Universidad de Buenos Aires
Facultad de Ciencias Exactas y Naturales
Departamento de Computación

Detecting clusters and boundaries: a twofold study on shape representation

Tesis presentada para optar al título de Doctor de la Universidad de Buenos Aires en el área Ciencias de la Computación

Mariano Tepper

Buenos Aires, 31 de Marzo de 2011

Comments

This work was done between March 2008 and March 2011 in the Departamento de Computación, Facultad de Ciencias Exactas y Naturales, Universidad de Buenos Aires, Argentina. During that time, I made a few academic visits to the Instituto de Ingeniería Eléctrica at the Facultad de Ingeniería, Universidad de la República, Uruguay, and the Laboratoire Traitement et Communication de l'Information (LTCI) at Telecom ParisTech, France. My advisors were Pablo Musé, Andrés Almansa and Marta Mejail.

The thesis board was composed by Alfred Bruckstein, Alvaro Pardo, Gregory Randall and Yann Gousseau.

Detecting clusters and boundaries: a twofold study on shape representation

Resumen

Las formas juegan un rol clave en nuestro sistema cognitivo: en la percepción de las formas yace el principio de la formación de conceptos. Siguiendo esta línea de pensamiento, la escuela de la Gestalt ha estudiado extensivamente la percepción de formas como el proceso de asir características estructurales encontradas o *impuestas sobre* el material de estímulo.En resumen, tenemos dos modelos de formas: pueden existir físicamente o ser un producto de nuestros procesos cognitivos.

El primer grupo está compuesto por formas que pueden ser definidas extrayendo los contornos de objetos sólidos. En este trabajo nos restringiremos al caso bidimensional. Decimos entonces que las formas del primer tipo son formas planares. Atacamos el problema de detectar y reconocer formas planares. Ciertas restricciones teóricas y prácticas nos llevan a definir una forma planar como cualquier pedazo de línea de nivel de una imagen.

Comenzamos por establecer que los métodos a contrario existentes para detectar líneas de nivel son usualmente muy restrictivos: una curva debe ser enteramente saliente para ser detectada. Esto se encuentra en clara contradicción con la observación de que pedazos de líneas de nivel coinciden con los contornos de los objetos. Por lo tanto proponemos una modificación en la que el algoritmo de detección es relajado, permitiendo la detección de curvas parcialmente salientes.

En un segundo acercamiento, estudiamos la interacción entre dos maneras diferentes de determinar la prominencia de una línea de nivel. Proponemos un esquema para competición de características donde el contraste y la regularidad compiten entre ellos, resultando en que solamente las líneas de nivel contrastadas y regulares son consderedas salientes.

Una tercera contribución es un algoritmo de limpieza que analiza líneas de nivel salientes, descartando los pedazos no salientes y conservando los salientes. Está basado en un algoritmo para detección de multisegmentos que fue extendido para trabajar con entradas periódicas.

Finalmente, proponemos un descriptor de formas para codificar las formas detectadas, basado en el Shape Context global. Cada línea de nivel es codificada usando shape contexts, generando así un nuevo descriptor semi-local. A continuación adaptamos un algoritmo existente de matching de formas a contrario para nuestro caso particular.

El segnudo grupo está compuesto por formas que no se corresponden con un objeto sólido, pero que están generadas por la integración de varios objetos sólidos. La formas más simples en este grupo son configuraciones de puntos en dos dimensiones. Las técnicas de agrupamiento pueden resultar útiles en estas situa-

ciones.

En un trabajo fundacional de 1971, Zahn trató el problema de encontrar agrupamientos perceptuales de acuerdo a la gestalt proximidad y propuso tres principiops básicos para algoritmos de clustering: (1) solamente importan las distancias entre puntos, (2) resultados estables entre diferentes ejecuciones e (3) independencia de la estrategia de exploración. Un tercer requerimiento implícito es crucial: los grupos pueden tener formas arbitrarias y un algoritmo para detectarlos debe ser capaz de lidiar con esto. En esta parte nos concentraremos en el diseño de algoritmos de agrupamiento que cumplan completamente los requemrimientos anteriores, imponiendo suposiciones mínimas sobre los datos a agrupar.

Comenzamos por analizar el problema de la validación de agrupamientos en una estructura jerárquica. Basándonos en métodos no-paramétricos para estimación de la densidad, proponemos calcular la prominencia de un determinado grupo. Luego, es posible elegir los grupos más salientes dentro de la jerarquía. En la práctica, el método muestra una preferencia por los grupos compactos y proponemos una simple heurística para corregir este tema.

En general, los métodos jerárquicos basados en grafos requieren calcular primero el grafo completo de distancias entre puntos. Por esta razón los métodos jerárquicos son considerados lentos. El más comúnmente utilizado, y el más rápido, algoritmo de entre ellos es el basado en el Árbol de Cubrimiento Mínimo (AGM). Proponemos por lo tanto un algoritmo para calcular el AGM evitando el paso intermedio de calcular el conjunto completo de distancias. Adicionalmente, el algoritmo puede ser fácilmente paralelizado. El método exhibe una buena performance para datos de baja dimensionalidad y permite un cálculo aproximado pero robusto en más altas dimensiones.

Finalmente proponemos un método para elegir subárboles agrupados dentro del AGM, mediante el cálculo de estadísticas de ejes simples. El método permite recuperar grupos con formas arbitrarias. También trabaja en situaciones ruidosas, donde el ruido es considerado como datos sin agrupar, permitiendo separarlos de los datos agrupados. También mostramos que la aplicación iterativa del algoritmo permite resolver un fenómeno llamado enmascaramiento, donde un grupo muy populoso impide la deteccion de otros menos populosos.

Palabras clave

Formas, líneas de nivel, agrupamientos, detección a contrario, árbol generador mínimo.

Detecting clusters and boundaries: a twofold study on shape representation

Abstract

Shape plays a key role in our cognitive system: in the perception of shape lies the beginning of concept formation. Following this lines of thought, the Gestalt school has extensively studied shape perception as the grasping of structural features found in or *imposed upon* the stimulus material. In summary, we have two models for shapes: they can exist physically or be a product of our cognitive processes.

The first group is formed by shapes that can be defined by extracting contours from solid objects. In this work we will restrict ourselves to the two dimensional case. Therefore we say that these shapes of the first type are planar shapes. We address the problem of detecting and recognizing planar shapes. A few theoretical and practical restrictions lead us to define a planar shape as any piece of meaningful level line of an image.

We begin by stating that previous a contrario methods to detect level lines are often too restrictive: a curve must be entirely salient to be detected. This is clearly in contradiction with the observation that *pieces* to level lines coincide with object boundaries. Therefore we propose a modification in which the detection criterion is relaxed by permitting the detection of partially salient level lines.

As a second approach, we study the interaction between two different ways of determining level line saliency: contrast and regularity. We propose a scheme for feature competition where contrast and regularity contend with each other, resulting in that only contrasted and regular level lines are considered salient.

A third contribution is a clean-up algorithm that analyses salient level lines, discarding the non-salient pieces and returning the salient ones. It is based on an algorithm for multisegment detection, which was extended to work with periodic inputs.

Finally, we propose a shape descriptor to encode the detected shapes, based on the global Shape Context. Each level line is encoded using shape contexts, thus generating a new semi-local descriptor. We then adapt an existing a contrario shape matching algorithm to our particular case.

The second group is composed by shapes that do not correspond to a solid object but are formed by integrating several solid objects. The simplest shapes in this group are arrangements of points in two dimensions. Clustering techniques might be helpful in these situations.

In a seminal work from 1971, Zahn faced the problem of finding perceptual clusters according to the proximity gestalt and proposed three basic principles for clustering algorithms: (1) only inter-point distances matter, (2) stable results across executions and (3) independence from the exploration strategy. A last im-

plicit requirement is crucial: clusters may have arbitrary shapes and detection algorithms must be capable of dealing with this. In this part we will focus on designing clustering methods that completely fulfils the aforementioned requirements and that impose minimal assumptions on the data to be clustered.

We begin by assessing the problem of validating clusters in a hierarchical structure. Based on nonparametric density estimation methods, we propose to compute the saliency of a given cluster. Then, it is possible to select the most salient clusters in the hierarchy. In practice, the method shows a preference toward compact clusters and we propose a simple heuristic to correct this issue.

In general, graph-based hierarchical methods require to first compute the complete graph of interpoint distances. For this reason, hierarchical methods are often considered slow. The most usually used, and the fastest hierarchical clustering algorithm is based on the Minimum Spanning Tree (MST). We therefore propose an algorithm to compute the MST while avoiding the intermediate step of computing the complete set of interpoint distances. Moreover, the algorithm can be fully parallelized with ease. The algorithm exhibits good performance for low-dimensional datasets and allows for an approximate but robust solution for higher dimensions.

Finally we propose a method to select clustered subtrees from the MST, by computing simple edge statistics. The method allows naturally to retrieve clusters with arbitrary shapes. It also works well in noisy situations, where noise is regarded as unclustered data, allowing to separate it from clustered data. We also show that the iterative application of the algorithm allows to solve a phenomenon called masking, where highly populated clusters avoid the detection less populated ones.

Keywords

Shapes, level lines, clusters, a contrario detection, minimum spanning tree.

Agradecimientos

Agradezco a los miembros del jurado, Alfred Bruckstein, Alvaro Pardo, Gregory Randall y Yann Gousseau, por aceptar evaluar mi trabajo. Sus comentarios y sugerencias tienen un gran valor para mí.

Este trabajo no hubiera sido posible sin la colaboración fundamental de muchas personas.

En primer lugar, quiero agradecer a Pablo Musé. Aunque, por motivos burocráticos que escaparon a nuestro control, Pablo no pudo ser oficialmente director de esta tesis, en la práctica lo fue. No hubiera podido llevar a buen término mi trabajo sin su constante apoyo y sin la riqueza, tanto científica como humana, de nuestras discusiones semanales a lo largo de estos tres años.

También quiero expresar mi reconocimiento para con Andrés Almansa, quien no deja de causarme admiración no sólo por su capacidad para generar ideas originales sino también por su calidad humana.

Quiero agradecer también a Marta Mejail por su confianza, afecto y amistad. Junto a ella conocí y aprendí a querer el procesamiento de imágenes.

Aunque ya lo mencioné como jurado quiero volver a resaltar la persona de Gregory Randall. En 2004, estaba mirando entre la lista de cursos de la Escuela de Informática (ECI) y me encontré con uno llamado “introducción a la visión por computadora”. Me llamó la atención el título y decidí anotarme. Gregory dictaba ese curso, que motivó mi interés en la imágenes. Gregory fue también el alma máter del programa ALFA “Computer Vision Foundations and Applications”, que me permitió vivir un año en Paris mientras me formaba profesionalmente. Para todos los que lo conocemos, Gregory es el compás que marca el “Sur”.

Un gracias enorme a mis padres, que siempre me dieron, desde el corazón, cuanto podían para que me desarrolle libremente. Gracias también a mis amigos tanto a los que siempre me apoyaron como a los que siempre me preguntan “cuando te vas a poner a laburar?”. Finalmente, gracias a Luciana que, con su apoyo constante, fue mi luz en los momentos más oscuros.

Contents

In looking to an object we reach out for it. With an invisible finger we move through the space around us, go out to the distant places where things are found, touch them, catch them, scan their surfaces, trace their borders, explore their texture. [...] Thus a tangible bridge is established between the observer and the observed thing, and over this bridge the impulses of light that emanate from the object travel to the eyes and thereby to the soul.

Arnheim, Visual Thinking [3], pp. 19.

CHAPTER

1 Introduction

Shape plays a key role in our cognitive system: in the perception of shape lies the beginning of concept formation.

Artists have implicitly acknowledged the importance of shapes since the dawn of times. Indeed, despite that lines do not divide objects from their backgrounds in the real world, line drawings are present in much of our earliest recorded art and, remarkably, remained unchanged through history, see Figure 1.1a. After the rediscovery of the ancient Greek's culture, Renaissance's artists used specially proportioned shapes (e.g. golden and harmonic proportions) to make their works more visually appealing, see Figure 1.1b. In the last century, many artists decided to explicitly exploit the powerful place of shape within our perception, giving birth to currents like cubism and abstract art.

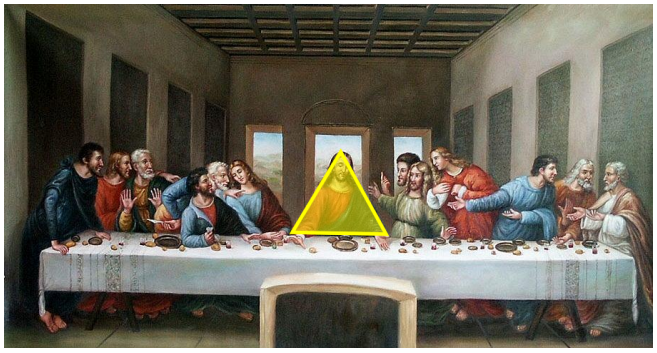
Although art may provide clues to understand shape perception, it tells us little from the formal point of view. Let us begin by defining what is a shape.

Phenomenologists [5] conceive shape as a subset of an image, digital or perceptual, endowed with some qualities permitting its recognition. In this sense, both concepts, shape and recognition, are intrinsically intertwined: one has to define what is a shape in such a way that its recognition can be performed.

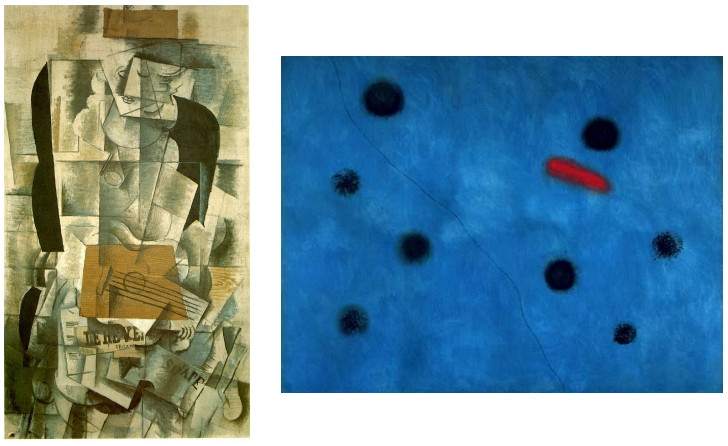
However shapes can also take many forms: they can exist physically as in Figure 1.2a, where the letters have real contours, or be a product of our cognitive processes as in Figure 1.2b, where we interpolate a contour from individual points. Here, the word "interpolate" is the key: we perceive both shapes as equivalent when, strictly speaking, they are not. In some way, our perception fills the gaps between the points to create an holistic percept.

Following this lines of thought, gestaltists [3] regard shape perception as the grasping of structural features found in or imposed upon the stimulus material. The Gestalt school has extensively studied phenomena that unveil and justify this definition. Kanizsa's triangle [70], see Figure 1.3a is a famous example where we see a triangle even if there are not three intersecting and nonparallel segments. In the second example, depicted in Figure 1.3b, dots are arranged in circular patterns

(a) Lines are used to convey the outer contours of the horses in a very similar way in these drawings, one from 15,000 BC (left: Chinese Horse, paleolithic cave painting at Lascaux, France) and the other from AD 1300 (right: Jen Jen-fa, detail from The Lean Horse and the Fat Horse, Peking Museum, China). Reprinted by permission from Macmillan Ltd: NATURE [29], copyright 2005.



(b) The triangle in Renaissance's composition depicted in Leonardo da Vinci's The Last Supper.

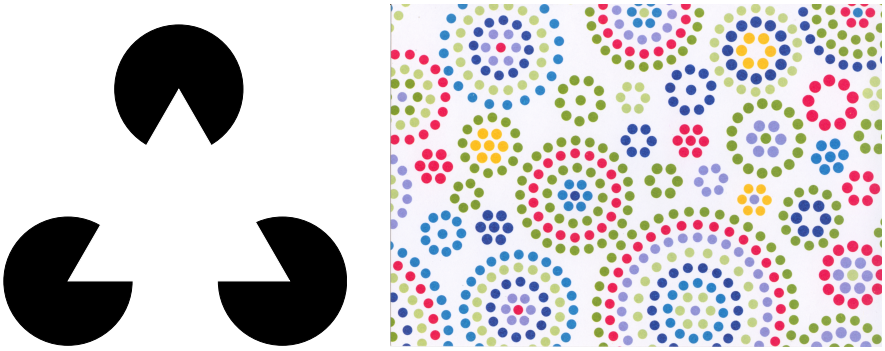


(c) In the 20-th century, shapes were made explicit, as in Georges Bracques's Woman with a Guitar and a painting from Joan Miró's Bleu Series.

Figure 1.1: Artist have long acknowledged the central and complex role of shapes in perception.

(a) (b)

Figure 1.2: Humans have no trouble for recognizing the Coca-Cola logo from points sampled along its contour. The edge detection / descriptor encoding / descriptor matching framework, nowadays classical in computer vision, may fail in such an easy task.



(a) (b)

Figure 1.3: (a) Kanizsa's triangle and (b) a dotted texture. Shape is perceived by a grouping phenomenon called amodal completion.

and the colour is used to enforce this effect. This phenomena were all grouped by the Gestalt theory into a cognitive process called amodal completion.

In summary, we have two models for shapes. We have, first, intrinsically defined shapes and, second, extrinsically defined ones.

The first group is formed by shapes that can be defined by extracting contours from solid objects. In this context, shapes are represented and analyzed from an infinite-dimensional approach in which a shape is the locus of an infinite number of points [78]. This point of view leads to the active contours formulation [74] or to level-sets methods [121]. Although these shapes can be defined in any number of dimensions, e.g. the contour of a three dimensional solid object is a surface, we will restrict ourselves to the two dimensional case. Therefore we say that these shapes of the first type are planar shapes.

The second group is composed by shapes that do not correspond to a solid object but are formed by integrating several solid objects. The simplest shapes in this group are arrangements of points in two dimensions. Shapes are then approximated by a finite-dimensional representation (a set of landmarks or samples), on

which various transformations may act to account for variability and to subsequently derive models [78]. This approach conducted to the study and analysis of manifolds [24, 48]

1.1 Planar Shapes

Planar shape recognition is one of the most active fields in computer vision and digital image processing. But what is shape recognition? It can be stated schematically that shape recognition is the ability to recognize that two shapes seem to be similar. Obviously, the previous sentence has many ambiguities that have to be resolved before answering the above question. The questions that naturally arise when tackling the problem are: What makes a shape to be similar to another? and finally, what is recognition?

Let us begin by the last question. It can be argued that recognition (re+cognition) is the process of awareness that occurs in thinking when some event, process, pattern, or object is identified as recurring. Thus in order for something to be recognized, it must be familiar. When the recognizer has correctly responded, this is a measure of understanding.

This definition opens some paths that lead to interesting concepts. From one side, it introduces the need for some notion of repetition (maybe not a strict but a relaxed one), that can be used to assess the recurrence of elements. From the other, it follows that some *a priori* knowledge is needed. It also gives a hint about the importance of recognition as a cognitive process for "further" high-level cognitive tasks.

Returning to the need of some *a priori* perceptual knowledge, it is a fact that it is not necessary to retain the totality of a shape with all its details for its later recognition, but only those qualities permitting its recognition:

> It appears likely that a major function of the perceptual machinery is to strip away some of the redundancy of stimulus, to describe or encode incoming information in a form more economical than that in which it impinges on the receptors. (Attneave [5], 1954)

Finding the right set of shape qualities is therefore crucial, and adding a minimality restriction is a major concern to achieve both *computational* and *memory* economy.

Now that we have introduced some conceptual principles about shape recognition, we can detail the perturbations that should not affect its realization. For the sake of completeness, we paraphrase the argumentative line in [85, 22], that leads to a set of requirements that have to be fulfilled by any shape representation, in order to be compatible with the recogintion process.

Contrast changes. According to the gestaltists Attneave and Wertheimer, shape perception is independent of the gray scale or of the measured colours.

> The concentration of information in contours is illustrated by the remarkable similar appearance of objects alike in contour and

> different otherwise. The "same" triangle, for example, may be either white on black or green on white. Even more impressive is the familiar fact that an artist's sketch, in which lines are substituted for sharp color gradients, may constitute a readily identifiable representation of a person or thing. (Attneave [5], 1954).
>
> I stand at the window and see a house, trees, sky. Theoretically I might say there were 327 brightnesses and nuances of colour. Do I have "327"? No. I have sky, house, and trees. It is impossible to achieve "327" as such. And yet even though such droll calculation were possible and implied, say, for the house 120, the trees 90, the sky 117 - I should at least have this arrangement and division of the total, and not, say, 127 and 100 and 100; or 150 and 177. (Wertheimer [138], 1938).

Occlusions and background modification. Humans are able to recognize shapes despite the presence of occlusions or differences in the background.

Noise and smoothing. These are inherent to any perceptual task and to any image generated according to Shannon's theory.

> It appears, then, that when some portion of the visual field contains a quantity of information grossly in excess of the observer's perceptual capacity, he treats those components of information which do not have redundant representation somewhat as a statistician treats "error variance", averaging out particulars and abstracting certain statistical homogeneities. (Attneave [5], 1954)

Geometric distortions and deformations. Human perception is constantly dealing with perspective effects and any automatic system must take them into account.

All these restrictions limit the possibilities for the choice of the elements of an image that form a planar shape. From these restrictions, a set of formal requirements for any planar shape representation can be derived:

Contrast invariance. We define an image as a function $u(x)$ where $u(x)$ represents the gray level or luminance in x. Our first task is to extract the topological information of an image, independent of the unknown contrast change function of the acquisition system. This function can be modeled as a continuous and growing function g. The observed data of an image u might be any $g(u)$. This simple argument leads to select the level sets [121], or level lines, as a contrast invariant image description [28]. The set of level lines is called the topographic map of an image.

Concentration of Information. The previous requirement leads us to define the set of level lines as a complete and contrast invariant image representation. Maybe in a contradictory way, many authors, like Attneave, argue that "information is concentrated along contours (regions where contrast changes abruptly)". This means that not all level lines are necessary to obtain a perceptually complete description. We call the selected lines, meaningful level lines. The selection uses an *a contrario* model, following gestaltic principles.

Occlusion and background-figure. Even the most adapted selection of level lines is not totally useful to describe shapes. When a shape A occludes partially another shape B, the level lines of the resulting image are in fact a concatenation of pieces of the level lines of A and B. It is therefore mandatory to cut or to dissect the level lines in pieces that allow to overcome this kind of situations.

Smoothing. During the acquisition, details much too fine to be perceptually relevant are introduced. It is necessary to use a suitable filtering mechanism. This not necessarily means that the input shape must be smoothed. From our point of view invariance to fine details must be handled by a subsequent suitable description and not by the planar shape detection process. We will show that this is indeed feasible.

Geometrical invariance. Representations must be invariant to weak projective transformations. It can be shown that all planar curves within a large class can be mapped arbitrarily close to a circle by projective transformations [4]. Moreover, full projective invariance is nor perceptually real (humans have great difficulties to recognize objects under strong perspective effects) nor computationally tractable. In this sense, affine invariance is the most we can impose in practice. At the same time, the effect of any optical acquisition system can be modeled by a convolution with a smoothing radial kernel. It does not commute with projective transformations and must be taken into account in the recognition process. A multiscale analysis is the only feasible way to treat it correctly. Both concepts, affine invariance and multiscale analysis must be consistently integrated.

According to the above concepts, we are now able to formally define a shape in the following way:

Definition 1. *We call planar shape of an image any piece of meaningful level line of an image.*

The last among these requirements, namely geometrical invariance, will not be covered in this work. Affine invariance is usually handled by proposing affine invariant descriptors [22]. Since acquisition smoothing does not commute with projective transformations we consider such an approach flawed by principle. A method for simulating affine transformations has been recently proposed [99], permitting to achieve true affine invariance while using similarity invariant representations. All the presented methods for planar shape detection/recognition can be directly embedded in a procedure where affine parameters (including scale) are simulated.

1.1.1 Planar Shape Proposal

In this work we address the problem of detecting and recognizing planar shapes.

We begin by stating that previous a contrario methods to detect level lines are often too restrictive. A curve must be entirely salient to be detected. This is clearly

in contradiction with the aforementioned observation that *pieces* of level lines coincide with object boundaries. Therefore we propose a modification in which the detection algorithm is relaxed by permitting the detection of partially salient level lines.

As a second approach, we study the interaction between two different ways of determining level line saliency: contrast and regularity. In general a contrasted level line will be regular and vice versa. Previous ways to combine both features have problems which we correct by proposing a scheme for feature competition. Contrast and regularity contend with each other, resulting in that only contrasted and regular level lines are considered salient.

A third contribution is a clean-up algorithm that analyses salient level lines, discarding the non-salient pieces and returning the salient ones. It is based on an algorithm for multisegment detection [61], which was extended to work with periodic inputs.

Finally, we propose a new shape descriptor to encode the detected shapes. It is based on a global (with respect to the image) shape descriptor called Shape Context [12]. Each level line is encoded using shape contexts, thus generating a new semi-local descriptor which we call Morphological Shape Context. We then propose an adapted version of the matching algorithm by Musé et al. [106] suitable for Morphological Shape Contexts.

1.2 Clusters as shapes

Manifolds are classically represented by point clouds, i.e. by samples. Nowadays, three-dimensional shape acquisition devices such as laser range scanners have become a popular source of point cloud generation. These scanners provide in general raw data in the form of (noisy) unorganized point clouds representing surface samples [92]. This source of data is becoming increasingly popular, creating new and broad applications. Hence, this representation must be tackled directly, without the need of the sometimes cumbersome and distorting intermediate steps of surface reconstruction.

Figure 1.4 represents an example shapes represented by point clouds. In this case, the point cloud is the result of a three-dimensional reconstruction from multiple views.

A classical assumption in manifold learning is that points lie on a single manifold with an intrinsic dimensionality lower than the one of the original space. However in many cases such assumption is clearly not true. For example, in Figure 1.4 points lie on two different surfaces, i.e. the hand and the head. In general terms, the single manifold assumption is not easy to verify.

Clustering techniques might be helpful at solving these issues as data can be partitioned with different rules in such a way that each set on the partition respects our assumptions. But to which clustering methods should we turn our attention to?

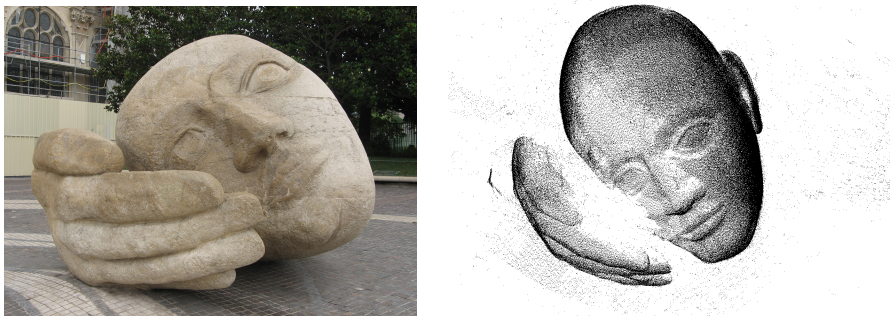


Figure 1.4: Example of shapes represented by point clouds. Reproduced from http://cmp.felk.cvut.cz/projects/is3d/#Pub1.

Over the years, many clustering methods have been proposed [67]. We are only interested in methods for clustering metric datasets. Non-numerical datasets are out of the scope of this work.

As we intend to cluster datasets in metric spaces, we must solely rely on the given metric. At this point we must make perfectly clear that the distance is chosen by the user of the clustering method. The user must not be forced to adapt the problem to the clustering methods but in opposition, adapt the clustering methods to the problem to solve.

In particular we are not interested on methods that rely on assumptions on the characteristics of the space to cluster. In other words and as an example, a method designed to work in an Euclidean space should not be used until we are sure that our data actually lives in this space! These kind of verifications are problematic and do not conduct to an unifying clustering paradigm.

Figure 1.5 illustrate the negative effect of using inappropriate metrics to analyze data. In this case, the metric is induced by the method to project a sphere on a plane. If we need visually accurate areas, the Mercator projection must be replaced by an equal-area projection .

In a seminal work from 1971, Zahn [143] faced the problem of finding perceptual clusters according to the proximity gestalt and proposed three basic principles for clustering algorithms.

Only inter-point distances matter. No extra information must be used apart from the distance between points.

Stable results. Results must remain stable for all runs of the detection process. Once its parameters fixed, an algorithm must not yield completely different results in different executions.

Independence from the exploration strategy. The order in which points are analyzed must not affect the outcome of the algorithm. In other words, a permutation of the dataset should not affect the results.

These principles form the conceptual grounds on which our work is based and impose certain restrictions to the type of algorithms we look for:

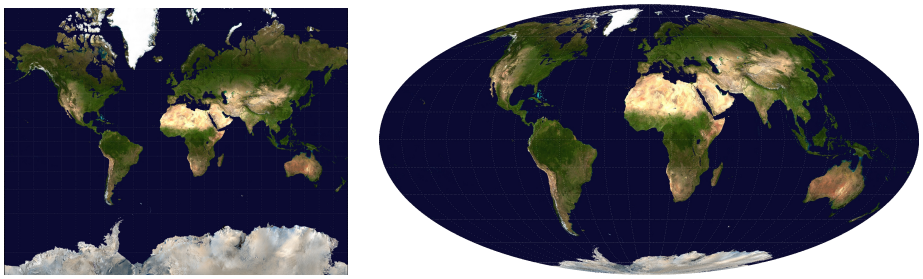


Figure 1.5: Using an inadequate metric might induce errors. The Mercator projection projection (left) gives the impression that Africa and Greenland have about the same area. The equal-area Mollweide projection (right) corrects the aberration.

Only inter-point distances matter. The natural algorithmic structure for modeling points and distances between points is a weighted graph where edge weights represent distances. Thus, according to this constraint, graphs are the only suitable underlying structure for clustering.

No random steps. Many clustering algorithms converge to a local minimum of a criterion of the partition badness. By initializing the algorithm with random seeds, one expects to have higher probability of finding the global minimum. As expected, random initializations are directly reflected on the obtained clustering. Although one could imagine a randomly initialized algorithm that yields stable results, this is hard in practice. We therefore desire algorithms not based on such initializations.

Independence from the exploration strategy. This requirement is the most simple to fulfill and most modern clustering algorithms do.

A last requirement is implicitly stated in our discussion: clusters may have arbitrary shapes and detection algorithms must be capable of dealing with this. A whole family of clustering algorithms that detect only elliptical clusters must be then discarded.

There are two main trends in clustering algorithms: partitional and hierarchical. Partitional methods aim at finding a partition that optimizes a certain criterion. On the other side, hierarchical methods produce a hierarchy of nested clusters. This hierarchy is useful for applications where a taxonomy is interesting; when it is not the case, a process to select clusters from the hierarchy must be used. Both, partitional and hierarchical, methods need a validation step which studies the quality and pertinence of the result.

1.2.1 Clustering Proposal

In this part we will focus on designing a clustering method that completely fulfills the aforementioned requirements and that impose minimal assumptions on the data to be clustered.

We begin by assessing the problem of validating clusters in a hierarchical structure. Based on nonparametric density estimation methods, we propose to compute the saliency of a given cluster. Then, it is possible to select the most salient clusters in the hierarchy. In practice, the method shows a preference toward compact clusters and we propose a simple heuristic to correct this issue.

In general, graph-based hierarchical methods require to first compute the complete graph of interpoint distances. For this reason, hierarchical methods are often considered slow. The most usually used, and the fastest hierarchical clustering algorithm is based on the Minimum Spanning Tree (MST). We therefore propose an algorithm to compute the MST while avoiding the intermediate step of computing the complete set of interpoint distances. This is achieved with a clever use of nearest neighbors search structures. Moreover, the algorithm can be fully parallelized with ease. The algorithm exhibits good performance for low-dimensional datasets and allows for an approximate but robust solution for higher dimensions.

Finally we propose a method to select clustered subtrees from the MST, by computing simple edge statistics. The method allows to retrieve clusters with arbitrary shapes. It also works well in noisy situations, where noise is regarded as unclustered data, allowing to separate it from clustered data. We also show that the iterative application of the algorithm allows to solve a phenomenon called masking, where highly populated clusters avoid the detection of less populated ones.

1.3 Overview of this thesis

The dissertation is organized in two main parts. The first part deals with the extraction of shape information from images. We present methods for extracting, encoding and matching planar shapes. The second part is focused on the analysis and detection of shapes described by point clouds. This representation naturally leads to facing clustering problems.

Part I: A study on Planar Shapes

This part (Chapter 2 to Chapter 4) is focused on planar shapes on images and covers the following topics: shape extraction, invariant encoding of shapes, shape matching and matching decision.

Chapter 2: Planar Shape Review We present surveys methods to detect, encode and match shapes in images. We describe the weaknesses and strengths of frequently used algorithms.

Chapter 3: Shape Extraction Here a method is proposed to select the most meaningful level lines based on perceptual principles. The proposed method aims at improving the original method proposed in [39]. It also improves some building

blocks of the method proposed in [23]. In short words, we detect curves that are only partially salient. A model to combine the "contrast" and the "good continuation" partial gestalts is also introduced; in this model both features compete thus yielding robust detections. We also propose a modification to the multisegment detection algorithm in [61] that allows to work with periodic sequences. This algorithm permits to roughly extract all pieces of level lines of an image, that coincide with pieces of edges.

The work presented in this chapter (joint work with P. Musé, A. Almansa and M. Mejail) will soon be submitted and available as a preprint.

Chapter 4: Shape Encoding and Matching This chapter is devoted to presenting a new shape recognition algorithm. It encodes level lines by using the shape context descriptor; we call this combination Morphological Shape Context (MSC). Then an a contrario formulation based on the framework in [106] for matching MSC is presented.

The a contrario matching approach proposed in this chapter was presented in ICIP 2009 [132] and more extensively in CIARP 2009 [133].

Part II: The proximity gestalt: a computational quest

This part (Chapter 5 to Chapter 8) is devoted to the clustering problem. Three algorithms are presented: two clustering methods and a technique to efficiently compute the MST.

Chapter 5: Clustering Review We describe different clustering algorithms based on different approaches: partitional, hierarchical, graph-based. Then we review some solutions to the often underestimated problem of validating a clustering. We point out the advantages and drawbacks of these methods.

Chapter 6: Clustering using graph-based density estimation We present an a contrario clustering algorithm that is a combination of hierarchical and density-estimation techniques that does not suffer from an a priori imposition of the cluster shapes. In this sense it extends the original method in [21]. Even if we do not impose cluster shapes a priori, the formulation results in an algorithm well suited for detecting compact clusters.

The a contrario clustering algorithm proposed in this chapter has been accepted for publication in Pattern Recognition.

Chapter 7: Efficient Minimum Spanning Tree An algorithm to compute the MST in metric datasets is proposed. Classically, MST-based clustering algorithms are considered too slow for large datasets because the distances between all points in the dataset must be computed. Consequently, they are discarded or patched by constraining the input graph. This patch may have unwanted consequences in the result that cannot be predicted in advance. The proposed algorithm does not

need to compute all distances. We show that for low-dimensional datasets, the algorithm is faster than classical MST algorithms. For datasets of higher dimensions, the algorithm allows to efficiently compute an approximate solution.

The proposed MST algorithm has been submitted to the Journal of Machine Learning Research and is available as a preprint at http://hal.archives-ouvertes.fr/hal-00583120/en/.

Chapter 8: Clustering using MST statistics The density estimation techniques in the algorithm from Chapter 6 leads to a computationally slow algorithm. In this chapter we present a second a contrario clustering algorithm, based on the MST. This formulation leads to an efficient algorithm that allows to recover arbitrarily shaped clusters. Classically the a contrario validation methods are presented with a set of candidates to test which among them are clusters. We present a new model in which the a contrario validation of clusters is used to create the candidate set, thus robustifying the detection process. We also illustrate a phenomenon called masking, were a highly populated cluster avoids from detecting other less populated clusters. We show that masking can be unveiled by iterating the clustering procedure.

The clustering algorithm proposed in this chapter has been submitted to the International Journal of Computer Vision and is available as a preprint at http://arxiv.org/abs/1104.0651.

Chapter 9 We present some conclusions, as well as perspectives and future work.

Part I

A study on Planar Shapes

CHAPTER

2 Planar Shape Review

Abstract

In the present chapter we briefly review different approaches to the problem of shape recognition, presenting methods to extract, encode and match planar shapes. Among shape extraction methods, we focus on the ones that provide closed curves representations, i.e. active contours, level sets and MSER. For the other stages (encoding and matching), we limit ourselves to present the main trends.

2.1 Shape Detection

In few words, shape detection is the process of extracting salient object contours from images. These contours take the form of closed Jordan Curves and there are a few different approaches devoted to extracting them, but all methods in one way or another, rely on analyzing the image gradient along the curve.

2.1.1 Active Contours

The Active Contour theory, a.k.a. snakes, proposes a variational approach for edge detection [74, 16]. Following Desolneux et al. [40] we will concentrate on the formulation by Kimmel and Bruckstein [76]. Snakes are image curves with optimal contrast and smoothness energy. Let $\gamma(s)$ be an image curve parameterized by its length $L(\gamma)$. The energy to maximize is

$$F(\gamma) = \frac{1}{L(\gamma)} \int_0^{L(\gamma)} g\left(Du(\gamma(s)) \cdot \gamma'(s)^{\perp}\right) ds \tag{2.1}$$

where Du is the image gradient, $\gamma'(s)^{\perp}$ is a unit vector normal to the curve and $g > 0$ is an even contrast function. When $g(\alpha) = \alpha$, the above functional is related with the Marr-Hildreth edge detector [87].

In this formulation, the curve smoothness is not made explicit but is nevertheless present: it is hidden in the function g. It can be shown [40] that if g has linear

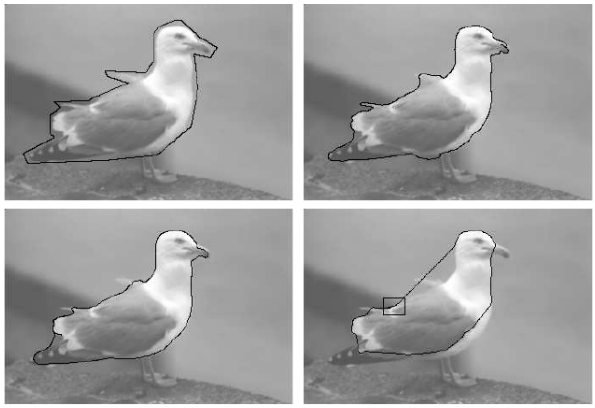


Figure 2.1: Contour optimization in function of g. An initial contour (top left) was first drawn by hand. It was optimized by the snake model for different functions g: $g(\alpha) = |\alpha|^{0.5}$ (top right), $g(\alpha) = |\alpha|$ (bottom left) and $g(\alpha) = |\alpha|^3$ (bottom right). As the power increases, the snake becomes less sensitive to low contrast edges and tends to smooth them or even to create straight shortcuts. Reproduced from [40]

or supra-linear growth, then the snake will tend to abandon the low contrasted parts of the contour.

2.1.2 The Topographic Map

Given an image u, the upper level set $\mathcal{X}_\lambda$ and the lower level set $\mathcal{X}^\lambda$ of value λ are subsets of $\mathbb{R}^2$ defined by

$$\mathcal{X}_\lambda = \{x \in \mathbb{R}^2 \mid u(x) \geq \lambda\} \tag{2.2}$$

$$\mathcal{X}^\lambda = \{x \in \mathbb{R}^2 \mid u(x) < \lambda\} \tag{2.3}$$

The set of upper (or lower) level sets of an image is sufficient to reconstruct it [90].

We define the boundaries of the connected components of a level set as a level line. Monasse [98] developed an efficient method to compute level lines by bilinear interpolation. These level lines have the following properties:

- level lines are closed Jordan Curves;
- level lines at different levels do not meet;
- by topological inclusion, level lines form a partially ordered set.

We call the set of level lines (along with their level) a topographic map. A tree structure is induced by the partial ordering in the topographic map.

The Mathematical Morphology school [90, 121] has extensively studied the topographic map and its level sets producing a hole set of tools for image analysis. Smoothing filters usually described by using Partial Differential Equations (PDE) can be proven to have equivalent formulation in terms of iterated morphological operators [63]. Hence, edge detectors can then be directly expressed by combining these operators.

Exploiting the tree structure of the topographic map, Monasse proposed an algorithm to select its contrasted level lines [97]. In the next chapter we will examine in depth the detection of perceptually significant level lines.

2.1.3 Maximally Stable Extremal Regions

There is a wide literature about Maximally Stable Extremal Regions (MSER). This method was introduced by Matas et al. [89] and has rapidly gained popularity in the computer vision field.

An image I is a discrete function $I : \mathcal{D} \subset \mathbb{Z}^2 \to \mathcal{S}$. Extremal regions are well defined on images if:

1. $\mathcal{S}$ is totally ordered
2. an adjacency relation $A \subset \mathcal{D} \times \mathcal{D}$ is defined.

Definition 2. *A region $\mathcal{Q}$ is a contiguous subset of $\mathcal{D}$, according to relation A, i.e. $\forall p, q \in \mathcal{Q}$ there exists a sequence $p, a_1, a_2, \ldots, a_n, q$, where $a_i \in \mathcal{Q}$, such that $A(p, a_1), \ldots, A(a_i, a_{i+1}), \ldots, A(a_n, q)$.*

The regions can be extracted with the algorithm by Najman and Couprie [108] that runs on quasi-linear time.

Definition 3. *Let $\mathcal{Q}$ be a region and let $p, q \in \mathcal{D}$ be respectively two points in the inner boundary and in the outer boundary of $\mathcal{Q}$, i.e. $\forall p \in \mathcal{Q}$ and $\forall q \notin \mathcal{Q}$ such that $A(p, q)$. The region $\mathcal{Q}$ is extremal if $\forall p, q,\ I(p) > I(q)$ (maximum intensity region) or $\forall p, q,\ I(p) < I(q)$ (minimum intensity region).*

Definition 4 (MSER)**.** *Let $\mathcal{Q}_1, \ldots, \mathcal{Q}_i, \mathcal{Q}_{i+1}, \ldots$ be a sequence of nested extremal regions, i.e. $\mathcal{Q}_i \subset \mathcal{Q}_{i+1}$. An extremal region $\mathcal{Q}_{i^*}$ is called maximally stable if and only if $q(i) = |\mathcal{Q}_{i+\Delta} \setminus \mathcal{Q}_{i-\Delta}| / |\mathcal{Q}_i|$ has a local minimum at i^*.*

From our point of view, extremal regions are non other than zero-order interpolated level sets. Maximal (minimal) extremal regions correspond to the upper (lower) level sets of an image. Maximal stability can be regarded as an alternative way of keeping contrasted level lines. However the algorithm has to be run twice to obtain both upper and lower MSER.

Cao et al. [22] suggested to directly extract MSER from the bilinearly interpolated topographic map and Gómez Fernández [59] proposed a detailed and fast implementation. Gómez Fernández also performed an experimental comparison of MSER boundaries with meaningful boundaries. To distinguish this formulation from classical MSER we will refer to it as Maximally Stable Shapes (MSS) and to their boundaries as Maximally Stable Boundaries (MSB).

Fast algorithms have been developed to calculate extremal regions. In [89] Matas et al. proposed an algorithm that has a $O(n \log\log n)$ complexity, and argued that $O(n\,\alpha(n))$ can be achieved. This last version was implemented in [103], along with other optimizations.

Figure 2.2: Comparison of MSER (upper and lower), maximally stable shapes, and meaningful boundaries.

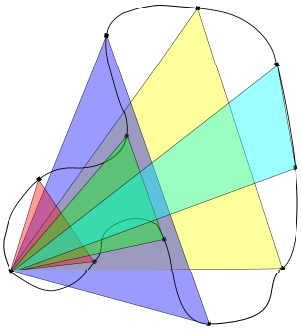


Figure 2.3: The shape can be approximated by using triangles whose (normalized) area is affine invariant.

Much research has been done using MSER. Donoser and Bischof [42] propose an algorithm for tracking MSERs in a video sequence. They use several features extracted from the MSER (e.g. center of mass, mean gray value and size). Forssén and Lowe [53] also use MSERs in combination with SIFT. Obdrzalek et al. [112] build affine invariant frames from geometrical affine invariants extracted from MSER. Sivic and Zisserman [126] use centers of mass of MSERs as stable keypoints for describing video frames. Matching is performed using SIFT descriptors in affine invariant regions under a text retrieval approach. This work has inspired Nistér and Stewénius who proposed a similar framework [111].

2.2 Shape Encoding

Shen et al. [123, 122] proposed an affine invariant shape encoding by using triangles whose vertices are points sampled along the shape contour, see Figure 2.3. To achieve full affine invariance, the process to sample the points must also be affine invariant.

A curve γ of length L can be parameterized by its arc length by defining

$$\gamma(s) = \frac{\int_0^s \left(x'(s)^2 + y'(s)^2\right)^{1/2}}{\int_0^L \left(x'(s)^2 + y'(s)^2\right)^{1/2}}, \tag{2.4}$$

where x' and y' are the derivatives in the horizontal and vertical direction, respectively. To achieve an affine invariant parametrization [95], arc length is usually replaced by affine length which has the following definition:

$$\gamma(s) = \frac{\int_0^s \left(x'(s)y''(s) + x''(s)y'(s)\right)^{1/3}}{\int_0^L \left(x'(s)y''(s) + x''(s)y'(s)\right)^{1/3}}, \tag{2.5}$$

where x'' and y'' are the second derivatives in the horizontal and vertical direction, respectively. The main disadvantage of this formulation is that high order derivatives are required for its computation. In practice these derivatives are avoided by convolving with the derivative of the Gaussian kernel.

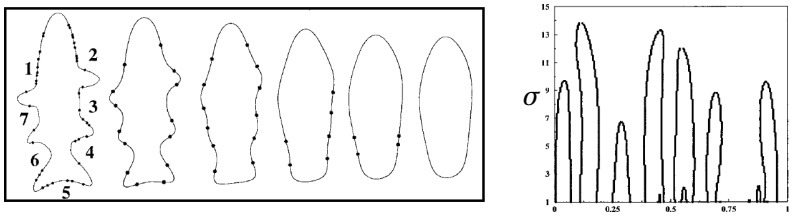


Figure 2.4: Left, as the curve is smoothed zero crossings of the curvature tend to disappear. Right, the evolution of these points is then analyzed as a shape descriptor (σ is the scale parameter).

Mokhtarian [96, 95] described shapes by applying a scale-space analysis to the curve and observing the scales at which zero crossing of the curvature disappear, see Figure 2.4. By changing the parametrization and the curvature definition accordingly one can achieve similarity invariance [96] or affine invariance [95]. As a side note, there is a more stable scheme, that does not involve any derivative computation, for affine curvature smoothing proposed by Moisan [94].

Zernike Moments [75] have long been used for shape description. For example, the method was included as a part of the MPEG-7 experiment on shape description [80]. These moments only provide a global description. More recently, different approaches [82, 118] renewed the interest in these descriptors by adding new features to them (e.g. rotation invariance). Cura et al. [38] showed that Zernike Moments have strong linear dependencies which can be corrected by PCA and a suitable weighting scheme.

Extensive research has been developed around the shape context method [12, 100]. The method will be explained in Chapter 4. For now it suffices to say that a shape context is a histogram of relative contour point positions with origin at another contour point. A histogram is built for each contour point. Mori et al. [100] also use the edge orientation to enrich the descriptor. In its basic definition, the contour is simply the output of an edge detection scheme (i.e. Canny's [20]). It is extremely hard to chain such contour points to define a closed curve. For this reason, the shape context is a global descriptor and it has to be patched to cope with cluttered scenes. For example, for recognizing CAPTCHAs[1] rectangular observation windows are used [101].

The relative positions of the contour points are usually computed using the Euclidean distance although other distances can be used. In order to achieve invariance to shape articulations, the inner distance has been proposed [84]. The inner distance is the length of the shortest path between two points in a given curve which does not cross the curve, see Figure 2.5. Although in certain situations these distance may be useful, in general we may want to distinguish 'I' shaped curves

[1]The Completely Automated Public Turing test to tell Computers and Humans Apart (CAPTCHA) is a type of challenge-response test used in computing to ensure that the response is not generated by a computer.

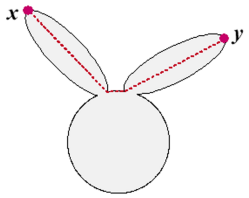
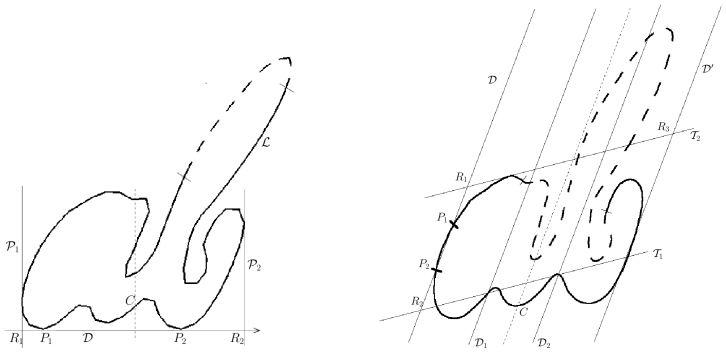


Figure 2.5: The inner distance can provide robustness to changes in the orientation of the "ears".



(a) Similarity invariant LLD (b) Affine invariant LLD

Figure 2.6: Local invariant frames can be built from the curve.

from 'L' shaped ones.

Mikolajczyk [93] proposed a SIFT-like shape descriptor [86]. In simple terms, it builds a local weighted histogram of contour point orientations. It can be regarded as a local shape context with additional angular information.

Lisani et al. [85] introduced a shape encoding algorithm known as Level Line Descriptor (LLD). They used obviously level lines as the shapes to encode. A small step of affine curvature smoothing allows to filter acquisition noise. In order to build invariant representations (up to either similarity or affine transformations), they define local frames for each level line, based on robust directions (tangent lines at flat pieces or bitangent lines). Figure 2.6 illustrate how to extract a similarity invariant frame from a bitangent and an affine invariant frame from a flat piece.

Obdrzálek and Matas [112] proposed a plethora of primitives for building affine invariant frames from MSER. Then, the image patch defined by each frame is encoded using a Discrete Cosine Transform (DCT). In this sense this is a hybrid shape/texture descriptor.

All presented affine invariant methods share the same problem: acquisition blur is not affine invariant. The convolution with a smoothing radial kernel does not commute with projective transformations and in particular the topology of an

image is affected by affine transformations (even those with unit determinant). For example, scale invariance has been solved by performing a multiscale analysis in a Gaussian scale-space [140, 139, 83]. The same approach was considered computationally prohibitive for all affine parameters. Recently, Morel and Yu [99] showed that affine invariance can also be obtained by simulations and proposed a numerical scheme that renders the problem tractable[2]. In short terms, this breakthrough allows to use good similarity invariant descriptors while in practice achieving affine invariant encoding.

2.3 Shape Matching

The decision step of whether two descriptors should be matched or not is the least studied of all the processes involved in visual recognition.

Let $\mathcal{F} = \{F^k \mid 1 \le k \le M\}$ be a database of M shapes. For each shape $F^k \in \mathcal{F}$ we have a set $\mathcal{T}^k = \left\{t_j^k \mid 1 \le j \le n_k\right\}$ where n_k is the number of points in the shape. Let $SC_{t_j^k}$ be the shape context of t_j^k, $1 \le j \le n_k$, $1 \le k \le M$. For simplicity, we denote

$$\mathcal{S} = \left\{SC_{t_j^k} \mid t_j^k \in \mathcal{T}^k,\ F^k \in \mathcal{F}\right\}. \tag{2.6}$$

Let us also suppose that we have a suitable distance $d(\cdot,\cdot)$ between Shape Contexts. We follow the choice made by Belongie et al. [12] of the χ^2 test statistic to compare two shape contexts SC, SC'.

$$d(SC, SC') = \frac{1}{2}\sum_{k=1}^{K}\frac{\left[SC(k) - SC'(k)\right]^2}{SC(k) + SC'(k)}, \tag{2.7}$$

where $SC(k)$ denotes the k-th bin of C. This is a classical choice when comparing histograms.

2.3.1 Bipartite Graph Matching

Belongie et al. [12] propose a global dissimilarity minimization, via bipartite graph matching.

Let $\mathcal{F}$ and $\mathcal{F}'$ be two databases of shapes such that $|\mathcal{S}| = |\mathcal{S}'|$ and let π be a permutation of $\mathcal{S}'$. The idea is to minimize the total cost of matching,

$$H(\pi) = \sum_{\substack{SC\in\mathcal{S}\\ SC'\in\mathcal{S}'}} d(SC, SC'). \tag{2.8}$$

This is an instance of the weighted bipartite matching problem, which can be solved in $O\left(|\mathcal{S}|^3\right)$ time using the Hungarian method. The matching is finally given by

$$\arg\min_{\pi} H(\pi). \tag{2.9}$$

[2] code and online demo available at http://www.ipol.im/pub/algo/my_affine_sift/

The input of the algorithms solving the problem is a complete graph, represented by its matrix of adjacency (i.e. the square matrix of distances).

Now, if $|\mathcal{S}| \neq |\mathcal{S}'|$ the adjacency matrix can be made square by completing it with infinite costs. Outliers are also handled by extending the adjacency matrix. One adds "dummy" nodes, with constant matching cost ϵ_d to each shape. Equation 2.9 warrants that a point will be matched to a "dummy" node whenever there is no real match available at smaller cost than ϵ_d. Thus, in practice, ϵ_d acts as a global threshold for detecting outliers. Belongie et al. [12] argue that this method handles outliers robustly. The assertion is at least questionable since global thresholds are known to behave poorly in detection problems. Moreover, the choice of ϵ_d is a non-trivial task.

A faster method can be achieved [100], by first performing a coarse matching

- with a downsampled query shape, i.e. by keeping only a small number of representative shape contexts or
- by quantizing the shape contexts in the target database, i.e. by clustering the shape contexts (e.g. with k-means) and only keeping the clusters centroids.

This coarse matching can then be refined with any other method.

2.3.2 The SIFT Matching Rule

Most methods use a nearest neighbor approach to match two sets of descriptors. The most widely used one is the method proposed by Lowe [86] to match SIFT descriptors.

Let $\mathcal{F}$ and $\mathcal{F}'$ be two databases of shapes. A descriptor $SC \in \mathcal{S}$ is matched to its nearest neighbor $SC_1 \in \mathcal{S}'$ if and only if

$$\frac{d(SC, SC_1)}{d(SC, SC_2)} < d^*,$$

where $SC_2 \in \mathcal{S}'$ is the second nearest neighbor of SC. The value d^* is a global parameter often fixed at 0.8.

This method has a double advantage. From one side, it is computationally fast, since computing the first and the second nearest neighbors can be made in $O\left(|\mathcal{S}'| \log |\mathcal{S}'|\right)$ for each SC. From one side, although d^* is global, the actual rejection threshold will depend ultimately on the distance from SC to all the descriptors in $\mathcal{S}'$. An adaptive threshold will behave more robustly than the global threshold presented in the previous section.

A difference with the bipartite matching algorithm is that two shape contexts in $\mathcal{S}$ can be matched with the same shape context in $\mathcal{S}'$. For some applications, this many-to-one matching can be considered a harmful but can be easily solved, for example, by only keeping the match with lowest cost. A second drawback is that a shape context in $\mathcal{S}$ cannot be matched with more than one shape context in $\mathcal{S}'$, thus avoiding one-to-many matches. There are no general workarounds for this issue.

Summary

In the present chapter we overviewed different approaches to the problem of shape recognition, presenting methods to extract, encode and match planar shapes. All of them are far from being perfect and have their own strengths and weaknesses. Motivated by this huge and yet-to-be-explored field, in the following chapters we will develop different methods for the three stages of planar shape recognition.

CHAPTER

3 Shape Extraction

Abstract

This chapter is devoted to the extraction of planar shapes from images. We present a method to select the perceptually significant (i.e. contrasted) level lines from the topographic map, using the Helmholtz principle. Contrarily to the classical formulation by Desolneux et al. [40] where level lines must be entirely salient, it allows to detect partially salient level lines, thus resulting in more robust and more stable detections. Reprising the work by Cao et al. [23], we then tackle the problem of combining two gestalts as a measure of saliency and propose a method that reinforces detections. We finally propose a new method for eliminating non-salient pieces of the previously selected level lines, extending a method to detect subsequences by Grompone et al. [61] to the periodic case.

In the introduction for this thesis, we have stated that two major requirements for any shape detection method are contrast invariance and concentration of information. The former requirement leads to define the set of level lines as a complete and contrast invariant image representation. The latter implies that not all level lines are necessary to obtain a perceptually complete description.

For extracting the level lines of an image (i.e. the topographic map, briefly explained in Chapter 2), we make use of the Fast Level Set Transform (FLST) [98]. In general, the topographic map is an infinite set, and so only quantized grey levels are considered, ensuring that the set is finite. Since level sets and their connected components are ordered by the inclusion relation, the FLST is a hierarchical representation and the topographic map may be embedded in a tree structure. To make things simple, a level line L_i is a descendant of another line L_j in the tree if and only if L_i is included in the interior of L_j.

After computing the topographic map of an image, the perceptually important lines in it have to be selected. The search will focus on unexpected configurations, rising from the perceptual laws of Gestalt Theory [70, 138]. The method

makes use of a perceptual principle called Helmholtz Principle [40] which states that conspicuous structures may be viewed as exceptions to randomness.

3.1 Meaningful Contrasted Boundaries

Within this framework, Desolneux et al. [39] proposed an algorithm to detect contrasted level lines in grey level images, called meaningful boundaries (MB). A definition of randomness has to be established: a background or a contrario model.

Let C be a level line of the image u and $x_0, x_1, \ldots, x_{n-1}$ denote n regularly sampled points of C, with geodesic distance two pixels, which in the a contrario noise model are assumed to be independent. In particular the gradients at these points are independent random variables (the image gradient norm $|Du|$ can be computed on a 2×2 neighborhood).

The curve detection algorithm consists in adequately rejecting the null hypothesis $\mathcal{H}_0$: *the values of $|Du|$ are i.i.d., extracted from a noise image with the same gradient histogram as the image u itself.*

Notation 1. *Let $H_c(\mu) \stackrel{\text{def}}{=} P(|Du| > \mu)$, where Du can be computed by a standard finite differences scheme on a 2×2 neighborhood.*

Definition 5. (Desolneux et al. [39]) *Let $\mathcal{C}$ be a finite set of N_{ll} level lines of u. A level line $C \in \mathcal{C}$ is an ε-meaningful boundary if*

$$\text{NFA}(C) \stackrel{\text{def}}{=} N_{ll}\ H_c(\min_{x \in C} |Du|(x))^{l/2} < \varepsilon \tag{3.1}$$

where l is the length of C. This number is called number of false alarms (NFA) of C.

Proposition 1. *The expected number of ε-meaningful boundaries in a random set E of random curves is smaller than ε.*

Proof. We refer to the work by Cao et al. [23] for a complete proof. □

To summarize, this algorithm claims that perceptually significant level lines correspond to the meaningful boundaries.

Definition 5 has some drawbacks. From one side, the use of the minimum or any punctual measure, for the case, can be an unstable measure in the presence of noise. From the other side, it demands the curve to be not likely to be *entirely* generated by noise (i.e. well contrasted). We already stated that *pieces* of level lines match object boundaries. Moreover, as seen on Figure 3.1, the use of the minimum contrast seems in contradiction with what we perceive. It is therefore too restrictive to impose such a constraint. Since we search for object boundaries, we think the natural model is to select level lines that have well contrasted parts.

In this direction, we propose to modify the definition of the number of false alarms of a curve, to support this new model.

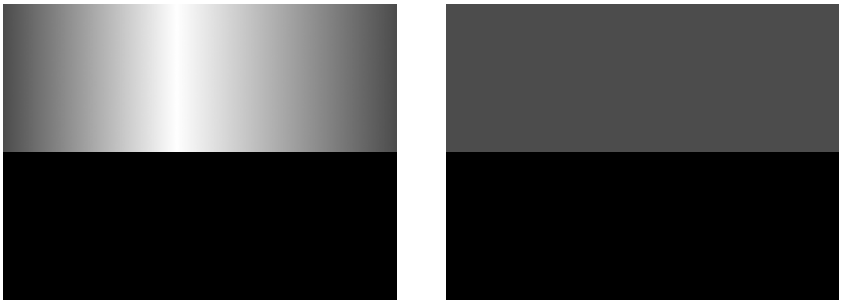


Figure 3.1: Conceptual consequence of using the minimum contrast to detect boundaries. The left image contains a gray gradient and an uniformly black region on its upper and lower halves respectively. The right image is constructed by putting in its upper half the minimum gray level on the left image's upper half. If our perception was tuned to use the minimum contrast to detect the boundary between the two regions, we would perceive that the image on the right is as contrasted as the one on the left, which is clearly not the case.

Notation 2. *Let $x_0, x_1, \ldots, x_{n-1}$ denote n regularly sampled points of C, with geodesic distance 2. For $x \in C$ denote by c_i $(0 \leq i < n)$ the contrast at x_i defined by $c_i = |Du|(x_i)$. We note by $\mathbf{M}$ the vector of the values c_i sorted in ascending order and by μ_k $(0 \leq k < n)$ the k-th value of $\mathbf{M}$.*

For $k \leq N \in \mathbb{N}$ and $p \in [0,1]$, let us denote by

$$\mathcal{B}(N,k;p) \stackrel{\text{def}}{=} \sum_{j=k}^{N} \binom{N}{j} p^j (1-p)^{N-j} \tag{3.2}$$

the tail of the binomial law. Desolneux et al. present a thorough study of the binomial tail and its use in the detection of geometric structures [40].

Following Meinhardt et al. [91], for a given curve, the probability under $\mathcal{H}_0$ that at least k among the n values c_j are greater than μ is given by the tail of the binomial law $\mathcal{B}(n,k,H_c(\mu))$, where $H_c(\mu) = P(|Du| > \mu)$. The regularized beta function, defined by

$$I(x;a,b) = \frac{\int_0^x t^{a-1}(1-t)^{b-1}\,dt}{\int_0^1 t^{a-1}(1-t)^{b-1}\,dt}, \tag{3.3}$$

can be regarded as an interpolation of the binomial tail to the continuous domain [40] and can be computed much faster. Thus it is interesting, and more convenient, to extend this model to the continuous case using the regularized incomplete beta function $I(H_c(\mu); l_1(k,2), l_2(k,2))$ where

$$l_1(k,s) = \frac{l}{s}\frac{n-k}{n} \tag{3.4}$$

$$l_2(k,s) = 1 + \frac{l}{s}\frac{k}{n}. \tag{3.5}$$

This represents the probability under $\mathcal{H}_0$ that, for a curve of length l, some parts with total length greater or equal than $l_1(k,2)$ have a contrast greater than μ.

Definition 6. *Let $\mathcal{C}$ be a finite set of N_{ll} level lines of u. A level line $C \in \mathcal{C}$ is an ε-meaningful boundary if*

$$\mathrm{NFA}_K(C) \stackrel{\text{def}}{=} N_{ll}\, K \min_{0\le k<K} I(H_c(\mu_k); l_1(k,2), l_2(k,2)) < \varepsilon \tag{3.6}$$

where K is a parameter of the algorithm. This number is called number of false alarms (NFA) of C. We also note

$$k_{\min} \stackrel{\text{def}}{=} \arg\min_{0\le k<K} I(H_c(\mu_k); l_1(k,2), l_2(k,2)) \tag{3.7}$$

The parameter K controls the number of points that we allow to be likely generated by noise. In the following, the terms ε-meaningful boundary and NFA will refer to Definition 6.

A classical lemma will be needed in the following.

Lemma 1. *Let X be a real random variable. Let $F(x) = \Pr(X \le x)$ be the repartition function of X. Then, for all $t \in (0,1)$,*

$$\Pr(F(X) < t) \le t.$$

In the same way, let $H(x) = \Pr(X \ge x)$. Then for all $t \in [0,1]$,

$$\Pr(H(X) < t) \le t.$$

Proof. We follow the exposition by Cao et al. [22]. Let us define the pseudo-inverse

$$F^{-1}(t) = \inf\{s, F(s) \ge t\}. \tag{3.8}$$

Because of the convention in its definition, F is right-continuous. Hence,

$$F \circ F^{-1}(t) \ge t. \tag{3.9}$$

Moreover, for all $x \in \mathbb{R}$,

$$F(x) < t \Leftrightarrow x < F^{-1}(t). \tag{3.10}$$

Indeed, let us first assume that $F(x) < t$. If $x \ge F^{-1}(t)$, then, since F is nondecreasing, we have $F(x) \ge F \circ F^{-1}(t) \ge t$, which is a contradiction. Conversely, let us assume $x < F^{-1}(t)$. Then, $F(x) \ge t$ would contradict the definition of $F^{-1}(t)$. This proves the equivalence. Hence,

$$\begin{aligned}
\Pr(F(X) < t) &= \Pr(X < F^{-1}(t)) \quad \text{by Equation 3.10} \\
&= \Pr(\exists y,\, y < F^{-1}(t),\, X \le y) \\
&= \sup_{y < F^{-1}(t)} F(y) \\
&\le t \quad \text{again by Equation 3.10}
\end{aligned} \tag{3.11}$$

The third equality is a basic convergence theorem of measure theory. Note that the last inequality is not strict, because of the passage to the limit.

The second part of the lemma is proved in the same way. Let us define the pseudo-inverse

$$H^{-1}(t) = \sup\{s,\, H(s) \geq t\}. \tag{3.12}$$

Because of the convention in its definition, H is left-continuous. Hence,

$$H \circ H^{-1}(t) \geq t. \tag{3.13}$$

Moreover, for all $x \in \mathbb{R}$,

$$H(x) < t \Leftrightarrow x > H^{-1}(t). \tag{3.14}$$

Indeed, let us first assume that $H(x) < t$. If $x \leq H^{-1}(t)$, then, since H is non-increasing, we have $H(x) \geq H \circ H^{-1}(t) \geq t$, which is a contradiction. Conversely, let us assume $x > H^{-1}(t)$. Then, $H(x) \geq t$ would contradict the definition of $F^{-1}(t)$. This proves the equivalence. Hence,

$$\begin{aligned}
\Pr(H(X) < t) &= \Pr(X > H^{-1}(t)) \quad \text{by Equation 3.14} \\
&= \Pr(\exists y,\, y > H^{-1}(t),\, X \geq y) \\
&= \inf_{y > H^{-1}(t)} H(y) \\
&\leq t \quad \text{again by Equation 3.14}
\end{aligned} \tag{3.15}$$

□

Proposition 2. *The expected number of ε-meaningful boundaries, obtained with Definition 6, in a finite random set E of random curves is smaller than ε.*

Proof. For this proof we follow the scheme from Proposition 12 in [22].

For all k, let us denote by L_1^k the random length of the pieces of C such that $|Du| \geq \mu_k$. From Definition 6, any curve C is ε-meaningful if there is at least one $0 \leq k < K$ such that $N_{ll}\ K\ I(H_c(\mu_k); L_1^k, l_2(k)) < \varepsilon$. Let us denote by $E(C,k)$ this event and recall that all probabilities are under $\mathcal{H}_0$:

$$\Pr(E(C,k)) \stackrel{\text{def}}{=} \Pr(I(Hc(\mu_k); X; l_2(k,2)) < \frac{\varepsilon}{N_{ll}\ K})$$

From Lemma 1, we denote

$$\begin{aligned}
X &= L_1^k & S(x) &= I(Hc(\mu_k); x; l_2(k,2)) \\
t &= \frac{\varepsilon}{N_{ll}\ K} & \Pr(S(X) < t) &= \Pr(E(C,k))
\end{aligned}$$

and finally

$$\Pr(E(C,k)) \leq \frac{\varepsilon}{N_{ll} \cdot K}.$$

The event defined by "C is ε-meaningful" is $E(C) = \bigcup_{0\le k<K} E(C,k)$. Let us denote by $\mathbb{E}_{\mathcal{H}_0}$ the mathematical expectation under $\mathcal{H}_0$. The expected number of ε-meaningful curves is defined as $\mathbb{E}_{\mathcal{H}_0}\left(\sum_{C\in\mathcal{C}} \mathbf{1}_{E(C)}\right)$ where $\mathbf{1}_A$ is the indicator function of the set A. Then

$$\mathbb{E}_{\mathcal{H}_0}\left(\sum_{C\in\mathcal{C}} \mathbf{1}_{E(C)}\right) \le \sum_{\substack{C\in\mathcal{C}\\ 0\le k<K}} \Pr\left(E(C,k)\right) \le \sum_{\substack{C\in\mathcal{C}\\ 0\le k<K}} \frac{\varepsilon}{N_{ll}\cdot K} = \varepsilon$$

□

This new model is an extension of the previous one, since $\mathrm{NFA}_K(C) = \mathrm{NFA}(C)$ when $K = 1$. In fact, Definition 6 is no other than a relaxation of Definition 5. We should expect to have new detections and to detect the same lines, with increased stability. This comes from the fact that several punctual measures are used and the minimum is taken over their probability. This was experimentally checked and some results can be seen in Section 4.3.

The choice of the value of K cannot be directly made. To begin, its effect is highly dependent on the length of the curve. It is totally different to allow 5 points with low contrast in a 20 points curve than in a 200 points curve. The value of K must be set in a way that the length of the curve is taken into account. In fact, K does not depend only on the curve length. For example, if a curve has a very highly contrasted part, it is very probable that this part corresponds to an object boundary. We would want to detect that curve, even if the curve is very long and the rest of it is poorly contrasted. In this case, we would need to allow for larger values of K. On the contrary, if a curve is evenly contrasted (poorly or not), it is unnecessary to set a high value of K. To summarize, the value of K has to be chosen as a function of the curve length and of the image contrast along the curve.

Definition 7. *Following Definition 6, for a given curve C, we set the value of K as*

$$\hat{K}_\varphi \stackrel{\text{def}}{=} \underset{i<n}{\arg\max}\left(\frac{\sum_{j=0}^{i}\mu_j}{\sum_{j=0}^{n-1}\mu_j} < \varphi\right) \tag{3.16}$$

where n is the number of regularly sampled independent points in C and $\varphi \in [0,1]$ is the new parameter of the detection algorithm.

This choice of K is indeed adaptive to the length and contrast of each level line. It is in fact quite stable for values of $\varphi < 0.05$. Greater values lead to an overdetection and, in general, no perceptually significant level lines appear (Figure 3.2). For example, all the experiments in this paper were produced using $\varphi = 0.02$.

In [23], other modifications are proposed to the basic meaningful boundaries algorithm. We will not discuss them in this work, since we are only interested in the redefinition of the NFA and its consequences.

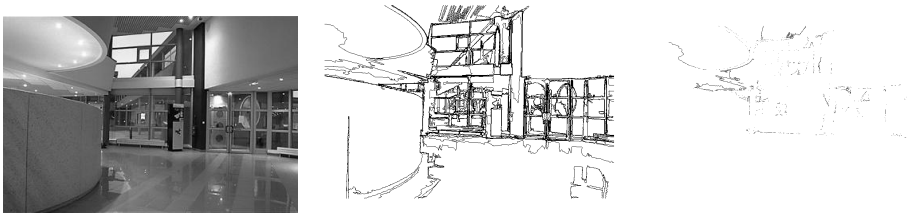


Figure 3.2: Changing the parameter φ does not significantly affect the result of the new meaningful boundaries method. On the left, original image; on the center, maximal meaningful boundaries with $\varphi = 0.02$; on the right, difference between $\varphi = 0.02$ and $\varphi = 0.04$.

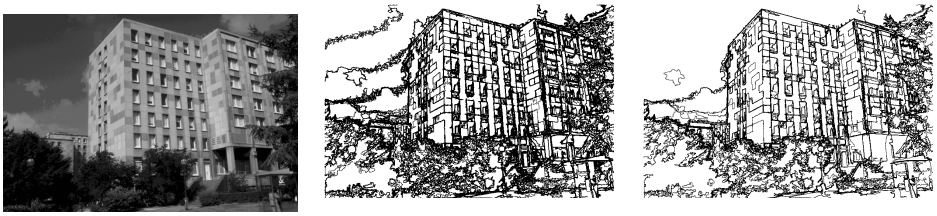


Figure 3.3: Effect of the maximality condition over the meaningful boundaries of an image. On the left, original image; on the center, meaningful boundaries with Definition 5; on the left maximal meaningful boundaries with Definition 5. In (b) we have 8987 level lines, and 517 in (c).

3.1.1 Maximal boundaries

Meaningful boundaries usually appear in parallel and redundant groups, because of interpolation. Since the meaningful level lines inherit the tree structure of the original tree, Desolneux et al. [40] use this structure to efficiently remove redundant boundaries.

Definition 8. (Monasse and Guichard [98]) *A monotone section of a level lines tree is a part of a branch such that each node has a unique son and where grey level is monotone (no contrast reversal). A maximal monotone section is a monotone section which is not strictly included in another one.*

Definition 9. (Desolneux et al. [39]) *A meaningful boundary is maximal meaningful if it has a minimal NFA in a maximal monotone section.*

Figure 3.3 shows an example of the reduction of the number of level lines caused by the maximality constraint. Parallel level lines are eliminated, leading to “thinner edges”.

3.1.2 Practical implications of the change in the NFA

We address now to the following question: is there a fundamental difference in practice between detecting with Definition 5 and detecting with Definition 6? The answer is that, given an image, this change implies noticeable differences in the detected curves. Indeed, the new definition of meaningful boundaries is more robust since the NFAs attained are much lower. Taking the minimum of probabilities is also more stable than taking the minimum on any punctual measure, see Figure 3.4.

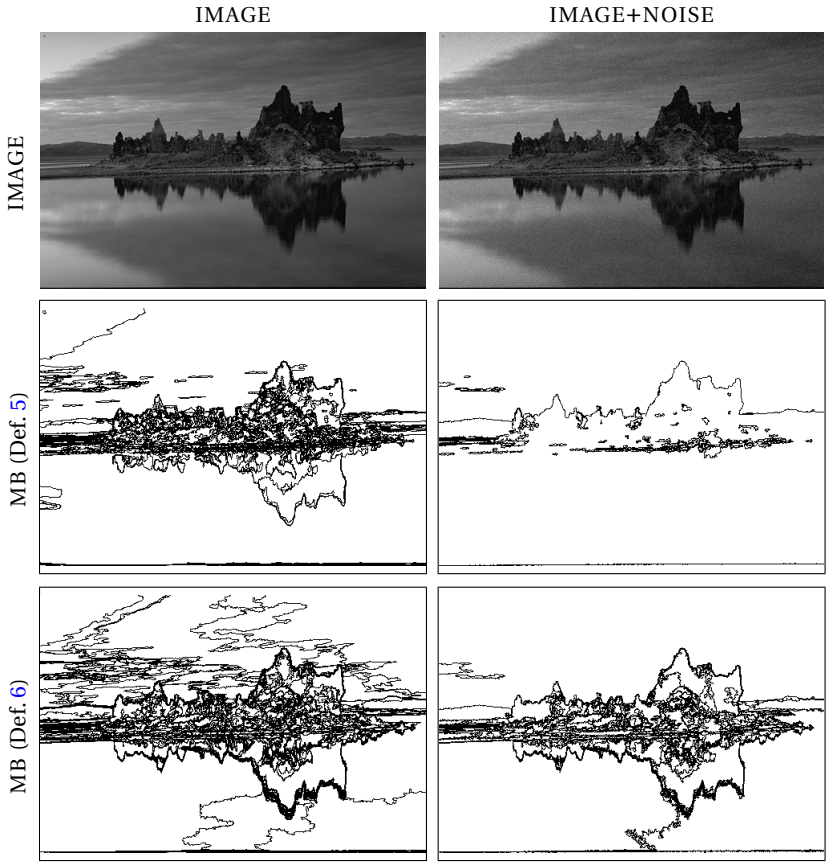


Figure 3.4: Noise contamination example. The image on the right is contaminated by a small amount of noise. Definition. 5 takes a minimum of punctual measures, thus the result is affected. On the counterpart, result with Definition 6 is less affected, as it deals with probabilities. Notice that no smoothing is performed previous to detection.

In some cases, by relaxing the meaningfulness threshold, visually better results can be achieved with Def. 5. More level lines are kept, but at the expense of having lower confidence on them. The key advantage with Def. 6 is that, for a given

threshold for ε, more level lines are selected.

One of the possible arguments against Def. 6 could be that it is no more than a shift of the threshold on the NFA. Specifically, that there exists a threshold $\varepsilon' > \varepsilon$ for which the detections using Def. 5 and ε' would be the same as using Def. 6 and ε. However, the assertion is clearly false, as shown in Fig. 3.5.

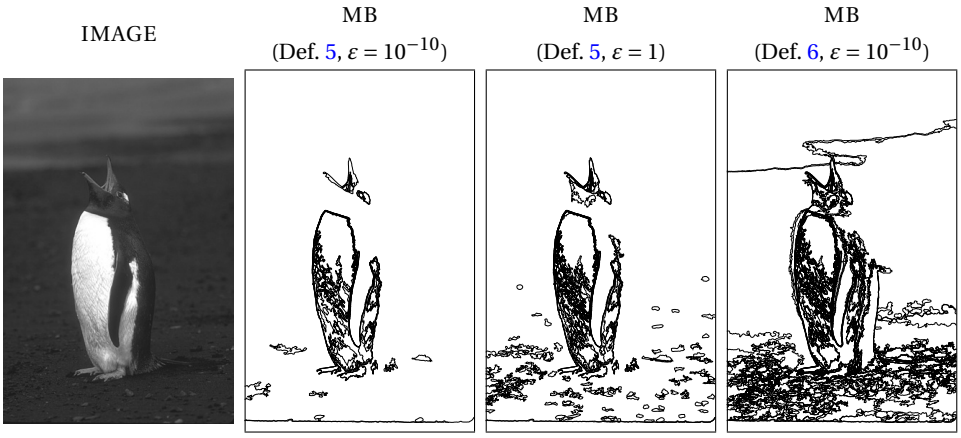


Figure 3.5: Def. 6 is not merely a shift of the threshold on the NFA from Def. 5: even relaxing the threshold to its limit ($\varepsilon = 1$), the result with the old method remains roughly the same. A lot of structure missed with Definition 5 is recovered with Definition 6.

Underdetection is far more dangerous than overdetection. Losing structure is critical in most applications (scene reconstruction, image matching, etc.) as it can end-up in a total failure. Detection noise can always be handled (or even tolerated) when the amount of noise does not occlude information, as in our case. Boundary detection is therefore more complete and reliable. This is experimentally checked in all examples, even if the difference is more striking in some examples than in others.

3.2 Combining contrast and good continuation

Let C be a rectifiable planar curve, parameterized by its length. Let l be the length of C and $x = C(s) \in C$. With no loss of generality, we assume that $s = 0$.

Definition 10. (Cao et al. [23]) *Let $s > 0$ be a fixed positive value such that $2s < l$. We call regularity of C at x (at scale s) the quantity*

$$R_s(x) = \frac{\max(|x - C(-s)|, |x - C(s)|)}{s} \tag{3.17}$$

where $|x_i - x_j|$ represents the length of the curve's portion between x_i and x_j.

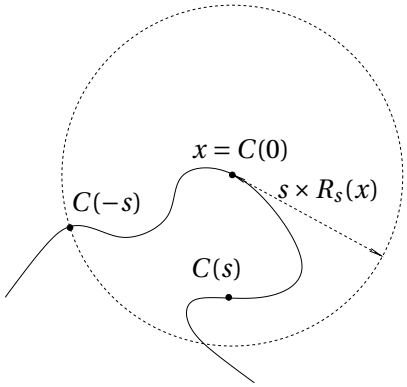


Figure 3.6: Reproduced from the work by Cao et al. [23]. The regularity at x is obtained by comparing the radius of the circle with s. The radius is equal to s if and only if the curve is a straight line. If the curve has a large curvature, the radius will be small compared to s.

Figure 3.6 visually explains the pertinence of this definition. Only when one of the subcurves $C((-s,0))$ or $C((0,s))$ is a line segment, $R_s(x) = 1$; in all other cases $R_s(x) < 1$. When s is small enough, regularity is inversely proportional to the curve's curvature around x [23].

The question about the choice of s arises naturally and was studied in detail by Cao et al. [23] and Musé [105]. We will limit ourselves to state that a larger value of s (thus at less local scale of analysis) is more robust to noise. On the other side, s should not be too large either. In practice, and following Cao et al. [23] one may safely set $s = 5$ or $s = 10$.

Let us denote by $H_s(r)$ the distribution of the regularity in white noise level lines, i.e.

$$H_s(r) = P(R_s(x) > r, x \in C, C \text{ is a white noise level line}), \tag{3.18}$$

which depends only on s and can be empirically estimated.

Again, the curve detection algorithm consists in adequately rejecting the null hypothesis $\mathcal{H}_0$*: the values of* $|R_s|$ *are i.i.d., extracted from a noise image.* It is natural to assume, in the background model, that contrast and regularity are independent.

Let us forget for the moment the aforementioned issues associated with the use of extremal (the minimum) statistics.

Definition 11. (Cao et al. [23]) *Let* C *be a level line in a finite set* $\mathcal{C}$ *of* N_{ll} *level lines of image* u*. Let*

$$\mu = \min_{x \in C} |Du|(x)$$
$$\rho = \min_{x \in C} R_s(x)$$

be respectively the minimal quantized contrast and regularity along C. The level line C is an ε-meaningful regular boundary if

$$\mathrm{NFA}_S(C) \stackrel{\text{def}}{=} N_{ll}\ H_s(\rho)^{l/2s} < \varepsilon. \tag{3.19}$$

The level line C is an ε-meaningful contrasted regular boundary if

$$\mathrm{NFA}_{\mathrm{CR}}(C) \stackrel{\text{def}}{=} N_{ll}\ H_c(\mu)^{l/2}\ H_s(\rho)^{l/2s} < \varepsilon. \tag{3.20}$$

Unfortunately, in their article [22] Cao et al. do not prove that the expected number of ε-meaningful contrasted regular boundaries in a finite set of random curves is smaller than ε. This fact is indeed annoying since the threshold ε is emptied of meaning. It is not by any means an easy proof and we have not found a solution yet. However, we have proven that by slightly changing the definition of meaningful contrasted regular boundaries (Definition 11) in the following manner

$$\mathrm{NFA}_{\mathrm{CR}}(C) \stackrel{\text{def}}{=} N_{ll}\ H_c(\mu)^{l^2/2s}\ H_s(\rho)^{l^2/2s}. \tag{3.21}$$

a proof, given in Appendix 3.B, can be built.

Although theoretically sound, meaningful contrasted regular boundaries defined by Equation 3.21 do not provide satisfactory results. This is a direct consequence of using l^2. With respect to contrasted meaningful boundaries (Definition 5) and even if the regularity term has high probability (say one), elevating the contrast term to a much larger number will shift the NFA of all curves towards zero. Irregular curves that were not meaningful by its contrast, might become meaningful regular boundaries. This is certainly an unwanted side effect.

An alternative way of combining regularity and contrast, which does not suffer from the aforementioned shifting effect, must be used. The following definition

$$\mathrm{NFA}_{\mathrm{CR}}(C_i) \stackrel{\text{def}}{=} N_{ll} \max\left(H_c(\mu_i)^{2l/2},\ H_s(\rho_i)^{2l/2s}\right) \tag{3.22}$$

exhibits some interesting properties:

- A contrasted but irregular curve will not be detected;
- A regular but non contrasted curve will not be detected;
- An irregular and non contrasted curve will not be detected;
- A regular and contrasted curve will be detected.

Both gestalts, i.e. contrast and good continuation, interact in a novel way: instead of cooperating, they compete. As the exponent in the contrast term is greater than the exponent in the regularity term ($l > l/s$), the contrast term will in general dominate the detections and the regularity will act as an additional sanity check.

Regarding the shifting phenomenon, we will still have it. However, $2l$ is much less aggressive than l^2 and its effect will be doubly mitigated: (1) since $l \gg 2$ and (2) because of the controlling effect of using the maximum.

Definition 6 introduced a relaxed version of meaningful contrasted boundaries which included Definition 5 as a particular case. We profit from such knowledge and also relax the definition of meaningful contrasted regular boundaries.

Definition 12. *Let $\mathcal{C}$ be a finite set of N_{ll} level lines of u. A level line $C \in \mathcal{C}$ is an ε-meaningful contrasted regular boundary if*

$$\mathrm{NFA}_K^{\mathrm{CR}}(C) \stackrel{\text{def}}{=} N_{ll}\, K_c\, K_s \max\left(\min_{0\le k<K_c} I_c(C,k)^2,\ \min_{0\le k'<K_s} I_s(C,k')^2\right) < \varepsilon, \tag{3.23}$$

where

$$I_c(C,k) = I(H_c(\mu_k); l_1(k,2), l_2(k,2))$$
$$I_s(C,k') = I(H_s(\rho_{k'}); l_1(k',2s), l_2(k',2s))$$

and K_c and K_s are parameters of the algorithm. This number is called number of false alarms (NFA) of C.

Here K_c and K_s have the same meaning that K in Definition 6 and the same strategy, detailed in Definition 7, can be used to set their values

Proposition 3. *The expected number of ε-meaningful contrasted regular boundaries in a finite random set E of random curves is smaller than ε.*

Proof. The same assumptions from the proof of Proposition 6 hold.

Let $X_i = \mathbf{1}_{C_i \text{ is meaningful}}$ and $N = \#E$. Let us denote by $\mathbb{E}_{\mathcal{H}_0}$ the mathematical expectation under $\mathcal{H}_0$. Then

$$\mathbb{E}\left(\sum_{i=1}^{N}\sum_{k=1}^{K_c}\sum_{k'=1}^{K_s} X_i\right) = \mathbb{E}\left(\mathbb{E}\left(\sum_{i=1}^{n}\sum_{k=1}^{k_c}\sum_{k'=1}^{k_s} X_i \mid N=n, K_c=k_c, K_s=k_s\right)\right)$$

We have assumed that N is independent from the curves and K_c, K_s are input parameters. Thus, conditionally to $N=n$, the law of $\sum_{i=1}^{N} X_i$ is the law of $\sum_{i=1}^{n} Y_i$ where

$$Y_i = \mathbf{1}_{n\,k_c\,k_s \max\left(\min_{0\le k<k_c} I_c(C_i,k)^2,\ \min_{0\le k'<k_s} I_s(C_i,k')^2\right) < \varepsilon}.$$

By the linearity of expectation

$$\mathbb{E}\left(\sum_{i=1}^{n}\sum_{k=1}^{k_c}\sum_{k'=1}^{k_s} X_i\right) = \mathbb{E}\left(\sum_{i=1}^{n}\sum_{k=1}^{k_c}\sum_{k'=1}^{k_s} Y_i\right) = \sum_{i=1}^{n}\sum_{k=1}^{k_c}\sum_{k'=1}^{k_s} \mathbb{E}(Y_i).$$

Since Y_i is a Bernoulli variable,

$$\begin{aligned}
&\mathbb{E}(Y_i) = \Pr(Y_i = 1)\\
&= \Pr\left(n\,k_c\,k_s \max\left(\min_{0\le k<k_c} I_c(C_i,k)^2,\ \min_{0\le k'<k_s} I_s(C_i,k')^2\right) < \varepsilon\right)\\
&= \sum_{l=0}^{\infty} \Pr\left(n\,k_c\,k_s \max\left(\min_{0\le k<k_c} I_c(C_i,k)^2,\ \min_{0\le k'<k_s} I_s(C_i,k')^2\right) < \varepsilon \;\middle|\; L_i = l\right) \Pr(L_i = l).
\end{aligned}$$

Let us finally denote by $\alpha_1 \dots \alpha_l$ the l independent values of $|Du|$ and $\gamma_1 \dots \gamma_{l/s}$ the l/s independent values of $|R_s|$. Again, we have assumed that L_i is independent of the gradient and regularity distributions in the image. Thus conditionally to $L_i = l$,

$$\Pr\left(n\,k_c\,k_s \max\left(\min_{0\le k<k_c} I_c(C_i,k)^2,\ \min_{0\le k'<k_s} I_s(C_i,k')^2\right) < \varepsilon \mid L_i = l\right) =$$
$$\Pr\left(n\,k_c\,k_s \max\left(\min_{0\le k<k_c} I_c(C_i,k)^2,\ \min_{0\le k'<k_s} I_s(C_i,k')^2\right) < \varepsilon\right) =$$
$$\Pr\left(\max\left(\min_{0\le k<k_c} I_c(C_i,k),\ \min_{0\le k'<k_s} I_s(C_i,k')\right) < \left(\frac{\varepsilon}{n\,k_c\,k_s}\right)^{1/2}\right) =$$
$$\Pr\left(\min_{0\le k<k_c} I_c(C_i,k) < \left(\frac{\varepsilon}{n\,k_c\,k_s}\right)^{1/2}\right)\Pr\left(\min_{0\le k'<k_s} I_s(C_i,k') < \left(\frac{\varepsilon}{n\,k_c\,k_s}\right)^{1/2}\right)$$

From proof of Proposition 2,

$$\Pr\left(\min_{0\le k<k_c} I_c(C_i,k) < \left(\frac{\varepsilon}{n\,k_c\,k_s}\right)^{1/2}\right)\Pr\left(\min_{0\le k'<k_s} I_s(C_i,k') < \left(\frac{\varepsilon}{n\,k_c\,k_s}\right)^{1/2}\right) \le$$
$$\le \left(\frac{\varepsilon}{n\,k_c\,k_s}\right)^{1/2}\left(\frac{\varepsilon}{n\,k_c\,k_s}\right)^{1/2} = \frac{\varepsilon}{n\,k_c\,k_s}$$

Finally

$$\mathbb{E}(Y_i) \le \frac{\varepsilon}{n\,k_c\,k_s} \quad \Rightarrow \quad \sum_{i=1}^{n}\sum_{k=1}^{k_c}\sum_{k'=1}^{k_s} \mathbb{E}(Y_i) \le \varepsilon. \tag{3.24}$$

□

3.2.1 Discussion

We will now examine the results of the proposed competition between contrast and good continuation.

The benefits of using meaningful contrasted regular boundaries are clear in Figure 3.7. In both examples, only using contrast produces an overdetection (level lines are detected in areas with texture, e.g. vegetation on the left, or exhibiting a slight gradient, e.g. the sky and the dome on the right) while only using good continuation produces an underdetection (e.g. the bridge on the left and the bell on the right). The combination of both gestalts corrects the issues by keeping the best from both worlds: most undesired level lines disappear while the desired ones are kept.

Although more complicated to analyze, Figure 3.8 further supports our claims. See the detail on Harrison Ford's sleeve: it is completely lost by using contrast, partially recovered by using good continuation and well recovered by combining them.

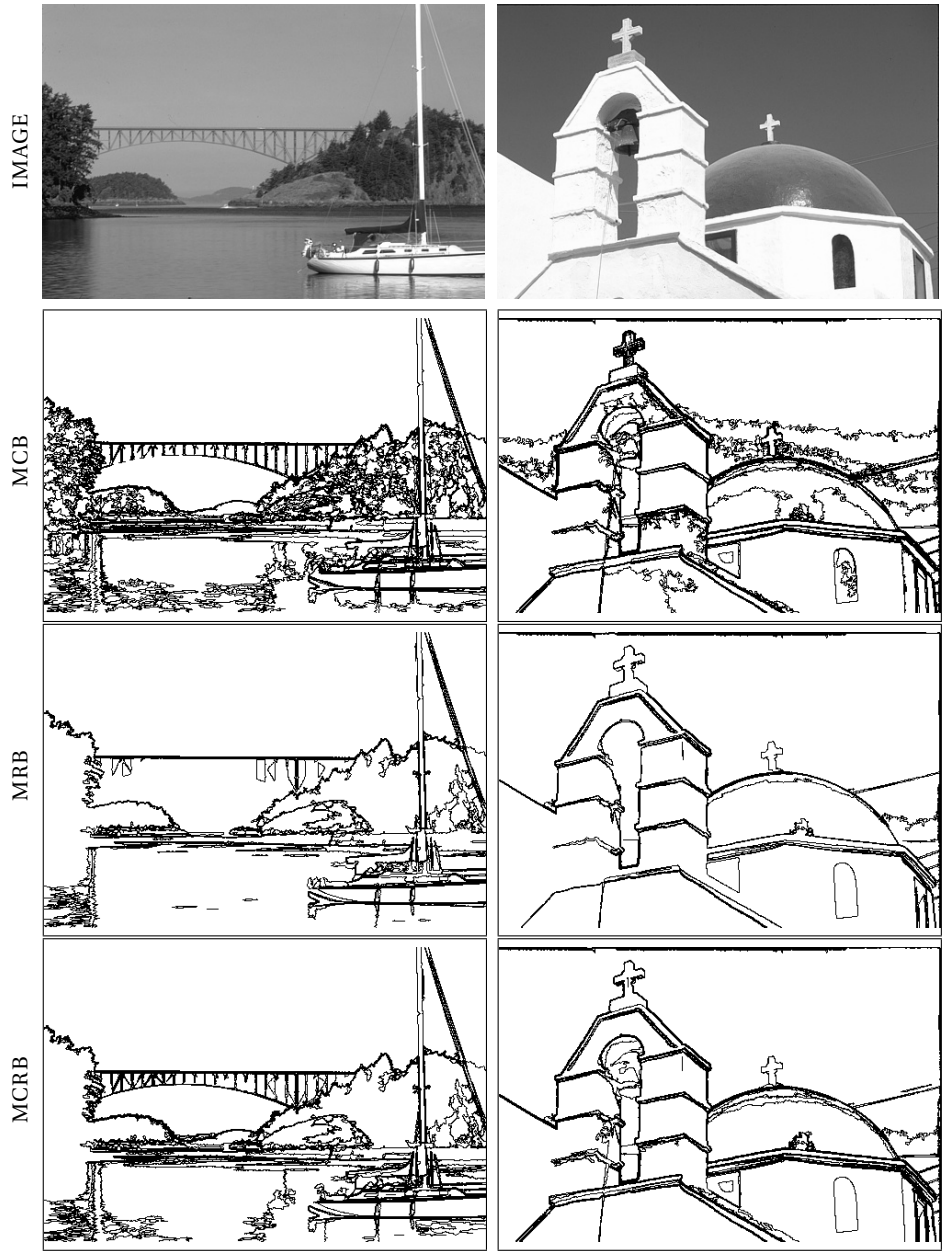


Figure 3.7: Comparison of Meaningful Contrasted Boundaries (MCB) using Definition 6, Meaningful Regular Boundaries (MRB) using Definition 11 and Meaningful Contrasted Regular Boundaries (MCRB) using Definition 12.

It is important to point out that in general, good continuation has a predominant effect over contrast. In the depicted examples, meaningful contrasted boundaries have lower NFAs than meaningful smooth ones. This explains the visual effect that we perceive at looking at the results: contrasted regular boundaries are basically regular boundaries reinforced by some contrasted parts.

The last example in Figure 3.9 is a real scene, extremely complicated from the edge detection point of view. In any case, all results are globally satisfactory. Noticeable differences between the methods are perceived by looking at the signs containing letters.

3.3 Detecting Periodic Subsequences

Grompone et al. [61] propose a method for accurately detecting straight line segments in a digital image. It is based on the Helmholtz principle and hence parameterless. In the authors' words, "at the core of the work lies a new way to interpret binary sequences in terms of unions of segments".

A sequence $S = (s_i)_{1 \leq i \leq L}$ of length L is binary if $\forall i,\ s_i \in \{0,1\}$. A subsequence $a \subseteq S$ is defined by a pair of indices $\left(a^{(1)}, a^{(2)}\right)$ with $1 \leq a^{(1)} < a^{(2)} \leq L$, such that $\left(\forall s_i,\ a^{(1)} \leq s_i \leq a^{(2)}\right)\ s_i \in a$.

Notation 3. *Given a binary sequence S of length L, an n-subsequence is an n-tuple $(a_1, \ldots, a_n)$ of n disjoints subsequences $a_i \subseteq S$. The set of all n-subsequences in S will be denoted by $\mathcal{M}(n, S)$.*

We define $k(a) = \#\{s_i \mid i \in [a^{(1)}, a^{(2)}] \wedge s_i = 1\}$ and $l(a) = a^{(2)} - a^{(1)} + 1$ (i.e. the length of a).

Notice that $\#\mathcal{M}(n, S) = \binom{L}{2n}$ [61].

Definition 13. *Given a binary sequence S of length L, an n-subsequence $(a_1, \ldots, a_n)$ in $\mathcal{M}(n, S)$ is said ε-meaningful if*

$$\mathrm{NFA}(a_1, \ldots, a_n) \stackrel{\text{def}}{=} \binom{L}{2n} \prod_{i=1}^{n} (l(a_i) + 1)\mathcal{B}(l(a_i), k(a_i), p) < \varepsilon$$

where $p = \Pr(s_i = 1), 1 \leq i \leq L$. This number is called number of false alarms (NFA) of $(a_1, \ldots, a_n)$.

A run in S is a maximal subsequence only containing ones, i.e.

$$\left(\forall i \in [a^{(1)}, a^{(2)}],\ s_i = 1\right) \wedge \left(a^{(1)} = 1 \vee s_{a^{(1)}-1} = 0\right) \wedge \left(a^{(2)} = L \vee s_{a^{(2)}+1} = 0\right).$$

One can restrict the search for n-subsequences to the ones where each of the n subsequences starts at a run start and ends at a run end [61]. We denote by R the number of runs in S.

Figure 3.8: Comparison of Meaningful Contrasted Boundaries (MCB) using Definition 6, Meaningful Regular Boundaries (MRB) using Definition 11 and Meaningful Contrasted Regular Boundaries (MCRB) using Definition 12.

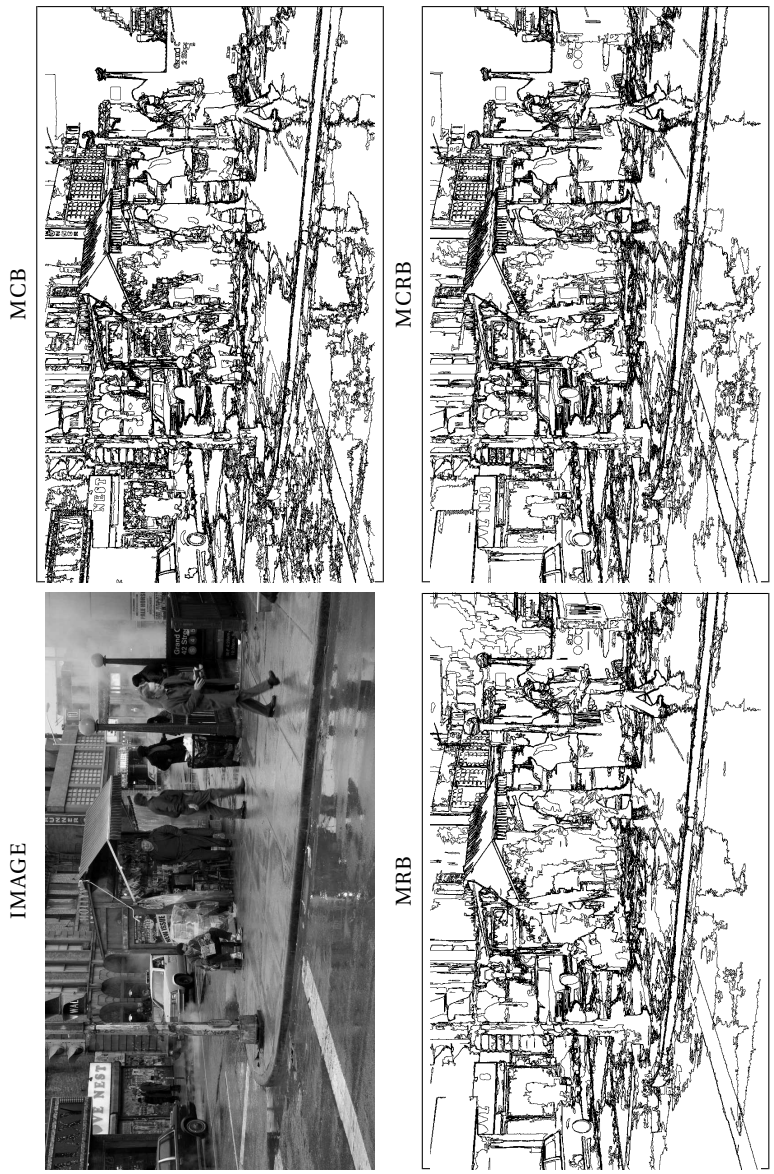


Figure 3.9: Comparison of Meaningful Contrasted Boundaries (MCB) using Definition 6, Meaningful Regular Boundaries (MRB) using Definition 11 and Meaningful Contrasted Regular Boundaries (MCRB) using Definition 12.

Figure 3.10: A sequence example: the sequence has length 128 and 64 points with value 1 represented by dashes. This sequence has been generated by randomly drawing a subset of size 64 from a set of size 128 with a uniform law over all possible such subsets.

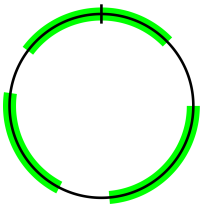


Figure 3.11: A periodic sequence where runs are represented in green. If treated as a non-periodic sequence, any subsequence detector would detect four subsequences at best, when in fact the desired result is to detect three subsequences.

Definition 14. *Given a binary sequence S, its maximal ε-meaningful subsequence $(a_1,\dots,a_n)^*$ is defined as*

$$(a_1,\dots,a_n)^* \stackrel{\text{def}}{=} \underset{\substack{1\le n\le R\\ (a_1,\dots,a_n)\in\mathcal{M}(n,S)}}{\arg\min} \text{NFA}(a_1,\dots,a_n).$$

Proposition 4. *The expected number of ε-meaningful n-subsequences in a random binary sequence is smaller than ε.*

Proof. We refer to the work by Grompone et al. [23] for a complete proof. □

We propose now to extend the above definitions to support periodic binary sequences. A binary sequence $S = (s_i)_{1\le i\le L}$ is made periodic by considering L its period. Periodic sequences are different in nature from their non-periodic counterparts, see Figure 3.11. A definition suitable for the periodic case is needed.

In the periodic case, a subsequence must be defined more carefully. A subsequence $a \subseteq S$ is defined by a pair of indices $\left(a^{(1)}, a^{(2)}\right)$:

- if $a^{(1)} < a^{(2)}$ then the non-periodic definition holds, i.e. $1 \le a^{(1)} < a^{(2)} \le L$, and $\left(\forall s_i,\ a^{(1)} \le s_i \le a^{(2)}\right)\ s_i \in a$. Such subsequences are intra-subsequences.
- if $a^{(1)} > a^{(2)}$, $\left(\forall s_i,\ 1 \le s_i \le a^{(2)} \vee a^{(1)} \le s_i \le L\right)\ s_i \in a$. Such subsequences are inter-subsequences.

Runs are modified accordingly to also cover inter-subsequences.

Notation 4. *Given a periodic binary sequence S of period L, a periodic n-subsequence is an n-tuple $(a_1,\dots,a_n)$ of n disjoints subsequences $a_i \subseteq S$. The set of all n-subsequences in S will be denoted by $\mathcal{M}(n,S)$.*

We define $k(a) = \#\{s_i \mid i \in [a^{(1)}, a^{(2)}] \wedge s_i = 1\}$ and the length of a as

$$l(a) = \begin{cases} a^{(2)} - a^{(1)} + 1, & \text{if } a \text{ is an intra-subsequence;} \\ a^{(2)} + L - a^{(1)} + 1, & \text{if } a \text{ is an inter-subsequence.} \end{cases}$$

Notice that $\#\mathcal{M}(n,S) = 2\binom{L}{2n}$ since from each pair of points in S two subsequences can be constructed.

Definition 15. *Given a periodic binary sequence S of period L, an n-subsequence $(a_1,\dots,a_n)$ in $\mathcal{M}(n,S)$ is said ε-meaningful if*

$$\mathrm{NFA}(a_1,\dots,a_n) \stackrel{\text{def}}{=} 2\binom{L}{2n}\prod_{i=1}^{n}(l(a_i)+1)\,\mathcal{B}(l(a_i),k(a_i),p) < \varepsilon$$

where $p = \Pr(s_i = 1), 1 \le i \le L$. This number is called number of false alarms (NFA) of $(a_1,\dots,a_n)$.

Proposition 5. *The expected number of ε-meaningful n-subsequences in a random periodic binary sequence is smaller than ε.*

Proof. This proof follows closely the one by Grompone et al. [23] but adapted to periodic sequences. The expected number of ε-meaningful n-subsequences is given by

$$\begin{aligned} \mathbb{E}\left(\sum_{(a_1,\dots,a_n)\in\mathcal{M}(n,S)} \mathbf{1}_{\mathrm{NFA}(a_1,\dots,a_n)<\varepsilon}\right) &= \sum_{(a_1,\dots,a_n)\in\mathcal{M}(n,S)} \mathbb{E}\left(\mathbf{1}_{\mathrm{NFA}(a_1,\dots,a_n)<\varepsilon}\right) \\ &= \sum_{(a_1,\dots,a_n)\in\mathcal{M}(n,S)} \Pr\left(\mathrm{NFA}(a_1,\dots,a_n) < \varepsilon\right). \end{aligned} \tag{3.25}$$

Then $\mathrm{NFA}(a_1,\dots,a_n) < \varepsilon$ implies that

$$\begin{aligned} 2\binom{L}{2n}\prod_{i=1}^{n}(l(a_i)+1)\,\mathcal{B}(l(a_i),k(a_i),p) &< \varepsilon \\ \prod_{i=1}^{n}\mathcal{B}(l(a_i),k(a_i),p) &< \frac{\varepsilon}{2\binom{L}{2n}\prod_{i=1}^{n}(l(a_i)+1)}. \end{aligned} \tag{3.26}$$

Let $U_i = \mathcal{B}(l(a_i),k(a_i),p)$ be a random variable and let α be a positive number,

$$\begin{aligned} &\Pr\left(\prod_{i=1}^{n} U_i < \alpha\right) = \\ &\qquad \sum_{u_2,\dots,u_n} \Pr\left(\prod_{i=1}^{n} U_i < \alpha \,\middle|\, U_2 = u_2,\dots,U_n = u_n\right)\Pr\left(U_2 = u_2,\dots,U_n = u_n\right). \end{aligned} \tag{3.27}$$

Since the a_i are disjoint, the U_i are independent then

$$\Pr\left(\prod_{i=1}^{n} U_i < \alpha\right) = \sum_{u_2,\ldots,u_n} \Pr\left(\prod_{i=1}^{n} U_i < \frac{\alpha}{u_2 \ldots u_n}\right) \Pr\left(U_2 = u_2, \ldots, U_n = u_n\right). \quad (3.28)$$

Now using that $\Pr(U_i < \alpha) < \alpha$ (see Lemma 1) and that $\Pr\left(U_2 = u_2, \ldots, U_n = u_n\right) \leq \Pr\left(U_2 \leq u_2, \ldots, U_n \leq u_n\right)$ one gets, since there are $l(a_i) + 1$ possible values for U_i,

$$\Pr\left(\prod_{i=1}^{n} U_i < \alpha\right) < \prod_{i=2}^{n} (l(a_i) + 1)\alpha < \prod_{i=1}^{n} (l(a_i) + 1)\alpha. \quad (3.29)$$

Let us recall that $\#\mathcal{M}(n, S) = 2\binom{L}{2n}$, then setting $\alpha = \frac{\varepsilon}{2\binom{L}{2n}\prod_{i=1}^{n}(l(a_i)+1)}$ gives the wanted result. □

The maximality rule from Definition 14 holds unchanged in the periodic case.

On the implementation side, Grompone et al. [61] describe a dynamic programming scheme for the non-periodic case that eases the heavy computational burden. We show now that implementing the algorithm for detecting periodic subsequences is indeed straightforward.

We begin by shifting the periodic sequence to transform inter-subsequences into intra-subsequences. A circular shift to the left is used, see Figure 3.12a. We first form a non-periodic sequence $S^{(2)}$ of length $2L$ from two periods of the periodic sequence S of period L, see Figure 3.12b. We say that $S^{(2)}$ is a 2-period sequence. Two key tricks allow us to solve the problem:

1. restrict the number of subsequences in all tested subsequences. Let R be the number of runs in S. Then it is sufficient to test only n-subsequences where $1 \leq n \leq R$. For example, in Figure 3.12b it is sufficient to look for n-subsequences where $n \leq 3$.
2. subsequences longer than L are not tested.

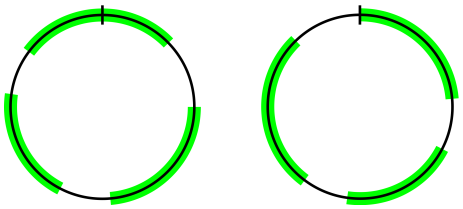
With these two restrictions, one can simply detect non-periodic subsequences in non-periodic sequence $S^{(2)}$ and the result will be optimal.

3.3.1 Boundary clean-up

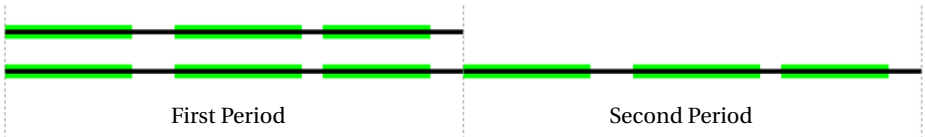
Following Cao et al. [23], Proposition 2 asserts that if a level line is a meaningful boundary, then it cannot be entirely generated in white noise (up to ε false detections on the average) but it can have parts that are likely to be contained in noise.

Cao et al. [23] propose to give an upper bound to the size of those parts. Assume that C is a piece of level line with L independent points, contained in a non-edge part, described by the noise model. The probability that L is larger than $l > 0$ needs to be estimated, knowing that $|Du| \geq \mu$. This is exactly the a posteriori length distribution $p(\mu; l) \stackrel{\text{def}}{=} P(L \geq l || Du| \geq \mu)$. The estimation of this distribution was studied by Cao et al. [22].

Let us now consider an image u with N_{ll} (quantized) level lines. Let us also denote by N_l the number of all possible sampled subcurves of these level lines.



(a) In the presence of an inter-subsequence, the periodic sequence is circularly shifted to the left to transform it into an intra-subsequence.



(b) As all three runs are intra-subsequences, a 3-subsequence should be detected

Figure 3.12: Periodic sequences where runs are represented in green. Subsequence detection must take place in a 2-period sequence to prevent from splitting inter-subsequences.

(N_l is the sum of the squared number of independent points of the lines if they are closed).

As in Proposition 2, it can be proved that $N_l \cdot p(\mu; l)$ is an upper bound of the expected number of pieces of lines of length larger than l with gradient larger than μ. For a fixed μ, let be l such that $N_l \cdot p(\mu; l) < \varepsilon$. Then, we know that, on the average, we cannot observe more than ε pieces of level line with a length larger than l and a gradient everywhere larger than μ.

Then one can define $\mathcal{L}(\mu) = \inf\{l, N_l \cdot p(\mu; l) < \varepsilon\}$ and keep every subcurve of any meaningful boundary with length equal or greater than $\mathcal{L}(\mu)$, where $|Du| \geq \mu$.

The value of μ can be seen as a new parameter of the method. Its value can be fixed arbitrarily using a conservative approach [22]. Letting $|Du|$ less than 1, means that edges with an accuracy less than one pixel may be detected. Thus, taking $\mu = 1$ is the least restrictive choice. For μ about 1, values of $\mathcal{L}(\mu)$ less than a few hundreds are obtained.

Since $\mathcal{L}(\mu)$ is a decreasing function of μ, fixing it at a small value produces large lengths. We are imposing that the contrasted pieces have to be very large and this is not always the case, as argued before. Furthermore the probability distribution $p(\mu; l)$ has to be estimated. We propose to take a different path to remove non-contrasted boundary parts.

In Definition 6, pieces of a meaningful boundary are explicitly allowed to be generated in white noise. We are certainly not interested in these pieces and this relaxation responds to the fact that we want to retrieve the remaining pieces of that boundary (i.e. edge region). The desired detection of contrasted parts in a boundary is very close in spirit to periodic subsequence detection.

Nevertheless there is a difference: the contrast of any boundary takes on real values. The former problem is solved by thresholding on the contrast. In this direction, we claim that a natural choice is $\mu_{k_{\min}}$ (see Definition 6). A maximal ε-meaningful boundary is thus converted into a periodic binary sequence.

We want to apply the periodic subsequence detection algorithm from Definitions 15 and 14 to that sequence. The only parameter left is $p = \Pr(s_i = 1), 1 \leq i \leq L$ and it is straightforward defined as $p \stackrel{\text{def}}{=} H_c(\mu_{k_{\min}})$ (see Notation 1).

We finally define the following clean-up rule:
For any meaningful boundary, keep every subcurve belonging to its maximal 1-meaningful subsequence.

This clean-up mechanism does not impose a minimal length to contrasted parts. The length is adjusted automatically, by choosing the more meaningful subsequence in the level line. As an additional advantage, there is no need to estimate any probability distribution. Figure 3.13 shows an example of the benefits of the proposed clean-up method over the one by Cao et al.. The classic version clearly produces underdetection, visually important structure is missed (notice the face in Fig. 3.13c). On the other hand, the new version produces some overdetection: small noisy parts are not eliminated but no important structure is lost. Fig. 3.14 shows two additional examples on images from the Berkeley database.

3.4 Conclusions

This work presents a novel contribution to the field of image structure retrieval. We think that the topographic map is an extremely well suited theoretical framework to perform that task. Mathematical Morphology has proved this in depth and extension with the work it developed. In that direction, we worked on the algorithm called Meaningful Boundaries. Some deep modifications are introduced in it.

First, the criterion of meaningfulness was relaxed. In the new definition, a level line can have a non-causal piece and still be considered perceptually important. We also provide an intuitive parameter that allows to deal with the length of that piece. This parameter is, as we stated above, dependent on the length and contrast of the curve, which is the natural choice.

Second, a new boundary clean-up algorithm is presented, based on Grompone et al.' multisegment detector. It benefits from some of the good properties of the new meaningful boundaries algorithm and outperforms a previous clean-up algorithm proposed by Cao et al.. Results are satisfactory. Some pieces that should be eliminated are not and some that should not are in fact eliminated. However this clean-up algorithm is an important first step and already conceptually presents all the good qualities we should expect from it.

Examples of the resulting image structure retrieval method were presented, soundly showing that its theoretical advantages are also validated in practice. The proposed method increases significantly the robustness and the stability of the

(a) IMAGE (b) MB ($\varepsilon = 1$)

(c) MB+CU (d) MB+PMS

Figure 3.13: Comparative examples of the results obtained with both clean-up algorithms. The clean-up algorithm by Cao et al. produces underdetection; the phenomenon is corrected by using periodic meaningful subsequences.

detections.

As a final remark, the maximality constraint presents some issues. All the packets of parallel level line pieces are not eliminated by it. We are currently exploring another kind of algorithm based on maximality along the gradient direction, to eliminate this effect.

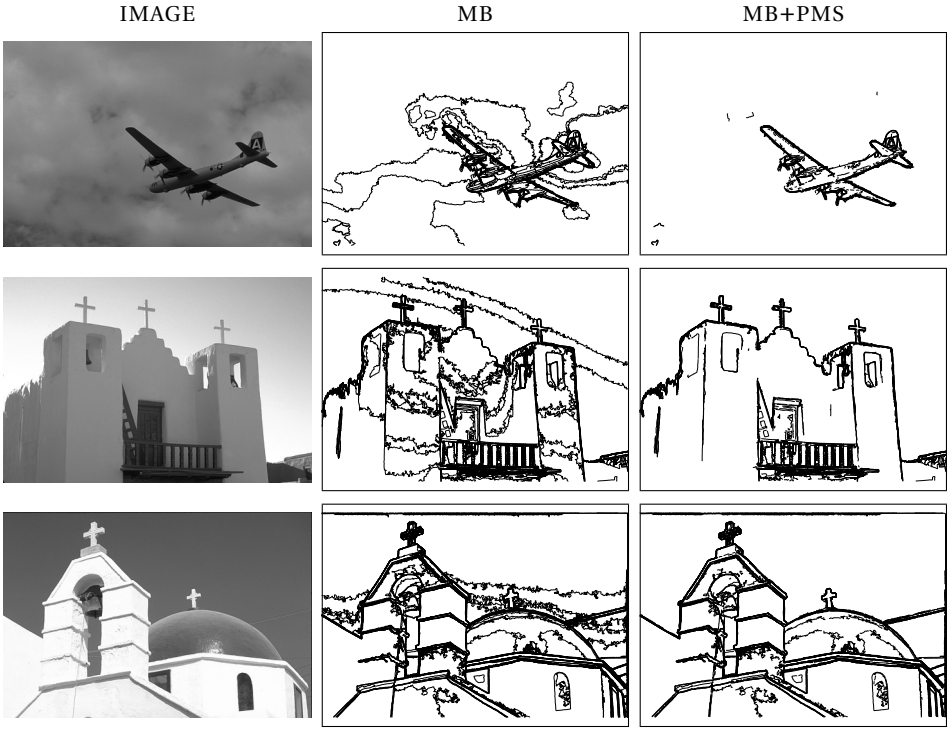


Figure 3.14: Results of the presented clean-up algorithm. On the left, the original image; on the center, its meaningful boundaries; on the right, its meaningful boundaries after clean-up.

3.A Appendix: The Incomplete Beta Function

Following the presentation in [115], the beta function, also called the Euler integral of the first kind, is a special function defined by

$$B(a,b) = \int_0^1 t^{a-1}(1-t)^{b-1}\,dt \tag{3.30}$$

for $a, b > 0$.

The incomplete beta function is a generalization of the beta function that replaces the definite integral of the beta function with an indefinite integral. The situation is analogous to the incomplete gamma function being a generalization of the gamma function.

The incomplete beta function is defined as

$$B(x;a,b) = \int_0^x t^{a-1}(1-t)^{b-1}\,dt \tag{3.31}$$

for $a, b > 0$. For $x = 1$, the incomplete beta function coincides with the complete beta function.

The regularized incomplete beta function (or regularized beta function, for short) is defined in terms of the incomplete beta function and the complete beta function:

$$I(x;a,b) = \frac{B(x;a,b)}{B(a,b)} \tag{3.32}$$

It has the limiting values

$$I(0;a,b) = 0 \qquad\qquad I(1;a,b) = 1 \tag{3.33}$$

and the symmetry relation

$$I(x;a,b) = 1 - I(1-x;b,a) \tag{3.34}$$

If a and b are both rather greater than one, then $I(x;a,b)$ rises from "near-zero" to "near-unity" quite sharply at about $x = a/(a+b)$.

Continued fractions are often powerful ways of evaluating functions that occur in scientific applications. A continued fraction looks like this:

$$f(x) = b_0 + \cfrac{a_1}{b_1 + \cfrac{a_2}{b_2 + \cfrac{a_3}{b_3 + \cfrac{a_4}{b_4 + \cdots}}}}$$

Printers prefer to write this as

$$f(x) = b_0 + \frac{a_1}{b_1+}\,\frac{a_2}{b_2+}\,\frac{a_3}{b_3+}\,\frac{a_4}{b_4+}\cdots$$

The regularized incomplete beta function has a continued fraction representation,

$$I(x;a,b) = \frac{x^a(1-x)^b}{aB(a,b)}\left[\frac{1}{1+}\,\frac{d_1}{1+}\,\frac{d_2}{1+}\cdots\right], \tag{3.35}$$

where

$$d_{2m+1} = -\frac{(a+m)(a+b+m)x}{(a+2m)(a+2m+1)}$$
$$d_{2m} = -\frac{m(b-m)x}{(a+2m-1)(a+2m)}$$

This continued fraction converges rapidly for $x < (a+1)/(a+b+2)$, taking in the worst case $O(\sqrt{\max(a,b)})$ iterations. For $x > (a+1)/(a+b+2)$ we can just use Equation 3.34 to obtain an equivalent computation where the continued fraction will also converge rapidly.

The values of a and b involved in the computations of this work end-up in systematic underflow errors. It was therefore mandatory to use an appropriate rescaling to be able to represent the probabilities correctly. The natural choice is to use the logarithm for such a task.

The term

$$\frac{x^a(1-x)^b}{aB(a,b)}$$

can be implemented in the logarithm without problems, taking advantage of the relation between the complete beta function and the gamma function [115]. The remaining term

$$\left[\frac{1}{1+}\frac{d_1}{1+}\frac{d_2}{1+}\cdots\right] \tag{3.36}$$

does not produce underflows and is not a problem.

What happens when we use Equation 3.34? Thankfully, in our case, when $I(x;a,b)$ produces underflows, $1-I(1-x;b,a)$ does not and we can directly apply the logarithm.

Finally, the last problematic case is when $x = 0$ since $I(0;a,b) = 0$ and $\log 0 = -\infty$. There is no solution to this issue. But we know that in our particular application $H_c(\mu) \neq 0$ for all μ over a level line, and we must not deal with this case.

3.B Appendix: The Mellin Transform

This section proves that meaningful contrasted regular boundaries (Definition 11) are theoretically correct, when the NFA is redefined in the following manner:

$$\mathrm{NFA}_{\mathrm{CR}}(C) \stackrel{def}{=} N_{ll}\, H_c(\mu)^{l^2/2s}\, H_s(\rho)^{l^2/2s}. \tag{3.37}$$

The Fourier transform offers powerful analytical tools to study the distribution of sums of independent random variables. Analogically, products of independent random variables can be studied using the Mellin transform. Epstein performed a thorough study of the Mellin transform [47]. We only include here the main results that are used in this work.

Definition 16. *The Mellin transform of a positive random variable ξ with continuous p.d.f. $f(x)$ is $\mathrm{E}(\xi^{s-1})$, $s \in \mathbb{C}$, where*

$$\mathrm{F}(s) = \mathrm{E}(\xi^{s-1}) = \int_0^\infty x^{s-1} f(x)\, dx. \tag{3.38}$$

Lemma 2. *Let ξ_1 and ξ_2 be two independent positive random variables with Mellin transforms $F_1(s)$ and $F_2(s)$ respectively, then the Mellin transform of the product $\eta = \xi_1\xi_2$ is $G(s) = F_1(s)F_2(s)$.*

Proof. It is immediate since

$$\begin{aligned}
\mathrm{G}(s) = \mathrm{E}(\eta^{s-1}) = \mathrm{F}(s) &= \mathrm{E}((\xi_1\xi_2)^{s-1}) \\
&= \int_0^\infty (x_1x_2)^{s-1} f(x_1x_2)\, d(x_1x_2) \\
&= \iint_0^\infty x_1^{s-1} x_2^{s-1} f(x_1) f(x_2)\, dx_1 dx_2 \\
&= \int_0^\infty x_1^{s-1} f(x_1)\, dx_1 \int_0^\infty x_2^{s-1} f(x_2)\, dx_2 \\
&= \mathrm{E}({\xi_1}^{s-1})\mathrm{E}({\xi_2}^{s-1}) \\
&= F_1(s)F_2(s).
\end{aligned} \tag{3.39}$$

□

Lemma 3. *Let ξ_1 and ξ_2 be two independent positive random variables with Mellin transforms $F_1(s)$ and $F_2(s)$ respectively, then the p.d.f. of the product $\eta = \xi_1\xi_2$ is*

$$g(y) = \int_0^\infty \frac{1}{x} f_1\left(\frac{y}{x}\right) f_2(x)\, dx.$$

Proof. It is immediate since

$$\begin{aligned}
\mathrm{G}(s) = \mathrm{E}(\eta^{s-1}) &= \int_0^\infty (t)^{s-1} g(t)\, dt \\
&= \iint_0^\infty (t)^{s-1} \frac{1}{x} f_1\left(\frac{t}{x}\right) f_2(x)\, dxdt \\
&= \iint_0^\infty (t)^{s-1} \frac{1}{x} \frac{x^{s-2}}{x^{s-2}} f_1\left(\frac{t}{x}\right) f_2(x)\, dxdt \\
&= \iint_0^\infty \left(\frac{t}{x}\right)^{s-1} f_1\left(\frac{t}{x}\right) x^{s-2} f_2(x)\, dxdt.
\end{aligned} \tag{3.40}$$

By performing the change of variables $u = t/x$

$$\begin{aligned}
\mathrm{G}(s) &= \iint_0^\infty \left(\frac{t}{x}\right)^{s-1} f_1\left(\frac{t}{x}\right) x^{s-2} f_2(x)\, dxdt \\
&= \iint_0^\infty u^{s-1} f_1(u) x^{s-1} f_2(x)\, dxdu \\
&= \mathrm{E}({\xi_1}^{s-1})\mathrm{E}({\xi_2}^{s-1}) \\
&= F_1(s)F_2(s).
\end{aligned} \tag{3.41}$$

□

As usual, ε-meaningful boundaries are correct is the following proposition holds.

Proposition 6. *The expected number of ε-meaningful contrasted regular boundaries, obtained with Equation 3.37, in a finite random set E of random curves is smaller than ε.*

Proof. We will follow the same discussion made by Cao et al. [23] for contrasted meaningful boundaries, i.e. Definition 5.

Assume that X is a real random variable described by the inverse repartition function $H(\mu) = \Pr(X \geq \mu)$. Assume that u is a random image such that the values $|Du|$ are independent with the same law as X. The same reasoning applies to $|R_s|$. Let now E be a set of random curves (C_i) in u such that $\#E$ (the cardinality of E) is independent from each C_i . For each i, we note

$$\mu_i = \min_{x \in C_i} |Du|(x)$$
$$\rho_i = \min_{x \in C_i} R_s(x).$$

We also assume that we can choose L_i independent (in contrast) points on C_i and that we can also choose L_i/s independent (in regularity) points on C_i (points that are afar at least by Nyquist's distance). We can think of the C_i as random walks with independent increments but since we choose a finite number of samples on each curve, the law of the C_i does not really matter. We assume that L_i is independent from the pixels crossed by C_i. We say that C_i is ε-meaningful if

$$\mathrm{NFA}_{\mathrm{CR}}(C_i) \overset{def}{=} N_{ll}\, H_c(\mu_i)^{L_i^2/s}\, H_s(\rho_i)^{L_i^2/s}$$

Let $X_i = \mathbf{1}_{C_i \text{ is meaningful}}$ and $N = \#E$. Let us denote by $\mathbb{E}_{\mathcal{H}_0}$ the mathematical expectation under $\mathcal{H}_0$. Then

$$\mathbb{E}_{\mathcal{H}_0}\left(\sum_{i=1}^{N} X_i\right) = \mathbb{E}_{\mathcal{H}_0}\left(\mathbb{E}_{\mathcal{H}_0}\left(\sum_{i=1}^{N} X_i \mid N = n\right)\right) \tag{3.42}$$

We have assumed that N is independent from the curves. Thus, conditionally to $N = n$, the law of $\sum_{i=1}^{N} X_i$ is the law of $\sum_{i=1}^{n} Y_i$ where $Y_i = \mathbf{1}_{n\, H_c(\mu_i)^{L_i}\, H_s(\rho_i)^{L_i/s} < \varepsilon}$. By linearity of expectation

$$\mathbb{E}_{\mathcal{H}_0}\left(\sum_{i=1}^{N} X_i \mid N = n\right) = \mathbb{E}\left(\sum_{i=1}^{n} Y_i\right) = \sum_{i=1}^{n} \mathbb{E}(Y_i)\,. \tag{3.43}$$

Since Y_i is a Bernoulli variable,

$$\begin{aligned}\mathbb{E}(Y_i) = \Pr(Y_i = 1) &= \Pr(n\, H_c(\mu_i)^{L_i^2/s}\, H_s(\rho_i)^{L_i^2/s} < \varepsilon) \\ &= \sum_{l=0}^{\infty} \Pr(n\, H_c(\mu_i)^{l^2/s}\, H_s(\rho_i)^{l^2/s} < \varepsilon \mid L_i = l)\, \Pr(L_i = l). \end{aligned} \tag{3.44}$$

Again, we have assumed that L_i is independent from the image gradient and regularity distributions. Thus conditionally to $L_i = l$, the law of $n\ H_c(\mu_i)^{L_i^2/s}\ H_s(\rho_i)^{L_i^2/s}$ is the law of $n\ H_c(\mu_i)^{l^2/s}\ H_s(\rho_i)^{l^2/s}$. Let us finally denote by $\alpha_1 \dots \alpha_l$ the l independent values of $|Du|$ and $\gamma_1 \dots \gamma_{l/s}$ the l/s independent values of $|R_s|$. We have

$$\begin{aligned}
&\Pr(n\ H_c(\mu_i)^{l^2/s}\ H_s(\rho_i)^{l^2/s} < \varepsilon) = \\
&\Pr\left(H_c(\mu_i)\ H_s(\rho_i) < \left(\frac{\varepsilon}{n}\right)^{s/l^2}\right) = \\
&\Pr\left(H_c(\min_{1\le k_1\le l} \alpha_{k_1})\ H_s(\min_{1\le k_2\le l/s} \gamma_{k_2}) < \left(\frac{\varepsilon}{n}\right)^{s/l^2}\right) = \\
&\Pr\left(\max_{1\le k_1\le l} H_c(\alpha_{k_1})\ \max_{1\le k_2\le l/s} H_s(\gamma_{k_2}) < \left(\frac{\varepsilon}{n}\right)^{s/l^2}\right) = \\
\prod_{k_1=1}^{l}\prod_{k_2=1}^{l/s} &\Pr\left(H_c(\alpha_{k_1})\ H_s(\gamma_{k_2}) < \left(\frac{\varepsilon}{n}\right)^{s/l^2}\right).
\end{aligned} \tag{3.45}$$

To continue, it is necessary to find the p.d.f of a product of independent random variables. The Mellin transform offers a solution to this problem. For commodity, we note $X = H_c(\alpha_{k_1})$ and $Y = H_s(\gamma_{k_2})$ and $Z = XY$. Then

$$\Pr\left(H_c(\alpha_{k_1})\ H_s(\gamma_{k_2}) < \left(\frac{\varepsilon}{n}\right)^{s/l^2}\right) = \Pr\left(Z < \left(\frac{\varepsilon}{n}\right)^{s/l^2}\right). \tag{3.46}$$

By using the Mellin transform, the p.d.f. of Z is

$$f_Z(t) = \int_0^\infty \frac{1}{w} f_X\left(\frac{t}{w}\right) f_Y(w)\ dw, \tag{3.47}$$

where f_X and f_Y are the p.d.f. of X and Y, respectively. We are interested in the c.d.f of Z, which is given by

$$\begin{aligned}
\Pr(Z < z) &= \int_0^z \int_0^\infty \frac{1}{w} f_X\left(\frac{t}{w}\right) f_Y(w)\ dw dt \\
\Pr(Z < z) &= \int_0^\infty \frac{1}{w} f_Y(w) \left(\int_0^z f_X\left(\frac{t}{w}\right)\ dt\right)\ dw
\end{aligned} \tag{3.48}$$

By performing the change of variables $u = t/w$

$$\begin{aligned}
\Pr(Z < z) &= \int_0^\infty \frac{1}{w} f_Y(w) \left(\int_0^{z/w} f_X(u) w\ du\right)\ dw \\
\Pr(Z < z) &= \int_0^\infty f_Y(w) \left(\int_0^{z/w} f_X(u)\ du\right)\ dw.
\end{aligned} \tag{3.49}$$

We know that $\int_0^{z/w} f_X(u)\ du = \Pr(X < z/w) \le z/w$ from Lemma 1 in Chapter 3 and

$$\Pr(Z < z) \le \int_0^\infty f_Y(w) \frac{z}{w}\ dw = z \int_0^\infty f_Y(w) \frac{1}{w}\ dw. \tag{3.50}$$

We now integrate by parts,

$$\int_0^q f_Y(w)\frac{1}{w}\,dw = \frac{1}{1}F_Y(1) + \int_0^q F_Y(w)\frac{1}{w^2}\,dw = Q(q).$$

Obviously $Q(0) = 0$ and $Q(w) \underset{w\to\infty}{\longrightarrow} 1$ since $F_Y(w) \underset{w\to\infty}{\longrightarrow} 1$, resulting in

$$\Pr(Z < z) \le z\int_0^\infty f_Y(w)\frac{1}{w}\,dw = z. \tag{3.51}$$

Reprising Equation 3.45,

$$\Pr\left(H_c(\alpha_{k_1})\ H_s(\gamma_{k_2}) < \left(\frac{\varepsilon}{n}\right)^{s/l^2}\right) \le \left(\frac{\varepsilon}{n}\right)^{s/l^2} \tag{3.52}$$

$$\prod_{k_1=1}^{l}\prod_{k_2=1}^{l/s}\Pr\left(H_c(\alpha_{k_1})\ H_s(\gamma_{k_2}) < \left(\frac{\varepsilon}{n}\right)^{s/l^2}\right) \le \prod_{k_1=1}^{l}\prod_{k_2=1}^{l/s}\left(\frac{\varepsilon}{n}\right)^{s/l^2} \tag{3.53}$$

$$= \left(\frac{\varepsilon}{n}\right)^{(s/l^2)(l/s)l} = \frac{\varepsilon}{n} \tag{3.54}$$

and finally

$$\sum_{i=1}^{n}\mathbb{E}(Y_i) = \sum_{i=1}^{n}\Pr(n\ H_c(\mu_i)^l\ H_s(\rho_i)^{l/s} < \varepsilon) \le \sum_{i=1}^{n}\frac{\varepsilon}{n} = \varepsilon. \tag{3.55}$$

□

CHAPTER

4 Shape Encoding and Matching

Abstract

In this chapter we focus on planar shape recognition which is usually addressed by encoding shapes with descriptors and matching these descriptors. We overview the shape context technique, and we present an improved version that leads to an intrinsic definition of semi-locality in this new descriptor. We then apply the a contrario shape matching framework for the case of shape contexts.

4.1 Morphological Shape Contexts

The shape context considers a sampled version of the image edge map as the shape to be encoded. The shape context of a point in the shape is a coarse histogram of the relative positions of the remaining points. The histogram bins are taken uniformly in log-polar space, making the descriptor more sensitive to positions of nearby sample points than to those farther away.

Let $\mathcal{T} = \{t_1, \ldots, t_n\}$ be the set of points sampled from the edge map of an input image. For each $t_i \in \mathcal{T}$, $1 \leq i \leq n$, the distribution of the $n-1$ remaining points in $\mathcal{T}$ is modeled relative to t_i as a log-polar histogram (Figure 4.1a). We denote by $\Theta \times \Delta$ a partition of the log-polar space $[0, 2\pi] \times (0, L]$ into A and B bins respectively, where $L = \max_{t_j \in \mathcal{T}} ||t_j - t_i||_2$. The histogram is defined as

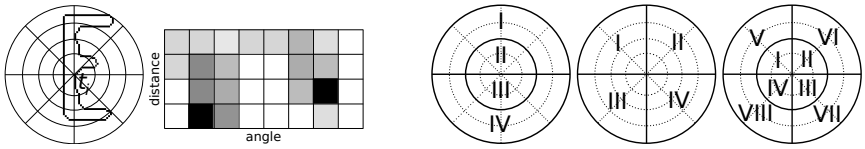
$$SC_{t_i}(\Theta_k, \Delta_m) = \#\{t_j \in \mathcal{T} \,:\, j \neq i, t_j - t_i \in (\Theta_k, \Delta_m)\}$$

where $0 < k \leq A$ and $0 < m \leq B$. The Shape Context of t_i (SC_{t_i}) is defined as a normalized version of $SC_{t_i}(\Theta_k, \Delta_m)$.

Figure 4.1a depicts both spatial and matrix representations of a shape context.

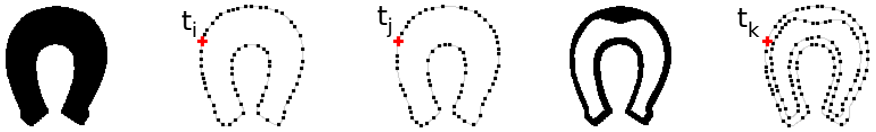
The collection of the shape contexts for every point in the shape is a redundant and powerful descriptor for that shape but has some drawbacks.

First, the sampling stage is performed by considering that the edge map corresponds to a Poisson process [12]. This hard-core model produces a non-deter-



(a) (b)

Figure 4.1: (a) Shape context of a character 'E'. Left, partition into bins around the point t_i; right, matrix representation of SC_{t_i} (darker means more weight). (b) Different ways to split a shape context. Doted lines separate bins and thick lines separate bin groupings.



(a) (b) (c) (d) (e)

Figure 4.2: (a) image horsehoe1; (b) sampled points from horsehoe1; (c) other sampled points from horsehoe1, with the same sampling process than those in (b); (d) image horsehoe2; (e) sampled points from horseshoe2 with the same sampling process than those in (b) and (c). The points t_i, t_j ad t_k are in the same position of the image.

ministic sampling algorithm which means that different runs of the sampling algorithm may give slightly different results. The immediate consequence is that two descriptors from exactly the same image, obtained at different times, may not be equal. In short terms, jitter noise is introduced in the descriptor. In Figure 4.2 the effect of the jitter noise is shown, making $d(SC_{t_i}, SC_{t_j}) \approx 0.11 \neq 0$ [1].

Second, from our point of view the main drawback of shape context is that it inherits the weaknesses from the edge map. We mentioned previously that extracting curves from the edge map is a hard problem. This fact has a great impact in shape encoding: there is no intrinsic distinction between what is global and what is not. An example is shown in Figure 4.2, where $d(SC_{t_i}, SC_{t_k}) \approx 0.3$ which is clearly above the jitter noise $d(SC_{t_i}, SC_{t_j})$. In short terms, a slight modification of the shape has a great impact on the distance. The question "Where does a shape begin and where does it end?" becomes absolutely non trivial. The efforts to overcome this issue lead to heuristic solutions.

As stated above, the topographic map provides a natural solution to these is-

[1] $d(\cdot,\cdot)$ is the χ^2 distance and is used throughout this paper

sues. Meaningful boundaries are much more suitable than the edge map for shape recognition. Meaningful boundaries are used as the set of shapes to be encoded and recognized from an image [22]. Maximal Stable Extremal Regions (MSER), which are very close in spirit to MB, have also been used for shape encoding [112].

The main idea is to exploit the benefits of the image structure representation defined in the previous section and to fuse it with shape context. We call this new descriptor Morphological Shape Context (MSC).

As in shape context, each shape in a given image is composed by a set of points. In MSC, we consider each curve (i.e. meaningful boundary) as a shape. When dealing with curves, the sampling stage is done in a very natural way, by arc-length parameterization, thus eliminating jitter noise. See Algorithm 1. In the resulting algorithm, shapes are extracted using the MB algorithm. Notice that we also might chose the MSS algorithm, see Chapter 2 Section 2.1.3. Let us redefine $\mathcal{T} = \{t_1, ..., t_n\}$ as the set of points sampled from a meaningful boundary of an image. The shape context is then computed for each sample point t_i, $1 \leq i \leq n$.

Algorithm 1 Compute the MSC of an image u

```
C ←extractCurves(u)
for all T ∈ C do
    T_S ←sample(T)
    for all p_i ∈ T_S do
        for all 0 < k ≤ A, 0 < m ≤ D do
            compute SC_{t_i}(α_k, d_m)
        end for
        normalize SC_{t_i}(α_k, d_m)
    end for
end for
```

Beside the advantages of the representation we described above, one of its keys is the natural separation between level lines (they do not intersect). It allows to go from a global shape encoding to a semi-global one in a natural way, i.e. without fixing any arbitrary threshold. The most powerful advantage is that individual objects present in the image can be matched separately, which was not possible in shape context.

The Level Line Descriptor [22] was designed to detect that two images share *exactly* the same shape. The "perceptual invariance" is only introduced in the matching stage. That is not what we are aiming for. We want to keep the intrinsic "perceptual invariance" given by the shape context and be able to detect that two images share two *similar* shapes, independently of the matching algorithm.

4.2 A Contrario Shape Context Matching

We reproduce the notation from Section 2.3 in Chapter 2,

Let $\mathcal{F} = \{F^k \mid 1 \le k \le M\}$ be a database of M shapes. For each shape $F^k \in \mathcal{F}$ we have a set $\mathcal{T}^k = \left\{t_j^k \mid 1 \le j \le n_k\right\}$ where n_k is the number of points in the shape. Let $SC_{t_j^k}$ be the shape context of t_j^k, $1 \le j \le n_k$, $1 \le k \le M$. For simplicity, we denote

$$\mathcal{S} = \left\{SC_{t_j^k} \mid t_j^k \in \mathcal{T}^k,\ F^k \in \mathcal{F}\right\}. \tag{4.1}$$

Let us also suppose that we have a suitable distance $d(\cdot,\cdot)$ between Shape Contexts. We follow the choice made by Belongie et al. [12] of the χ^2 test statistic to compare two shape contexts SC, SC'.

$$d(SC, SC') = \frac{1}{2}\sum_{k=1}^{K} \frac{\left[SC(k) - SC'(k)\right]^2}{SC(k) + SC'(k)}, \tag{4.2}$$

where $SC(k)$ denotes the k-th bin of C. This is a classical choice when comparing histograms.

All these efforts aim at reducing the number of false correspondences but are not truly successful: none of the above methods gives a clear-cut answer to the problem of deciding if two descriptors are similar. In our particular case, we need to find out whether two shapes look alike or not.

Cao et al. [22] and then Rabin et al. [116] shown that the *a contrario* framework is specially well suited for matching shape and SIFT descriptors respectively. The a contrario detection framework is based on the Helmholtz Principle that, for our application, states that a match is meaningful when it is not likely to occur in a context where noise overwhelms the information. In particular this section summarizes the concepts introduced by Tepper et al. [132] for matching shape contexts.

The aforementioned framework is specially suited for shape matching. Let $\{SC_i \mid 1 \le i \le n\}$ and $\{SC'_j \mid 1 \le j \le m\}$ be two sets of shape contexts from two different shapes. We want to see if both shapes look alike. The distances between SC_i and SC'_j can be seen as observations of a random variable D that follows some unknown random process.

What we would really want to do is to perform an hypothesis test, for each pair (SC_i, SC'_j) and some appropriately chosen distance $d(\cdot,\cdot)$, where
$\mathcal{H}_0$*: a small* $d(SC_i, SC'_j)$ *is observed due to a realization of randomness, i.e. because the database is large.*
$\mathcal{H}_1$*: a small* $d(SC_i, SC'_j)$ *is observed because of some causality, i.e. because the shapes look alike.*

On one hand, $P(D \mid \mathcal{H}_0)$ can be modeled with relative ease, even if the model is not perfectly realistic. On the other hand, it is not possible to model $P(D \mid \mathcal{H}_1)$ because we assume no other information but the observed set of features. Hence, the full hypothesis test can not be done: we can not control type II errors.

However controlling type I errors, i.e. the number of false correspondences under $\mathcal{H}_0$, is enough to make a sound answer to our decision problem. In other

words, low probabilities under $\mathcal{H}_0$ are not likely to happen by chance and are, on the contrary, causal.

We define the distance between two shape contexts and estimate the probability of occurrence of a given match under $\mathcal{H}_0$. It is essential [106] to split the shape context into independent features (its importance will be clarified in Section 4.2.1).

We assume that each shape context is split in C independent features [132] that we denote $SC^{(i)}_{t_j^k}$ with $1 \leq i \leq C$ (see Figure 4.1b for an example).

Let Q be a query shape and q a point of Q. We define

$$d_j^k = \max_{1 \leq i \leq C} d_j^{k(i)} \tag{4.3}$$

$$d_j^{k(i)} = d(SC_q^{(i)}, SC^{(i)}_{t_j^k}). \tag{4.4}$$

We can now formally state the a contrario hypothesis

$\mathcal{H}_0$: *the distances $d_j^{k(i)}$ are observations of C identically distributed independent random variables $D^{(i)}$, $1 \leq i \leq C$ that follow some stochastic process.*

The matching algorithm consists in adequately rejecting $\mathcal{H}_0$. Then the probability of false alarms is defined as

$$P(D \leq \delta \mid \mathcal{H}_0) = P(\max_{1 \leq i \leq C} D^{(i)} \leq \delta \mid \mathcal{H}_0) \tag{4.5}$$

$$= \prod_{i=1}^{C} P(D^{(i)} \leq \delta \mid \mathcal{H}_0) \tag{4.6}$$

$P(D^{(i)} \leq \delta \mid \mathcal{H}_0)$ represents the likeliness of occurrence of a distance lower than δ under $\mathcal{H}_0$ in $\mathcal{F}$.

Definition 17. *The number of false alarms (NFA) of the pair (q, t_j^k) in the database $\mathcal{F}$ is*

$$\mathrm{NFA}(q, t_j^k) \overset{def}{=} \left(\sum_{k'=1}^{M} n_{k'} \right) \cdot \prod_{i=1}^{C} P(D^{(i)} \leq d_j^k \mid \mathcal{H}_0). \tag{4.7}$$

We say the pair (q, t_j^k) is an ε-meaningful match if $\mathrm{NFA}(q, t_j^k) < \varepsilon$.

The probabilities $P(D^{(i)} \leq d_j^k \mid \mathcal{H}_0)$ can be estimated as the cumulative histograms of the distances $d_j^{k(i)}$, $1 \leq i \leq C$, $1 \leq k \leq M$ and $1 \leq j \leq n_k$.

This provides a simple rule to decide whether a single pair (q, t_j^k) does match or not. From one side, this is a clear advantage over other matching methods since we have an individualized assessment for the quality of each possible match. From the other side, the threshold is taken on the probability instead of directly on the distances. Setting a threshold directly on the distances d_j^k (or $d_j^{k(i)}$ for the case) is hard, since distances do not have an absolute meaning. If all the shapes in the database look alike, the threshold should be very restrictive. If they differ significantly from each other, a relaxed threshold would suffice.

Thresholding on the probability is more robust and stable. More stable, since the same threshold is suitable for different database configurations. More robust, since we explicitly control false detections. The expected number of ε-meaningful matches in a random set of random matches can be proven to be smaller than ε [22].

4.2.1 Partitioning the Shape Context

As stated above, the features in which the shape context is split must be independent to go from Equation 4.5 to Equation 4.6.

A shape is represented by a set of sample points drawn from the contours of an object. Belongie et al. perform a somewhat uniform sampling along the contour. This responds to the assumption that the points follow a Poisson process [119]. This is a fundamental property to take advantage of when splitting the shape context.

The shape context can be directly split by grouping its bins. Since the points are assumed to be uniformly distributed, any way to group the bins (without overlapping), produce a set of independent features. Figure 4.1b shows different ways to split the shape context (the 2D polar histogram boundaries are indicated by thick lines). We can see each group indicated with roman numerals: for $C = 4$ one can half distances and angles or only partition angles, for $C = 8$ distances are halved and angles are split in four quadrants.

Once we know that the shape context can be split, the question is: why is it necessary to split it? The probabilities $P(D^{(i)} \leq \delta \mid \mathcal{H}_0)$ are estimated in practice using the cumulated histograms of the distances $d_j^{k(i)}$. Each bin can be at least $1/N$. If we take the number of features $C = 1$, from Def. 17 the NFA of any pair of features is greater than $N \cdot 1/N = 1$. This means that on the average we would have at least one false alarm per query, which is not by any means an acceptable bound.

It is therefore important to choose $C > 1$, so that the NFA of any pair of features is greater than $N \cdot 1/N^C = 1/N^{C-1}$. This means that we can reach lower values for the NFA by splitting the shape context into independent features.

4.3 Discussion

In this section we illustrate the performance of the presented methods with three different examples. All the experiments in this paper were produced using $\varphi = 0.02$ for the computation of MB. In both *a contrario* algorithms taking $\varepsilon = 1$ should suffice but we set $\varepsilon = 10^{-10}$ for MB and $\varepsilon = 10^{-2}$ for matching to show the degree of confidence achievable without affecting the results. We also include results using different base shapes than MB, i.e. MSB (see Chapter 2 Section 2.1.3).

In the first example, we tested the approach in a video sequence from South Park, which is textureless and composed only by contours. In Figure 4.3, meaningful matches between two consecutive frames are depicted. White dots represent

the centers of the MSC. In Figure 4.3c, both frames are overlapped to show moving shapes. Note that in Figures 4.3a and 4.3b there are no matches in these areas.

The second example, displayed in Figure 4.4, is closely related to the first one. Here texture is present and a non-rigid character is moving on the foreground. The matches between frames 3 and 4 of the sequence are shown. Only shapes not occluded by the movement are matched. The channel logo is correctly matched since it is located in the foreground and it is not affected by any motion.

Finally, in Figure 4.5 an application to content-based video retrieval is shown. We searched for the parental guidance logo in a video sequence with 8434 frames, see Figure 4.4a for example frames. Figure 4.5b depicts the number of matches for each frame of the video. The logo is present in three intervals ([0, 76], [2694, 2772] and [4891, 4969]) which coincide with the three spikes. These spikes are clearly higher than spurious matches in the rest of the video. The second and third spike are smaller than the first one, in those intervals the logo is only at 66% of its original size. This is achieved without any multiscale processing. The size of the shape context adapts naturally to the size of the encoded level line; this introduces limited scale invariance to the method. The limitation can be simply explained by stating that the image topology changes with blur and hence with zoom. However, the method is fairly resisting to zooms when the zooming effect is not too drastic

In Figure 4.5c the best match (the correct one) has a NFA of $2.45 \cdot 10^{-9}$ and the worst one (the wrong one), of $9.99 \cdot 10^{-3}$. At $\varepsilon = 10^{-4}$ all matches are correct.

The same experiment as in Figure 4.5c using shape context gives 3 matches instead of the 29 obtained using MSC (Figure 4.5d). All MSC matches are correct and all shape context matches are wrong: the global shape context approach is unable to match semi-local shapes.

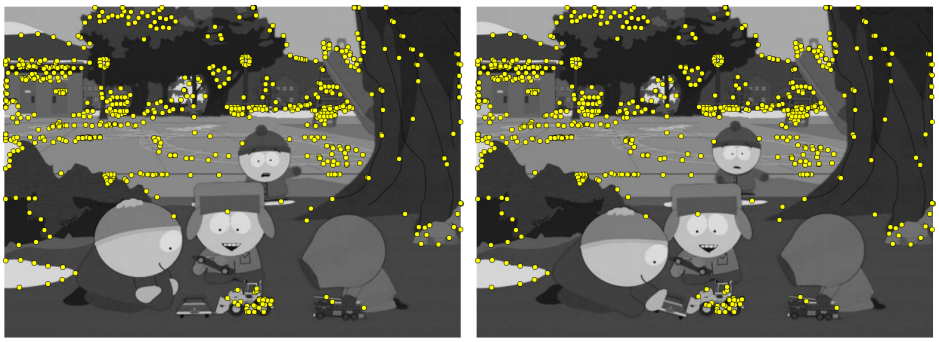
The examples show that semi-locality in the MSC is a key feature to match shapes in contexts where other shapes are present: when very similar images present little differences (Figure 4.3), when different foregrounds occlude the same background (Figure 4.4), when the query is not present or surrounded by a large set of shapes (Figure 4.5). MSC provides a novel approach to deal with such contexts, proving itself successful where shape context is not.

4.4 Future work

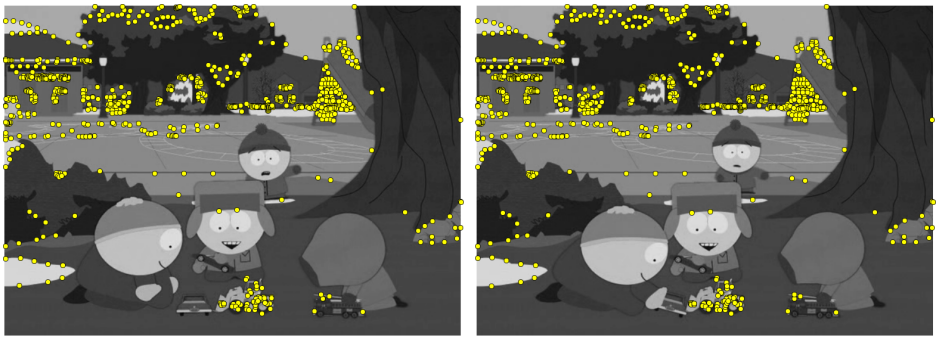
The present chapter opens a fair number of possibilities for further extensions.

Throughout this chapter we used the classical χ^2 statistic test as a distance to compare shape contexts. Obviously this is not the only available choice. The χ^2 distance relies completely on the accuracy of the binning process. In general, this is not the case since noise (or other small perturbations) may cause points to move from a bin to another.

The Earth mover's distance (EMD) is used in probability theory to establish a measure of dissimilarity between two (normalized) probability distributions. Informally, if the distributions are interpreted as piles of dirt, the EMD is the mini-



(a) MSC encoded from meaningful boundaries

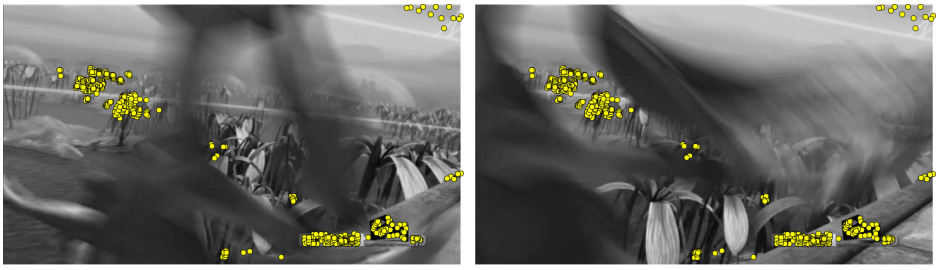


(b) MSC encoded from maximally stable boundaries

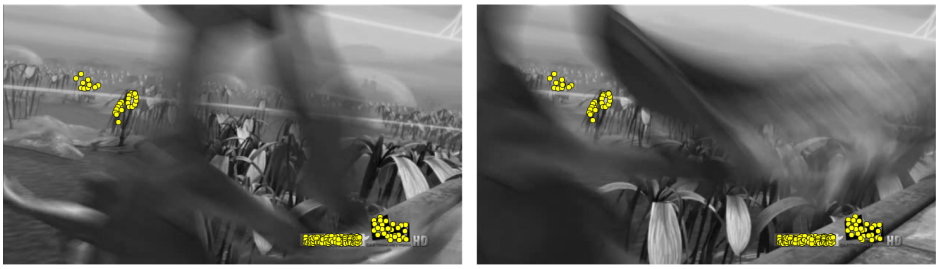
(c) Overlapped frames

Figure 4.3: There are 874 and 901 matches coherent with a similarity transformation in Figures (a) and (b) respectively. Both frames are shown overlapped to show moving shapes.

(a) Video sequence



(b) MSC encoded from maximally stable boundaries



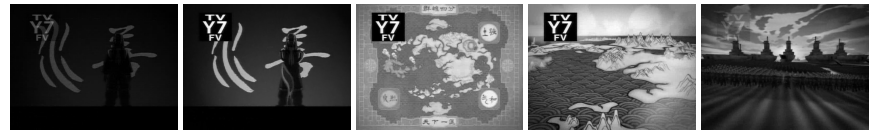
(c) MSC encoded from meaningful boundaries

Figure 4.4: A video sequence with a non-rigid character moving on the foreground. The channel logo is in the bottom right. Matching between frames 3 and 4: there are 193 and 551 meaningful matches (yellow dots) coherent with a similarity transformation in Figures (c) and (b) respectively.

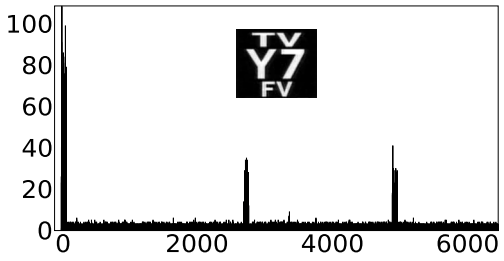
mum cost of moving the dirt to turn a pile into the other. If the domain is discrete, e.g. as histograms, the computation of the EMD becomes an instance of a transportation problem, which can be solved by the Hungarian algorithm.

Rabin et al. [116] proposed a circular version of the EMD, which is specially suited for angular distributions. It was originally designed for matching SIFT descriptors, which share a loosely similar binning process with shape contexts. The authors shown that the circular EMD significantly improves the matching process. Hence it is natural and straightforward to match shape contexts using the circular EMD.

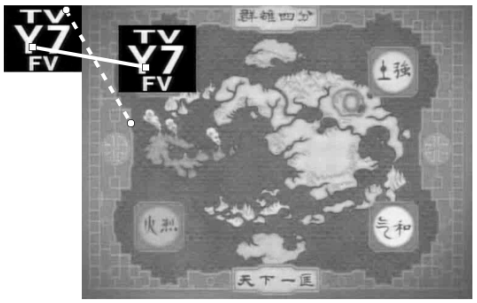
From a different point of view, psychologists have long acknowledged that human perception assigns more importance to points where the curvature is high [5]. More recent studies (see Appendix 4.A) support this claim and extend it by stating that concavities are perceptually more salient than convexities. This suggests that a distance between shapes should account for these phenomena.



(a)



(b)



(c)

(d)

(e)

Figure 4.5: (a) Example video frames where the parental logo fades in and then fades out. (b) Number of matches per frame of a video with the displayed query. (c) Best (solid line) and worst (dashed line) matches for a target frame. (d) Detail of the logo area with matched points in black dots over the meaningful boundaries. (e) Another frame with matched points in colors dots over the meaningful boundaries.

Another major line for improvement is to extend the degree of invariance of shape contexts. These descriptors are naturally translation invariant and partially scale invariant. The partial scale invariance can be easily explained: although the size of the descriptor adjust to the shape to encode, no blur is taken into account. This kind of solution, as shown by Musé et al. [105], is not fully scale invariant as blur changes the image topology. The degree of invariance can be augmented by the following ways

zoom Shape contexts can become fully scale invariant by embedding the level line detection process into a multiscale approach. This kind of approach is the only real solution to scale invariance and has been extensively explored with successful results.

rotations SIFT descriptors are made rotation invariant by rotating the hole descriptor using a locally privileged orientation. Rabin et al. [116] have robustified this approach by choosing such orientation using a contrario techniques.

affinities Classically, full affine invariance was considered computationally prohibitive. Recently, Yu and Morel [99] showed that this belief is false and proposed a method to simulate affine parameters. Along with the continual growth in computational power, this method allows for full affine invariance, while using similarity invariant descriptors.

We used MSC with maximally stable boundaries and with meaningful boundaries. These two formulations could be merged and one could compare directly MSC from both representations. We already mentioned other descriptors (e.g., SIFT and LLD) and in this direction it would be possible to use all sort of descriptors and combine them in a final and global decision step.

In any shape/object recognition method, there is a final check for spatial consistency. In general this check is crucial since the matching methods do not directly control false matches. The check for spatial consistency prunes the set of matches from these undesired ones. The already classical approach is to use RANdom SAmple Consensus (RANSAC) [51], where a parametric transformation is defined a priori and the largest subset supporting the same transformation is chosen.

In this section, the experiments were performed using exactly this scheme with similarity transformations. Notice that even if we control the expected number of false alarms, false alarms can occur anyway. Two shapes from two different images can be strikingly similar but not actually correspond to the same physical 3D object. To mention a simple 2D example, characters 'm', 'E' and '3' could be matched, depending on the typography, and not be false matches from a strict shape criterion.

The need for spatial coherence is undoubtful. RANSAC approaches fail when one wishes to detect multiple instances of the the same object in the same image. The iteration of the RANSAC process is the usual path to cope with this requirements. An alternative approach is to cluster the set of matches. All descriptors have information about the nature of the object they encode (e.g. scale, orientation, location). This information was successfully incorporated into the coherence

check by Cao et al. [22] using a clustering algorithm. Then, it would be natural to group coherent shape contexts by using the clustering techniques we presented in the first part of this thesis. The only reason for not presenting these experiments are schedule constraints...

4.A Appendix: Information along contours

Feldman and Singh [49] have made a quantitative analysis from the viewpoint of information theory of Attneave's claim:

> Information is concentrated along contours (i.e. regions where color changes abruptly), and is further concentrated at those points on a contour at which its direction changes most abruptly (i.e. at angles or peaks of curvature). (Attneave [5], 1954)

Here we briefly explain their work. According to Shannon's formulation, the surprisal of a measure x is

$$u(x) = -\log p(x) \tag{4.8}$$

where $p(x)$ is the distribution of x.

We consider now the case of a planar curve parameterized by its arc length. Let L be its length, we take n uniformly spaced points. The arc length between them is obviously $\Delta s = \frac{L}{n}$. From point to point, the tangent changes by some angle $\alpha \in (-\pi;\pi)$. We adopt the convention that positive angles correspond to clockwise turns.

If we want to know whether some observed values of α are informative, we must first assume a prior distribution for $p(\alpha)$. A natural choice could be the von Mises distribution centered on $\alpha = 0$, that is,

$$p(\alpha) = A \exp(b \cos\alpha) \tag{4.9}$$

where b is a parameter for the spread of the distribution and A is a normalizing constant depending only on b.

Combining Equation 4.8 and 4.9 we get

$$u(\alpha) = -\log p(\alpha) = -\log A - b\cos\alpha, \tag{4.10}$$

that is,

$$u(\alpha) \propto -\cos\alpha. \tag{4.11}$$

We define now the curvature κ as

$$\kappa \approx \frac{\alpha}{\Delta s} \tag{4.12}$$

and thus $p(\kappa)$ follows a von Mises distribution with mean 0 and spread parameter $b(\Delta s)^2$,

$$p(\kappa) \approx A' \exp\left[b(\Delta s)^2 \cos(\kappa \Delta s)\right]. \tag{4.13}$$

To understand intuitively this scaling of b, $\frac{1}{b}$ can be regarded as the equivalent of the variance in a Gaussian distribution. Therefore the surprisal of κ is

$$u(\kappa) \approx -\log A' - b(\Delta s)^2 \cos(\kappa \Delta s), \tag{4.14}$$

that is,

$$u(\kappa) \propto -\cos(\kappa \Delta s). \tag{4.15}$$

The cosine function decreases monotonically from 0 to π, and thus $u(\alpha)$ and $u(\kappa)$ increase monotonically. We finally obtain a proof that the surprisal of a point in a curve (or the information coded by that point) increases monotonically with the magnitude of the curvature (as cosine is a symmetric function).

A similar argument can be presented for closed curves, where the total turning angle must add up to 2π. A more natural choice for the expected value would now be $\frac{2\pi}{n}$. This yields

$$p(\alpha) = A\exp\left[b\cos\left(\alpha-\frac{2\pi}{n}\right)\right] \tag{4.16}$$

$$u(\alpha) = -\log A - b\cos\left(\alpha-\frac{2\pi}{n}\right) \tag{4.17}$$

$$u(\kappa) \approx -\log A' - b(\Delta s)^2\cos\left(\kappa\Delta s-\frac{2\pi}{n}\right) \tag{4.18}$$

The $u(\kappa)$ is minimal when α turns $\frac{2\pi}{n}$ towards the interior of the curve. Points with negative curvature κ are now more surprising (carry more information) than points with positive curvature.

In this same direction, Barenholtz et al. [8] made a psychophysical series of tests that adds light to the relative importance of convex and concave parts of a closed curve, see Figure 4.6. Humans are more sensitive to changes in concavities. Other studies from the same research group support this claim [9].

This would suggest that, when calculating the distance between two curves, the distance should be more sensitive to differences in concavities. This would lead to the definition of a new "visually adapted" curve distance.

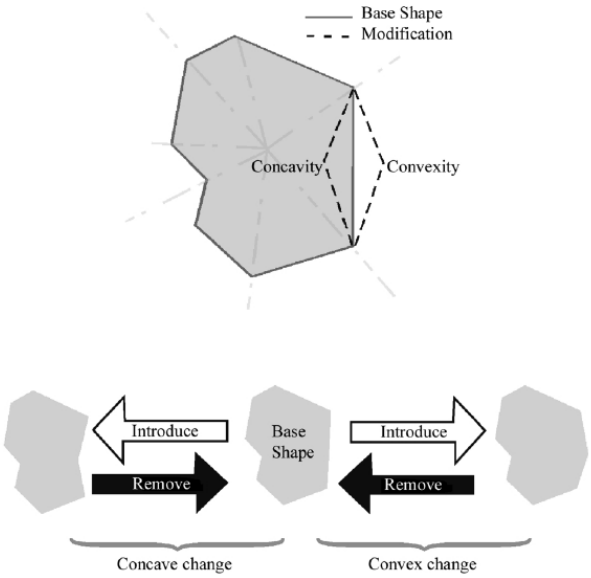


Figure 4.6: Sketch of the psychophysical experiment made by Barenholtz et al.. Their results indicate that human perception is more sensitive to changes in concavity than in convexity. Reproduced from [8].

Part II

The proximity gestalt: a computational quest

CHAPTER

5 Clustering Review

Abstract

In this chapter we briefly review different clustering techniques from the two main approaches: partitional and hierarchical clustering. Among partitional methods, spectral clustering, Mean Shift and graph-based approaches are covered. We finally overview different approaches to the problem of cluster validation.

Clustering is an unsupervised learning method in which a set of observations is assigned into subsets (called clusters) so that observations within the same cluster are similar in some sense. Clustering is an interesting problem for many domains, such as image and signal analysis, bioinformatics or medical sciences. It has been applied to image segmentation [110, 35, 50], object class and shape recognition [81, 21], gene network analysis [41] and internet databases analysis [41, 52], among others. Nowadays, the need for exploratory data analysis has become of extreme importance due to the increase in both volume and variety of data. Many domains are in need of computational techniques that do not rely on strong a priori knowledge.

Despite its intuitive simplicity, it is extremely hard to provide a formal definition of what a cluster is. Different authors provide different definitions. Very often definitions are derived from the algorithm being used, rather than the opposite.

Unfortunately, the lack of a unified definition makes it difficult to find a unifying clustering theory. A plethora of methods to assess or classify clustering algorithms have been developed, some of them with very interesting results. To cite a few [77, 71, 24]. For a broad perspective of clustering techniques, we refer the reader to the excellent overview of clustering methods recently reported by Jain [67].

Formally, we want to find clusters in a feature set $X = \{x_i \in \mathbb{R}^H\}_{i=1\ldots N}$ where H is the dimension of the feature space. We assume X is embedded in a metric space

with metric d. In the following the terms point and feature have the same meaning and we will use one or the other depending on the context.

5.1 Partitional Clustering

The goal of partitional clustering is to identify the partition $\mathcal{P}$ that optimizes a criterion function. Usually, the partition size $k = |\mathcal{P}|$ is given. We denote by P_i the i-th partition of $\mathcal{P}$ for $1 \leq i \leq k$.

We will not address parametric methods as mixture decomposition since we assume no a priori knowledge on the underlying probability density function. The standard method for solving these, from our point of view, restricted problems is Expectation-Maximization (EM) algorithm [15].

The simplest and most widely used family of criterion functions is the one of related minimum variance criterion [45]. The energy to be minimized is

$$E = \frac{1}{2} \sum_{m=1}^{k} |P_m| \, \langle d_m \rangle, \tag{5.1}$$

where $\langle d_m \rangle$ is the average distance between points in the m-th cluster, given by

$$\langle d_m \rangle = \frac{1}{|P_m|^2} \sum_{x_i \in P_m} \sum_{x_j \in P_m} d(x_i, x_j). \tag{5.2}$$

If d is the squared Euclidean distance, it is equivalent to minimize the sum of squared errors over $\mathcal{P}$

$$J(\mathcal{P}) = \sum_{m=1}^{k} \sum_{x \in P_m} \| x - \langle P_m \rangle \|_2^2, \quad \text{where} \quad \langle P_m \rangle = \frac{1}{|P_m|} \sum_{x \in P_m} x. \tag{5.3}$$

When k grows, the squared error $J(\mathcal{P})$ decreases [67], hence it can be minimized only for a fixed number of clusters. The minimization of $J(\mathcal{P})$ is known to be an NP-hard problem [43]. Hence, it is usually solved with an iterative greedy algorithm called k-means [67]. This is probably the most widely known clustering algorithm.

Strictly speaking, this criterion only makes sense when clusters are isotropic, multivariate normally distributed. But this assumption is often overlooked in practice.

In the following we present some algorithms that reflect modern views of partitional clustering, allowing to detect arbitrarily shaped clusters or to avoid specifying the number of clusters.

5.1.1 Spectral Methods

A thorough overview of the spectral graph theory was presented by Chung [33]. Here we briefly recall spectral graph concepts used for clustering.

Let $G = (V, E)$ be an undirected graph with a vertex set $V = \{v_i\}_{i=1...N}$, where N is the cardinal of V, and an edge set $E \subseteq V \times V$. We consider graphs to be non-reflexive, i.e. $\forall\ v \in V,\ (v, v) \notin E$. We also define an edge weighting function $\omega : E \rightarrow \mathbb{R}$ such that $\forall\ e \in E,\ \omega(e) > 0$.

We denote by $W \in \mathbb{R}^{N \times N}$ the matrix defined by $W_{ij} = \omega((v_i, v_j))$, $1 \leq i, j \leq N$. We denote by $D \in \mathbb{R}^{N \times N}$ the diagonal matrix such that $D_{ii} = \sum_j W_{ij}$. The Laplacian of G is defined to be the matrix

$$\begin{aligned} \mathcal{L} &= D^{-1/2}(D - W)D^{-1/2} \\ &= I - D^{-1/2}WD^{-1/2}. \end{aligned} \tag{5.4}$$

$\mathcal{L}$ is symmetric positive semidefinite [110] and its eigenvalues lie in the interval $[0, 2]$. Von Luxburg [137] analyzed the advantages of this form of the Laplacian.

Ng et al. [110] and Fowlkes et al. [54] studied the utility of employing multiple eigenvectors of $\mathcal{L}$ to embed each feature into an M-dimensional space ($M \ll N$). To build the embedding, we compute the $N \times N$ matrices A and Λ such that $\left(D^{-1/2}WD^{-1/2}\right)A = A\Lambda$ and the values $\lambda_i = \Lambda_{ii}$ are sorted in ascending order. The columns of A are the eigenvectors of $\mathcal{L}$ and $1 - \lambda_i$ are its eigenvalues. The M-dimensional embedding is the result of keeping the first M columns of A forming the matrix A_M. The normalized form of A_M is defined as

$$\overline{A}_M \stackrel{\text{def}}{=} D^{-1/2}A_M. \tag{5.5}$$

The resulting spectral clustering procedure is as follows:

1. build G_o from $V = X$ using $\omega((v_i, v_j)) = d(x_i, x_j)$ where d is an application specific kernel distance,
2. compute the embedding $\overline{A}_M$ from G_o,
3. apply a clustering algorithm to the rows of $\overline{A}_M$, as they form tight clusters.

Each feature is finally assigned to the cluster to which its corresponding row belongs. This procedure provides a relaxation of minimizing the Normalized Cut of the original graph [124], given by

$$\text{NCut}(C_1, C_2) \stackrel{\text{def}}{=} \frac{\text{f}(C_1, C_2)}{\text{f}(C_1, V)} + \frac{\text{f}(C_1, C_2)}{\text{f}(C_2, V)} \tag{5.6}$$

where $C_1, C_2 \in V$, $C_1 \cap C_2 = \emptyset$ and $\text{f}(C, C') = \sum_{u \in C, v \in C'} \omega((u, v))$. This criterion simply states that intra-cluster weights must be minimized with respect to inter-cluster ones.

Ng et al. [110] and Fowlkes et al. [54] use k-means for the final clustering stage. It is well known that k-means presents two drawbacks: (1) it is very sensitive to the initialization and (2) the number of clusters has to be manually specified. The former is usually addressed by performing several random initializations and then choosing the optimal partition with respect to Ncut, see Figure 5.1. The latter is an open question as choosing k is a difficult model-selection problem. Popular approaches to circumvent this issue are Akaike's Information Criterion

(AIC), Bayesian Information Criterion (BIC) [18] or Minimum Description Length (MDL) [10]. While these methods rely on firm theoretical background, results usually differ from a human made choice.

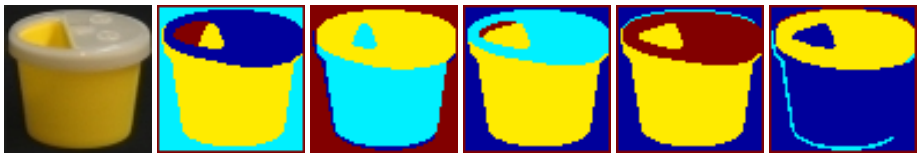


Figure 5.1: Example of image segmentation from 3×3 color patches, obtained using Normalized Cuts and k-means with $k = 4$. In each example k-means was executed 5 times and the segmentation that minimizes the Normalized Cut is chosen. Due to the clusters difference in density and in cardinality, the method yields unstable results.

Dhillon et al. [41] showed that solving this eigenvector problem is equivalent to computing kernel k-means with a suitable kernel. This formulation obviously inherits the aforementioned problems of k-means.

Yu and Shi [142] developed a discretization algorithm for the final clustering step, which seeks the discrete solution closest to the continuous optimum by rotating the normalized eigenvectors. In their algorithm the number of clusters is an input parameter and must be equal to the dimensionality of the embedding; this choice does not necessarily lead to optimal results. Moreover, as pointed out by Zelnik-Manor and Perona, this iterative method can easily get stuck in local minima and thus does not reliably find the optimal alignment [144]. This claim was also confirmed in our experiments.

An algorithm were the number of clusters is not specified by the user was introduced by Zelnik-Manor and Perona [144]. They designed a cost function to test the degree of alignment of $\overline{A}_M$ with the coordinate axes. Then, they align $\overline{A}_M$ with the coordinate axes by computing the minimum-cost rotation. Finally, for all m from 2 up to M, the minimum cost function of the first m columns of $\overline{A}_M$ is computed and the number of clusters is set to m. Unfortunately, the algorithm always return between 2 up to M clusters.

Ozertem et al. [113] propose to use the Mean Shift algorithm to build the adjacency matrix. The matrix is built by using the clusters found with Mean Shift for determining locality, instead of directly using a kernel distance.

As a final word, Nadler and Galun [107] showed that spectral methods may not reveal clusters of different sizes and scales. This assertion holds when clusters are intertwined or sufficiently near each other. However in practice, when clusters (even of different sizes and scales) are well separated, spectral methods perform well.

5.1.2 Mean Shift

The rationale behind the density estimation-based nonparametric clustering approach is that the feature space can be regarded as the empirical probability density function (p.d.f.) of the represented parameter. Dense regions in the feature space thus correspond to local maxima of the p.d.f., i.e. to the modes of the unknown density.

Density estimation can be performed by using kernels. This technique is also known as Parzen windows [57]. The kernel density estimator of the p.d.f f is given by

$$\hat{f}(x) = \frac{1}{N}\sum_{i=1}^{N} K_{\sigma_i}(x - x_i), \tag{5.7}$$

where K_{σ_i} is a suitable kernel. Usually uniform or Gaussian kernels are used but other choices are possible.

From the above equation, local maxima of f can be estimated. These local maxima are stationary points of a gradient ascent algorithm. This ascendent path is obtained by performing the so-called mean-shift iterations [35]. First, an initial estimate $\hat{x}$ is chosen. In each iteration the empirical mean

$$m(x) = \frac{1}{N\hat{f}(x)}\sum_{i=1}^{N} x_i K_{\sigma_i}(x - x_i) \tag{5.8}$$

is computed and we set $\hat{x} = m(x)$. The iteration of these simple steps converges to a local maximum of the p.d.f.. The volume that only includes the set of points that converge to the same local maximum is defined as an attraction basin. Attraction basins can be easily computed by starting the mean-shift iterations for all points in the feature set X. Finally, each attraction basin defines a cluster.

Carreira-Perpiñán [26] showed that when using a Gaussian kernel mean-shift is an Expectation-Maximization (EM) algorithm [15] and when the kernel is non-Gaussian, mean-shift is a generalized EM algorithm. Notice however that the number of clusters is not specified a priori.

5.2 Clustering with Neighborhood Graphs

5.2.1 Relative Neighborhood graphs

Let X be a set of points and let d be a metric. Two points $x_i, x_j \in X$ are said to be relative neighbors if the following condition holds

$$\forall x_k \in X,\ k \neq i, j,\ d(x_i, x_j) \leq \max\left[d(x_i, x_k), d(x_j, x_k)\right] \tag{5.9}$$

The graph formed by adding an edge for each pair of relative neighbors is called the Relative Neighborhood Graph (RNG).

Bandyopadhyay [7] proposed a clustering algorithm based on analyzing the RNG. It is based on the assumption that if x_i and x_j are relative neighbors, and

they belong to distinct clusters, then $d(x_i, x_j) > d(x_i, x_k)$ for all x_k in the same cluster as x_i. Then we search for a suitable threshold δ to separate intra-cluster and inter-cluster edges. The method is detailed in Algorithm 2.

Algorithm 2 Compute the clustering of feature set X using metric d using Bandyopadhyay's algorithm

Require: $X \neq \emptyset$
Ensure: S is a clustered partition of X
compute the RNG $R = (X, E_R)$ using d
$\tilde{E}_R \leftarrow$ sort E_R by non-decreasing edge weight.
$\tilde{E}_R \leftarrow$ eliminate edges in $\tilde{E}_R$ with duplicated weights.
$m \leftarrow \min_{(u,v)\in\tilde{E}_R} d(u,v)$
$M \leftarrow \max_{(u,v)\in\tilde{E}_R} d(u,v)$
if $M < 2m$ **then**
    $S \leftarrow \{X\}$
    return
end if
$t \leftarrow \frac{1}{2}\left(\tilde{E}_R(2) - \tilde{E}_R(1) + \tilde{E}_R(|\tilde{E}_R|) - \tilde{E}_T(|\tilde{E}_R| - 1)\right)$
if $\left(\exists j,\ 1 \le j \le |\tilde{E}_R| - 1\right)\ \tilde{E}_R(j+1) - \tilde{E}_R(j) \ge t \ \wedge\ \tilde{E}_R(j+1) \ge 2m$ **then**
    $\delta \leftarrow \tilde{E}_R(j)$
else
    $S \leftarrow \{X\}$
    return
end if
$E_R \leftarrow E_R - \{(u,v) \mid d(u,v) > \delta\}$
$\mathcal{R} \leftarrow$ connected components of R.
if $|\mathcal{R}| = 1$ or $|\mathcal{R}| > \sqrt{|X|}$ **then**
    $S \leftarrow \{X\}$
    return
else
    $S \leftarrow \emptyset$
    for all $R_i = (X_i, E_{R_i}) \in \mathcal{R}$ **do**
        add the result of recursively clustering X_i to S
    end for
end if

The method works well in many cases, see Figure 5.2. However, the method is filled with hard-to-justify heuristic parameters:. First, the maximum number of clusters is hardcoded to $\sqrt{|X|}$ using a "rule of thumb" in the author's words [7]. The underlying idea is that a cluster must contain in average more than $\sqrt{|X|}$ points. If a cluster contains less points but is tight enough, it should be detected. Second, the longest edge must have twice the length of the shortest edge. Third and more importantly, the threshold is defined using the average of the difference between

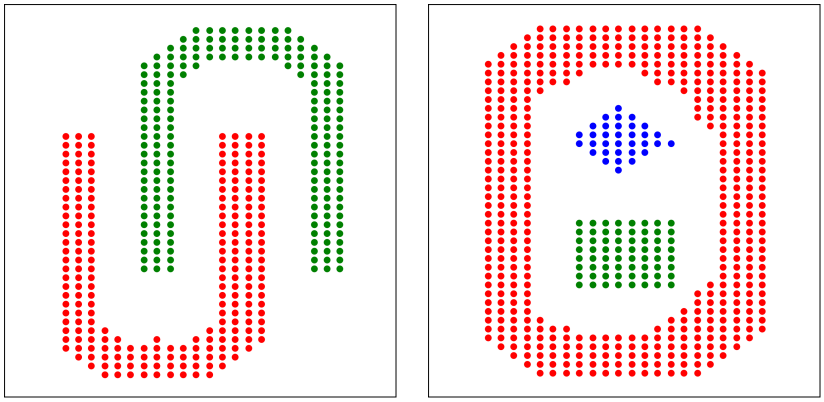


Figure 5.2: Results with Bandyopadhyay's method.

the two shortest edges and of the difference between the two longest edges. For nonuniform clusters, it may not be the best choice. For example, if the point set is a mixture of two Gaussians, such average may not reflect the Gaussians standard deviations.

Bandyopadhyay argues that this process may produce an oversplitting phenomenon [7]. To correct it, they also set a minimum number of points for each cluster.

5.2.2 Using the MST: Zahn's method

Hierarchical clustering methods aim at correctly detecting clusters in point group hierarchies. The classical choice to partition a feature set in hierarchical clustering, is globally thresholding the hierarchy at a fixed level [68].

To our knowledge, Zahn [143] made the first attempt to find a local rule to partition a feature set using the minimal spanning tree (or single-link hierarchy). Locally adaptive thresholds provide more realistic and useful partitions.

Definition 18. *Let $T = (X, E_T)$ be the minimum spanning tree of X. We define the k-neighborhood of $e \in E_T$ as the edges $e' \in E_T$ such that there is a path of length $k-1$ between an endpoint of e and an endpoint of e'.*

Definition 19. *We say $e \in E_T$ is an inconsistent edge if its weight $\omega(e)$ is significantly larger than the average of edge weights in the k-neighborhood of e.*

In this context, the significance can be measured by computing standard deviations or ratios. Zahn suggested to cluster a feature set by eliminating the inconsistent edges in this minimum spanning tree, see Figure 5.3.

This foundational method however remains highly heuristic and the choice for the significance depends on the nature of the clusters, turning clustering into a chicken-and-egg problem.

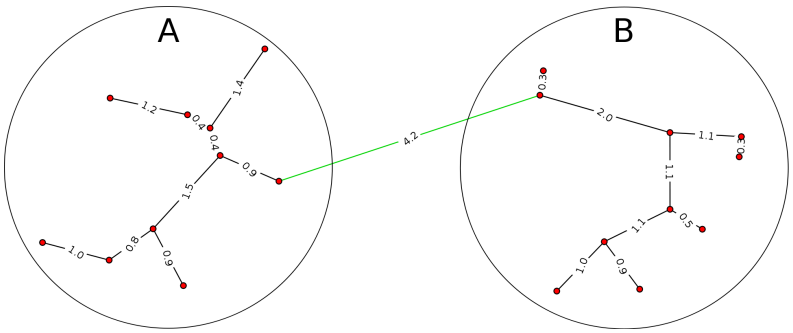


Figure 5.3: Let A∪B be the neighborhood of central edge (in green) whose length is 4.2. The average length is 0.93, which is significantly shorter than 4.2. The central edge is then inconsistent.

5.2.3 Using the MST: Felzenszwalb and Huttenlocher' method

Following the same line of work, Felzenszwalb and Huttenlocher [50] proposed a similar partitioning criterion but thresholds are locally adaptive, thus providing more realistic and useful partitions.

Let X be a feature set and d a suitable metric. Let $T = (X, E_T)$ be the minimum spanning tree of X. We say $C \in X$ is a component if T_C is a (connected) subtree of T with node set C. For a component C we define

$$\max(C) = \max_{\substack{v_i, v_j \in C \\ (v_i, v_j) \in E_T}} d(v_i, v_j) + \tau(C) \tag{5.10}$$

The function τ can take many forms, but can be simply defined as $\tau(C) = s/|C|$, where s acts as a scale parameter [50]. Then, given two components C_1 and C_2 we define

$$\min(C_1, C_2) = \min(\max(C_1), \max(C_2)) \tag{5.11}$$

From a conceptual point of view, the above rule is very similar to Zahn's: the core difference is replacing the average by the maximum and adding a local correction factor to avoid oversplitting. In this way, Felzenszwalb and Huttenlocher' algorithm is non other than Kruskal's [36] with an additional criterion to avoid merging certain connected subcomponents in the MST. That is, instead of constructing a MST, a minimum spanning forest is built. Given two disjoint components C_1 and C_2 they are only merged if

$$\min(C_1, C_2) \geq \min_{\substack{v_i \in C_1 \\ v_j \in C_2}} d(v_i, v_j) \tag{5.12}$$

Each tree in the forest is finally detected as a cluster. The process is described in Algorithm 3.

Algorithm 3 Compute the clustering of feature set X using metric d using Felzenszwalb and Huttenlocher' algorithm.

Require: $X \neq \emptyset$
Ensure: S is a clustered partition of X
1: compute the MST $T = (X, E_T)$ using d
2: $\tilde{E}_T \leftarrow$ sort E_T by non-decreasing edge weight.
3: $S \leftarrow \{\{v_1\}, \ldots, \{v_n\}\}$
4: **while** $\tilde{E}_T \neq \emptyset$ **do**
5: $(v_i, v_j) \leftarrow \text{head}(\tilde{E}_T)$
6: let C_i and C_j be the sets of S containing v_i and v_j respectively.
7: **if** $(C_i \neq C_j) \wedge (d(v_i, v_j) \leq \min(C_i, C_j))$ **then**
8: $S \leftarrow \left(S - C_i - C_j\right) \cup \left\{C_i \cup C_j\right\}$
9: **end if**
10: $\tilde{E}_T \leftarrow \text{tail}(\tilde{E}_T)$
11: **end while**

5.3 Other approaches

Ensemble clustering is a rapidly growing framework [129]. It is based on the concept that different clustering algorithms impose different organizations. For a given problem and in the absence of a priori information about the nature of the data to cluster, any given algorithm may impose an organization that do not correspond to the true natural organization. However, by combining results from different clustering algorithms a more comprehensive solution can be achieved. Vega-Pons and Shulcloper made a thorough review (in spanish) of this subfield [135].

A slightly more constraint clustering problem is the k-way graph partitioning problem which is defined as follows: given a graph $G = (V, E)$ with $|V| = n$, partition V into k subsets, $V_1, V_2, \ldots, V_k$ such that $V i \cap V j = \emptyset$ for $i \neq j$, $|V_i| = n/k$, $\bigcup_i V_i = V$, and the number of edges of E whose incident vertices belong to different subsets is minimized.

To solve this problem, a multilevel approach technique proved successful [64, 73]. The size of the graph is reduced by collapsing vertices and edges, the smaller graph is partitioned, and then uncoarsened to construct a partition for the original graph. In the uncoarsening phase, the coarse partition is refined by interpolation. This kind of mechanism provides not only good results but fast algorithms, since the partitioning phase is performed on a reduced graph.

Karypis and Kumar [73] proposed an algorithm called METIS that produces a graph partitioning with nearly equal-sized partitions. Extending the idea, Dhillon et al. [41] dropped the equal-size requirement and proposed a multilevel approach in combination with spectral graph methods (to be precise, with kernel k-means). Cour et al. [37] also propose a multilevel spectral approach, but instead of interpolating the original coarse partition, all scales interact in parallel to construct a partition of the original uncoarsened graph.

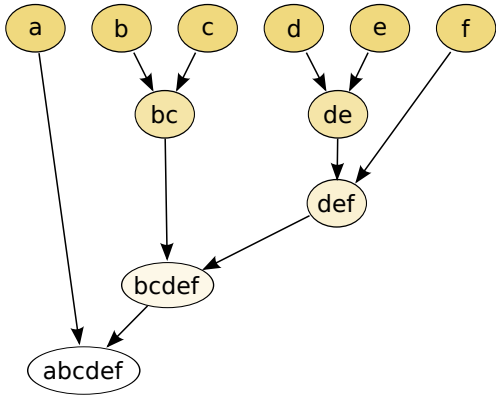


Figure 5.4: Dendrograms are recursive structures. Each node is a singleton (i.e. a leaf) or contains other nodes. Reproduced from http://en.wikipedia.org/wiki/Dendrogram

5.4 Hierarchical clustering

We have seen that partitional methods build a single partition with k clusters. Hierarchical methods seek to construct a hierarchy of clusters. This hierarchies are recursive in nature and are often represented by dendrograms, see Figure 5.4. The recursivity is natural in many domains were a taxonomy is needed.

These clustering methods can be divided in two classes

divisive: a top-down approach is followed in which the data is recursively partitioned [13, 102].

agglomerative: in this bottom-up approach, each point starts as a singleton cluster, and the closest pair of clusters $C_i, C_j \subset X$, in the sense of a chosen cluster dissimilarity measure $\delta(C_i, C_j)$, is iteratively merged [68].

Agglomerative methods are usually computationally simpler and in the following we will focus on them.

Different choices for $\delta(C_i, C_j)$ yield different algorithms

centroid-link: $\delta(C_i, C_j) = \min_{\substack{x_i \in C_i \\ x_j \in C_j}} d(x_i, x_j)$

This algorithm is conceptually close to k-means, since both minimize variance. In this sense, it generates compact isotropic clusters.

complete-link: $\delta(C_i, C_j) = \frac{1}{|C_i|\,|C_j|} \sum_{x_i \in C_i} \sum_{x_j \in C_j} d(x_i, x_j)$

This algorithm also favors compact isotropic clusters, as the merged clusters are those whose farthest ends are closer.

single-link: $\delta(C_i, C_j) = \min_{\substack{x_i \in C_i \\ x_j \in C_j}} d(x_i, x_j)$

Here the nearest-neighbor points determine the nearest subsets. If one lets the procedure evolve up to having a single cluster containing all points, the

result is a minimum spanning tree. We will explore this structure in depth in Chapters 7 and 8.

The hierarchy building process is based on local optimizations and it would be possible to think that partitional algorithms, which optimize a global criterion, obtain "better" results. Results are certainly better if the parameters of the partitional method (the criterion to optimize, the number of clusters or the kernel shape and size for mean-shift) coincide with the natural structure of the data. For example, if one knows that the feature set is a mixture of k isotropic Gaussians, then k-means will perform extremely well. Since we are following a blind approach about the nature of the data, hierarchical structures emerge as appealing creatures.

Since their outputs are nested series of partitions, ranging from $|X| = N$ clusters to one single cluster, one can imagine methods to determine the number of clusters as stopping rules in the merging process. If stopping rules are correctly designed, hierarchical methods would also be able to detect clusters having different densities or different number of points. We have already seen a successful example in Felzenszwalb and Huttenlocher' algorithm, see Section 5.2.3.

Partitions can thus be extracted from the hierarchy. For example, in Figure 5.4 one can extract $\mathcal{P} = \{\{a\},\{b\},\{c\},\{d\},\{e\},\{f\}\}$ but also $\mathcal{P}' = \{\{a\},\{b,c\},\{d,e,f\}\}$. Notice that the number of partitions that can be extracted from a hierarchy is far smaller than the number of possible partitions. Although this can be seen as a disadvantage, this "data-oriented" sampling of partitions makes the validation task computationally feasible.

It is interesting to point out what occurs from the theoretical point of view. Starting from three simple properties:

scale invariance if all distances are expanded or shrunk, we get the same result;

richness if the clustering algorithm is seen as a function from the feature space to a partitions space, the function's image must not be a subset of the partitions space;

consistency when intra-cluster distances are shrunk and inter-cluster distances are expanded, we get the same result;

Kleinberg [77] proved that no clustering scheme satisfying these conditions simultaneously can exist. Interestingly, Carlsson and Mémoli [25] proposed a characterization theorem that can be interpreted as a relaxation of Kleinberg's impossibility result in that by allowing the output of clustering methods to be hierarchical, one obtains existence and uniqueness.

5.5 Validating clusters

We have seen that there exist many clustering formulations which may produce very different results. Thus, the need of evaluating such results arises. The focus is now on determining which results are more accurate (in terms both of better data description and model simplicity). Many formulations assume that the data is indeed clustered, but what happens if it is not the case? Detecting such cases is also a main part of the process of cluster validation.

5.5.1 Validation indices

One may want to compare results from different partitional algorithms, results from the same algorithm with different parameter choices or different partitions extracted from a hierarchy of clusters. The most straightforward notion to perform these comparisons is to establish a similarity measure between partitions. Such measures are often called indices and many of them have been proposed over the years [135].

One of the most popular indices is the Rand index [117]. Let X be a set of n features and let $\mathcal{P}_1$ and $\mathcal{P}_2$ be two partitions of X. Finally,

- let N_{11} be the number of pairs of elements in X that are in the same set in P_1 and in the same set in P_2,

$$N_{11} = \#\big\{(x_i, x_j) \mid x_i, x_j \in X, \\ (\exists P_1 \in \mathcal{P}_1)\ x_i, x_j \in P_1 \wedge (\exists P_2 \in \mathcal{P}_2)\ x_i, x_j \in P_2\big\} \quad (5.13)$$

- let N_{00} be the number of pairs of elements in X that are in different sets in P_1 and in different sets in P_2,

$$N_{00} = \#\big\{(x_i, x_j) \mid x_i, x_j \in X, \\ (\nexists P_1 \in \mathcal{P}_1)\ x_i, x_j \in P_1 \wedge (\nexists P_2 \in \mathcal{P}_2)\ x_i, x_j \in P_2\big\} \quad (5.14)$$

- let N_{10} be the number of pairs of elements in X that are in the same set in P_1 and in different sets in P_2,

$$N_{10} = \#\big\{(x_i, x_j) \mid x_i, x_j \in X, \\ (\exists P_1 \in \mathcal{P}_1)\ x_i, x_j \in P_1 \wedge (\nexists P_2 \in \mathcal{P}_2)\ x_i, x_j \in P_2\big\} \quad (5.15)$$

- let N_{01} be the number of pairs of elements in X that are in different sets in P_1 and in the same set in P_2,

$$N_{01} = \#\big\{(x_i, x_j) \mid x_i, x_j \in X, \\ (\nexists P_1 \in \mathcal{P}_1)\ x_i, x_j \in P_1 \wedge (\exists P_2 \in \mathcal{P}_2)\ x_i, x_j \in P_2\big\} \quad (5.16)$$

The Rand index R is then defined as

$$\mathrm{RI}(\mathcal{P}_1, \mathcal{P}_2) = \frac{N_{00} + N_{11}}{N_{00} + N_{11} + N_{10} + N_{01}} = \frac{N_{00} + N_{11}}{\binom{n}{2}}. \quad (5.17)$$

Notice that $\mathrm{RI}(\mathcal{P}_1, \mathcal{P}_2) \in [0,1]$ and takes the value 1 when the two partitions are identical and 0 when $N_{00} = N_{11} = 0$. In practice, this last scenario is too extreme and has very little probability of occurrence. The index should actually reflect the measure of similarity between two partitions compared to the measure they would have by chance.

Hubert and Arabie proposed a method to correct this issue [66]. Let $\mathcal{P}_1 = \left\{P_1^{(1)}, P_1^{(2)}, \ldots, P_1^{(k)}\right\}$ and $\mathcal{P}_2 = \left\{P_2^{(1)}, P_2^{(2)}, \ldots, P_1^{(m)}\right\}$ Let us denote

$$
\begin{aligned}
n_{ij} &= \#\left\{(x_i, x_j) \mid x_i, x_j \in X,\ x_i, x_j \in P_1^{(i)} \wedge x_i, x_j \in P_2^{(j)}\right\} \\
a_i &= \sum_{j=1}^{m} n_{ij} \\
b_j &= \sum_{i=1}^{k} n_{ij} \\
N &= \sum_{i=1}^{k} \sum_{j=1}^{m} n_{ij}
\end{aligned}
$$

The Adjusted Rand Index ARI takes into account the expected value between two random partitions and is defined as

$$
\mathrm{ARI}(\mathcal{P}_1, \mathcal{P}_2) = \frac{\sum_{ij} \binom{n_{ij}}{2} - \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}\right] / \binom{N}{2}}{\frac{1}{2}\left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}\right] - \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}\right] / \binom{N}{2}}. \tag{5.18}
$$

The Adjusted Rand Index equals 1 when the Rand index equals 1. However the Adjusted Rand Index equals 0 when the index equals the expected value between two random partitions. Vinh et al. [136] further refined this index by using concepts from information theory such as the expected mutual information.

The underlying model of randomness behind this definition is the permutation model, in which partitions (i.e. clusterings) are generated randomly from the space of permutations with fixed number of clusters and points in each cluster. However, the background model is to simplistic: a fixed the number of points in each partition (i.e. cluster) is not realistic in many applications.

5.5.2 The Je(2)/Je(1) stopping rule

A relatively simple validation scheme for hierarchical methods was developed by Duda and Hart [45]. The Je(2)/Je(1) stopping rule is used for determining whether or not a cluster should be split into two subclusters.

For the two cluster solution the total within sum of squared distances about centroids of the two clusters, namely Je(2), is computed. For the single cluster solution, the same quantity, this time about the centroid of the concerned cluster, is computed and is denoted by Je(1). The method considers a null hypothesis where all features come from a normal distribution, whose mean and variance are empirically estimated over the whole dataset. The null hypothesis of one single Gaussian cluster is rejected if the ratio Je(2)/Je(1) is smaller than a critical value, specified by means of a significance level for the hypothesis testing.

This methods suffers from the same issues than the Adjusted Rand Index, that is an oversimplistic background model that may not be adapted to real clustering problems.

5.5.3 Testing randomness in the MST

It has been long observed [143] that the analysis of the Minimum Spanning Tree can reveal useful information about the structure of a feature set, see Section 5.2.2. As in the previously presented validation methods, randomness plays a key role. Indeed,

> Presence of clusters in data can be generally identified by density variations of patterns. Hence, our meaning of lack of structure in the data corresponds to a uniform distribution of data; departure from uniformity indicates the existence of possible clusters. (Jain et al. [69])

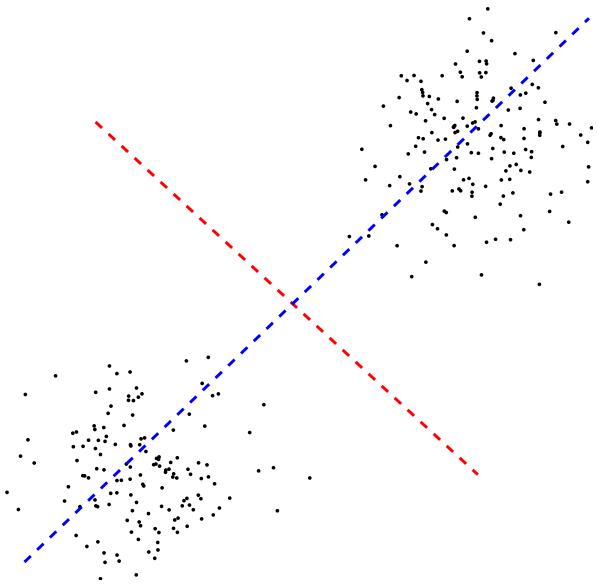
An interesting reflexion is pointed out by Hoffman and Jain [65] about uniformity. To be able to use uniformity as base for a decision rule, the support, or sampling windows, of the data must be known. Of course, this is rarely the case. Hoffman and Jain supply this uncertainty by using the convex hull of the observed feature set [65]. We will discuss this in Chapter 8.

Let us suppose we have two samples of size n and m, respectively, from distributions F_x and F_y both defined on $\mathbb{R}^H$. We define the null hypothesis $\mathcal{H}_0$ as $F_x = F_y$ and the alternative hypothesis $\mathcal{H}_1$ as $F_x \neq F_y$. A weighted graph of interpoint distances is created. Then we compute the MST of this graph and we remove the edges whose endpoints lie on different samples. The test statistic is defined as the number of resulting subtrees. $\mathcal{H}_0$ is rejected for a small number of subtrees. If both samples correspond to two separated and compact clusters there will be only two subtrees, as a single MST edge would have been removed.

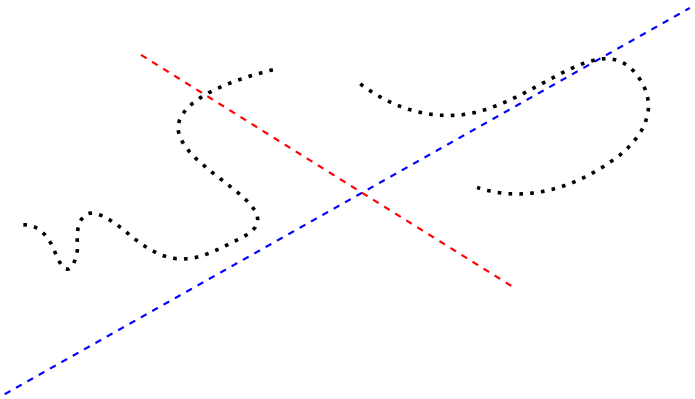
As stated, this is a two-sample test. Actually, for given data set we do not have two samples but only one. Choosing the samples is crucial, as a wrong choice can yield incorrect results. An example is given in Figure 5.5a, where two different sample choices are depicted.

Much later, Jain [69] addressed the problem of generating two suitable samples. Unfortunately, the method relies on heuristic assumptions. First, the sum of the MST edge's lengths is normalized to one. Then we discard edges whose normalized length exceeds the value 6. Then both samples are determined by analyzing the average edge length of the resulting subtrees. Subtrees with small average length form the first sample and subtrees with large average length define the second sample. There are two hard thresholds in the above process which ought to be tuned to build both samples correctly. To summarize, the test will only produce a correct result if both samples are extremely well chosen.

A second issue lies in the test itself. Counting the number of subtrees is a suitable statistic for relatively compact and isotropic clusters. However, for elongated clusters, it may produce incorrect results. In Figure 5.5b the removal of any MST edge will create only two subtrees; hence, as the number of subtrees does not char-



(a) Only one MST edge will cross the red frontier, generating two subtrees, while many edges will cross the blue frontier, generating many subtrees.



(b) Removing any MST edge creates two subtrees.

Figure 5.5: Two different ways of splitting point sets, depicted by the red and green frontiers.

acterize the clusters, the test will yield non-significant results.

Notice that this test is not a clustering algorithm. The test is able to separate background from non-background samples, but does not say anything about the structure of the clustered sample: a second algorithm has be used to actually cluster it.

As final point, it is important to mention that there is no analytical expression for the distribution of MST edges under the uniform distribution (or any parametric distribution, for the case) [65]. This distribution has to be empirically estimated by sampling from the background distribution.

Summary

In this chapter we studied and analyzed different approaches to clustering. We reviewed some main trends in partitional methods and presented the alternative hierarchical approach. We also discussed the problem of cluster validation. In the following we will devote ourselves to presenting new clustering validation methods, based on the a contrario framework.

CHAPTER

6 Clustering using graph-based density estimation

Abstract

In this chapter we present a new clustering method which is parameterless, independent from the original data dimensionality and from the shape of the clusters. It only takes into account inter-point distances and it has no random steps. The method performs an a contrario validation of clusters in a hierarchical structure by using graph-based nonparametric density estimation.

6.1 Introduction

Variations of the minimal spanning tree or limited neighborhood set approaches have been extensively explored [7, 50] (see Chapter 5, Section 5.2). The criteria in most works are based on local properties of the graph. Since perceptual grouping implies an assessment of local properties versus global properties, exclusively local methods must be discarded or patched. For example, Felzenszwalb's method [50] makes use of the minimal spanning tree but is forced to add an ad hoc global criterion to correct local observations.

As early as in 1971 Zahn established in a pioneering work [143] the conceptual grounds on which this work is based. He faced the problem of finding perceptual clusters according to the "proximity" gestalt principle [138]. He proposed three key arguments:

1. **Only inter-point distances matter**. The characteristics of the metric space should not be used. In other terms, we must look for solutions that do not rely on any assumptions about the chosen metric. It imposes graphs as the only suitable underlying structure for clustering.
2. **No random steps**. Results must remain equal for all runs of the detection process. In particular, random initializations are forbidden.
3. **Independence from the exploration strategy**. The order in which points are analyzed must not affect the outcome of the algorithm.

The ultimate goal of our work is to propose a clustering method which follows Zahn's principles. In recent years, a theory for the quantitative analysis of gestalts, called Computational Gestalt [39], was developed and since then many refinements followed. By studying Zahn's principles in the light of the Computational Gestalt theory, we introduce a perceptually driven clustering method. It only takes into account inter-point distances and it has no random steps. As a consequence, it is independent from the original data dimensionality and from the shape of the clusters.

This work is structured as follows. In Section 6.2 we present a new parameterless clustering method. In Section 6.3 we show results and then provide final remarks in Section 6.4.

6.2 A Contrario Clustering

In this section we present a method that overcomes the issues presented in the previous section. It is not dependent of random initializations and clusters are detected without manually specifying its number. The presented method is general, in the sense that it can be applied to any feature set X equipped with a suitable distance d.

We start by presenting a method in which ours is inspired. Then we introduce theoretical definitions in Section 6.2.1. The automatic choice of detection thresholds is discussed in Section 6.2.2. Sections 6.2.3 and 6.2.4 address the simplification and revision, respectively, of the obtained clusters.

In the scope of the Computational Gestalt programme [40], Cao et al. proposed a cluster detection method [21]. The main idea is to measure the statistical significance of a set of points as being a cluster. Let us recall its basic definition.

For $k \leq N \in \mathbb{N}$ and $p \in [0,1]$, let us denote by

$$\mathcal{B}(N,k;p) \stackrel{\text{def}}{=} \sum_{j \geq k}^{N} \binom{N}{j} p^j (1-p)^{N-j} \tag{6.1}$$

the tail of the binomial law. See Desolneux et al. for a thorough study of the binomial tail and its use in the detection of geometric structures [40].

Let $\mathcal{R}$ be a set of H-dimensional hyper-rectangles parallel to the coordinates axes and centered at the origin.

Definition 20. *Let $C \subset X$ be a subset of k points out of the N data points. We call number of false alarms (NFA) of C,*

$$\text{NFA}(C) \stackrel{\text{def}}{=} |\mathcal{R}| \cdot N \cdot \min_{\substack{x_i \in C \\ R \in \mathcal{R} \\ C \subset R + x_i}} \mathcal{B}(N-1, k-1, \pi_i) \tag{6.2}$$

where $R + x_i$ is the rectangle R translated to x_i and $\pi_i = \Pr(x \in x_i + R)$ is its probability. We say that C is an ε-meaningful group if $\text{NFA}(C) < \varepsilon$.

The term $|\mathcal{R}| \cdot N$ is the number of tests and the remaining part of Equation 6.2 represents a Probability of False Alarms (PFA). $\mathcal{B}(N-1, k-1, \pi_i)$ corresponds to the probability that at least $k-1$ out of the $N-1$ remaining points fall into $x_i + R$. The detection algorithm consists in exploring the group of points given by a dendrogram, identifying ε-meaningful groups as clusters and then performing an additional pruning, based on the inclusion properties of the dendrogram. A similar technique will be described in Section 6.2.3.

Notice that the above detection rule does not conform to Zahn's first argument. Indeed, inter-points distances are not directly taken into account and hyper-rectangles are used instead.

Another drawback of this approach is that a set $\mathcal{R}$ of hyper-rectangles parallel to the axes must be properly chosen. This choice is application specific since $\mathcal{R}$ is intrinsically related to cluster size/scale. For example, an exponential choice for the discretization of the rectangle space is made by Cao et al. [21] introducing a bias for small rectangles (since they are more densely sampled).

Each cluster must be fitted by an axis-aligned hyper-rectangle $R \in \mathcal{R}$, meaning that clusters with arbitrary shapes are not detected. Even hyper-rectangular but diagonal clusters may be missed or oversegmented.

The probability law π_i for each hyper-rectangle $R \in \mathcal{R}$, assuming no specific structure in the data, must be known a priori or estimated. The complexity of computation of the probability of an hyper-rectangle then depends on the dimension H and suffers from the "curse of dimensionality".

6.2.1 Graph-based A Contrario Clustering

We now propose a new method to find clusters in graphs that is independent from their shape and from the dimension H. We first build a weighted undirected graph $G = (V, E)$ where $v_i \in V$ is identified with a feature $x_i \in X$ in a metric space (X, d) and the weighting function ω is defined in terms of the corresponding distance function

$$\omega((v_i, v_j)) = d(x_i, x_j). \tag{6.3}$$

A subgraph G' of G is a connected graph $G' = (V', E')$ in which $V' \subseteq V$ and $E' \subseteq V' \times V'$.

Definition 21. *We define the* non-compactness *of a subgraph G' of a graph G as*

$$c(G') \stackrel{\text{def}}{=} \frac{\Omega(E')}{\Omega(E)} \quad \textit{where} \quad \Omega(E) = \sum_{e \in E} \omega(e). \tag{6.4}$$

We say G' is p-compact if $c(G') = p$.

Non-compactness is an estimation of the local vertex density, which plays the role of the relative volume π_i of the hyper-rectangles in Definition 20. It can be interpreted in the spirit of non-parametric density estimation. Parzen methods [57]

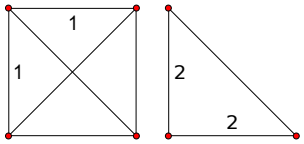


Figure 6.1: Two different subgraphs $G'_1 = (V'_1, E'_1)$ and $G'_2 = (V'_2, E'_2)$ such that $\Omega(E'_1) = \Omega(E'_2) = 4 + 2\sqrt{2}$. Suppose they are respectively embedded in two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ such that $\Omega(E_1) = \Omega(E_2)$, G'_1 would have the same non-compactness as G'_2 while having more nodes.

locally estimate density at a given point by computing the distances to its neighbors

$$\mathrm{p}(x) \approx \frac{1}{Nh} \sum_{i=1}^{N} Q\left(\frac{||x - x_i||}{h}\right), \tag{6.5}$$

where Q is a smoothing kernel. Choosing a kernel Q that evaluates to one in G' and to zero elsewhere, and picking $x \in X$, yields

$$\mathrm{p}(x) \approx \frac{1}{Nh^2} \sum_{i=1}^{K} \omega((v, v_i)), \tag{6.6}$$

where $v \in V$ is the vertex associated to feature x. Finally

$$\Pr(v \in V') \approx \frac{1}{Nh^2} \sum_{j=1}^{K} \sum_{i=1}^{K} \omega((v_j, v_i)). \tag{6.7}$$

By normalizing $\Pr(v \in V')$, $c(G')$ is obtained.

In our approach, non-compactness plays an important role in cluster detection. Informally, a cluster is a subgraph with two properties:

- its vertices are sufficiently near each other with respect to the remaining vertices, i.e. small non-compactness, and
- the number of its vertices is sufficiently large.

A detection scheme must propose a balance between the density of the cluster and its size. A small set must be very dense to be perceptually noticeable while larger sets are more clearly perceived even if they are less dense. A set of points will be more striking, and more compact, as it gets farther away from the rest of the points.

The non-compactness of a subgraph models how tight its vertices are but is not sufficient to characterize clusters. This can be seen in Figure 6.1 where $G'_1 = (V'_1, E'_1)$ and $G'_2 = (V'_2, E'_2)$ are two different subgraphs such that $\Omega(E'_1) = \Omega(E'_2)$. Suppose they are respectively embedded in two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ such that $\Omega(E_1) = \Omega(E_2)$, then G'_1 would have the same non-compactness as G'_2 while having more nodes.

Suppose we wanted to statistically model all possible instances of clustered data. First, we would need a model for each possible cluster shape (e.g. Gaussian).

Each individual model would have its own parameter set. Next, since we seek for a general model, it must support data distributed in multiple clusters with different shapes. In other terms, we need to integrate the individual models into a mixture. For example, k-means is aimed to detect mixtures of k Gaussian distributions.

These different mixtures can be seen as a parametric family $\mathcal{F}$ of distributions. This family is parametric on the number of clusters and on the parameters of the distribution modeling each cluster. Such a family has colossal cardinality and parameter set. Even if we restrict ourselves to a mixture of k Gaussian clusters for H dimensional data, the problem is hard enough. Since there are $H(H+1)/2+H$ parameters for each Gaussian (determined by an $H \times H$ covariance matrix and $H \times 1$ mean vector), the number of parameters is $k(H(H+1)/2+H)$ for each k. A family of such mixtures, parametric on k, is already untestable in practice.

Assume we were to model the problem described above as a classical hypothesis test:
$\mathcal{H}_1$*: the observed features have a particular distribution* $F \in \mathcal{F}$*, i.e. the data is clustered.*
$\mathcal{H}_0$*: the observed features are distributed more randomly, e.g. uniformly.*

Modeling $\mathcal{H}_1$ means to model $\mathcal{F}$. Modeling $\mathcal{F}$, even if it was feasible, would require an a priori characterization of what a cluster is. Due to the reasons that have just been exposed, we are not interested in defining such a model. We prefer instead to follow the classical claims of the Computational Gestalt School [40]; our detection algorithm will be driven by the Helmholtz principle: no perception in white noise. We will only concentrate on modeling $\mathcal{H}_0$ and consider that low probabilities of occurrence under $\mathcal{H}_0$ are indeed causal instead of casual.

Testing randomness to detect clusters is not a new concept. To cite a few, Hoffman, Jain et al. [65, 69] perform such kind of tests. The works by Cao et al. [21] and by Desolneux et al. [40] also follow this line of research.

We are interested in a general clustering method but, in accordance to our model, in applications where the cluster shapes and the number of clusters are known a priori, the full hypothesis test could be performed, using similar but more simple a contrario techniques.

Given a subgraph G' and its non-compactness, we consider its number of vertices k as a realization of a random variable K. We can then model the probability that a p-compact subgraph G' has at least k vertices due to a realization of randomness. We call this probability $\Pr(G' \mid \mathcal{H}_0)$.

Definition 22. *Let G' be a p-compact subgraph of G, we define the probability of false alarms (PFA) of G' as*

$$\mathrm{PFA}(G') \stackrel{\mathrm{def}}{=} \Pr(G' \mid \mathcal{H}_0) = \mathcal{B}(|V|, |V'|; p) \tag{6.8}$$

where $|\cdot|$ denotes the cardinality of a set.

The probability of false alarms quantifies the unlikeliness of occurrence of a p-compact subgraph G' of G with at least $|V'|$ nodes among $|V|$ under $\mathcal{H}_0$. In

other terms the unlikeliness of occurrence of a configuration of at least $|V'|$ features among $|V|$ with density p.

6.2.2 Learning detection thresholds

To detect unlikely dense subgraphs, a threshold is necessary on the PFA. In the classical *a contrario* framework, a new quantity is introduced: the Number of False Alarms (NFA), i.e. the product of the PFA by the number of tests (see Definition 20). The NFA has a more intuitive meaning than the PFA, since it is an upper bound on the expectation of the number of false detections [40]. The threshold is then easily set on the NFA.

To use the NFA one has to be able to compute (or at least analytically estimate) the number of tests being performed. In our setting, this is not possible, since it is equivalent to computing the number of subgraphs for a graph, which is an astronomical quantity (e.g. approximately 10^{300} for $N = 1000$).

An alternative solution, proposed by Burrus [19], consists in estimating the threshold directly on the PFA by Monte Carlo simulations, following the actual search heuristics instead of trying to bound the total number of tests in a full search. Furthermore, this solution allows to estimate not only a global threshold but partial thresholds, by splitting the detected subgraphs into different categories, each with a dedicated threshold. In this work we follow this approach.

Let $\mathcal{G}$ be the set of all subgraphs of a graph G and let $\mathcal{J}_K : \mathcal{G} \to \{1, 2, \ldots, K\}$ be a hash function used to divide $\mathcal{G}$ into K categories. We define the exploration strategy $\mathcal{S} \subseteq \mathcal{G}$ as a set of subgraphs to be analyzed during detection and learning. In Section 6.2.3 we analyze exploration strategies in depth. As an example we define the basic universal strategy $\mathcal{S}_U = \mathcal{G}$.

We already stated that our detection algorithm is ruled by the principle of no perception in white noise. It is therefore clear that subgraphs in $\mathcal{S}$ with a PFA that is likely to occur in white noise have to be discarded. A direct approach would be to generate several random graphs, compute the PFA of their subgraphs and select a threshold such that all of them are discarded (up to a certain confidence level).

Given a hash function $\mathcal{J}_K$ and an exploration strategy $\mathcal{S}$, Algorithm 4 performs exactly that procedure to compute a set of thresholds $\delta(\varepsilon) = \{\delta_k(\varepsilon)\}_{k=1\ldots K}$. Thresholds are first initialized. For each category k, we run Q simulations. For each simulation $q = \{1 \ldots Q\}$, the number d_q of subgraphs G' such that $\mathcal{J}_K(G') = k$ and $\mathrm{PFA}(G') < \delta_k(\varepsilon)$ is counted. Then the empirical mean m_k and deviation s_k of $d = \{d_q\}_{q=1\ldots Q}$ are computed.

An upper bound u of the expectation μ_k of m_k is computed by performing a Student's t-test. We are looking to approximate ε/K by u and $\delta_k(\varepsilon)$ is adjusted accordingly. We refer to Burrus [19] for further details.

Definition 23. *We say that a subgraph G' is an ε-meaningful cluster if*

$$\mathrm{PFA}(G') < \delta_{\mathcal{J}_K(G')}(\varepsilon) \tag{6.9}$$

Algorithm 4 Compute $\delta(\varepsilon)$ for N vertices using exploration strategy $\mathcal{S}$ by Q Monte Carlo simulations.

for all $k \in \{1 \dots K\}$ **do**
  initialize $\delta_k(\varepsilon)$
  repeat
    for all $q \in \{1 \dots Q\}$ **do**
      build a graph G_q with N vertices at random from the distribution of $\mathcal{H}_0$.
      apply $\mathcal{S}$ to G_q.
      $d_q \leftarrow \#\{G' \in \mathcal{S},\ \mathrm{PFA}(G') < \delta_k(\varepsilon) \ \wedge\ \mathcal{J}_K(G') = k\}$
    end for
    Compute the empirical mean m_k and deviation s_k of $d = \{d_q\}_{q=1\dots Q}$
    Compute a confidence interval on the expectation μ_k of d using the property $\Pr(\mu_k \leq m_k) = \mathrm{F}_{Q-1}\left(\frac{m_k-\mu_k}{s_k}\sqrt{Q-1}\right)$ where $\mathrm{F}_n(x)$ is the distribution function of a Student law with n degrees of freedom.
    For a chosen confidence level, if the estimated upper bound of μ_k is greater than ε/K, increase $\delta_k(\varepsilon)$ otherwise decrease $\delta_k(\varepsilon)$.
  until convergence of $\delta_k(\varepsilon)$
end for

for a properly computed set of thresholds $\delta(\varepsilon)$.

We define the number of false alarms (NFA) of G' *as*

$$\mathrm{NFA}(G') \stackrel{\mathrm{def}}{=} \frac{\varepsilon}{\delta_{\mathcal{J}_K(G')}(\varepsilon)}\mathrm{PFA}(G') \tag{6.10}$$

Note that subgraphs consisting of a single node must certainly not be detected. From Definition 22 they cannot be detected, i.e. $\mathcal{B}(|V|, 1; 0) = 1$. As we look for rare events, subgraphs with probability of occurrence in noise equal to 1 are never detected.

Lemma 4. *The expected number of* ε*-meaningful clusters in a random graph* G *is lower than* ε.

Proof. By construction of $\delta_k(\varepsilon)$, the number of meaningful subgraphs G' in a random graph G is lower than ε/K. By linearity of expectation, if there are less than ε/K errors in average for each category, then there are globally less than ε errors in average. □

Figure 6.2 depicts the profile of the learned set of thresholds $\delta(\varepsilon)$ for the exploration strategy explained in Section 6.2.3 and different graph sizes. There are N vertices, $K = 10$ categories and the hash function for a graph $G' = (V'E')$ is simply $\mathcal{J}_K(G') = \lfloor(|V'|+1)\cdot K/N\rfloor$. Medium-small and compact subgraphs are more frequent than large and compact subgraphs causing that thresholds for the first

categories are more restrictive than thresholds for the last ones. Note that the evolution of the set of thresholds $\delta(\varepsilon)$ with size is smooth, allowing the computation of intermediate sets of thresholds by interpolation.

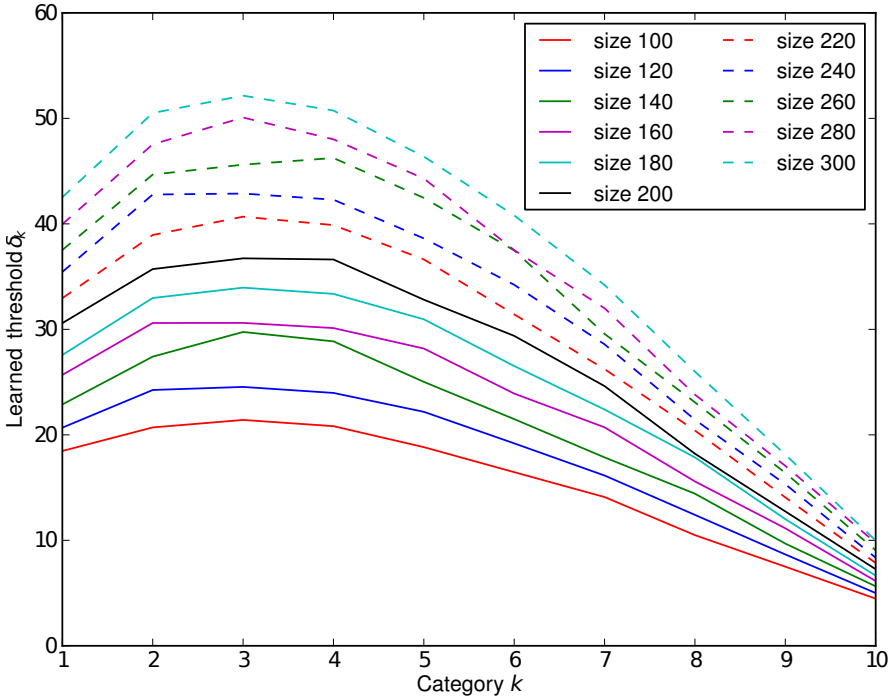


Figure 6.2: Learned set of thresholds $\delta(\varepsilon)$ for $\varepsilon = 1$ for sizes ranging from $N = 100$ to $N = 300$ (in negative logarithmic scale). Thresholds evolve smoothly with size, consequently they can be safely interpolated for intermediates sizes. Larger subgraphs are rarer and thresholds are therefore less restrictive than for smaller subgraphs.

The role of the hash function is to correct a bias that might be introduced by the exploration strategy. If the sizes of the clusters one seeks to detect are well enough sampled by the exploration strategy, choosing $K = 1$ should suffice.

6.2.3 Eliminating redundancy

While each meaningful cluster is relevant by itself, the whole set of meaningful clusters exhibits, in general, high redundancy [21]. Indeed, a very meaningful cluster G_1 usually remains meaningful when it is slightly enlarged or shrunk into a graph G_2. If, e.g. $G_2 \subset G_1$, this question is easily answered by comparing NFA(G_1) and NFA(G_2), see Definition 23. The NFA is used instead of the PFA to allow comparisons that span different categories. The group with the smallest NFA must of course be preferred. The above criterion is implemented by the following pruning rule

for all ε-meaningful clusters G_1, G_2 **do**
  if $G_2 \subset G_1 \ \vee \ G_1 \subset G_2$ **then**
    eliminate $\arg\max(\mathrm{NFA}(G_1), \mathrm{NFA}(G_2))$
  end if
end for

that indeed produces the desired result but is computationally intractable. Moreover, for a given graph, exploring the whole set of its subgraphs to compute each PFA is already intractable. An exploration algorithm is therefore needed. Hierarchical clustering methods are well suited for this task.

Definition 24. *A hierarchy $\mathcal{T}$ of a graph $G = (V, E)$ is a set such that $\forall\ T \in \mathcal{T}$*

- $\exists\ v \in V,\ T = \{v\}$ *or*
- $\exists\ T_1, T_2 \in \mathcal{T},\ T = T_1 \cup T_2$.

Depending on the direction they build the hierarchy, these clustering methods can be agglomerative (bottom-up) or divisive (top-down). The former are usually computationally simpler.

Any hierarchical algorithm [67] can be used. In this work we focus on the agglomerative algorithm called minimal spanning tree. Zahn's work [143] concentrates on stating the good properties of minimal spanning trees to detect perceptual clusters.

The construction of the minimal spanning tree starts by considering each single point as a cluster and iteratively merges the pair of clusters containing the closest nearest-neighbor points. It can be found using Kruskal's method (Figure 6.3).

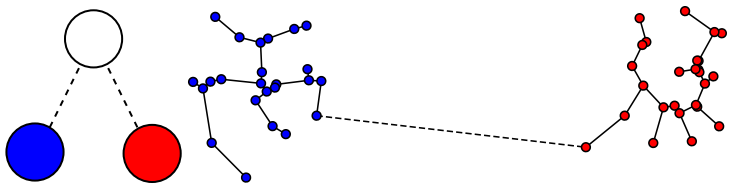


Figure 6.3: Part of a minimal spanning tree. The blue node set and the red node set are linked by the dashed edge, creating a new node in the minimal spanning tree.

We will restrict ourselves to explore the node sets contained in a hierarchy and to compute PFA on the subgraphs induced by them. Finally we apply the aforementioned pruning rule profiting from the inclusion properties of the tree structure.

Definition 25 (Exploration strategy)**.** *Given a graph $G = (V, E)$, we denote by $G_C = (C, E')$ the subgraph such that*

$$\forall\ (c_a, c_b) \in E,\ c_a \in C\ \wedge\ c_b \in C \Rightarrow (c_a, c_b) \in E', \tag{6.11}$$

and we say that G_C is induced by C. For a given hierarchy $\mathcal{T}$ we define the exploration strategy as $\mathcal{S}_\mathcal{T} = \{G_T\}_{T \in \mathcal{T}}$.

Definition 26 (Maximal ε-meaningful cluster)**.** *For a learned set of thresholds $\delta(\varepsilon)$, we say that $G_T \in \mathcal{S}_\mathcal{T}$ is a maximal ε-meaningful cluster if and only if*

1. *G_T is an ε-meaningful cluster, see Definition 23,*
2. *all meaningful descendants are less meaningful,*
   $\forall\ T' \in \mathcal{T},\ T' \subset T,\ \mathrm{NFA}(G_{T'}) > \mathrm{NFA}(G_T)$,
3. *all meaningful ancestors are less meaningful,*
   $\forall\ T' \in \mathcal{T},\ T \subset T',\ \mathrm{NFA}(G_{T'}) \geq \mathrm{NFA}(G_T)$.

The proposed clustering algorithm simply consists on detecting maximal ε-meaningful clusters. Definition 26 is closely related to the maximality rule by Cao et al. [21] but is simpler and it might be regarded as an implementation of the exclusion principle [40]. In our experiments, we found no need to include a measure of meaningfulness for a pair of subgraphs.

According to Definition 25 the subgraph G_C is the largest subgraph in G not containing more vertices than C. Why not use instead the partial trees provided by the hierarchy as in Figure 6.3? Because in the father (represented in white) inter-cluster edges (in dashed line) are under-represented with respect to intra-cluster edges (in solid lines).

To explain this situation, let $G = (V, E)$ be a hypothetical base graph. Let A be the blue node set and B the red node set respectively and let them induce subgraphs $G_A = (A, E_A)$ and $G_B = (B, E_B)$. Let us denote the father of both G_A and G_B by $G_{A\cup B} = (A \cup B, E_A \cup E_B \cup E_{AB})$ where $E_{AB} = \{(v_A, v_B) \in E,\ v_A \in A \wedge v_B \in B\}$. Then

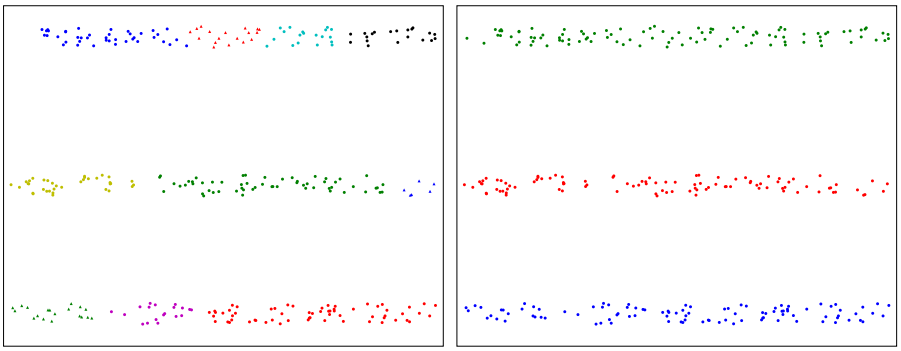
$$c(G_{A\cup B}) = \frac{\Omega(E_A) + \Omega(E_B) + \Omega(E_{AB})}{\Omega(E)} \tag{6.12}$$

Let us compare $\mathrm{PFA}(G_{A\cup B})$ with $\mathrm{PFA}(G_A)$. The non-compactness $c(G_{A\cup B})$ grows by $\frac{\Omega(E_B)+\Omega(E_{AB})}{\Omega(E)}$ with respect to $c(G_A)$. Since B is tight, $\Omega(E_B)$ is small and the growth is mainly determined by $\Omega(E_{AB})$. Meanwhile, $E_A \cup E_B \cup E_{AB}$ grows by $|E_B| + |E_{AB}|$ with respect to E_A. If $|E_{AB}|$ is small (in our case 1), the growth in size is mainly determined by $|E_B|$. The same reasoning can be applied to $\mathrm{PFA}(G_{A\cup B})$ and $\mathrm{PFA}(G_B)$. In summary, $\Omega(E_{AB})$ has to be very large to compensate for the relatively large growth in size. If it is not the case, the father will be more meaningful than its two children. Only very separated clusters will be detected separately. To correct this situation, inter-cluster edges have to be better sampled. A reasonable choice is using the subgraphs induced by A and B.

Comaniciu and Meer [35] state that "arbitrarily structured feature spaces can be analyzed only by nonparametric methods since these methods do not have embedded assumptions". They classify nonparametric clustering methods into two classes: *hierarchical clustering* and *density estimation*. Regarding the first class, they regard the detection of clusters in a hierarchy as being a non-trivial task. The proposed approach, maximal ε-meaningful clusters, merges these two main trends. It detects clusters in a hierarchy by using density estimation. The hierarchy provides candidate groups in a natural manner thus, from one side, allowing a reduced effort in the density estimation step and, from the other side, providing the cardinality of such groups as important information, see Definition 23.

6.2.4 Revising elongated clusters

Non-elongated clusters are preferred by our detection algorithm. Actual elongated clusters are separated in several non-elongated maximal meaningful clusters as in Figure 6.4a. Detecting non-elongated clusters is equivalent to detecting clusters with the same intrinsic dimension as the embedding. To correct this issue, a revision is needed.



(a) (b)

Figure 6.4: Maximal meaningful clusters (a) before and (b) after revising elongated clusters. In (a) 10 clusters are found while in (b) only 3 remain.

For a given graph $G = (V, E)$, let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two subgraphs of G. We define

$$\mathrm{L}(G_1, G_2) \stackrel{\text{def}}{=} \min\left(\max_{e\in E_1}\omega(e),\ \max_{e\in E_2}\omega(e)\right). \tag{6.13}$$

This equation is very similar to the one proposed by Felzenszwalb [50] to detect clusters in a hierarchical structure except there is no extra scale parameter.

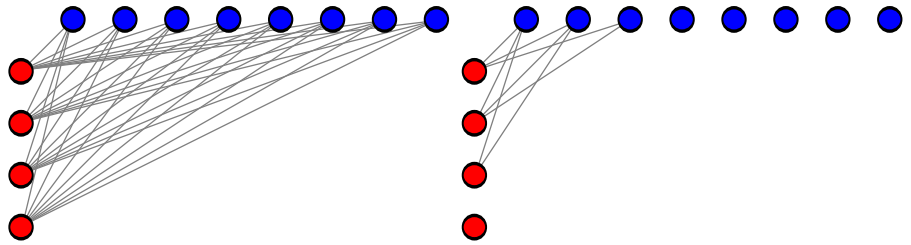
Let $G_T = (V_T, E_T)$ be a maximal ε-meaningful cluster, $G_F = (V_F, E_F)$ its father and $G_S = (V_S, E_S)$ its sibling in $\mathcal{T}$. Let us define

$$G_{T\cup S} \stackrel{\text{def}}{=} (V_F,\ E_T \cup E_S \cup E_{TS}) \tag{6.14}$$

where $E_{TS} = \{e \in E_F,\ \omega(e) \leq \mathrm{L}(G_T, G_S)\}$. Long edges connecting the extremes of the G_F are eliminated from $G_{T\cup S}$ by using a local connection between G_T and G_S. The effect of this local connection rule is shown in Figure 6.5.

If $\mathrm{PFA}(G_{T\cup S}) < \mathrm{PFA}(G_T)$, G_F is replaced in $\mathcal{S}_\mathcal{T}$ by $G_{T\cup S}$ and maximality in $\mathcal{S}_\mathcal{T}$ is recomputed. This procedure is repeated until convergence. Note that convergence is guaranteed since changes are propagated up in the tree, stopping at the root in the worst case. This heuristic is able to correct for oversplitting of elongated clusters present in the embedding, as seen in Figure 6.4b.

Algorithm 5 summarizes the complete proposed detection approach.



(a) (b)

Figure 6.5: Effect of locally connecting clusters. The subgraphs G_T (in blue) and G_S (in red) are siblings in $\mathcal{T}$. For clarity, only inter-cluster edges are depicted. (a) Their father G_F in $\mathcal{T}$. (b) The locally connected graph $G_{T\cup S}$.

Algorithm 5 For a point set X and an appropriate kernel distance d compute the set $\mathcal{M}$ of maximal ε-meaningful clusters.

build G_o from X using d
compute the hierarchy $\mathcal{T}$ from G_o
$\mathcal{M} = \emptyset$
for all $G_T \in \mathcal{S}_\mathcal{T}$ **do**
  if G_T is maximal ε-meaningful **then**
    add G_T to $\mathcal{M}$
  end if
end for
repeat
  choose $G_T \in \mathcal{M}$
  find its sibling G_S, its father G_F in $\mathcal{S}_\mathcal{T}$ and compute $G_{T\cup S}$
  if PFA($G_{T\cup S}$) < PFA(G_T) **then**
    replace G_T (and possibly G_S) by $G_{T\cup S}$ in $\mathcal{M}$
    replace G_F by $G_{T\cup S}$ in $\mathcal{S}_\mathcal{T}$
  end if
until no more replacements are performed

6.3 Experimental results

Although the method may be applied to any metric space, we will use spectral methods as they produce tighter clusters which are more suitable for the presented clustering approach, see Chapter 5, Section 5.1.1. We will then cluster embedding $\overline{A}_M$, where its rows are the input features $X = \{x_i\}_{i=1\ldots N}$, identifying x_i with the i-th row of $\overline{A}_M$ and by defining d as the usual Euclidean distance in $\mathbb{R}^M$. Notice that this distance may represent a more complex semidistance in the original feature space $\mathbb{R}^H$. Summarizing, we take the following steps:

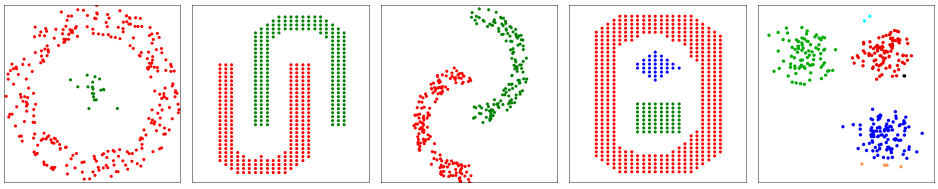
1. build G_o from X using d;
2. compute the embedding $\overline{A}_M$ from G_o;

3. compute maximal ε-meaningful clusters (Algorithm 5) using the rows of $\overline{A}_M$ as the feature set

As every spectral clustering technique, there are two main values to be tuned: the scale of the kernel distance and parameter M (see Equation 5.5 and Algorithm 5). The former was fixed by manually choosing the scale that yields the best results with k-means. Anyhow, once the scale fixed, all compared methods analyze the transformed feature space (i.e. the embedding) an should provide results independently of that choice. The parameter M can be interpreted as an estimation of the number of groups [110]. For example, when using k-means, M is usually set equal to k. In the case of meaningful clusters, the value of M was determined empirically and can be seen as an overestimation of the largest possible number of groups.

The proposed approach is successful at finding perceptually clear 2D clusters even when clusters have arbitrary shapes (Figure 6.6). By only fixing the parameters needed to compute the embedding, i.e. its dimension M and the standard deviation of the Gaussian kernel used for the distance, in all cases the clustering is successful. No extra parameter is needed as fixing $\varepsilon = 1$ is sufficient for stable detections.

Figure 6.6e presents an interesting case since some maximal clusters are not meaningful (represented in cyan, black and orange). The scene is composed of a mixture of three Gaussians. Peripheral points, i.e. located in low density areas, are harder to merge. Note that spectral methods are unable to deal with highly intertwined clusters as features are mapped to a single manifold in the Euclidean embedding.

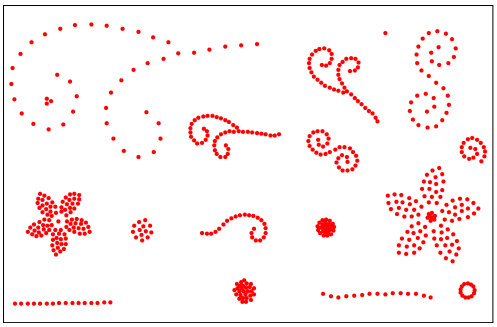


(a) (b) (c) (d) (e)

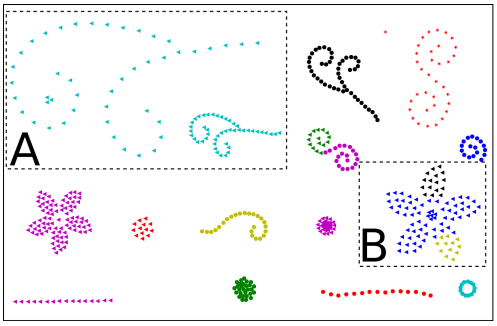
Figure 6.6: 2D points clustering examples. In (a), (b), (c) and (d) all groups are meaningful. In (e) maximal clusters are shown: only red, green and blue groups are meaningful while cyan, black and orange are not and are finally discarded by our algorithm.

Figure 6.7 depicts a comparison between results using k-means and our algorithm. Both start from the same embedding. We consider that there are 15 clusters in Figure 6.7a. For k-means the correct number of clusters was set. The combination of groups with very high density and groups with low density causes the random initialization in k-means to fail, see Figure 6.7b. It creates under-split clusters, e.g. rectangle A, and over-split clusters, e.g. rectangle B. Maximal

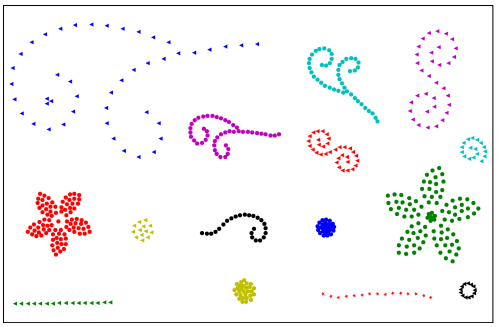
ε-meaningful clusters perform correctly with no parameter tuning (i.e. the number of clusters is automatically found by the algorithm), see Figure 6.7c.



(a)



(b)



(c)

Figure 6.7: (a) Original scene. Starting from the same embedding, result with (b) k-means, where we manually set the correct number of clusters, and (c) maximal meaningful clusters. k-means incorrectly merges some clusters (zone A) and incorrectly splits others (zone B).

The next experiment aims at comparing our results with Mean Shift, see Chapter 5, Section 5.1.2. [1].

[1] http://www.mathworks.com/matlabcentral/fileexchange/10161-mean-shift-clustering

Figure 6.8 presents an experiment were Mean Shift is used to cluster the dataset in Figure 6.7a. Different density estimations were performed, by varying the kernel size. Clearly, results are suboptimal. The main disadvantage we see in the density estimation step is that a global kernel size must be chosen. Such a strategy is unable to cope with clusters of different densities and spatial sizes. Choosing a small kernel causes to correctly detect dense clusters at the price of oversplitting less denser ones. On the contrary, a large kernel corrects the oversplitting of less denser clusters but introduces undersplitting for the denser ones. Our method also uses non-parametric density estimation, but the scale is not fixed in advance. As shown by the Parzen windows interpretation of the non-compactness, the "kernels" we use are determined by the candidate sets given by the hierarchy and thus multiscale density estimation if performed.

In Figure 6.9 we show segmentation results on synthetic images. A precision should be made regarding this experiment as well as all segmentation experiments in this paper: our goal here is not to present a new segmentation method, but just to illustrate the performance of proposed clustering technique by means of segmentation examples. For this reason, we simply consider that the vectors to be clustered are the set of single color image pixels or 3×3 color image patches, depending on the experiment. Notice that there is no term imposing image spatial connectivity of clusters.

Observing Figure 6.9, if the random initialization procedure picks an appropriate seed, Normalized Cuts with k-means may perform reasonably well when setting the correct k. Still, sometimes the resulting clusters can be degraded by noise, as observed on the top figure in the second column. When k is not well chosen ($k = 3$ in our example) results are poor, as on the third column. Results on the fourth column show that our method is successful without any further parameter tuning.

Figure 6.10 presents more image segmentation results. Results on the second column are among the best we could obtain with k-means, by carefully choosing k by hand (we set k to 3, 6, 6, 6 and 11 respectively). In general, results are correct. In all cases except for the one on the first row, the number of clusters k had to be overestimated with respect to the number of visually perceived regions. In some cases, some regions are overconnected (see the blue region on the first row and the red region on second row) a fact that could not be corrected by slightly increasing k.

There have been previous attempts to automatically detect the number of clusters in Normalized Cuts (or spectral clustering), e.g. by Zelnik-Manor and Perona [144]. Their work has similar goals but starts from different requirements: they indeed make use of the characteristics of the eigenspace. Moreover, since their method is based on selecting the minimum of an alignment cost function, it is not able to detect non-clustered data as such. Our method is specifically designed for this task.

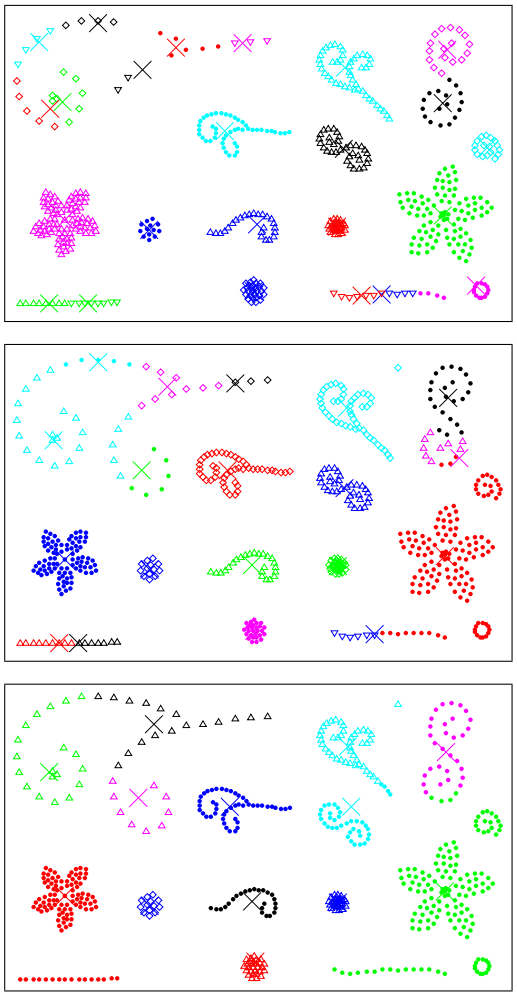


Figure 6.8: Results with Mean Shift for the point set from Figure 6.7a. The local maxima of the estimated density are signaled by x marks. Even when varying the kernel size, results are clearly suboptimal.

Results on the third column were obtained with Zelnik-Manor and Perona's method [144][2]. In this case the features are individual color pixels, since we did not find better results by using patches. In one case, on the fifth row, the method oversegments the image, while in the others the image is undersegmented. In these cases, boundaries between regions seem somewhat away from perceived regions.

The proposed algorithm, whose results are depicted on the fourth column, performs well in all cases, being able to correctly separate perceptually evident clusters. Contrarily to the other two methods, small clusters are not arbitrarily

[2] code available at http://webee.technion.ac.il/~lihi/Demos/SelfTuningClustering.html

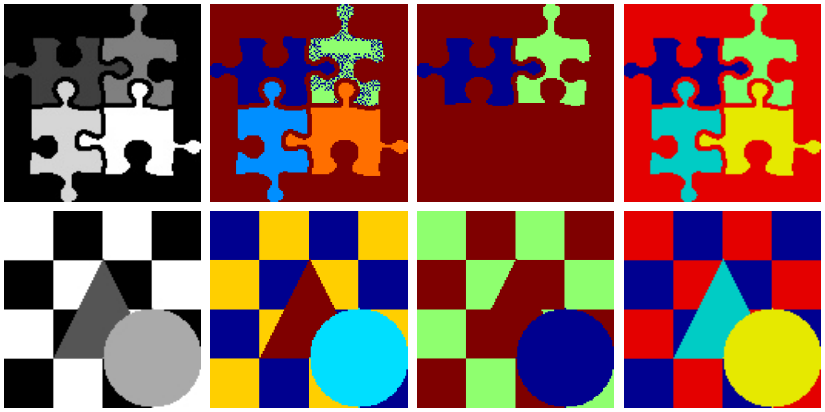


Figure 6.9: Comparison of image segmentations. On the first column,original images with 5 and 4 regions respectively. On the second column, segmentations obtained with k-means by correctly setting k to 5 and 4 respectively. On the third column, segmentations obtained with k-means by setting $k = 3$. On the fourth column, segmentations obtained with our method.

merged to the closest larger cluster but remain undetected. In simpler terms, some patches are detected as not belonging to any cluster: they are considered as noise. In accordance to this claim and since we are clustering patches (without any kind of rotation invariance) the ones that lie on the boundaries between objects are not classified. This is a desirable feature since boundary patches are of different nature from non-boundary patches.

Figure 6.11 presents more segmentation results on images from the COIL-100 database [109]. The same remarks from the previous examples hold. In general, our method correctly finds the clusters and outperforms Zelnik-Manor and Perona's method, although in some cases relatively big areas remain detected as unclustered patches.

The Berkeley Segmentation Dataset [88] is often used to perform segmentation experiments. Nowadays, to our knowledge, an exhaustive review of the clustering methods reported in the literature shows that there exists no clustering approach that is able to correctly and automatically segment all images in such a varied and complex dataset. Hence, we selected a subset of this database that we consider that should be easier to segment. For the sake of completeness, experiments on this subset are also included.

For the final set of experiments we compare our method with the algorithm by Cour et al. [37]. They use the Normalized Cut framework, but using a multiscale decomposition[3]. The final embedding for clustering is constructed by using the information on the different scales and applying inter-scale constraints to ensure overall consistency. The final clustering step is performed by using the discretiza-

[3] http://www.seas.upenn.edu/~timothee/software/ncut_multiscale/ncut_multiscale.html

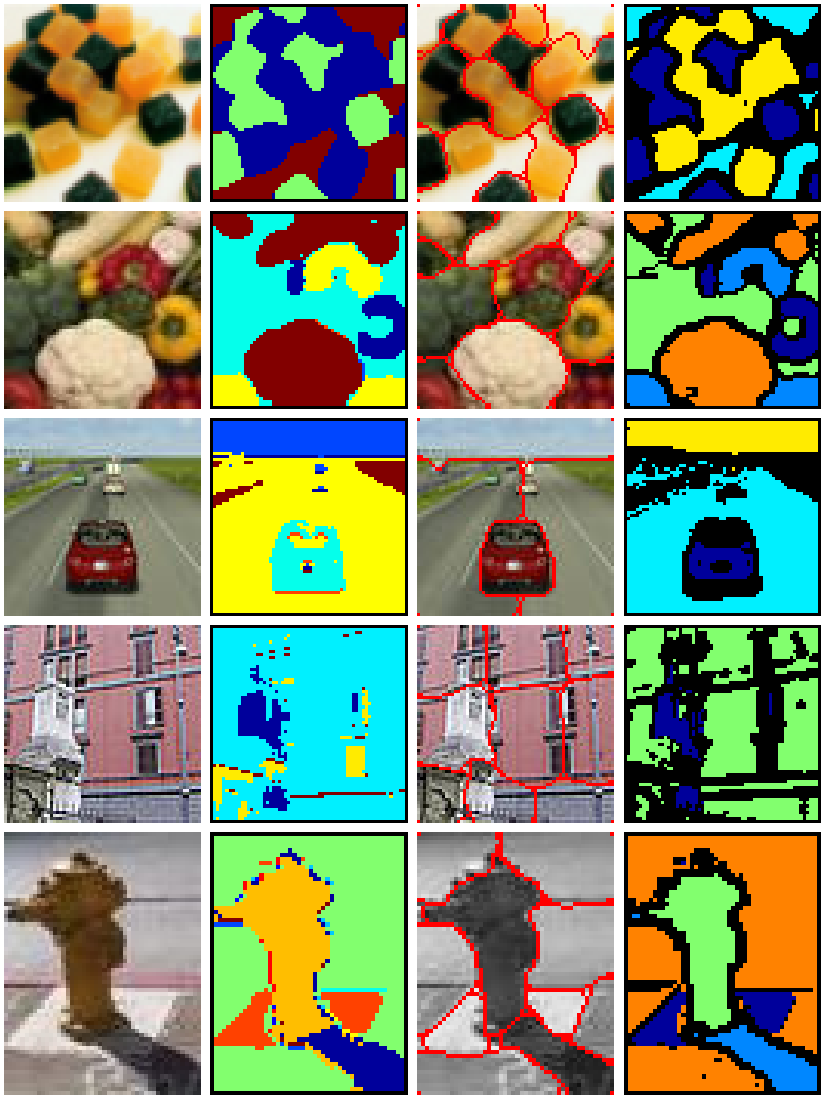


Figure 6.10: Comparison of image segmentations from 3×3 color patches. On the first column, original images. On the second column, results with k-means by tuning k by hand. On the third column, results with Zelnik-Manor and Perona's method. On the fourth column, maximal meaningful clusters: in all examples, non-detected areas are depicted in black.

tion algorithm by Yu and Shi [142] (see Chapter 5, Section 5.1.1).

In Figures 6.12, 6.13 and 6.14 we use Cour's algorithm to construct the embeddings and then compare the clustering results obtained with different methods:

YS Yu and Shi's algorithm [142].

ZMP Zelnik-Manor and Perona's algorithm (restricted to the final assignment of points to clusters) [142].

MMC+R maximal meaningful clusters, revising elongated clusters.
MMC–R maximal meaningful clusters, without revising elongated clusters (i.e. by omitting lines from 9 to 16 on Algorithm 5).

It is important to note that we do not propose a specifically designed method to solve the final assignment problem in Normalized Cuts, but a general clustering algorithm. In this work, we use this algorithm to cluster sets of point within the Normalized Cuts framework.

In Figures 6.12 and 6.13, the spectral embedding is constructed with $M = 3$. This is based on two reasons. First, choosing three regions seems to be a reasonable choice in both experiments. Second, visual inspection of point clouds is easier (otherwise, dimensionality reduction techniques should be applied and results would actually depend also on the performance of these techniques).

By looking directly at the embeddings in Figures 6.12 and 6.13, it is straightforward to see that the clusters detected by YS and by ZMP differ from the results that one should have expected. The proposed method yields detections which seem to be more adequate to the point clouds structure. In Figure 6.12 we perceive that the segmentation obtained with the elongated clusters revision step (Section 6.2.4) is globally better than the one which omits this step. The opposite situation occurs in Figure 6.13, where disabling the revision allows to detect the balcony. A side effect is that the sky is split in three regions, which roughly correspond to different brightness that results from the *degradé* of the sky.

In Figure 6.14, the embeddings are constructed with $M = 10$. In some cases, YS and ZMP perform better while in others the proposed method produces more satisfactory results. All methods oversplit or undersplit clusters in different cases. In general, we think there is no clear winner for these relatively complex scenes. However, both in ZMP and in the proposed approach, contrarily to YS, the number of clusters is not chosen in advance. Moreover, in contrast to ZMP, our method is general in the sense that it was not specifically designed for Normalized Cuts and can be used in other scenarios.

6.4 Final Remarks

The proposed method satisfies Zahn's requirements for a perceptual clustering technique. On the one hand, the algorithm does not involve any random choice since it is completely deterministic. On the other hand, once distances have been computed, the method is independent from the dimension of the points (in this case, the dimension of the embedding). Its running time is not affected by an increase in dimensionality: it does not suffer from the "curse of dimensionality". In addition using a more complicated, time consuming distance function is transparent to our method.

The number of clusters is automatically determined, eliminating a classical parameter that is usually hard to choose. It is replaced by ε which has a more intuitive meaning: it controls the average number of false detections. Tuning its

value is not necessary since setting $\varepsilon = 1$ is sufficient in practice.

Detection thresholds are easily computed from ε by performing Monte Carlo simulations. These thresholds are well adapted to accept/reject non-clustered data. Experimental results support this claim. Indeed, our method correctly finds the number of clusters and the detected clusters are perceptually significant. Moreover, detection is highly stable since clusters have NFAs well below the estimated thresholds.

Results show that the clustering technique is shape independent. Although our base algorithm has a bias towards non-elongated clusters, a simple heuristic rule is able to correct that situation and handle correct results for a wide range of shapes.

Finally, the exploration rule in Section 6.2.3 allows for a reasonable computational complexity of $\mathrm{O}\left(N^2 \cdot \log N\right)$, detailed in 6.A. When N is large, although the total complexity is a low-degree polynom, handling a fully connected graph is costful, no matter how simple the computed operations are. The implementation in its current state can not handle graphs with more than twenty thousands nodes. The computation time of maximal meaningful clusters for a graph of such size takes between one and three minutes.

6.A Temporal complexity

Kruskal's algorithm for computing the minimal spanning tree has a complexity of $\mathrm{O}\left(|E| \cdot \log |E|\right)$, as the edge set E in G must be sorted. There are faster algorithms such as Prim's but optimal computation of the minimum spanning tree is not the goal of this work. Kruskal's algorithm is sped-up by using a union-find algorithm on a disjoint-sets data structure [130]. After sorting edges, union-find allows to build the minimal spanning tree $\mathcal{T}$ in quasi-linear time. More precisely, its worst-case complexity is $\mathrm{O}\left(|E| \cdot \alpha(|E|)\right)$ where α is the extremely slow-growing inverse Ackermann function. In practice $\alpha(M) < 4$.

Computing the set of edges of each node in $\mathcal{T}$ can be done in $\mathrm{O}\left(|E| \cdot \log |E|\right)$. The computation of the binomial tail is done by using the incomplete beta function which is constant in time [115].

Since there are $|V|$ nodes in G, $|\mathcal{T}| = 2|V| - 1$. Computing PFA requires therefore $2|V| - 1$ computations. All nodes in $\mathcal{T}$ are examined during the maximality check, which also amounts to $2|V| - 1$ computations. Maximal meaningful clusters algorithm itself is therefore linear in the number of nodes, i.e. $\mathrm{O}\left(|V|\right)$.

As G is fully connected, $|V| \leq |E| = \frac{|V| \cdot (|V|-1)}{2}$ and since $|V| = |X| = N$, $\mathrm{O}\left(|E| \cdot \log |E|\right) = \mathrm{O}\left(N^2 \cdot \log N\right)$.

Figure 6.11: Comparison of image segmentations from 3×3 color patches. On the first column, images from the COIL-100 database. On the second column, results with k-means by tuning k by hand. On the third column, results with Zelnik-Manor and Perona's method. On the fourth column, maximal meaningful clusters: in all examples, non-detected areas are depicted in black.

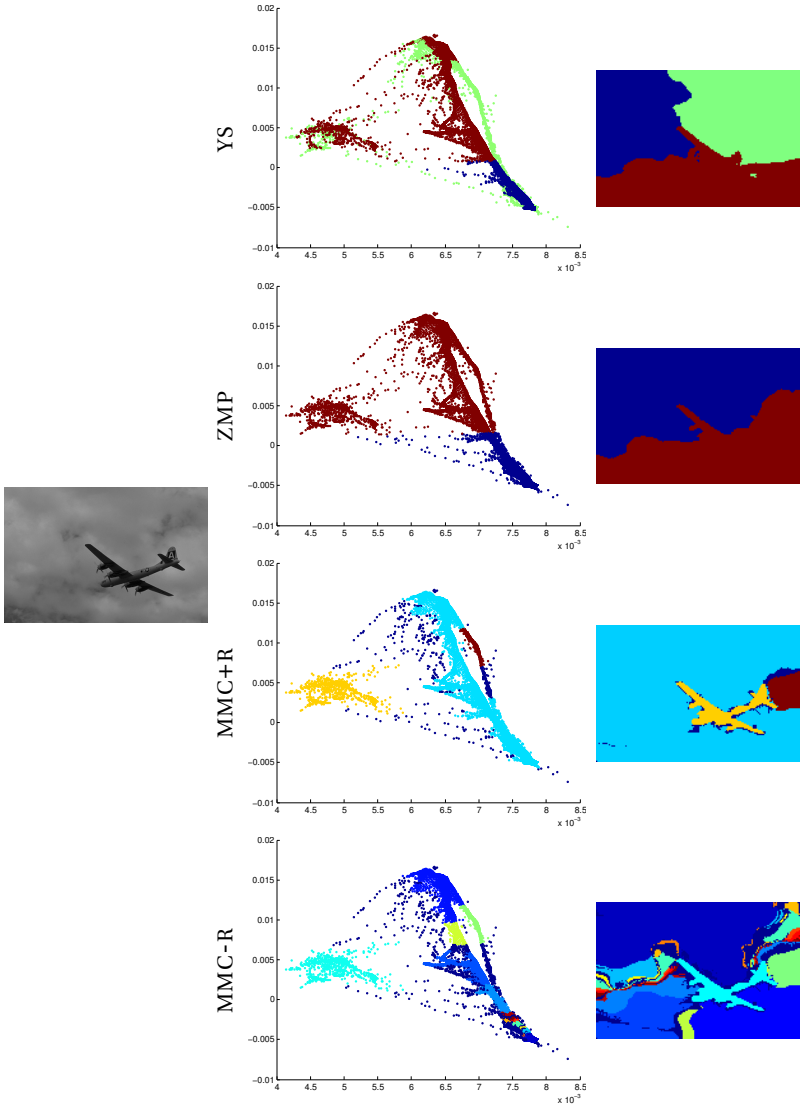


Figure 6.12: In the original image, we perceive two or three main regions: the plane and one or two regions on the textured sky. On the center column, the point cloud on the left, which corresponds to the airplane, is clearly separated from the rest. Neither YS nor ZMP detect it as an individual cluster. The proposed method is able to detect it as a separate cluster.

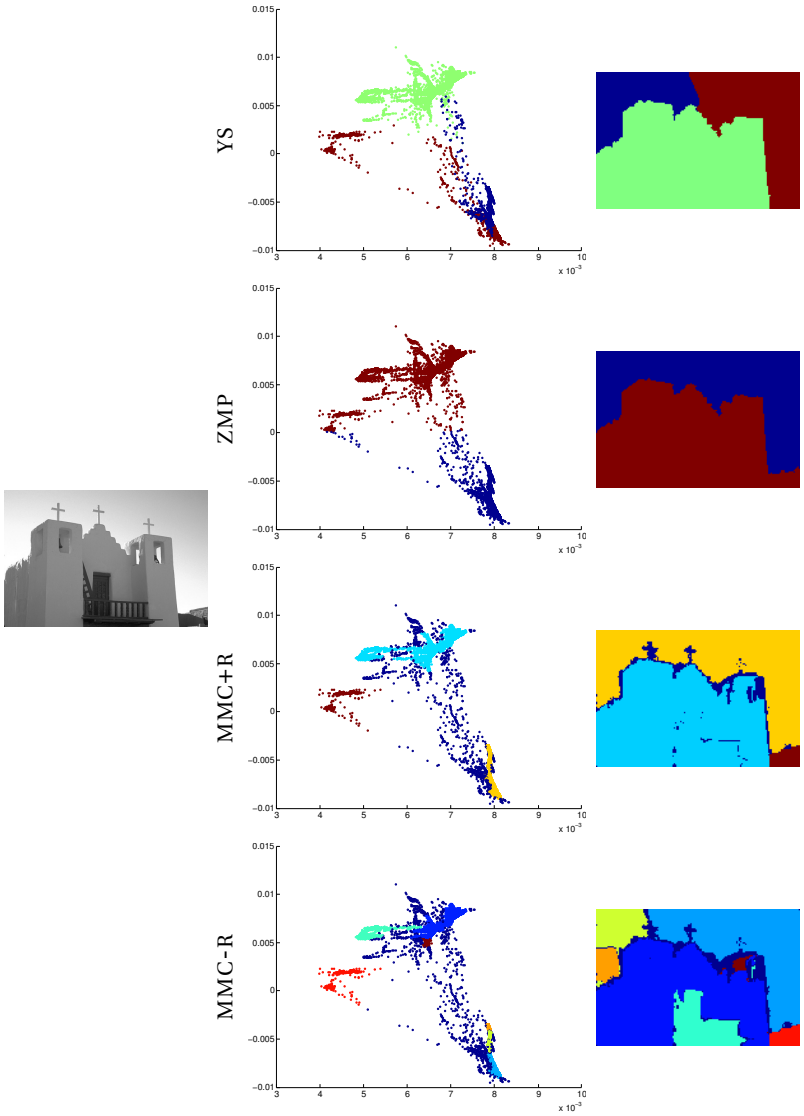


Figure 6.13: Four main regions are perceived in the original image: sky, church, balcony and bottom right dark area. On the center column, the point cloud on the left, which corresponds to bottom right are on the original image, is clearly separated from the rest. Neither YS nor ZMP detect it as an individual cluster. The proposed method is able to detect it as a separate cluster. In MMC–R, the balcony stands out as a separate region, and the sky is split in three regions (due to the *degradé* of the sky).

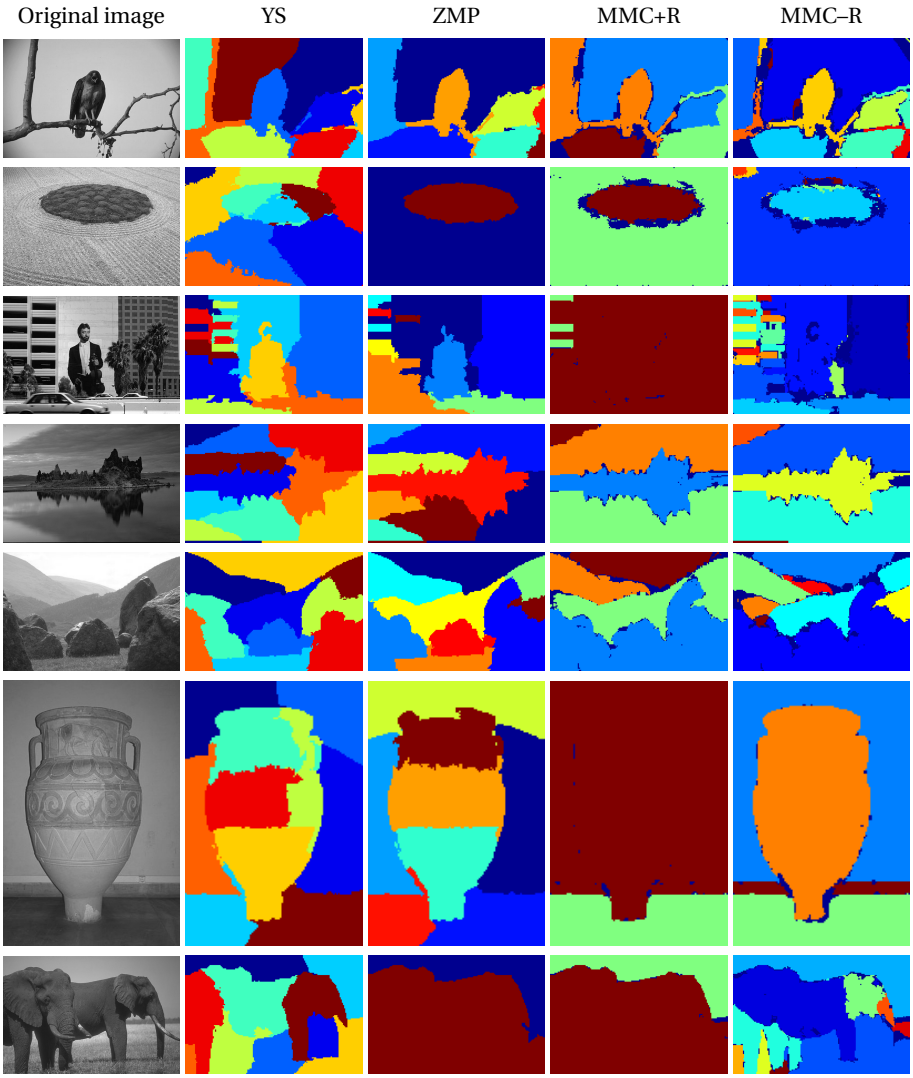


Figure 6.14: All compared methods produce better results for some images and worst ones for others. None of them clearly outperforms the others: depending on the image, clusters are over or undersplit.

CHAPTER

7 Efficient Minimum Spanning Tree

Abstract

Computing the minimum spanning tree (MST) is a common task in the pattern recognition and the computer vision fields. However, little work has been done on efficient general methods for solving the problem on large datasets where graphs are complete and edge weights are given implicitly by a distance between vertex attributes. In this chapter we propose a generic algorithm that extends the classical Boruvka's algorithm by using nearest neighbors search structures to reduce significantly time and memory performances. The algorithm can also compute in a straightforward way approximate MSTs thus further improving speed. Experiments show that the proposed method outperforms classical algorithms on large low-dimensional datasets by several orders of magnitude. Finally, to illustrate the usefulness of the proposed algorithm, we focus on a classical computer vision problem: image segmentation. We extend a local MST-based clustering algorithm by Felzenszwalb and Huttenlocher [50], thus permitting a global scene analysis.

7.1 Introduction

The computation of the minimum spanning tree (MST) is a classical problem in computer science. For an undirected weighted graph, it can be simply stated as finding a tree that covers all vertices, called a spanning tree, with minimum total edge cost. It is taught in every course of algorithms and data structure as an example where greedy strategies are successful and it is regarded as one of the first historical foundations of operations research.

The history of the MST problem up to 1985 was reviewed by Graham and Hell [60]. Maybe the two most widely known algorithms to compute the MST are Prim's and Kruskal's [36]. There is a third classical algorithm by Boruvka [60] that mysteriously remained shadowed by the other two. This fact is emphasized by the fact that Boruvka's algorithm is also known as Sollin's algorithm, despite the fact that Sollin re-discovered it independently years later.

Algorithm	Model	Time
Prim [36]	pointer-based	$O(m \log n)$
Kruskal [36]	pointer-based	$O(m \log n)$
Boruvka [60]	pointer-based	$O(m \log n)$
Karger [72]	random-access	$O(m)$
Chazelle [32]	pointer-based	$O(m\alpha(m,n))$

Table 7.1: Major MST algorithms, their computational model and their time complexity (n and m are the number of nodes and the number of edges, respectively). Karger's and Chazelle's are amortized complexities. All algorithms have a spatial complexity of $O(m)$ at least.

Under a restricted random-access computational model, Karger etal [72] published a randomized algorithm that runs on expected linear time. Up to date, Chazelle's [32] is the fastest pointer-based algorithm to compute the MST. Table 7.1 summarizes the complexity of classical and state-of-the-art algorithms. Pettie and Ramachandran [114] proposed an optimal theoretical algorithm which runs in time $O(\mathcal{T}^*(m,n))$ where n (respectively m) is the number of vertices (respectively edges) of the graph and $\mathcal{T}^*$ is the minimum number of edge-weight comparisons needed to determine the solution.

The MST algorithm is particularly interesting for many data analysis tasks in computer vision and pattern recognition. A clear example is clustering, where the classical single-link hierarchical algorithm [68] can be proved to be equivalent to computing the MST. In a seminal work, Zahn [143] studied the benefits of using the MST for clustering, which were lately checked in psychophysical experiments by Dry et al. [44]. More recently, the MST received much attention maybe due to the growth in the size of clustering datasets (e.g. [27, 50]). The approximate MST (AMST), suboptimal but faster, also received attention for the same reasons Lai et al. [79].

We now slightly change the definition of the problem to a form more suitable for feature sets analysis (e.g. clustering).

Definition 27. *Given a set M and a function $d : M \times M \to \mathbb{R}$ such that, $\forall x, y \in M$,*

- $d(x,y) \geq 0$ *(non-negativity),*
- $d(x,y) = 0 \Leftrightarrow x = y$ *(identity of indiscernibles),*
- $d(x,y) = d(y,x)$ *(symmetry) and*
- $d(x,z) \leq d(x,y) + d(y,z)$ *(triangle inequality).*

Then d and the pair (M,d) are said to be a metric on M and a metric space, respectively.

Definition 28. *Given a metric space (M,d) and feature set $X \subseteq M$, the MST of X is defined as the MST of the weighted undirected graph $G = (V,E)$ where each $v_i \in V$ is identified with a feature $x_i \in X$, $E = V \times V$ and the graph's weighting function*

$\omega : E \to \mathbb{R}$ *is defined as*

$$\omega((v_i, v_j)) = d(x_i, x_j). \tag{7.1}$$

In other terms, the problem remains the same but the graph is now complete by definition. In such a context, and as the feature set gets larger, all the previous algorithms are worthless (as explained later).

The problem is classically addressed by using metric spaces with exploitable specific characteristics, i.e. the Euclidean MST is contained in the Delaunay triangulation of X [46]. Recent work has aimed at building an AMST [79] through a clever use of space-filling curves. The fractal nature of such curves imposes the use of a scale parameter, which is not easy to set automatically.

Bently and Friedman [14] and Murtagh [104] addressed the problem of using nearest neighbors search structures to compute the MST. The approach proved successful; moreover, using such structures allows in addition to compute the AMST in a natural and straightforward way. Both works are outdated and a revision in the light of novel nearest neighbors techniques and increasing computational power is much needed. More recently, Leibe et al. [81] also use nearest neighbors techniques for hierarchical clustering using the average-link criterion. Although they improved the method's performance, their algorithm is not suitable for extremely large datasets.

In this chapter we address the MST problem without computing all distances in E. We build on Boruvka's approach, summarized in Algorithm 6, by an appropriate use of nearest neighbors search techniques.

Algorithm 6 Compute the minimum spanning tree of X using metric d

Require: $X \neq \emptyset$
Ensure: $T = (X, E_T)$ is the minimum spanning tree of X using metric d

$E_T \leftarrow \emptyset$
while $|E_T| < |X| - 1$ **do**
  $E' \leftarrow \emptyset$
  for each connected component C of T **do**
    $(u_m, v_m) \leftarrow \underset{u \in C,\ v \notin C}{\arg\min}\, d(u, v)$
    $\delta_m \leftarrow d(u_m, v_m)$
    $E' \leftarrow E' \cup \{(u_m, v_m, \delta_m)\}$
  end for
  while $E' \neq \emptyset$ **do**
    $(u_m, v_m, \delta_m) \leftarrow \underset{(u,v,\delta) \in E'}{\arg\min}\, d$
    $E' \leftarrow E' \setminus \{(u_m, v_m, \delta_m)\}$
    if $E_T \cup \{(u_m, v_m, d_m)\}$ does not contain cycles **then**
      $E_T \leftarrow E_T \cup \{(u_m, v_m, d_m)\}$
    end if
  end while
end while

7.1.1 MST Complexity

Let us define $n = |X|$. The tasks to be performed are:

- creation of n singletons in $O(n)$ time,
- $n-1$ unions of sets and
- n^2 operations to find whether two nodes are in the same subtree or not.

The disjoint-sets data structure [130] gives extremely efficient operations on disjoint sets. The amortized complexity then is $O\left(n^2\ \alpha(n^2, n)\right)$ per iteration, where alpha is the extremely slow growing inverse Ackermann function, i.e. in practice $\alpha(n^2, n) \leq 4$.

The number of connected components will be reduced by at least a factor of 2 in each iteration. At most $\log n$ iterations must be performed. Therefore, the complexity is $O\left(n^2\ \alpha(n^2, n)\ \log n\right)$ which can be approximated without fear by $O\left(n^2\ \log n\right)$.

The previous paragraphs contain an implicit fact: all $n(n-1)/2$ distance computations must be performed giving birth to a double-sided problem:

in space : storing all $n(n-1)/2$ results for $n \geq 10^5$ is prohibitive,

in time : even if results are not stored, for $n \geq 10^5$ the overall running-time is also prohibitive.

Keep in mind that, in modern pattern recognition applications, feature sets of 10^5 points or more are becoming common [56].

The rest of the paper is structured as follows. In Section 7.2 we propose a general approach to compute the MST using nearest neighbors search structures. Section 7.3 deals with search structures and in particular with a slight modification needed to compute the MST. Section 7.4 shows empirical results of the proposed approach on a synthetic dataset and Section 7.5 shows results on real image segmentation examples. Finally, some final remarks and future work are presented in Section 7.6.

7.2 A Nearest Neighbors Approach

Let us assume that we have a function $\mathrm{NN}_d(A, b)$ that returns the nearest neighbor $a \in A$ of b using metric d. We will discuss such functions and their implementation in Section 7.3.

The term $\underset{u \in C,\ v \notin C}{\arg\min}\, d(u, v)$ in line 5 of Algorithm 6 can be straightforwardly expressed in terms of finding the nearest neighbor in the set $V \setminus C$:

$$u_m = \underset{u \in C}{\arg\min}\, d(u, \mathrm{NN}_d(V \setminus C, u)), \tag{7.2}$$

$$v_m = \mathrm{NN}_d(V \setminus C, u_m). \tag{7.3}$$

Let us define a constraint function $\rho : X \to \{0, 1\}$. We propose to modify the function $\mathrm{NN}_d(A, b)$ by adding an additional constraint ρ on the returned element.

We denote it by $\mathrm{NN}_{d,\rho}(A,b)$. In summary, it returns the nearest neighbor $a \in A$ of b using metric d such that $\rho(a) = 1$. By setting

$$\rho(v) = (v \notin C) \tag{7.4}$$

we have

$$\mathrm{NN}_d(V \setminus C, u) = \mathrm{NN}_{d,\rho}(V, u). \tag{7.5}$$

This kind of problem is sometimes referred to as Foreign Nearest Neighbors in the literature.

We are sure that the desired node v_m is among the k nearest neighbors of u where $k = |C| + 1$. Therefore in the worst case, using a naive approach, $\mathrm{NN}_{d,\rho}$ amounts to perform a k-nearest neighbors search and then a simple check among them by using ρ. Note that k is a dynamic (growing) quantity and it is not possible to fix it in advance. The problem is of a different nature than finding the MST in a constrained degree graph.

Rewriting Algorithm 6 in terms of these new elements results in Algorithm 7. Our work is similar to Bentley and Friedman' [14]. They showed the pertinence of using nearest neighbors search structures to compute the MST in a Kruskal-like algorithm. Although they showed that the use of nearest neighbors search structures was fruitful for the computation of MSTs, their work received little attention.

Algorithm 7 Compute the minimum spanning tree of feature set X with metric d using nearest neighbors structures.

Require: $X \neq \emptyset$
Ensure: $T = (V, E_T)$ is the minimum spanning tree of G using d

$E_T \leftarrow \emptyset$
while $|E_T| < |V| - 1$ **do**
  $E' \leftarrow \emptyset$
  for all connected components C of T **do**
    $u_m \leftarrow \underset{u \in C}{\arg\min}(u, \mathrm{NN}_{d,\rho}(V, u))$
    $v_m \leftarrow \mathrm{NN}_{d,\rho}(V, u_m)$
    $\delta_m \leftarrow d(u_m, v_m)$
    $E' \leftarrow E' \cup \{(u_m, v_m, \delta_m)\}$
  end for
  while $E' \neq \emptyset$ **do**
    $(u_m, v_m, \delta_m) \leftarrow \underset{(u,v,\delta) \in E'}{\arg\min}\, d$
    $E' \leftarrow E' \setminus \{(u_m, v_m, \delta_m)\}$
    if $E_T \cup \{(u_m, v_m, d_m)\}$ does not contain cycles **then**
      $E_T \leftarrow E_T \cup \{(u_m, v_m, d_m)\}$
    end if
  end while
end while

Beside using of nearest neighbors, Bentley and Friedman [14] also use priority queues to prune the number of nearest neighbors searches performed during the algorithm. First let us explain that Kruskal's algorithm is greedy: it creates a forest (i.e. a set of trees) where each isolated edge is a tree and gradually merges these trees by adding the smallest edge whose endpoints lie on different trees. They propose to store the nodes of the partial (i.e. already computed) forest, along with their foreign nearest neighbors, in a single and global priority queue where the priority of a node is the inverse of the distance to its foreign nearest neighbor. The use of a priority queue is indeed interesting in this context, as the next edge to add to the MST is at the top of the priority queue. The top of the queue is removed and the top-priority foreign nearest neighbors is added to the MST. This node is also added to the queue, after computing its foreign nearest neighbors. Additionally, the priority queue must be updated, since disjoint connected components are merged and some foreign nearest neighbors might not be foreigners anymore.

Note that it may not be necessary to update the entire priority queue. This is because the current priority of each of these nodes (the priority before the insertion in the MST) serves as an upper bound of its real priority (the priority after the insertion in the MST). The real priority of a node needs only to be computed when its current priority is on the top of the queue. Thus, by using a global priority queue Bentley and Friedman were able to speed up a Kruskal-like algorithm.

We already stated our interest in building on Boruvka's algorithm, hence a global priority queue is not suitable in this case. Alternatively, we propose to use several priority queues, one for each connected component in the partial MST. Each queue, only holds the nodes in its respective connected component, and their foreign nearest neighbor. After an insertion in the MST, two connected components are merged and their priority queues are also merged.

Note that the space complexity is still $O(n)$. In the first iteration, there are n queues, each of length 1. In the second iteration there are roughly $n/2$ queues, each of length 2, and so on.

Algorithm 8 is the result of this modification. The operators top and pop return the top-priority element in the queue and remove it, respectively.

7.2.1 Approximate MST

We stated that our approach allows to compute approximate MSTs. Indeed, if we simply relax the search by finding the approximate nearest neighbors we end up with an approximate minimum spanning tree algorithm. Approximate nearest neighbors queries are much faster than exact ones, specially in high-dimensional spaces.

Typically, $\text{ANN}_d(X, u, \eta)$ ensures that, if the true nearest neighbor is at distance δ, the approximate nearest neighbor is at a distance lower than $\delta(1+\eta)$. Note that AMSTs can also be obtained by using a probability bound on the nearest neighbor distance [134].

Lai et al. [79] have previously studied AMSTs. Their approximation is obtained

Algorithm 8 Compute the minimum spanning tree of X with metric d using nearest neighbors structures and multiple priority queues.

Require: $X \neq \emptyset$
Ensure: $T = (V, E_T)$ is the minimum spanning tree of G using d

for all $v \in V$ **do** {each node is a connected component}
  $v_m \leftarrow \mathrm{NN}_{d,\rho}(V, v)$
  $d_m \leftarrow d(v, v_m)$
  add (v, v_m) to the empty priority queue $Q_{\{v\}}$ with priority d_m
end for
$E_T \leftarrow \emptyset$
while $|E_T| < |V| - 1$ **do**
  $E' \leftarrow \emptyset$
  for all connected components C of T **do**
    $(v, v_m) \leftarrow \mathrm{top}(Q_C)$
    while $v_m \in C$ **do** {not a foreign nearest neighbors}
      $\mathrm{pop}(Q_C)$
      $u_m \leftarrow (v, \mathrm{NN}_{d,\rho}(V, v))$
      $\delta_m \leftarrow d(v, u_m)$
      add (v, u_m) to Q_C with priority δ_m
      $(v, v_m) \leftarrow \mathrm{top}(Q_C)$
    end while
    $\delta_m \leftarrow d(v, v_m)$
    $E' \leftarrow E' \cup \{(v, v_m, \delta_m)\}$
  end for
  for all $(u, v) \in E'$ **do**
    $C \leftarrow$ connected component of T containing u
    $C' \leftarrow$ connected component of T containing v
    merge Q_C and $Q_{C'}$ into $Q_{C \cup C'}$
  end for
  while $E' \neq \emptyset$ **do**
    $(u_m, v_m, \delta_m) \leftarrow \operatorname{argmin}_{(u,v,\delta) \in E'} d$
    $E' \leftarrow E' \setminus \{(u_m, v_m, \delta_m)\}$
    if $E_T \cup \{(u_m, v_m, d_m)\}$ does not contain cycles **then**
      $E_T \leftarrow E_T \cup \{(u_m, v_m, d_m)\}$
    end if
  end while
end while

by using space-filling structures, i.e. Hilbert curves. Their work differs from ours in two central points. First, our algorithm allows to combine MSTs and AMSTs in a single framework, in which the only difference between them is a relaxation parameter. Their work is restricted to AMSTs. Second, Hilbert curves are fractal and the space-filling accuracy follows an exponential scale. It relies on a scale parameter that has a non-intuitive meaning and which is difficult to choose. It is not straightforward to set automatically a suitable scale for a given point set configuration. The relaxation parameter in our method has a clear interpretation and it is easy to monitor its effect.

7.3 Nearest Neighbors Search Structures

The problem of efficiently finding nearest neighbors has received much attention, as it is in the core of a great variety of problems in pattern recognition and classification. To name a few, non-parametric density estimation [57], non-linear manifold learning [131] and descriptors matching [86].

A plethora of nearest neighbors search structures have been proposed over the years:

- Kd-trees and randomized Kd-trees [13, 125],
- VP-trees and MVP-trees [141, 17],
- M-trees and their extensions [34, 127, 128, 145],
- hashing-based methods [2],
- hierarchical k-means trees [102]
- list-of-clusters [30, 31].

The above list is, of course, non-exhaustive.

All metric search structures, except hashing-based methods, exploit, one way or another, the metric's triangular inequality to reduce the number of distance computations to be performed in a search. Hashing-based methods aim at finding hash functions which ensure that the probability of collision is much higher for objects that are close to each other than for those that are far apart.

7.3.1 List-of-clusters

Now we turn our attention to the list-of-clusters structure [30, 31]. It is reported to be very efficient and resistant to the intrinsic dimensionality of the data set. Our experiments, shown in Table 7.2, support this claim. It can also be implemented in primary and in secondary memory. Furthermore, it has the advantage of being easy to implement and understand, as it does not involve complex data structures. For this reason we choose list-of-clusters to explain the modifications needed to find foreign nearest neighbors in depth.

The core of the list-of-clusters construction algorithm is the choice of a center $c \in X$ and a radius r_c. The possible choices for them will be discussed later. Let us define

Size	Dimensions	List-of-clusters	kd-tree	k-means tree	Linear
10^4	2	0.04	0.22	0.1	2.76
	3	0.03	0.32	0.17	2.64
	4	0.08	0.4	0.45	2.59
	5	0.04	0.61	1.11	2.56
	10	0.19	1.72	2.84	3.23
	20	0.8	5.62	4.16	4.48
10^5	2	0.31	1.33	0.3	47.65
	3	0.49	2.04	0.91	52.84
	4	0.72	3.07	2.42	50.19
	5	0.6	4.34	5.47	49.88
	10	2.7	14.51	53.52	59.26
	20	18.39	70.68	66.07	65.21

Table 7.2: Running-time comparison of search structures in the size and in the dimensionality of the dataset. The dataset is composed by uniformly distributed points. The values are the average of 1000 random queries and are expressed in milliseconds.

- **internal elements**:

$$I_{X,c,r_c} \stackrel{def}{=} \{x \in X - \{c\},\ d(c,x) \leq r_c\},$$

- **external elements**:

$$E_{X,c,r_c} \stackrel{def}{=} \{x \in X,\ d(c,x) > r_c\}.$$

A bucket is defined as the tuple (c, r_c, I_{X,c,r_c}). The process is repeated inside E_{X,c,r_c} recursively, producing at the end a list of buckets. This procedure is depicted in a recursion-free manner, in Algorithm 9. List-of-clusters, where only external elements are split, can be seen as a degenerated VP-tree, where external and internal elements are split.

There are two decisions to make at the core of the building algorithm: the selection of the center c and the radius r_c [31]. At iteration i, for selecting the center c_i, we chose the element farthest to c_{i-1} in the remaining set. The objective of such a choice is to minimize the overlap between regions. By using partitions of fixed size, the radius r_c is then easily deduced.

Now we focus our attention to search itself. The original search algorithm in the list-of-clusters structure iterates through the list of buckets and performs exhaustive searches only when needed (determined by triangular inequalities). An exhaustive search occurs within a bucket's internal elements. It can be written in

Algorithm 9 Build a list-of-clusters from X using metric d

Require: $X \neq \emptyset$
Ensure: $L_{X,d}$ is the list of clusters of X
  $L_{X,d} \leftarrow \emptyset$
  while $X \neq \emptyset$ **do**
    select a center $c \in X$
    select a radius r_c
    $I \leftarrow \{x \in X - \{c\},\ d(c,x) \leq r_c\}$
    $X \leftarrow X - I - \{c\}$
    $L_{X,d} \leftarrow L_{X,d} : (c, r_c, I)$
  end while

the following terms

$$x_m \leftarrow \arg\min_{x \in I} d(x,q) \tag{7.6}$$

$$d_m \leftarrow d(x_m, q). \tag{7.7}$$

The standard list-of-clusters search algorithm [31] is shown in Algorithm 10, with the introduction of the constraint ρ in the exhaustive search (lines 10 and 9). The operators head and tail return the first element in the list and remove it, respectively. Adding constraint ρ to any other search structure is similar.

Algorithm 10 can be very easily modified to find approximate nearest neighbors. It is sufficient to rewrite lines 4 and 8 as follows

$$d_c / r \leq 1 + \eta \tag{7.8}$$

$$d_c / (r_c + r) \leq 1 + \eta, \tag{7.9}$$

where η is a relaxation parameter. Note that more complex nearest neighbors formulations can be used, such as probabilistic bounds [134].

7.4 Experimental Results

As distance computations are the dominating speed factor, we measure performance and complexity as a function of them. We sample points from a uniform distribution in the unit hyper-cube. We tested with four different dimensionalities $\mathbb{R}^2$, $\mathbb{R}^5$, $\mathbb{R}^{10}$ and $\mathbb{R}^{20}$. We compared the following methods:

Bvka: all distances are precomputed and stored in memory and then Algorithm 6 is performed.

Bvka-O: Algorithm 7 where an online linear search is used to compute nearest neighbors.

Bvka-LOC: Algorithm 7 where nearest neighbors are computed online by using the list-of-clusters search structure.

Bvka-PQ-LOC: Algorithm 8 where nearest neighbors are computed online by using list-of-clusters search structure.

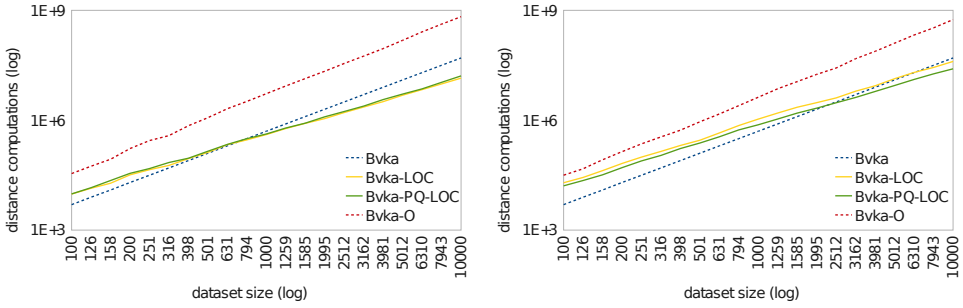
Algorithm 10 Search for the nearest neighbor p of q in the list-of-clusters $L_{X,d}$ with initial radius r and restriction ρ

Require: $L_{X,d}$ is not empty and $r > 0$
Ensure: $p \in X$ is the nearest neighbor of q if $\exists x \in X,\ d(x,q) \leq r$

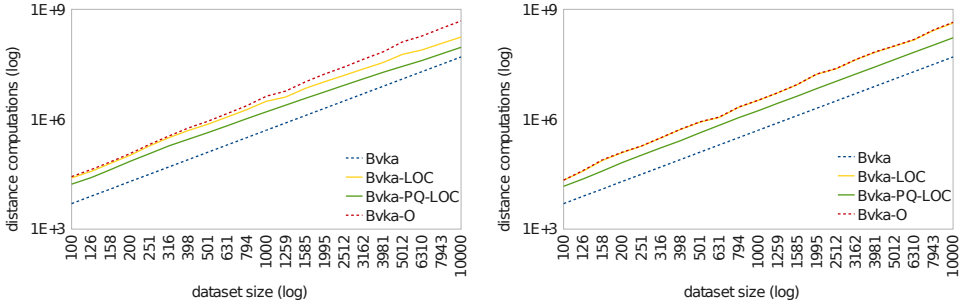
repeat
  $(c, r_c, I) \leftarrow \text{head}(L_{X,d})$
  $d_c \leftarrow d(c,q)$
  if $d_c \leq r$ **then**
    $p \leftarrow c$
    $r \leftarrow d_c$
  end if
  if $d_c \leq r_c + r$ **then**
    $x_m \leftarrow \arg\min_{\substack{x \in I \\ \rho(x)=1}} d(x,q)$
    $d_m \leftarrow d(x, x_m)$
    if $d_m \leq r$ **then**
      $p \leftarrow x_m$
      $r \leftarrow d_m$
    end if
  end if
  $L_{X,d} \leftarrow \text{tail}(L_{X,d})$
until $L_{X,d}$ is not empty or $d_c \leq r_c - r$

Method	Solution	Distances computed	Distances stored	Space	Search speed
Bvka	MST	$n(n-1)/2$	all	$\mathrm{O}(n^2)$	—
Bvka-O	MST	$\mathrm{O}(n^2 \log n)$	none	$\mathrm{O}(1)$	linear
Bvka-LOC	MST	$\mathrm{O}(\bar{s} n \log n)$	none	$\mathrm{O}(n)$	sub-linear
Bvka-PQ-LOC	MST	$\mathrm{O}(\bar{s} n \log n)$	$n-1$	$\mathrm{O}(n)$	sub-linear
Bvka-A η	AMST	$\mathrm{O}(\bar{s} n \log n)$	$n-1$	$\mathrm{O}(n)$	sub-linear

Table 7.3: The methods compared in this chapter. $\bar{s}$ stands for average number of distance operations needed to complete a nearest neighbors search. The space required by the nearest neighbors search structure is $\mathrm{O}(n)$.

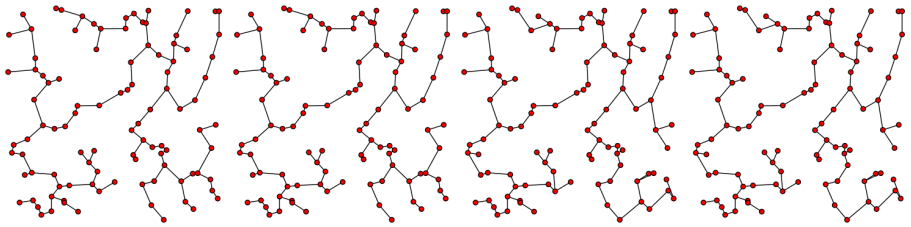


(a) $X \subset \mathbb{R}^2$ (b) $X \subset \mathbb{R}^5$



(c) $X \subset \mathbb{R}^{10}$ (d) $X \subset \mathbb{R}^{20}$

Figure 7.1: Comparison in the number of distance computations as $|X|$ grows. The radii in the list-of-clusters were chosen such that each bucket has $\sqrt{|X|/2}$ internal elements. Both scales are logarithmic.



(a) MST (b) AMST ($\eta = 0.1$) (c) AMST ($\eta = 0.2$) (d) AMST ($\eta = 0.5$)

Figure 7.2: Comparison of the MST (using Bvka) vs the AMST (using Bvka-A η) for several levels of relaxation η.

Method	$\mathbb{R}^2$	$\mathbb{R}^5$	$\mathbb{R}^{10}$	$\mathbb{R}^{20}$
Bvka	2	2	2	2
Bvka-O	2.14	2.12	2.13	2.15
Bvka-LOC	1.58	1.66	1.92	2.15
Bvka-PQ-LOC	1.61	1.6	1.87	2.03

Table 7.4: Slopes of the different curves in Figure 7.1 in a log-log scale. In low dimensions, Bvka-LOC is better than any classical algorithm while Bvka-PQ-LOC resists better the dimensionality increase.

Bvka-A η: Bvka-PQ-LOC modified to compute the AMST by using approximate nearest neighbors (see Equations 7.8 and 7.9) where η is the relaxation parameter.

A summary of these methods is presented in Table 7.3. Note that the reduced memory complexity of the algorithm warranties that we will be able to treat large datasets without "out of memory" issues.

Comparisons were made for relatively small feature sets ($|X| \leq 10^4$) to be able to compare with a classical MST implementation. A summary of our results is shown in Figure 7.1. At a first approach and as expected, all algorithms have a polynomial dependency on the dataset size, since they are linear in a log-log plot. In fact, we are not aiming at improving the worst case complexity of Boruvka's algorithm, but at improving its expected performance.

Results are indeed encouraging. Our method exhibits a very strong performance improvement in low dimensions, see Figures 7.1a and 7.1b. Bvka-LOC and Bvka-PQ-LOC in both cases outperforms Bvka several orders of magnitude.

We can also notice a strong performance degradation of Bvka-LOC with the increase of dimensionality, see Figures 7.1c and 7.1d. The only cause is the nearest neighbors search structure. It is a well known fact that the performance of nearest neighbors search structures tends to become linear in high-dimensions. In any case, our method is generic: any nearest neighbor structure can be used. Another structure may provide better results in high dimensions and we plan to explore these issues in future work.

Table 7.4 summarizes the results from Figure 7.1 by analyzing the slope of the different curves. The proposed approach lowers in practice the number of distance computations needed to solve the problem. The quadratic profiles of Bvka and Bvka-O are reduced to supralinear (e.g. $n^{1.6}$ approximately) by Bvka-LOC and Bvka-PQ-LOC. As stated, the latter shows a computational cost which is less sensitive to an increase in dimensionality.

We provide a simple example of the incidence of using the AMST, shown in Figure 7.2. We use X uniformly distributed on the square $[0,1]^2$ and Euclidean distance. Computing the MST required 9613 distance computations with our algorithm, while taking 9155, 8705 and 7840 with $\eta = 0.1$, $\eta = 0.2$, $\eta = 0.5$ respectively.

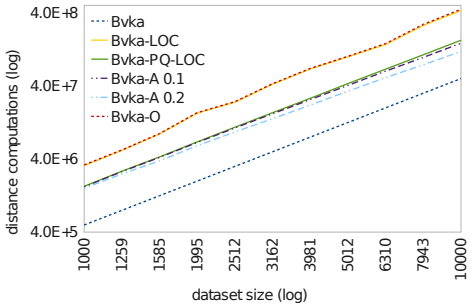


Figure 7.3: Comparison in the number of distance computations of the MST and the AMST algorithms for $\eta = 0.1$ and $\eta = 0.2$ with $X \subset \mathbb{R}^{20}$.

There is an important improvement in performance while the number of topology changes is small. Moreover, when carefully inspected, these changes are reasonable. It is a well known fact that (even little) jitter noise in the dataset greatly affects the topology of the MST [27]: computing the AMST can be seen as perturbing the dataset with such a noise.

Usually η is chosen to be quite small, and its use has more meaning in large and high-dimensional datasets. In our toy example, keeping η small does not introduce changes in the topology of the tree. We exaggerated η to show actual topology changes.

A performance comparison between MSTs and AMSTs is shown in Figure 7.3. We use X uniformly distributed in the hyper-cube $[0,1]^{20}$ and Euclidean distance. As argued before Bvka-LOC's performance tends to Bvka-O's in high-dimensions. Bvka-A greatly improves the performance: it is 1.7 and 1.62 times faster than Bvka-O and Bvka-LOC respectively when $|X| = 10^4$.

Computing the MST for $|X| = 10^5$ is not possible with classical algorithms on standard computers, since approximately $5 \cdot 10^9$ distances must be computed and stored. This means more than 18.6 GB if we use 32 bits to store each computed distance. Using minimum memory (less than 20 MB), we were able to compute the MST using Euclidean distance, without, explicitly nor implicitly, exploiting the nature of the Euclidean space (i.e. without relying on Delaunay triangulations). Table 7.5 presents the resulting running times for all considered algorithms. Again, these results can be improved, as we did not perform any tuning of the list-of-clusters.

Moreover, since parallelization is straightforward, it can be exploited to boost the performance. Every iteration in both nearest neighbors-searching cycles, i.e. lines 4 to 9 in Algorithm 7 and lines 9 to 20 in Algorithm 8, can be run in parallel since it operates on a single disjoint connected component.

Finally, more efficient search algorithms can be implemented for a given nearest neighbors structure that might increase the performance of the proposed algorithms, such as the best-bin-first or an optimized depth-first [120].

Dimensions	Bvka-PQ-LOC	Bvka-A 0.1	Bvka-A 0.2
$\mathbb{R}^2$	32	27	23
$\mathbb{R}^5$	85	63	48

Table 7.5: Running times (in seconds) on an Intel Core 2 Duo at 2.2 GHz for 10^5 uniformly distributed points using Euclidean distance

7.5 Application to Image Segmentation

We introduced Felzenszwalb and Huttenlocher' clustering method in Chapter 5 Section 5.2. In the original article [50] the method is used for image segmentation. The authors argue that:

> "There are several possible ways of determining which feature points to connect by edges. We connect each point to a fixed number of nearest neighbors. Another possibility is to use all the neighbors within some fixed distance δ. In any event, it is desirable to avoid considering all $O(n^2)$ pairs of feature points." [50]

They also test a version of their algorithm in which the graph layout is determined by the spatial neighborhood in the image. The previous citation is true in a context where all $n(n-1)/2$ distances are needed to compute the real MST.

On one side, it is clear that pruning the complete graph yields a faster algorithm. Nowadays a small resolution image has $640 \times 480 = 307200$ pixels. Using algorithms available in the literature, finding the MST is intractable.

On the other side, from a conceptual point of view, pruning is dangerous. Results can be severely affected, as discussed by Fowlkes and Malik [55] and Cour et al. [37]. The use of k-nearest neighbors graphs or fixed radius graphs also brings a new complication: both parameters have to be carefully selected to yield a connected graph. For example, if k is fixed such that the resulting graph is not fully connected, an image where all pixels have the same color would end up being segmented!

Based on the concepts of Section 7.2, we propose a new version of Felzenszwalb and Huttenlocher's algorithm. This version drops the original requirements of feature locality, becoming global. It exploits the fact that the proposed algorithm makes tractable the problem of computing the MST for a set of $10^5 \sim 10^6$ features.

In the following experiments, the color value in RGB space for each pixel is used as a feature in $\mathbb{R}^3$ and Euclidean distance is used to construct the MST. As stated above, any metric could have been used without changing the algorithm hence, in this sense, the algorithm is parametric on the metric.

For our tests we compared three different versions of Felzenszwalb and Huttenlocher's algorithm:

FH-Grid: Algorithm 3 using the image grid connectivity to prune the complete

graph

FH-Bvka-NN: Algorithm 3 using the complete graph and the proposed algorithm to compute the MST.

FH-Bvka-A η: Algorithm 3 using the complete graph and the proposed algorithm to compute the AMST.

We tested these algorithms on images from the Berkeley segmentation dataset [88]. Since their size is $321 \times 481 = 154401$ pixels, computing the MST with any classical (or even with Chazelle's state-of-the-art algorithm) is not possible.

The results on Figures 7.4, 7.5 and 7.6 (first three rows) show the importance of the role of global mechanisms in image segmentation.

Human visual system is able to capture and to use global characteristics. For example, we perceive a blue sky as a whole, even when a part appears through a hole in some object. Obviously, FH-Grid can not reproduce such behavior while global methods perform well.

By using the image connectivity, artifacts are introduced (clearly visible in all examples). Although this can be corrected by using locality directly on the color space, scale parameters are left free and must be correctly tuned to obtain satisfactory results.

Another issue can be observed in the methods obtained with FH-Grid: the number of regions is badly overestimated. Accuracy in the localization of visually perceived region frontiers comes at the price of over-segmenting the image.

As a counter part, all global methods we tested have difficulties when segmenting images where several objects share common colors. This is also natural and is a widely adopted mechanism by animals in the form of camouflage. Animals copy colors and structures from their environment to avoid being detected by predators or preys. More careful and local inspection are necessary to detect them.

The last three rows in Figures 7.4, 7.5 and 7.6 illustrate the stability of using FH-Bvka-A with different levels of approximation compared to the exact FH-Bvka-NN: changing η maintains the global structures in the segmentation.

7.6 Final Remarks

The dominating factor when computing the MST of a feature set X is the number of distance computations to be performed. We presented a method for computing the MST based on a clever use of nearest neighbors search structures. It has $\mathrm{O}\left(n^2\right)$ and $\mathrm{O}\left(n\right)$ time and space complexities respectively. However, in practice it outperforms classical algorithms for large, and low dimensional, datasets.

The same algorithm with a slight modification can also be used to compute the AMST: instead of finding nearest neighbors, one finds approximate nearest neighbors. In high-dimensional datasets, we showed the performance increase that results from using AMSTs. Moreover, in our tests the AMST, as computed, has an stable behavior.

To show the pertinence of the proposed algorithm, we use it to improve a state-of-the-art image segmentation algorithm proposed by Felzenszwalb and Hutten-

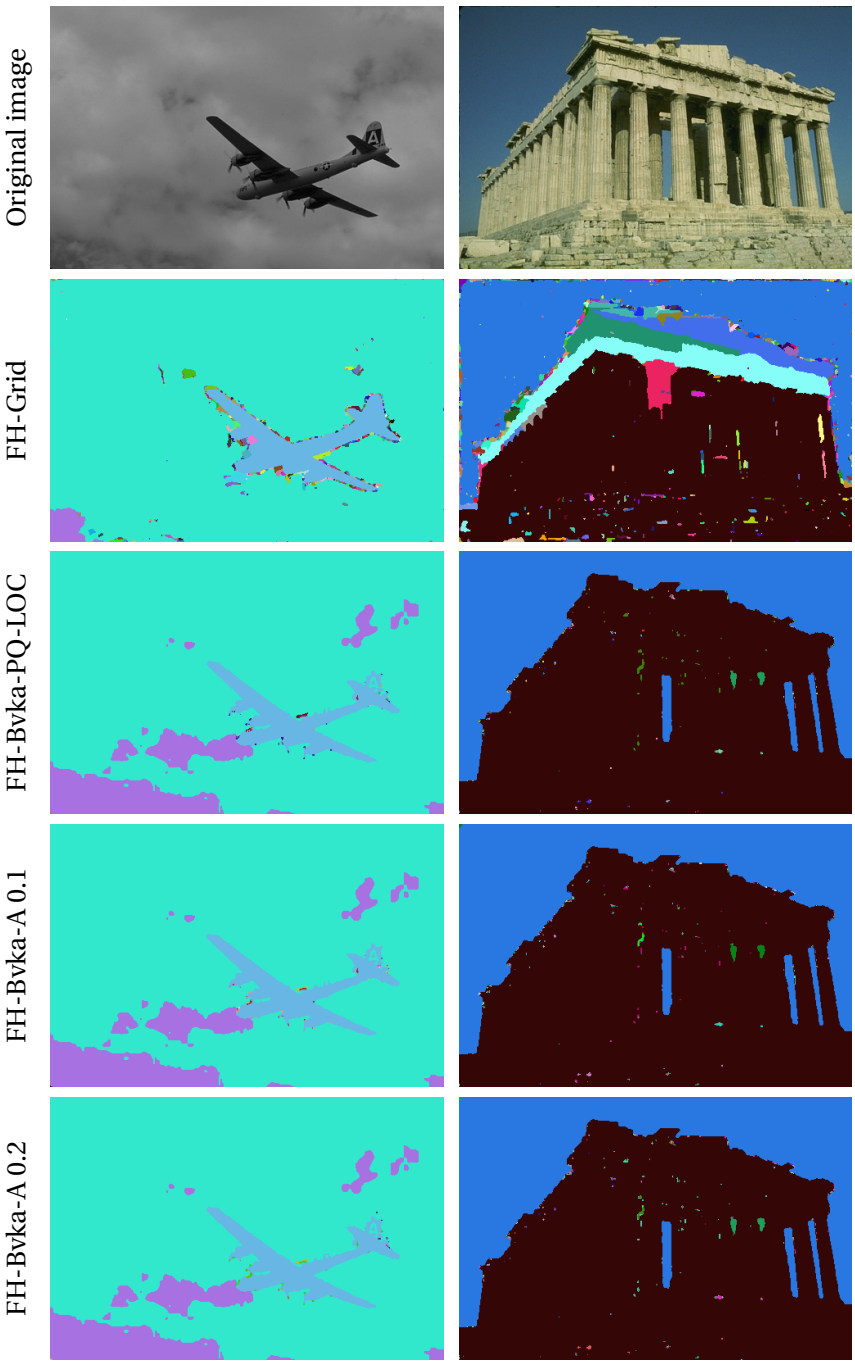


Figure 7.4: Effect of using local, complete and approximate complete graphs. The oversegmentation produced by FH-Grid has been corrected by all the proposed methods. Notice also that the approximate versions of FH-Bvka-A show an stable behavior.

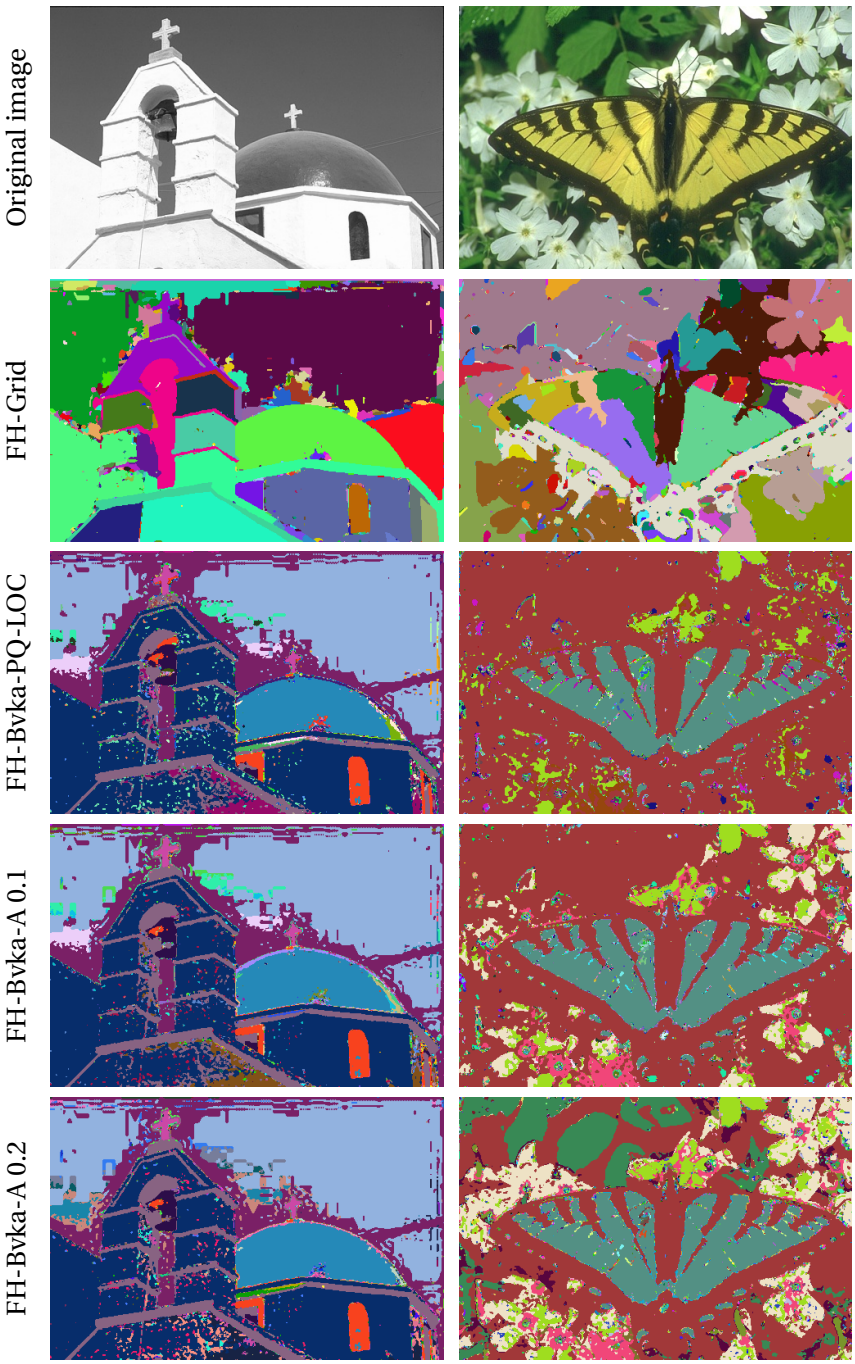


Figure 7.5: Effect of using local, complete and approximate complete graphs. The oversegmentation produced by FH-Grid has been corrected by all the proposed methods. Notice also that the approximate versions of FH-Bvka-A show an stable behavior.

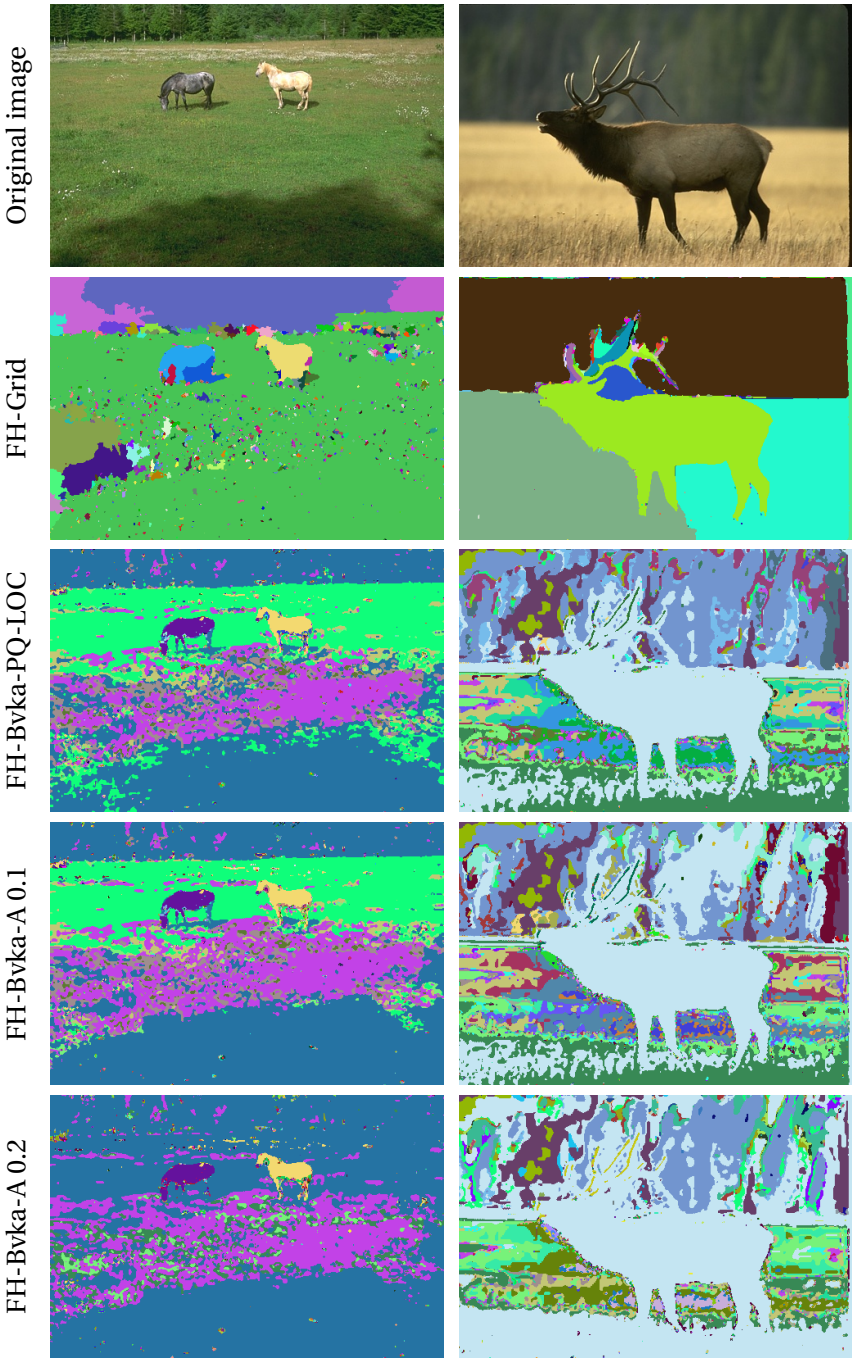


Figure 7.6: Effect of using local, complete and approximate complete graphs. The oversegmentation produced by FH-Grid has been corrected by all the proposed methods. Notice also that the approximate versions of FH-Bvka-A show an stable behavior.

locher [50]. Due to the memory and performance limitations of classical MST algorithms, Felzenszwalb and Huttenlocher's method is restricted to finding segmentations based on local connections or forced to rely on arbitrary connectivity parameters. Thanks to our input, a global segmentation can be obtained, without modifying the overall algorithm and without any extra parameter.

There are three conceptual main lines for future work. The first consists on performing an experimental evaluation of nearest neighbors search structures and their incidence on the performance of the proposed algorithm. This includes the evaluation of different criteria in list-of-clusters for selecting the centers and the radii. Second, we did not explore other search algorithms [120] which may reduce the number of distance computations per query. Finally, when using approximate MSTs, the trade-off between enhanced speed and accuracy must be explored more carefully.

Last, from the implementation point of view, the proposed algorithms can be parallelized without any reformulation. Moreover, in list-of-clusters, the exhaustive search within a bucket can be implemented using vectorial processors as the bucket size is fixed.

CHAPTER

8 Clustering using MST statistics

Abstract

In this chapter we propose a new clustering method that can be regarded as a numerical method to compute the proximity gestalt. The method analyzes edge length statistics in the MST of the dataset and provides an a contrario cluster detection criterion. The approach is fully parametric on the chosen distance and can detect arbitrarily shaped clusters. The method is also automatic, in the sense that only a single parameter is left to the user. This parameter has an intuitive interpretation as it controls the expected number of false detections. We show that the iterative application of our method can (1) provide robustness to noise and (2) solve a masking phenomenon in which a highly populated and salient cluster dominates the scene and inhibits the detection of less-populated, but still salient, clusters.

8.1 Introduction

Human perception is extremely adapted to group similar visual objects. Based on psychophysical experiments using simple 2D figures, the Gestalt school studied the perceptual organization, and identified a set of rules that govern human perception [138]. Each of these rules focuses on a single quality, or gestalt, many of which have been unveiled over the years.

One of the earlier and most powerful gestalts is proximity, which states that spatial or temporal proximity of elements may be perceived as a single group. Of course, the notion of distance is heavily embedded in the proximity gestalt. This is clearly illustrated in Figure 8.1. Two possible distances between the bars B_1 and B_2 that could be considered are

$$d_M(B_1,B_2) = \max_{\substack{p_1\in B_1\\ p_2\in B_2}} ||p_1-p_2||,$$

$$d_m(B_1,B_2) = \min_{\substack{p_1\in B_1\\ p_2\in B_2}} ||p_1-p_2||.$$

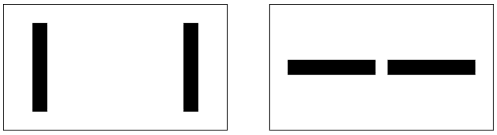


Figure 8.1: Two experiments with black bars. We perceive the bars on the left as more separated than the ones on the right. Nevertheless, there exists distances between sets that cannot capture the difference.

In this particular example $||.||$ denotes the euclidean norm. According to distance d_M, the bars are exactly at the same distance in both experiments, while according to distance d_m the bars on the right are closer to each other. In this case, the distance d_m seems to be more consistent with our perception.

The conceptual grounds on which our work is based were laid by Zahn in a seminal paper from 1971 [143]. Zahn faced the problem of finding perceptual clusters according to the proximity gestalt and proposed three key arguments:

1. **Only inter-point distances matter**. This imposes graphs as the only suitable underlying structure for clustering.
2. **No random steps**. Results must remain stable for all runs of the detection process. In particular, random initializations are forbidden.
3. **Independence from the exploration strategy**. The order in which points are analyzed must not affect the outcome of the algorithm.

These conceptual statements, together with the preference for d_m over d_M or other distances between sets, led Zahn to use the Minimum Spanning Tree (MST) as a building block for clustering algorithms. (The MST is the tree structure induced by the distance d_m [36].) Recently, psychophysical experiments performed by Dry et al. [44] supported this choice. In these experiments individuals were asked to connect points of 30 major star constellations, to show the structure they perceive. Two examples of constellations are shown in Figure 8.2. The outcome of these experiments was that, among five relational geometry structures, the MST and the Relative Neighborhood Graph (RNG) exhibit the highest degree of agreement with the empirical edges. In the RNG, explained in Chapter 5 Section 5.2.1, two points p and q are connected by an edge whenever there does not exist a third point r that is closer to both p and q than they are to each other. The MST is a subgraph of the RNG. Nonetheless the diagonal variance of both groups might suggest that sometimes other links not present nor in the MST nor in the RNG are used.

Zahn [143] suggested to cluster a feature set by eliminating the inconsistent edges in the minimum spanning tree. That is, instead of constructing a MST and as a consequence of the eliminations, a minimum spanning forest is built.

Since then, variations of the limited neighborhood set approaches have been extensively explored. The criteria in most works are based on local properties of the graph. Since perceptual grouping implies an assessment of local properties versus global properties, exclusively local methods must be discarded or patched.

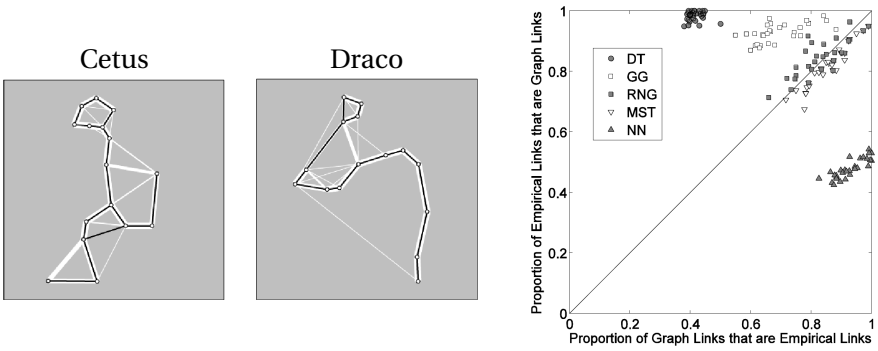


Figure 8.2: Left and middle: example constellations shown in black and the aggregated empirical structure shown in white. The number of persons that chose an edge is represented by the edge's width. Right: proportional overlap between graph and empirical structure links for Delaunay Triangulation (DT), Gabriel Graph (GG), Relative Neighborhood graph (RNG), Minimum Spanning Tree (MST), and Nearest Neighbors (NN). Each data point represents one of the 30 stimuli. Reproduced from [44].

For example, Felzenszwalb and Huttenlocher [50] and Bandyopadhyay [7] make use of the MST and RNG respectively, see Chapter 5 Sections 5.2.1 and 5.2.3. However, in order to correct local observations and to produce a reasonable clustering, they are forced to consider additional ad hoc global criteria.

The computation of the MST requires previous computation of the complete graph. This is a major disadvantage of MST-based clustering methods, that impose severe restrictions both on time and memory. The obvious workaround is to prune a priori the complete graph (e.g. in image segmentation, the image connexity is exploited), but unfortunately it might produce artifacts in the final solution. In Chapter 7 we proposed an efficient method to compute the MST on metric datasets. The use of this method allows for a significant performance boost over previous MST-based methods (e.g. [21, 50]), thus permitting to treat large datasets.

From an algorithmic point of view, the main problem with the Gestalt rules is their qualitative nature. Desolneux et al. developed a detection theory which seeks to provide a quantitative assessment of gestalts [40]. This theory is often referred as Computational Gestalt Theory and it has been successfully applied to numerous gestalts and detection problems [23, 62, 116]. It is primarily based on the Helmholtz principle which states that no structure is perceived in white noise. In this approach, there is no need to characterize the elements one wishes to detect but contrarily, the elements one wishes to avoid detecting.

In the light of this framework, Desolneux et al. analyzed the proximity gestalt, proposing a clustering algorithm [40]. It is founded on the idea that clusters are groups of points contained in a relatively small area. In other words, by counting points and computing the area that encloses them, one can assess the exceptionality of a given group of points.

The method proposed by Desolneux et al. [40] suffers from some problems. First, it can only be applied to points in an Euclidean 2D space. Second, in order to compute the enclosing areas, the space has to be discretized a priori and such discretization is used to compute the enclosing areas; of course, different discretizations lead to different results. Last, two phenomena called collateral elimination and faulty union in [21] occur when an extremely exceptional cluster hides or masks other less but still exceptional ones.

Cao et al. [21] continued this line of research extending the clustering algorithm to higher dimensions and corrected the collateral elimination and faulty union issues, by introducing what they called indivisibility criterion. However, as their method is also based on counting points on a given region, it is still required that a set of candidate regions is given a priori. The set of test regions is defined to be a set $\mathcal{R}$ of hyper-rectangles parallel to the axes and of different edge lengths, centered at each data point. The choice of $\mathcal{R}$ is application specific since it is intrinsically related to cluster size/scale. For example, an exponential choice for the discretization of the rectangle space is made by Cao et al. [21] introducing a bias for small rectangles (since they are more densely sampled). Then each cluster must be fitted by an axis-aligned hyper-rectangle $R \in \mathcal{R}$, meaning that clusters with arbitrary shapes are not detected. Even hyper-rectangular but diagonal clusters may be missed or oversegmented. A probability law modeling the number of points that fall in each hyper-rectangle $R \in \mathcal{R}$, assuming no specific structure in the data, must be known a priori or estimated. Obviously, this probability depends on the dimension of the space to be clustered.

In Chapter 6 we introduced the concept of graph-based a contrario clustering. A key element in this method is that the area can be computed from a weighted graph, where the edge weight represents the distance between two points, using non-parametric density estimation. Since only distances are used, the dimensionality of the problem is reduced to one. However, since this method is conceived for complete graphs, it suffers from a high computational burden.

There is an additional concept behind clustering algorithms that was not stated before: a point, to belong to a cluster, must be similar to all points in the cluster or only to some of them? All the described region-based solutions imply choosing the first option since, in some sense, all distances within a group are inspected. Table 8.1 shows on which side some algorithms are. Since our goal is to detect arbitrarily shaped clusters, we must place ourselves in the second group. We can do this by using the MST.

Our goal is to design a clustering method that can be considered a quantitative assessment of the proximity gestalt. Hence we propose a clustering method based on analyzing the distribution of distances of MST edges. The formulation naturally allows to detect clusters of arbitrary shapes. The use of trees, as minimally connected graphs, also leads to a fast algorithm.

The approach is fully automatic in the sense that the user input only relates to the nature of the problem to be treated and not the clustering algorithm itself. Strictly speaking it involves one single parameter that controls the degree

a point must be similar	
to all points in the cluster	to at least one point in the cluster
k-means	single-link algorithm [57]
Cao et al. [21]	Mean Shift [35]
Tepper et al. (Chapter 6)	Felzenszwalb and Huttenlocher [50]

Table 8.1: Conceptually there are two different ways to form a cluster. To belong to a cluster a point must be similar to all points in the cluster or to at least one point in the cluster. All algorithms explicitly or implicitly chose one or the other.

of reliability of the detected clusters. However, these methods can be considered parameter-free, as the result is not sensitive to the parameter value.

As the method relies on the sole characterization of non-clustered data, it is thus capable of detecting non-clustered data as such. In other words, in the absence of clustered data, the algorithm yields no detections.

We finally illustrate a masking phenomena where a highly populated cluster might occlude or mask less populated ones, showing that the iterative application of the MST-based clustering method is able to cope with this issue, thus solving very complicated clustering problems.

8.2 A New Clustering Method: Proximal Meaningful Forest

We now propose a new method to find clusters in graphs that is independent from their shape and from their dimension. We first build a weighted undirected graph $G = (X, E)$ where X is a set of features in a metric space (M, d) and the weighting function ω is defined in terms of the corresponding distance function

$$\omega((v_i, v_j)) = d(x_i, x_j). \tag{8.1}$$

8.2.1 The Minimum Spanning Tree

Informally, the Minimum Spanning Tree (MST) of an undirected weighted graph is the tree that covers all vertices with minimum total edge cost.

Given a metric space (M, d) and feature set $X \subseteq M$, we denote by $G = (X, E)$ the undirected weighted graph where $E = X \times X$ and the graph's weighting function $\omega : E \to \mathbb{R}$ is defined as

$$\omega((x_i, x_j)) = d(x_i, x_j) \quad \forall x_i, x_j \in X. \tag{8.2}$$

The MST $T = (X, E_T)$ of the feature set X is defined as the MST of G. A very important and classical property of the MST is that a hierarchy of point groups can be constructed from it.

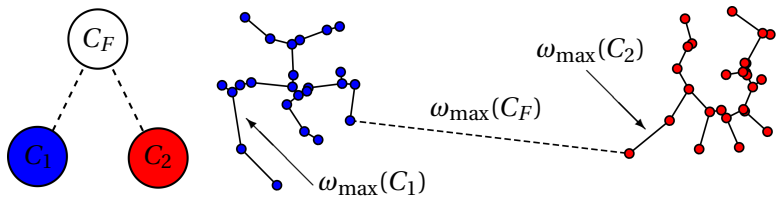


Figure 8.3: Part of a minimal spanning tree. The blue node set and the red node set are linked by the dashed edge, creating a new node in the minimal spanning tree.

Notation 5. *Let $T = (X, E_T)$ be the minimum spanning tree of X. For a group of points $C \in X$, we denote*

$$E(C) = \{(v_i, v_j) \mid v_i, v_j \in C \wedge (v_i, v_j) \in E_T\} \tag{8.3}$$

The edges in $E(C)$ are sorted in non-decreasing order, i.e.

$$\forall\ e_i, e_j \in E(C),\ i < j \Rightarrow \omega(e_i) \leq \omega(e_j)$$

Definition 29. *Let $T = (X, E_T)$ be the minimum spanning tree of X. A component $C \subseteq X$ is a set such that the graph $G = (C, E(C))$ is connected and*

- $\exists\ v \in V,\ C = \{v\}$ *or*
- $\nexists\ C' \in X,\ C \subset C' \quad \wedge \quad \omega_{\max}(C) > \omega_{\max}(C')$,

where $\omega_{\max}(C) = \max_{e \in E(C)} \omega(e)$. A single-link hierarchy $\mathcal{T}$ is the set of all possible components.

It is important to notice what the single-link hierarchy implies: given two components $C_1, C_2 \in \mathcal{T}$, it suffices that there exists a pair of vertices, one in C_1 and one in C_2 that are sufficiently near each other to generate a new component $C_F \in \mathcal{T}$, such that $C = C_1 \cup C_2$ and

$$\omega_{\max}(C_F) = \min_{\substack{v_i \in C_1, v_j \in C_2 \\ (v_i, v_j) \in E_T}} \omega((v_i, v_j)). \tag{8.4}$$

An example is depicted in Figure 8.3. The direct consequence of this fact is that the use of the single-link hierarchy for clustering provides a natural way to deal with clusters of different shapes.

All minimum spanning tree algorithms are greedy. From Definition 29 and Equation 8.4, in the single-link hierarchy the component $C_F = C_1 \cup C_2$ is the father of C_1 and C_2 and

$$\omega_{\max}(C_F) \geq \omega_{\max}(C_1) \tag{8.5}$$

$$\omega_{\max}(C_F) \geq \omega_{\max}(C_2). \tag{8.6}$$

With the objective of finding a suitable partition and to the best of our knowledge, Felzenszwalb and Huttenlocher [50] were the first to compare $\omega_{\max}(C_F)$ with

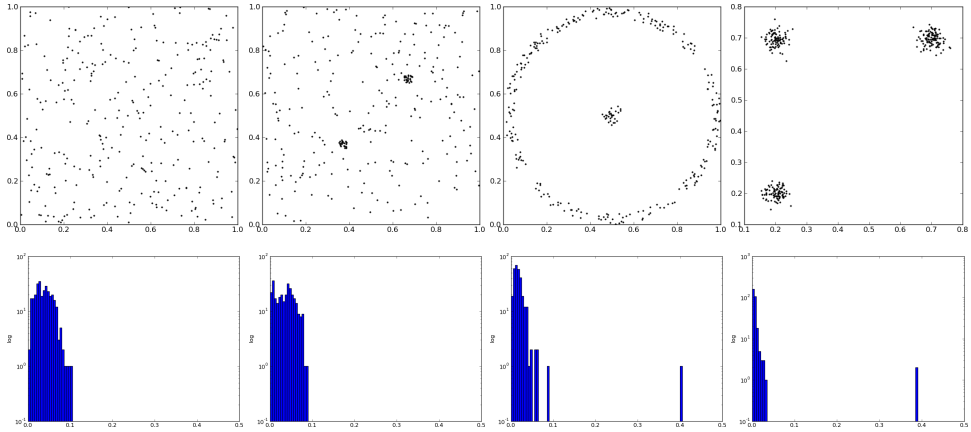


Figure 8.4: Histograms (in logarithmic scale) of MST edges from different point configurations. The non-clustered case (first column) differs from the other cases. Notice that clustered configurations also differ from each other.

$\omega_{\max}(C_1)$ and $\omega_{\max}(C_2)$, with an additional correction factor τ. Reprising Chapter 5 Section 5.2.3 with our new notation, C_1 and C_2 are only merged if

$$\min\left[\omega_{\max}(C_1)+\tau(C_1),\ \omega_{\max}(C_2)+\tau(C_2)\right] \geq \omega_{\max}(C_F). \tag{8.7}$$

In practice τ is defined as $\tau(C) = s/|C|$ where s plays the role of a scale parameter. The above definition presents a few problems. First, τ (i.e. s) is a global parameter and experiments show that clusters with different sizes and densities might not be recovered with this approach (Figure 8.10a). Second, there is not an easy rule to fix τ or s and, although it can be related with a scale parameter, there is no way to predict which specific value is best suited for a particular problem.

The exploration of similar ideas, while bearing in mind their shortcomings, leads us to a new clustering method.

8.2.2 Proximal Meaningful Forest

First, let us observe that the edge length distribution of an MST of a configuration of clustered points differs significantly from that of an unclustered point set (Figure 8.4). As a general idea, by knowing how to model unclustered data, one could detect clustered data by measuring some kind of dissimilarity between both.

Concretely, we are looking to evaluate the probability of occurrence, under the background model (i.e. unclustered data), of a random set $\mathscr{C}$ which exhibit the characteristics of a given observed set C. Both sets have the same cardinality, i.e. $|E(C)| = |E(\mathscr{C})| = K$.

The general principle has been previously explored. In 1983, following the same rationale Hoffman and Jain [65] proposed a similar idea: to perform a test of randomness. They built a null hypothesis using the edge length distribution

of the MST and they performed a single test analyzing whether the whole dataset belongs to the random model or not by computing the difference between the theoretical and the empirical CDF. Jain et al. [69] further refined this work, by using heuristic computations to separate the dataset into two or more subsets which were then tested using a two sample test statistic. Barzily et al. recently continued this line of work [11]. This approach introduces a bias towards the detection of compact (i.e. non-elongated) and equally sized clusters [11].

Notation 6. *Let $\mathcal{P}$ be a partition of $\mathbb{R}$, and $P \in \mathcal{P}$ such that $\omega_{\max}(C_F) \in P$.*

We also denote by e_i the i-th edge of $E(C)$ and by γ_i the i-th edge of $E(\mathscr{C})$. Following Equation 8.7 which proved successful as a decision rule to detect clusters and associating it with Equations 8.5 and 8.6, we compute

$$\begin{aligned}\Pr\Big(\omega(\gamma_1) < \omega(e_1), \ldots, \omega(\gamma_K) < \omega(e_K) \mid \omega_{\max}(\mathscr{C}_F) \in P\Big) \\ = \prod_{i=1}^{K} \Pr\Big(\omega(\gamma_i) < \omega(e_i) \mid \omega_{\max}(\mathscr{C}_F) \in P\Big) \\ \leq \Pr\Big(\omega_{\max}(\mathscr{C}) < \omega_{\max}(C) \mid \omega_{\max}(\mathscr{C}_F) \in P\Big)^K \end{aligned} \tag{8.8}$$

Unfortunately, conditioning by $\omega_{\max}(\mathscr{C}_F) = \omega_{\max}(C_F)$ is not practical: $\omega_{\max}(\mathscr{C}_F)$ is a real random variable and thus the event has null probability.

Definition 30. *Let $C \in X$ be a component of the single-link hierarchy $\mathcal{T}$ induced by the minimum spanning tree $T = (X, E_T)$ such that $|C| > 1$. We define the probability of false alarms (PFA) of C as*

$$\mathrm{PFA}(C) \stackrel{\text{def}}{=} \Pr(C \mid \mathcal{H}_0) = \Pr\Big(\omega_{\max}(\mathscr{C}) < \omega_{\max}(C) \mid \omega_{\max}(\mathscr{C}_F) \in P\Big)^K \tag{8.9}$$

The constraint $|C| > 1$ is needed since $E(C) = \emptyset$ when $|C| = 1$. Note that sets consisting of a single node must certainly not be detected. Conceptually, even when they are isolated, they constitute an outlier and not a cluster. We simply do not test such sets.

To detect unlikely dense subgraphs, a threshold is necessary on the PFA. In the classical *a contrario* framework, a new quantity is introduced: the Number of False Alarms (NFA), i.e. the product of the PFA by the number of tested candidate clusters. The NFA has a more intuitive meaning than the PFA, since it is an upper bound on the expectation of the number of false detections [40]. The threshold is then easily set on the NFA.

Definition 31 (Number of false alarms)**.** *We define the number of false alarms (NFA) of C as*

$$\mathrm{NFA}(C) \stackrel{\text{def}}{=} (|X| - 1) \cdot \mathrm{PFA}(C) \tag{8.10}$$

Notice that, by definition, $|X| - 1$ is the number of non-singleton sets in the single-link hierarchy.

Definition 32 (Meaningful component)**.** *A component C is ε-meaningful if*

$$\mathrm{NFA}(C) < \varepsilon \tag{8.11}$$

In the following, it is important to notice a fact about the single-link hierarchy. The components are mainly determined by the sorted sequence of the edges from the original graph; this follows directly from Kruskal's algorithm [36]. However, the components are independent of the differences between the edges in that sorted sequence: only the order matters and not the actual weights of the edges.

We reproduce Lemma 1 in Chapter 3.

Lemma 5. *Let X be a real random variable and let $F(x) = P(X \leq x)$ be the cumulative density function of X. Then for all $t \in (0,1)$,*

$$\Pr(F(X) < t) \leq t \tag{8.12}$$

Lemma 6. *The expected number of ε-meaningful clusters in a random single-link hierarchy (i.e. issued from the background model) is lower than ε.*

Proof. We follow the scheme of Proposition 1 from the work by Cao et al. [23]. Let $\mathcal{T}$ be random single-link hierarchy. For brevity let $M = |X| - 1$. Let Z_i be a binary random variable equal to 1 if $C_i \in \mathcal{T}$ is meaningful and 0 else. Let Y_i be a binary random variable equal to 1 if

$$M \cdot \Pr\Big(\omega_{\max}(\mathscr{C}) < \omega_{\max}(C_i) \mid \omega_{\max}(\mathscr{C}_F) \in P_i\Big)^{K_i} < \varepsilon \tag{8.13}$$

and 0 else. Let us denote by $\mathbb{E}(X)$ the expectation of a random variable X in the a contrario model. We then have

$$\mathbb{E}\left(\sum_{i=1}^{M} Z_i\right) = \mathbb{E}\left(\mathbb{E}\left(\sum_{i=1}^{M} Z_i \mid M\right)\right). \tag{8.14}$$

Of course, M is independent from the sets in $\mathcal{T}$. Thus, conditionally to $M = m$, the law of $\sum_{i=1}^{M} Z_i$ is the law of $\sum_{i=1}^{M} Y_i$. By linearity of expectation,

$$\mathbb{E}\left(\sum_{i=1}^{M} Z_i \mid M = m\right) = \mathbb{E}\left(\sum_{i=1}^{m} Y_i\right) = \sum_{i=1}^{m} \mathbb{E}(Y_i). \tag{8.15}$$

Since Y_i is a Bernoulli variable,

$$\begin{aligned}\mathbb{E}(Y_i) &= \Pr(Y_i = 1) = \Pr\left(M \cdot \Pr\Big(\omega_{\max}(\mathscr{C}) < \omega_{\max}(C_i) \mid \omega_{\max}(\mathscr{C}_F) \in P_i\Big)^{K_i} < \varepsilon\right) \\ &= \sum_{k=0}^{\infty} \Pr\left(M \cdot \Pr\Big(\omega_{\max}(\mathscr{C}) < \omega_{\max}(C_i) \mid \omega_{\max}(\mathscr{C}_F) \in P_i\Big)^{K_i} < \varepsilon \;\middle|\; K_i = k\right) \cdot \Pr(K_i = k).\end{aligned} \tag{8.16}$$

We have assumed that K_i is independent from $\Pr\Big(\omega_{\max}(\mathscr{C}) < \omega_{\max}(C_i) \mid \omega_{\max}(\mathscr{C}_F) \in P_i\Big)$. Thus, conditionally to $K_i = k$, the law of $M \cdot \Pr\Big(\omega_{\max}(\mathscr{C}) < \omega_{\max}(C_i) \mid \omega_{\max}(\mathscr{C}_F) \in P_i\Big)^{K_i}$ is the law of $M \cdot \Pr\Big(\omega_{\max}(\mathscr{C}) < \omega_{\max}(C_i) \mid \omega_{\max}(\mathscr{C}_F) \in P_i\Big)^{k}$. We have

$$\begin{aligned}
&\Pr\left(m \cdot \Pr\Big(\omega_{\max}(\mathscr{C}) < \omega_{\max}(C_i) \mid \omega_{\max}(\mathscr{C}_F) \in P_i\Big)^{k} < \varepsilon\right) \\
&= \Pr\left(\Pr\Big(\omega_{\max}(\mathscr{C}) < \omega_{\max}(C_i) \mid \omega_{\max}(\mathscr{C}_F) \in P_i\Big) < \left(\frac{\varepsilon}{m}\right)^{1/k}\right) \\
&= \Pr\left(\max_{\gamma \in E(\mathscr{C})} \Pr\Big(\omega(\gamma) < \omega_{\max}(C_i) \mid \omega_{\max}(\mathscr{C}_F) \in P_i\Big) < \left(\frac{\varepsilon}{m}\right)^{1/k}\right) \\
&\qquad = \prod_{j=1}^{k} \Pr\left(\Pr\Big(\omega(\gamma_j) < \omega_{\max}(C_i) \mid \omega_{\max}(\mathscr{C}_F) \in P_i\Big) < \left(\frac{\varepsilon}{m}\right)^{1/k}\right) \leq \frac{\varepsilon}{m}. \quad (8.17)
\end{aligned}$$

The last implication follows from Lemma 5. This term does not depend on l, thus

$$\begin{aligned}
&\sum_{k=0}^{\infty} \Pr\left(M \cdot \Pr\Big(\omega_{\max}(\mathscr{C}) < \omega_{\max}(C_i) \mid \omega_{\max}(\mathscr{C}_F) \in P_i\Big)^{K_i} < \varepsilon \;\Big|\; K_i = k\right) \cdot \Pr(K_i = k) \\
&\qquad \leq \frac{\varepsilon}{m} \sum_{k=0}^{\infty} \Pr(K_i = k) = \frac{\varepsilon}{m}. \quad (8.18)
\end{aligned}$$

Hence,

$$\mathbb{E}\left(\sum_{i=1}^{M} Z_i \mid M = m\right) \leq \varepsilon. \quad (8.19)$$

This finally implies $\mathbb{E}\left(\sum_{i=1}^{M} Z_i\right) \leq \varepsilon$, what means that the expected number of meaningful clusters is less than ε. □

8.2.3 The background model

The distribution $\Pr\Big(\omega_{\max}(\mathscr{C}) < \omega_{\max}(C) \mid \omega_{\max}(\mathscr{C}_F) \in P\Big)$ is not known a priori. Moreover, up to our knowledge there is no analytical expression for the cumulative edge distribution under $\mathcal{H}_0$ for the MST [65]. We estimate this distribution by performing Monte Carlo simulations of the background process.

Algorithm 11 Compute $\Pr\Big(\omega_{\max}(\mathscr{C}) < \omega_{\max}(C) \mid \omega_{\max}(\mathscr{C}_F) \in P\Big)$ for a set of N points by Q Monte Carlo simulations.

for all q such that $1 \leq q \leq Q$ **do**
  $X \leftarrow$ draw N points from the background point process.
  build the single-link hierarchy $\mathcal{T}_k$ form the MST of X.
end for
compute a conditional histogram from the set $\{\mathcal{T}_k\}_{q=1\ldots Q}$

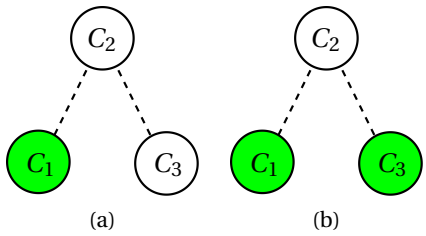


Figure 8.5: Example of collateral elimination. Three components C_1, C_2, C_3 such that $C_1 \subset C_2$, $C_3 \subset C_2$ and $\text{NFA}(C_1) < \text{NFA}(C_2) < \text{NFA}(C_3) < \varepsilon$. (a) The classical maximality rule only selects C_1 as a maximal component. (b) The scheme in Algorithm 12 selects C_1 and C_3.

Classically, one defines a point process and a sampling window. Hoffman and Jain [65] point out that the sampling window for the background point process is usually unknown for a given dataset. They use the convex hull arguing that it is the maximum likelihood estimator of the true sampling window for uniformly distributed two-dimensional data. In the experiments from Section 8.3, we simply use the minimum hiper-rectangle that contains the whole dataset as the sampling window. However, there are problems where theirs intrinsic characteristics allow to define other background processes that do not involve a sampling window.

8.2.4 Eliminating redundancy

While each meaningful cluster is relevant by itself, the whole set of meaningful components exhibits, in general, high redundancy: a meaningful component C_1 can contain another meaningful component C_2 [21]. This question can be answered by comparing $\text{NFA}(C_1)$ and $\text{NFA}(C_2)$ using Definition 32. The group with the smallest NFA must of course be preferred. Classically, the following rule

for all ε-meaningful clusters C_1, C_2 **do**
  if $C_2 \subset C_1 \ \vee \ C_1 \subset C_2$ **then**
    eliminate $\arg\max(\text{NFA}(C_1), \text{NFA}(C_2))$
  end if
end for

would have been used to perform the pruning of the set of meaningful components. Unfortunately, it leads to a phenomenon described in [21], where it was called collateral elimination. A very meaningful component can hide another meaningful sibling, as illustrated in Figure 8.5.

The single-link hierarchy offers an alternative scheme to prune the redundant set of meaningful components, profiting from the inclusion properties of the dendrogram structure. It is non-other than the exclusion principle, defined first by Desolneux et al. [40], which states that

> Let A and B be groups obtained by the same gestalt law. Then no point

x is allowed to belong to both A and B. In other words each point must either belong to A or to B.

A simple scheme for applying the exclusion principle is shown in Algorithm 12.

Since we are choosing the components that are more in accordance with the proximity gestalt, we call the resulting components Proximal Meaningful Components (PMC). Then, we say that the set of all proximal meaningful components is a Meaningful Clustered Forest (MCF).

Algorithm 12 Eliminate redundant components from the set $\mathcal{M}$ of meaningful components.

$\mathcal{F} \leftarrow \emptyset$
while $\mathcal{M} \neq \emptyset$ **do**
  $C_{\min} \leftarrow \arg\min_{C \in \mathcal{M}} \mathrm{NFA}(C)$
  eliminate $C_{\min}$ from $\mathcal{M}$
  eliminate all components C from $\mathcal{M}$ such that $C \subset C_{\min}$
  eliminate all components C from $\mathcal{M}$ such that $C_{\min} \subset C$
  add $C_{\min}$ to $\mathcal{F}$
end while
$\mathcal{M} \leftarrow \mathcal{F}$

8.3 Experiments on Synthetic examples

As a sanity check, we start by testing our method with simple examples. Figure 8.6 present clusters which are well but not linearly separated. The meaningful clustered forest describes correctly the structure of the data.

Figure 8.7 shows an example of cluster detection in a dataset overwhelmed by outliers. The data consists of 950 points uniformly distributed in the unit square, and 50 points manually added around the positions $(0.4, 0.4)$ and $(0.7, 0.7)$. The figure shows the result of a numerical method involving the above NFA. The background distribution is chosen to be uniform in $[0,1]^2$. Both visible clusters are found and their NFAs are respectively 10^{-15} and 10^{-8}. Such low numbers can barely be the result of chance.

The case of mixture of Gaussians, shown in Figure 8.8, provides an interesting example. On the tails, points are obviously sparser and the distance to neighboring points grows. Since we are looking for tight components, the tail might be discarded, depending on the Gaussian variance.

The example in Figure 8.9 consists of a very complex scene, composed of clusters with different densities, shapes and sizes. Proximal components (i.e. we avoid testing $\mathrm{NFA} < \varepsilon$) are displayed. Even when no decision about the statistical significance is made, the recovered clusters describe, in general, the scene accurately. Some oversplitting can be detected in proximal components. When a decision is made and only meaningful components are kept, we realize that the oversplit

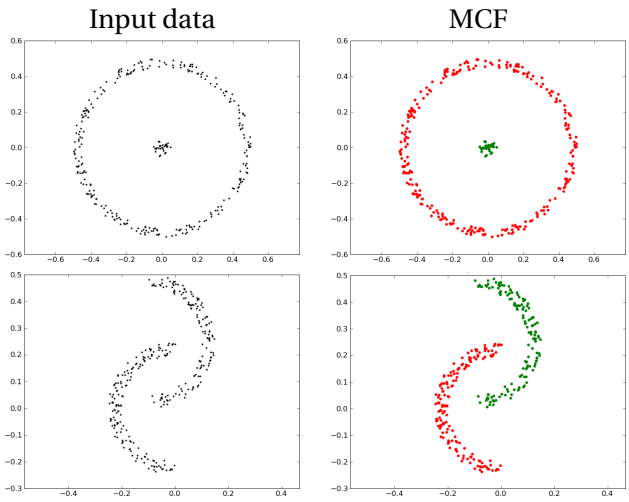


Figure 8.6: The meaningful clustered forest correctly describes the points organization, even when clusters have arbitrary shapes.

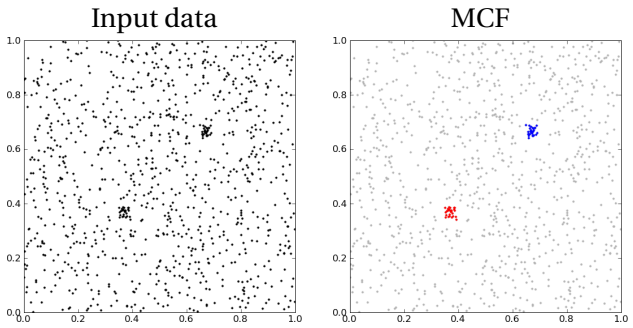


Figure 8.7: Similar experiment as performed by Cao et al. in Figure 2 [23]. Clustering of twice 25 points around $(0.4, 0.4)$ and $(0.7, 0.7)$ surrounded by 950 i.i.d. points, uniformly distributed in the unit square. Exactly two proximal meaningful components are detected.

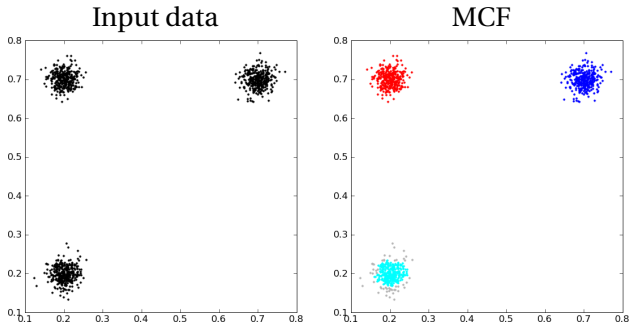


Figure 8.8: Clusters are correctly recovered in the mixture of three Gaussians. However some points are detected as noise (depicted in gray) in the tails.

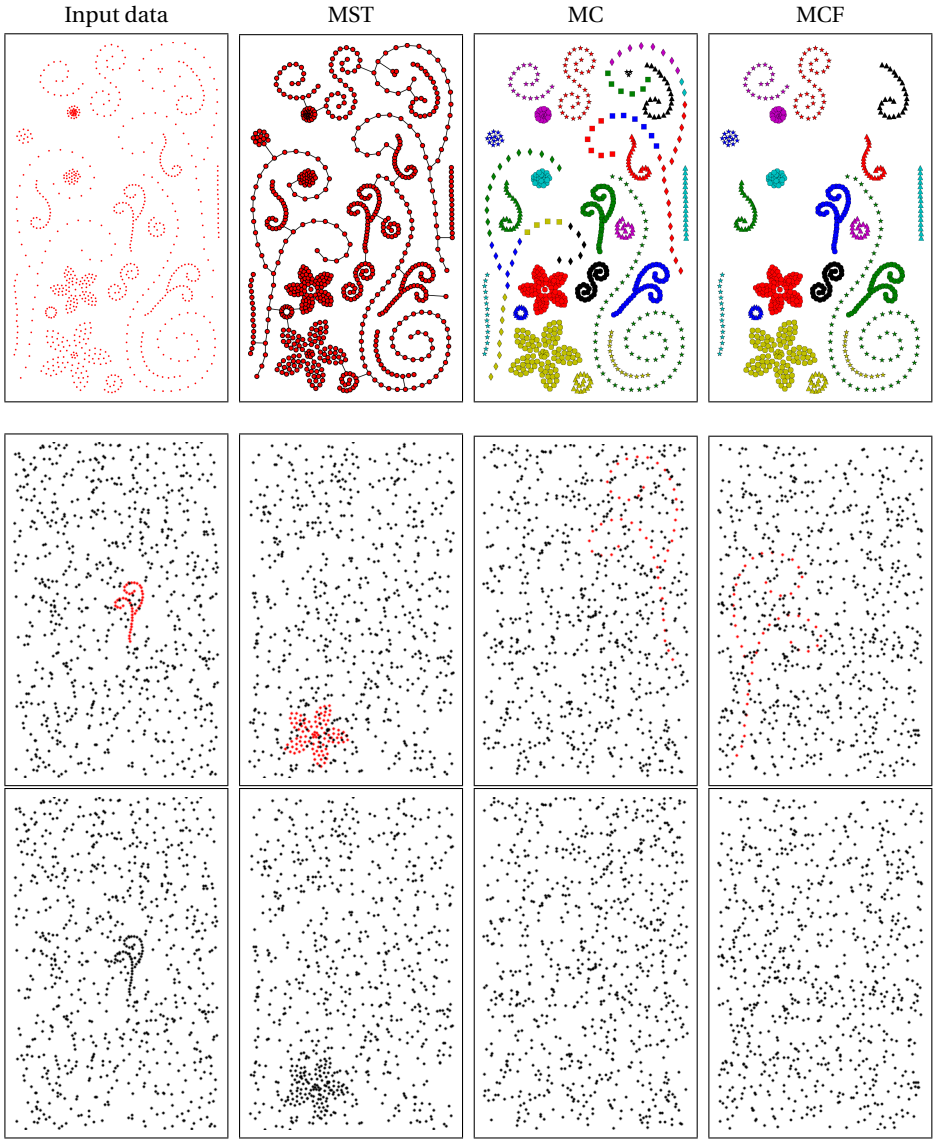
figures are not meaningful. As a sanity check, in Figure 8.9a we plot some of the detected structures superimposed to a realization of the background noise model. The input data in Figure 8.9 contains 764 points and for a given shape in it, with W points, we plot the shape and $764 - W$ points drawn from the background model. Among proximal components, the meaningful ones can be clearly perceived while non-meaningful ones are unnoticed.

Our results are compared with Felzenszwalb and Huttenlocher' algorithm (denoted by FH in the following), that was briefly described in Section 5.2.3 Chapter 5, and with Mean Shift [35, 58]. Mean Shift performs a non-parametric density estimation (using sliding windows) and finds its local maxima. Clusters are determined by what Comaniciu and Meer call "basins of attraction" [35]: points are assigned to a local maximum following an ascendent path along the density gradient [1]. Figure 8.10 presents an experiment were FH and Mean Shift are used, respectively, to cluster the dataset in Figure 8.9. Different runs were performed, by varying the kernel/scale size. Clearly, results are suboptimal in both cases. Both algorithms share the same main disadvantage: a global scale must be chosen a priori. Such a strategy is unable to cope with clusters of different densities and spatial sizes. Choosing a fine scale leads to a correct detection of dense clusters, at the price of oversplitting less denser ones. On the contrary, a coarser scale corrects the oversplitting of less denser clusters but undersplits the denser ones.

8.4 Handling MST Instability

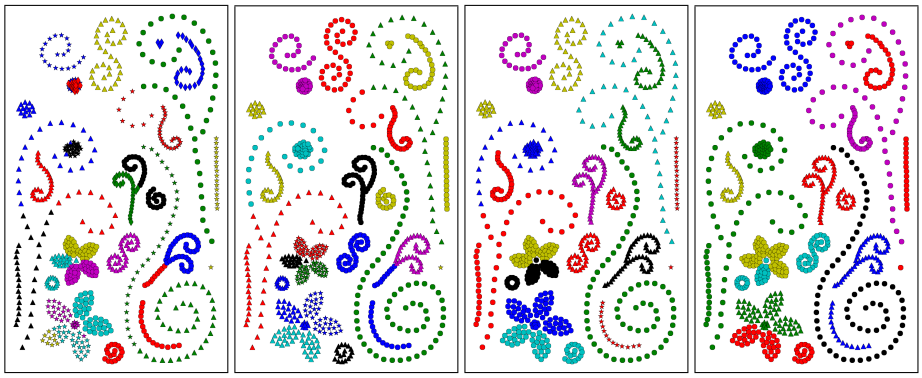
A seemingly obvious but interesting phenomenon occurs when noise is added to clustered data. Suppose we have data with two well separated clusters. In the absence of noise, it exists an MST edge linking both clusters. If noise is added to the data, the edge would probably disappear and be replaced by a sequence of edges. The length of the original linking edge is larger than the length of the

[1] http://www.mathworks.com/matlabcentral/fileexchange/10161-mean-shift-clustering

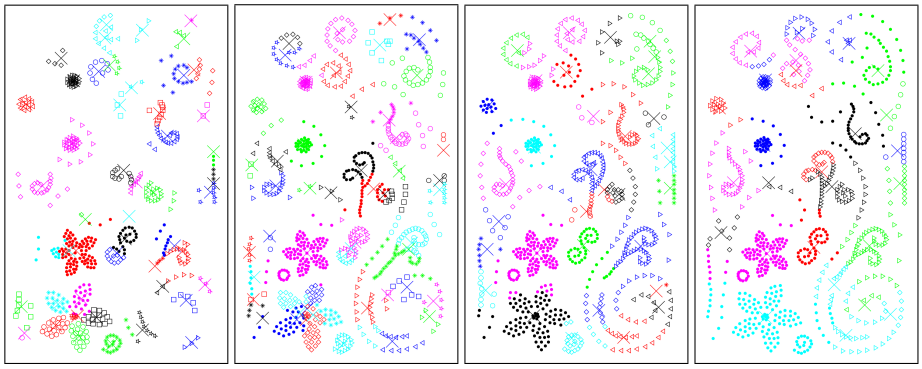


(a) Shapes drawn against noise. Shapes are respectively plotted in red and in black on the top and bottom rows.

Figure 8.9: In this example, the meaningful clustered forest correctly describes the organization of the points configuration. Only small or less denser figures are discarded. Indeed, meaningful components are clearly perceived in noise while non-meaningful components are not.



(a) Felzenszwalb and Huttenlocher' algorithm results at different scales.



(b) Mean Shift results for different values of kernel sizes.

Figure 8.10: The same points configuration as in Figure 8.9. At all scales, FH and Mean Shift fail to correctly detect the organization. Under and oversplitting occur in all cases.

endpoints of the sequence. The direct consequence is an increase in the NFA of the two clusters. Depending on the magnitude of that increase, the clusters might potentially be split into several proximal meaningful components. See Figure 8.13.

In short terms, noise affects the ideal topology of the MST. The oversplitting phenomenon can be corrected by iterating the following steps:

1. detecting the meaningful clustered forest,
2. add the union of points in the meaningful clustered forest to a new input dataset,
3. remove the points in the meaningful clustered forest and replace them with noise,
4. iterate until convergence,
5. re-detect the meaningful clustered forest on the new noise-free dataset built along all iterations.

The MST of the set formed by merging the meaningful clustered forests from all iterations has the right topology. In other words this MST resembles the MST of the

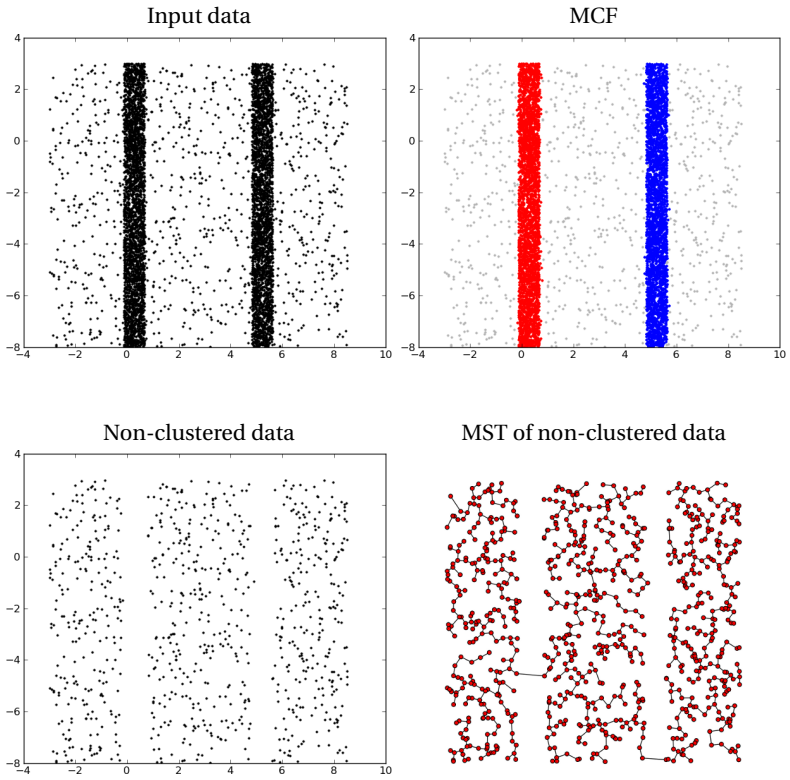


Figure 8.11: Removing PMC can generate artifacts, i.e. holes, in the remaining data. These holes might create edges in the MST of the non-clustered data, that certainly violate the background model.

original data without noise. Then, detection of meaningful clustered forest can be performed without major trouble. We say that these detections form a stabilized meaningful clustered forest.

The above method implicitly contains a key issue in step 3. Detected points must be replaced with others which have a completely different distribution (i.e. the background distribution) but must occupy the same space. Figure 8.11 contains an example of the need for such a strong requirement. Pieces of background data might become "disconnected, or to be precise connected by artificially created new edges. In one dimension, these holes are easily contracted, but when the dimensionality increases the contracting scheme gains more and more degrees of freedom.

This noise filling procedure can be achieved by using the Voronoi diagram [6] of the original point set. In the Voronoi diagram, each point lies on a different cell. To remove a point amounts to emptying a cell. Then the set of empty cells can be used as a sampling window to draw points from the background model. Notice that this procedure is actually generic since the Voronoi tesselation can be

generalized to higher dimensional metric spaces [6].

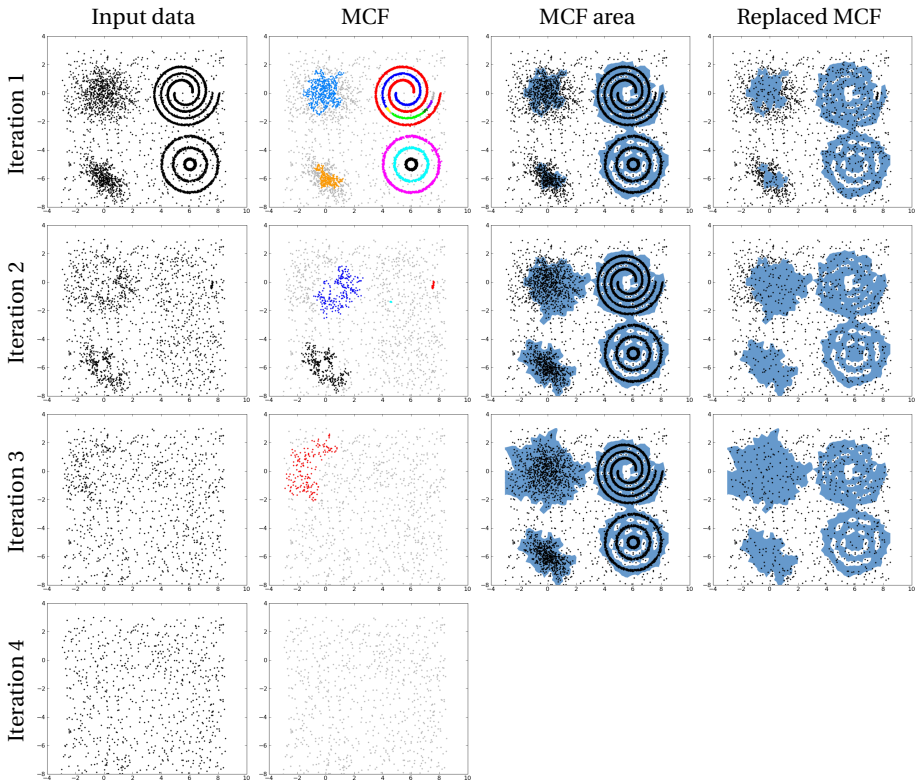
The process simulates replacing detected components with noise from the background model. Due to the same nature of the Voronoi diagram, the process is not perfect: in the final iteration, the resulting point set is quasi but not exactly distributed according to the background model. A small bias is introduced, causing a few spurious detections in the MCF. To correct this issue it suffices to set $\varepsilon = 10^{-2}$, as these detections have NFAs slightly lower than one and actual detections have really low NFAs. Of course this new thresholding could be avoided if a more accurate flling procedure was used.

Algorithm 13 illustrates steps 1 to 4 of the correcting method. An example is shown in Figure 8.12, where four iterations are required until convergence.

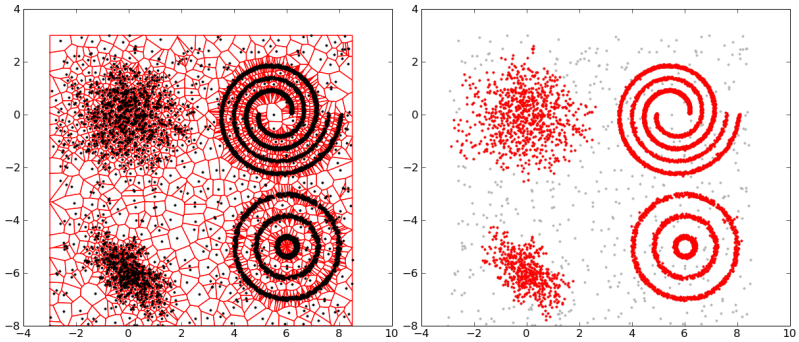
Algorithm 13 Stabilize point set X returning the set $\mathcal{F}$ of non-background points.

$\mathcal{F} \leftarrow \emptyset$
$\mathcal{V} \leftarrow$ cells from Voronoi diagram of point set X intersected with the minimum rectangle enclosing X.
$X' \leftarrow X$
$\mathcal{M} \leftarrow$ meaningful clustered forest of X'
while $\mathcal{M} \neq \emptyset$ **do**
  $\mathcal{V}' \leftarrow \emptyset$
  $X' \leftarrow \emptyset$
  for all $C \in \mathcal{M}$ **do**
    for all $\mathbf{p} \in C$ **do**
      add $V \in \mathcal{V}$ to $\mathcal{V}'$ such that $\mathbf{p} \in V$.
      if $\mathbf{p} \in X$ **then**
        add $\mathbf{p}$ to X'.
        add $\mathbf{p}$ to $\mathcal{F}$.
      end if
    end for
  end for
  $a \leftarrow \sum_{V \in \mathcal{V}} \text{area}(V)$
  $a_{\mathcal{M}} \leftarrow \sum_{V \in \mathcal{V}'} \text{area}(V)$
  $n_{\mathcal{M}} \leftarrow \sum_{C \in \mathcal{M}} |C|$
  $n \leftarrow a_{\mathcal{M}} \cdot (|X| - N_{\mathcal{M}}) / (a - a_{\mathcal{M}})$
  $B \leftarrow$ draw n points $\mathbf{q}_i$, $1 \leq i \leq n$, from the background model such that $(\exists V \in \mathcal{V}')\ q_i \in V$.
  $X' \leftarrow X' \cup B$
  $\mathcal{M} \leftarrow$ meaningful clustered forest of X'
end while

Figure 8.13 shows a second example of the stabilization process, followed by the detection of the stabilized meaningful clustered forest. The NFAs of the de-

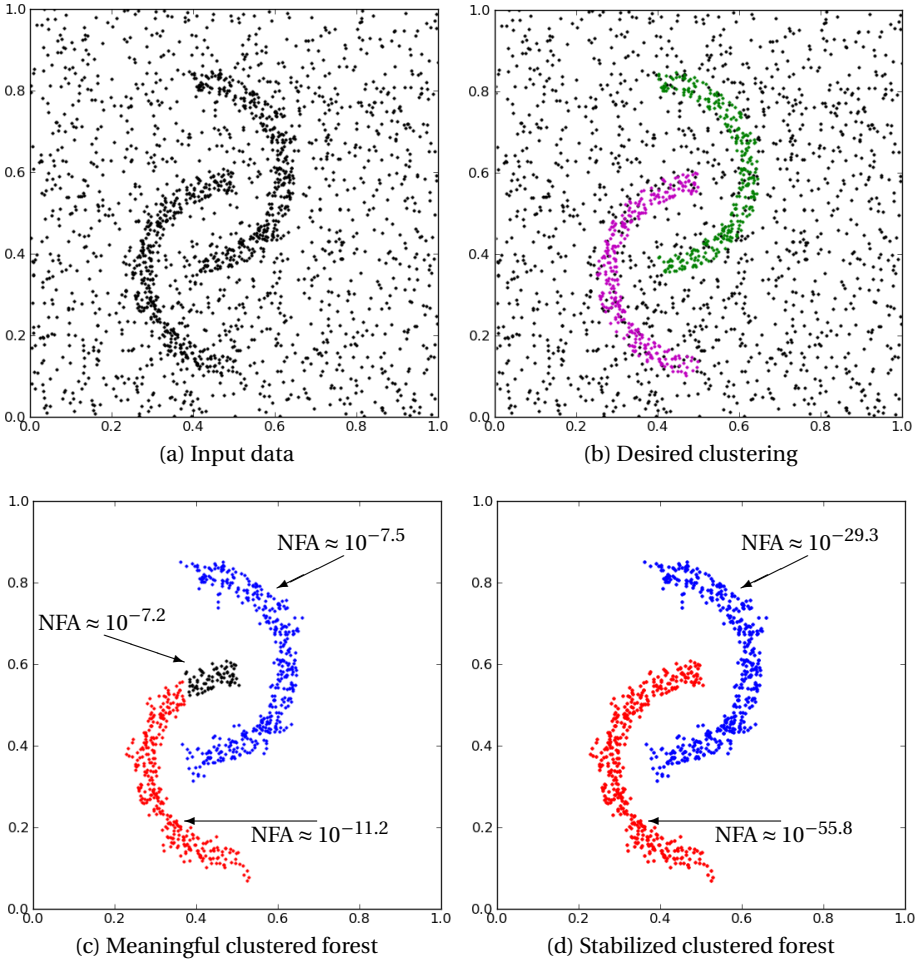


(a) In each iteration, the MCF is detected and the cells on the Voronoi diagram corresponding to points in the MCF are emptied and filled with points distributed according to the background model. In the fourth iteration, no MCF is detected and thus the algorithm stops.



(b) Left, original Voronoi diagram. Right, resulting non-background points in red.

Figure 8.12: Iteratively detecting the MCF and replacing it with points from the background model, converges and separates background from non-background data.



(a) Input data (b) Desired clustering

(c) Meaningful clustered forest (d) Stabilized clustered forest

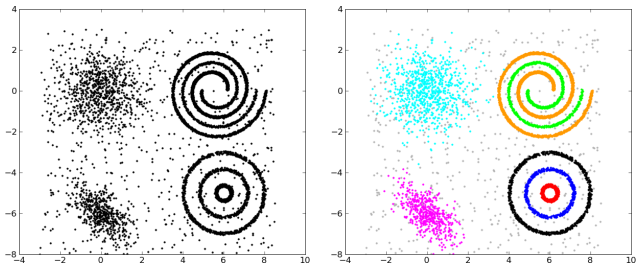
Figure 8.13: Noise might affect the stability of the meaningful clustered forest, causing to oversplit the clusters. Algorithm 13 converges in two iterations. Then, one can detect the meaningful clustered forest among non-background points, yielding a stabilized meaningful clustered forest.

tected components are also included. The very low attained NFAs, account for the success of the procedure.

8.5 The Masking Challenge

In 2009, Jain [67] stated that no available algorithm could cluster the dataset in Figure 8.14a and obtain the result in Figure 8.14b. The dataset is interesting because it brings to light a new phenomenon: a cluster with many points can "dominate" the scene and hide other clusters that could be meaningful.

A similar behavior occurs in vanishing point detection, as pointed out by Al-



(a) Input data (b) Desired clustering

Figure 8.14: According to Jain [67] no available algorithm can correctly cluster this dataset.

mansa et al. [1]. A vanishing point is a point in an image to which parallel line segments not frontoparallel appear to converge; in some sense one can regard this point as a collection of such parallel line segments. Sometimes this procedure will still miss some weak vanishing points which are "masked" by stronger vanishing points composed of much more segments. These may not be perceived at first sight, but only if we manage to unmask them by getting rid of the "clutter" in one way or another. Almansa et al. propose to eliminate these detected vanishing points and look for new vanishing points among the remaining line segments.

In our case, this very same approach cannot be followed. Masked clusters are not completely undetected but partially detected. Removing such cluster parts and re-detecting would cause oversegmentation. We propose instead to only remove the most meaningful proximal component and then iterate. The process ends when the masking phenomenon disappears, that is:

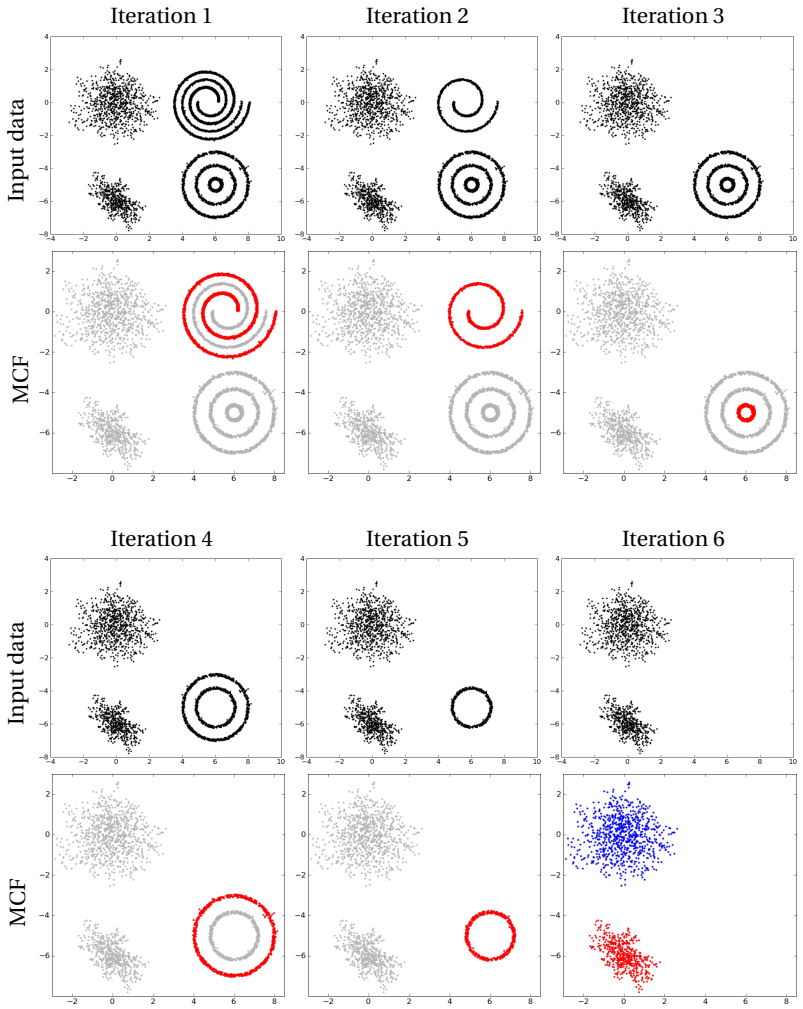
- when there are no unclustered points, or
- no MCF is detected.

Algorithm 14 shows a detail of this unmasking scheme. Summarizing, first non-background points are detected using the stabilization process in Algorithm 13 and then the unmasking process takes place.

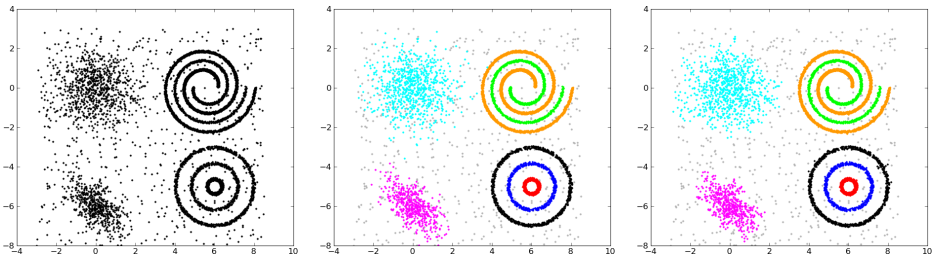
From a total number of 7000 points in Figure 8.14a, the outer spiral (in orange in in Figure 8.14b) has 2514 points, i.e. almost 36% of the points. The detection of the unmasked MCF in Figure 8.15d correct all masking issues. Moreover, they are extremely similar to the desired clustering in Figure 8.15c. The difference is that clusters absorb background points that are within or near them. Indeed, these background points are statistically indistinguishable from the points from the cluster that absorbs them.

8.6 Three-dimensional point clouds

We tested the proposed algorithm with three-dimensional point clouds. We put two point clouds in the same scene at different positions, thus building two scenes



(a) In each iteration, the MCF is detected and the most meaningful component is removed from the dataset. In the sixth iteration, all points are clustered and thus the algorithm stops.



(b) Input data (c) Desired clustering (d) Unmasked MCF.

Figure 8.15: Iteratively detecting the MCF and removing from the dataset its most meaningful component, converges and corrects the masking phenomenon. The detected MCF is extremely similar to the desired clustering. The difference is that clusters absorb background points that are within of near them.

Algorithm 14 Compute the unmasked meaningful clustered forest $\mathcal{U}$ from the point set $\mathcal{F}$ of non-background points.

$\mathcal{U} \leftarrow \emptyset$
while $\mathcal{M} \neq \emptyset$ **do**
  $\mathcal{M} \leftarrow$ stabilized meaningful clustered forest of $\mathcal{F}$
  if $|F| = \sum_{C \in \mathcal{M}} |C|$ **then**
    $\forall C \in \mathcal{M}$, add C to $\mathcal{U}$.
  else
    $C_{\min} \leftarrow \arg\min_{C \in \mathcal{M}} \mathrm{NFA}(C)$
    for all p $\in C_{\min}$ **do**
      remove p from $\mathcal{F}$
    end for
    add $C_{\min}$ to $\mathcal{U}$.
  end if
end while

in Figures 8.16 and 8.17. In both cases uniformly distributed noise was artificially added. The skeleton hand and the bunny are formed by 3274 and by 3595, respectively. In Figure 8.16, 3031 noise points were added to total 9900 points. In Figure 8.16, 7031 noise points were added to total 13900 points and both shapes were positioned closer to each other and in such a way that no linear separation exist between them. In both cases the result is correct

In Figure 8.16, the MCF is oversplit but the stabilization process discussed in Section 8.4 corrects the issue. In Figure 8.16, although the same phenomenon is possible, it does not occur in this realization of the noise process.

8.7 Final Remarks

In this chapter we propose a new clustering method that can be regarded as a numerical method to compute the proximity gestalt. The method relies on analyzing edge distances in the MST of the dataset. The direct consequence is that our approach is fully parametric on the chosen distance.

The proposed method present several novelties over other MST-based formulations. Some formulations have preference for compact clusters as they extract their clustering detection rule from characteristics that are not intrinsic to the MST. Our method only focuses on the length of the MST edges; hence, it does not present such preference. Other formulations analyze the data at a fixed local scale, thus introducing a new method parameter. We have shown through examples that these local methods can fail when the input data has clusters with different sizes and densities. In these same examples, our method perform well without the need of introducing any extra parameter.

The method is also automatic, in the sense that only a single parameter is left

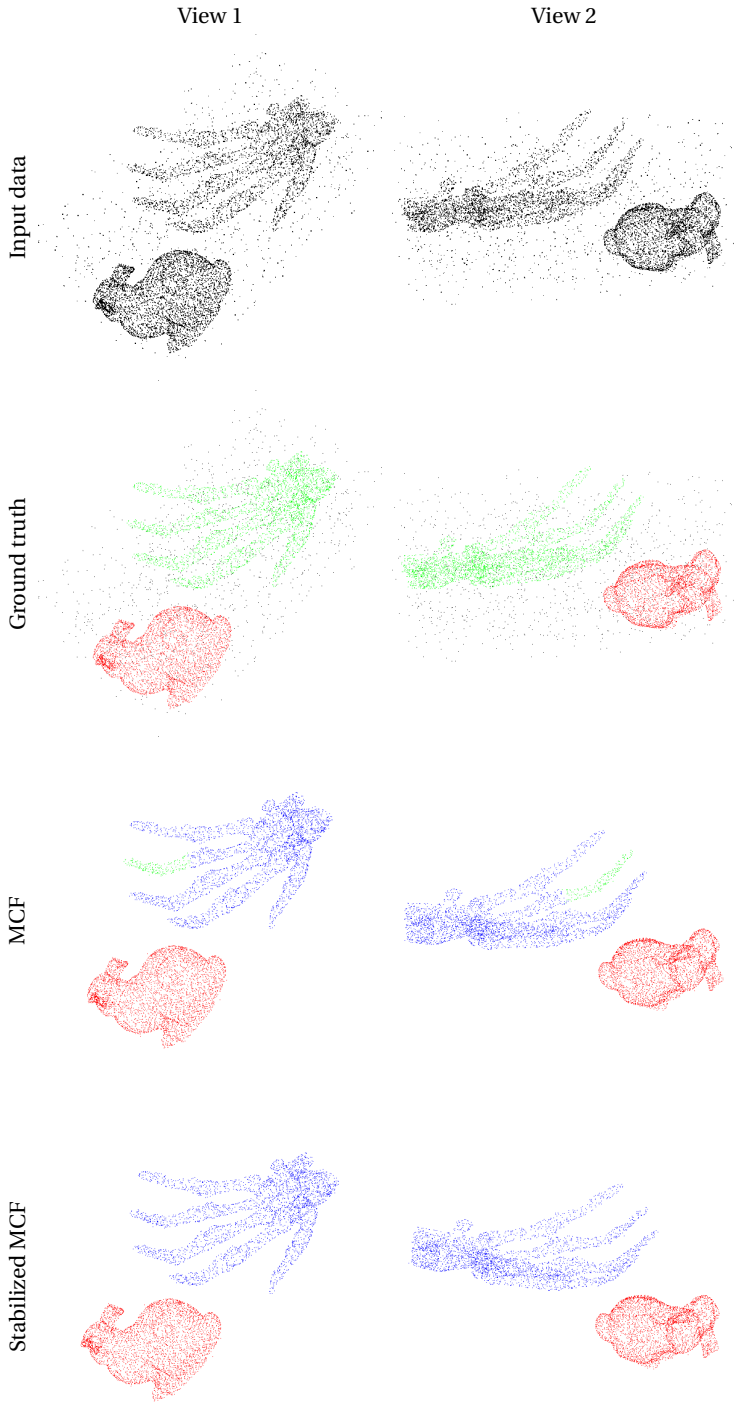


Figure 8.16: Two point clouds with artificially added noise. In this case, noise perturbed the MCF (see the finger of the skeleton hand). This effect is corrected by the stabilization process.

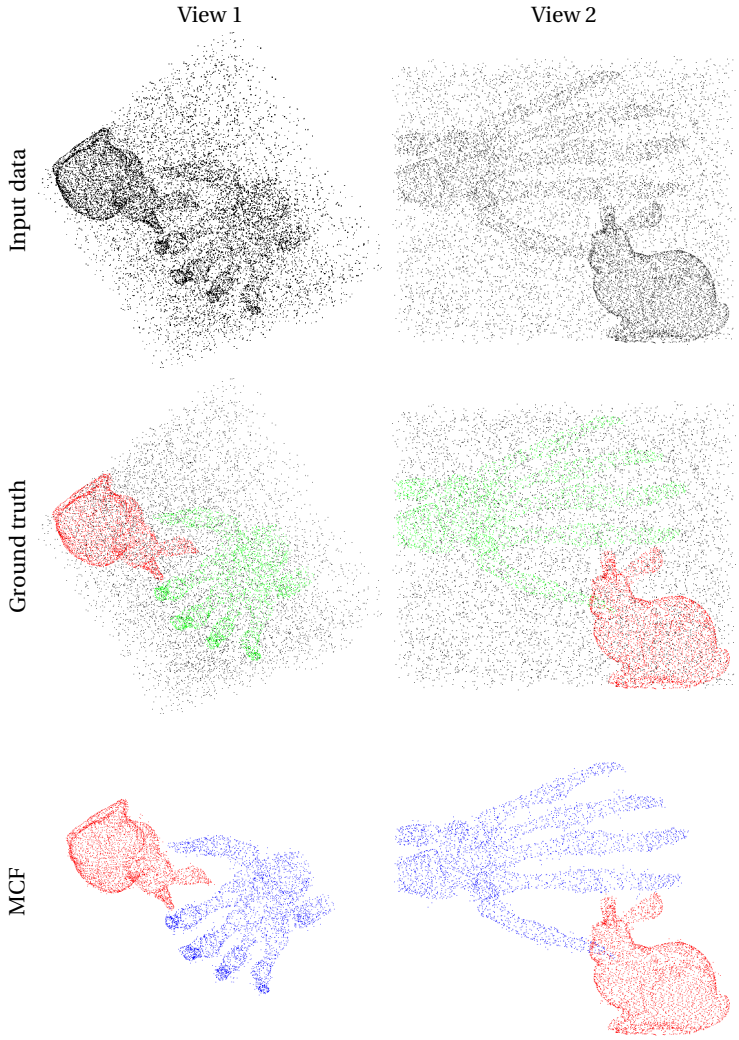


Figure 8.17: Two point clouds with artificially added noise. Both shapes are close to each other and are not linearly separable. The result of the stabilization process is omitted as detections do not change.

to the user. This parameter has an intuitive interpretation as it controls the expected number of false detections. Moreover, setting it to 1 is sufficient in practice.

Robustness to noise is an additional but essential feature of the method. Indeed, we have shown that the iterative application of our method can be used to treat noisy data, producing quality results.

We also studied the masking phenomenon in which a highly populated and salient cluster dominates the scene and inhibits the detection of less-populated, but still salient, clusters. The proposed method can be iteratively used to avoid such inhibitions from happening, yielding promising results.

As future work, it would be interesting to study the MST edge distribution under different point distributions. From the theoretical point of view, it can bring light to the method correctness. In practice, it would allow to replace the simulated background models by their analytical counterparts.

CHAPTER

9 Conclusions

In this thesis we studied two shape representations, namely planar shapes and clusters. With the former we refer to shapes that correspond to object boundaries detected in an image. The latter tackles point clouds that may represent object surfaces, manifolds, etc.

In the first part of this work, we address the detection and recognition of planar shapes and propose different techniques to improve and complement the shape recognition framework presented in [22].

The second part was devoted to the study of clustering techniques with the objective of detecting arbitrarily shaped clusters using proximity. Several algorithms are proposed that focus on the detection process itself and on the algorithmic performance of the process.

9.1 Main contributions

Let us now summarize the main contributions of this thesis to planar shapes:

- We have extended the classical a contrario method for detecting salient level lines. The extension involves allowing that a level line must not be entirely salient to be detected. In particular, if contrast is used as a measure of saliency, it suffices that only some parts are contrasted to detect the level line as a meaningful one. This extension have two benefits: (1) from a conceptual point of view, it is in accordance that pieces of level lines correspond to object boundaries; (2) from a practical angle, the threshold between contrasted and non-contrasted parts is found automatically and will prove useful in the following processing stage.
- We have explored the removal of the non-salient (i.e. non-contrasted) parts of a meaningful level line. The previous method to perform this clean-up relied on an a priori estimation of the length of level lines in noise images. We followed a different approach, which does not involve any a priori information, by detecting periodic binary subsequences. The method is based on

an algorithm to detect binary subsequences that was extended for handling periodic sequences (remember that level lines are closed Jordan curves).

- We proposed a method to use two gestalts, contrast and good continuation, as a combined saliency measure to detect level lines. In this approach, both measures compete for the "control" of the level line: the least salient of both dominates. The resulting effect is a reinforcement of the detections since only contrasted and smooth boundaries are detected.
- We adapted the shape context descriptor to work with actual shapes, instead of edge maps, thus converting a global into a semi-local descriptor. The semi-locality is completely natural, not depending on any user-defined parameter. We finally applied the a contrario shape matching framework for the problem of matching shape contexts.

The main contributions of this thesis to clustering can be summarized as follows:

- We proposed an a contrario clustering method that permits to validate individual clusters in a hierarchical structure. The validation is achieved by using graphs to perform nonparametric density estimation. In contrast to previous methods, datasets with any number of dimensions can be easily handled and the shape of clusters is not imposed a priori. As an example, we have successfully applied this method to detect clusters in the Normalized Cuts framework.
- A method for efficiently computing the Minimum Spanning Tree (MST) is also introduced. The method avoids computing the complete set of interpoint distances by a clever use of nearest neighbors search structures. We have shown that the method is very efficient for large and low-dimensional datasets. For high dimensions, an approximate MST can be computed which has been proven very stable.
- We finally presented a second a contrario clustering algorithm. Although it is also a validation scheme for clusters in a hierarchical structure, it is based on different principles than the aforementioned one. It is specifically designed to work with the MST and its hierarchical version, i.e. the single-link algorithm, by computing edge length statistics. It is thus capable of detecting arbitrarily shaped clusters very efficiently. We address the problem of masking in two forms. The first manifestation is when noise avoids from correctly detecting clusters. The second manifestation occurs when a highly populated cluster avoids from detecting other less populated, but still meaningful, clusters. We show that by iteratively applying the proposed clustering algorithm, both phenomena can be unveiled.

9.2 Future work

The experimental results presented in this thesis are satisfactory and promising, both on the planar shape and on the clustering side. However many problems remain unsolved or their solution has to be improved.

In Chapter 4 we have already discussed in detail possible extensions to our work on planar shapes. The most important ones are:

- Explore other distances between shapes that provide better technical results (e.g. the circular EMD [116]) or that are inspired by psychophysical results on human perception.
- Simulation of the affine parameters by using a multiscale analysis combined with the ASIFT simulation procedure [99].
- Obtain a decisive answer to whether groups of shapes do match or not. Clustering techniques offer an interesting way to address this problem [21].

On the clustering side, the main lines of improvement and future development are:

- Provide a fully parallelized implementation of the algorithm to compute MSTs. Such implementation should act on two different levels. First, the algorithm itself can be run in parallel by using multiple processors. Second, the nearest neighbors algorithms can be implemented for GPUs, thus greatly accelerating the search process.
- The clustering algorithm presented in Chapter 8, uses edge length statistics of the MST. Human perception was the inspiration to use the MST. However, for applications where perceptually-inspired methods are not essential, it is also possible to imagine a similar approach by using different hierarchical algorithms. The most straightforward one is the use of the complete-link hierarchy which would lead to a fast algorithm to detect compact clusters.
- So far, we have only used proximity as a measure for clustering while in many applications other measures might improve the results. Specially in the case of 3D point clouds, good continuation (i.e. smoothness) is a leading key to correctly recover many shapes.
- Finally, we will mention the use of the proposed a contrario clustering techniques for ensemble clustering. The methods can be used in this framework in two different ways. First, results might be greatly improved by perturbing the original dataset with a given noise, clustering each of them and finally obtaining a more robust solution by combining all the individual solutions. Last but not least, different clustering results, obtained with our own algorithms or with others, can be combined by using the proposed clustering techniques. Single-link algorithms have long been used for clustering by combining results from other algorithms (which may be adapted to handling non-numerical data, for example). The application of the presented techniques in this scenario is indeed straightforward.

Bibliography

[1] A. Almansa, A. Desolneux, and S. Vamech. Vanishing point detection without any a priori information. *Transactions on Pattern Analysis and Machine Intelligence*, 25(4):502–507, 2003.

[2] A. Andoni and P. Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. *Communications of the ACM*, 51(1):117–122, January 2008.

[3] Rudolf Arnheim. *Visual Thinking*. University of California Press, April 1969.

[4] K. Astrom. Fundamental limitations on projective invariants of planar curves. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 17(1):77–81, 1995.

[5] F. Attneave. Some informational aspects of visual perception. *Psychological Review*, 61(3):183–193, May 1954.

[6] F. Aurenhammer. Voronoi diagrams - a survey of a fundamental geometric data structure. *ACM Computing Surveys*, 23(3):345–405, September 1991.

[7] S. Bandyopadhyay. An automatic shape independent clustering technique. *Pattern Recognition*, 37(1):33–45, January 2004.

[8] E. Barenholtz, E. Cohen, J. Feldman, and M. Singh. Detection of change in shape: An advantage for concavities. *International Journal of Cognitive Science*, 89(1):1–9, August 2003.

[9] E. Barenholtz and J. Feldman. Visual comparisons within and between object parts: evidence for a single-part superiority effect. *Vision Research*, 43(15):1655–1666, July 2003.

[10] A. Barron, J. Rissanen, and B. Yu. The minimum description length principle in coding and modeling. *IEEE Transactions on Information Theory*, 44(6):2743–2760, 1998.

[11] Z. Barzily, Z. Volkovich, B. Akteke Öztürk, and G. W. Weber. On a Minimal Spanning Tree Approach in the Cluster Validation Problem. *Informatica*, 20(2):187–202, 2009.

[12] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522, 2002.

[13] J. Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517, September 1975.

[14] J. Bentley and J. Friedman. Fast Algorithms for Constructing Minimal Spanning Trees in Coordinate Spaces. *IEEE Transactions on Computers*, 27(2):97–105, 1978.

[15] C. Bishop. *Pattern Recognition and Machine Learning*. Springer, 1st ed. 2006. corr. 2nd printing edition, October 2007.

[16] A. Blake and M. Isard. *Active Contours*. Springer-Verlag, 1998.

[17] T. Bozkaya and M. Ozsoyoglu. Distance-based indexing for high-dimensional metric spaces. In *Proceedings of the 1997 ACM SIGMOD international conference on Management of Data*, volume 26, pages 357–368, New York, NY, USA, June 1997. ACM.

[18] K. Burnham and D. Anderson. *Model selection and multimodel inference: a practical information-theoretic approach*. Springer, 2nd edition, July 2002.

[19] N. Burrus, T. M. Bernard, and J. M. Jolion. Image segmentation by a contrario simulation. *Pattern Recognition*, 42(7):1520–1532, 2009.

[20] J. Canny. A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, November 1986.

[21] F. Cao, J. Delon, A. Desolneux, P. Musé, and F. Sur. A Unified Framework for Detecting Groups and Application to Shape Recognition. *Journal of Mathematical Imaging and Vision*, 27(2):91–119, February 2007.

[22] F. Cao, J. L. Lisani, J. M. Morel, P. Musé, and F. Sur. *A Theory of Shape Identification*, volume 1948 of *Lecture Notes in Mathematics*. Springer, 2008.

[23] F. Cao, P. Musé, and F. Sur. Extracting Meaningful Curves from Images. *J. Math. Imaging Vis.*, 22(2-3):159–181, 2005.

[24] G. Carlsson and F. Mémoli. Characterization, Stability and Convergence of Hierarchical Clustering methods. *Journal of Machine Learning Research*, 11:1425–1470, 2010.

[25] G. Carlsson and F. Mémoli. Characterization, Stability and Convergence of Hierarchical Clustering methods. *Journal of Machine Learning Research*, 11:1425–1470, 2010.

[26] M. Carreira-Perpiñán. Gaussian mean shift is an EM algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29:2007, 2005.

[27] M. Carreira-Perpiñán and R. Zemel. Proximity graphs for clustering and manifold learning. In *In Advances in Neural Information Processing Systems*, volume 17, pages 225–232, 2005.

[28] V. Caselles and J. M. Morel. Topographic Maps and Local Contrast Changes in Natural Images. *International Journal of Computer Vision*, 33:5–27, 1999.

[29] P. Cavanagh. The artist as neuroscientist. *Nature*, 434(7031):301–307, March 2005.

[30] E. Chavez and G. Navarro. An Effective Clustering Algorithm to Index High Dimensional Metric Spaces. In *Proceedings of the Seventh International Symposium on String Processing Information Retrieval*, pages 75+, Washington, DC, USA, 2000. IEEE Computer Society.

[31] E. Chávez and G. Navarro. A compact space decomposition for effective metric indexing. *Pattern Recognition Letters*, 26(9):1363–1376, 2005.

[32] B. Chazelle. A minimum spanning tree algorithm with inverse-Ackermann type complexity. *Journal of the ACM*, 47(6):1028–1047, November 2000.

[33] F. Chung. *Spectral Graph Theory (CBMS Regional Conference Series in Mathematics, No. 92)*. American Mathematical Society, February 1997.

[34] P. Ciaccia and M. Patella. Bulk Loading the M-tree. In *In Proceedings of the 9th Australasian Database Conference*, pages 15–26, 1998.

[35] D. Comaniciu and P. Meer. Mean shift: a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, August 2002.

[36] T. Cormen, C. Leiserson, R. Rivest, and C. Stein. *Introduction to Algorithms*. McGraw-Hill Science / Engineering / Math, 2nd edition, December 2003.

[37] T. Cour, F. Benezit, and J. Shi. Spectral Segmentation with Multiscale Graph Decomposition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 1124–1131, Washington, DC, USA, 2005. IEEE Computer Society.

[38] E. Cura, M. Tepper, and M. Mejail. Content-Based Emblem Retrieval Using Zernike Moments. In *Iberoamerican Congress on Pattern Recognition, CIARP 2010*, pages 79–86, Sao Paulo, Brazil, November 2010. IAPR.

[39] A. Desolneux, L. Moisan, and J. M. Morel. Edge Detection by Helmholtz Principle. *Journal of Mathematical Imaging and Vision*, 14(3):271–284, 2001.

[40] A. Desolneux, L. Moisan, and J. M. Morel. *From Gestalt Theory to Image Analysis*, volume 34. Springer-Verlag, 2008.

[41] I. Dhillon, Y. Guan, and B. Kulis. Weighted Graph Cuts without Eigenvectors A Multilevel Approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(11):1944–1957, 2007.

[42] M. Donoser and H. Bischof. Efficient Maximally Stable Extremal Region (MSER) Tracking. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1 (CVPR'06)*, volume 1, pages 553–560. IEEE, 2006.

[43] P. Drineas, A. Frieze, R. Kannan, S. Vempala, and V. Vinay. Clustering Large Graphs via the Singular Value Decomposition. In *Machine Learning*, volume 56, pages 9–33, 2004.

[44] M. Dry, D. Navarro, K. Preiss, and M. Lee. The Perceptual Organization of Point Constellations. In *Annual Meeting of the Cognitive Science Society*, 2009.

[45] R. Duda, P. Hart, and D. Stork. *Pattern Classification (2nd Edition)*. Wiley-Interscience, 2 edition, November 2001.

[46] W. Eddy, A. Mockus, and S. Oue. Approximate single linkage cluster analysis of large data sets in high-dimensional spaces. *Computational Statistics & Data Analysis*, 23(1):29–43, 1996.

[47] B. Epstein. Some Applications of the Mellin Transform in Statistics. *The Annals of Mathematical Statistics*, 19(3):370–379, September 1948.

[48] P. Etyngier. *Statistical learning, Shape Manifolds & Applications to Image Segmentation*. PhD thesis, École Nationale des Ponts et Chaussées, April 2008.

[49] J. Feldman and M. Singh. Information along Contours and Object Boundaries. *Psychological Review*, 112(1):243–252, 2005.

[50] P. Felzenszwalb and D. Huttenlocher. Efficient Graph-Based Image Segmentation. *International Journal of Computer Vision*, 59(2):167–181, September 2004.

[51] M. Fischler and R. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, June 1981.

[52] G. Flake, R. Tarjan, and K. Tsioutsiouliklis. Graph Clustering and Minimum Cut Trees. *Internet Mathematics*, 1(4):385–408, 2004.

[53] P. E. Forssén and D. Lowe. Shape Descriptors for Maximally Stable Extremal Regions. In *IEEE International Conference on Computer Vision*, volume CFP07198-CDR, Rio de Janeiro, Brazil, October 2007. IEEE Computer Society.

[54] C. Fowlkes, S. Belongie, F. Chung, and J. Malik. Spectral Grouping Using the Nyström Method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):214–225, 2004.

[55] C. Fowlkes and J. Malik. How much does Globalization help Segmentation? Technical report, 2004.

[56] A. Frank and A. Asuncion. UCI machine learning repository, 2010.

[57] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, second edition, 1999.

[58] K. Fukunaga and L. Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on Information Theory*, 21(1):32–40, January 1975.

[59] F. Gómez Fernández. Líneas de Nivel y Shape Context. Master's thesis, Departamento de Computación, Facultad de ciencias Exactas y Naturales, Universidad de Buenos Aires, March 2010.

[60] R. Graham and P. Hell. On the history of the minimum spanning tree problem. *Annals Of The History Of Computing*, 7(1):43–57, 1985.

[61] Grompone, Jérémie Jakubowicz, Jean-Michel Morel, and Gregory Randall. On Straight Line Segment Detection. *Journal of Mathematical Imaging and Vision*, 32(3):313–347, November 2008.

[62] R. Grompone von Gioi, J. Jakubowicz, J. M. Morel, and G. Randall. LSD: A Fast Line Segment Detector with a False Detection Control. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(4):722–732, April 2010.

[63] F. Guichard, J. M. Morel, and R. Ryan. Contrast invariant image analysis and PDE's. http://www.cmla.ens-cachan.fr/Membres/morel.html.

[64] B. Hendrickson and R. Leland. A multilevel algorithm for partitioning graphs. In *Proceedings of the 1995 ACM/IEEE Conference on Supercomputing (CDROM)*, Supercomputing '95, New York, NY, USA, 1995. ACM.

[65] Richard Hoffman and Anil K. Jain. A test of randomness based on the minimal spanning tree. *Pattern Recognition Letters*, 1(3):175–180, March 1983.

[66] L. Hubert and P. Arabie. Comparing partitions. *Journal of Classification*, 2(1):193–218–218, December 1985.

[67] A. K. Jain. Data clustering: 50 years beyond K-means. *Pattern Recognition Letters*, 31(8):651–666, September 2010.

[68] A. K. Jain and R. C. Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1988.

[69] A. K. Jain, Xiaowei Xu, Tin K. Ho, and Fan Xiao. Uniformity testing using minimal spanning tree. In *International Conference on Pattern Recognition*, pages 281–284. IEEE Computer Society, 2002.

[70] G. Kanizsa. *Organization in Vision: Essays on Gestalt Perception*. Praeger, 1979.

[71] R. Kannan, S. Vempala, and A. Vetta. On clusterings: Good, bad and spectral. *Journal of the ACM*, 51(3):497–515, May 2004.

[72] D. Karger, P. Klein, and R. Tarjan. A Randomized Linear-Time Algorithm to Find Minimum Spanning Trees. *Journal of the ACM*, 42(2):321–328, 1995.

[73] G. Karypis and V. Kumar. A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs. *SIAM Journal on Scientific Computing*, 20(1):359–392, 1999.

[74] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, January 1988.

[75] A. Khotanzad and Y. H. Hong. Invariant image recognition by Zernike moments. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 12(5):489–497, 1990.

[76] R. Kimmel and A. Bruckstein. Regularized Laplacian Zero Crossings as Optimal Edge Integrators. *International Journal of Computer Vision*, 53:225–243, July 2003.

[77] J. Kleinberg. An impossibility theorem for clustering. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems*, pages 446–453. MIT Press, 2002.

[78] H. Krim and A. Yezzi, editors. *Statistics and Analysis of Shapes*. Modeling and Simulation in Science, Engineering and Technology. Birkhäuser Boston, May 2006.

[79] C. Lai, T. Rafa, and D. Nelson. Approximate minimum spanning tree clustering in high-dimensional space. *Intelligent Data Analysis*, 13(4):575–597, 2009.

[80] L. Latecki, R. Lakamper, and U. Eckhardt. Shape descriptors for non-rigid shapes with a single closed contour. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, volume 2000, pages 424–429, 2000.

[81] B. Leibe, K. Mikolajczyk, and B. Schiele. Efficient Clustering and Matching for Object Class Recognition. In *Proceedings of British Machine Vision Conference*, 2006.

[82] S. Li, M. C. Lee, and C. M. Pun. Complex Zernike Moments Features for Shape-Based Image Retrieval. 39(1):227–237, 2009.

[83] T. Lindeberg. *Scale-Space Theory in Computer Vision*, volume 256 of *The Springer International Series in Engineering and Computer Science*. Kluwer Academic Publishers, Norwell, MA, USA, 1994.

[84] H. Ling and D. Jacobs. Shape classification using the inner-distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29:286–299, 2007.

[85] J. L. Lisani, Moisan, J. M. Morel, and P. Monasse. On the theory of planar shape, 2002.

[86] D. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110, November 2004.

[87] D. Marr and E. Hildreth. Theory of Edge Detection. *Proceedings of the Royal Society of London. Series B, Biological Sciences*, 207(1167):187–217, 1980.

[88] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings of the International Conference on Computer Vision*, volume 2, pages 416–423. IEEE Computer Society, 2001.

[89] J. Matas, O. Chum, U. Martin, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *Proceedings of the British Machine Vision Conference*, volume 1, pages 384–393, London, 2002.

[90] G. Matheron. *Random Sets and Integral Geometry*. John Wiley & Sons, NY, USA, 1975.

[91] E. Meinhardt, E. Zacur, A. Frangi, and V. Caselles. 3D Edge Detection by Selection of Level Surface Patches. *Journal of Mathematical Imaging and Vision*, October 2008.

[92] F. Mémoli and G. Sapiro. Computing with point cloud data. In *Statistics and analysis of shapes*, Model. Simul. Sci. Eng. Technol., pages 201–229. Birkhäuser Boston, Boston, MA, 2006.

[93] K. Mikolajczyk, A. Zisserman, and C. Schmid. Shape Recognition with Edge-Based Features. In *Proceedings of the British Machine Vision Conference*, volume 2, pages 779–788, 2003.

[94] L. Moisan. Affine plane curve evolution: a fully consistent scheme. *Image Processing, IEEE Transactions on*, 7(3):411–420, 1998.

[95] F. Mokhtarian and S. Abbasi. Retrieval of Similar Shapes under Affine Transform. page 657. 1999.

[96] F. Mokhtarian, S. Abbasi, and J. Kittler. Efficient and robust retrieval by shape content through curvature scale space. In *International Workshop on Image Databases and MultiMedia Search*, pages 35–42, 1996.

[97] P. Monasse. *Morphological representation of digital images and application to registration*. PhD thesis, Université Paris IX-Dauphine, Paris, France, June 2000.

[98] P. Monasse and F. Guichard. Fast Computation of a Contrast Invariant Image Representation. *IEEE Transactions on Image Processing*, 9(5):860–872, May 2000.

[99] J. M. Morel and G. Yu. ASIFT: A New Framework for Fully Affine Invariant Image Comparison. *SIAM Journal on Imaging Sciences*, 2(2):438–469, 2009.

[100] G. Mori, S. Belongie, and J. Malik. Efficient shape matching using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(11):1832–1837, 2005.

[101] G. Mori and J. Malik. Recognizing objects in adversarial clutter: breaking a visual CAPTCHA. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages I–134–I–141. IEEE Computer Society, 2003.

[102] M. Muja and D. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *In VISAPP International Conference on Computer Vision Theory and Applications*, pages 331–340, 2009.

[103] E. Murphy-Chutorian and M. Trivedi. N-tree disjoint-set forests for maximally stable extremal regions. In *British Machine Vision Conference*, volume II, pages 739–748, September 2006.

[104] F. Murtagh. A Survey of Recent Advances in Hierarchical Clustering Algorithms. *The Computer Journal*, 26(4):354–359, November 1983.

[105] P. Musé. *On the definition and recognition of planar shapes in digital images*. PhD thesis, École Normal Supérieure de Cachan, October 2004.

[106] P. Musé, F. Sur, F. Cao, Y. Gousseau, and J. M. Morel. An A Contrario Decision Method for Shape Element Recognition. *International Journal of Computer Vision*, 69(3):295–315, September 2006.

[107] B. Nadler and M. Galun. Fundamental Limitations of Spectral Clustering. In *Advances in Neural Information Processing Systems*, volume 19, pages 1017–1024. MIT Press, 2007.

[108] L. Najman and M. Couprie. Building the Component Tree in Quasi-Linear Time. *Image Processing, IEEE Transactions on*, 15(11):3531–3539, 2006.

[109] S. Nene, S. Nayar, and H. Murase. Columbia Object Image Library (COIL-100). Technical report, Columbia University, 1996.

[110] A. Ng, M. Jordan, and Y. Weiss. On Spectral Clustering: Analysis and an algorithm. In T. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems*, pages 849–856. MIT Press, 2001.

[111] D. Nister and H. Stewenius. Scalable Recognition with a Vocabulary Tree. In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2161–2168, Washington, DC, USA, 2006. IEEE Computer Society.

[112] S. Obdrzálek and J. Matas. Object Recognition using Local Affine Frames on Distinguished Regions. In Paul L. Rosin, David A. Marshall, Paul L. Rosin, and David A. Marshall, editors, *British Machine Vision Conference 2002*. British Machine Vision Association, 2002.

[113] U. Ozertem, D. Erdogmus, and R. Jenssen. Mean shift spectral clustering. *Pattern Recognition*, 41(6):1924–1938, June 2008.

[114] S. Pettie and V. Ramachandran. An optimal minimum spanning tree algorithm. *Journal of the ACM*, 49(1):16–34, January 2002.

[115] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. *Numerical Recipes in C*. Cambridge University Press, Cambridge, UK, 2nd edition, 1992.

[116] J. Rabin, J. Delon, and Y. Gousseau. A Statistical Approach to the Matching of Local Features. *SIAM Journal on Imaging Sciences*, 2(3):931–958, 2009.

[117] W. Rand. Objective Criteria for the Evaluation of Clustering Methods. *Journal of the American Statistical Association*, 66(336):846–850, 1971.

[118] J. Revaud, G. Lavoue, and A. Baskurt. Improving Zernike Moments Comparison for Optimal Similarity and Rotation Angle Retrieval. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(4):627–636, 2009.

[119] B. D. Ripley. Modelling Spatial Patterns. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(2):172–212, 1977.

[120] H. Samet. Depth-First K-Nearest Neighbor Finding Using the MaxNearestDist Estimator. *International Conference on Image Analysis and Processing*, 0:486+, 2003.

[121] J. Serra. *Image Analysis and Mathematical Morphology*. Academic Press, Inc, Orlando, FL, USA, 1983.

[122] D. Shen, H. Ip, and Khwang. Affine invariant detection of perceptually parallel 3D planar curves. *Pattern Recognition*, 33(11):1909–1918, November 2000.

[123] D. Shen, W. Wong, and H. Ip. Affine-Invariant Image Retrieval By Correspondence Matching of Shapes, May 1999.

[124] J. Shi and J. Malik. Normalized Cuts and Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[125] C. Silpa-Anan and R. Hartley. Optimised KD-trees for fast image descriptor matching. volume 0, pages 1–8, Los Alamitos, CA, USA, 2008. IEEE Computer Society.

[126] J. Sivic and A. Zisserman. Video Google: A Text Retrieval Approach to Object Matching in Videos. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, volume 2, pages 1470–1477 vol.2, Los Alamitos, CA, USA, April 2003. IEEE Computer Society.

[127] T. Skopal, J. Pokorný, M. Krátký, and V. Snášel. Revisiting M-Tree Building Principles. In *Advances in Databases and Information Systems*, volume 2798, pages 148–162. Springer, September 2003.

[128] T. Skopal, J. Pokorný, and V. Snášel. Nearest Neighbours Search Using the PM-Tree. In *Database Systems for Advanced Applications*, pages 803–815, 2005.

[129] A. Strehl and J. Ghosh. Cluster ensembles — a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3:583–617, March 2003.

[130] R. Tarjan and J. Van Leeuwen. Worst-case Analysis of Set Union Algorithms. *Journal of the ACM*, 31(2):245–281, April 1984.

[131] J. Tenenbaum, V. De Silva, and J. Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, 290(5500):2319–2323, December 2000.

[132] M. Tepper, D. Acevedo, N. Goussies, J. Jacobo, and M. Mejail. A decision step for shape context matching. In *IEEE International Conference on Image Processing*, 2009.

[133] M. Tepper, F. Gómez, P. Musé, A. Almansa, and M. Mejail. Morphological Shape Context: Semi-locality and Robust Matching in Shape Recognition. In Eduardo Bayro-Corrochano and Jan-Olof Eklundh, editors, *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, volume 5856/2009 of *Lecture Notes in Computer Science*, chapter 15, pages 129–136. Springer, Berlin, Heidelberg, 2009.

[134] J. Toyama, M. Kudo, and H. Imai. Probably correct k-nearest neighbor search in high dimensions. *Pattern Recognition*, 43(4):1361–1372, April 2010.

[135] S. Vega-Pons and J. Ruiz-Shulcloper. Combinación de agrupamientos: un estado del arte. Technical report, CENATAV, La Habana, Cuba, January 2010.

[136] N. Vinh, J. Epps, and J. Bailey. Information theoretic measures for clusterings comparison: is a correction for chance necessary? In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 1073–1080, New York, NY, USA, 2009. ACM.

[137] U. von Luxburg, M. Belkin, and O. Bousquet. Consistency of spectral clustering. *Annals of Statistics*, 36(2):555–586, Apr 2008.

[138] M. Wertheimer. *Laws of organization in perceptual forms*, pages 71–88. Routledge and Kegan Paul, 1938.

[139] A. Witkin. Scale-space filtering: A new approach to multi-scale description. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 9, pages 150–153. IEEE Signal Processing Society, March 1984.

[140] A. Witkin. Scale-space filtering. In *International Joint Conference on Artificial Intelligence*, pages 329–332, San Francisco, CA, USA, 1987. Morgan Kaufmann Publishers Inc.

[141] P. Yianilos. Data structures and algorithms for nearest neighbor search in general metric spaces. In *Proceedings of the fourth annual ACM-SIAM Symposium on Discrete algorithms*, pages 311–321, Philadelphia, PA, USA, 1993. Society for Industrial and Applied Mathematics.

[142] S. Yu and J. Shi. Multiclass spectral clustering. In *IEEE International Conference on Computer Vision*, pages 313–319 vol.1. IEEE Computer Society, 2003.

[143] C. T. Zahn. Graph-Theoretical Methods for Detecting and Describing Gestalt Clusters. *Transactions on Computers*, C-20(1):68–86, 1971.

[144] L. Zelnik-Manor and P. Perona. Self-tuning spectral clustering. In *Advances in Neural Information Processing Systems*, volume 17, pages 1601–1608. MIT Press, 2004.

[145] X. Zhou, G. Wang, J. Xu Yu, and G. Yu. M^+-tree: a new dynamical multidimensional index for metric spaces. In *Proceedings of the 14th Australasian Database Conference*, pages 161–168, Darlinghurst, Australia, Australia, 2003. Australian Computer Society, Inc.