COMPUTER VISION BASED REUSE DETECTION IN DIGITAL ARTWORKS

by

Furkan Işıkdoğan

B.S, Computer Engineering, Yıldız Technical University, 2011

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Computer Engineering

Boğaziçi University

2013

ACKNOWLEDGEMENTS

Foremost, I would like express my sincere gratitude to my thesis supervisor, Prof. Albert Ali Salah, for his guidance and support. I really appreciate his continuous help regardless of physical distance.

I am grateful to Prof. Alan Bovik for inspiring me with his insightful lectures, supporting my future research, and helping me achieve my academic goals.

I would like to thank Prof. Lale Akarun for her valuable lectures, which have formed the basis of my research.

I would like to thank Prof. Songül Albayrak for guiding me through my undergraduate studies, accepting to be in my jury, and for reviewing my thesis.

I would like to thank TÜBİTAK, the Scientific and Technological Research Council of Turkey, for supporting me financially with the National Scholarship Program for M.Sc. Students.

Last but not least, I would like to thank my family for everything they have done for me throughout my life.

ABSTRACT

COMPUTER VISION BASED REUSE DETECTION IN DIGITAL ARTWORKS

Image reuse refers to the use of visual elements of existing images in order to create new ones. In this thesis, we study the automatic image reuse detection problem in digital artworks, which is a relatively under-studied problem of image retrieval. We introduce two novel image reuse datasets: an artificial dataset that simulates different types of reuse systematically, and an annotated natural dataset that includes a set of digital artworks that are crawled from the web. Based on the natural dataset, we propose a taxonomy which identifies the primary types of reuse and manipulations. Then, for image reuse detection, we evaluate different feature extraction and classification methods that are commonly used for image copy detection, content-based image retrieval, and computer analysis of artworks. The features we use include, color histograms, Histogram of Oriented Gradient (HOG) descriptors, and the Scale Invariant Feature Transform (SIFT) descriptor and its color-based variants. We use the bag-of-visual-words (BoW) approach with the SIFT descriptors. We also present a novel image description algorithm, called the Affine Invariant Salient Patch (AISP) descriptor, which provides a foreground sensitive description of images by fitting concentric ellipses to the most salient region in an image and extracting features from each track. Our results show that the AISP method can be suitable for reuse detection with its compactness and good retrieval accuracy, especially in images with prominent foreground objects. On the other hand, the use of the SIFT descriptors in a BoW model can be more advisable in a more natural setting and for cluttered scenes.

ÖZET

DİJİTAL SANAT ESERLERİNDE BİLGİSAYARLA GÖRÜ TABANLI TEKRAR KULLANIM TESPİTİ

İmge işlemede tekrar kullanım, var olan bir imgenin görsel bileşenlerinin yeni bir imge oluşturmak için kullanılmasını ifade etmektedir. Bu tezde literatürde daha önce kapsamlı olarak ele alınmamış bir konu olan dijital sanat eserlerinde tekrar kullanımın otomatik olarak tespit edilmesi üzerinde çalıştık. Bu çalışmada iki yeni veri seti sunuyoruz: farklı tekrar kullanım biçimlerini sistematik olarak benzeten sentetik bir veri seti ve internetten derlenmiş, alt bilgi içeren görüntülerden oluşan doğal bir veri seti. Doğal veri setindeki görüntülerden yola çıkarak temel tekrar kullanım ve düzenleme tiplerini tanımlayan bir taksonomi önerdik. Daha sonra, imgelerde kopya tespiti, içerik tabanlı imge erişimi ve sanat eserlerinin bilgisayar analizi gibi problemlerde yaygın olarak kullanılan yöntemleri –renk histogramları, HOG, BoW modelinde kullanılan SIFT tanımlayıcısı ve renk-tabanlı varyantları– görüntü tekrar kullanım tespiti için kullanımını değerlendirdik. Var olan yöntemlere ek olarak, AISP adını verdiğimiz, görsel olarak belirgin olan alanlara uydurulan eş merkezli elipslerden çıkarılan özelliklerden faydalanan, ön plan hassasiyeti olan bir görüntü tanımlama algoritması önerdik. Sonuçlarımız, görüntülerin kompakt bir şekilde karakterize edilmesine olanak sağlayan AISP yönteminin kabul edilebilir doğruluk oranları ile tekrar kullanım tespiti problemi için özellikle nesnelerin belirgin olarak ön planda olduğu durumlarda kullanışlı olabileceğini, bununla birlikte görüntülerin sanatçılar tarafından oluşturulduğu daha doğal durumlarda SIFT tanımlayıcılarının BoW modeli ile kullanımının tavsiye edilebilir olduğunu göstermektedir.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS . . . iii
ABSTRACT . . . iv
ÖZET . . . v
LIST OF FIGURES . . . viii
LIST OF TABLES . . . xi
LIST OF SYMBOLS . . . xii
LIST OF ACRONYMS/ABBREVIATIONS . . . xiv
1. INTRODUCTION . . . 1
1.1. Motivation . . . 1
1.2. Related Work . . . 2
1.2.1. Image Copy and Manipulation Detection . . . 2
1.2.2. Content Based Image Retrieval . . . 5
1.2.3. Saliency Estimation . . . 7
1.2.4. Digital Artwork in Computer Vision . . . 8
1.3. Challenges . . . 10
1.4. Contributions . . . 11
1.5. Outline of the Thesis . . . 11
2. TAXONOMY . . . 12
2.1. Types of reuse . . . 12
2.2. Types of manipulations . . . 16
3. METHODOLOGY . . . 20
3.1. Color Histograms . . . 20
3.2. Histograms of Oriented Gradients . . . 21
3.3. Scale Invariant Feature Transform . . . 22
3.4. Color-based Variants of SIFT . . . 25
3.5. Bag of Words Model . . . 26
3.6. Saliency Based Segmentation . . . 27
3.7. Affine Invariant Salient Patch Descriptors . . . 29
3.7.1. Salient Patch Detection . . . 29

3.7.2. Affine Invariant Feature Extraction 30
3.7.3. Addition of Global Features and Normalization 34
3.8. Matching 35
4. EXPERIMENTS 36
4.1. Datasets 36
4.1.1. Artificial Dataset 36
4.1.2. Natural Dataset 38
4.2. Experimental Setup 41
4.3. Preliminary Experiments 41
4.4. Matching Results 42
4.4.1. Experiments on the Artificial Dataset 43
4.4.2. Experiments on the Natural Dataset 48
4.5. Comparison in terms of dimensionality and computation time 56
5. CONCLUSION 58
REFERENCES 60

LIST OF FIGURES

Figure 1.1. Sub-image retrieval example in the work of Yan Ke *et al.* 4

Figure 1.2. Partial-duplicate image search example in the work of Zhou *et al.* 4

Figure 2.1. A taxonomy of types of reuse. 12

Figure 2.2. An example of partial reuse: the flower on the left image is used in the right image. 13

Figure 2.3. An example of direct reuse. 14

Figure 2.4. An example of use as a background. 14

Figure 2.5. An example of remake. 14

Figure 3.1. Keypoint descriptor computation. 24

Figure 3.2. Illustration of reused image matching using SIFT features. 24

Figure 3.3. An illustration of the bag-of-words model. 26

Figure 3.4. Example saliency-based segmentation results on our artificial dataset. 28

Figure 3.5. Example of salient patch extraction. 30

Figure 3.6. Illustration of AISP descriptors on an image. 32

Figure 3.7. Quiver plot of the gradient image of the salient patch. 33

Figure 4.1. Artificial dataset generation. 37

Figure 4.2. Example images from the natural dataset. 39

Figure 4.3. Cumulative matching accuracies for BoW model with different parameters. 42

Figure 4.4. Summary of the results on the artificial dataset. 43

Figure 4.5. Cumulative matching accuracies of the methods for direct reuse on the artificial dataset. 45

Figure 4.6. Cumulative matching accuracies of the methods for partial reuse on the artificial dataset with no transformation. 46

Figure 4.7. Cumulative matching accuracies of the methods for partial reuse on the artificial dataset with aspect ratio change and blur. 46

Figure 4.8. Cumulative matching accuracies of the methods for partial reuse on the artificial dataset with color change and rotation. 47

Figure 4.9. CMC plots for the direct reuse category on the natural dataset. . . 48

Figure 4.10. CMC plots for the partial reuse category on the natural dataset. . 49

Figure 4.11. CMC plots for the remake remake category on the natural dataset. 50

Figure 4.12. CMC plots for the use as a background category on the natural dataset. 50

Figure 4.13. Summary of the results on the natural dataset for the four types of reuse. 51

Figure 4.14. CMC plots for the most frequently observed types of manipulations on the natural dataset. 53

Figure 4.15. CMC plots for less frequently observed types of manipulations on the natural dataset. 54

Figure 4.16. CMC plots for the least frequently observed types of manipulations on the natural dataset. 55

Figure 4.17. Summary of the results on the natural dataset for nine different types of manipulations. 56

LIST OF TABLES

Table 2.1. Observation frequencies of reuse types. 15

Table 2.2. Observation frequencies of modifications. 19

Table 4.1. Example tuples of annotations. 40

Table 4.2. Numbers of images for different types of reuse and manipulation. . 40

Table 4.3. Computation time and dimensionalities of the methods. 57

LIST OF SYMBOLS

a	Semi-major axis of an ellipse
b	Semi minor axis of an ellipse
$\mathbf{B}$	Binary segmentation mask
c	Number of occurrence
$\mathbf{C}$	Covariance matrix
d	Euclidean distance
d_s	Standardized Euclidean distance
$D_r(r_k, r_i)$	Color distance between regions r_k and r_i
$\mathbf{F}$	Reflection (flip/mirroring) matrix
$f(c)$	Frequency of color c
$\mathbf{G_x}$	Horizontal gradient image
$\mathbf{G_y}$	Vertical gradient image
$\mathbf{H}$	A histogram
$\mathbf{I_d}$	Destination image
$\mathbf{I_h}$	Hue component of an image
$\mathbf{I_o}$	Source (original) image
$\mathbf{I_s}$	Saturation component of an image
$\mathbf{I_v}$	Value (brightness) component of an image
$I(x, y)$	Pixel value at position x,y in an image
IDF	Inverse document frequency
K	Number of colors in an image
$\mathbf{M}$	Edge magnitude matrix
m_x	Horizontal component of an image centroid
m_y	Vertical component of an image centroid
n	Number of histogram bins
N	Number of retrieved images
p_i	Occurrence frequency of colors that lie in the range of i^{th} bin in an image
$\mathbf{R}$	Rotation matrix

$\mathbf{S}$	Scaling matrix
s_x	Horizontal scaling factor
s_y	Vertical scaling factor
$S(r_k)$	Saliency value of region r_k
$\mathbf{T}$	Edge orientation matrix
TF	Term frequency
$\mathbf{V_c}$	Matching candidate feature vector
$\mathbf{V_q}$	Query image feature vector
$w(r_i)$	Pixel count of region r_i
w_i	Visual word with an index i
$\mathbf{X}$	Pixel position matrix for a reused region on an image
α	Opacity parameter
μ	Central moments
σ	Standard deviation
θ_m	Principal orientation

LIST OF ACRONYMS/ABBREVIATIONS

AISP	Affine Invariant Salient Patch
BoW	Bag of Words
CBIR	Content Based Image Retrieval
CMC	Cumulative Matching Characteristic
DCT	Discrete Cosine Transform
HOG	Histograms of Oriented Gradients
HSV	Hue, Saturation, Value
PCA	Principal Component Analysis
PDF	Probability Distribution Function
RC	Region-based Contrast
SIFT	Scale Invariant Feature Transform
SURF	Speeded Up Robust Features

1. INTRODUCTION

Advances in multimedia technologies have led to an increase in the volume of digital artworks that are created and shared on the internet, especially on social networks for digital arts, which allow users to access a tremendous number of images. Images can be manipulated or reused in order to create new pieces of visual art by using image manipulation software, or by traditional techniques.

In this work, the term *digital artwork* refers to two-dimensional images that are available on digital media, including but not limited to digital photographs, computer generated images, and scanned drawings. We use the term *reuse* for any case in which an image is used as a source or reference in production of another artwork, and we aim to examine the types of reuse and provide a computer vision based framework that facilitates automatic detection of reuse in digital artwork.

1.1. Motivation

Automatic detection of reuse provides functionality for numerous tasks including source image location, similar image retrieval [1,2], popularity and influence analysis [3], image manipulation and forgery detection [4–6], and copyright violation detection [5,7, 8]. For example, the automatic location of source images can provide more information about a piece of digital art by listing a set of source images that are used in the composition. The source image information –i.e. the number of shared sources between images– can be used in image search as a semantic variable in addition to low-level image features that are extracted from whole images. This information can also be useful for image influence analysis for discovering the relationships between different genres of digital art and for measuring popularity for a specific piece of art. Last but not least, retrieval of images that use a specific source image can help artists or third parties detect possible copyright violations and also measure the popularity of their artwork.

1.2. Related Work

This section first reviews the previous work in the areas that are related to the image reuse detection problem: image copy detection and content-based image retrieval. Section 1.2.1 outlines the methodologies and datasets used in image copy detection, and Section 1.2.2 surveys image matching techniques for content-based image retrieval, the topics that show similarity to our topic, reuse detection, in terms of used methods. Image saliency estimation, which refers to the estimation of the regions or pixels that stand out in an image, can be useful for region of interest detection in digital artwork. Section 1.2.3 gives a brief review of saliency estimation and saliency-based segmentation methods. Finally, Section 1.2.4 summarizes some of the relevant work in computer vision to study digital artworks.

1.2.1. Image Copy and Manipulation Detection

Digital images can be easily copied and manipulated using graphics editing software. In order to prevent copyright violations, watermarking –which is defined as altering a work imperceptibly to embed a message [9]– has been widely used. Many copy detection algorithms focus on copyright protection and propose methods that can be alternatives to watermarking. We do not study watermarking techniques here since copyright protection is not the main concern but one of the practical applications of image reuse detection. However, since the concept of reuse includes minor modifications of the original image, an efficient image copy detection algorithm can be a helpful starting point in building a framework for reuse detection. Thus, in this section we briefly overview some methods related to copy detection.

Copy detection is achieved by either matching images directly, or by seeking resemblance in some feature space. The feature space can incorporate different properties. For example, Kim [7] uses an ordinal measure of discrete cosine transform (DCT) coefficients to build a copy detection framework. In the framework, images are first resampled to 8x8 pixel reduced-size images, using intensity averaging. Then, a set of coefficients are calculated over the images using the DCT. The magnitudes of AC

coefficients are ranked to obtain a rank matrix, and the comparison of images is made by using the L_1 distance between the corresponding rank matrices. The author tests the system on five sets of modified copies of images, and the experimental results show accurate detection rates invariant to various types of color transformations and image filtering techniques. However, the system suffers from high dependence on the rotation of the images, which makes it less convenient for reuse detection since geometric transformations constitute a significant part of the manipulations in the reuse of pictorial elements. In order to handle rotations, Wu *et al.* [10] adopt an elliptical partitioning strategy, which results with superior performance in matching rotated images and similar performance in other transformations. Xu *et al.* [11] use multi-resolution histograms for image copy detection. Although their experiments show better results addressing the geometric distortions compared to the work of Kim [7] and Wu *et al.* [10], the system still does not seem to have an acceptable tolerance to cropping and rotation for reuse detection.

An approach that is more efficient in partial image copy detection is the use of local descriptors instead of extracting features from the whole image. Local descriptors such as Scale Invariant Feature Transform (SIFT) [12] and Speeded Up Robust Features (SURF) [13] can handle the geometric transformations with their invariance to scaling, rotation, and translation. In [5], Ke *et al.* propose a near-duplicate and sub-image retrieval system (see Figure 1.1), which uses PCA-SIFT [14], which is a relatively compact variant of SIFT, as a local descriptor in order to detect modified –particularly cropped– versions of original images. PCA-SIFT reduces the dimensionality of the SIFT descriptor from 128 to 36 by using a precomputed eigenspace –which is generated using a set of natural scene images– for feature projection. They use a dataset that is generated by applying 40 different transformations to a set of images downloaded from an online art gallery, and their experimental results display very high precision and recall rates.

Zhou *et al.* [15] address the partial-duplicate image search problem, which is a

Figure 1.1. Sub-image retrieval example in the work of Yan Ke *et al.* [5]: Query image (left), retrieved images (middle and right) (Figure from [5])

Figure 1.2. Partial-duplicate image search example in the work of Zhou *et al.* [15] (Figure from [15])

very similar case to sub-image retrieval and reuse detection. Similar to the content based image retrieval systems that we review in Section 1.2.2, they use the bag-of-visual-words (BoW) with SIFT descriptors. In BoW model, each image is represented by a bag-of-visual words, where the visual words are generated by quantizing the feature space that consists of feature sets that are extracted from a corpora of images (see Section 3.5). In the vector quantization stage, Zhou *et al.* assume that the query and duplicated images share similar descriptors and orientation values. Hence, in addition to descriptor vectors they also quantize the orientation values of keypoints. Their proposed approach seems to be an effective way to retrieve partially duplicated images (see Figure 1.2). The authors propose a straightforward solution to the rotation invariance problem, which is to run the query with different orientation values and the same descriptors. Although changing the rotation value is computationally much cheaper than rotating the image, querying every possible orientation might not be feasible for complete rotation invariance.

Another keypoint based approach is proposed by Zhao *et al.* [8] for near-duplicate image detection. The authors propose a two-stage detection scheme, where the first stage consists of selection of a small set of candidate keypoints using the BoW model, and the second stage involves keypoint matching for the candidate keypoint pairs. They report that their two-stage detection scheme speeds up the matching process without significant degradation in matching accuracy.

1.2.2. Content Based Image Retrieval

Content-based image retrieval (CBIR) systems differ from copy detection systems in that they usually focus on the retrieval of similar or related images in terms of color or texture based features rather than detection of near-duplicate images. In CBIR, images are usually –but not necessarily– evaluated under n classes, such as *animals, buildings, people, plants, etc.*, and a match is considered correct if the images fall in the same class.

As the reused images usually fall between the categories of duplicate and related

images, recent CBIR systems are reviewed here as a part of the relevant literature. Comprehensive literature reviews on CBIR are available [1, 2]. In this section we review only the work that target problems that show parallels to reuse detection in digital artwork.

Many early CBIR methods used global features [16, 17], which are generally computed over images in a uniform way. However, the CBIR systems that use low-level features usually suffer from the *semantic gap*, which is defined as "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." [1] In order to shrink the semantic gap, region-based and keypoint-based approaches have been used. As an example, Carson *et al.* [18] propose a region-based image retrieval algorithm, *Blobworld*, which automatically segments images into regions called *blobs* and compares images based on features extracted from these regions. The authors evaluate the Blobworld algorithm on an image dataset of 10,000 images, using the color and texture histograms that are computed over entire images as a baseline method for comparison. They show that the Blobworld algorithm performs better than the baseline approach.

As a higher-level way of image representation, local descriptors detect salient interest points on images and represent images by the features that are extracted from local regions the keypoints. Local descriptors are commonly integrated into the bag-of-visual-words (BoW) paradigm for CBIR since local features usually have a very high and they are computationally expensive to compare. The use of SIFT features in BoW (see Section 3.5) is common for higher-level semantic retrieval tasks [19–22]. Clustering can be a bottleneck in a BoW-based image retrieval system, since it is a computationally difficult problem. In order to speed-up the vocabulary building step, two fundamental strategies can be adopted: reducing the dimensionality and tweaking the clustering step. To reduce the dimensionality, in [22], a small and useful subset of SIFT features are selected by an unsupervised preprocessing procedure, and only the resultant features are used in order to obtain a more compact and distinctive set of descriptors. To improve the clustering step, Philbin *et al.* [21] use SIFT features for object retrieval and study scalable clustering methods, which can be used for building

an improved visual vocabulary, namely, approximate k-means and hierarchical k-means.

As an alternative to the standard BoW model, Hörster *et al.* [20] use Latent Dirichlet Allocation (LDA) [23], which is a generative probabilistic model, in the representation of images. The authors use the SIFT features for feature extraction. For visual word computation, they merge k-means clustering results on multiple non-overlapping feature subsets. They evaluate their system on an image dataset of over 246,000 images downloaded from the photo sharing website Flickr [24]. They report that their system outperforms previous methods, such as probabilistic latent semantic analysis based approaches. Based on the results of their evaluation, the authors assert that their framework is suitable for large-scale datasets.

1.2.3. Saliency Estimation

Visual saliency can be defined as the distinctiveness of an element which makes it pop out of its environment. There are two main types of factors that determine saliency: *bottom-up* and *top-down*. Bottom-up saliency solely relies on low-level perceptual information, such as high contrast to surrounding regions in terms of color and shape. When a region is different from its neighbors in an image, it attracts our attention. In addition to the perceptual information, *top-down* cues, such as looking for an object in a specific color, can also determine visual saliency [25, 26].

Visual saliency information can be useful for estimating the regions that are more likely to be reused in digital artworks. Since we do not have user-defined *top-down* cues for every piece of digital art, we focus only on *bottom-up* saliency, and we briefly overview some of the saliency detection methods.

In *bottom-up* saliency estimation, one of the most well-known early methods is presented by Itti *et al.* [27], where the authors propose a biologically inspired model based on the architecture in [28]. The model generates a topographical map of saliency for any image without any prior knowledge about the content, using color, intensity, and orientation based cues.

In a more recent work, Hou *et al.* [29] propose a method that produces a saliency map by analyzing the log-spectrum of the input image. Hou *et al.* use Inverse Fourier Transform in order to generate the saliency map. In another *bottom-up* approach, Bruce and Tsotsos [30] present an information maximization based saliency model of overt attention –which refers to an observable shift in sensory organs (i.e. eye movements).

To use *top-down* cues in saliency estimation, Judd *et al.* [31] present a dataset that includes eye tracking experiments and propose a supervised model that combines *top-down* and *bottom-up* information by training a classifier from human eye tracking data. The authors use different levels of features to analyze their data. They use low-level features such as color, intensity, and orientation based cues; a horizon level predictor; and high-level features such as face and person detectors.

In applications of image saliency estimation, there are several examples of use of saliency for object segmentation. Achanta *et al.* [32] use a simple k-means algorithm after saliency calculation for segmenting objects. Fu *et al.* [33] propose an automatic segmentation method called Saliency Cut, which incorporates saliency detection and graph-cuts. Cheng *et al.* [34] make use of Felzenszwalb *et al.*'s algorithm [35] for segmentation. They compute the saliency over segmented regions of an image and over corresponding saliency maps, and apply thresholding to the saliency maps to extract objects.

We use Cheng *et al.*'s method [34] for segmentation in the salient patch extraction step in the AISP algorithm that we describe in Section 3.7, since it provides saliency information at a region level. The implementation details of the method are explained in Section 3.6.

1.2.4. Digital Artwork in Computer Vision

Computer analysis of digital artwork involves computer vision and machine learning based tasks, including feature extraction, comparison, classification, and clustering problems. More specifically, the tools of computer vision are used for artist identifi-

cation [36], artistic image classification [37], art collection visualization [3], similarity analysis [38,39], authenticity assessment [40], and aesthetic evaluation of paintings [41].

For the artist identification and artwork classification problems, Johnson *et al.* [36] describe three wavelet-based approaches on brush analysis and artist identification, and they evaluate the methods on a dataset of 101 paintings. In [37], Carneiro *et al.* present an artistic image dataset of 988 expert-annotated images and evaluate several retrieval methods, including the BoW model and their extension of the inverted label propagation method [42].

For artwork collection visualization, Buter *et al.* [3] have implemented a toolkit, which uses a variety of features to characterize images, including color statistics, edge and corner densities, saliency-based statistics, and number of faces on images. For classification, Buter *et al.* use k-nearest neighbor, naive Bayes, nearest mean, and linear support vector machine classifiers, through which images are automatically classified by the artist who produced the image. Given two sets of images by different artists, classifiers are used to visualize image sets in a subspace that maximizes class separation.

For similarity analysis, Graham *et al.* [38] apply multidimensional scaling on the mean similarity data, which is based on human ratings for similarity of pairs of paintings, and they observe the correlation between the human ratings and basic image statistics. They conclude that basic image statistics can be a good estimator of human-perceived similarity, especially for landscape painting.

The work of Shamir *et al.* [39] is another example of unsupervised computer analysis of artworks. They use a dataset of 994 paintings including the work of 34 different painters, where they measure similarities between the paintings of different artists using shape, texture, edge, and color based features. Based on their experimental results, they assert that artists can be grouped by their artistic movements automatically by computer analysis of artworks.

Other problems that have been studied in computer analysis of artwork include

authenticity and aesthetic evaluations. Berezhnoy *et al.* [40] address the authenticity problem and perform color and texture analysis on the paintings of Vincent van Gogh. By using similarity, it becomes possible to detect whether a new work sufficiently resembles the known works of van Gogh to be classified as one. Li *et al.* [41] study the aesthetic evaluation of art. They analyze images using global features (color and edge distributions, brightness, and blurriness of images) and color-based features computed over different regions of images, and they evaluate their methodology using human consensus based ground truth.

1.3. Challenges

Reuse detection in digital artwork is a high-level semantic task, which can be challenging even for humans. Unlike image copy detection, in the case of image reuse, pictorial elements in images can be reused in any scale and amount. A small object in an image can constitute a major part in another image that reuses the source image. In most cases, global descriptors perform poorly in detecting partial correspondences. Compared to global descriptors, local descriptors are more robust to occlusions and geometric transformations. However, they generally produce high dimensional representations of images, which are not very suitable for retrieval tasks unless some further processing is applied. Although the BoW approach alleviates the dimensionality problem by representing images by the count of quantized keypoint descriptors, the vector quantization step in BoW puts a heavy load on the processor and reduces the scalability of the system. In addition, the types of reuse and modifications can vary for different artists and genres of digital art, making it more difficult to develop a global system that works well for almost all cases. Another challenge to reuse detection is that images can have similar contents without reusing parts from each other. For example, a famous architectural structure can be depicted by several artists even if they do not influence each other. An ideal image reuse detection system should be able to detect even a small amount of reuse without retrieving false positive images.

1.4. Contributions

In this work we propose a taxonomy of reuse in digital artwork and evaluate image description methods for different types of reuse. We aim to discover the methods that can be useful for detecting different types of reuse and manipulations. We also introduce one artificial and one natural image reuse dataset in order to test the retrieval performances of the evaluated methods.

1.5. Outline of the Thesis

The rest of the thesis is organized as follows: Chapter 2 describes our proposed taxonomy of types of reuse and manipulations. Chapter 3 describes the methods that we employ in reuse detection. Chapter 4 introduces the datasets and presents the experimental results. Finally, Chapter 5 gives conclusions and discusses possible future work.

2. TAXONOMY

In digital artworks, visual elements can be modified and reused in numerous ways such as using as a background, mapping as a texture or using the pictorial elements as a whole in a modified version of the artwork. Furthermore, reuse of visual elements can include various types of image manipulations such as scaling, rotation, color manipulation and alpha blending. In this chapter, we propose a classification for different types of reuse (Section 2.1) and image manipulations that are observed in reuse (Section 2.2), using 1200 annotated images in our natural dataset (Section 4.1).

2.1. Types of reuse

A taxonomy of the various types of reuse makes it easier to observe the different properties of modifications and discuss the effectiveness of proposed methods on different branches of reuse. Using the natural dataset that we describe in Section 4.1, we have categorized the possible types types of reuse in four main branches: partial reuse, direct reuse, use as a background, and remake/inspiration (Figure 2.1).

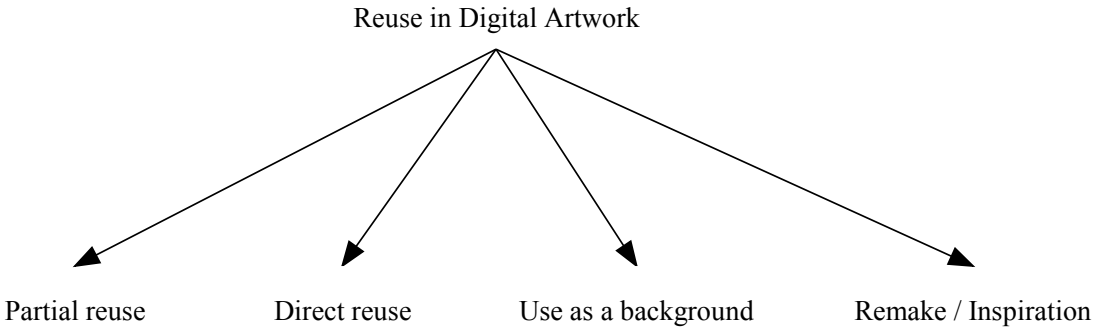


Figure 2.1. A simple taxonomy of types of reuse

Partial reuse denotes the use of a selected area of an original image and superimposition of the selected patch on another image. In other words, in this type of reuse the source image is used as an object in the destination image. By definition, the im-

ages in this category usually reuse two or more images and may include alpha blending, simple geometric and affine transforms such as translation, rotation and scaling. An example of partial reuse is shown in Figure 2.2 where the left image[1] is used in the right image [2] .

Figure 2.2. An example of partial reuse: the flower on the left image is used in the right image

Direct reuse refers to the use of an image as a whole and may involve manipulations such as insertion or removal of objects, addition of frames or captions, color and texture filters, and background manipulations. Generally, the images in this category either make use of only one source image or have a source image that constitute the vast majority of the destination image. In essence, direct reuse is the case that either the work is based on one image, or the whole image is used where a small amount of cropping is acceptable. An example of direct reuse is shown in Figure 2.3 where the left image[3] is used in the right image [4] .

[1]http://fa-stock.deviantart.com/art/Water-lily-3444-110882698
[2]http://cold-malina.deviantart.com/art/fairy-lily-329416909
[3]http://rafalhyps.deviantart.com/art/White-Cat-FOR-FREE-83189707
[4]http://vampiremindfreak.deviantart.com/art/Painted-life-346800877

Figure 2.3. An example of direct reuse: original image (left), modified image (right)

Figure 2.4. An example of use as a background: original image (left), modified image (right)

Figure 2.5. An example of remake: original image (left), modified image (right)

Use of an image as a background is especially very common in scenic images. A typical example in this category is shown in Figure 2.4 where the left image[5] is used as a background in the right image [6] .

The branch of remake includes remake or inspirational use such as paintings, sketches, and comics based on a source image. In most cases the source image is depicted in another medium and scanned afterwards. Using different type of media makes it harder to detect correspondences due to significant amount of alterations in the texture, edges and colors. The case of inspirational use is even more challenging to detect since the referred artwork may not contain any direct copies of pictorial elements. An example of remake is shown in Figure 2.5 where the left image[7] is used as a background in the right image [8] .

We have collected a database of N images from the internet, containing original and reused images. We refer to this database as the *natural dataset*, as opposed to the *artificial dataset* which will be described later. The observation frequencies of different types of reuse in the natural dataset are summarized in Table 2.1. As we can see from the table, the frequencies do not sum up to 100% since one instance may fall into more than one categories in many cases. Specifically, the direct reuse and background categories have a considerable overlap, since background images are generally used as a whole without excessive cropping. In this classification only the direct and partial reuse categories are considered to be mutually exclusive.

Table 2.1. Observation frequencies of reuse and manipulation types.

Partial reuse	**Direct reuse**	**Use as a background**	**Remake**
27%	47%	44%	6%

[5]http://wyldraven.deviantart.com/art/Moon-STOCK-147754760
[6]http://a7md3mad.deviantart.com/art/The-Moon-Rises-352388527
[7]http://lugubrum-stock.deviantart.com/art/Lugubrum-stock-regaleira5-52478084
[8]http://tig3r-eye.deviantart.com/art/Lost-Green-Spaces-357104509

2.2. Types of manipulations

We made an analysis of the most common types of manipulations in image reuse: color manipulation, translation, texture manipulation, text overlay, rotation, aspect ratio change, transparency and alpha blending, mirroring, and duplication.

Color manipulations include brightness and contrast change, color replacement, hue and saturation shift, tint and shades, and color balance change. Let an image is represented in the HSV color space, with hue ($\mathbf{I_h}$), saturation ($\mathbf{I_s}$), and brightness ($\mathbf{I_v}$) color channels. A color manipulation can be expressed as,

$$\mathbf{I_h} = a\mathbf{I_h} + d \tag{2.1}$$

$$\mathbf{I_s} = b\mathbf{I_s} + e \tag{2.2}$$

$$\mathbf{I_v} = c\mathbf{I_v} + f \tag{2.3}$$

where a, b, c, d, e, and f are constants.

Color manipulations are found to be the most popular transformations in our dataset with a 60% frequency of observation.

Translation is defined as changing the pixel locations of a reused area on an image, which is a more inclusive description than the definition of translation in geometry. Given a matrix $\mathbf{X}$ that lists the pixel positions of the pixels that lie in the reused area in an image, we can define translation as follows:

$$I_d(\mathbf{X} + \mathbf{o}) = I_s(\mathbf{X}) \tag{2.4}$$

where $\mathbf{I_o}$ and $\mathbf{I_d}$ are the source and destination images, respectively, and $\mathbf{o}$ is an offset vector that consists of horizontal and vertical offset values. Note that in our notation parentheses right next to the matrices denote indices, and boldface letters are not used

when the matrices are accessed by their indices.

A reused image is labeled as translated unless the pixel positions are preserved when the source and destination images are scaled to the same size. Thus, transformations that violate that property such as aspect ratio change, and mirroring are also considered as subsets of translation. Translation has been observed in 52% of the instances in the dataset.

Texture manipulations are the transformations that alter the texture of the image, including (excessive) blurring/sharpening, overlaying/embossing texture, and texture effects such as glowing edges and mosaic tiles effects. This category include a variety of operations including linear and non-linear image filters. More than 22% of the instances in the dataset involve texture manipulations.

Text overlay is also a frequently encountered type of manipulation with a 15% frequency of observation in the dataset. Examples of text overlay include image captions, poster and flyer designs. Small-font text that lies on the margins (i.e. author signature) is not considered text overlay in this classification and not included in this category.

Other types of modifications that we examine are observed relatively less than the major ones. Rotation, aspect ratio change, alpha blending, mirroring, and duplication are observed in 6%, 6%, 4%, 4%, and 4% of the instances in the database, respectively. The observation frequencies of different modifications are summarized in Table 2.2.

Rotation (with translation) can be formalized as follows:

$$I_d(\mathbf{R} \times \mathbf{X} + \mathbf{o}) = I_s(\mathbf{X}) \tag{2.5}$$

where $\mathbf{R}$ is a rotation matrix:

$$\mathbf{R} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \tag{2.6}$$

for a rotation angle θ.

Aspect ratio change occurs in case of non-proportional scaling, which can be defined as

$$I_d(\mathbf{S} \times \mathbf{B}) = I_s(\mathbf{B}) \tag{2.7}$$

where $\mathbf{S}$ is a scaling matrix:

$$\mathbf{S} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \tag{2.8}$$

in which s_x and s_y are horizontal and vertical scaling factors and $s_x \neq s_y$.

Alpha blending refers to partially transparent use of an original image and defined as

$$I'_d(\mathbf{B}) = \alpha I_s(\mathbf{B}) + (1 - \alpha) I_d(\mathbf{B}) \tag{2.9}$$

where α is an opacity parameter that takes values between 0 (completely transparent) and 1 (completely opaque).

Mirroring denotes horizontal or vertical flipping, which is defined as:

$$I_d(\mathbf{T} \times \mathbf{B} + \mathbf{o}) = I_s(\mathbf{B}) \tag{2.10}$$

where **F** is a transformation matrix for mirroring:

$$\mathbf{F} = \begin{bmatrix} t_x & 0 \\ 0 & t_y \end{bmatrix} \tag{2.11}$$

For a horizontal flip the parameters are assigned as $t_x = 1$ and $t_y = -1$, whereas for a vertical flip they are assigned as $t_x = -1$ and $t_y = 1$. An offset **o** is added to positions in order to realign the image.

Duplication implies multiple use of one element in an original image (i.e. using a tree multiple times to create a forest image).

Table 2.2. Observation frequencies of modifications.

Color manipulation	60%
Translation	52%
Texture manipulation	22%
Text overlay	15%
Rotation	6%
Aspect ratio change	6%
Alpha blending	4%
Mirroring	4%
Duplication	4%

In this taxonomy a general classification scheme is proposed for different types of modifications. The transformations that we classify here are not mutually exclusive, on the contrary, combinations of different modifications are observed in most cases. More detailed information about the distributions of different types of reuse and modifications in the natural dataset is given in Section 4.1.2.

3. METHODOLOGY

In this chapter, we describe the methods that we use for image reuse detection. We first summarize popular image description methods that are used in matching-based tasks in computer vision, such as content-based image retrieval, image copy detection, and object recognition. Then, we explain a saliency-based segmentation approach in Section 3.6, which constructs a basis for our proposed algorithm, the affine invariant salient patch descriptors. We describe the implementation details of our algorithm in Section 3.7. Finally, we explain our matching methods in Section 3.8.

3.1. Color Histograms

Color histograms describe the probability distribution of colors on an image but do not contain locations of colors or any other spatial information. An $n-bin$ histogram $\mathbf{H}$ defined as follows:

$$\mathbf{H} = \{p_1, p_2, p_3, ..., p_n\} \tag{3.1}$$

where p_i denotes the occurrence frequency of the color intensities that lie in $[\frac{i-1}{K/n}, \frac{i}{K/n})$ range, and K is the number of colors in the image.

In addition to the histogram of the whole image, we also compute histograms over the cells on a 2×2 grid on the image which sums up to five color histograms for each image including the overall histogram. In this way we add simple spatial information to the descriptor in order to provide a better description of an image. As a predictable drawback, this approach reduces the rotation and translation invariance of the descriptors.

In this work, we choose the number of bins as eight, and we compute histograms for each of the three color channels of images. For each image, we obtain 120-dimensional ($\#bins \times \#channels \times \#cells = 8 \times 3 \times (1 + 4)$) feature vectors in

total.

3.2. Histograms of Oriented Gradients

Histograms of Oriented Gradients (HOG) [43] define edge characteristics of images using the gradient orientation histograms of cells that form the image. The implementation of HOG can be summarized in five main steps; division of the image into fixed sized spatial cells, gradient computation, edge angle and magnitude computation, accumulation of weighted votes for gradient orientation over each cell, and normalization of contrast over overlapping blocks, which are defined as larger spatial regions that consist of cells.

Gradients are computed in a straightforward way by using a discrete derivative filter as:

$$G_x(x, y) = I(x + 1, y) - I(x - 1, y) \tag{3.2}$$

$$G_y(x, y) = I(x, y + 1) - I(x, y - 1) \tag{3.3}$$

where $\mathbf{I}$ denotes the pixel intensity matrix of the image, x and y indicate the pixel indices, and $\mathbf{G_x}$ and $\mathbf{G_y}$ denote the horizontal and vertical gradient components of the image.

Then, the magnitude and orientation values are calculated for each pixel as:

$$M(x, y) = \sqrt{G_x(x, y)^2 + G_y(x, y)^2} \tag{3.4}$$

$$T(x, y) = \arctan(G_y(x, y)/G_x(x, y)) \tag{3.5}$$

where $\mathbf{M}$ and $\mathbf{T}$ correspond to the magnitude and orientation matrices, respectively.

The descriptors are obtained by counting the magnitude-weighted orientation votes of each pixel within each cell. The resultant orientation histogram characterizes the edges in the cells with the weighted occurrences of orientations that fall in predefined angle intervals.

In the final step, values of histograms within blocks –which are defined as spatial patches consisting of 2×2 cells– are locally normalized to the range $[0, 1]$ in order to obtain enhanced descriptors robust to local variations in gradient strengths.

3.3. Scale Invariant Feature Transform

Scale Invariant Feature Transform (SIFT) [12] is a local descriptor that is widely used for image matching tasks. SIFT is able to achieve scale and rotation invariance and also partially invariant to changes in illumination and 3D camera viewpoint.

The procedure can be summarized in four stages; scale-space extrema detection, keypoint localization, orientation assignment, and computation of keypoint descriptors. The first three steps form the keypoint detection procedure, and the last step consists of characterizing the keypoints that are detected in the previous steps. The keypoint detection method can be replaced by other strategies such as dense sampling, which will be explained later (see Section 4.3).

A scale space is constructed by blurring out the original image progressively by convolving with a Gaussian function to get a set of images with different levels of blurriness, each of which is called an octave. Then, the original image is resampled to half of its size to generate the second octave, and the procedure is iterated to construct all scales in each octave. Then, the consecutive Gaussian images are subtracted in order to get the Difference of Gaussians, which approximates the Laplacian of Gaussians efficiently. In order to find extrema points, each sample point is compared with its 26 neighbors in the scale space and labeled as a candidate keypoint only if it is greater

than all of its neighbors or smaller than all of them.

In order to find accurate keypoint locations a 3D quadratic function is fitted to the local sample points by using a Taylor expansion of the scale space function, and the extrema are located by getting the derivative of this function and setting it to zero. Later, this function is also used for elimination of low contrast keypoints and edge responses.

An orientation histogram with 36 bins, each covering an angle of 10 degrees, is used for a patch of pixels of the Gaussian blurred image around the keypoint. The use of orientation histograms shows some similarity with the Histograms of Oriented Gradients [43] approach. The Gaussian image with the closest scale is selected by using the scale of the keypoint in order to provide scale invariance. Samples are weighted by their gradient magnitudes and a Gaussian-weighted circular window. The peaks are assigned as the dominant orientation of the keypoint. Then, the other orientations and coordinates are rotated relative to the dominant orientation in order to accomplish rotation invariance. Additionally, any other local peak within 80 percent of the dominant orientation is used to create another keypoint with that orientation.

Finally, a descriptor is created for each keypoint as identification of the corresponding keypoint. Firstly, gradient magnitudes and orientations are computed for each sample point in the keypoint patch. Then, an orientation histogram is created by letting each point to vote, weighted by a Gaussian window. Figure 3.1 illustrates the computation of the keypoint descriptor.

In order to compensate for the illumination changes, the gradient magnitudes are normalized by thresholding the values greater than 0.2 in the unit feature vector, and then renormalizing the vector to the unit length so that the distribution of orientations gains more emphasis than large magnitudes.

An example of matching reused digital artwork[9] [10] with SIFT descriptor is illustrated in Figure 3.2.

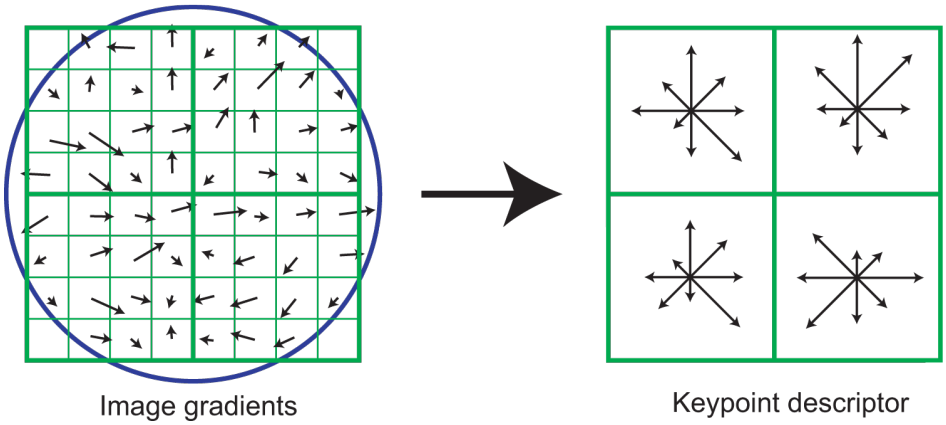


Figure 3.1. Keypoint descriptor computation (Figure from [12])

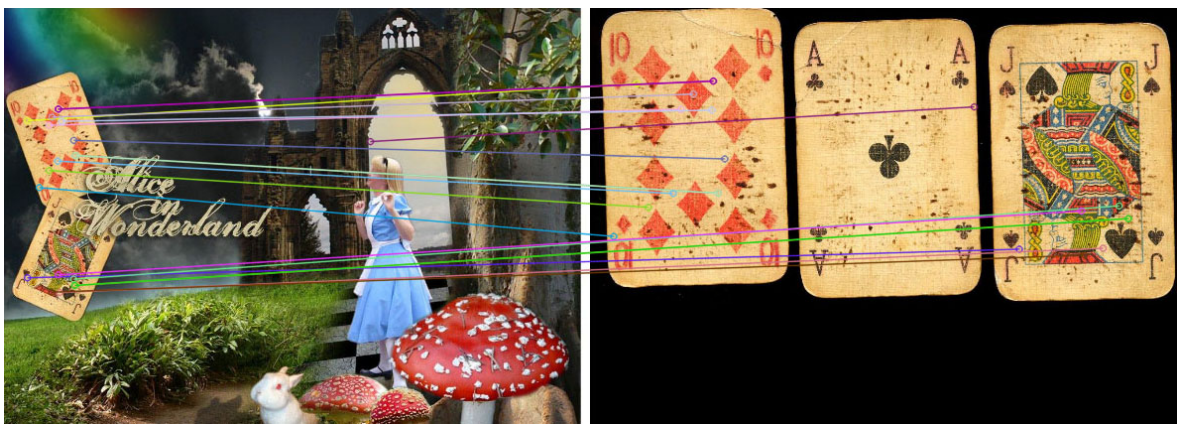


Figure 3.2. Illustration of reused image matching using SIFT features

[9]http://koukla-loves.deviantart.com/art/Alice-In-Wonderland-71323393
[10]http://temabinastock.deviantart.com/art/Stock-000150-1076209

3.4. Color-based Variants of SIFT

In previous studies, various color-based extensions of the SIFT descriptor are proposed and used. For the image reuse detection problem, we chose and evaluated three variations of the SIFT descriptor; RGB-SIFT, OpponentSIFT, and C-SIFT; which are shown to give better overall performance as compared to many other color descriptors as reported in [44].

The RGB-SIFT descriptor computes the SIFT features over each of the RGB color channels instead of using a single intensity channel. Similarly, the OpponentSIFT descriptor uses two color channels (O_1 and O_2) and one intensity channel (O_3) that are defined in the opponent color space. The O_1, O_2, and O_3 channels are defined as follows:

$$O_1 = (R - G)/\sqrt{2} \tag{3.6}$$

$$O_2 = (R + G - 2B)/\sqrt{6} \tag{3.7}$$

$$O_3 = (R + G + B)/\sqrt{3} \tag{3.8}$$

The C-SIFT descriptor normalizes the color components, O_1 and O_2, by the intensity channel, O_3, in the opponent color space in order to increase invariance to intensity changes. The channels are normalized as in the following equations:

$$C_1 = O_1/O_3 \tag{3.9}$$

$$C_2 = O_2/O_3 \tag{3.10}$$

For the keypoint detection step, dense sampling is also used and evaluated as an alternative method to the sparse salient point detection in the SIFT descriptor. In dense sampling, every n^{th} pixel is marked as a keypoint in a uniform way, and the SIFT features are calculated for each keypoint.

3.5. Bag of Words Model

The bag-of-words (BoW) model was first proposed for information retrieval in text collections. In the domain of natural language processing, BoW is used for characterizing a document as a collection of words and their number of occurrences. Similarly, in computer vision, bag-of-visual-words model uses counts of visual words to represent images.

Bosch *et al.* [45] describe the four steps of BoW model as: (i) point of interest detection, (ii) computation of local keypoint descriptors over the detected interest points, (iii) quantization of descriptors in order to create a visual vocabulary, (iv) and computation of histograms of words, which is generated by counting the occurrences of visual words in each image. An illustrative figure showing the four steps of BoW is shown in Figure 3.3.

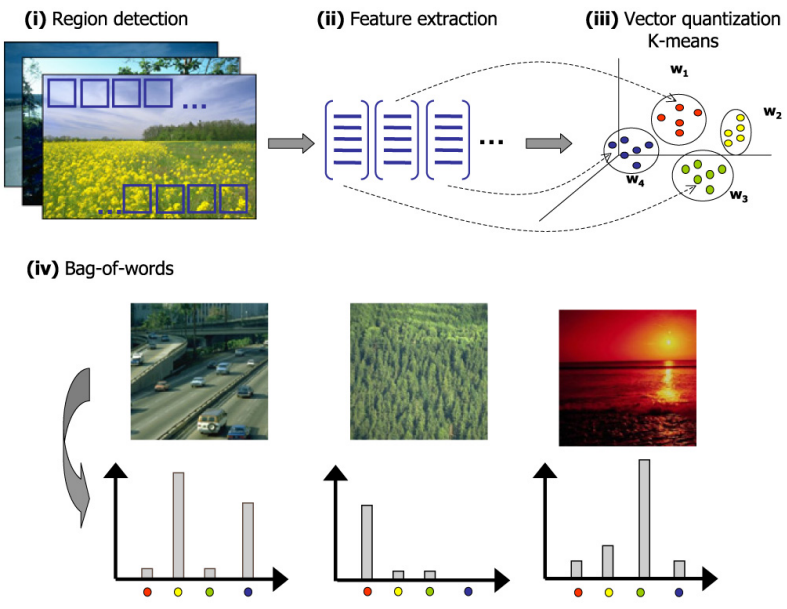


Figure 3.3. An illustration of the bag-of-words model (Figure from [45])

Note that step (i), (ii), and (iv) are called for each image, whereas step (iii) is done offline for a set of features. In this work, we use the SIFT descriptor for local

feature computation and k-means clustering algorithm for vector quantization in step (iii).

The BoW model enables us to represent images with a smaller number of features compared to local descriptors without any quantization. On the other hand, as van de Sande *et al.* report in [46], the vector quantization stage in visual vocabulary generation constitutes a bottleneck and decreases the scalability of the system.

3.6. Saliency Based Segmentation

Saliency of a region or a pixel on an image refers to its prominence and uniqueness relative to its neighbors. Intuitively, we can assume that more salient regions are more probable to be reused. In this work saliency detection is studied as a heuristic for reused area estimation and a tool for segmentation.

We adopt the *region-based contrast* (RC) approach of Cheng *et al.* [34] for salient region detection. The first step here is to segment the image into regions by using the graph-based image segmentation technique of Felzenszwalb and Huttenlocher [35]. After segmentation, saliency values of regions are calculated as,

$$S(r_k) = \sum_{r_k \neq r_i} w(r_i) D_r(r_k, r_i) \tag{3.11}$$

where $S(r_k)$ is the saliency value of region r_k, $w(r_i)$ is the pixel count of region r_i used for weighting and $D_r(r_k, r_i)$ is the color distance between regions r_k and r_i defined as,

$$D_r(r_1, r_2) = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} f(c_{1,i}) f(c_{2,j}) D(c_{1,i}, c_{2,j}). \tag{3.12}$$

Here, $D(c_{1,i}, c_{2,j})$ is the color distance between colors $c_{1,i}$ and $c_{2,j}$ in L, a, b color space, and $f(c_{1,i})$ and $f(c_{2,j})$ denote the frequencies (or probabilities) of the i^{th} and j^{th} colors in the first and second regions respectively among all the colors in the corresponding region, where the colors in each color-channel are quantized into 12 bins in order to

reduce the complexity. In the second part of the equation a simple color distance is used as a color contrast metric, and in the first part of the equation the probabilities of colors in PDF are used as weight functions to highlight the contrast between prominent colors.

In addition to the contrast-based functions, a spatial weighting term is used to emphasize the effect of a high contrast to closer regions, since contrast to distant regions is a relatively weak indicator of saliency. For example, a blue region can be considered salient when it has a red surrounding region even if the majority of the distant regions (i.e. image background) are also blue. Thus, saliency value for a region r_k is redefined as,

$$S(r_k) = \sum_{r_k \neq r_i} exp(-D_s(r_k, r_i)/\sigma_s^2) w(r_i) D_r(r_k, r_i) \tag{3.13}$$

where σ_s^2 is a parameter for adjusting the spatial weighting, which is set to 0.4 by default.

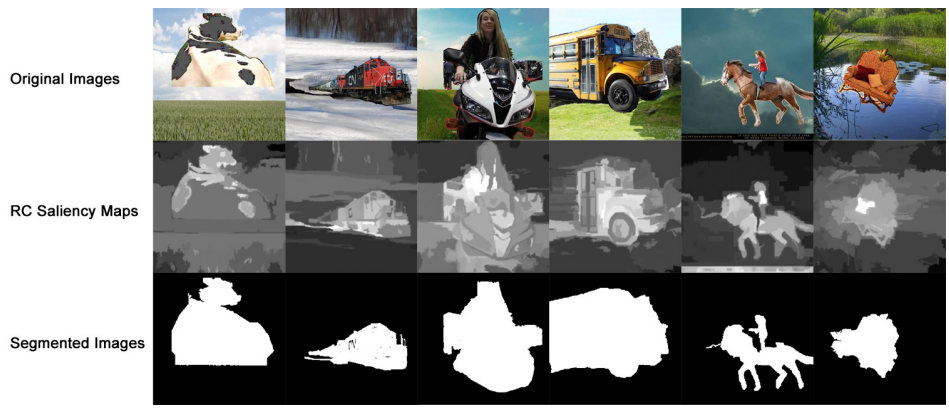


Figure 3.4. Example saliency-based segmentation results on our artificial dataset

In order to obtain the final segmentation masks for salient regions, a threshold is applied to the RC saliency maps. As explained in [34], an initial segmentation is obtained by applying a fixed threshold to the saliency map. Then, as provided by the

authors, the GrabCut [47] algorithm is used to enhance the segmentation iteratively.

Given sample instances in our artificial dataset (see Section 4.1) as input images, saliency map and segmentation examples are shown in Figure 3.4.

3.7. Affine Invariant Salient Patch Descriptors

In this section we explain our proposed algorithm, Affine Invariant Salient Patch (AISP) descriptors [48], for image retrieval and reuse detection. We can summarize our algorithm in three stages; salient patch detection, affine invariant feature extraction, and addition of global features.

3.7.1. Salient Patch Detection

In many cases of reuse, the foreground objects in source images tend to be more salient as compared to the background. Especially, stock images are good candidates for saliency-based segmentation since they are usually composed of a complex foreground object and a relatively simple background. Exploiting this feature, saliency maps are used in this method to estimate the location of the foreground object, assuming that there is only one major foreground object in each source image.

First, an RC saliency map is computed, and a segmentation mask is generated by applying a threshold to the saliency map. To recap, the segmentation process that was explained in Section 3.6 can be summarized as: computation of histogram-based saliency values, segmentation of the image into regions, and for each region distance-weighted average saliency computation.

Then, the segmentation mask is applied to the image, and the region of interest is extracted by cropping out the black margins (see Figure 3.5). The region of interest will be referred as salient patch in the rest of this chapter.

(a) Source Image (b) Salient Region (c) Salient Patch

Figure 3.5. Example of salient patch extraction [48]

3.7.2. Affine Invariant Feature Extraction

AISP descriptors achieve affine invariance by describing the salient region as a partitioned elliptical region. The first step in feature extraction is computing the best fitting ellipse to the salient patch. Image moments of the binary segmentation mask, which is computed by using the salient region detector that explained in the previous section, is used to compute the best fitting ellipse.

Let B be a segmentation mask, then the horizontal and vertical components of the image centroid are calculated as follows:

$$m_x = \frac{\sum_x \sum_y xB(x,y)}{\sum_x \sum_y B(x,y)} \tag{3.14}$$

$$m_y = \frac{\sum_x \sum_y yB(x,y)}{\sum_x \sum_y B(x,y)} \tag{3.15}$$

Then, the central moments of the binary image are found using the following

equation:

$$\mu_{ij} = \frac{\sum_x \sum_y (x - m_x)^i (y - m_y)^j B(x, y)}{\sum_x \sum_y B(x, y)} \tag{3.16}$$

Using the covariance matrix of the binary shape, the principal orientation of the binary image is computed as Equation 3.18, and we set the main orientation of the descriptor to θ_m.

$$C = \begin{bmatrix} \mu_{20} & \mu_{11} \\ \mu_{11} & \mu_{02} \end{bmatrix} \tag{3.17}$$

$$\theta_m = \frac{1}{2} \tan^{-1} \left(\frac{2\mu_{11}}{\mu_{20} - \mu_{02}} \right) \tag{3.18}$$

Then, the semi-major and semi-minor axis lengths of the best fitting ellipse are found using the following equations, which are described in [49]:

$$a = \sqrt{\mu_{20} + \mu_{02} + \sqrt{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2}} \tag{3.19}$$

$$b = \sqrt{\mu_{20} + \mu_{02} - \sqrt{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2}} \tag{3.20}$$

After calculating the fundamental variables that defines the properties for the best fitting ellipse, the salient patch is segmented into four equal breadth concentric elliptical tracks as shown in Figure 3.6.

The image is converted from RGB to HSV color space for further processing.

Figure 3.6. Illustration of AISP descriptors on an image [48]

Then, for each track on the salient patch, 8-bin color histograms are computed for each of the color channels, and an 8-bin edge orientation histogram is obtained by using the brightness channel.

The edge histograms are calculated using a similar strategy to the Histograms of Oriented Gradients [43] approach. For each elliptical track, pixels cast magnitude-weighted votes for their orientation.

An edge orientation histogram, $H(i, \theta)$, is defined as

$$\mathbf{t_i} = [\mathbf{x_i y_i}] \tag{3.21}$$

$$H(i, \theta_k) = \sum_j M(\theta_{k-1} < T(\mathbf{x_{i,j}}, \mathbf{y_{i,j}}) \leq \theta_k) \tag{3.22}$$

where $\mathbf{t_i}$ is a vector that contains the x and y indices of the pixels in the i^{th} elliptical track, θ_k denotes the upper boundary angle for the k^{th} bin ($\theta_0 = 0$), and $M(x, y)$ and $T(x, y)$ refer to the edge magnitude and aligned orientation of the corresponding pixel,

respectively, which are calculated as:

$$M(x,y) = \sqrt{G_x(x,y)^2 + G_y(x,y)^2} \tag{3.23}$$

$$T(x,y) = \tan^{-1}(G_y(x,y)/G_x(x,y)) - \theta_m \tag{3.24}$$

where G_x and G_y are the horizontal and vertical gradients defined by Equations 3.2 and 3.3. Figure 3.7 shows the edge orientations and magnitudes of the example salient patch (balloons) using a Quiver plot. In the example, the image is downsampled and θ_m is assumed to be zero for simplicity.

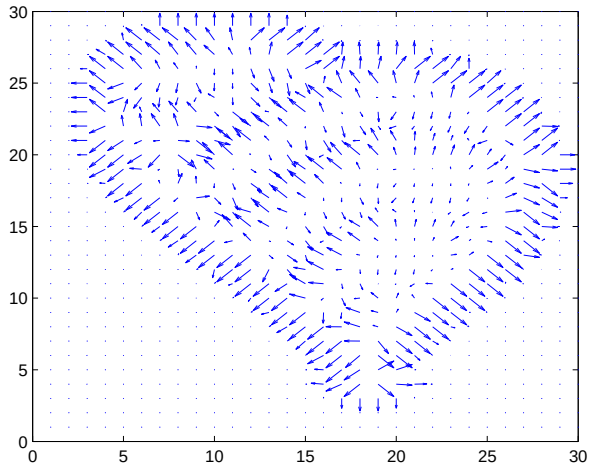


Figure 3.7. Quiver plot of the gradient image of the salient patch

As described by Equation 3.24, rotation invariance is achieved by simply subtracting the main orientation θ_m (Equation 3.18) from all orientation values in T. Since the main orientation is obtained by using the binary segmentation mask of the salient patch, the edge-based features that defined here are not robust to 180 degree rotations and flips. In his context, color and edge based features are both incorporated to the final AISP descriptor in order to let different types of features alleviate weaknesses of each other.

3.7.3. Addition of Global Features and Normalization

Achieving an optimal level of context independence for foreground objects is important in many cases of reuse. For example, when we search an image database with a query image that has a picture of a red car in a desert, in most cases we expect to see more instances of the red car reused in different contexts than the desert used with different objects in the retrieved images. However, if there are other images that reuse both the car and desert, it would be useful to retrieve them first. Furthermore, use as a background constitutes a large class in digital artworks, and even though a salient patch is a good heuristic of the region of interest it may fail to detect the actual region of interest, and in some cases the detected region might not even have an intersection with the reused region.

In this context, the AISP algorithm adopts a two-level feature extraction strategy and incorporates global features to the final descriptor, as well as the features extracted from the salient patch. Similar to the previous step, global features are extracted by computing color and edge histograms over the whole image, and the features are added to the descriptor. The global features are treated like an additional track in the descriptor and have the same weight as a single track.

Finally, all histograms are normalized to be a unit vector as:

$$\hat{H_j} = \frac{H_j}{||H_j||} \tag{3.25}$$

where $||H_j||$ is the norm of histogram H_j, $\hat{H_j}$ is the corresponding normalized vector.

The number of tracks and bins are set to four and eight respectively in the default descriptor, which results in a 160-dimensional feature vector for each image (See Equation 3.26).

$$\#dimensions = ((\#tracks + 1) \times \#histograms \times \#bins) \tag{3.26}$$

3.8. Matching

We use two different matching methods for different descriptors. For the color histograms, HOG, and AISP, we rank the candidate matches, V_c, according to their standardized Euclidean distances to the query image feature vector, V_q, in descending order. Standardized Euclidean distance, d_s, between N-dimensional vectors V_c and V_q, is defined by Equation 3.27, where each dimension is scaled by its standard deviation σ_i.

$$d_s(c,q) = \sqrt{\sum_{i=1}^{N} \frac{1}{\sigma_i^2}(V_c - V_q)^2} \tag{3.27}$$

For the SIFT descriptor we use the BoW model. First, we apply term frequency–inverse document frequency weighting before matching. Term frequency TF_{ij} of a visual word within an image is defined by the histogram of visual words that we use to represent the images in the BoW model. Term frequency can be expressed in mathematical terms as follows:

$$TF_{ij} = \frac{c_{ij}}{\sum_k c_{kj}} \tag{3.28}$$

where c_{ij} is the number of occurrence of i^{th} visual word in j^{th} image.

In order to accentuate the contribution of the words that occur less in a collection, which are likely to have more discriminative power, inverse document frequency defines word rarity as:

$$IDF_i = \log \frac{|D|}{|\{w_i \in D\}|} \tag{3.29}$$

where IDF_i is the inverse term frequency of a word w_i, $|D|$ is the number of instances in the collection, and $|\{w_i \in D\}|$ is the number of documents that the word w_i appears in.

4. EXPERIMENTS

In this chapter we first introduce two novel datasets (Section 4.1). Then, we explain our experimental setup in Section 4.2 and our preliminary experimental results in Section 4.3. We evaluate our methodology for the presented datasets in terms of matching accuracy (Section 4.4) and system requirements (Section 4.5).

4.1. Datasets

In this section we present two novel image datasets for image reuse. The first dataset consists of synthetically modified and combined images that aims to simulate different types of reuse in a systematic way. The second dataset is a natural dataset that consists of a collection of artwork crawled from the web. We use the artificial dataset to perform evaluation under controlled conditions and the natural dataset to assess the performance of the methods under more natural settings.

4.1.1. Artificial Dataset

In digital artworks, reused elements in images include multiple types of manipulations in most cases. In order to simplify evaluation, we use an artificial dataset that enables us to perform controlled experiments.

The artificial dataset is generated by using sets of foreground and background images and applying various transformations automatically in order to evaluate the effectiveness of different methods on different types of reuse. The dataset consists of a total of 4698 images where 1271 of the images are the source images, 1785 images are the reused versions of the source images, and the remaining 1642 images are the noise/orphan images that are processed among the source images but have no reused versions. Each modified image in the dataset has only one corresponding source image, whereas one source image may have more than one reused copy.

In the artificial dataset, we simulated two main types of reuse: partial and direct reuse. In the partial reuse category, five types of modifications are defined as: color change, rotation, aspect ratio change, blur, and translation with no transformation. We generated the synthetically reused images by superimposing random foreground objects on different background images with systematic transformations. We used the 2012 PASCAL Visual Object Classes [50] dataset to obtain a set of segmented foreground objects. For each type of transformation, we used a randomly selected subset of 255 images and their corresponding segmentation masks to extract the foreground area (see Figure 4.1). We selected the background images from the natural dataset.

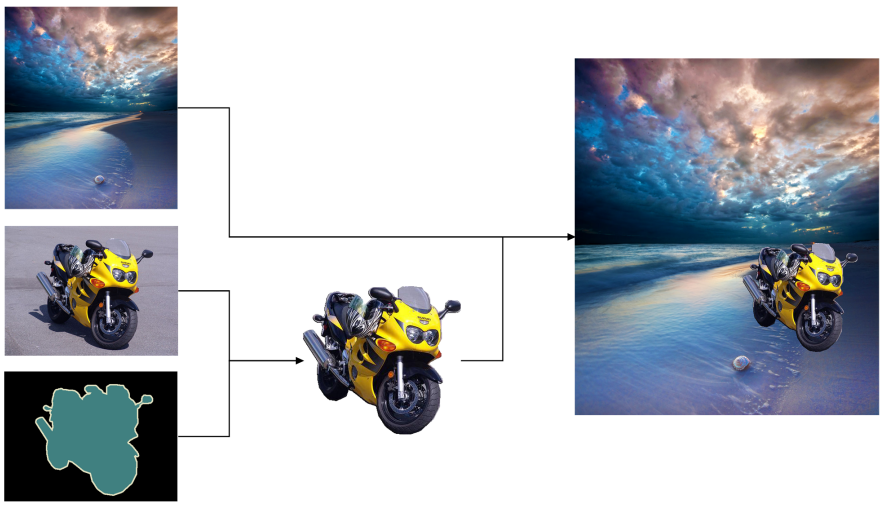


Figure 4.1. Artificial dataset generation for partial reuse

In the direct reuse category, only two types of modifications are defined, which are blur and color change, since the use of other transforms –translation, rotation, and aspect ratio change– in the direct reuse category would have blurred the distinction between direct and partial reuse. The images in the direct reuse category includes solely the transformed copies of the images in PASCAL VOC dataset and does not involve segmentation masks and background images.

We randomized the parameters for each type of transformation in order to prevent memorization of a linear transform with certain parameters as a result of overtraining.

For each image, we selected a random value for the specified transformation parameter within the following ranges: background image boundaries for translation, [0 90] degrees for rotation, [0.5 2] for aspect ratio, [0 10] sigma and 5-pixel radius for Gaussian blur, and for color change [0 50] percent circular value shift for each of the color channels. (see Section 2.2)

4.1.2. Natural Dataset

The natural dataset was based on an idea of Dr. Almila Akdag Salah to use stock images to explore diffusion of style and influence on *deviantArt*, the artwork sharing website. *deviantArt* [51] is a social network for sharing artworks, with more than 26 million registered users. These users upload some of their work under the category of *stock images*. These images are usually published under an open license and free to use by others. Therefore, they are ideal samples for reuse detection. We collect these images, as well as the images that use these stock images. We crawled more than 16,000 images from the most popular images in certain categories on *deviantArt*. We selected the *stock images* category as the root directory and downloaded the images under the following subcategories: *animals, food, nature, places, plants, and premade backgrounds*. An example image from each category is shown in Figure 4.2[11] [12] [13] [14] [15] [16] .

Many artists on *deviantArt* notify authors of the source images that they reuse via image comments. We made use of the comments for each source image to find the images that reuse a source image and to associate the images with their reused counterparts. We used regular expressions to follow the correct types of links since the comments may include plenty of irrelevant links in addition to the relevant entries. Our image crawler uses a depth limited recursive search to download reused images (children images) and relates them to their source images (parent images).

[11] http://nickistock.deviantart.com/art/RAWWWR-81716482
[12] http://4k1.deviantart.com/art/Have-A-Bite-159633970
[13] http://serapstock.deviantart.com/art/Red-Morning-I-100397186
[14] http://enchanted-stock.deviantart.com/art/Stock-032-55000908
[15] http://sourcow.deviantart.com/art/Flower-STOCK-55502670
[16] http://little-spacey.deviantart.com/art/Cosmic-Beach-Premade-Background-280452929

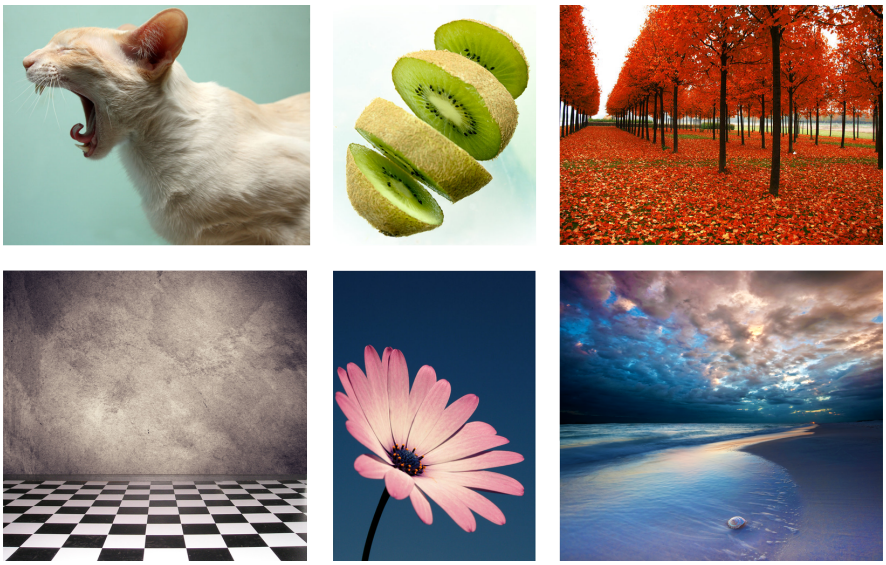


Figure 4.2. Example images from the natural dataset

We annotated a total of 1200 images by labeling each image for the reuse and manipulation types that we described in Chapter 2. An annotation tuple includes the id numbers of the image and its source, a link to the image, types of reuse that the image belongs to, and the types of modifications that are observed in the image. A block of example annotations are given in Table 4.1, where the images #2-6 reuse the image #1.

For the annotated dataset, we selected 200 images in each of the six subcategories, and we excluded any parent image with no children. Since the dataset is generated by crawling the most popular images, most parent images have several children, where the average number of children per parent image is found as 7.33.

Since the types of reuse and manipulations are not equally popular, the numbers of images that belong to certain types and categories are not equally distributed. The numbers of images for all six categories, four types of reuse, and nine types of modifications are shown in Table 4.2.

Table 4.1. Example tuples of annotations.

ID	1	2	3	4	5	6
Parent ID	0	1	1	1	1	1
Link	-	-	-	-	-	-
Partial reuse	0	1	1	0	0	0
Direct reuse	0	0	0	1	0	0
Remake	0	0	0	0	0	1
Use as a background	0	0	0	1	1	0
Color manipulation	1	0	1	1	1	0
Translation	0	1	1	0	1	0
Texture manipulation	0	0	1	1	0	0
Text overlay	0	0	0	1	0	0
Rotation	0	1	0	0	0	0
Aspect ratio change	0	1	0	0	0	0
Alpha blending	0	0	1	0	0	0
Mirroring	0	0	0	1	0	0
Duplication	0	1	0	0	0	0

Table 4.2. Numbers of images for different types of reuse and manipulation.

	Places	**Animals**	**Food**	**Premade**	**Nature**	**Plants**	**Sum**	**%**
Partial Reuse	28	104	92	8	15	76	323	27
Direct Reuse	136	35	49	152	113	83	568	47
Remake	4	33	23	0	4	11	75	6
Use as a background	155	1	3	176	163	28	526	44
Color Manipulation	145	100	91	136	121	124	717	60
Translation	101	113	106	94	98	116	628	52
Texture Manipulation	63	25	14	50	43	69	264	22
Text Overlay	32	25	14	40	43	26	180	15
Rotation	4	10	34	3	5	18	74	6
Aspect Ratio Change	13	9	16	17	8	7	70	6
Alpha Blending	9	6	7	1	8	19	50	4
Mirroring	6	15	4	3	14	5	47	4
Duplication	5	7	15	2	3	14	46	4

4.2. Experimental Setup

In order to assess the feature descriptors for detecting image reuse, we design several experiments. In our experiments, we separate the database into several partitions. The gallery contains the set of original images. The test set contains images that reuse images from the gallery. In each experiment, the usefulness of the descriptor is evaluated with a retrieval paradigm. Given a query image I, the feature descriptor is used to rank the gallery images in terms of probability of reuse in the query image. We use the cumulative match characteristic (CMC) curve to compare different descriptors. In the CMC curve, rank n in the x axis denotes the retrieval accuracy over the entire set, when retrieving the original reused image among the first n candidates is accepted as a match.

4.3. Preliminary Experiments

We chose the keypoint sampling strategy and number of visual words experimentally. For the all images in the natural dataset, we ran the standard SIFT descriptor with two sampling strategies: sparse salient keypoint detection, and dense sampling. For sparse sampling, we used the default keypoint detector of SIFT, and for dense sampling, we sampled every 8^{th} pixel. Then, we generated a BoW codebooks with different vocabulary sizes. We selected the number of clusters for the BoW model as 160, 320, 640, 1280, and 2560, which are factors of the default size of the AISP descriptor. The first 20 rank retrieval accuracies for the above-mentioned parameters are shown in Figure 4.3. In this experiment, the gallery contains 144 images, and 1056 query images were evaluated.

By observing the results of our experiment, dense sampling seems to work better for the reuse detection problem. This result also shows some parallels with the results of Nowak *et al.* [52], as the authors state that dense sampling provides better results than more complicated methods for image classification –although it is not the same problem that we study here. Based on our preliminary experimental results, we selected uniform dense sampling as our default sampling strategy.

As expected, the performance increases with increasing number of clusters (visual words). However, increasing the number of clusters does not improve the performance significantly after some point as observed in the figure. Thus, we selected the number of clusters as 1280 in the rest of the experiments.

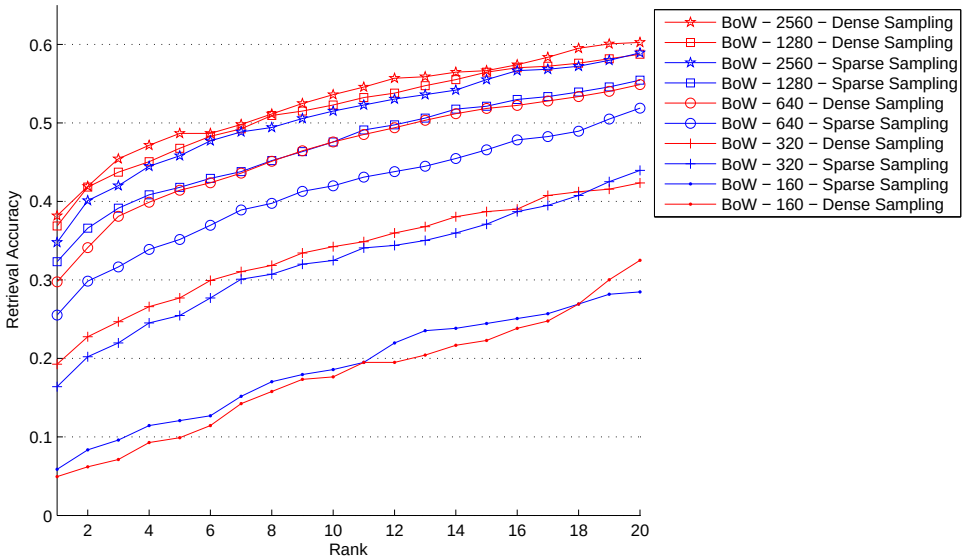


Figure 4.3. Cumulative matching accuracies for BoW model with different parameters

4.4. Matching Results

In this section, we evaluate the methods that we explained in Chapter 3: color histograms, HOG features, SIFT descriptors with BoW model, and the AISP descriptors. We compare the local and global feature components of the AISP features in addition to the final AISP descriptor. We report and illustrate the retrieval accuracies of the above-mentioned methods on the artificial and natural datasets using the CMC curves.

4.4.1. Experiments on the Artificial Dataset

We studied the effectiveness of the methods for the different types of manipulations using the artificial dataset. We ran experiments for combinations of types of reuse and manipulations.

To summarize the results, we calculated an overall performance for each method by taking the average of the first, fifth, tenth, and twentieth rank retrieval accuracies. Figure 4.4 summarizes the results in a bar graph.

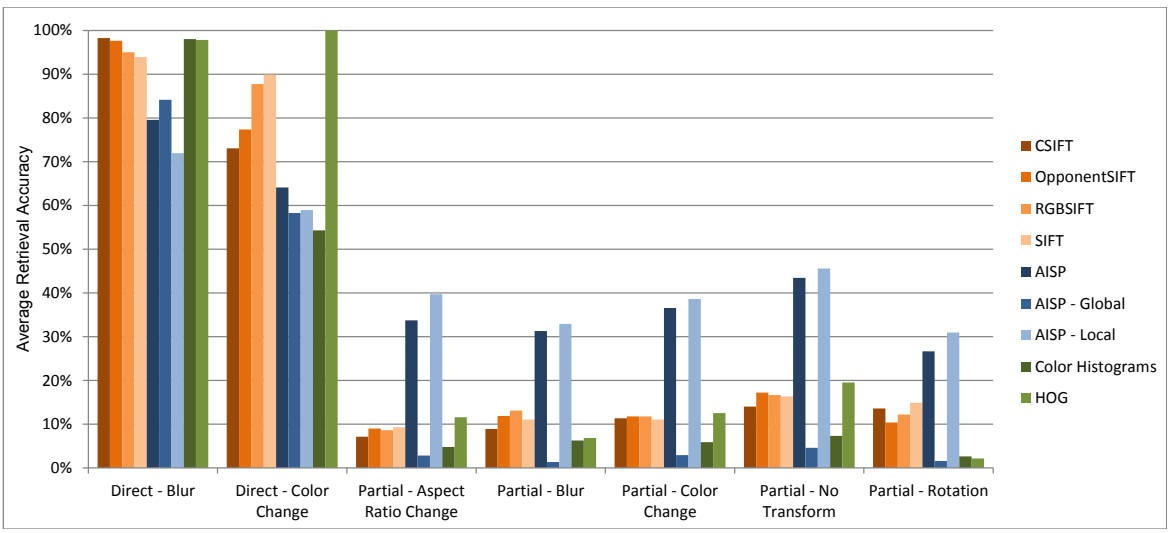
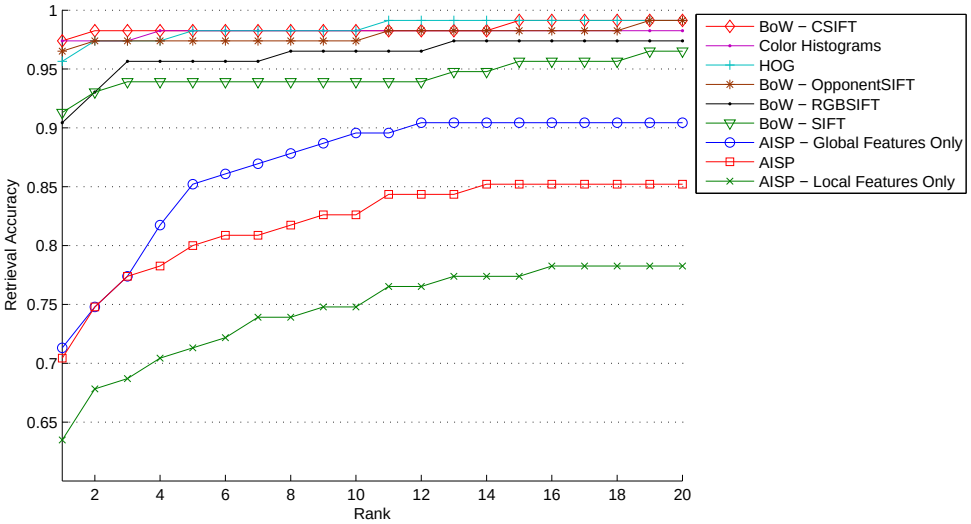


Figure 4.4. Summary of the results on the artificial dataset

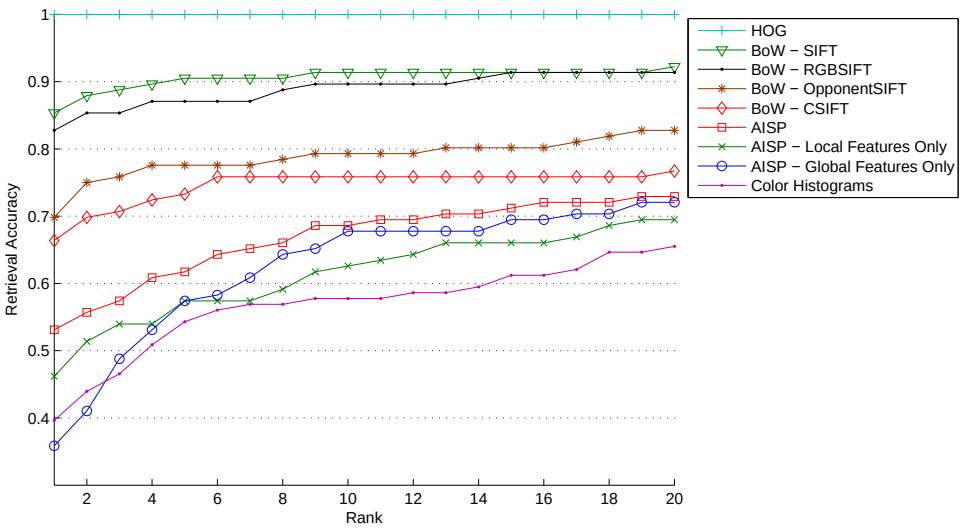
For the direct reuse category, we observe from Figure 4.5 that the AISP descriptors do not perform as good as the other methods. This result can be explained by the dependence of the AISP algorithm on the salient patch detection results. When a parent image is used as a whole or constitute a major part in a child image, different salient sub-regions on the images might be labeled as the salient patches. Unlike the SIFT descriptor, AISP relies on a single salient region. As a consequence, a non-matching pair of salient patches may result in poor matching accuracy.

Under color transforms (see Figure 4.5 b), the HOG descriptor outperformed the other methods since it does not employ color-based features. Even though edges can be deteriorated by excessive color manipulations, in this experiment, the edge orientation histograms are preserved enough for accurate matching.

Figure 4.6, Figure 4.7, and Figure 4.8 compare the retrieval accuracies for the case of partial reuse. The first figure presents the results for the partially reused images with no other systematic transformation, and the second and third figures show the results for the instances with the previously explained transformations: aspect ratio change, blur, color change, and rotation. The figures show that the global features fail in most cases where the source image is reused partially. The main reason of this result is that the foreground objects in the artificial dataset images are context independent, since the foreground and background images are combined randomly. Thus, the global descriptors produced poor results for these cases. However, the retrieval accuracies of the global descriptors still seem to be better than random, since the global descriptors can still successfully match images if the reused area on the image is large enough and the amount of manipulations are admissible. Contrary to other global features, the HOG descriptor displayed relatively better results, which may be due to the contribution of the strong edges that enclose the reused foreground objects on the images, where strong edges arise as a result of the foreground-background contrast in the artificial dataset images. The background-foreground contrast also facilitates the salient patch detection. As shown in Figures 4.6, 4.7, and 4.8, the AISP features showed notably higher performance than the other methods for all the five cases of partial reuse that are studied here. Among the SIFT descriptors, the use of color features seem to decrease retrieval performance when there is a significant amount of color change as in Figure 4.5 (b). In other cases, none of the SIFT descriptors outperformed the others for all cases.



(a) Blur



(b) Color change

Figure 4.5. Cumulative matching accuracies of the methods for direct reuse on the artificial dataset

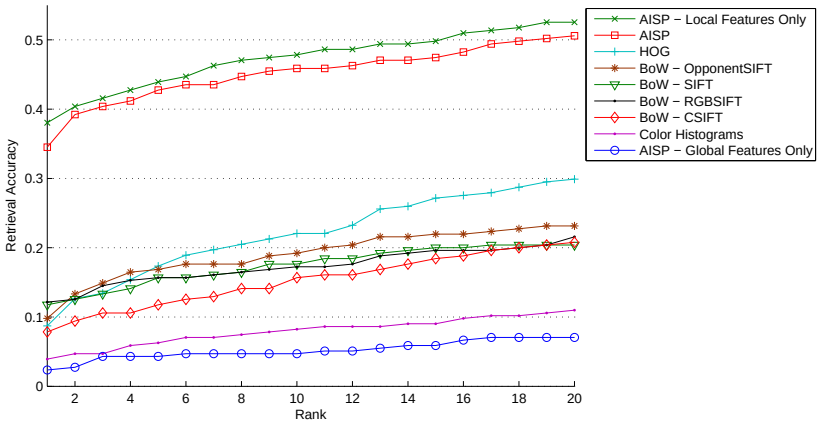
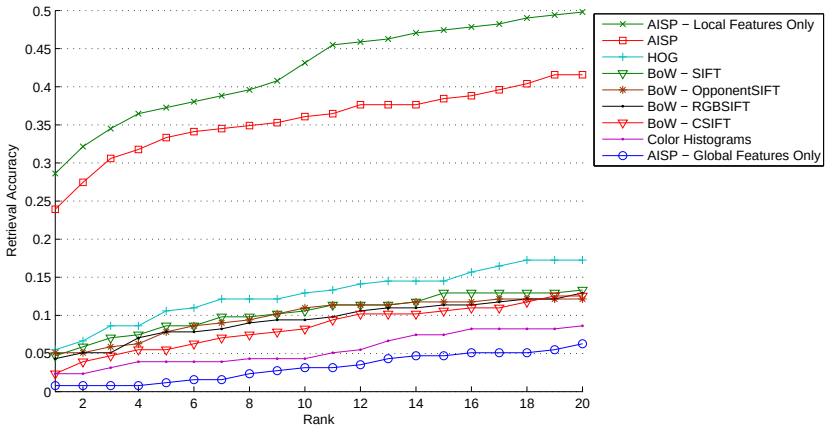
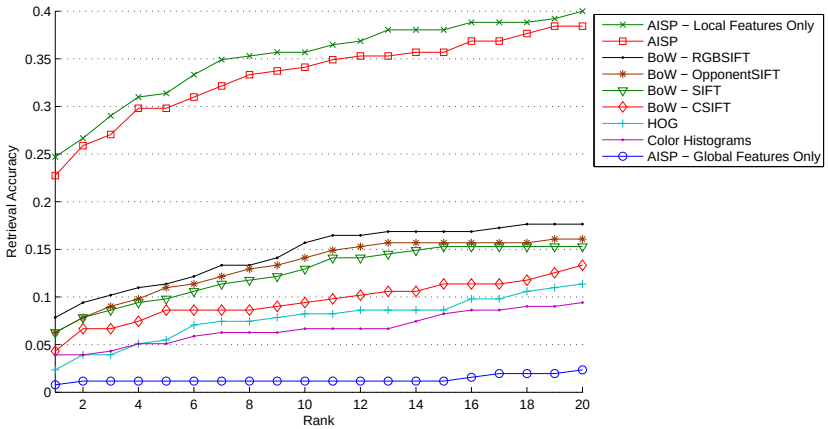


Figure 4.6. Cumulative matching accuracies of the methods for partial reuse on the artificial dataset with no transformation

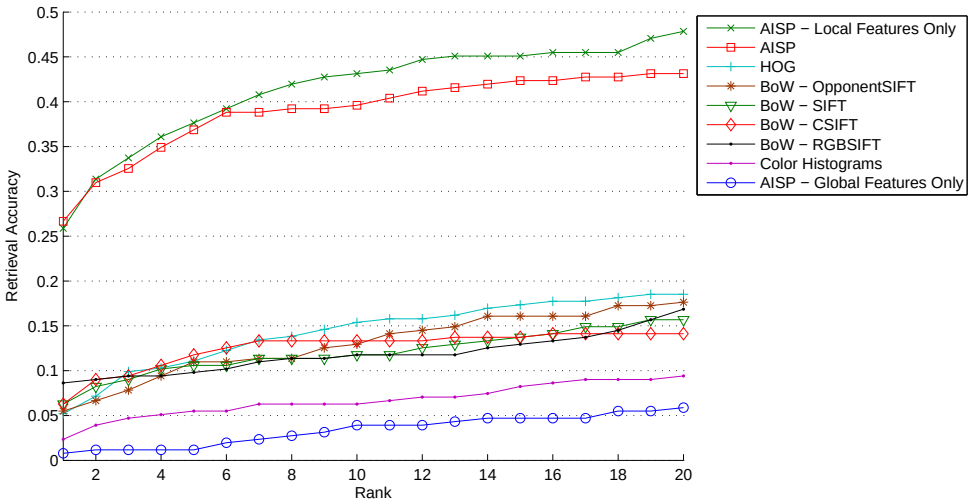


(a) Aspect ratio change

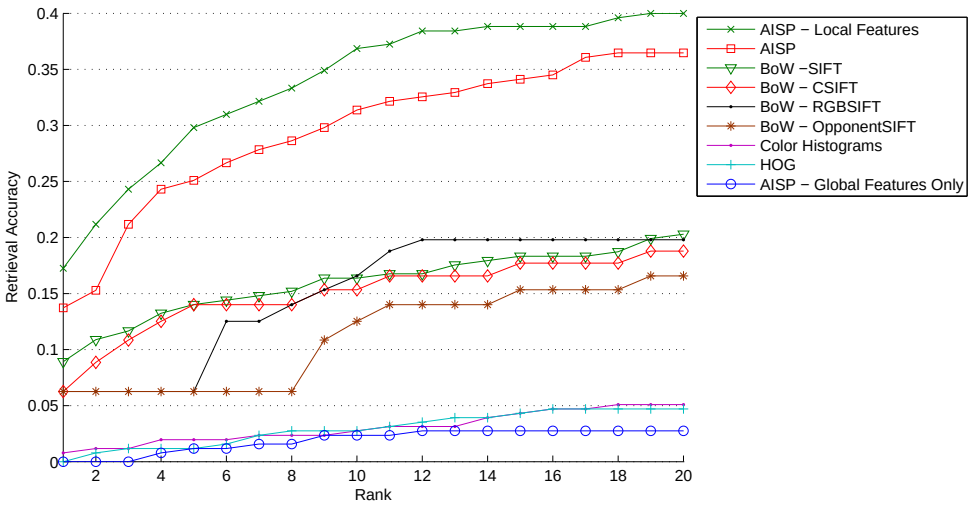


(b) Blur

Figure 4.7. Cumulative matching accuracies of the methods for partial reuse on the artificial dataset with aspect ratio change and blur



(a) Color change



(b) Rotation

Figure 4.8. Cumulative matching accuracies of the methods for partial reuse on the artificial dataset with color change and rotation

4.4.2. Experiments on the Natural Dataset

In order to evaluate the methods in our framework in a more natural setting, we performed experiments on the natural dataset that we described earlier. We compared the methods for each of the four types of reuse and nine types of manipulations. As we stated earlier, our gallery in the natural dataset consists of 144 source images, which makes our problem similar to a 144-class image classification problem.

Figure 4.9 presents the results of our experiments on the direct reuse category on the natural dataset. As it is seen from the figure, for the direct reuse category, the experiments on the natural dataset showed similar results as the artificial dataset experiments, in which the AISP descriptors failed to outperform the other methods. In this category, the standard SIFT descriptor gave the best retrieval results.

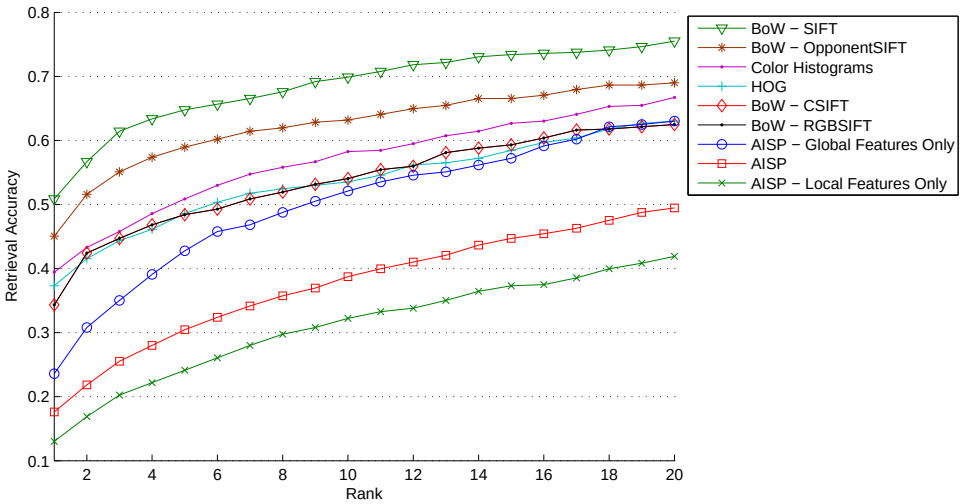


Figure 4.9. CMC plots for the direct reuse category on the natural dataset

For the partial reuse category, the methods showed similar retrieval accuracies in general, where the local descriptors tended to give more accurate results as shown in Figure 4.10. A highlight in the figure is that the combination of local and global features in the AISP descriptor can provide better retrieval results than both local and global features alone. Partial reuse category in the natural dataset can be a good example of

this, since it includes images with objects that are usually reused independently, yet not completely in a context independent fashion in the destination image.

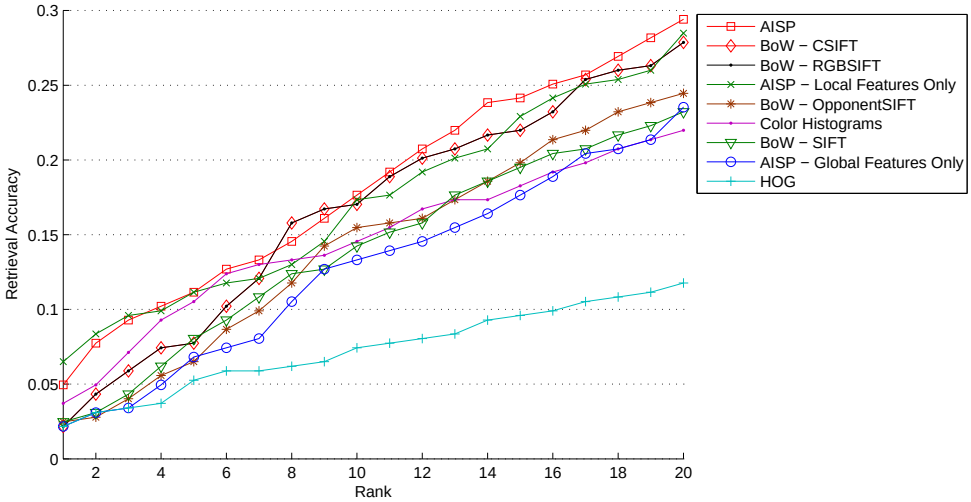


Figure 4.10. CMC plots for the partial reuse category on the natural dataset

The remake category is relatively less restricted than the other types of reuse and consists of a variety of images. Images can be similar to their source images in terms of color, texture, edgeness, or another criterion. Thus, the results for the remake category might not be as informative as other branches of reuse. CMC curves for the remake category are plotted on Figure 4.11.

When source images are used as backgrounds, the AISP algorithm tends to fail, since it is intrinsically foreground sensitive and looks for a foreground object in every instance, even if it does not exist. Our experimental results have also confirmed this weakness as presented in Figure 4.12, in which the local feature-based AISP descriptor ranks as the worst method. The standard SIFT descriptor showed the best performance, similar to the direct reuse category.

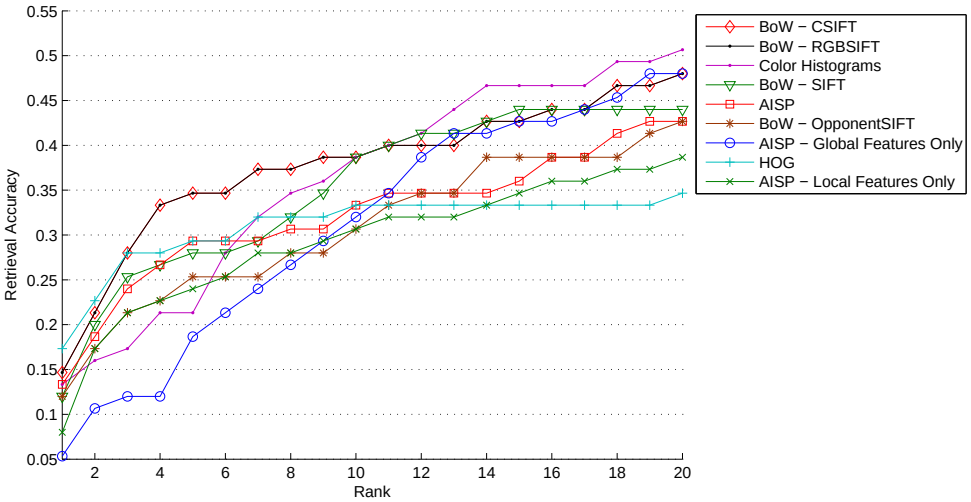


Figure 4.11. CMC plots for the remake category on the natural dataset

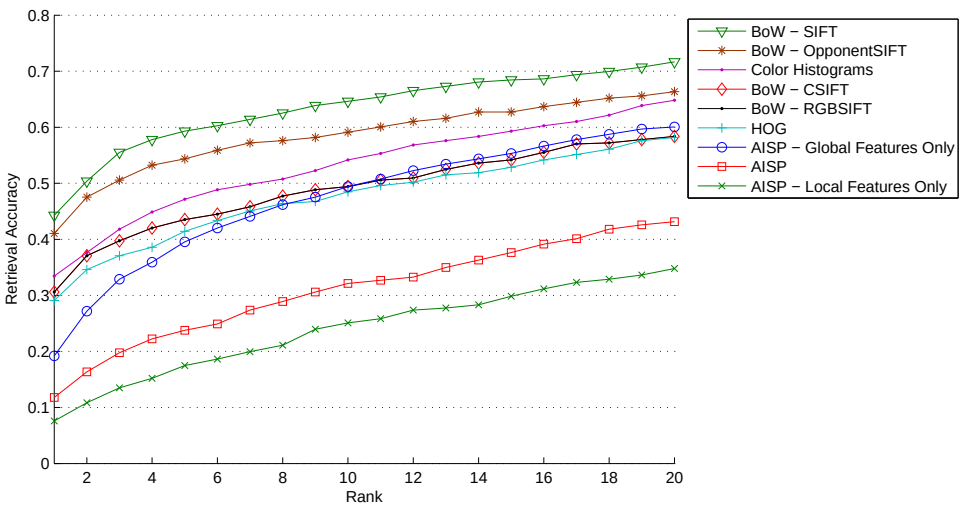


Figure 4.12. CMC plots for the use as a background category on the natural dataset

The experimental results for the four types of reuse are summarized in Figure 4.13.

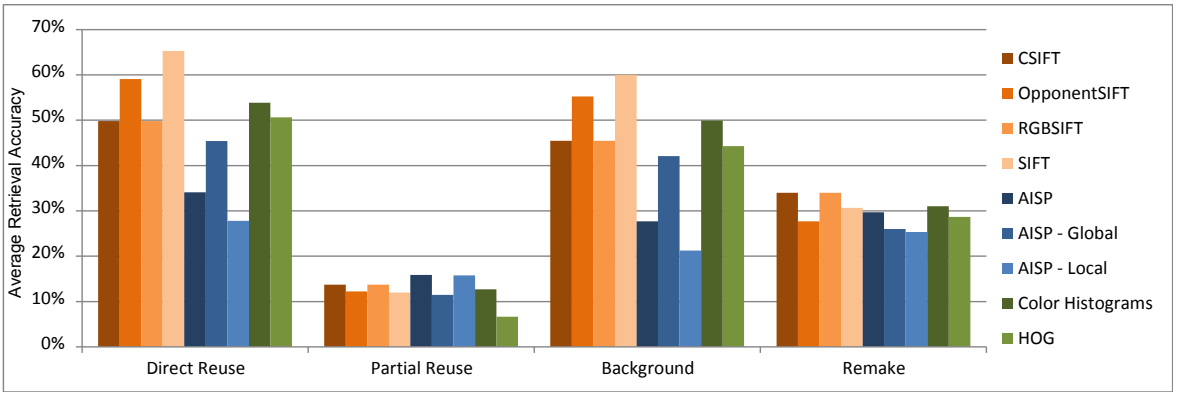


Figure 4.13. Summary of the results on the natural dataset for the four types of reuse

We grouped the types of manipulations according to their frequency of occurrence. Figure 4.14 compares the retrieval results for the most common types of transformations that are observed in our dataset: color manipulation, translation, and texture manipulation. Figure 4.15 shows the results for relatively less common modifications: text overlay, rotation, and aspect ratio change. Finally, Figure 4.16 gives the CMC plots for the least common types of manipulations: alpha blending, mirroring, and duplication. Note that the results may not reflect the actual effects of the corresponding types of manipulations, since each image usually involve more than one type of manipulation, and the number of manipulations tends to increase with decreasing occurrence probability. As a result, characterizability of the corresponding manipulation also decreases with decreasing probability of occurrence. The average numbers of different types of transformations are found as 2.29 for the first group, 2.71 for the second group, and 3.31 for the last group.

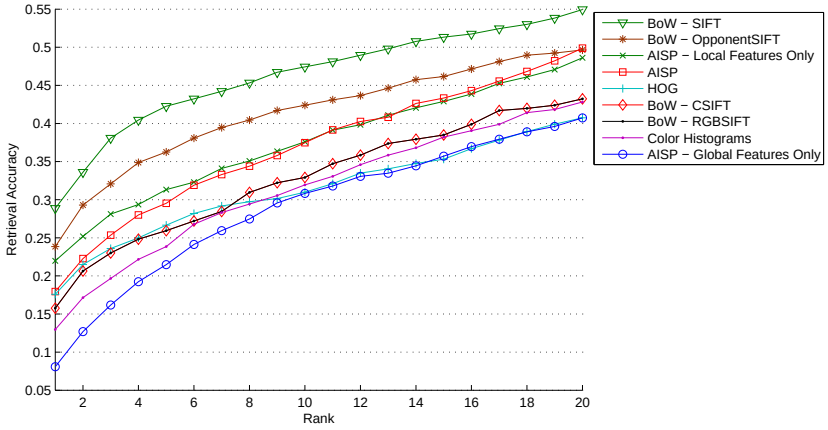
Some highlights from our results for the types of manipulations are listed as follows. Color-based features performed slightly worse than the other descriptors on the images that include color manipulations. In this respect, the standard SIFT descriptor displayed the best performance for the color manipulation category. However, color

based descriptors were still able to retrieve the correct source images, since excessive manipulation of colors is not common among naturally reused images. HOG features showed poor performance on cropped and/or translated images, since they are not robust to translations. In our implementation, they are computed over the whole image, and during matching we do not test every possible frame combination. This would have been too computationally expensive.

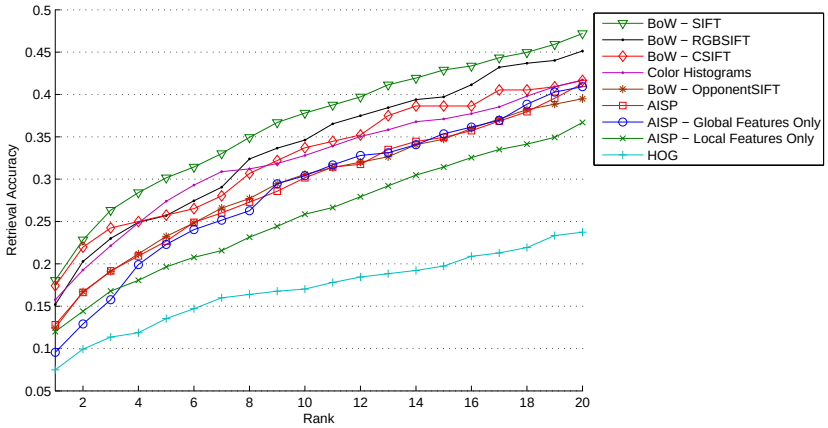
Overlay image captions are usually observed in the direct reuse and background categories. The low performance of AISP descriptors in this category can be explained by the noise in saliency maps caused by text, since the letters usually have a high contrast to their neighbor regions, although they are usually not the reused part in an image.

Rotation and aspect ratio changes are evaluated independently on the artificial dataset, where each image contains only one type of transformation. For these transformations, the results here are not as informative as the artificial dataset results, since rotation and aspect ratio changes are usually not observed solely in the natural dataset.

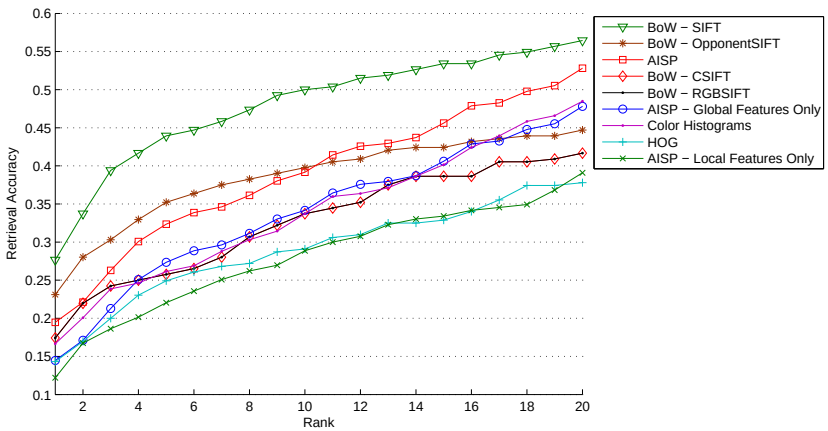
The results for the third group of manipulations (alpha blending, mirroring, and duplication) might not be as significant as the previous groups (color manipulation, translation, texture manipulation, text overlay, rotation, and aspect ratio change), since they highly depend on individual instances and have a large overlap with other types of manipulations. Even though the results in this group can be weak, we observe that alpha blending and rotation are challenging types of manipulation for all the reuse detection methods that we have evaluated.



(a) Color manipulation

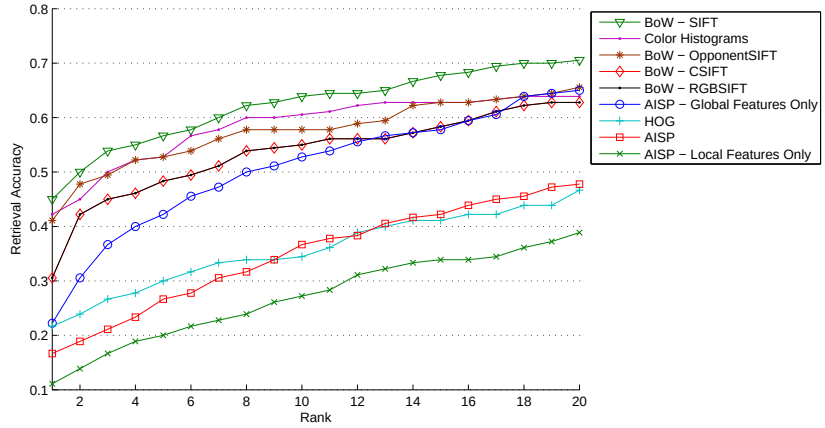


(b) Translation

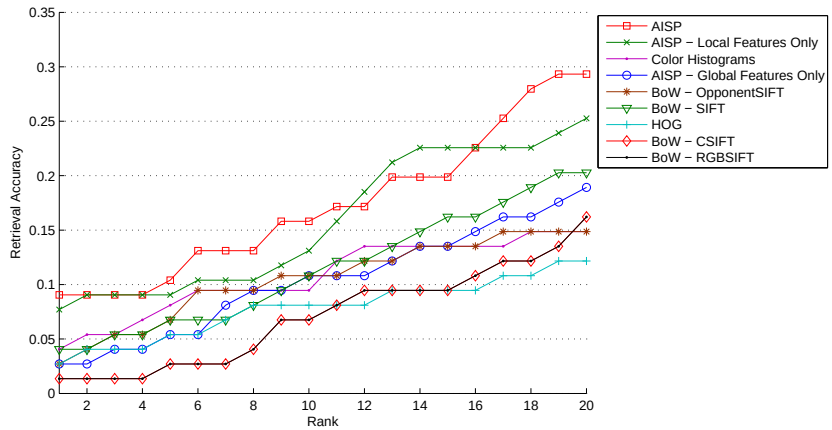


(c) Texture manipulation

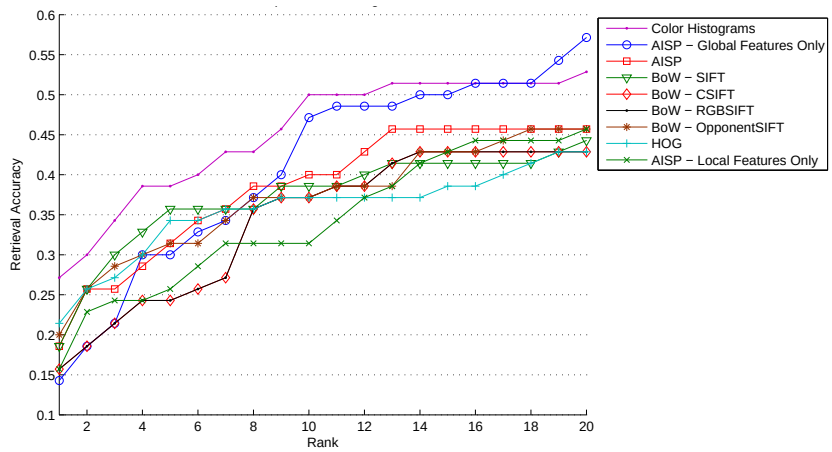
Figure 4.14. CMC plots for the most frequently observed types of manipulations on the natural dataset



(a) Text overlay

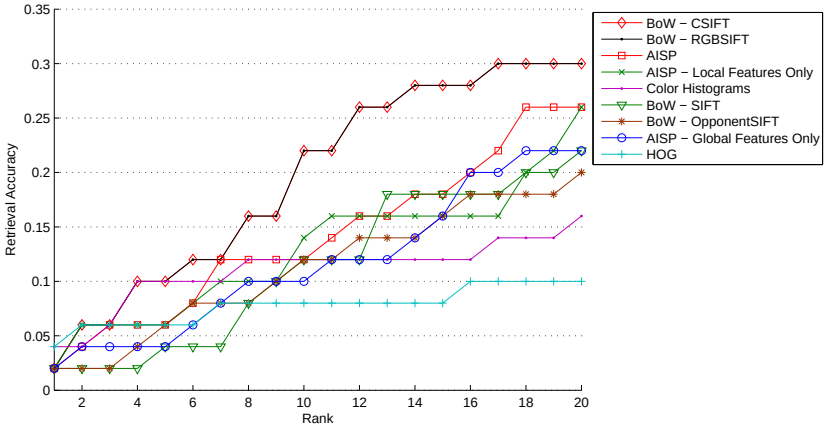


(b) Rotation

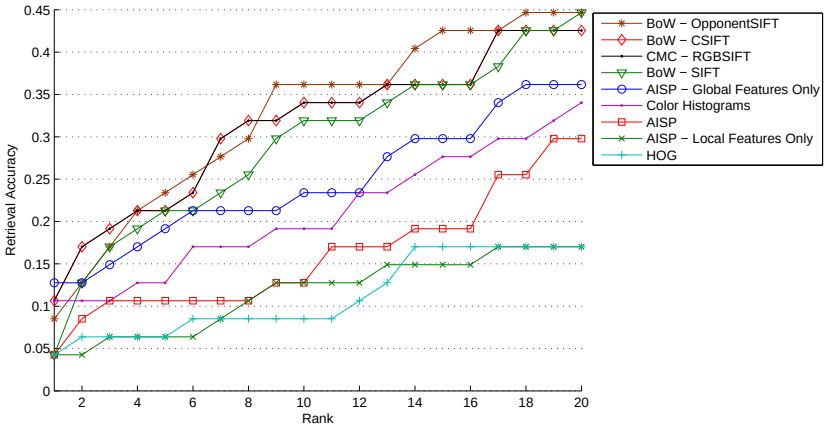


(c) Aspect ratio change

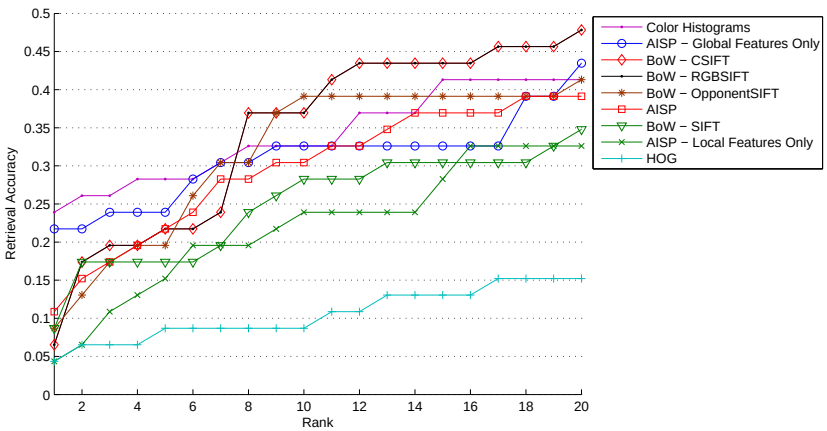
Figure 4.15. CMC plots for less frequently observed types of manipulations on the natural dataset



(a) Alpha blending



(b) Mirroring



(c) Duplication

Figure 4.16. CMC plots for the least frequently observed types of manipulations on the natural dataset

Experimental results for each of the nine types of manipulations –color manipulation, translation, texture manipulation, text overlay, rotation, aspect ratio change, alpha blending, mirroring, and duplication– are summarized in Figure 4.17.

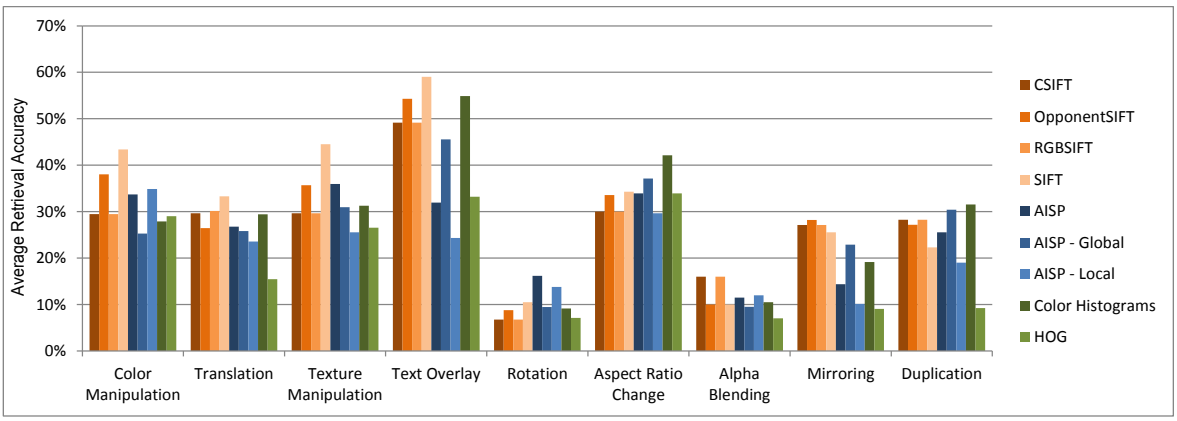


Figure 4.17. Summary of the results on the natural dataset for the nine types of manipulations

4.5. Comparison in terms of dimensionality and computation time

In this section we compare the methods in terms of time and memory. We performed the experiments on a computer with Intel Core i3 2.26GHz processor and 6GB memory. We ran the methods for all 1200 images in the natural dataset and computed an *average time consumption per image* value for each method. We also report the dimensionalities of the methods for comparison. For the SIFT descriptors, the reported dimensionalities are averages for the natural dataset. For the BoW, the reported codebook generation time is the offline time elapsed during the clustering of the SIFT features that are extracted from 144 original images in the natural dataset. Table 4.3 reports the computation time and dimensionality for each method.

Even though the AISP descriptor runs faster than the SIFT descriptors, its relative computational cost is high as we consider the dimensionality of the descriptor.

This is mainly a result of the computationally expensive salient patch detection stage, which constitutes the bottleneck in the AISP algorithm.

Table 4.3. Computation time and dimensionalities of the methods.

Method	**Dimensionality**	**Computation time**
AISP	160	1.526 seconds/image
Color histograms	120	0.199 second/image
HOG	384	0.709 second/image
SIFT	110592	1.628 seconds/image
OpponentSIFT	331776	1.957 seconds/image
CSIFT	331776	1.938 seconds/image
RGBSIFT	331776	1.916 seconds/image
BoW - 160	160	13.19 minutes (codebook computation)
BoW - 320	320	21.37 minutes (codebook computation)
BoW - 640	640	55.34 minutes (codebook computation)
BoW - 1280	1280	115.1 minutes (codebook computation)
BoW - 2560	2560	180.58 minutes (codebook computation)

5. CONCLUSION

In this work, we studied the image reuse detection problem in digital artworks. We first overviewed existing approaches in the related areas of research: image copy and manipulation detection, content-based image retrieval, saliency estimation, and computer analysis of artwork. Then, we examined the methods that could be useful for the reuse detection problem: color histograms, HOG, SIFT, RGB-SIFT, OpponentSIFT, and C-SIFT descriptors. In addition to the existing methods, we also proposed a novel image description method, the Affine Invariant Salient Patch (AISP) descriptor, which makes use of image saliency to estimate a foreground-region in an image and emphasizes its contribution to the final description of the image.

For the evaluation of our methodology, we proposed two novel datasets, one artificial and one natural. We proposed a taxonomy that identifies the types of reuse and modifications in digital artworks. For each branch of reuse and type of modification that are addressed in our proposed taxonomy, we ran the methods in our framework and compared the retrieval results. Based on our experimental results we showed that our method, AISP, can be a lightweight alternative to SIFT-BoW approach, especially for the case of partial reuse, where a selected region in a source image is reused in a different context in another image. However, with its high overall performance, the use of SIFT descriptors with a BoW model is recommended when image manipulations are not limited to one type of transformation and there is no prior information about the characteristics of the manipulations.

The major strength of the AISP method is its compactness and scalable computation. The AISP descriptor produces low-dimensional vectors to characterize images, and it does not require a super-linear preprocessing step such as vector clustering. A major weaknesses of AISP is that it highly depends on the salient patch detection result and relies on a single foreground region instead of multiple candidate regions. As a result, a failure to detect the actual region of interest is likely to result in a mismatch, even though global features are also integrated into the final descriptor to alleviate

this dependence. As the results also showed, the AISP descriptors fell behind other methods for the direct reuse and background categories, in which the reused region is usually the background itself.

In future work we hope to empower our descriptor with a more robust and computationally efficient foreground segmentation method. In order to determine the regions that are most likely to be reused, a machine learning system can be trained using an annotated natural dataset that includes manually segmented regions. Another improvement can be detecting multiple candidate salient patches and using a limited number of the patches to retrieve multiple images of origin. In addition, a weighting scheme can be developed to adjust the contribution of the candidate patches and the detected background. Parametrizing the importance of color and edge based features can also help optimize the retrieval performance for different types of transformations. To improve retrieval accuracy, different methods can be selected or fused, using a set of functions that help determine which methods to use based on image content (i.e. image entropy). In this work, we drew our conclusions from datasets with limited annotation, since creating a large annotated dataset requires a lot of time and manual effort. It is possible to extend our research with a larger natural dataset with more detailed annotation.

REFERENCES

1. Smeulders, A., M. Worring, S. Santini, A. Gupta and R. Jain, "Content-based Image Retrieval at the End of the Early Years", *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, Vol. 22, No. 12, pp. 1349 –1380, 2000.

2. Liu, Y., D. Zhang, G. Lu and W.-Y. Ma, "A Survey of Content-based Image Retrieval with High-level Semantics", *Pattern Recognition*, Vol. 40, No. 1, pp. 262 – 282, 2007.

3. Buter, B., N. Dijkshoorn, D. Modolo, Q. Nguyen, S. van Noort, B. van de Poel, A. Ali and A. Salah, "Explorative Visualization and Analysis of a Social Network for Arts: the Case of deviantART", *Journal of Convergence*, Vol. 2, No. 1, 2011.

4. Bayram, S., I. Avcibas, B. Sankur and N. Memon, "Image Manipulation Detection", *Journal of Electronic Imaging*, Vol. 15, No. 4, pp. 041102–041102, 2006.

5. Ke, Y., R. Sukthankar, L. Huston, Y. Ke and R. Sukthankar, "Efficient Near-duplicate Detection and Sub-image Retrieval", *In ACM Multimedia*, pp. 869–876, 2004.

6. Fridrich, A. J., B. D. Soukal and A. J. Lukáš, "Detection of Copy-move Forgery in Digital Images", *in Proceedings of Digital Forensic Research Workshop*, Citeseer, 2003.

7. Kim, C., "Content-based Image Copy Detection", *Signal Processing: Image Communication*, Vol. 18, No. 3, pp. 169–184, 2003.

8. Zhao, W.-L. and C.-W. Ngo, "Scale-rotation Invariant Pattern Entropy for Keypoint-based Near-duplicate Detection", *Image Processing, IEEE Transactions on*, Vol. 18, No. 2, pp. 412–423, 2009.

9. Cox, I., M. Miller, J. Bloom and M. Miller, *Digital Watermarking*, Morgan Kaufmann, Berlin, 2001.

10. Wu, M.-N., C.-C. Lin and C.-C. Chang, "Novel Image Copy Detection with Rotating Tolerance", *Journal of Systems and Software*, Vol. 80, No. 7, pp. 1057 – 1069, 2007.

11. Xu, Z., H. Ling, F. Zou, Z. Lu and P. Li, "Robust Image Copy Detection Using Multi-resolution Histogram", *Proceedings of the International Conference on Multimedia Information Retrieval*, MIR '10, pp. 129–136, ACM, New York, NY, USA, 2010.

12. Lowe, D., "Distinctive Image Features from Scale-invariant Keypoints", *International Journal of Computer Vision*, Vol. 60, No. 2, pp. 91–110, 2004.

13. Bay, H., A. Ess, T. Tuytelaars and L. Van Gool, "Speeded-up Robust Features (SURF)", *Computer Vision and Image Understanding*, Vol. 110, No. 3, pp. 346–359, 2008.

14. Ke, Y. and R. Sukthankar, "PCA-SIFT: A More Distinctive Representation for Local Image Descriptors", *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, Vol. 2, pp. II–506, IEEE, 2004.

15. Zhou, W., Y. Lu, H. Li, Y. Song and Q. Tian, "Spatial Coding for Large Scale Partial-duplicate Web Image Search", *Proceedings of the International Conference on Multimedia*, MM '10, pp. 511–520, ACM, New York, NY, USA, 2010.

16. Niblack, C. W., R. Barber, W. Equitz, M. D. Flickner, E. H. Glasman, D. Petkovic, P. Yanker, C. Faloutsos and G. Taubin, "QBIC project: Querying Images by Content, Using Color, Texture, and Shape", *IS&T/SPIE's Symposium on Electronic Imaging: Science and Technology*, pp. 173–187, International Society for Optics and Photonics, 1993.

17. Pentland, A., R. W. Picard and S. Sclaroff, "Photobook: Content-based Manipulation of Image Databases", *International Journal of Computer Vision*, Vol. 18, No. 3, pp. 233–254, 1996.

18. Carson, C., M. Thomas, S. Belongie, J. M. Hellerstein and J. Malik, "Blobworld: A System for Region-based Image Indexing and Retrieval", *Visual Information and Information Systems*, pp. 509–517, Springer, 1999.

19. Shapovalova, N., C. Fernández, F. X. Roca and J. Gonzàlez, "Semantics of Human Behavior in Image Sequences", *Computer Analysis of Human Behavior*, pp. 151–182, Springer, 2011.

20. Hörster, E., R. Lienhart and M. Slaney, "Image retrieval on Large-scale Image Databases", *Proceedings of the 6th ACM International Conference on Image and Video Retrieval*, CIVR '07, pp. 17–24, ACM, New York, NY, USA, 2007.

21. Philbin, J., O. Chum, M. Isard, J. Sivic and A. Zisserman, "Object Retrieval with Large Vocabularies and Fast Spatial Matching", *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pp. 1 –8, June 2007.

22. Turcot, P. and D. Lowe, "Better Matching with Fewer Features: The Selection of Useful Features in Large Database Recognition Problems", *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pp. 2109 –2116, 2009.

23. Blei, D. M., A. Y. Ng and M. I. Jordan, "Latent Dirichlet Allocation", *the Journal of Machine Learning Research*, Vol. 3, pp. 993–1022, 2003.

24. "Flickr, A photo sharing website", `http://www.flickr.com/`, 2004, accessed January 2013.

25. Itti, L., "Visual Salience", *Scholarpedia*, Vol. 2, No. 9, p. 3327, 2007.

26. Frintrop, S., "Computational Visual Attention", *Computer Analysis of Human Behavior*, pp. 69–101, Springer, 2011.

27. Itti, L., C. Koch and E. Niebur, "A Model of Saliency-based Visual Attention for Rapid Scene Analysis", *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, Vol. 20, No. 11, pp. 1254 –1259, Nov 1998.

28. Koch, C. and S. Ullman, "Shifts in Selective Visual Attention: Towards the Underlying Neural Circuitry", *Human Neurobiology*, Vol. 4, pp. 219 –227, 1985.

29. Hou, X. and L. Zhang, "Saliency detection: A Spectral Residual Approach", *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pp. 1–8, IEEE, 2007.

30. Bruce, N. D. and J. K. Tsotsos, "Saliency, Attention, and Visual search: An Information Theoretic Approach", *Journal of Vision*, Vol. 9, No. 3, 2009.

31. Judd, T., K. Ehinger, F. Durand and A. Torralba, "Learning to Predict Where Humans Look", *Computer Vision, 2009 IEEE 12th International Conference on*, pp. 2106–2113, IEEE, 2009.

32. Achanta, R., F. Estrada, P. Wils and S. Süsstrunk, "Salient Region Detection and Segmentation", *Computer Vision Systems*, pp. 66–75, Springer, 2008.

33. Fu, Y., J. Cheng, Z. Li and H. Lu, "Saliency Cuts: An Automatic Approach to Object Segmentation", *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pp. 1–4, IEEE, 2008.

34. Cheng, M.-M., G.-X. Zhang, N. Mitra, X. Huang and S.-M. Hu, "Global Contrast Based Salient Region Detection", *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pp. 409 –416, June 2011.

35. Felzenszwalb, P. and D. Huttenlocher, "Efficient Graph-based Image Segmenta-

tion", *International Journal of Computer Vision*, Vol. 59, pp. 167–181, 2004.

36. Johnson, C. R., E. Hendriks, I. J. Berezhnoy, E. Brevdo, S. M. Hughes, I. Daubechies, J. Li, E. Postma and J. Z. Wang, "Image Processing for Artist Identification", *Signal Processing Magazine, IEEE*, Vol. 25, No. 4, pp. 37–48, 2008.

37. Carneiro, G., N. P. da Silva, A. Del Bue and J. P. Costeira, "Artistic Image Classification: An Analysis on the PRINTART Database", *Computer Vision–ECCV 2012*, pp. 143–157, Springer, 2012.

38. Graham, D. J., J. D. Friedenberg, D. N. Rockmore and D. J. Field, "Mapping the Similarity Space of Paintings: Image Statistics and Visual Perception", *Visual Cognition*, Vol. 18, No. 4, pp. 559–573, 2010.

39. Shamir, L. and J. A. Tarakhovsky, "Computer Analysis of Art", *Journal on Computing and Cultural Heritage (JOCCH)*, Vol. 5, No. 2, p. 7, 2012.

40. Berezhnoy, I. E., E. O. Postma and J. van den Herik, "Computerized Visual Analysis of Paintings", *Proceedings 16th International Conference of the Association for History and Computing*, pp. 28–32, 2005.

41. Li, C. and T. Chen, "Aesthetic Visual Quality Assessment of Paintings", *Selected Topics in Signal Processing, IEEE Journal of*, Vol. 3, No. 2, pp. 236–252, 2009.

42. Carneiro, G., "Graph-based Methods for the Automatic Annotation and Retrieval of Art Prints", *Proceedings of the 1st ACM International Conference on Multimedia Retrieval*, p. 32, ACM, 2011.

43. Dalal, N. and B. Triggs, "Histograms of Oriented Gradients for Human Detection", *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, Vol. 1, pp. 886–893, IEEE, 2005.

44. Van De Sande, K., T. Gevers and C. Snoek, "Evaluating Color Descriptors for

Object and Scene Recognition", *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, Vol. 32, No. 9, pp. 1582–1596, 2010.

45. Bosch, A., X. Muñoz and R. Martí, "Which is the Best Way to Organize/Classify Images by Content?", *Image and Vision Computing*, Vol. 25, No. 6, pp. 778 – 791, 2007.

46. van de Sande, K. E., T. Gevers and C. G. Snoek, "Empowering Visual Categorization with the GPU", *Multimedia, IEEE Transactions on*, Vol. 13, No. 1, pp. 60–70, 2011.

47. Rother, C., V. Kolmogorov and A. Blake, "GrabCut: Interactive Foreground Extraction Using Iterated Graph Cuts", *ACM Trans. Graph.*, Vol. 23, No. 3, pp. 309–314, Aug. 2004.

48. Isikdogan, F. and A. Salah, "Affine Invariant Salient Patch Descriptors for Image Retrieval", *International Workshop on Image and Audio Analysis for Multimedia Interactive Services*, Paris, France, 2013.

49. Teague, M., "Image Analysis via the General Theory of Moments", *Journal of the Optical Society of America*, Vol. 70, pp. 920–930, Aug 1980.

50. Everingham, M., L. Van Gool, C. K. I. Williams, J. Winn and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results", http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

51. "deviantART, A social artwork network", `http://www.deviantart.com/`, 2000, accessed January 2013.

52. Nowak, E., F. Jurie and B. Triggs, "Sampling Strategies for Bag-of-features Image Classification", *Computer Vision–ECCV 2006*, pp. 490–503, Springer, 2006.