ANALYSIS OF IMAGE TRANFORMS FOR SKETCH-BASED RETRIEVAL

FELIX STÜRMER
(230127)

Diploma Thesis

Technische Universität Berlin
Fakultät IV - Elektrotechnik und Informatik
Computer Graphics

22. Oktober 2012

ERKLÄRUNG DER SELBSTSTÄNDIGKEIT

Hiermit erkläre ich an Eides statt, dass ich die vorliegende Arbeit selbständig und eigenhändig sowie ausschließlich unter Verwendung der aufgeführten Quellen und Hilfsmittel angefertigt habe.

Berlin, den 22. Oktober 2012

Felix Stürmer

ABSTRACT

The steady increase in quantity of digital images creates demand for more efficient and accurate means to search and categorize visual information in large databases. This thesis describes the Fast Discrete Curvelet Transform (FDCT) in the context of sketch-based image retrieval. The properties of the FDCT suggest, that it might be well suited to represent features found in hand-drawn sketches. After examining the general structure of image retrieval systems, several variations of processing pipelines are implemented, that use sketches as query input to retrieve similar images from a database using global and local descriptors. The quality of the retrieved results is evaluated using two benchmarks, that cover both cross-domain and intra-domain scenarios. The results show, that retrieval methods based on the FDCT can compete with previously published algorithms, even for cross-domain queries. It is also shown that the choice between descriptors based on global image characteristics or local bag-of-features descriptors depends on the nature of the images used.

ZUSAMMENFASSUNG

Der stete Zuwachs an digitalen Bildern führt zu einem zunehmenden Bedarf an effizienten Methoden, große Bilddatenbanken zu durchsuchen und deren Inhalt zu kategorisieren. Diese Diplomarbeit beschreibt die Fast Discrete Curvelet Transform (FDCT) im Kontex der Bildersuche anhand von Handzeichnungen. Die Eigenschaften der FDCT legen nahe, dass sie gut geeignet sein könnte um Features in Handzeichnungen abzubilden. Nach der Untersuchung der grundsätzlichen Struktur von Systemen zur Bildersuche werden mehrere Variationen eines solchen Systems implementiert, das Handzeichnungen zur Suche mittels globaler und lokaler Deskriptoren verwendet. Die Qualität der Suchergebnisse wird anhand von zwei Benchmarks gemessen, die sowohl domänenübergreifende als auch domäneninterne Abfragen abdecken. Die Ergebnisse zeigen, dass Suchmethoden auf Basis der FDCT auf gleichem Niveau wie zuvor veröffentlichte Algorithmen liegen. Sie zeigen außerdem, dass die Wahl zwischen Deskriptoren, die auf globalen Bildeigenschaften basieren, und solchen, die lokale bag-of-features Ansätze verfolgen, stark von der Art des verwendeten Bildmaterials abhängig ist.

ACKNOWLEDGMENTS

The creation of this thesis would not have been possible without the help of quite a few people. First and foremost, I want to express my gratitude to my lovely girlfriend Lisa, who lifted me up when motivation failed me and supported me throughout my studies. Similarly, I'm grateful to my parents and the rest of my family for giving me this unique chance.

I would also like to thank Prof. Dr. Alexa and Prof. Dr. Bickel for giving me the opportunity to write this thesis. In addition, my thanks go to Mathias Eitz for supervising my work and giving me valuable advice and feedback during its creation.

Last but not least, I want to thank my fellow student Robert Oskamp (TU Berlin) for the great cooperation during our studies, for many enjoyable discussions and for proofreading this document.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

1.1 MOTIVATION

With the proliferation of cheap and powerful mobile devices over the last years, the amount of images uploaded to the internet each day is growing seemingly without limits. New medical devices produce high-resolution image data in large amounts as the costs of building and running these devices go down. Detailed imagery in various wavelengths of the sky and the planets in our solar system is captured by more and more telescopes on earth and in space. The amount of visual information created today is so large, that no human could ever hope to gain a comprehensive overview. That is why increasing attention is being directed at computer-aided classification and retrieval of those information.

At the core of the research into content-based image retrieval (CBIR) lies the need to be able to access the growing repositories of visual data in a convenient and efficient manner. In this context "convenient" describes the ability for the user to express the query without a complex reformulation of the intent to make it accessible to the query processor. At the same time the computational efficiency becomes more important as the amount of data to search grows. This issue becomes even more critical as the use of mobile, power-limited devices increases across many areas of application, such as autonomous vehicles or handheld augmented reality devices.

Research into text-based information retrieval has brought into existence many statistical methods to query a potentially large body of text using text as the query input. This preserves the close mapping of the intent of the user to the expression of the query and thereby makes the process accessible to users without knowledge about the internal workings of the retrieval system. Providing the means to access a large amount of visual data using a system with similar properties has turned out not to be an easy problem to solve. Using text-based querying for that purpose depends on the ability to reliably label visual data, which would require solving the general object recognition problem first [45]. To avoid that obstacle and to free the retrieval system from the requirement of translating between textual and visual information, many methods to search an image database using visual similarity have been developed.

While the goals of those systems are very similar, they differ considerably in many aspects of the processing pipeline. The query input ranges from example images over drawings to predicates describing color and shape distribution. Similarly, the structure and content of the databases

and the means by which the systems query and rank the results vary significantly. This thesis focuses on evaluating a system that uses hand-drawn sketches as inputs to query databases of either photographs or contour images. The Fast Discrete Curvelet Transform [9] is used to analyse the images, because it should be especially adept at representing curve-like discontinuities of various sizes.

1.2 OUTLINE

Chapter 2 presents the structure of the problem and prior solutions. The following Chapter 3 proposes several variations of a particular solution using the Fast Discrete Curvelet Transform [9]. The experimental setup and its results are documented in Chapter 4 and discussed in Chapter 5. In Chapter 6 several possible conclusions are drawn and pointers towards future research are given.

2 BACKGROUND & RELATED WORK

2.1 GENERAL CHALLENGES OF COMPUTER VISION

2.1.1 *The Semantic Gap*

One of the core insights of computer vision in general and content based image retrieval in particular probably is that human perception is inseparably linked to interpretation by the brain. As a human individual there is no way to directly access visual information without them having been filtered and weighted by one's personal experiences and cultural context. Therefore, when people talk about visual similarity of images, it usually includes a large degree of semantic similarity unconciously added to the perception. The difference between that mode of perception and the current algorithmic ways to analyse visual data has been eloquently coined *the semantic gap* by Smeulders et al. [45]:

> "The semantic gap is the lack of coincidence between the information that one can extract from the visual data and the interpretation that the same data have for a user in a given situation."

Having had that realisation can guide the decision of a researcher or designer of such systems.

2.1.2 *The Sensory Gap*

In addition to the semantic ambiguity described above, another major obstacle of computer vision impacts a CBIR system: *the sensory gap*. This term has also been coined by Smeulders et al. [45], who define it as follows:

> "The sensory gap is the gap between the object in the world and the information in a (computational) description derived from a recording of that scene."

That terse definition includes a multitude of conditions, that can affect an image, which a CBIR system operates on:

ILLUMINATION The brightness or direction of the illumination can hide or accent edges and texture properties in the scene. Similarly, the color of the illumination influences the recorded color information in the image.

RESOLUTION The imaging resolution sets a lower limit on the size of features that can be correctly recognised by any algorithm. As in all signal processing applications, aliasing of high frequency components of the image can introduce further ambiguities. [42]

OCCLUSION Depending on the viewpoint of the recording and the composition of the scene, distinguishing parts of depicted objects may be occluded by other objects or objects may be only partially inside the recorded image.

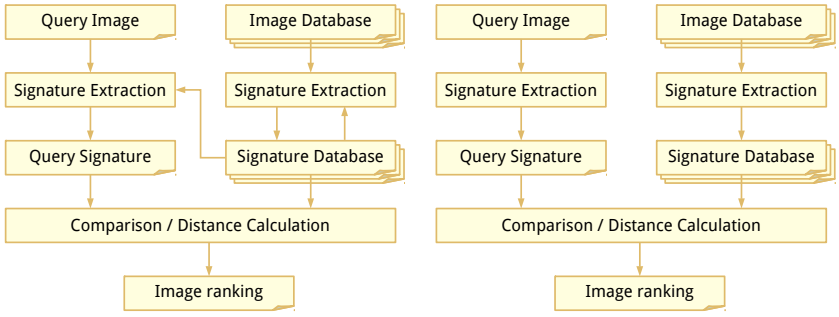
PERSPECTIVE An object's proportions can be distorted by the imaging perspective.

An ideal CBIR system would use feature extraction and comparison methods that can account and correct for such conditions.

2.2 ANATOMY OF A CBIR SYSTEM

The inner workings of most CBIR systems can best be examined by looking at the processing pipeline each query has to go through. The coarse sequence of computational steps is almost the same in all such systems (Figure 1):

1. Acquire the image.
2. Extract the signature using a feature extraction algorithm.
3. Compare the signature to a database containing the signatures of the images to search within.
4. Rank the database images by similarity using the comparison results.



(a) Local features (b) Global features

Figure 1: The processing pipeline for CBIR using both local and global features is very similar. The main difference is in the signature extraction step, in which local features are selected, weighted and/or compressed depending on the results of the signature extraction of the other images in the database.

2.2.1 *Image Acquisition*

The format in which the images are available to the system determines the maximum amount of information available to subsequent analysis steps.

A significant part of the preprocessing usually done after acquisition depends on the broadness of the image domain. The concept of the domain encompasses and describes the variability of many possible image parameters like illumination or composition and is therefore closely related to the sensory gap described above. The narrower the image domain is, the more assumptions the system can make about images from that domain. By their very nature, the domain of sketch based image retrieval systems is usually very broad. It contains the sketches created by the user to query the database as well as the images in the database itself, which can be of a completely different nature, e.g. photographs or paintings.

Another factor usually is the accepted input format of the feature extraction algorithm. Many algorithms like SIFT [25] or SURF [5] are defined for single-channel data, but some have been specifically developed to operate on multi-channel images, like cSIFT [1] and Yang and Xiao's YIQ-based descriptor [52].

2.2.2 *Signature Extraction*

The signature of an image is its representation in the following comparison step. Therefore it should describe the image using its most discriminatory features compared to all other images in the database. Due to the effects of the *sensory gap* discussed above, there is, at the moment, no definitive way to determine the discriminatory power of features in general, even though knowledge about the image domain can guide the descisions. The signature composition depends on both, the kind of features extracted from the image and the way these features are encoded.

Over the last two decades, a wide variety of feature descriptors have been published, which mostly focus on specific types of features. Some techniques use color histograms [49], while others include spatial relations between colors in a region [46] [13] [24]. Many descriptors attempt to capture texture characteristics in an affine-invariant way [41]. Manjunath et al. [29] use Gabor wavelets for image retrieval, which Rubner and Thomasi [39] combine with the earth mover's distance to bypass segmentation. Another class of descriptors focuses on representing shapes using detection of edges and salient points. Lowe [25] developed the now widely adopted SIFT descriptor, that employs clustering of salient points. More recently, the SURF desciptor [5] uses Haar wavelets to deliver comparable performance. Several publications combine feature types to arrive at a more comprehensive descriptor. Oliva and Torralba

[34] capture various properties like "roughness" and "openness" to characterize the whole scene.

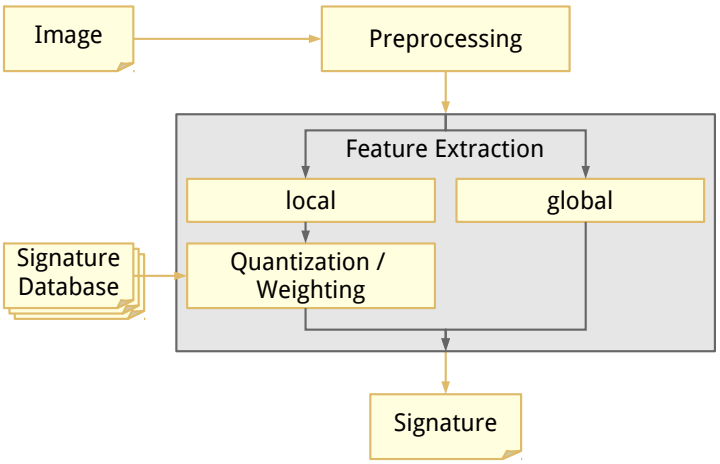


Figure 2: Signature extraction in CBIR systems

2.2.2.1 *Global Features*

Aside from the nature of the features captured by a descriptor, there are also differences in the geometrical scope the features are derived from. Global feature descriptors attempt to capture the structure of the whole scene or describe the distribution of properties across the image like the binary Haar color descriptor published by Utenpattanant et al. [49]. Some global algorithms subdivide the image into regular segments and derive the localized distribution of features for each division. Lazebnik at al. [22] [23] improve upon this concept by creating feature pyramids using iterative subdivision for multiscale analysis. While the computational complexity of those global approaches is usually quite low, they are especially susceptible to problems like partial occlusion or reflections within the scene. The spatial envelope descriptor [34] combines a global spectrogram with locally derived spectral information to produce an overall image descriptor.

2.2.2.2 *Local Features*

In contrast to the global approach, many CBIR systems employ "bag-of-features" descriptors, that represent the image as an unsorted collection of local features extracted from small patches of the image. The unsorted nature of the feature collection leads to loss of large-scale geometric structures that can be counteracted by a suitable choice of the patch sizes. When the local feature descriptors are invariant to rotation, scale or similar deformations, the sensitivity to viewpoint variations or occlusions decreases. The prominent SIFT descriptor [25] achieves this by selecting the feature locations such that they can be normalized with respect to scale, orientation and limited 3D projections. The SURF de-

scriptor by Bay et al [5] gives similar results, but has reduced computational requirements. The HOG descriptor [12] calculates histograms of gradient directions on regular grid cells to describe the local angular distribution of edges.

2.2.2.3 *Dimensionality Reduction*

The signatures produced by local descriptors are often large sets of vectors, that are themselves of considerable size. For example, the SIFT descriptor describes each image using about 1000 local feature vectors of 160 values each. Such large numbers of vectors are expensive to store and compare, which is why one of several data reduction methods is commonly used.

PRINCIPAL COMPONENT ANALYSIS The Principle Component Analysis (PCA) is a transformation, that computes the orthogonal basis best suited to describe the variance of the data. An n-dimensional data set is linearly mapped to a coordinate system, in which the direction of the first axis a_1 is the direction with the largest variance in the data. The following axes' a_i, $i \in \{2, \ldots, n\}$ directions correspond to the orthogonal directions with the next-largest variances in descending order. By choosing the p largest component vectors and performing an inverse transformation of the PCA-transformed data, a projection of the original data in p dimensions can be obtained. Due to the choice of the vectors for the inverse transformation, the projection discards only the parts of each observation that vary the least between all observations.

PCA has been applied to the face recognition problem using intensity images (eigenfaces) [48], wavelets (waveletfaces) [16] and more recently curvelets (curveletfaces) [28]. Ke et al. [21] also used it to improve the robustness of the SIFT [25] descriptor. To overcome the limited between-class discrimination of PCA, it has been combined with Linear Discriminant Analysis (LDA), yielding even better results [27].

VISUAL WORDS AND CLUSTERING Instead of reducing the size of the individual feature vectors, the bag-of-features approach collects the feature vectors into a single signature vector to represent the image. This is done by determining a codebook of representative feature vectors, the "visual words" [44], and assigning each local feature vector to the most similar visual word. A histogram of the distribution of visual words can then be calculated as a signature for each image.

To create the codebook, the large number of feature vectors extracted from local patches of each image in the database are grouped into clusters of similar vectors. The optimal number of clusters is usually determined experimentally and varies with other processing parameters such as the sampling strategy [33] [53].

The most common clusting method used in numerous publications [55] [44] [11] [17] [50] is k-means clustering. This algorithm uses the

euclidean distance as a metric to assign each observation x_p to the nearest cluster S_i with mean m_i, $i \in \{1, \ldots, k\}$. The goal is to minimize the variance within each cluster:

$$\sum_{i=1}^{k} \sum_{x_p \in S_i} \|x_p - m_i\|^2$$

Lloyd's algorithm is the usual way to calculate a k-means partition. It requires a set of k intial cluster centers, that are often randomly chosen from the dataset or randomly generated. The cluster centers m_i are then iteratively adjusted until no reassignment takes place in two consecutive iterations t and $t+1$. In each iteration, each oberservation x_i is assigned to exactly one cluster S_i using

$$S_{i,t} = \left\{ x_p : \|x_p - m_{i,t}\| \leqslant \|x_p - m_{j,t}\| \quad \forall j \in \{1, \ldots, k\} \right\}.$$

The centers are then recalculated as

$$m_{i,t+1} = |S_{i,t}|^{-1} \sum_{x_p \in S_{i,t}} x_p.$$

The greedy nature of the algorithm and the random initialization mean that it is merely a heuristic and can converge on a local minimum, that is not a global minimum. Other clustering methods work hierarchically by agglomerating and dividing the data. Applications of that have been published by Nister and Stewenius [32] and Philbin et al. [37], who recursively divide or merge partitions to optimize the vocabulary.

Once the clustering algorithm has converged, the cluster centers m_i will be used as the visual words of the codebook. To create the signature vector of an image, its feature vectors x_p will be quantized by grouping them into sets S_i, $i \in \{1, \ldots, k\}$ such that

$$S_i = \left\{ x_p : \|x_p - m_i\| \leqslant \|x_p - m_j\| \quad \forall j \in \{1, \ldots, k\} \right\}.$$

The final image signature $\tilde{I}$ then is the vector of the cardinalities of these sets:

$$\tilde{I} = \left(|S_1|, |S_2|, \ldots, |S_{k-1}|, |S_k| \right)$$

2.2.3 *Comparison and Ranking*

2.2.3.1 *Weighting*

In the vocabulary of visual words created via clustering, a set of representatives for a group of features has been defined. How distinctive the presence of visual word in the signature of a specific image actually is has not yet been taken into account though. To remedy this, weighting schemes are often used, that modify the signatures to enhance distinguishing characteristics and make the ranking more accurate.

LINEAR SUPPORT VECTOR MACHINES The concept of support vector machines (SVMs) is usually employed when a feature needs to be classified into one or more classes. While such classification problems occur frequently in computer vision in the fields of object detection [38] [11], human detection [12] or scene classification [53], the class concept is too limited for general image retrieval. To achieve the classification, a SVM constructs a hyperplane, that optimally separates two sets of points $x_i \in \mathbb{R}^n$ with labels $y_i \in \{-1, 1\}$ in a training dataset $\{(x_i, y_i)\}$. The hyperplane can be defined using its normal vector w and offset b as

$$w \cdot x - b = 0.$$

To turn the problem into an optimization problem, the plane is split up into two parallel hyperplanes described by

$$w \cdot x - b = 1 \text{ and } w \cdot x - b = -1.$$

The region between these two planes is characterized by their distance $\frac{2}{\|w\|}$, which needs to be minimized while satisfying

$$\begin{aligned} w \cdot x_i - b &\geqslant \ \ 1 \quad \text{if } y_i = 1 \qquad \text{or} \\ w \cdot x_i - b &\leqslant -1 \quad \text{if } y_i = -1 \end{aligned}$$

for all i. It is possible to transform this into an equivalent quadratic optimization problem solvable by standard quadratic programming algorithms.

Even though image retrieval as described in this thesis is not a simple classification problem, linear SVMs can still be of use. Guyon et al. [20] showed, that the components of w can be used as weights describing the discriminative power of feature vectors. This was applied to image retrieval by Shrivastava et al. [43]. There, the authors trained a SVM for each query image I_q with the query image's signature constituting one class and the database images' signatures x_i making up the other class. The pairwise similarity could then easily be obtained from the learned weights w_q as

$$S(I_q, I_i) = w_q^T x_i.$$

That way, common features in the query signature are effectively downvoted, while unique features are assigned larger weights.

TF-IDF Along with the visual word analogy from the field of text retrieval, the statistical method of TF-IDF weighting [4] has been applied to CBIR [44]. It is a technique to weight features (terms) in a document in relation to their occurence in the document and the whole database. The term frequency $tf_{i,j}$ is the number of normalized occurences of term t_i in document d_j. The normalization can be performed in different ways. The most common ones are dividing by the total number of words

n_j in the document or the maximum term count $\max\{tc_{i,j} : \forall t_i \in d_j\}$ of any term in the document j:

$$tf_{i,j} = \frac{tc_{i,j}}{n_j} \text{ or } tf_{i,j} = \frac{tc_{i,j}}{\max\{tc_{i,j} : \forall t_i \in d_j\}}$$

The occurence of a term in the database D is measured as the logarithm of the quotient of the overall number of documents $|D|$ and the number m_i of documents containing the term t_i:

$$idf_i = \log \frac{|D|}{m_i}$$

Therefore, the total weight $w_{i,j}$ of a term is given as

$$w_{i,j} = tf_{i,j} \cdot idf_i = \frac{tc_{i,j}}{n_j} \cdot \log \frac{|D|}{m_i}$$

assuming the document length is used for normalization.

2.2.3.2 *Distance Metrics*

To retrieve the similar images from the database, a ranking of database images in respect to their similarity to the query image must be obtained. Since each image is represented by a signature vector, this means the pairwise distance between the query image's signature and each database image's signature can be used to sort the list of query results. This section will describe several distance metrics and similarity measure candidates, which can be used to that end.

EUCLIDEAN DISTANCE The simplest and probably most widely used distance metric is the euclidean distance. Its two-dimensional variant is derived from the formula of Pythagoras. Generalized to n-dimensional points $p = (p_1, p_2, \ldots, p_n)$ and $q = (q_1, q_2, \ldots, q_n)$, it can be written as

$$d_{EUCL}(p, q) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}.$$

MAHALANOBIS DISTANCE If the distance metric is used for clustering or classification of datasets with non-spherical within-class distributions, the euclidean distance will naturally not perform well. A common alternative is the Mahalanobis distance, that incorporates the correlation of the dataset into the result. The distance of a point $p = (p_1, p_2, \ldots, p_n)$ to a cluster with mean $m = (m_1, m_2, \ldots, m_n)$ is

$$d_{MAHA}(p, m) = \sqrt{(p - m)^T S^{-1} (p - m)},$$

where S is the cluster's covariance matrix. In practice, it has been used by Mikolajczyk and Schmid [31] and Sivic and Zisserman [44] to compare feature vectors of the SIFT [25] descriptor.

COSINE DISTANCE A metric sometimes used in information retrieval applications is the cosine distance or cosine similarity. The similarity is defined as the cosine of the angle between two vectors $p = (p_1, p_2, \ldots, p_n)$ and $q = (q_1, q_2, \ldots, q_n)$ when interpreted geometrically:

$$\cos(\theta_{p,q}) = \frac{p \cdot q}{\|p\|\|q\|}$$

This means that the metric effectively normalizes the vectors in respect to their euclidean length. That effect is often used in text retrieval to achieve invariance regarding the document size.

The distance measure derived from the cosine similarity is

$$d_{COS} = 1 - \frac{p \cdot q}{\|p\|\|q\|}.$$

EARTH MOVER'S DISTANCE The Earth Mover's Distance (EMD) is an application of the discrete transportation problem, which was first introduced into the field of computer vision by Peleg et al. [36]. The use of the algorithm as a signature comparison metric in image retrieval was published by Rubner et al. [40]. At the abstract level, the distance is calculated as the amount of work necessary to transform one signature into the other. A main advantage of the EMD over most other distance measures is that it accounts for inter-bin distances (ground distances) in binned distributions. Another useful property is the ability to compare distributions of different sizes, e.g. for partial matching. For signatures $P = \{(p_1, w_{p_1}), \ldots, (p_n, w_{p_n})\}$ and $Q = \{(q_1, w_{q_1}), \ldots, (q_m, w_{q_m})\}$ of bin centers p_i and q_i with bin sizes w_{p_i} and w_{q_i} the pairwise ground distances can be represented in a $n \times m$ matrix D. These ground distances $d_{i,j}$ between two bins is then interpreted as the costs of moving a unit of goods from one bin to the other. The optimal solution minimizes the overall costs by finding flow values $f_{i,j}$ between each pair p_i and q_i, that satisfy the constraints

$$\begin{aligned}
f_{i,j} &\geqslant 0 \\
\sum_{j=1}^{n} f_{i,j} &\leqslant w_{p_i} \\
\sum_{i=1}^{m} f_{i,j} &\leqslant w_{q_j} \\
\sum_{i=1}^{n}\sum_{j=1}^{m} f_{i,j} &= \min\left(\sum_{i=1}^{n} w_{p_i}, \sum_{j=1}^{m} w_{q_j}\right)
\end{aligned}$$

for all $1 \leqslant i \leqslant n$ and $1 \leqslant j \leqslant m$. The distance between signatures P and Q can then be calculated as

$$d_{EMD}(P,Q) = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m} d_{i,j} f_{i,j}}{\sum_{i=1}^{n}\sum_{j=1}^{m} f_{i,j}}.$$

Rubner et al. [40] propose using a simplex-based algorithm to solve the transportation problem, that achieves good performance by exploiting the specific problem structure. Zhang at al. [54] use Rubner's implementation to train a support vector machine for image classification with Gaussian kernels scaled by the EMD.

HISTOGRAM INTERSECTION A very inexpensive way to compare two distributions has been shown by Swain and Ballard [47], namely the histogram intersection technique. While they used it to compare color histograms, it should work equally well for binned distributions generated using vector quantization. The normalized similarity measure between two histograms $P = (p_1, \ldots, p_n)$ and $Q = (q_1, \ldots, q_n)$ is

$$s_{HI}(P,Q) = \frac{\sum_{i=1}^{n} \min(p_i, q_i)}{\sum_{i=1}^{n} q_i}$$

A variation of the above definition treats partial matches as perfect matches by adjusting the denominator:

$$s_{HI}(P,Q) = \frac{\sum_{i=1}^{n} \min(p_i, q_i)}{\min\left(\sum_{i=1}^{n} p_i, \sum_{i=1}^{n} q_i\right)}$$

2.3 IMAGE TRANSFORMATIONS FOR FEATURE EXTRACTION

2.3.1 *The Gabor Filter*

In content based image retrieval, the inability of the Fourier transform to localize the frequency components in time disqualifies it as a suitable analysis method. Instead, wavelets are often used to obtain a time-frequency representation of the signal. A particularly successful application to CBIR was the use of gabor wavelets as proposed by Manjunath and Ma [29]. The mother Gabor wavelet $g(x,y)$ with the sinus frequency W and gaussian scaling parameters σ_x, σ_y is given as

$$g(x,y) = \left(\frac{1}{2\pi\sigma_x\sigma_y}\right) \exp\left(-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right) + i \cdot 2\pi W x\right)$$

and thus its Fourier transform can be written as

$$\hat{g}(u,v) = \exp\left(-\frac{1}{2}\left(\frac{(u-W)^2}{\sigma_u^2} + \frac{v^2}{\sigma_v^2}\right)\right),\ \sigma_u = \frac{1}{2\pi\sigma_x} \text{ and } \sigma_v = \frac{1}{2\pi\sigma_y}.$$

The individual wavelets $g_{m,n}(x,y)$ are generated from $g(x,y)$ by scaling and rotation by $\theta = \frac{n\pi}{K}$ for all $0 < n < K$ orientations:

$$g_{m,n}(x,y) = a^{-m} g\left(a^{-m}(x\cos\theta + y\sin\theta), a^{-m}(-x\sin\theta + y\cos\theta)\right).$$

To avoid redundancy due to overlap, the scaling parameters σ_u and σ_v are chosen in a way that the frequency spectra can be tiled as in Figure 3. A normalization to zero mean can be added to remove the influence of the input's intensity value scale.

The individual coefficients can be obtained by convolving each filter of the filter bank with the signal.

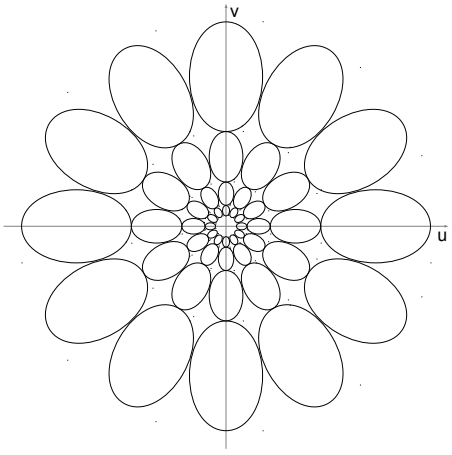


Figure 3: Tiling of Gabor wavelets. Note that due to the elliptical shape of the wavelets, some parts of the spectrum are left uncovered. (Image inspired by Manjunath and Ma [29])

2.3.2 *The Continuous Curvelet Transform*

The formulation of the continuous curvelet transform (CCT) by Candes and Donoho [7] was based on Candes' previous definition and expansion of the ridgelet transform [6]. In that publication they looked at the state of research into efficient representations of edge discontinuities. They based their research on two realisations:

1. A nonadaptive approach of signal approximation can compete with many of the adaptive schemes prevalent in previous research. At the same time the non-adaptivity comes with a greatly reduced computational overhead and reduced requirements for a priori knowledge. Obtaining that knowledge in the presence of blurred or noisy data can sometimes be unfeasible.

2. Wavelet transforms can represent point singularities in a signal of up to two dimensions in a near-ideal manner, but fail to perform equally well on edges: Given a two-dimensional object in signal f, that is smooth except for discontinuities along a curve, a wavelet approximation $\tilde{f}_m^W$ from the m largest coefficients exhibits an error of

$$\|f - \tilde{f}_m^W\|^2 \propto m^{-1}, \text{ for } m \to \infty$$

since up to $O(2^j)$ localized wavelets are needed to represent the signal along the edge. That falls short of what an approximation $\tilde{f}_m^T$ using a series of m adapted triangles could achieve:

$$\|f - \tilde{f}_m^T\|^2 \propto m^{-2}, \text{ for } m \to \infty$$

They showed that a similarly precise approximation can be achieved by combining Candes' ridgelet analysis [6] with smart windowing functions and bandpass filters. The steps of the transformation were described as follows:

1. Decomposition of the signal into subbands of scale-dependent size
2. Partitioning of each subband into squares
3. Normalisation of each square to unit scale
4. Analysis of each square in an orthonormal ridgelet system

The result was the formulation of a decomposition that matched the parabolic scaling law $\mathit{width} \propto \mathit{length}^2$ often observed in curves.

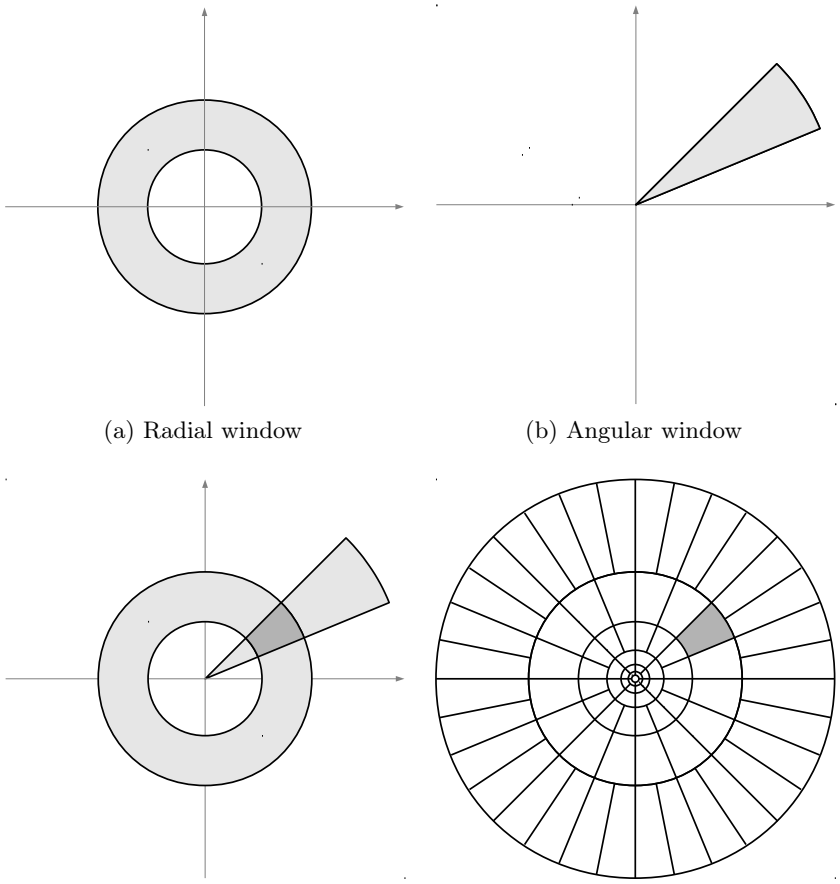
The above formulation became known as the curvelet 99 transform when Candes and Donoho revised it soon after [8]. The new version is not dependent on ridgelets and aims to remove some shortcomings of the curvelet 99 transform, namely a simpler mathematical analysis, fewer parameters and improved efficiency regarding digital implementations, which will be described later.

The curvelet transform in $\mathbb{R}^2$ works by localising the curvelet waveforms in the time domain. The "mother" curvelet waveform $\varphi_j(x)$ is defined using two frequency domain windows $W(r)$, the "radial window" (Figure 4a), and $V(t)$, the "angular window" (Figure 4b). These windows must obey the admissibility condition for wavelets. They can be combined in U_j (Figure 4c):

$$U_j(r,\theta) = 2^{\frac{-3j}{4}} W(2^{-j}r) V(\frac{2^{\lfloor \frac{j}{2} \rfloor}\theta}{2\pi}).$$

The waveform φ_j can then be expressed as being the inverse Fourier transform of $\hat{\varphi}_j = U_j$ and all curvelets of a scale 2^{-j} can be derived by

- rotating φ_j by a sequence of equispaced rotation angles $\theta_l = 2\pi \cdot 2^{-\lfloor \frac{j}{2} \rfloor} \cdot l$ with $l = 0, 1, \ldots$ such that $0 < \theta_l < 2\pi$ and



(a) Radial window (b) Angular window

(c) Combined window (d) Complete coronisation

Figure 4: The window $W(2^{-j}r)$ at scale 2^j (a) is combined with the window $V(t)$ (b) to form a support wedge for the curvelet (c). The wedge roughly obeys a $\mathit{width} \propto \mathit{length}^2$ relation. (d) shows the wedge within a schema of the complete tiling in frequency domain.

- translating φ_j by a sequence of offsets $k = (k_1, k_2) \in \mathbb{Z}^2$:

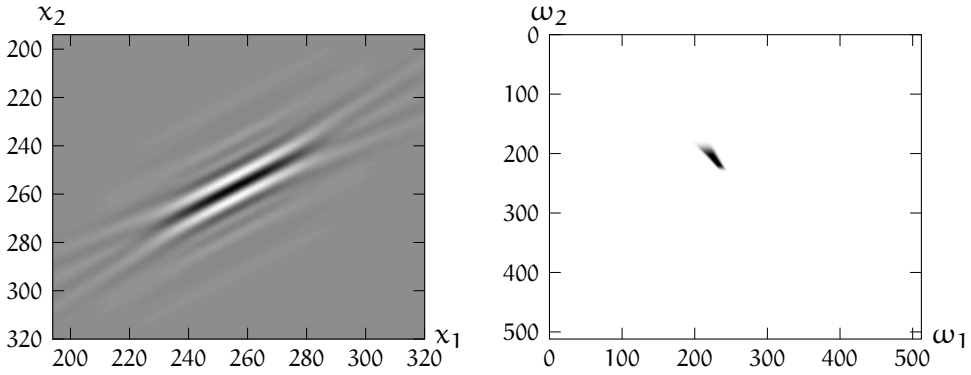
$$\varphi_{j,k,l}(x) = \varphi_j(R_{\theta_l}(x - x_k^{(j,l)})), \tag{1}$$

where $x = (x_1, x_2)$, R_θ is the rotation matrix for angle θ and $x_k^{(j,l)} = R_{\theta_l}^{-1}(k_1 \cdot 2^{-j}, k_2 \cdot 2^{-\frac{j}{2}})$. Figure 5a and Figure 5b show example waveforms and their supports in frequency domain.

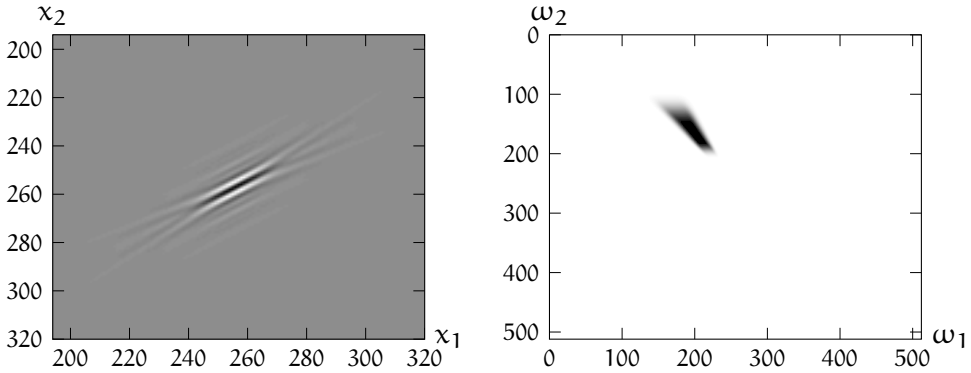
Each curvelet coefficient $c(j, l, k)$ can then be calculated as the inner product of $f \in L^2(\mathbb{R}^2)$ and curvelet $\varphi_{j,l,k}$:

$$c(j, l, k) := \langle f, \varphi_{j,l,k} \rangle = \int_{\mathbb{R}^2} f(x)\overline{\varphi_{j,l,k}(x)}dx \tag{2}$$

As visible in Figure 4d curvelets also have non-directional components at the coarsest scale, similar to those found in the wavelet transform. Those curvelets will be defined using a special low-pass filter window



(a) Coarse Curvelet waveform in time and frequency domain



(b) Fine Curvelet waveform in time and frequency domain

Figure 5: Curvelet waveforms on coarse (a) or fine (b) scale with time domain shown on the left and frequency domain shown on the right side (from Candes et al. [9]).

W_0, which is characterized as being the remainder of the tiling not covered by the previously described radial windows:

$$W_0(r)^2 = 1 - \sum_{j \geqslant 0} W(2^{-j}r)^2$$

With the help of this window, defining the coarse scale curvelet $\varphi_{j_0,k}$ via its Fourier transform $\hat{\varphi}_{j_0}$ is possible:

$$\hat{\varphi}_{j_0}(\omega) = 2^{-j_0} W_0(2^{-j_0}|\omega|)$$
$$\varphi_{j_0,k}(x) = \varphi_{j_0}(x - 2^{-j_0}k),$$

where $k = (k_1, k_2) \in \mathbb{Z}^2$.

Note that, in contrast to the Gabor wavelets (Figure 3), there is no gap in the curvelet tiling, so no information is lost.

2.3.3 *The Fast Discrete Curvelet Transform*

Based on the above definition of the continuous curvelet transform, a team around the authors of the original curvelet publication presented

two digital, discrete implementations of the transform: the Fast Discrete Curvelet Transform (FDCT) [9]. The implementations have been described in 2D and 3D, but since this paper deals exclusively with 2D images, the explanation below will also be restricted to two dimensions.

The digital versions of the transforms operate on arrays $f[t_1, t_2]$ with $0 \leqslant t_1, t_2 < n$ to produce coefficients $c^D(j, l, k)$ in a way consistent with the continuous version (Equation 2):

$$c^D(j, l, k) := \sum_{0 \leqslant t_1, t_2 < n} f[t_1, t_2] \overline{\varphi^D_{j,l,k}[t_1, t_2]}. \tag{3}$$

Since the windows used in the continuous form are based on rotations and dyadic coronae, they are not well suited for use with cartesian arrays. The discrete formulation substitutes them with appropriate concepts. Instead of concentric annuli, the window function W_j^D generates concentric, square "rings" using the square windows $\Phi_j(\omega_1, \omega_2) = \phi(2^{-j}\omega_1)\phi(2^{-j}\omega_2)$, with ϕ being a low-pass 1D window:

$$W_j^D(\omega) = \sqrt{\Phi^2_{j+1}(\omega) - \Phi^2_j(\omega)}, \quad j \geqslant 0.$$

The rotation matrix R_θ is replaced by the shear matrix S_θ to create the combined window function

$$U^D_{j,l} := W_j^D(\omega) V_j(S_{\theta_l}\omega).$$

The sequence θ_l is defined as a sequence of equispaced slopes $\tan(\theta_l) := l \cdot 2^{-\lfloor \frac{j}{2} \rfloor}$ with $l = -2^{\lfloor \frac{j}{2} \rfloor}, \ldots, 2^{\lfloor \frac{j}{2} \rfloor} - 1$.

Special attention must be paid to creating the windows, that touch the diagonals, to ensure

$$\sum_j \sum_l |U^D_{j,l}(\omega)|^2 = 1$$

holds, so the tiling obeys the admissibility condition just like in the continuous case. Figure 6 shows a tiling of all $U^D_{j,l}$.

2.3.3.1 *FDCT using unequispaced FFTs*

The first implementation of the discrete curvelet transform transfers the input array $f[t_1, t_2]$, $0 \leqslant t_1, t_2 < n$ into the Fourier domain to obtain $\hat{f}[n_1, n_2]$:

$$\hat{f}[n_1, n_2] = \sum_{t_1, t_2 = 0}^{n-1} f[t_1, t_2] e^{-\frac{i2\pi(n_1 t_1 + n_2 t_2)}{n}}, \quad -\frac{n}{2} \leqslant n_1, n_2 < \frac{n}{2}$$

The obtained Fourier samples need to be interpolated for each pair of scale j and angle l to match the grid of the sheared support window $U_j^D[n_1, n_2]$. The authors achieve this by resampling $\hat{f}$ on the grid implied by the sheared window for each angle via a series of 1D fast

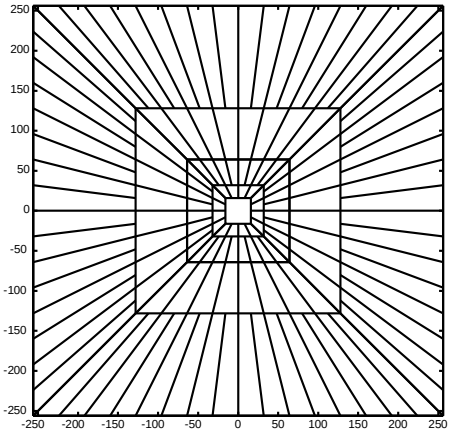


Figure 6: Discrete frequency tiling using concentric squares (from Candes et al. [9])

Fourier transforms. These transforms represent a polynomial interpolation of each "column" of the parallelogram P_j containing the sheared window (Figure 7a), that can be computed with a $O(n^2 \log n)$ complexity in a sufficiently exact approximation.

This yields an object $\hat{f}[n_1, n_2 - n_1 \tan\theta_l]$ for $(n_1, n_2) \in P_j$, that can be multiplied with the window U_j^D described above in order to create a localized "wedge" with the orientation θ_l:

$$f_{j,l}^D[n_1, n_2] = \hat{f}[n_1, n_2 - n_1 \tan\theta_l] U_j^D[n_1, n_2]$$

The discrete curvelet coefficients $c^D(j, l, k)$ can then be calculated by applying the inverse 2D Fourier transform:

$$c^D(j, k, l) = \sum_{n_1, n_2 \in P_j} \hat{f}[n_1, n_2 - n_1 \tan\theta_l] U_j^D[n_1, n_2] e^{i2\pi\left(\frac{k_1 n_1}{L_{1,j}} + \frac{k_2 n_2}{L_{2,j}}\right)},$$

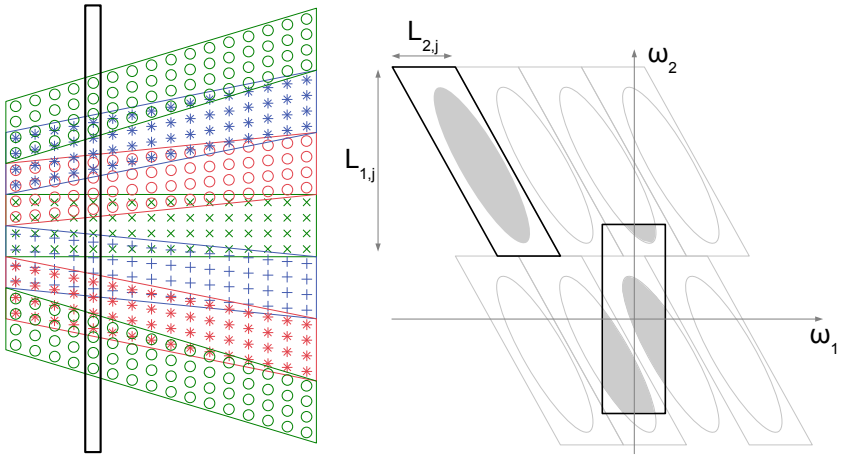
in which $L_{1,j}$ and $L_{2,j}$ are the length and width of the rectangle supporting U_j^D.

2.3.3.2 *FDCT using wrapping*

As before, FDCT using wrapping first calculates $\hat{f}[n_1, n_2]$ with $-\frac{n}{2} \leqslant n_1, n_2 < \frac{n}{2}$ as the Fourier transform of the input f. The sample is then localized by multiplying it with a window $U_{j,l}^D$ for each angle j and scale l:

$$d_{j,l}[n_1, n_2] = U_{j,l}^D \hat{f}[n_1, n_2]$$

To avoid the computationally costly interpolation step required in the USFFT approach, this method keeps the rectangular grid of the input signal. Because an axis-aligned bounding box of the window U_j^D in Fourier domain cannot maintain the $width \propto length^2$ proportions of the window, applying an inverse Fourier transform on such a bounding



(a) Sheared USFFT tiling (b) Sheared tiling for wrapping

Figure 7: (a) illustrates the respective grid for each parallelogram containing several sheared support windows of the "east quadrant". The box highlights one of the columns, that represent one of the 1D polynomial interpolation problems solved for resampling. In (b), the parallelogram $P_{j,l}$ is shown on top of the tilted tiling of a curvelet in frequency domain. Due to the periodicity, the rectangle in the center contains the same curvelet, but has a much smaller axis aligned bounding box for the FFT to operate on. Both images have been adapted from Candes et al. [9].

box in general would lead to significant oversampling of the coefficients and thereby increase the memory requirements for fine scale curvelets beyond that of the the USFFT approach. In order to circumvent that, the authors utilize the periodic nature of the Fourier transform and propose generating a periodically wrapped version of the fourier samples. For $P_{j,l}$ as the bounding parallelogram of $U^D_{j,l}$, $L_{1,j}$ and $L_{2,j}$ are the period lengths by which to translate $P_{j,l}$ in the horizontal and vertical direction to produce a suitable tiling for each orientation θ_l (Figure 7). Thus, the wrapped, localized data are

$$f^D_{j,l}[n_1, n_2] = W d_{j,l}[n_1, n_2] = \sum_{m_1 \in \mathbb{Z}} \sum_{m_2 \in \mathbb{Z}} d_{j,l}[n_1 + m_1 L_{1,j}, n_2 + m_2 L_{2,j}]$$

with $0 \leqslant n_1 < L_{1,j}$ and $0 \leqslant n_2 < L_{2,j}$, which gives a rectangle of size $L_{1,j}$ times $L_{2,j}$.

Again, the discrete curvelet coefficients $c^D(j, l, k)$ can then be collected using the inverse 2D Fourier transform:

$$c^D(j, k, l) = \sum_{n_1, n_2 \in P_j} f^D_{j,l}[n_1, n_2] e^{i2\pi\left(\frac{k_1 n_1}{L_{1,j}} + \frac{k_2 n_2}{L_{2,j}}\right)}.$$

2.3.4 *gPb Contour Detection*

Maire et al. [26] describe an improvement of the contour detector published by Martin et al. [30], that includes global information in addition to local cues. The orientation-specific local parameters G_i extracted from a circular neighborhood around the location (x, y) are brightness, color and texture gradients on three scales. They are summarized as a coefficient $mPb(x, y, \theta)$ using a weighted sum with weights α_i:

$$mPb(x, y, \theta) = \sum_{i=1}^{9} \alpha_i G_i(x, y, \theta)$$

The global component $sPb(x, y, \theta)$ is the result of applying a filterbank of directional gaussian derivatives to a set of k generalized eigenvectors v_j, $j \in \{1, \ldots, k\}$. The linear system these eigenvectors are obtained from has an affinity matrix derived from the intervening contour cue [18]. The linear combination of the individual directional derivatives then represents the large-scale contours in the image:

$$sPb(x, y, \theta) = \sum_{j=1}^{k} \frac{1}{\sqrt{\lambda_j}} sPb_{v_j}(x, y, \theta)$$

A further linear combination of the local component mPb and the global component sPb with learned weights α_i and γ provides a detailed map of contours in the image while limiting the amount of clutter compared to a purely local contour detector:

$$gPb(x, y, \theta) = \sum_{i=1}^{9} \alpha_i G_i(x, y, \theta) + \gamma \cdot sPb(x, y, \theta)$$

From these directional contour maps, Arbeláez et al. derived a hierarchical contour detector [3], that conditionally joins adjacent regions to obtain closed-contour maps of high quality.

3 PROPOSED SOLUTION

The image processing pipeline described in this thesis aims to be suitable for content based image retrieval using hand-drawn sketches for querying. The main interest was to evaluate, how well the Fast Discrete Curvelet Transform (FDCT) [9] is able to represent the lines in hand-drawn sketches as well as salient edges in photos or paintings. To explore the effects of preprocessing and signature extraction, several variations of the pipeline have been implemented. The used preprocessing steps include applying the Sobel operator, extracting a Canny edge map or determining segment borders using the gPb algorithm published by Arbelaez at al. [2]. Signatures are constructed using both, global curvelet features, and a bag-of-features approach similar to what was described by Sivic and Zisserman [44] and Eitz et al. [15].

The following sections will describe the variations of the processing stages *image acquisition*, *signature extraction*, and *ranking*. To reference the individual variations unambiguously, labels like LUMA will be introduced for each component.

3.1 IMAGE ACQUISITION

Since one premise of the system is that hand-drawn sketches are compared with a large body of images from various sources, a division into two input domains seems obvious. The first domain, the domain of query sketches, is quite narrow, because we can characterize its members as binary images with large, smooth areas separated by discontinuities along curves. The database images, that make up the second domain, are not subject to such limitations. They may be color photographs (Figure 8a), paintings, computer renderings or black-and-white sketches.

LUMA Because the Fast Discrete Curvelet Transform used in every variant of the signature extraction step takes a single 2D matrix as input, images with more than one color channel need to be reduced to one channel. The RGB values from the benchmark dataset have therefore been converted to greyscale images using the definition of luma according to ITU standards [35] (Figure 8b). Each pixel with red, green and blue values (R, G, B) is mapped to a luminance value Y using

$$Y = \frac{299}{1000}R + \frac{587}{1000}G + \frac{114}{1000}B.$$

SOBEL In order to make comparing the query sketch to the database images more effective, an edge extraction algorithm can be applied to

each database image. The Sobel operator calculates horizontal and vertical gradients by convolving the image with the 3×3 kernels

$$K_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \text{ and } K_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

to obtain the directional gradients G_x and G_y. The overall response is the gradient magnitude G at each pixel location (Figure 8c):

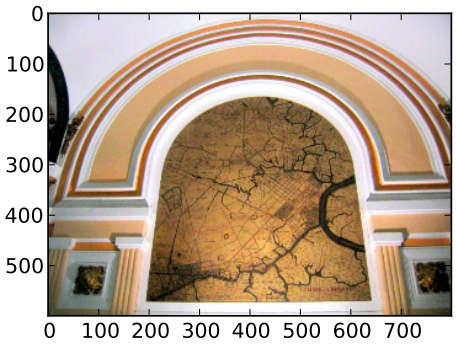
$$G = \sqrt{G_x^2 + G_y^2}$$

CANNY A slightly more complex way to extract edges is the Canny edge detector [10]. Initially, the image is smoothed via a convolution with a small Gaussian kernel to reduce the susceptibility to noise, even though this increases the localization error of the edge detection. On the smoothed image the gradient magnitude is calculated using the Sobel operator described above. The angle of the gradient can be calculated from the directional gradients G_x and G_y using

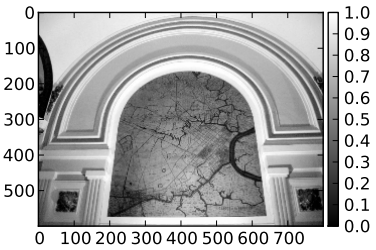
$$\Theta = \arctan \frac{G_x}{G_y}$$

and quantized into bins for $0°$, $45°$, $90°$ and $135°$. Thin edges can be obtained from the gradient magnitudes by performing non-maximum suppression along the direction perpendicular to the gradient direction, e.g. a pixel is marked as being on a $90°$ edge if its magnitude is larger than the magnitudes north and south of its location. To avoid lines being broken up by noisy fluctuations, the edges are traced along their direction and gaps are filled in if the signal within the gap is above a certain threshold. The result is a binary edge map of the whole image (Figure 8d).

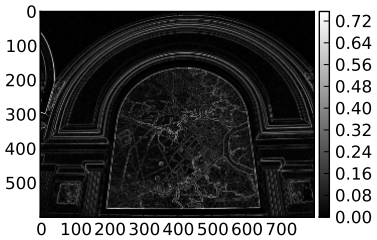
SEGMENT What a human sketches as a line in an image is often a boundary between two regions with different color or texture characteristics. Therefore the output from image segmentation algorithms can also indicate the location of edges. The hierarchical segmentation algorithm gPb-owt-ucm published by Arbeláez et al. [3] [2] was chosen because it represents the most recent advances in contour detection and it incorporates both local and global image information. On the gPb contour detector described in Section 2.3.4 they apply the Oriented Watershed Transform, that merges adjacent regions of an over-segmented image. The criterion for merging is the strength of the boundary shared by the two regions.



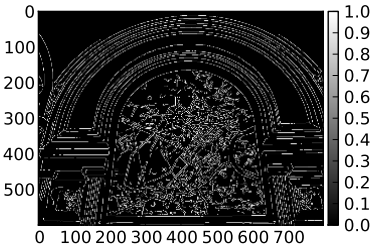
(a) Original image



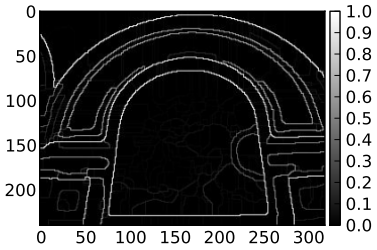
(b) Image after luma conversion



(c) Image after Sobel operator



(d) Image after Canny operator



(e) Image after gPb contour detection

Figure 8: An example image from the benchmark dataset [15], that has been processed using various edge extraction algorithms.

3.2 SIGNATURE EXTRACTION

Each method of signature extraction described in this section has the Fast Discrete Curvelet Transform at its heart. The curvelet transform has two main parameters, that influence the result: The number of angles N_θ used at the coarsest scale and the number of scales N_j, which corresponds to the number of concentric squares shown in Figure 7. Experiments conducted to determine the optimal values of these parameters have shown that using more scales than 4 does not provide benefits that would justify the increased amount of processing time. Similar findings were reported by Mandal et al. [27] and Guha and Wu [19]. Furthermore, since the coarsest scale is non-directional, as explained in Section 2.3.2, it is ignored in further computations. Finally, since the curvelet coefficients can have positive or negative sign, only the absolute value of the coefficient is used in the calculations of the mean values.

The response image generated by the FDCT for each pair of scale and angle is too large to be considered for the signature directly. Therefore the response image $C_{s,\theta}$ for scale s and angle θ is subdivided into n^2 equally sized grid cells $G_{s,\theta,x,y}$ with $x, y \in \{1, \ldots, n\}$:

$$C_{s,\theta} = \begin{bmatrix} G_{s,\theta,1,1} & G_{s,\theta,1,2} & \cdots & G_{s,\theta,1,n} \\ G_{s,\theta,2,1} & G_{s,\theta,2,2} & \cdots & G_{s,\theta,2,n} \\ \vdots & \vdots & \ddots & \vdots \\ G_{s,\theta,n,1} & G_{s,\theta,n,2} & \cdots & G_{s,\theta,n,n} \end{bmatrix}$$

For each of these grid cells, the mean $\bar{C}_{s,\theta}$ is calculated:

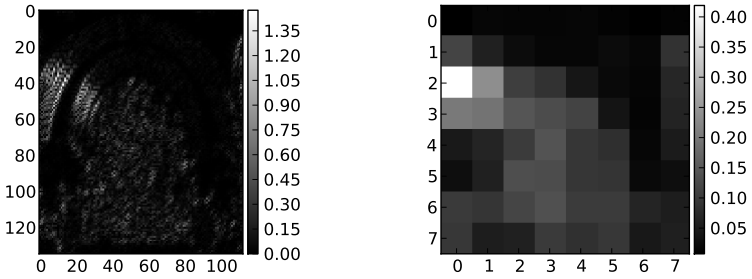
$$\begin{aligned} \bar{C}_{s,\theta} &= \begin{bmatrix} mean(G_{s,\theta,1,1}) & mean(G_{s,\theta,1,2}) & \cdots & mean(G_{s,\theta,1,n}) \\ mean(G_{s,\theta,2,1}) & mean(G_{s,\theta,2,2}) & \cdots & mean(G_{s,\theta,2,n}) \\ \vdots & \vdots & \ddots & \vdots \\ mean(G_{s,\theta,n,1}) & mean(G_{s,\theta,n,2}) & \cdots & mean(G_{s,\theta,n,n}) \end{bmatrix} \\ &= \begin{bmatrix} \bar{c}_{s,\theta,1,1} & \bar{c}_{s,\theta,1,2} & \cdots & \bar{c}_{s,\theta,1,n} \\ \bar{c}_{s,\theta,2,1} & \bar{c}_{s,\theta,2,2} & \cdots & \bar{c}_{s,\theta,2,n} \\ \vdots & \vdots & \ddots & \vdots \\ \bar{c}_{s,\theta,n,1} & \bar{c}_{s,\theta,n,2} & \cdots & \bar{c}_{s,\theta,n,n} \end{bmatrix} \end{aligned}$$

3.2.1 *Global Features*

MEAN The global approach to signature extraction simply takes the family of matrices $\bar{C}_{s,\theta}$ and concatenates them as the image signature.

3.2.2 *Local Features*

The local feature extraction methods used here follow the bag-of-features approach, that aims to represent an image using a set of local feature



(a) Curvelet coefficients (b) Means on an 8×8 grid

Figure 9: The curvelet coefficients for a specific scale and angle can be seen in (a). In (b) the response image has been subdivided into 8×8 cells and the mean value of each cell is shown.

descriptions or "visual words" similar to what was described by Sivic et al. [44].

3.2.2.1 *Sampling*

Both of the feature vector extraction methods described above use dense, overlapping sampling of a grid of mean values. By using $m \times m$ windows, the small-scale spatial relationship between features can be captured. The overlap helps to avoid misinterpretation of features on grid boundaries that would occur with dense, non-overlapping sampling. Each window encodes the geometric relationships between cells within a neighborhood. Evaluations by Nowak et al. [33] have shown that random sampling, which dense sampling is a special case of, outperforms keypoint-based sampling for large enough numbers of samples. Dense sampling on grids has previously been successfully used by Lazebnik et al. [22] [23]. The R-HOG descriptor [12] also uses a dense grid for sampling with overlapping windows to improve matching performance.

PMEAN Continuing from the set of matrices $\bar{C}_{s,\theta}$, this algorithm densly samples each matrix by sliding a window of size $m \times m, m < n$ across it (Figure 10). This results in $(n-m+1)^2$ parts $\bar{W}_{s,\theta,u,v}$ with $u, v \in \{1, \ldots, n-m+1\}$:

$$\bar{W}_{s,\theta,u,v} = \begin{bmatrix} \bar{c}_{s,\theta,u,v} & \bar{c}_{s,\theta,u,v+1} & \cdots & \bar{c}_{s,\theta,u,v+m} \\ \bar{c}_{s,\theta,u+1,v} & \bar{c}_{s,\theta,u+1,v+1} & \cdots & \bar{c}_{s,\theta,u+1,v+m} \\ \vdots & \vdots & \ddots & \vdots \\ \bar{c}_{s,\theta,u+m,v} & \bar{c}_{s,\theta,u+m,v+1} & \cdots & \bar{c}_{s,\theta,u+m,v+m} \end{bmatrix}$$

For each pair (u, v) these matrices are concatenated in a consistent way and stored as the feature vectors of the image. That way, the algorithm derives $(n-m+1)^2$ vectors of length $N_s \cdot N_{\theta_s} \cdot m^2$ from each image, where N_{θ_s} is the number of angles at the scale s.

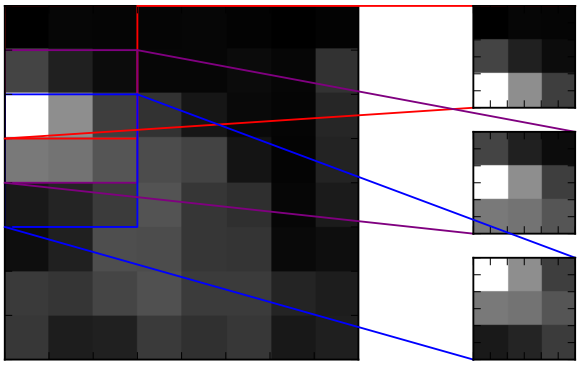


Figure 10: The 8×8 mean coefficient grid $\bar{C}_{s,\theta}$ is densely sampled using windows of 3×3 grid cells to produce 36 patches per scale and angle.

PMEAN2 In most parts, this variant is identical with the previously described PMEAN algorithm with the exception of the final signature feature vectors. Instead of concatenating all vectors $\bar{W}_{s,\theta,u,v}$, each scale is handled separately. Therefore, each combination of (u, v, s) produces a vector of length $N_{\theta_s} \cdot m^2$.

3.2.2.2 *Clustering and Vector Quantization*

CLUSTER The set of feature vectors extracted from the images are numerous and large. In order to create meaningful image signatures from these vectors, the whole set is condensed into a dictionary using k-means clustering. As already discussed in Section 2.2.2.3, the goal thereof is to derive a dictionary of predefined size that contains the visual words corresponding to the most discriminating features of the images in the image database. A universally optimal size for the codebook does not seem to exist, as Nowak et al. [33] observed an increase in accuracy up to 1000 words, but overfitting for some sampling algorithms beyond that. At the same time Yang et al. [53] report optimal sizes of 20000 to 80000 depending on the image database. The findings of Eitz et al. [15] agree with Nowak et al. in that a size of 1000 visual words was optimal for sketches, which is why that size was chosen for the following evaluation.

VQ The dictionary generated in the clusting step is then used to quantize the feature vectors extracted by the previous PMEAN and PMEAN2 procedures and record their number in a signature vector as explained in Section 2.2.2.3. To improve the degree to which this vector describes the image it was generated from, its components are then weighted. The TF-IDF statistic detailed in Section 2.2.3.1 is used to calculate a factor for each value that diminishes common, non-distinguishing visual words and boosts the importance of rare or unique words.

3.3 RANKING

As the last step of the pipeline, ranking operates on the signatures produced by prior extraction steps. It outputs a sequence of database images, sorted by the distance to the query image in ascending order. The distance can be determined using various metrics and similarity measures, which have been detailed in Section 2.2.3.2. This section lists the metrics used in the experiments and labels them for later reference.

L_2 The simplest and most widely used distance metric calculates the euclidean distance between the query image's signature and each database image's signature.

COS Following the definition of the cosine distance in Section 2.2.3.2, this calculates the distance of two signatures vectors via the angle between them.

HI When comparing histograms of features, the histogram intersection measure s_{HI} has been shown to be superior to the euclidean distance in most of the cases [51]. It has the added benefit of allowing for partial matches in a signature. As can be seen from the definition in Section 2.2.3.2, the result lies within $[0, 1]$ with 1 being a perfect match. Therefore, $1 - s_{HI}$ is used as a distance value to sort the result list.

HIB The binary variant of the histogram intersection measure converts the bin counts into a boolean representation, where a 1 indicates the presence of a codeword irrespective of the actual count, and a 0 denotes a codeword's absence. The resulting sequence of zeros and ones is then treated like in the HI measure.

EMD The Earth Mover's Distance is solved using a simplex algorithm variant, which has an exponential worst case complexity. That makes it computationally more expensive than the linear complexity measures described previously.

4 EXPERIMENTAL RESULTS

This chapter will present the benchmarking methods as well as the specific processing pipelines constructed from the steps explained in Chapter 3. A detailed description of the experimental results for each pipeline variation will follow. Many pipelines were tested with varying parameters, some of which are common to all experiments and some of which are specific to the implementation.

4.1 BENCHMARKING METHOD

A usual way to evaluate the performance of retrieval systems is to calculate the ratio of true positive and false positive matches and visualize it in a receiver operating characteristic (ROC) curve. While that approach is well suited for benchmarking binary decision algorithms, it is not appropriate for retrieval problems, that do not feature a well-defined "correct" solution. An alternative approach is looking at the recall and precision characteristics defined as

$$\text{recall} = \frac{\text{number of correct positive results}}{\text{total number of positives}}$$
$$\text{precision} = \frac{\text{number of correct positive results}}{\text{total number of results}}$$

Even though this metric works better for algorithms that return a set of results, it is still based on the notion of a "positive match", which requires an a priori classification of the benchmark data. For intra-domain evaluations the sketch dataset created by Eitz et al. [14] is used. It consists of 20.000 hand-drawn sketches obtained via crowd-sourcing, that are evenly divided into 250 categories. To speed up computations, 50 of those categories are chosen to derive precision-recall statistics. From each category, an image is randomly selected as the query and the rest is used as positive results. In this case, both the query images and the database images are from the sketch domain, so the effectiveness of the retrieval process without preprocessing biases can be examined.

Since sketch-based image retrieval systems are most likely to be used in interactive search applications of some form, it is desirable to assess the performance in relation to the results a human would achieve. Therefore the benchmark used to evaluate the retrieval pipelines in cross-domain applications corresponds to the method described by Eitz et al. [15], in which the authors create a benchmark dataset and perform a user study with 28 participants to define "ground truth" rankings. The dataset is divided into 31 groups of one sketch and 40 images

each. Participants ranked the 40 images within each group by assigning scores indicating the similarity to the corresponding sketch in a controlled study environment. Each sketch/image pair's final ground truth ranking is calculated as the mean of the scores assigned by all participants.

To compare a ground truth ranking $x = (x_1, x_2, \ldots, x_n)$ to a ranking $y = (y_1, y_2, \ldots, y_n)$ produced by a retrieval system, the Kendall rank correlation coefficient τ_B is used. It measures the similarity of the orderings by grouping all pairs $p_{i,j} = \{(x_i, y_i), (x_j, y_j)\}$, $i, j \in \{1, \ldots, n\}$ into 5 sets:

$$
\begin{aligned}
p_{i,j} &\in C && \text{if } x_i < x_j \text{ and } y_i < y_j \\
p_{i,j} &\in D && \text{if } x_i < x_j \text{ and } y_i > y_j \\
p_{i,j} &\in T_x && \text{if } x_i = x_j \text{ and } y_i \neq y_j \\
p_{i,j} &\in T_y && \text{if } x_i \neq x_j \text{ and } y_i = y_j \\
p_{i,j} &\in T_{xy} && \text{if } x_i = x_j \text{ and } y_i = y_j
\end{aligned}
$$

From that, the correlation value τ_B in the interval $[-1, 1]$ can be calculated as

$$\tau_B = \frac{|C| - |D|}{\sqrt{(|C| + |D| + |T_x|)(|C| + |D| + |T_y|)}}.$$

The higher τ_B is, the more pairs in x and y have a similar ordering. Since the values are only compared within each ranking, the result is independent of each rankings' scaling, making it ideal for comparison of different distance metrics.

4.2 CROSS-DOMAIN RESULTS

Some of the variations below include a Canny edge detector as a preprocessing step. The parameter σ determines the size of the Gaussian kernel used by the Canny algorithm. A value of $\sigma = 1.5$ was determined to be appropriate for the image dataset used in the benchmark. Values larger than $\sigma = 2$ tend not to detect any edges, while smaller values produced more "false" edges resulting from noise in the images.

The SEGMENT preprocessing step could unfortunately not use the same 1024×768 pixel images as inputs as the other preprocessors, because the implementation provided by Arbelaez et al. [2] could not be run on any machine available due to extreme memory requirements. Therefore, the images in the database are rescaled to 320×240 pixels for the evaluation of this algorithm. This might lead to loss of small features and puts limits on some parameters like the grid size G.

All pipeline variants utilize the FDCT to extract curve information. The number of scales and the number of angles at the coarsest scale will be called N_s and N_θ respectively. Based on experimentation and other publications using the curvelet transform [27] [19], 4 scales and 12 angles

are used in almost all cases, since larger values did not consistently lead to better results. The size of the codebook is set at 1000 visual words, as values beyond that did not show improvements in evaluations performed by Nowak at al. [33]. Eitz et al. [15] reported similar optimal values of 500 to 1000 visual words and attributed the low number to the sparsity of edge-like features in sketches.

Throughout the correlation coefficient graphs in the following sections, the results obtained by Eitz et al. [15] for the HoG, Spark and SHoG descriptors are included.

The first two sections present the resulting mean τ_B values obtained using the rank correlation method described above, grouped by preprocessing steps and sampling method. Afterwards, the influence of parameter variations and the distribution of results within selected pipelines are examined.

4.2.1 *Global Features*

The processing steps applied to the database images and the query images in the pipelines based on global features are almost identical, with the exception of the CANNY, SOBEL and SEGMENT steps for the database images. The curvelet responses are averaged on a grid and the features are ranked using the L_2 and the COS distance measures.

LUMA+MEAN The most straightforward combination of processing steps consists of a LUMA input image, on which the means of $G \times G$ grid cells is computed for each scale and angle (Figure 11). Varying G determines the feature size that can be encoded best. Setting $G = 12$ seems to yield the best correlation, although the advantage over $G = 8$ and $G = 16$ is below 0.01. The distance measure COS outperforms the L_2 measure (Table 1). A possible explanation would be that the COS measure normalizes the magnitude of the feature vector as discussed in Section 2.2.3.2.

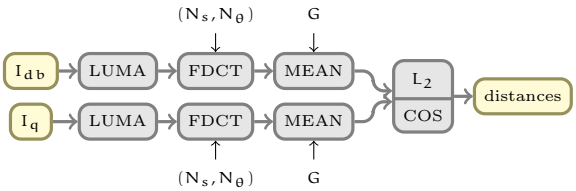


Figure 11: Global LUMA+MEAN Pipelines

N_s	N_θ	G	Metric	Mean τ_B correlation coefficient
4	12	4	COS	0.177
4	12	8	L_2	0.14
4	12	8	COS	0.183
4	12	12	L_2	0.14
4	12	12	COS	0.191
4	12	16	COS	0.186

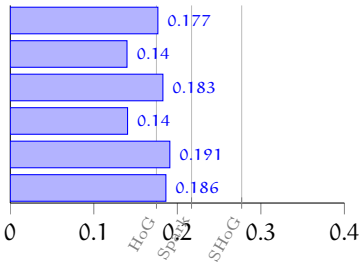


Table 1: Global LUMA+MEAN Results

CANNY+MEAN In addition to reading the images like in the previous LUMA+MEAN configuration, this pipeline applies a CANNY processing step to the database images in an attempt to bring the query and database image domains closer together (Figure 12). Again, the COS distance measure produces the best rankings (Table 2). Surprisingly, the Canny edge detector does not lead to increased performance in comparison to the plain LUMA preprocessing step.

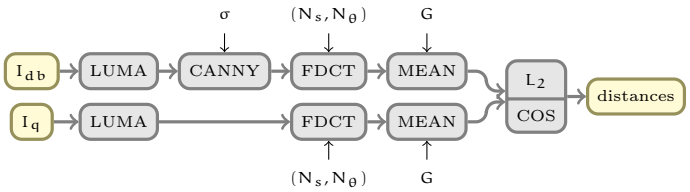


Figure 12: Global CANNY+MEAN Pipelines

N_s	N_θ	G	σ	Metric	Mean τ_B correlation coefficient
4	12	4	1.5	COS	0.179
4	12	8	1.5	L_2	0.095
4	12	8	1.5	COS	0.188
4	12	12	1.5	L_2	0.096
4	12	12	1.5	COS	0.184
4	12	16	1.5	COS	0.182

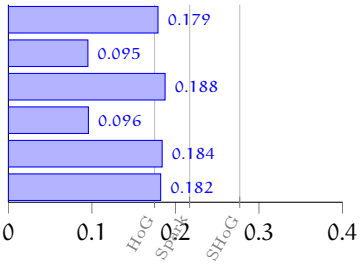


Table 2: Global CANNY+MEAN Results

SOBEL+MEAN The SOBEL step used in this variant also attempts to bring the database images into the sketch domain (Figure 13). The results are slightly better than with the CANNY preprocessor (Table 3).

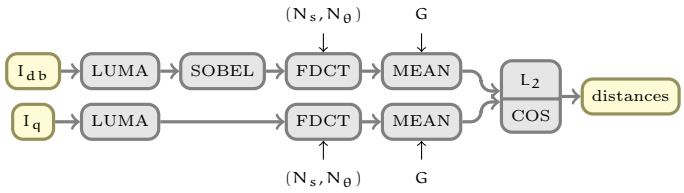


Figure 13: Global SOBEL+MEAN Pipelines

N_s	N_θ	G	Metric	Mean τ_B correlation coefficient
4	12	8	L_2	0.15
4	12	8	COS	0.19
4	12	12	L_2	0.152
4	12	12	COS	0.2

Table 3: Global SOBEL+MEAN Results

SEGMENT+MEAN With the gPb contour detector in the SEGMENT step to find edges in the database images (Figure 14), the L_2 distance metric produces results comparable to the COS metric in the CANNY+MEAN variant (Table 4). Unlike in the other cases, the COS distance measure performs worse than the L_2 metric.

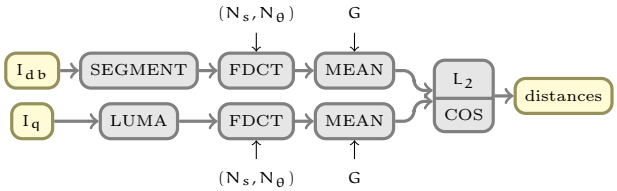


Figure 14: Global SEGMENT+MEAN Pipelines

N_s	N_θ	G	Metric	Mean τ_B correlation coefficient
4	12	4	L_2	0.188
4	12	4	COS	0.168
4	12	8	L_2	0.171
4	12	8	COS	0.169

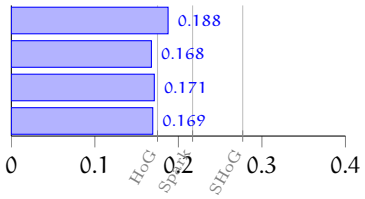


Table 4: Global SEGMENT+MEAN Results

4.2.2 *Local Features*

The local feature extraction pipelines utilize the same preprocessing steps as the ones based on global features. The curvelet transform remains unchanged as well. The curvelet coefficients are sampled using the PMEAN and PMEAN2 strategies to produce features from local patches. From these feature vectors, a codebook with 1000 entries is generated and the feature vectors from both the database images and the query images are quantized and weighted using the TF-IDF method. The metrics used for ranking are the generic cosine and L_2 distance measures as well as histogram intersection and the earth mover's distance.

LUMA+PMEAN(2) Analogous to the global LUMA+MEAN variant, this pipeline configuration directly uses the LUMA image representations as input for the curvelet transform (Figure 15). For both, the PMEAN (Table 5a) and PMEAN2 (Table 5b) sampling method, histogram intersection and the cosine measures are far superior to the L_2 metric and even the EMD metric, although the results cannot compete with the global LUMA+MEAN version.

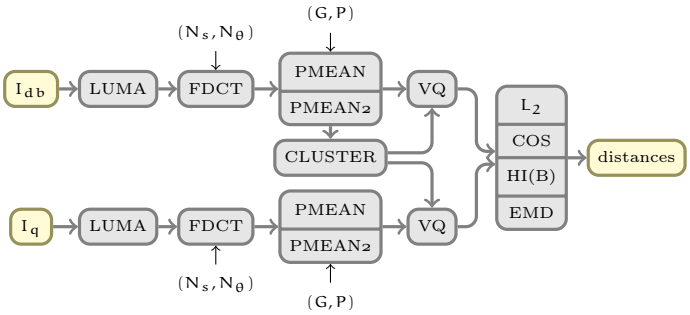
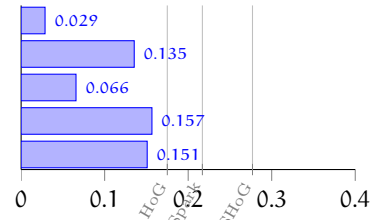


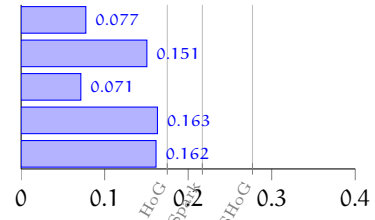
Figure 15: Local LUMA+PMEAN(2) Pipelines

N_s	N_θ	G	P	Metric	Mean τ_B correlation coefficient
4	12	8	3	L_2	0.029
4	12	8	3	COS	0.135
4	12	8	3	EMD	0.066
4	12	8	3	HI	0.157
4	12	8	3	HIB	0.151



(a) Local LUMA+PMEAN Results

N_s	N_θ	G	P	Metric	Mean τ_B correlation coefficient
4	12	8	3	L_2	0.077
4	12	8	3	COS	0.151
4	12	8	3	EMD	0.071
4	12	8	3	HI	0.163
4	12	8	3	HIB	0.162



(b) Local LUMA+PMEAN2 Results

Table 5: Local LUMA+PMEAN(2) Results

CANNY+PMEAN(2) Using the Canny edge detector and PMEAN sampling (Figure 16), the rank correlation coefficients for COS and HI exceed all previous results (Table 6a) with the highest value achieved being 0.22. Treating the coefficients on different scales separately using the PMEAN2 sampling method is inferior to PMEAN in this case.

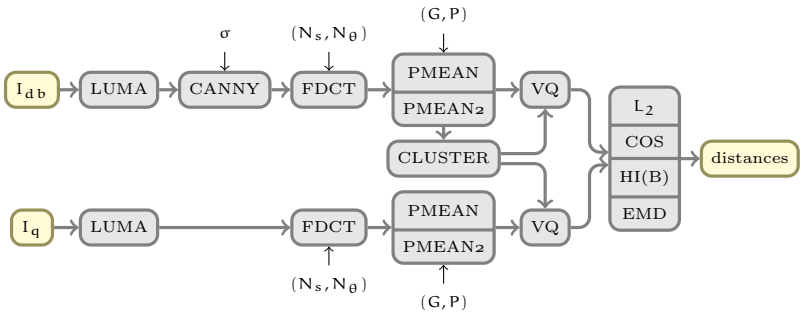
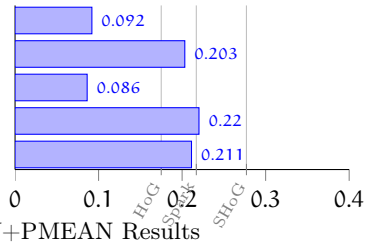


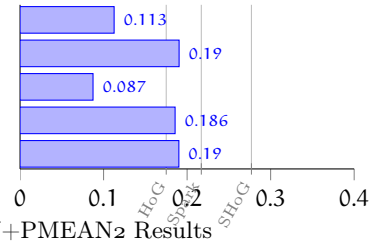
Figure 16: Local CANNY+PMEAN Pipelines

N_s	N_θ	G	P	σ	Metric	Mean τ_B correlation coefficient
4	12	8	3	1.5	L_2	0.092
4	12	8	3	1.5	COS	0.203
4	12	8	3	1.5	EMD	0.086
4	12	8	3	1.5	HI	0.22
4	12	8	3	1.5	HIB	0.211



(a) Local CANNY+PMEAN Results

N_s	N_θ	G	P	σ	Metric	Mean τ_B correlation coefficient
4	12	8	3	1.5	L_2	0.113
4	12	8	3	1.5	COS	0.19
4	12	8	3	1.5	EMD	0.087
4	12	8	3	1.5	HI	0.186
4	12	8	3	1.5	HIB	0.19



(b) Local CANNY+PMEAN2 Results

Table 6: Local CANNY+PMEAN(2) Results

SOBEL+PMEAN(2) Preprocessing the database images with the Sobel operator (Figure 17) results in slightly lower correlation coefficients (Table 7). The PMEAN2 sampling algorithm yields better results for the L_2 and EMD metrics, but worse for COS, HI and HIB (Table 7b).

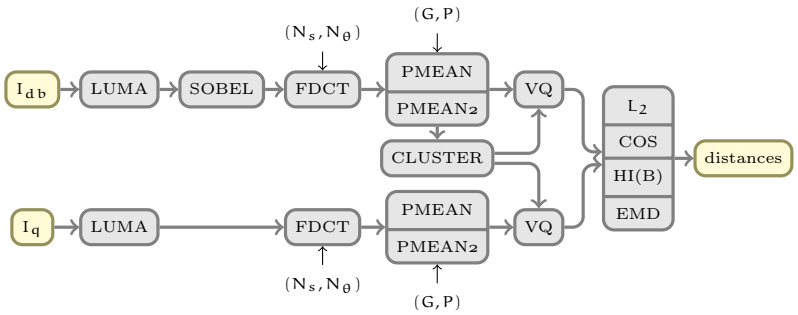
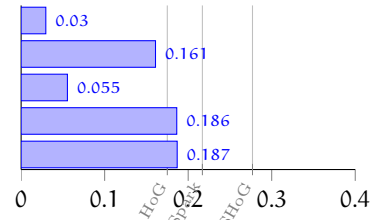


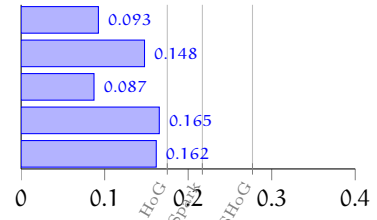
Figure 17: Local SOBEL+PMEAN(2) Pipelines

N_s	N_θ	G	P	Metric	Mean τ_B correlation coefficient
4	12	8	3	L_2	0.03
4	12	8	3	COS	0.161
4	12	8	3	EMD	0.055
4	12	8	3	HI	0.186
4	12	8	3	HIB	0.187



(a) Local SOBEL+PMEAN Results

N_s	N_θ	G	P	Metric	Mean τ_B correlation coefficient
4	12	8	3	L_2	0.093
4	12	8	3	COS	0.148
4	12	8	3	EMD	0.087
4	12	8	3	HI	0.165
4	12	8	3	HIB	0.162



(b) Local SOBEL+PMEAN2 Results

Table 7: Local SOBEL+PMEAN(2) Results

SEGMENT+PMEAN(2) The SEGMENT step (Figure 18) leads to similar, although slightly lower results than the Sobel operator (Table 8). Neither PMEAN nor PMEAN2 have a consistent advantage across the different distance measures.

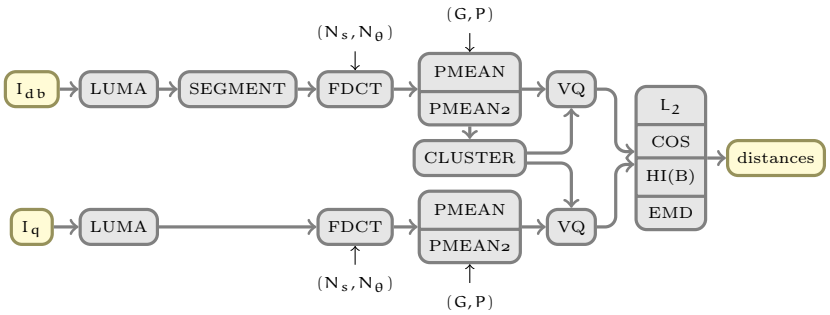
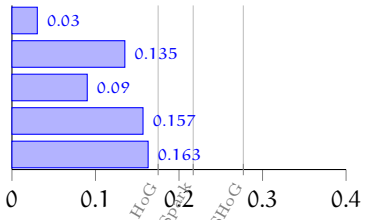


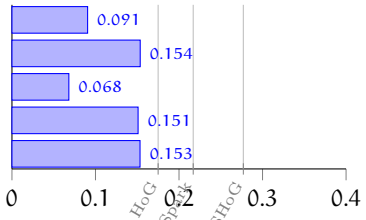
Figure 18: Local SEGMENT+PMEAN(2) Pipelines

N_s	N_θ	G	P	Metric	Mean τ_B correlation coefficient
4	12	8	3	L_2	0.03
4	12	8	3	COS	0.135
4	12	8	3	EMD	0.09
4	12	8	3	HI	0.157
4	12	8	3	HIB	0.163



(a) Local SEGMENT+PMEAN Results

N_s	N_θ	G	P	Metric	Mean τ_B correlation coefficient
4	12	8	3	L_2	0.091
4	12	8	3	COS	0.154
4	12	8	3	EMD	0.068
4	12	8	3	HI	0.151
4	12	8	3	HIB	0.153



(b) Local SEGMENT+PMEAN2 Results

Table 8: Local SEGMENT+PMEAN(2) Results

4.2.3 *Parameter Variations*

In an attempt to improve the results, the best performing configurations presented in the previous sections will be re-used with varying parameter values (Table 9). The following sections compare the results of changing the parameters N_θ, P, G and σ.

Preproc.	Sampling	G	P	σ	Metric	Mean τ_B correlation coefficient
CANNY	MEAN	8		1.5	COS	0.188
CANNY	PMEAN	8	3	1.5	HI	0.22
CANNY	PMEAN2	8	3	1.5	COS	0.19
LUMA	MEAN	12			COS	0.191
SOBEL	MEAN	12			COS	0.2
SOBEL	PMEAN	8	3		HIB	0.187

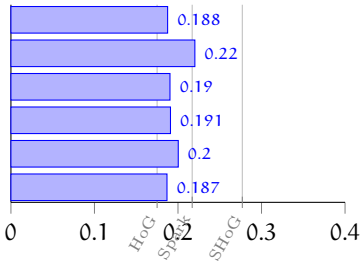


Table 9: Best Performing Configurations with default assumptions $N_s = 4$ and $N_\theta = 12$.

4.2.3.1 *Curvelet Angles*

The parameter N_θ controls the number of angles, the curvelet coronization is divided into (Section 2.3.3). Therefore, it determines how finely the angles of the lines are resolved and how sensitive to angular differences the descriptor is.

As indicated by literature and exploratory experiments, an angular subdivision of $N_\theta = 12$ appears to be optimal (Table 10). Larger values lead to worse, but constant results.

Preproc.	Sampling	N_s	N_θ	Metric	Mean τ_B correlation coefficient
CANNY	PMEAN	4	4	HI	0.183
CANNY	PMEAN	4	8	HI	0.207
CANNY	PMEAN	4	12	HI	0.22
CANNY	PMEAN	4	16	HI	0.207
CANNY	PMEAN	4	20	HI	0.208

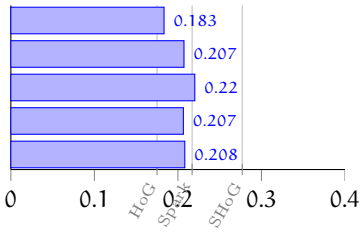


Table 10: Influence of N_θ on the results of CANNY+PMEAN for $G = 8$, $P = 3$ and $\sigma = 1.5$.

4.2.3.2 *Grid and Patch Sizes*

All of the MEAN, PMEAN and PMEAN2 sampling methods use a regular grid to divide the curvelet coefficients into cells, in which the mean of the coefficients is calculated. Using a small number G of subdivisions means that smaller features might vanish within a large grid cell, unable to influence the mean value. A finer subdivision allows for smaller features to be represented at the risk of cutting apart larger features that lie on the grid lines. In addition to G, the local sampling methods PMEAN and PMEAN2 are influenced by the number of grid cells that make up a patch. As explained in Section 3.2.2, a patch captures the geometric relationships within a $P \times P$ neighborhood of cells. It thus defines an upper limit on the size of a feature that can be represented atomically. The results (Table 11) indicate, that a ratio of $\frac{P}{G} \approx \frac{1}{3}$ lead to a locally optimal solution. This means, that features and their local composition in a neighborhood of about $\frac{1}{3}$ of the image's width and height are best suited to discriminate the images. This is similar to the 25% optimum determined by Eitz et al. [15].

Preproc.	Sampling	G	P	Metric	Mean τ_B correlation coefficient
CANNY	PMEAN	5	2	HI	0.177
CANNY	PMEAN	6	2	HI	0.208
CANNY	PMEAN	8	2	HI	0.184
CANNY	PMEAN	8	3	HI	0.22
CANNY	PMEAN	8	4	HI	0.192
CANNY	PMEAN	12	3	HI	0.205
CANNY	PMEAN	12	4	HI	0.217

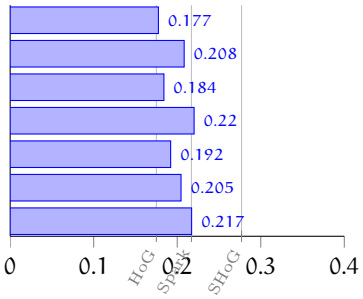


Table 11: Influence of grid parameters P and G on the results for $N_s = 4$, $N_\theta = 12$ and $\sigma = 1.5$.

4.2.3.3 *Canny Sigma*

In the CANNY preprocessing step, the parameter σ for the Gaussian smoothing kernel can have a potentially large influence. It controls the spread of the Gaussian distribution used for smoothing before the edge detection takes place. Larger values lead to more smoothing, which

makes the process less dependent on image noise, but may cause the loss of important edge information. The value $\sigma = 1.5$ yields the best correlation coefficients. Values larger than 2 tended to prevent any edge detection in images of the benchmark dataset. Compared to the other parameters though, the influence appears to be small except for the extreme values.

Preproc.	Sampling	σ	Metric	Mean τ_B correlation coefficient
CANNY	PMEAN	0.5	HI	0.193
CANNY	PMEAN	1.0	HI	0.214
CANNY	PMEAN	1.5	HI	0.22
CANNY	PMEAN	2.0	HI	0.218
CANNY	PMEAN	2.5	HI	0.187

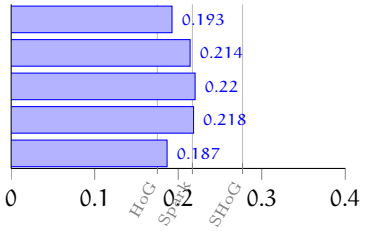


Table 12: Influence of the canny smoothing parameter σ on the results for $N_s = 4$, $N_\theta = 12$, $G = 8$ and $P = 3$.

4.2.4 *Result Distribution*

Each correlation value in the above graphs is the mean of the correlation values for 32 benchmark query images with a ground truth ranking of 40 result images. Calculating the mean correlation per query image for the best performing configurations shown in Table 9 shows, that the retrieval pipelines deal quite well with most image categories (Figure 19). At the same time the correlation is almost zero for two of the categories and even significantly below zero for two more across all pipeline configurations. This suggests that those query images or image sets have characteristics, that confound the algorithms. Figure 20 shows the query images with correlations below zero and examples from the corresponding image set.

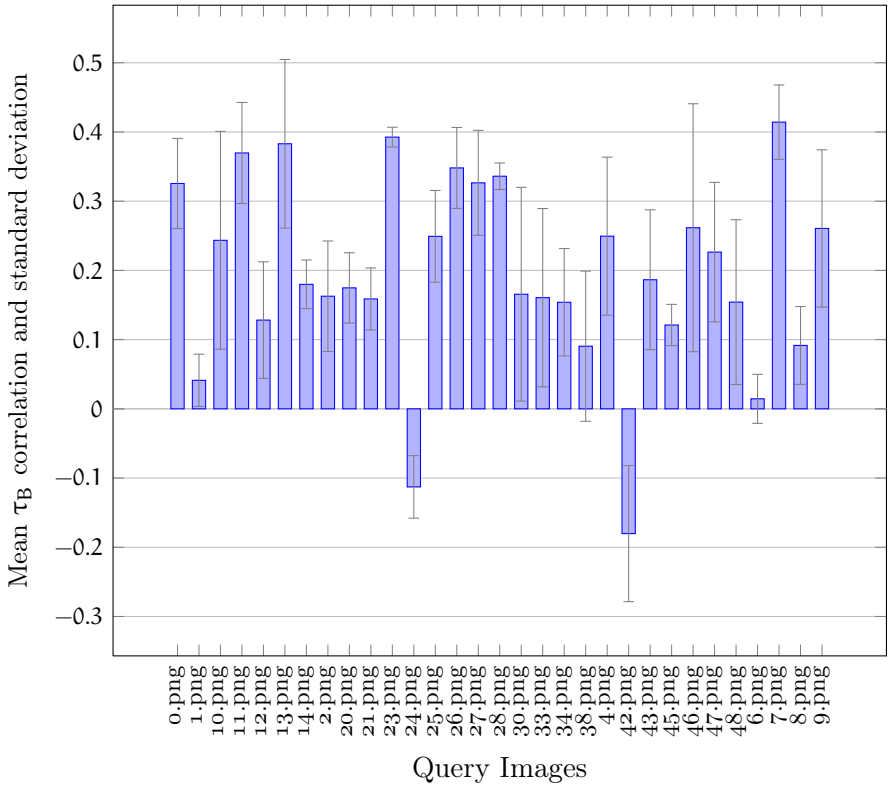


Figure 19: The distribution of mean correlations for the configurations of Table 9 shows drastically worse performance for some queries.

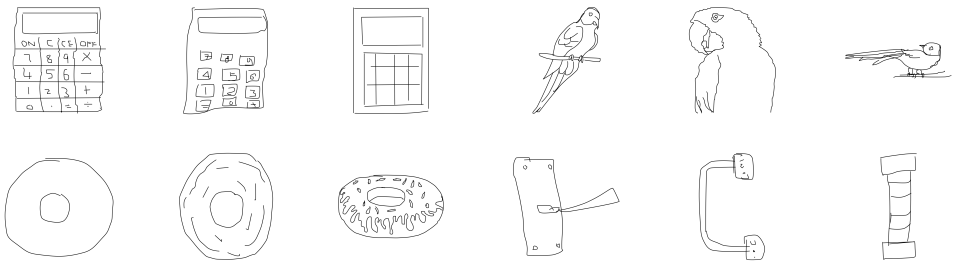
Figure 20: Outlier query images `24.png` (top) and `42.png` (bottom) with two example responses.

4.3 INTRA-DOMAIN RESULTS

As described in Section 4.1, the precision-recall benchmark on a dataset consisting purely of sketches is meant to remove the edge detection preprocessing steps as possible biases. The pipeline configurations used are the global LUMA+MEAN variant (Figure 11) with the L2 and COS distance measures and the local LUMA+PMEAN and LUMA+PMEAN2 pipelines (Figure 15). Based on the results in Section 4.2, a grid cell number of $G = 12$ and the cosine distance measure are used for the global pipelines. The local LUMA+PMEAN and LUMA+PMEAN2 variants are tested with $G = 8$ and a patch size of $P = 3$ in combination with the histogram intersection distance metric HI. The size of the codebooks remains at 1000 visual words. In all cases the Fast Discrete Curvelet Transform uses the parameters $N_s = 4$ and $N_\theta = 12$.

As the graphs in Figure 22 show, all four descriptors exhibit a large variation of precision values across the different categories. The mean average precision shows a small advantage of the global descriptors with $MAP(Q_{MEAN+L2}) = 0.139$ and $MAP(Q_{MEAN+COS}) = 0.150$ over the local variants with $MAP(Q_{PMEAN+HI}) = 0.129$ and $MAP(Q_{PMEAN2+HI}) = 0.120$ respectively.

Figure 23 breaks the results down and displays the average precision for each category. Here, it is apparent, that both global and local descriptors deal well with some categories like "calculator", "pear" or "donut", while the average precision values for the "parrot" and "door handle" categories stay well below 0.1. Figure 21 displays a few examples from two successful categories and two "difficult" categories. This is in line with the observations made by the creators of the benchmark dataset [14], that state that the images appear to be of varying difficulty for computational classification.



(a) (b)

Figure 21: The descriptors perform well on the categories "calculator" and "donut" (a), which exhibit a high degree of symmetry. The sketches from the categories "parrot" and "door handle" (b) do not have a uniform orientation or perspective or contain visually different representations of the object.

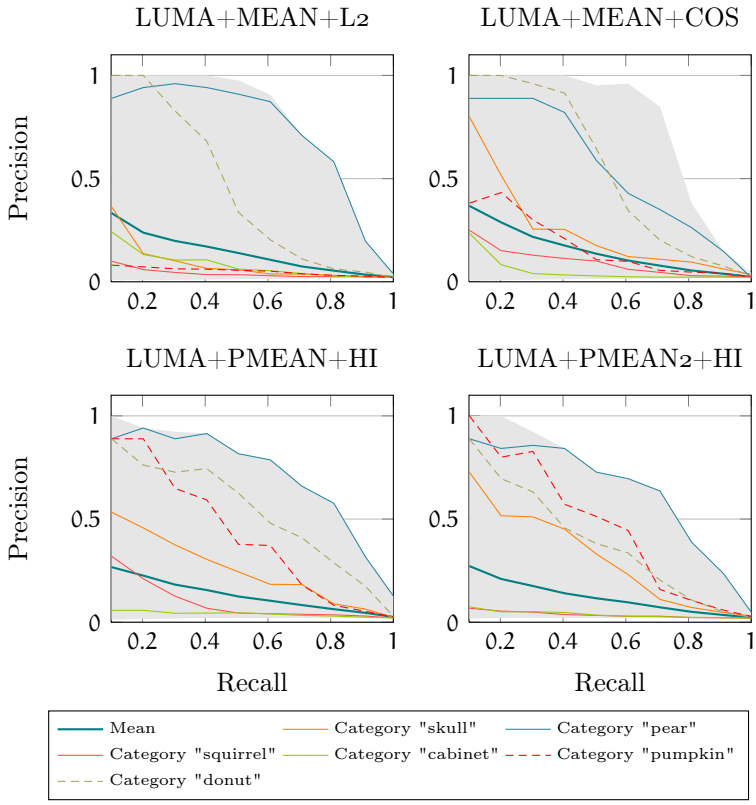


Figure 22: The graphs show the results of applying the global LUMA+MEAN with the L2 and COS distance measures and the local LUMA+PMEAN and LUMA+PMEAN2 pipelines to intra-domain retrieval of sketches. The areas indicate the overall spread of values across all categories while the line plots show results for selected categories.

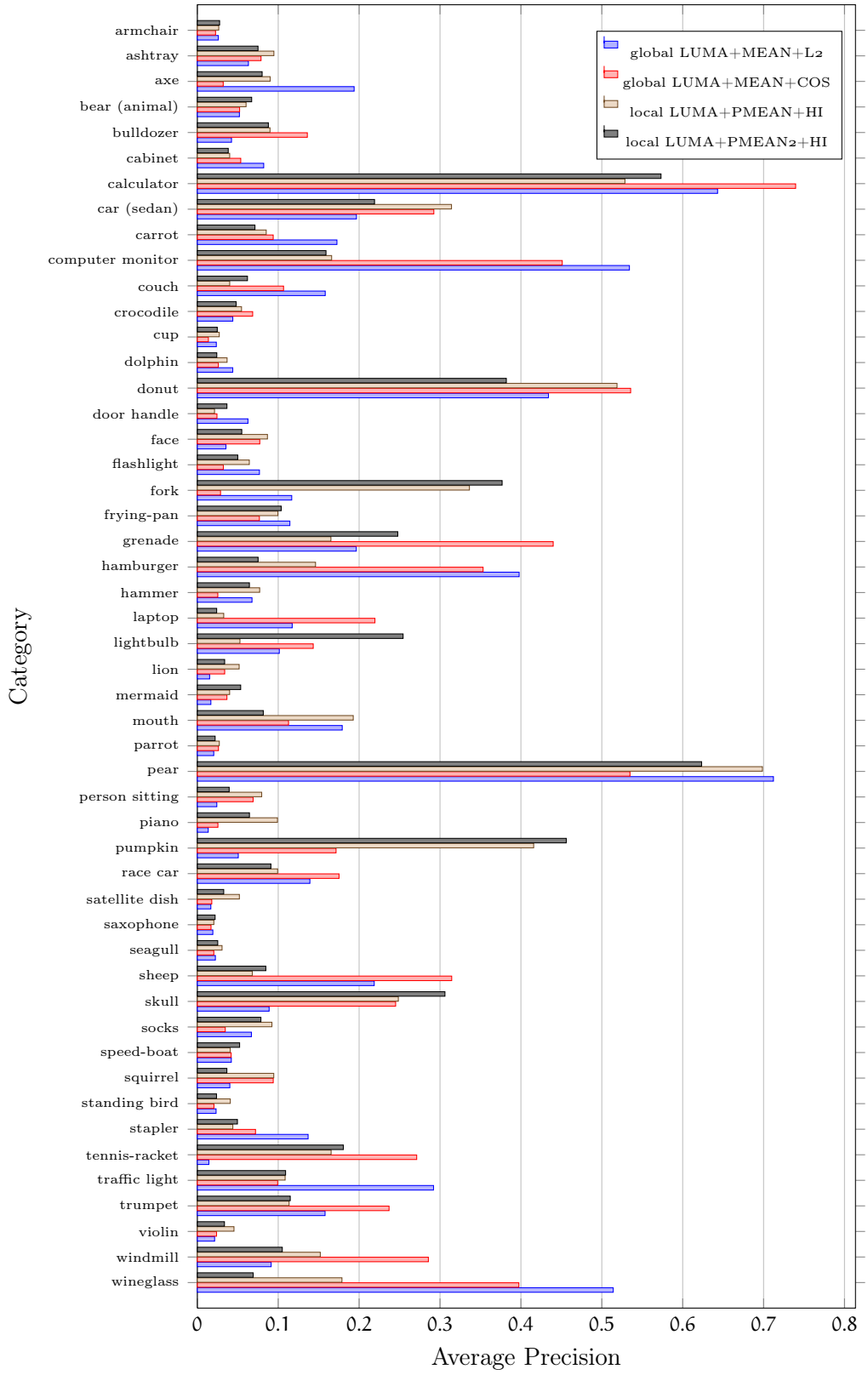


Figure 23: The average precision of the LUMA+MEAN, LUMA+PMEAN and LUMA+PMEAN2 pipelines broken down by query category.

5 DISCUSSION

Based on the results displayed previously, one can conclude that for cross-domain retrieval, an approach based on Canny edge detection and local features performs better than global approaches. Of the similarity measures used to compare the images' signatures, histogram intersection provides the most consistent performance. The advantage over the global descriptors, however, is not large, and poorly chosen parameter values can lead to the global descriptors outperforming the local variants.

When both the query image and the database images are sketches, the situation is reversed. The precision and recall statistics for the global LUMA+MEAN pipelines show a slight advantage over the local LUMA+PMEAN and LUMA+PMEAN2 variants. This is probably strongly influenced by the images used in the retrieval benchmark, as will be discussed below.

5.1 STRUCTURAL INFLUENCES

The intention underlying this thesis was to perform an evaluation of the applicability of the Fast Discrete Curvelet Transform to sketch-based image retrieval. It therefore seemed reasonable to otherwise choose methods and algorithms that are frequently used in this area of research in order to minimize the number of unknowns involved. The choice of edge detection steps includes established algorithms such as the Sobel and Canny operators as well as the recently published gPb contour detector [2]. The simple sampling methods are designed based on prior research, that suggests, that keypoint-based sampling barely provided any advantages at the cost of increased processing and complexity [33]. Finally, the clustering and ranking process is constructed using the often-used k-means clustering algorithm, several established distance metrics and the TF-IDF weighting scheme, that has been successfully applied to information retrieval for some time.

Considering the large differences between the image domains used in the cross-domain evaluation, it is not surprising that the most successful pipeline configurations involve edge detecting preprocessing steps. But even without such steps, that try to bridge the gap between the query and database image domains, the Curvelet transform on plain images produces viable results in some configurations. For example, the global LUMA+MEAN pipeline (Figure 11) combined with the cosine similarity measure performs almost as good as the best global configuration

(Table 1). This indicates that the scale-specificity enables the Curvelet transform to extract meaningful edges from photographs on its own.

5.2 PARAMETER CHOICES

Since many of the processing steps are based on commonly-used algorithms, literature already presented reasonable starting values for the evaluation. As the experiments in Section 4.2.3 show, the initial values already produce competitive results. The best parameter values seem to strike a balance between losing information due to small resolution and becoming overly sensitive to noise or unrelated image background. For local sampling methods, a neighborhood size of $\frac{1}{3}$ of the image dimensions repeatedly performs best. A value of $\sigma = 1.5$ for the Gaussian blur of the Canny edge detector appears to be suitable to extract the edges that correspond to a human sketch of the object or scene. The advantage of an angluar resolution larger than $N_\theta = 12$ for the curvelet transform is probably limited by to the poor accuracy of hand-drawn sketches.

5.3 BENCHMARK DATASET INFLUENCES

Before making generalized statements based on the results above, several properties of the benchmark datasets must be taken into account. In particular, the fact that the global descriptor outperforms the local descriptor in the intra-domain benchmark can probably be attributed to the nature of the images. As stated by Eitz et al. [14], the sketches have been scaled to a fixed size of the bounding box and centered in a 256×256 pixel image. This constitutes a form of preprocessing that is suited to bypass the lack of translation invariance of the global descriptor. While that invariance would be expected to be an advantage of the local descriptors in general, it can be a confounding factor in this case, which may contribute to the relatively poor results. Whether translation invariance beyond the slight fuzzyness introduced by the sampling method would be desirable at all, clearly depends on the images involved and the expectations of the user. If the user sketches a scene in order to find photographs with similar composition, disregarding the location of drawn features would be counterproductive.

So while the intra-domain dataset's normalization of the images seems to favor global descriptors, the cross-domain dataset mixes query intents and image types. Some sketches and images depict single objects with varying degrees of context and background, while others capture whole scenes or objects within a larger composition. Differences in descriptor performance within such a diverse image set are to be expected. And indeed, as shown in Figure 19, there are a few query images in the cross-domain benchmark, for which the proposed solutions consistently do not lead to good rankings.

But even in the intra-domain evaluation (Figure 22) there are significant differences in descriptor performance between different sketch categories. Reasons for that might be that the sketches or images contain too much distracting patterns or that the datasets contain several distinct representations of the same object, that are visually not very similar. When examining the results for each category (Figure 23), it is noticeable that the descriptors perform best for categories containing images with a high degree of symmetry and barely any variation in the orientation or perspective.

The overall picture is, that each set of pipeline components and parameter values might only be suitable for a limited range of image types. When the system is designed for a more specific purpose, its performance can probably exceed the results shown above by a large margin.

6 CONCLUSION

This thesis examined the suitability of image descriptors for sketch-based image retrieval. In particular, the theoretical background of the Fast Discrete Curvelet Transform was explained and its use in combination with established retrieval system architectures was demonstrated. Querying a database of photographs using hand-drawn sketches was used as a representative example of cross-domain image retrieval.

The theoretical discussion in literature of the Curvelet transform's advantages over related algorithms such as the Gabor filter indicate, it might be especially well suited for sketch-based image retrieval. To evaluate this, common structural features of retrieval systems were examined and based on that several processing pipeline variations, that utilize the Fast Discrete Curvelet Transform, were implemented. This included descriptors relying on global image information as well as ones using local neighborhoods to extract image features. The performance of these pipelines was measured using both a cross-domain benchmark and an intra-domain benchmark.

The results showed, that the curvelet-based descriptors can compete with other descriptors described in literature, although the implementations used in this paper did not exceed the best among those. For cross-domain retrieval a local feature descriptor, that performed edge detection on the photographs, was most successful. The intra-domain evaluation on the other hand resulted in a global descriptors showing slightly better performance. In both cases the advantage of one type over the other was small and might be attributed to limitations of the benchmark dataset. Also noticeable in both cases was a broad distribution of the resulting values for different query images and categories, that was consistent across several descriptors. This could be an indication, that the semantic and sensory gaps were too large for the algorithms to overcome, for example when the photographs or sketches contained large amounts of clutter or a category included ambiguous representations of an object.

6.1 FUTURE RESEARCH

As discussed above, the descriptors performed far below average for a few queries and categories. Looking at why the algorithms behaved so differently with specific images or finding common properties in the outlier images could give precious hints for making the descriptors applicable more generally. In the same way, it could provide insights into

which descriptors would be best for specific applications, such as medical image analysis or industrial quality control.

Another possible improvement could be to use algorithmically or heuristically determined values for the parameters, which were static in the above benchmarks, on a per-image basis. It would even be thinkable to dynamically exclude or include certain preprocessing steps depending on characteristics of the image and features in the database. If a quick analysis determines that a sobel operator extracts too few edges, for example, it could be replaced by a Canny edge extraction.

To make the retrieval process more robust despite the varying nature of the images, a database that contains several signatures extracted using different descriptors for each image can enable a CBIR system to create multiple rankings. This would be especially useful in scenarios, in which the user can interactively manipulate the query preferences to express the intent behind the query.

The results presented above show that it can be worth considering the FDCT when building such systems in the future.

BIBLIOGRAPHY

[1] A.E. Abdel-Hakim and A.A. Farag. "CSIFT: A SIFT Descriptor with Color Invariant Characteristics." In: *IEEE Conference on Computer Vision and Pattern Recognition*. Vol. 2. 2006, pp. 1978–1983.

[2] P. Arbelaez et al. "Contour detection and hierarchical image segmentation." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33.5 (2011), pp. 898–916.

[3] P. Arbeláez et al. "From contours to regions: An empirical evaluation." In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2009, pp. 2294–2301.

[4] R. Baeza-Yates, B. Ribeiro-Neto, et al. *Modern information retrieval*. Vol. 82. Addison-Wesley New York, 1999.

[5] H. Bay et al. "Speeded-up robust features (SURF)." In: *Computer Vision and Image Understanding* 110.3 (2008), pp. 346–359.

[6] E. J. Candes. "Ridgelets: theory and applications." PhD thesis. Stanford University, 1998.

[7] E. J. Candes and D. L. Donoho. *Curvelets: A surprisingly effective nonadaptive representation for objects with edges*. Tech. rep. Curves and Surfaces. DTIC Document, 2000.

[8] E. J. Candes and D. L. Donoho. "New tight frames of curvelets and optimal representations of objects with piecewise C2 singularities." In: *Communications on pure and applied mathematics* 57.2 (2004), pp. 219–266.

[9] E. Candes et al. "Fast discrete curvelet transforms." In: *Multiscale modeling and simulation* 5.3 (2006), pp. 861–899.

[10] J. Canny. "A computational approach to edge detection." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 6 (1986), pp. 679–698.

[11] G. Csurka et al. "Visual categorization with bags of keypoints." In: *Workshop on statistical learning in computer vision, European Conference on Computer Vision*. 2004, pp. 1–22.

[12] N. Dalal and B. Triggs. "Histograms of oriented gradients for human detection." In: *IEEE Conference on Computer Vision and Pattern Recognition*. Vol. 1. 2005, pp. 886–893.

[13] Y. Deng et al. "An efficient color representation for image retrieval." In: *IEEE Transactions on Image Processing* 10.1 (2001), pp. 140–147.

[14] M. Eitz, J. Hays, and M. Alexa. "How do humans sketch objects?" In: *ACM Transactions on Graphics* 31.4 (2012), 44:1–44:10.

[15] M. Eitz et al. "Sketch-based image retrieval: benchmark and bag-of-features descriptors." In: *IEEE Transactions on Visualization and Computer Graphics* 17.11 (Nov. 2011), pp. 1624–1636.

[16] G. C. Feng, P. C. Yuen, and D. Q. Dai. "Human face recognition using PCA on wavelet subband." In: *Journal of Electronic Imaging* 9 (2000), pp. 226–233.

[17] R. Fergus et al. "Learning object categories from Google's image search." In: *IEEE International Conference on Computer Vision*. Vol. 2. 2005, pp. 1816–1823.

[18] C. Fowlkes, D. Martin, and J. Malik. "Learning affinity functions for image segmentation: Combining patch-based and gradient-based approaches." In: *IEEE Conference on Computer Vision and Pattern Recognition*. Vol. 2. 2003, pp. 54–61.

[19] T. Guha and Q. M.J Wu. "Curvelet Based Feature Extraction." In: *Face Recognition*. InTech, 2010, pp. 35–42.

[20] I. Guyon et al. "Gene selection for cancer classification using support vector machines." In: *Machine learning* 46.1 (2002), pp. 389–422.

[21] Y. Ke and R. Sukthankar. "PCA-SIFT: A more distinctive representation for local image descriptors." In: *IEEE Conference on Computer Vision and Pattern Recognition*. Vol. 2. 2004, pp. 506–513.

[22] S. Lazebnik, C. Schmid, and J. Ponce. "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories." In: *IEEE Conference on Computer Vision and Pattern Recognition*. Vol. 2. 2006, pp. 2169–2178.

[23] S. Lazebnik, C. Schmid, J. Ponce, et al. "Spatial pyramid matching." In: *Object Categorization: Computer and Human Vision Perspectives* (2009), pp. 401–415.

[24] H. Y. Lee, H. K. Lee, and Y. H. Ha. "Spatial color descriptor for image retrieval and video segmentation." In: *IEEE Transactions on Multimedia* 5.3 (2003), pp. 358–367.

[25] D. G. Lowe. "Object recognition from local scale-invariant features." In: *IEEE International Conference on Computer Vision*. Vol. 2. 1999, pp. 1150–1157.

[26] M. Maire et al. "Using contours to detect and localize junctions in natural images." In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2008, pp. 1–8.

[27] T. Mandal, Q. M. Jonathan Wu, and Y. Yuan. "Curvelet based face recognition via dimension reduction." In: *Signal Processing* 89.12 (2009), pp. 2345–2353.

[28] T. Mandal and Q. M.J Wu. "Face recognition using curvelet based PCA." In: *International Conference on Pattern Recognition*. 2008, pp. 1–4.

[29] B. S. Manjunath and W. Y. Ma. "Texture features for browsing and retrieval of image data." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 18.8 (1996), pp. 837–842.

[30] D. R. Martin, C. C. Fowlkes, and J. Malik. "Learning to detect natural image boundaries using local brightness, color, and texture cues." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26.5 (2004), pp. 530–549.

[31] K. Mikolajczyk and C. Schmid. "Scale & affine invariant interest point detectors." In: *International Journal of Computer Vision* 60.1 (2004), pp. 63–86.

[32] D. Nister and H. Stewenius. "Scalable recognition with a vocabulary tree." In: *IEEE Conference on Computer Vision and Pattern Recognition*. Vol. 2. 2006, pp. 2161–2168.

[33] E. Nowak, F. Jurie, and B. Triggs. "Sampling strategies for bag-of-features image classification." In: *European Conference on Computer Vision* (2006), pp. 490–503.

[34] A. Oliva and A. Torralba. "Modeling the shape of the scene: A holistic representation of the spatial envelope." In: *International Journal of Computer Vision* 42.3 (2001), pp. 145–175.

[35] *Parameter values for the HDTV standards for production and international programme exchange*. International Telecommunication Union, 2002.

[36] Shmuel Peleg, Michael Werman, and Hillel Rom. "A Unified Approach to the Change of Resolution: Space and Gray-Level." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 7 (1989), pp. 739–742.

[37] J. Philbin et al. "Object retrieval with large vocabularies and fast spatial matching." In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2007, pp. 1–8.

[38] M. Pontil and A. Verri. "Support vector machines for 3D object recognition." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20.6 (1998), pp. 637–646.

[39] Y. Rubner and C. Tomasi. "Texture-based image retrieval without segmentation." In: *IEEE International Conference on Computer Vision*. Vol. 2. 1999, pp. 1018–1024.

[40] Y. Rubner, C. Tomasi, and L. J. Guibas. "A metric for distributions with applications to image databases." In: *IEEE International Conference on Computer Vision*. 1998, pp. 59–66.

[41] F. Schaffalitzky and A. Zisserman. "Viewpoint invariant texture matching and wide baseline stereo." In: *IEEE International Conference on Computer Vision*. Vol. 2. 2001, pp. 636–643.

[42] C. E. Shannon. "Communication in the presence of noise." In: *Proceedings of the IEEE* 86.2 (1998), pp. 447–457.

[43] A. Shrivastava et al. "Data-driven Visual Similarity for Cross-domain Image Matching." In: *ACM Transaction of Graphics* 30.6 (Dec. 2011), 154:1–154:10.

[44] J. Sivic and A. Zisserman. "Video Google: A text retrieval approach to object matching in videos." In: *IEEE International Conference on Computer Vision*. 2003, pp. 1470–1477.

[45] A. W.M Smeulders et al. "Content-based image retrieval at the end of the early years." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22.12 (2000), pp. 1349–1380.

[46] M. Stricker and A. Dimai. "Color indexing with weak spatial constraints." In: *Storage and Retrieval for Image and Video Databases IV* 2670 (1996), pp. 29–40.

[47] M. J. Swain and D. H. Ballard. "Color indexing." In: *International Journal of Computer Vision* 7.1 (1991), pp. 11–32.

[48] M. A. Turk and A. P. Pentland. "Face recognition using eigenfaces." In: *IEEE Conference on Computer Vision and Pattern Recognition*. 1991, pp. 586–591.

[49] A. Utenpattanant, O. Chitsobhuk, and A. Khawne. "Color descriptor for image retrieval in wavelet domain." In: *International Conference on Advanced Communication Technology*. Vol. 1. 2006, pp. 818–821.

[50] J. Winn, A. Criminisi, and T. Minka. "Object categorization by learned universal visual dictionary." In: *IEEE International Conference on Computer Vision*. Vol. 2. 2005, pp. 1800–1807.

[51] J. Wu and J. M. Rehg. "Beyond the Euclidean distance: Creating effective visual codebooks using the histogram intersection kernel." In: *IEEE International Conference on Computer Vision*. 2009, pp. 630–637.

[52] Guang Yang and Yingyuan Xiao. "A Robust Similarity Measure Method in CBIR System." In: *Congress on Image and Signal Processing*. IEEE, 2008, pp. 662–666.

[53] J. Yang et al. "Evaluating bag-of-visual-words representations in scene classification." In: *Proceedings of the International Workshop on Multimedia Information Retrieval*. 2007, pp. 197–206.

[54] J. Zhang et al. "Local features and kernels for classification of texture and object categories: A comprehensive study." In: *International Journal of Computer Vision* 73.2 (June 2007), pp. 213–238.

[55] L. Zhu, A. B Rao, and A. Zhang. "Theory of keyblock-based image retrieval." In: *ACM Transactions on Information Systems* 20.2 (2002), pp. 224–257.

COLOPHON

This document was typeset using the typographical look-and-feel `classicthesis` developed by André Miede. The style was inspired by Robert Bringhurst's seminal book on typography "*The Elements of Typographic Style*". `classicthesis` is available for both LaTeX and LyX:

`http://code.google.com/p/classicthesis/`

Happy users of `classicthesis` usually send a real postcard to the author, a collection of postcards received so far is featured here:

`http://postcards.miede.de/`

Final Version as of October 22, 2012 (`classicthesis` version 0.1).