FELIX STÜRMER

SKETCH-BASED IMAGE RETRIEVAL USING CURVELETS

SKETCH-BASED IMAGE RETRIEVAL USING CURVELETS

FELIX STÜRMER

An Evaluation of Curvlet-Based Cross-Domain Descriptors for Sketch-Based Image Retrieval

January 2012 – version 0.1

[September 18, 2012 at 15:15 – classicthesis version 0.1]

ABSTRACT

Short summary of the contents...

ACKNOWLEDGMENTS

acknowledgments go here...

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LISTINGS

ACRONYMS

1 INTRODUCTION

1.1 MOTIVATION

Paragraph about increase in visual data, mobile cameras, medicine, etc...

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 [43]. 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 predicate 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 full-color images or contour images. The fast discrete curvelet transform [9] is used to analyse image segments.

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 analysed 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 specific 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 [43]:

> 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 [43], where it's defined 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. [40]

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.

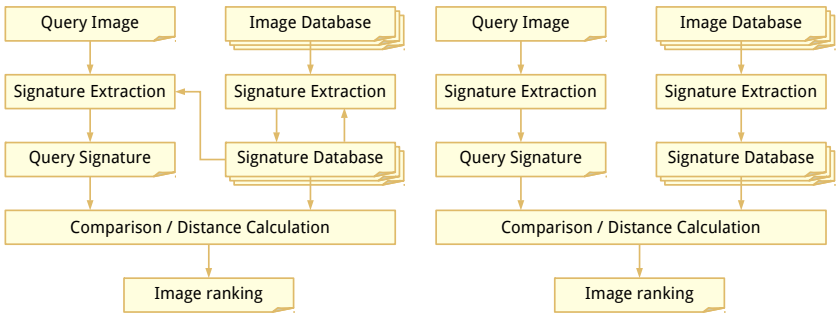
PERSPECTIVE An object's proportions can be distorted by the imaging perspective.

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 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 image from that domain. By their very nature, the domain of sketch based image retrieval systems is usually very broad. It contains the sketches create 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. photos or paintings.

Another factor usually is the accepted input format of the feature extraction algorithm. Many algorithms like SIFT [23] 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 [50].

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 [47], while others [44] [13] [22] include spatial relations between colors in a region. Many descriptors attempt to capture texture characteristics, such as [39] and [27]. Rubner and Thomasi [37] combine Gabor filters and the earth mover's distance and thereby bypass segmentation. Another class of descriptors focuses on representing shapes using detection of edges and salient points. Lowe [23] 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 [32] capture various scene properties like "roughness" and "openness".

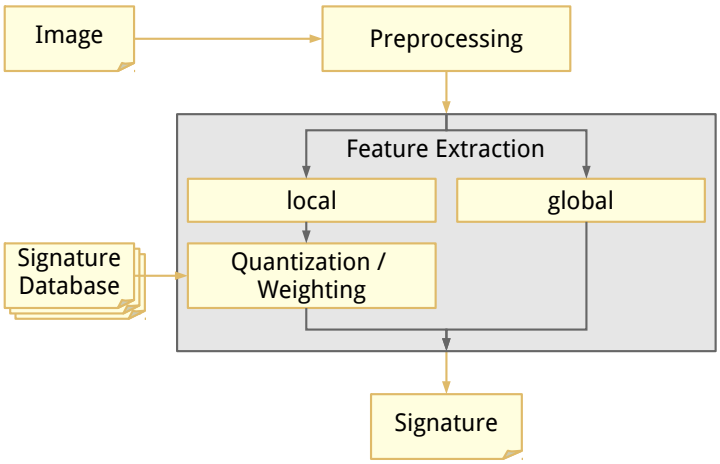


Figure 2: Signature extraction in CBIR systems

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 in [47]. Some global algorithms subdivide the image into regular segments and derive the localized distribution of features for each division. [20] and later [21] 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 [32] 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 [23] achieves this by selecting the feature locations such that they can be normalized with respect to scale, orientation and limited 3D projections. The SURF descriptor 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 vector, 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, so 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) [46], wavelets (waveletfaces) [15] and more recently curvelets (curveletfaces) [26]. In [19] it was used to improve the robustness of the SIFT [23] descriptor. To overcome the limited between-class discrimination of PCA, it has been combined with Linear Discriminant Analysis (LDA), yielding even better results [25].

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" [42], 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 [31] [51].

The most common clusting method used in numerous publications [53] [42] [11] [16] [48] 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 not a global minimum. Other clusting methods are hierarchical agglomorative and divisive algorithms as used in [30] and [35], that 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 cardinality 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 *Distance Metrics*

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 in [29] and [42] to compare feature vectors of the SIFT [23] descriptor.

COSINE DISTANCE A metric sometimes used in information retrieval applications is the coside 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.

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 in [34]. The use of the algorithm as a signature comparison metric in image retrieval was published by Rubner et al. [38]. 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

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

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

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

In [38] the authors propose using a simplex-based algorithm to solve the transportation problem, that achieves good performance by exploiting the specific problem structure. Zhang at al. [52] 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 [45], 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:

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

2.2.3.2 *Weighting*

LINEAR SUPPORT VECTOR MACHINES The concept of support vector machines (SVMs) is usually is usually employed when a feature needs to be classified into one or more classes. While such classification problem occur frequently in computer vision in the fields of object detection [36] [11], human detection [12] or scene classification [51], 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. [18] 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. in [41]. 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 [42]. 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 of documents m_i 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.

2.3 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 particulaly successful application to CBIR was the use of gabor wavelets as proposed by Manjunath and Ma [27]. 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 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.

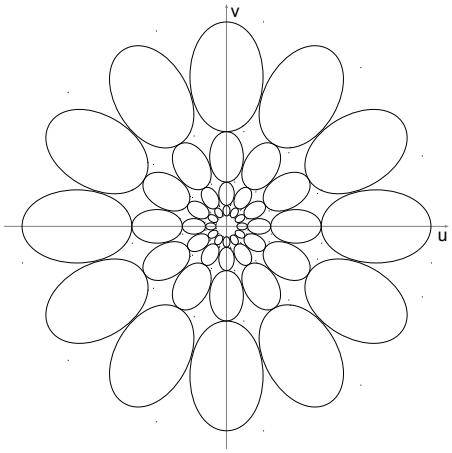


Figure 3: Tiling of Gabor wavelets. Note that due to the elliptical shape of the wavelets, there are some parts of the spectrum left uncovered.

2.3.2 *The Continuous Curvelet Transform*

The formulation of the continuous curvelet transform (CCT) by Candes and Donoho in [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 unfeasable.

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

The above formulation became known as the curvelet 99 transform when Candes and Donoho revised it soon after in [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). A combined frequency window U_j (Figure 4c) can then be defined as

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

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

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

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

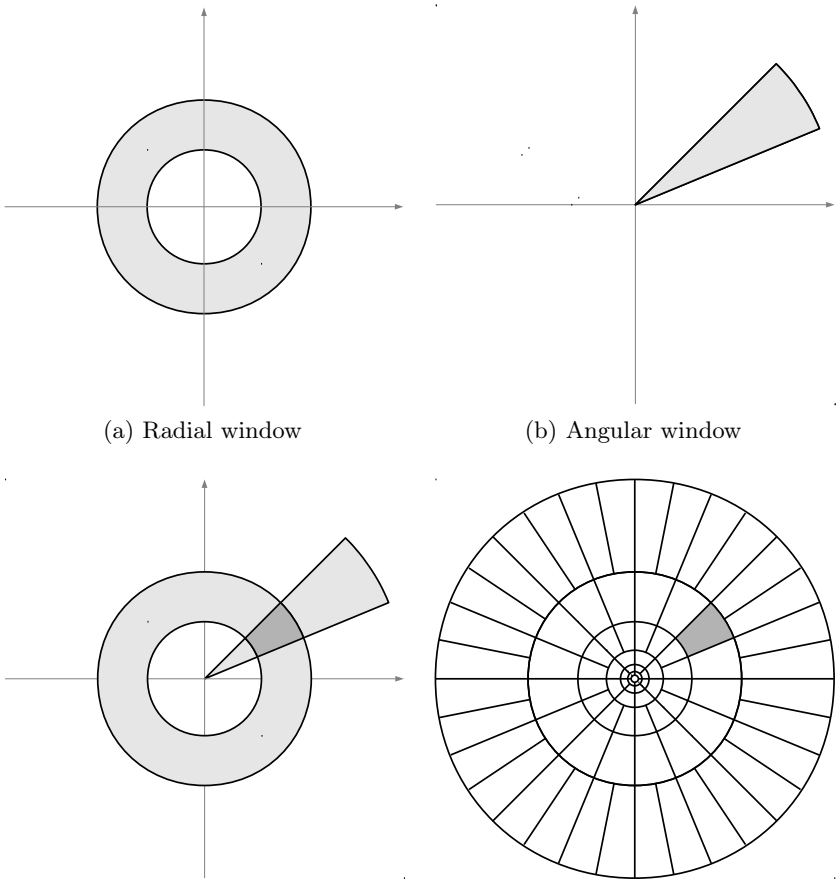
where $x = (x_1, x_2)$, R_θ the rotation matrix for angle θ and $x_k^{(j,l)} = R_{\theta_l}^{-1}(k_1 \cdot 2^{-j}, k_2 \cdot 2^{-\frac{j}{2}})$.

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

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



(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 $\mathtt{width} \propto \mathtt{length}^2$ relation. (d) shows the wedge within a schema of the complete tiling in frequency domain.

Using the window defining the coarse scale curvelet $\varphi_{j_0,k}$ via its Fourier transform is straightforward:

$$\varphi_{j_0,k}(x) = \varphi_{j_0}(x - 2^{-j_0}k), \widehat{\varphi_{j_0}}(\omega) = 2^{-j_0}W_0(2^{-j_0}|\omega|), \tag{3}$$

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 between the curvelets, 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):

$$c^D(j, l, k) := \sum_{0 \leqslant t_1, t_2 < n} f[t_1, t_2] \overline{\varphi^D_{j,l,k}[t_1, t_2]}. \tag{4}$$

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 squares "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 $l = -2^{\lfloor \frac{j}{2} \rfloor}, \ldots, 2^{\lfloor \frac{j}{2} \rfloor} - 1$. The construction of the corner windows are Figure 5 shows a tiling of all $U^D_{j,l}$.

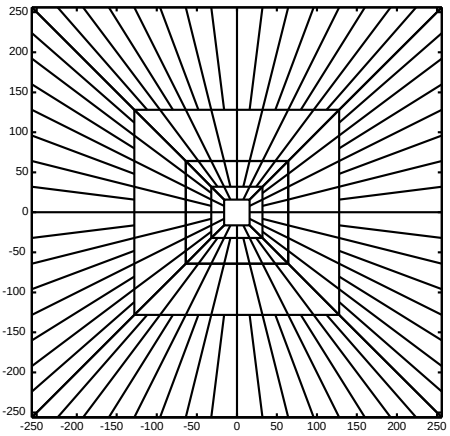


Figure 5: Discrete frequency tiling using concentric squares

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 Fourier transforms. These transforms represent a polynomial interpolation of each "column" of the parallelogram P_j containing the sheared window (Figure 6a), that can be computed with a $O(n^2 \log n)$ complexity in a sufficiently exact approximation.

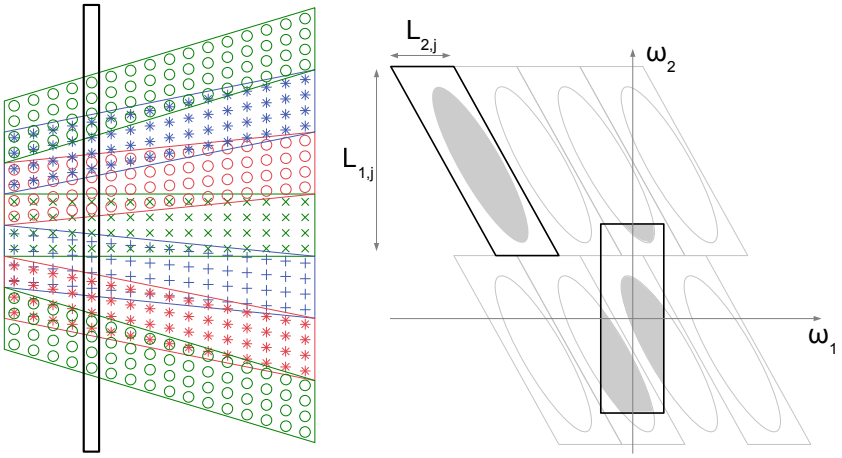
This yields an object $\hat{f}[n_1, n_2 - n_1 \tan \theta_l]$ for $(n_1, n_2) \in P_j$, that can be multiplied with the window U_j^D described above in order to create a localized "wedge" with the orientation θ_l:

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

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

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

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



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

Figure 6: (a) illustrates the respective grid for each parallelogram containing the 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 hat a much smaller axis aligned bounding box for the FFT to operate on.

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 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 $U_{j,l}^D$, $L_{1,j}$ 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 6). Thus, the wrapped, localized data are

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

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:

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

2.3.4 *gPb Contour Detection*

In [24], Maire et al. describe an improvement of the contour detector published in [28] 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 combonent $sPb(x, y, \theta)$ 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 [17]. The linear combination of the individual directional derivatives then represents the large-scale contours in the image:

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

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:

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

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 in [2]. Signatures are constructed using both, global curvelet features, and a bag-of-features approach similar to what was described in [42] and [14].

The following sections will describe the variations of the processing stages *image acquisition*, *signature extraction*, and *ranking*. To reference the individual variations unambigiously, 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 assumptions. They may be color photographs (Figure 7a), 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 [33] (Figure 7b). 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 7c):

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

CANNY A slightly more complex way to extract edges is the Canny edge detector [10]. Intially, 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

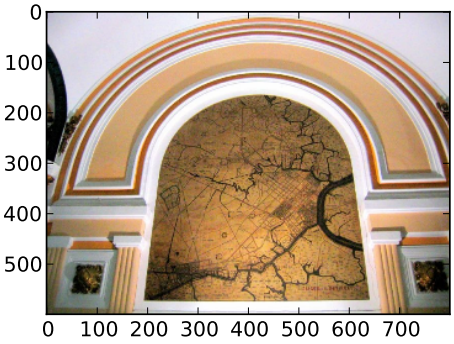
$$\Theta = \arctan \frac{G_x}{G_y}$$

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 fluctiations, 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 7d).

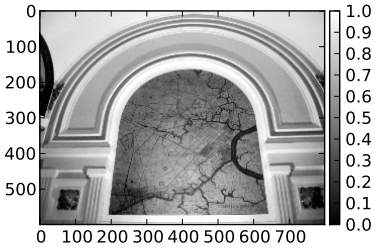
SEGMENT What a human sketches as a line in an image is often a boundary between two regions with different color or texture characteristics. Therefore the output from image segmentation algorithms can also indicate to location of edges. The hierarchical segmentation algorithm gPb-owt-ucm published by 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 2.3.4 they apply the Oriented Watershed Transform, that merges adjacent regions of an over-segemented image. The criterion for merging is the strength of the boundary shared by the two regions.

3.2 SIGNATURE EXTRACTION

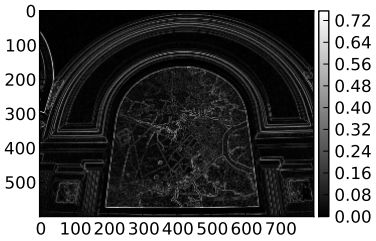
Each method of signature extraction described in this section has the fast discrete curvelet transform at its heart. The curvelet transform has



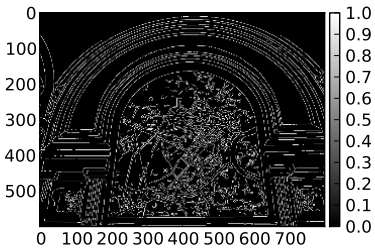
(a) Original image



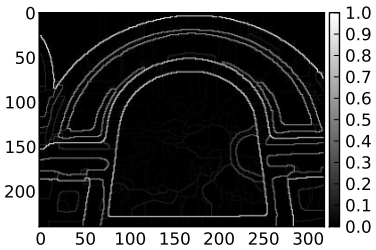
(b) Image after luma conversion



(c) Image after Sobel operator



(d) Image after Canny operator



(e) Image after gPb contour detection

Figure 7: Image acquisition variants

two main parameters, that influence the result: The number of angles N_θ used at the coarest scale and the number of scales N_j, which corresponds to the number of concentric squares shown in 6. 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. Furthermore, since the coarsest scale is non-directional, as explained in section 2.3.2, it is ignored in further computations.

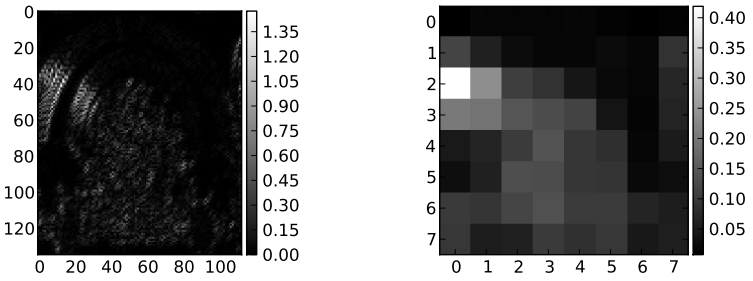
The response image generated by the FDCT for each pair of scale and angle is too large to be considered for the signature directly. Therefore

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 $x, y \in 1, 2, \ldots, n$:

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

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

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



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

Figure 8: 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 are shown.

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

descriptions or "visual words" similar to what was described in [42]. The exact way to extract the words will be detailed towards the end of this section.

The set of visual words extracted from the images are diverse and hard to compare. In order to create meaningful image signatures from these words, the whole set is condensed into a dictionary using k-means clustering. As already discussed in 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. [31] observe an increase in accuracy up to 1000 words, but overfitting for some sampling algorithms beyond that. At the same time [51] report optimal sizes of 20000 to 80000 depending on the image database. In [14] a size of 1000 visual words was found optimal for sketches.

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 9). This results in $n - m + 1$ 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 $u \cdot v$ 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.

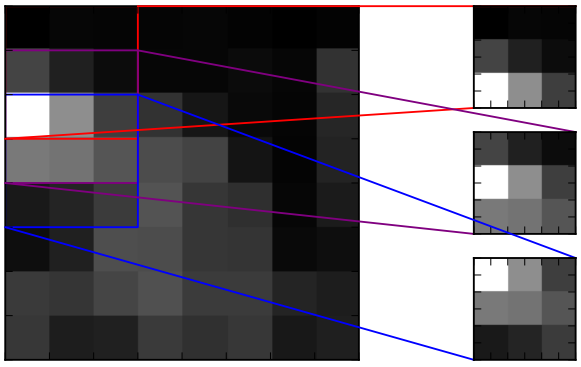


Figure 9: The 8×8 mean coefficient grid $\bar{C}_{s,\theta}$ is densely sampled using windows of 3×3 grid cells to produce 36 patches per scale and angle.

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 $N_{\theta_s} \cdot m^2$.

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. Evaluations by Nowak et al. [31] 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 in [20] and [21]. The R-HOG descriptor [12] also uses a dense grid for sampling with overlapping windows to improve matching performance.

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 similarity to the query image in descending order. The similarity can be determined using various metrics and similarity measures, which have been detailed in 2.2.3.1. 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.

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 [49]. It has the added benefit of allowing for partial matches in a signature. As can be seen from the definition in 2.2.3.1, the result lies within $[0, 1]$ with 1 being a perfect match. Therefore, $1 - s_{HI}$ is used to sort the result list.

EMD The Earth Mover's Distance is solved using a simplex algorithm veriant, 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 method 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 don't feature a well-defined "correct" solution. An alternative approach is looking the recall and precision characteristics defined as

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

$$1 - \text{precision} = \frac{\text{number of false positive results}}{\text{total number of results}}$$

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 systems that assign a degree of similarity to each item in the result set, this would require applying a threshold for classification. In addition to the distorting influence of an unsuitable thresholding choice, this method completely disregards the match quality within the positive result set, that can be important information to user of such a retrieval system.

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 corresponds to the method described in [14], 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 VARIANTS AND RESULTS

4.2.1 *Global Features*

TBD

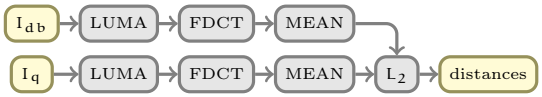


Figure 10

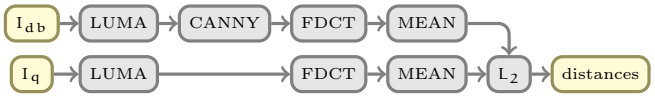


Figure 11

4.2.2 *Local Features*

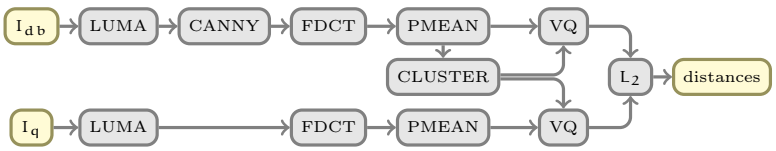

I_{db}
LUMA
CANNY
FDCT
PMEAN
VQ
CLUSTER
L_2
distances
I_q
LUMA
FDCT
PMEAN
VQ

Figure 12

5 ANALYSIS

Analysis goes here...

6 CONCLUSION

Conclusion goes here...

BIBLIOGRAPHY

[1] A.E. Abdel-Hakim and A.A. Farag. "CSIFT: A SIFT Descriptor with Color Invariant Characteristics." In: *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Vol. 2. 2006, pp. 1978 –1983. DOI: 10.1109/CVPR.2006.95.

[2] P. Arbelaez et al. "Contour detection and hierarchical image segmentation." In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 99 (2011), 1–1.

[3] P. Arbeláez et al. "From contours to regions: An empirical evaluation." In: *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. 2009, 2294–2301. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5206707 (visited on 08/30/2012).

[4] R. Baeza-Yates, B. Ribeiro-Neto, et al. *Modern information retrieval*. Vol. 82. Addison-Wesley New York, 1999. (Visited on 09/11/2012).

[5] H. Bay et al. "Speeded-up robust features (SURF)." In: *Computer Vision and Image Understanding* 110.3 (2008), 346–359. URL: http://www.sciencedirect.com/science/article/pii/S1077314207001555 (visited on 08/13/2012).

[6] E. J. Candes. "Ridgelets: theory and applications." PhD thesis. Stanford University, 1998. URL: http://www-stat.stanford.edu/~candes/papers/thesis.ps (visited on 08/01/2012).

[7] E. J. Candes and D. L. Donoho. *Curvelets: A surprisingly effective nonadaptive representation for objects with edges*. Tech. rep. DTIC Document, 2000. URL: http://oai.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=ADP011978 (visited on 08/01/2012).

[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), 219–266. URL: http://onlinelibrary.wiley.com/doi/10.1002/cpa.10116/abstract (visited on 08/01/2012).

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

[10] J. Canny. "A computational approach to edge detection." In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 6 (1986), 679–698. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4767851 (visited on 08/21/2012).

[11] G. Csurka et al. "Visual categorization with bags of keypoints." In: *Workshop on statistical learning in computer vision, ECCV*. Vol. 1. 2004, p. 22.

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

[13] Y. Deng et al. "An efficient color representation for image retrieval." In: *Image Processing, IEEE Transactions on* 10.1 (2001), 140–147. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=892450 (visited on 08/24/2012).

[14] M. Eitz et al. "Sketch-based image retrieval: benchmark and bag-of-features descriptors." In: *Visualization and Computer Graphics, IEEE Transactions on* 99 (2010), 1–1. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5674030 (visited on 08/14/2012).

[15] 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), p. 226.

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

[17] C. Fowlkes, D. Martin, and J. Malik. "Learning affinity functions for image segmentation: Combining patch-based and gradient-based approaches." In: *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*. Vol. 2. 2003, II–54. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1211452 (visited on 08/30/2012).

[18] I. Guyon et al. "Gene selection for cancer classification using support vector machines." In: *Machine learning* 46.1 (2002), 389–422. URL: http://www.springerlink.com/index/w68424066825vr3l.pdf (visited on 09/10/2012).

[19] Y. Ke and R. Sukthankar. "PCA-SIFT: A more distinctive representation for local image descriptors." In: *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*. Vol. 2. 2004, II–506. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1315206 (visited on 08/31/2012).

[20] S. Lazebnik, C. Schmid, and J. Ponce. "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories." In: *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*. Vol. 2. 2006, 2169–2178. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1641019 (visited on 08/27/2012).

[21] S. Lazebnik, C. Schmid, J. Ponce, et al. "Spatial pyramid matching." In: (2009). URL: http://hal.inria.fr/inria-00548647/ (visited on 09/10/2012).

[22] H. Y. Lee, H. K. Lee, and Y. H. Ha. "Spatial color descriptor for image retrieval and video segmentation." In: *Multimedia, IEEE Transactions on* 5.3 (2003), 358–367. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1223563 (visited on 08/14/2012).

[23] D. G. Lowe. "Object recognition from local scale-invariant features." In: *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*. Vol. 2. 1999, 1150–1157. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=790410 (visited on 08/13/2012).

[24] M. Maire et al. "Using contours to detect and localize junctions in natural images." In: *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. 2008, 1–8. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4587420 (visited on 08/30/2012).

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

[26] T. Mandal and Q. M.J Wu. "Face recognition using curvelet based PCA." In: *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*. 2008, 1–4.

[27] B. S. Manjunath and W. Y. Ma. "Texture features for browsing and retrieval of image data." In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 18.8 (1996), 837–842. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=531803 (visited on 08/16/2012).

[28] D. R. Martin, C. C. Fowlkes, and J. Malik. "Learning to detect natural image boundaries using local brightness, color, and texture cues." In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 26.5 (2004), 530–549. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1273918 (visited on 08/30/2012).

[29] K. Mikolajczyk and C. Schmid. "Scale & affine invariant interest point detectors." In: *International journal of computer vision* 60.1 (2004), 63–86. URL: http://www.springerlink.com/index/H37T7833M7037173.pdf (visited on 08/27/2012).

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

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

[32] 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), 145–175. URL: http://www.springerlink.com/index/K62TG81W8352G71H.pdf (visited on 08/16/2012).

[33] *Parameter values for the HDTV standards for production and international programme exchange*. 2002. URL: http://www.itu.int/dms_pubrec/itu-r/rec/bt/R-REC-BT.709-5-200204-I!!PDF-E.pdf.

[34] Shmuel Peleg, Michael Werman, and Hillel Rom. *A Unified Approach to the Change of Resolution: Space and Gray-Level*. 1989.

[35] J. Philbin et al. "Object retrieval with large vocabularies and fast spatial matching." In: *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*. 2007, 1–8.

[36] M. Pontil and A. Verri. "Support vector machines for 3D object recognition." In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 20.6 (1998), 637–646. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=683777 (visited on 08/27/2012).

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

[38] Y. Rubner, C. Tomasi, and L. J. Guibas. "A metric for distributions with applications to image databases." In: *Computer Vision, 1998. Sixth International Conference on*. 1998, 59–66. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=710701 (visited on 09/06/2012).

[39] F. Schaffalitzky and A. Zisserman. "Viewpoint invariant texture matching and wide baseline stereo." In: *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*. Vol. 2. 2001, 636–643. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=937686 (visited on 08/24/2012).

[40] C. E. Shannon. "Communication in the presence of noise." In: *Proceedings of the IEEE* 86.2 (1998), 447–457. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=659497 (visited on 07/23/2012).

[41] A. Shrivastava et al. "Data-driven Visual Similarity for Cross-domain Image Matching." In: *ACM Transaction of Graphics (TOG) (Proceedings of ACM SIGGRAPH ASIA)*. 2011.

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

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

[44] M. Stricker and A. Dimai. "Color indexing with weak spatial constraints." In: *Storage and Retrieval for Image and Video Databases IV* 2670 (1996), 29–40. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.20.3010&rep=rep1&type=pdf (visited on 08/24/2012).

[45] M. J. Swain and D. H. Ballard. "Color indexing." In: *International journal of computer vision* 7.1 (1991), 11–32. URL: http://www.springerlink.com/index/N231L41541P12L1G.pdf (visited on 09/06/2012).

[46] M. A. Turk and A. P. Pentland. "Face recognition using eigenfaces." In: *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR'91., IEEE Computer Society Conference on*. 1991, 586–591. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=139758 (visited on 08/31/2012).

[47] A. Utenpattanant, O. Chitsobhuk, and A. Khawne. "Color descriptor for image retrieval in wavelet domain." In: *Advanced Communication Technology, 2006. ICACT 2006. The 8th International Conference*. Vol. 1. 2006, 4–pp. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1625694 (visited on 08/14/2012).

[48] J. Winn, A. Criminisi, and T. Minka. "Object categorization by learned universal visual dictionary." In: *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*. Vol. 2. 2005, 1800–1807. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1544935 (visited on 08/24/2012).

[49] J. Wu and J. M. Rehg. "Beyond the Euclidean distance: Creating effective visual codebooks using the histogram intersection kernel." In: *Computer Vision, 2009 IEEE 12th International Conference on*. 2009, 630–637. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5459178 (visited on 09/18/2012).

[50] Guang Yang and Yingyuan Xiao. "A Robust Similarity Measure Method in CBIR System." In: IEEE, 2008, pp. 662–666. ISBN: 978-0-7695-3119-9. DOI: 10.1109/CISP.2008.185. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4566386 (visited on 04/16/2012).

[51] J. Yang et al. "Evaluating bag-of-visual-words representations in scene classification." In: *Proceedings of the international workshop on Workshop on multimedia information retrieval*. 2007, 197–206.

URL: http://dl.acm.org/citation.cfm?id=1290111 (visited on 09/04/2012).

[52] J. Zhang et al. "Local features and kernels for classification of texture and object categories: A comprehensive study." In: *Computer Vision and Pattern Recognition Workshop, 2006. CVPRW '06. Conference on*. June 2006.

[53] L. Zhu, A. B Rao, and A. Zhang. "Theory of keyblock-based image retrieval." In: *ACM Transactions on Information Systems (TOIS)* 20.2 (2002), 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/

DECLARATION

Put your declaration here.

Berlin, January 2012

Felix Stürmer