UNIVERSITAT POLITÈCNICA DE CATALUNYA

Universitat Politècnica de Catalunya

Teoria del Senyal i Comunicacions

Visual Object Analysis Using Regions and Interest Points

Ph.D. Thesis Proposal

by Carles Ventura Royo

Advisors:
Prof. Xavier Giró i Nieto
Prof. Verónica Vilaplana Besler

Barcelona, April 2013

Abstract

This dissertion research explores two of the most-used image models for object detection, 3D reconstruction, visual search: region-based and interest-points image representations; and will try to provide a new image model to take advantage of the strengths and overcome the weaknesses of both approaches. More specifically, we will focus on the gPb-owt-ucm segmentation algorithm and the SIFT local features since they are the most contrasted techniques in their respective fields. Furthermore, using an object retrieval benchmark, this dissertation research will analyze three basic questions: (*i*) the usefulness of an interest points hierarchy based on a contour strength signal, (*ii*) the influence of the context on both interest points location and description, and (*iii*) the analysis of regions as spatial support for bundling interest points.

Contents

Chapter 1

Motivation

Object detection, image matching, query by example, etc. are some of the applications that in the recent years have kept image processing and computer vision fields active. The most common input of these applications is a digital image, which can be defined as a 2-d matrix of pixel colors. In images, objects are represented at low level as a set of pixels, most of which have some perceptual characteristics that follows some model. However, trying to indentify these models is a difficult task because instances of a same object can vary drastically due to different factors such as scale, illumination and viewpoint changes, rotation, occlusion, etc. that makes object detection and recognition more challenging. With this goal, high-level algorithms are usually based on richer image representations that take into account either homogeneous or saliency characteristics. This dissertation research will explore two of the most used: region-based and interest-points image representations.

Region-based image representation considers an image as a set of groups of coherent pixels, called regions, obtained by a segmentation algorithm. This way, an object is described as a set of regions and their internal properties, or descriptors, can be used to perform a query by example or to detect those areas where there is possibly a face.

Interest-points image representation is one of the most popular approaches for state-of-the-art applications, such as image classification, object detection, object recognition, and object retrieval. It represents the image as a set of interest points or local features. An object, for instance, can be described as a structured set of local features.

The main objective of this dissertation research will be to analyze the relationship between region-based and interest-points image representations and try to propose a new hybrid representation that takes advantage of the strengths and overcomes the weaknesses of both approaches. This Thesis Proposal is structured in three main parts: (*i*) Section 2 analyzes how hierarchical region-based segmentation and interest points representation can benefit one from the other, (*ii*) Section 3 focuses on exploring the influence of the context on interest points location and description, and (*iii*) Section 4 analyzes the use of regions as spatial support for bundling interest points. Each of these three parts is divided into the following subsections: (*i*) Related work, (*ii*) Proposed approach, and (*iii*) Experiments. Finally, Section 5 shows the results of some preliminary experiments.

Chapter 2

Hierarchical segmentations and interest points

2.1 Related work

Several image representations have been used for many computer vision problems such as object recognition/detection, object recognition, image classification, etc. First, in Section 2.1.1 we describe one of the state-of-the-art segmentation algorithm: the gPb-owt-ucm. Then, we introduce different interest point detectors and descriptors in Section 2.1.2. Finally, in Section 2.1.3, other image representations such as Bag-of-Features, Spatial Pyramid, Histograms of Oriented Gradients (HOG) and Pyramid Histograms of Orientation Gradients (PHOG).

2.1.1 The gPb-owt-ucm segmentation algorithm

Up to now, the gPb-owt-ucm [AMFM11] is among the state-of-art segmentation algorithms. Segmentation techniques are evaluated on the Berkeley Segmentation Dataset [MFTM01] benchmark and gPb-owt-ucm gives the best performance. It consists of 3 different blocks: (*i*) the gPb contour detector, (*ii*) the Oriented Watershed Transform (OWT), and (*iii*) the Ultrametric Contour Map (UCM).

The gPb contour detector

The gPb contour detector aims at obtaining a set of images where the pixels of each image represent the boundary strength at a given orientation. With this goal, it couples multiscale local brightness, color and texture cues to a powerful globalization framework using spectral clustering.

The local cues are computed by applying oriented gradient operators at every location in the image. Thus, a circular disk split into two half-disks by a diameter at angle θ is placed at location (x, y). The gradient magnitude at (x, y) is defined by the χ^2 distance between the two half-disk intensity histograms. Figure 2.1 shows the histograms obtained from splitting a disk in two halves at orientation $\pi/4$ over the brightness channel. The result of comparing the two histograms for each image pixel at this orientation is also

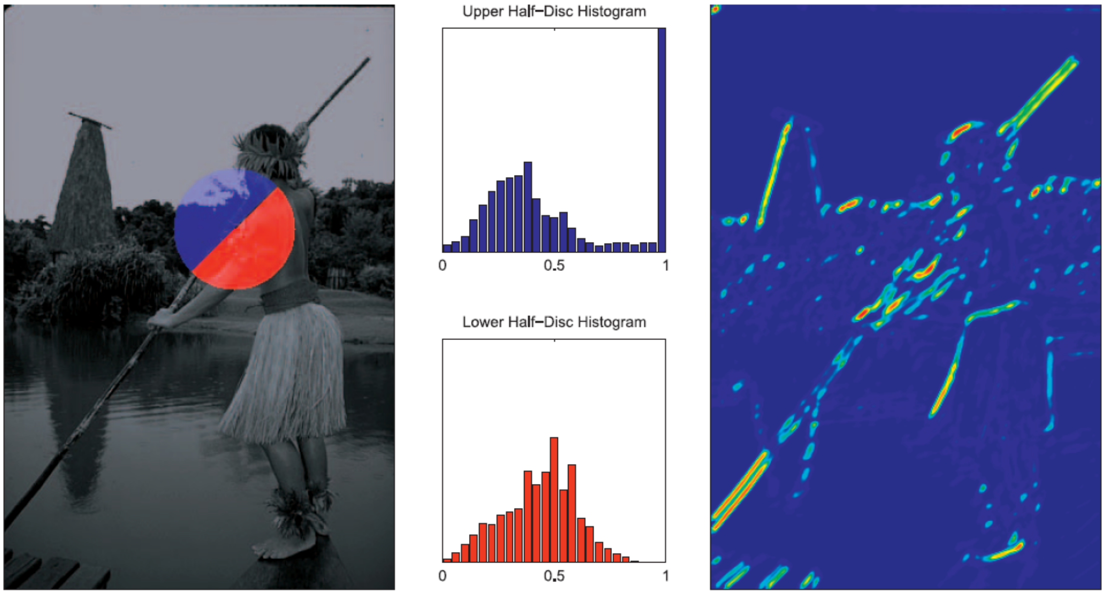


Figure 2.1: Oriented gradient of histograms. Figure taken from [AMFM11]. The left image show a circular disk placed at a pixel from the brightness channel of an image and split by a diameter at angle $\theta = \pi/4$. The central image shows the intensity histograms resulting from the upper and lower half-disks. The right image represents the gradient magnitude at the given orientation for each pixel from the brightness channel.

shown.

The gradient magnitudes are computed over four feature channels (brightness, chrominance and texture) and at different scales, all of which are linearly combined into a multiscale oriented signal:

$$mPb(x, y, \theta) = \sum_{s} \sum_{i} \alpha_{i,s} G_{i,\sigma(i,s)}(x, y, \theta) \tag{2.1}$$

where s indexes scales, i indexes feature channels, and $G_{i,\sigma(i,s)}(x, y, \theta)$ measures the histogram difference in channel i between two halves of a disc of radius $\sigma(i, s)$ centered at (x, y) and divided by a diameter at angle θ. The parameters $\alpha_{i,s}$ weight the relative contribution of each gradient signal. Figure 2.2 shows a general scheme of $G_{i,\sigma(i,s)}(x, y, \theta)$ computation. Figure 2.3 shows the resulting contour signal for each channel at two specific orientations and the maximum response over eight orientations. The $mPb(x, y)$ contour signal resulting from averaging the contour signals across all four channels and across three scales is also shown.

The spectral Pb (sPb) detector is derived from the eigenvectors of a spectral clustering. Since eigenvectors $\mathbf{v_k}$ carry contour information (see Figure 2.4), they are treated as images and convolved with Gaussian directional derivative filters at multiple orientations θ to obtain oriented signals $\{\nabla_\theta \mathbf{v_k}(x, y)\}$. The contour information from n different eigenvectors is combined to provide the spectral component of the gPb contour detector:

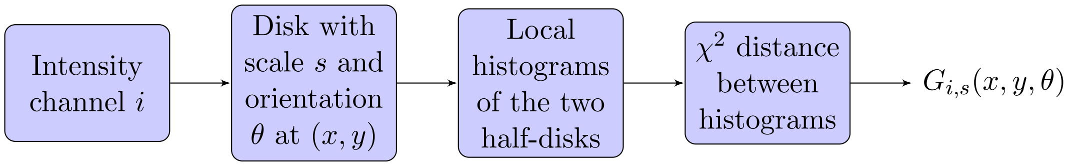

Intensity channel i
Disk with scale s and orientation θ at (x, y)
Local histograms of the two half-disks
χ^2 distance between histograms
$G_{i,s}(x, y, \theta)$

Figure 2.2: General scheme for local cues. For each intensity channel i (brightness, chrominance and texture), gradient intensity is computed at each pixel (x, y) and orientation θ by placing on it a disk with scale s split at orientation θ and computing the χ^2 distance between the local histogram from the two disk halves.

$$sPb(x,y,\theta) = \sum_{k=1}^{n} \frac{1}{\sqrt{\lambda_k}} \nabla_\theta \mathbf{v_k}(x,y) \tag{2.2}$$

where λ_k are their corresponding eigenvalues. Figure 2.5 shows a general scheme of how the spectral component is computed.

The gPb contour detector results from a linear combination of the multiscale mPb and the spectral sPb oriented signals. Whereas mPb fires at all the edges, the sPb extracts only the most salient curves in the image.

The Oriented Watershed Transform

The Oriented Watershed Transform (OWT) constructs the set of initial regions from the oriented contour signal $gPb(x, y, \theta)$. The Watershed Transform consists in placing a water source in each regional minimum, to flood the relief from sources, and build barriers when different sources are meeting.

First, region minimal of the non-oriented contour signal $gPb(x, y)$ (computed as $\max_\theta gPb(x, y, \theta)$) are taken as seed locations for homogeneous segments and standard watershed transform is applied (see Middle Left image from Figure 2.6). As a result, the catchment basins of the minima provide the regions of the finest partition and the corresponding watershed arcs the possible locations of the boundaries. Unfortunately, simply weighting each arc by the mean value of $gPb(x, y)$ for the pixels on the arc can produced artifacts as shown in Figure 2.6 (Middle image). For instance, horizontal watershed arcs near to strong vertical contours are erronously upweighted due to a high magnitude of $gPb(x, y)$ over those pixels caused by the strong vertical gradients. To correct this problem, Oriented Watershed Transform enforces consistency between the strength of the boundaries (watershed arcs) and the underlying oriented contour signal. This is done by estimating the orientation $o(x, y)$ at each pixel on an arc from the local geometry of the arc itself and assigning each arc pixel a boundary strength of $gPb(x, y, o(x, y))$ instead of $gPb(x, y)$. Therefore, the boundary strength assigned to an arc by the OWT can be interpreted as an estimate of the probability of that arc being a true contour. Figure 2.6 shows how artifacts are suppressed by applying the OWT.

The Ultrametric Contour Map

Finally, the Ultrametric Contour Map is the hierarchical region tree which results from an agglomerative clustering by iteratively merging the most similar regions, i.e. the

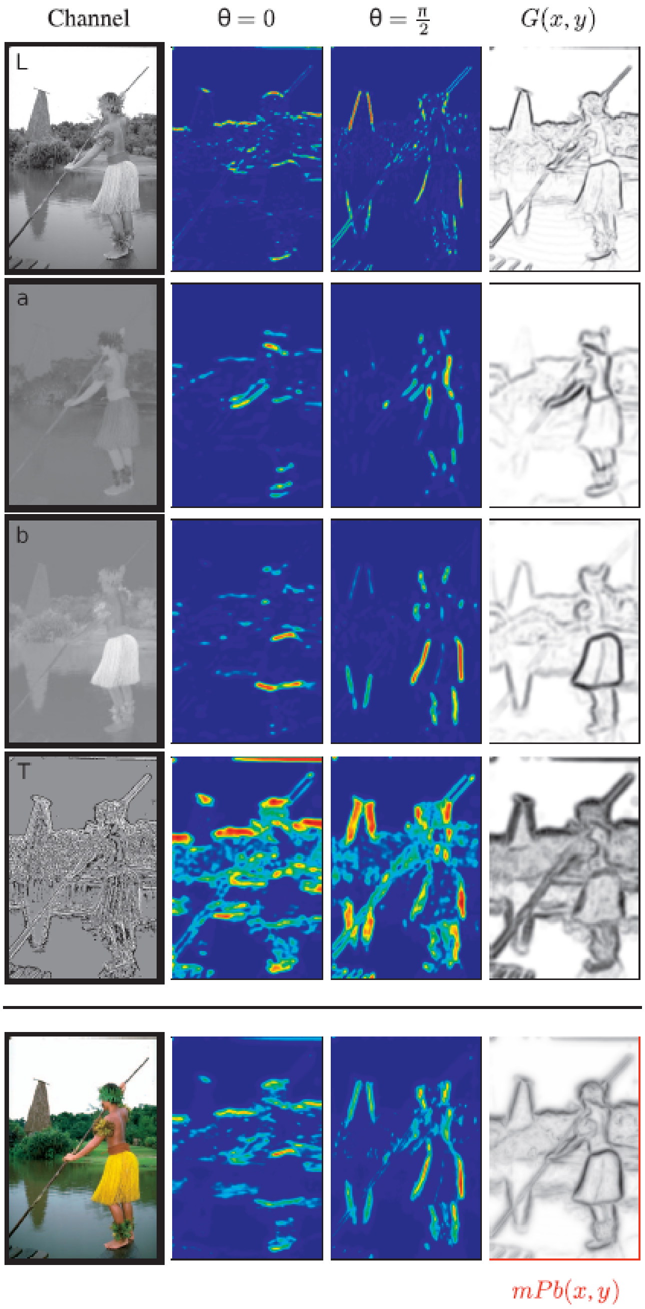


Figure 2.3: Multiscale Pb. Figure taken from [AMFM11]. Images from top to down show the four different channels and the original image. Images from left to right display oriented gradient of histograms (as outlined in Figure 2.1) for $\theta = 0$ and $\theta = \pi/2$, and the maximum response over eight orientations ($G(x, y)$ signal). At the bottom, beside the original image, the combination of oriented gradients across all four channels and across three scales is displayed.

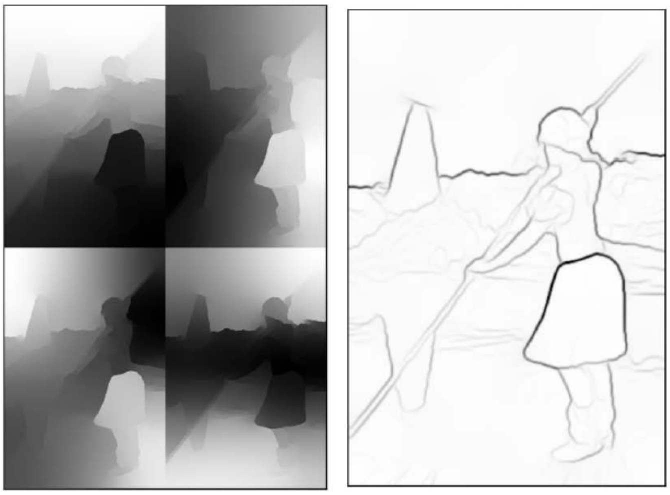


Figure 2.4: Spectral Pb. Figure taken from [AMFM11]. Left image shows first four eigenvectors resulting from spectral clustering. Right image shows the contour signal resulting from computing the gradients of the eigenvectors.

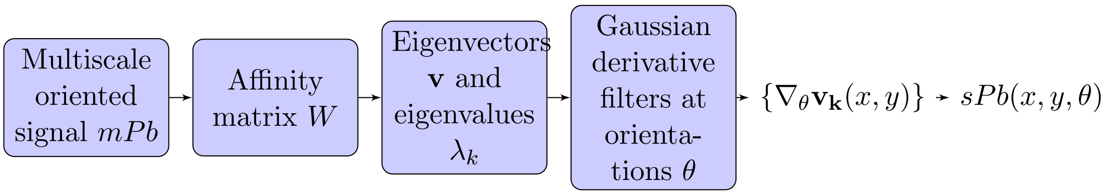

Multiscale oriented signal mPb
Affinity matrix W
Eigenvectors $\mathbf{v}$ and eigenvalues λ_k
Gaussian derivative filters at orientations θ
$\{\nabla_\theta \mathbf{v_k}(x,y)\}$
$sPb(x,y,\theta)$

Figure 2.5: General scheme for spectral component of the gPb contour detector.

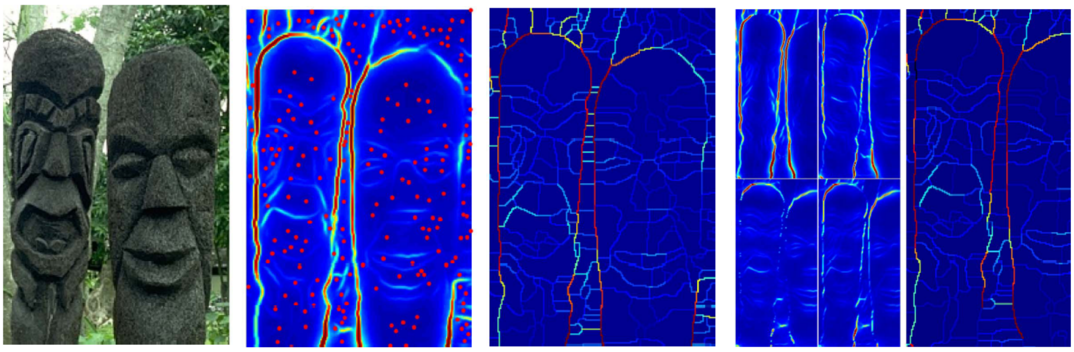


Figure 2.6: Watershed Transform. Figure taken from [AMFM11]. Left image shows the original image. Middle left image shows the non-oriented contour signal $gPb(x,y)$ resulting from the maximum response of $gPb(x,y,\theta)$ and local minima used as seed locations for the Watershed Transform. Middle image shows the weights assigned to watershed arcs by averaging $gPb(x,y)$ along them. This weighting scheme produces artifacts, such as the strong horizontal contours in the small gap between two statues. Middle right image shows $gPb(x,y,\theta)$ for four different orientations. Right image displays the result of reweighting the watershed arcs according to the magnitude of $gPb(x,y,\theta)$ at the orientation given by the arc. As a result, artifacts are suppressed as their orientations do not agree with the $gPb(x,y,\theta)$ signal.

two adjacent regions which are separated by the minimum weight contour. As a result, the base level of this hierarchy respects the weak contours and is an oversegmentation, whereas the upper levels respect only strong contours, resulting in an undersegmentation. Figure 2.7 shows an example of the whole gPb-owt-ucm segmentation algorithm.

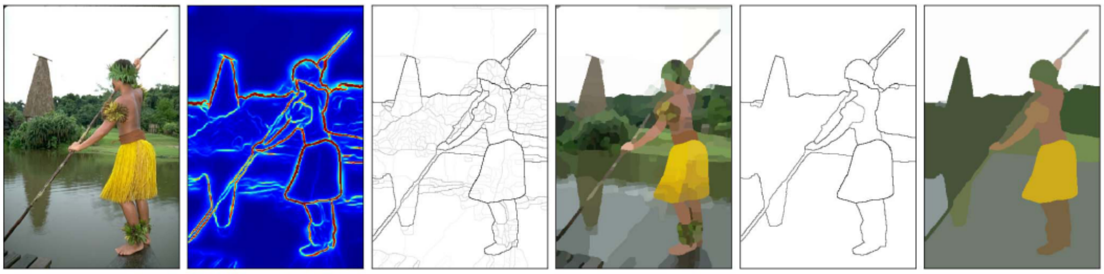


Figure 2.7: Hierarchical segmentation from contours. Figure taken from [AMFM11]. From left to right: image, maximal response of gPb over orientations, weighted contours resulting from OWT using gPb as input, initial oversegmentation resulting from OWT and corresponding to the finest level of UCM, contours obtained by thresholding UCM at level 0.5, and segmentation obtained by thresholding UCM at level 0.5.

2.1.2 Interest points

The interest-point based image representation consists of two parts: (i) the interest points detectors, and (ii) the interest point descriptors. Next, different techniques are described for both parts.

Detectors

An image can be represented as a set of interest points distributed in the image plane. A desirable property of any interest point detector is scale invariance and affine invariance, that is, its result does not vary under a scale change and an affine transformation, respectively. The work in [TM08] presents an exhaustive survey and classification of interest points detectors.

One of the first well-known detectors is the Hessian detector [Bea78]. A grayscale image can be understood as a function from the image plane to $\mathbb{R}$. In this context the Hessian matrix is defined as the matrix containing the second-order image derivatives, which encode information about how the normal to the image surface changes spatially. The trace of this matrix, known as Laplacian, is used to detect interest points.

The Harris detector [HS88] is also widely known and it is based on detecting corner points in the image. To do so, the structure tensor or second moment matrix is computed, which describes the main directions of the gradient of the image at the neighborhood of a given location. The determinant and the trace of this matrix was proposed as a measure of cornerness of the points of the image. The work in [MS04] provided variations of Harris detector that are scale invariant and affine invariant, named Harris Laplace and Harris Affine.

Junctions are the subset of corner points defined by the intersection points of three or more intensity surfaces in an image [Bey91]. The work in [MAFM08] defines the gPb contour detector (introduced in Section 2.1.1) and propose to use it for localizing junctions, which may be viewed as points at which two or more distinct contours intersect. More specifically, given a set of contours C_i specified by the contour detector gPb and their corresponding weights $w_i = |C_i| = \sum_{(x,y)\in C_i} gPb(x,y)$, i.e. the total contrast of contour C_i, junction location for an image neighborhood I_N is computed by an EM-style algorithm:

- Estimate the optimal junction location $L = (x_L, y_L)$ by minimizing its weighted distance from the contours $\{C_i\} \in I_N$

  $$L = argmin_{(x,y)\in I_N} \sum_i w_i d(C_i,(x,y))$$

  where $d(C_i,(x,y))$ is the distance from contour C_i to point (x,y).

- Update the weight w_i of each contour C_i in order to select only those contours passing close to the junction

  $$w_i = |C_i| exp(-d(C_i, L)^2/\epsilon^2)$$

  where ϵ is a factor controlling the distance tolerance

- Repeat the above two steps for a set number of iterations or until the optimal junction location L reaches a fixed point.

Saliency is another distinctive characteristic which has been used in a number of computer vision algorithms. The work in [Gil98] defines saliency in terms of local signal complexity or unpredictability. The idea is to find a point neighborhood with high complexity as a measure of saliency or information content. The method measures the change in entropy of a gray-value histogram computed in a point neighborhood. The search was extended to scale [KB01] and affine [KZB04] parameterized regions. Since color provides additional information, some methods as [IKN98] use color for building saliency maps.

The work in [MCUP02] proposes a technique, known as Maximally Stable Extremal Regions detector (MSER detector), to find correspondences from two images of a scene taken from different points and in different conditions. To do so, the local features they use as anchor are the so-called *extremal regions*, whose set is shown to be closed by perspective transformations and monotonic transformations of intensity, so they are reliable for finding correspondences between views. Extremal regions are found using a region-based image representation known as *Component Tree*.

Efficient implementations The set of interest points in an image can be very large so efficient implementations to compute the interest points are desirable.

The technique called Difference of Gaussians (DoG) [Low04] is used to efficiently approximate the Laplacian used by other detectors, as explained above. It is proved that the Laplacian is equal to the derivative in the scale direction. A difference in scale can be approximated by a blurring by a Gaussian. The derivative is therefore approximated by the difference between the image and a blurred version of it.

The technique known as Speeded Up Robust Features (SURF) [BETVG08] computes a fast approximation of the Hessian matrix based on the well-known integral images introduced in [VJ01].

Descriptors

Apart from the interest point themselves, algorithms based on local features compute local descriptors. A large variety of feature descriptors has been proposed, such as Gaussian derivatives, moment invariants, complex features, steerable filters, phase-based local features, and descriptors representing the distribution of smaller-scale features within the interest point neighbourhood. Based on the latter, one of the most known and used local descriptor is the so-called Scale Invariant Feature Transform (SIFT) [Low04], which is invariant to affine transformations, partially invariant to illumination changes and robust to local geometric distortion. The major stages of computation used to generate the set of local features are the following:

- Scale-space extrema detection. It searches over all scales and image locations to identify potential interest points that are invariant to scale and orientation. It is implemented efficiently by using a difference-of-Gaussian function convolved with the image, $D(x, y, \sigma)$, which can be computed from the difference of two nearby scales separated by a constant multiplicative factor k. In order to detect the local maxima and minima of $D(x, y, \sigma)$, each sample point is compared to its eight neighbors in the current image and nine neighbors in the scale above and below. It is selected only if it is larger than all of these neighbors or smaller than all of them.

- Keypoint localization. At each candidate location, a detailed model is fit to determine location and scale and reject points that have low contrast or are poorly localized along an edge and, therefore, are unstable to small amounts of noise.

- Orientation assignment. One or more orientations are assigned to each keypoint location based on local image gradient directions. Image data is transformed relative to the assigned orientation, scale, and location for each feature to be invariant to these transformations. Dominant directions of local gradients correspond to the peaks in an 36-bin orientation histogram which is formed from the gradient orientations of sample points within a region around the keypoint. The highest peak in the histogram is used to assign the keypoint orientation. Other local peaks within 80% of the highest peak are used to create multiple keypoints at the same scale location and scale but different orientation. Figure 2.8 shows an example where keypoints are displayed as vectors indicating scale, orientation, and location.

- Keypoint descriptor. Local image gradients in a region around each keypoint are transformed into a representation that is invariant to local shape distortion and change in illumination. First, the image gradient magnitudes and orientations are sampled around the keypoint location using its scale and weighted a Gaussian function that gives less emphasis to gradients that are far from the center of the descriptor. Then, 8-bin orientation histograms are created over 4×4 sample regions (see Figure 2.9). Therefore, each keypoint is represented by a 128 $(4 \times 4 \times 8)$ element feature vector. The contribution of each gradient to its corresponding orientation bin depends on the gradient magnitude. However, trilinear interpolation is used to distribute the sample into adjacent histogram bins, avoiding that the descriptor changes abruptly as a sample shifts smoothly from one orientation to another. Furthermore, using less orientation bins than in the orientation assignment stage (36 bins) allows for significant shift in gradient positions. Finally, the feature vector is modified to reduce the effects of illumination change. First, the vector is normalized

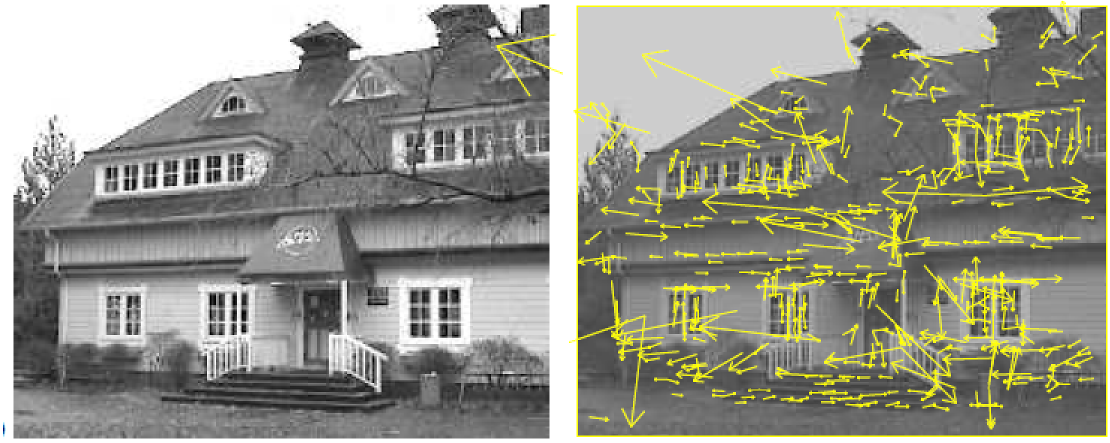


Figure 2.8: Figure taken from [Low04]. The right image shows the keypoints extracted from the original image on the left. Keypoints are displayed as vectors indicating scale, orientation, and location.

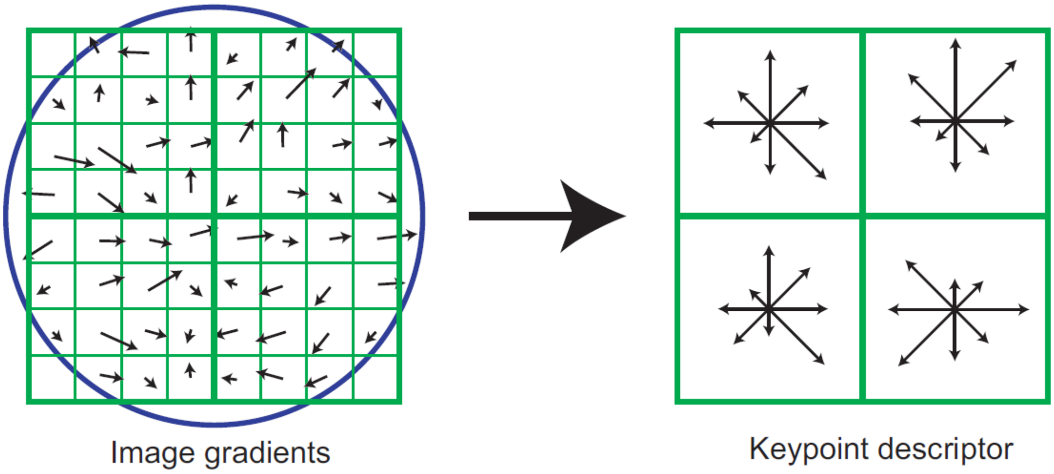


Figure 2.9: Figure taken from [Low04]. A keypoint descriptor is created by first computing the gradient magnitude and orientation at each image sample point in a region around the keypoint location, as shown on the left. These are weighted by a Gaussian window, indicated by the overlaid circle. These samples are then accumulated into orientation histograms summarizing the contents over 4x4 subregions, as shown on the right, with the length of each arrow corresponding to the sum of the gradient magnitudes near that direction within the region. This figure shows a 2×2 descriptor array computed from an 8×8 set of samples, whereas SIFT uses 4x4 descriptors computed from a 16x16 sample array.

> to unit length. Then, values in the unit feature vector are threshold to each be no larger than 0.2 and renormalized to unit length in order to reduce the influence of large gradient magnitudes. This means that the distribution of orientations has greater emphasis than matching the magnitudes for large gradients.

Another descriptor very popular is Speeded Up Robust Features (SURF) [BETVG08]. It describes a distribution of Haar-wavelet responses within the interest point neighborhood. The first step consists of constructing a square region centered around the interest point,

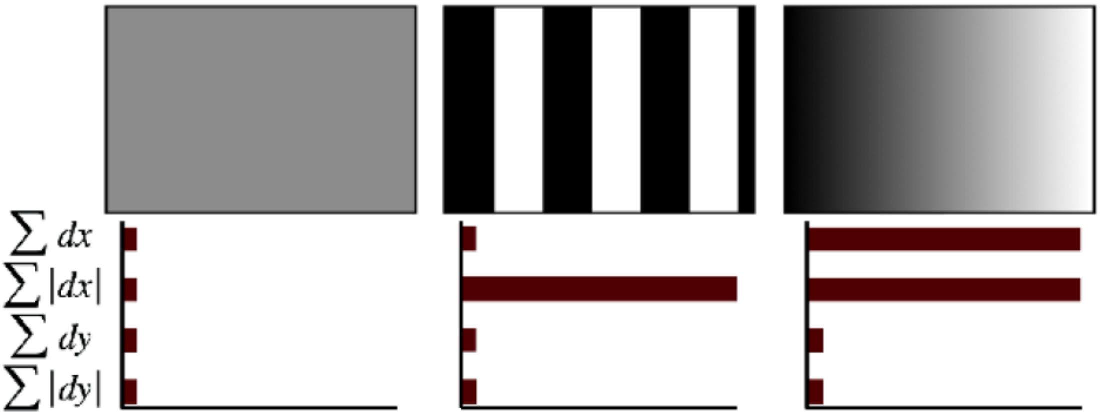


Figure 2.10: Figure taken from [BETVG08]. The descriptor components of a sub-region represent the different image intensity patterns. For a homogeneous region (left), all Haar-wavelet responses are relatively low. For a region of vertical stripes (middle), the value of $\sum |d_x|$ is high, but all other remain low. For a region with gradually increasing intensity in x direction, both values $\sum d_x$ and $\sum |d_x|$ are high.

and oriented along the keypoint orientation. The region is split up regularly into smaller 4×4 square sub-regions and, for each sub-region, a few simple features at 5×5 regularly spaced sample points are computed. Then, horizontal and vertical Haar wavelet responses d_x and d_y are summed up over each subregion, as well as their absolute values $|d_x|$ and $|d_y|$. Hence, each sub-region has a four-dimensional descriptor vector $\mathbf{v}$ for uts underlying intensity structure $\mathbf{v} = (\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|)$. This results in a descriptor vector for all 4×4 sub-regions of length 64. Figure 2.10 shows the properties of the descriptor for three distinctively different image intensity patterns within a subregion. The work in [BETVG08] also proposes an extented version of the SURF descriptor called SURF-128, which adds a couple of similar features. It uses the same sums as before, but splits these values up further. The sums of d_x and $|d_x|$ are computed separately for $d_y < 0$ and $d_y \geq 0$, as well as the sums of d_y and $|d_y|$ are also split up according to the sign of d_x. This extended version is more distinctive and not much slower to compute, but slower to match due to its higher dimensionality. It comes out to perform better.

2.1.3 Other image representations

Bag-of-Features

Traditionally, local features such as SIFT and SURF are clustered in unsupervised ways to create visual words [SZ03], which are equivalent to words for text retrieval. After clustering, each cluster center is taken as a visual word and a corresponding visual vocabulary is generated. Figure 2.11 shows examples of regions belonging to particular clusters, i.e. which will be treated as the same visual word. With the visual vocabulary, each image local feature is assigned its nearest visual word and, therefore, the image can be represented as a vector of visual word frequencies. This image representation is known as Bag-of-VisualWords (BoVW) and as Bag-of-Features (BoF), which is the dual of the extended Bag-of-Words (BoW) used in text retrieval, where each document is represented as a vector of word frequencies. Despite the lack of spatial information, the BoF image

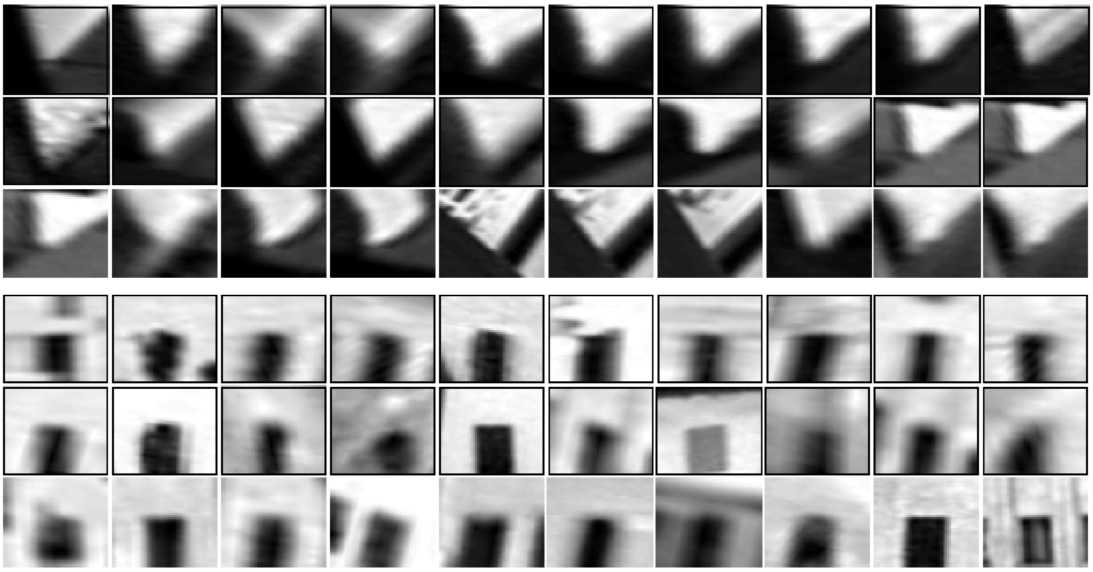


Figure 2.11: Figure taken from [SZ03]. It shows two sets of regions from two different clusters. Each region set is assigned the same visual word.

representation has been very popular in computer vision and visual content analysis in recent years. It has shown promising results for a wide variety of applications such as object recognition, image and video annotation, video event recognition, etc. Furthermore, the framework of traditional information retrieval [BYRN$^+$99], i.e. the inverted file structured indexing and TF-IDF (Term Frequency Inverted Document Frequency) weighting, has been adopted as a solution for large-scale image and video retrieval.

Spatial Pyramid

Bag-of-Features methods, which represent an image as an orderless collection of local features, disregard all information about the spatial layout of the features and are incapable of capturing shape or of segmenting an object from its background. The work in [LSP06] proposes to partition the image into increasingly fine sub-regions and compute histograms of local features found inside each sub-region. Figure 2.12 shows a toy example of a three-level pyramid.

Histograms of Oriented Gradients (HOG)

The basic idea in [DT05] is that local object appearance and shape can often be characterized by the distribution of local instensity gradients or edge directions, even without precise knowledge of the corresponding gradient or edge positions. It is usually computed to represent image windows rather than the whole image. Thus, in object detection and retrieval applications, classic exhaustive window sliding approach is usually combined with this representation. In practice, this is implemented by dividing the image window into small spatial regions, which are called cells. Then, for each cell, a local 1-D histogram of gradient directions or edge orientations is computed over the pixels of the cell. Each pixel calculates a weighted vote for an edge orientation histogram channel based on the orientation of the gradient element centred on it, and the votes are accumulated into

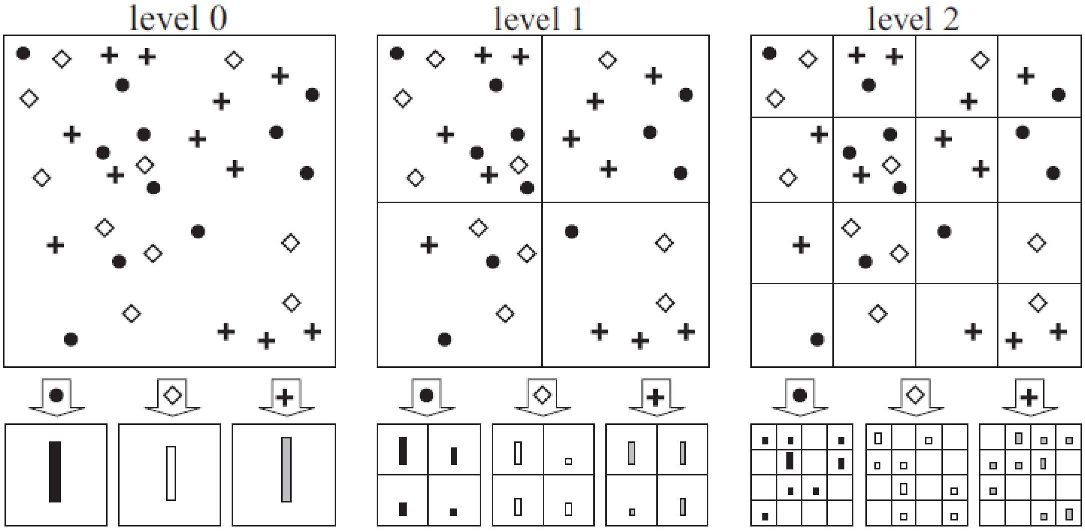


Figure 2.12: Figure taken from [LSP06]. Toy example of a three-level pyramid. The image has three feature types, indicated by circles, diamonds, and crosses. At the top, we subdivide the image at three different levels of resolution. Next, for each level of resolution and each channel, we count the features that fall in each spatial bin.

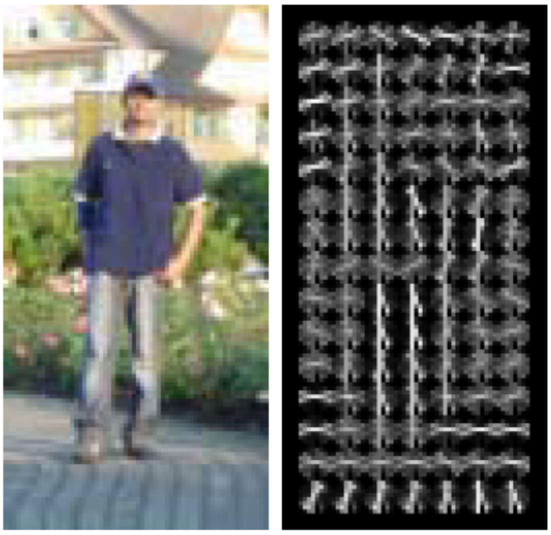


Figure 2.13: Figure taken from [DT05]. Image of a pedestrian (left) is divided into 7×15 cells, over which 1-D histograms of gradient directions are computed (right).

orientation bins over the cells. The combined histogram entries form the representation. Figure 2.13 shows an example of the HOG descriptor for an image of a pedestrian. In fact, HOG descriptor was designed for human detection. SIFT descriptor [Low04] was one of its precursors in the use of orientation histograms.

Pyramid of Histograms of Orientation Gradients (PHOG)

The work in [BZM07] proposes an image descriptor named Pyramid of Histograms of Orientation Gradients (PHOG), which is mainly inspired by two sources: (*i*) the image

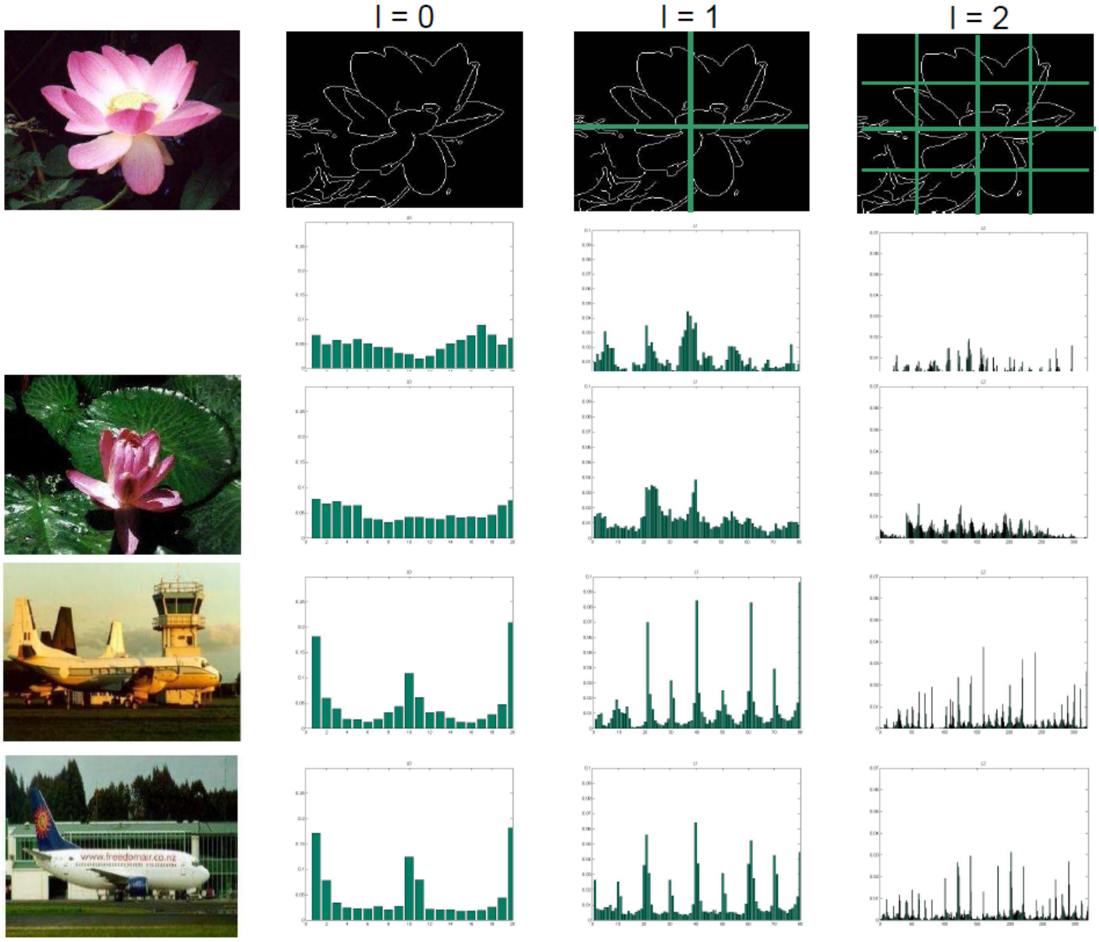


Figure 2.14: Figure taken from [BZM07]. Top row images shows an input image and grids for three levels over its contour image. The image below its level represent the concatenation of the histograms of oriented gradients computed over the grid cells at the corresponding level. The following row shows an example for an image of the same category (flower) and the last two rows two images from a different category (aeroplanes).

pyramid representation [LSP06], and (*ii*) the Histogram of Oriented Gradients (HOG) [DT05]. The PHOG descriptor represents an image by its local shape and the spatial layout, together with a spatial pyramid kernel. It consists of a histogram of orientation gradients over each image subregion at different resolution levels. Each image is divided into a sequence of increasingly finer spatial grids by repeatedly doubling the number of divisions in each axis direction. The idea is illustrated in Figure 2.14. If only the coarsest level is used, then the descriptor reduces to a global edge or orientation histogram. On the other hand, if only the finest level is used, the descriptor captures the distribution of local intensity gradients or edge directions, such as HOG descriptor [DT05]. The final PHOG descriptor for the image is a concatenation of all the HOG vectors computed over them.

2.2 Proposed approach

We propose to analyze if there exists any relationship between the gPb-owt-ucm segmentation algorithm and interest points. In such a case, we would like to study if one family could benefit from the other, in both directions.

First, we consider how gPb-owt-ucm could benefit from interest points. In this direction, we propose to analyze whether the interest points scales could be used to adaptively select the appropriate scale at each location. This idea emerges from [AMFM11], in which the authors detected that high resolution images of complex scenes require more than a naive weighted average of signals across the scale range to achive good detection of contours at any scale. This is because such an average (see Equation 2.1) blurs information, resulting in good performance for medius-scale contours, but poor detection of both fine-scale and large-scale contours. The authors claimed that it would be desirable to adaptively select the appropriate scale at each location.

Second, we consider how interest points could take advantage of gPb-owt-ucm. In this direction, we propose to analyze whether a hierarchy of interest points obtained from gPb-owt-ucm could improve object retrieval performance. This idea is based on the intuition that interest points located on strong contours could be more relevant than the ones located on smoother contours. In such case, we could use from $gPb(x, y, \theta)$, which has rich information about contours strength at different orientations, to establish a hierarchy among the interest points. More formally, we propose two ways of weighting a keypoint:

- Without considering the keypoint orientation:

$$W(k_i) = \max_{\theta} gPb(x(k_i), y(k_i), \theta) \tag{2.3}$$

  where $W(k_i)$ is the weight assigned to the keypoint k_i, θ refers to the different orientations considered by gPb contour detector, and $x(k_i)$ and $y(k_i)$ are the location coordinates of the keypoint k_i. The weight is assigned taking the contour strength at the orientation which gives the maximum value.

- Considering the keypoint orientation:

$$W(k_i) = gPb(x(k_i), y(k_i), \theta(k_i)) \tag{2.4}$$

  where $\theta(k_i)$ is the keypoint orientation. Therefore, the weight is assigned taking the contour strenght at the orientation given by the keypoint.

Once each keypoint has been assigned a weight, we could sort them and create a hierarchy where keypoints with the largest weights would form the upper part, whereas the ones with the smallest weights would belong to the bottom of the hierarchy. This hierarchy can be used in the matching process for object retrieval or in the classification process for object recognition as a top-down approach, making more robust the process and decreasing the computational cost. Suppose that we have a hierarchy of L levels, where level $l = 0$ consists of the keypoints with the smallest weights and level $l = L-1$ contains the ones with the largest weights. For object retrieval, matching between keypoints from level l is only considered if matching between between keypoints from level $l+1$ exceeds a minimum threshold. Analogously, for object recognition, each classifier is applied over keypoints at level l only if the classifier at level $l+1$ exceeds a minimum threshold.

For object retrieval, another way would be using directly the weights instead of creating a hierarchy of keypoints:

$$d(O_q, O_t) = \sum_i W(k_i) d(k_i, m(k_i)) \tag{2.5}$$

where O_q and O_t are the query object and a target object from the database respectively, k_i is the ith keypoint from O_q, $W(k_i)$ is its weight, and $m(k_i)$ is its matched keypoint from O_t.

Furthermore, we also propose another way to benefit from gPb-owt-ucm segmentation algorithm, which consists in using the junctions detected as interest points as in [MAFM08]. Junctions are defined as the points where watershed arcs intersect. Since each watershed arc is assigned a weight, each junction can also be assigned a weight as a function of the weights corresponding to the arcs from which the junction is created. Therefore, we propose to analyze which would be the impact on object retrieval performance of replacing interest points by the junctions detected using the gPb contour detector. A similar approach for object recognition using junctions has been adopted in [WBW+10].

2.3 Experiments

With regard to the evaluation of gPb-owt-ucm benefiting from interest points, we propose to use the precision-recall framework [MFM04] on the Berkeley Segmentation Datasets (BSDS300 and BSDS500) [MFTM01]. This benchmark operates by comparing machine generated contours to human ground-truth data and allows evaluation of segmentations in the same framework by regarding region boundaries as contours.

On the other hand, we propose to study the object retrieval performance by using only the interest points at an increasing depth in the hierarchy. Therefore, given a hierarchy of keypoints with L levels, we will compute the retrieval performance using keypoints belonging to top level $L-1$, keypoints belonging to the two uppermost levels $L-1, L-2$ and so on until considering keypoints from all levels $L-1, L-2, ..., 1, 0$. We expect that the upper the level, the more discriminative their keypoints. In other words, the performance increase of including keypoints from level $l-1$ to keypoints from levels $L-1, ..., l$ decreases as l also decreases. We plan to use the TRECVID benchmark dataset provided for the Instance Search task [SOK06].

With regard to object recognition, we also plan to analyze the impact of using a hierarchy of keypoints in bechmark datasets such as MSRC [SWRC06] and PASCAL [EVGW+10].

Chapter 3

Use of context in object retrieval and recognition

3.1 Related work

In object recognition and retrieval, there have been suggestions that a bounding box may be able to provide some degree of context and may actually be beneficial. Many state-of-the-art techniques are based on exhaustive search over the image by using a sliding window with multiple scales to find the best object positions [DT05] [FGMR10] [HJS09] [VJ04] [ZCYF10]. [DT05] proposes grids of histograms of oriented gradient (HOG) descriptors for human detection and uses a detection window which includes a margin around the person on all four sides to provide a significant amount of content that helps detection. Experimental results showed that decreasing the context decreases the performance. The approach of [FGMR10] builds on a framework that represents objects by a collection of parts arranged in a deformable configuration, which is referred to as deformable part models. These models are trained using a discriminative procedure that only requires bounding boxes for the objects. [HJS09] proposes a two stage sliding window object localization method that combines the efficiency of a linear classifier for pre-selection with the robustness of a sophisticated non-linear one for scoring. [VJ04] describes a face detector framework that uses a method for combining successively more complex classifiers in a cascade structure over sub-windows at different locations and scales. [ZCYF10] presents a latent hierarchical learning method for object detection where an object is represented by a 3-layer tree structure model. The first layer has one root node that represents the entire object. The root node has 9 child nodes at the second layer in a 3 by 3 grid layput, and each of them has 4 child nodes at the third layer. All tree nodes are rectangular windows over which HOG descriptors are computed.

However, one of the contributions of [ZMLS07] is the evaluation of background features, which shows the pitfalls of training on datasets with uncluttered or highly correlated backgrounds. Thus, this causes overfitting and yields disappointing results on test sets with more complex backgrounds. Their experiments reveal that the features on the objects themselves play the key role for recognition. Thus, using foreground (FF) and background (BF) features together does not improve the performance despite the discriminative information contained by backgrounds. To determine whether background features provides additional cues for classification, they examine the change in performance when the original background features from an image are replaced by two specially constructed alternative sets: random (BF-RAND) and constant natural scene (BF-CONST) backgrounds. They carry out the experiments for the following set ups:

- FF. It denotes the foreground features.
- AF. It denotes the features extracted from the original image, i.e., a combination of FF and BF.
- AF-RAND. It denotes the combination of FF and BF-RAND.
- BF-RAND. It denotes the combination of FF and BF-CONST.

In order of decreasing performance, these combinations are: FF, AF-CONST, AF, AFRAND. FF always gives the highest results, indicating that object features play the key role for recognition, and recognition with segmented images achieves better performance than without segmentation. Mixing background features with foreground features does not give higher recognition rates than FF alone.

In [ME07], it is confirmed that correct spatial support is important for object recognition. Thus, knowing the right spatial support leads to substantually better recognition performance for a large number of object categories, especially those that are not well approximated by a rectangle, such as sheep, bike and airplane object categories. Although classic rectangular sliding window approaches are known for outstanding results on faces, pedestrians, and front/side views of cars (all rectangular-shaped objects), they have trouble distinguishing foreground from background when the bounding box does not cover well an object (see Figure 3.1). They have also demonstrated remarkable performance recognizing more complicated categories, but in datasets such as Caltech-101 where there is a single object per image and with relatively correlated backgrounds. In [ME07], for each object in an anotated dataset (Microsoft Research Cambridge dataset), they estimate its class label in two scenarios: (*i*) using only the pixels inside the object's ground-truth support region, and (*ii*) using all pixels in the object's tight bounding box. Experiments show that objects that are poorly approximated by rectangles see the largest improvement (over 50%) when object's ground-truth support regions are used and that categories that do not show improvement with better spatial support already have remarkable performance. Overall, the recognition performance using ground-truth segments is 15% better than using the bounding boxes. In the same direction, in [vdSUGS11] the authors have adopted segmentation as a selective search strategy for object recognition. Instead of an exhaustive search, which needs constraining the computation per location, using segmentation to generate a limited set of locations allows to compute the more powerful yet expensive BoF.

In [RVG$^+$07], the authors go further and propose integrating segmentation into the BoF framework [FPZ03] considering each region as a stand-alone image by masking and zero padding the original image. Then the signature of the region is computed as in regular BoF, but discarding any feature that falls entirely outside its boundary. As a consequence, masking greatly enhances the contrast of the region boundaries making features along the boundaries more shape-informative. Furthermore, coarse spatial information is

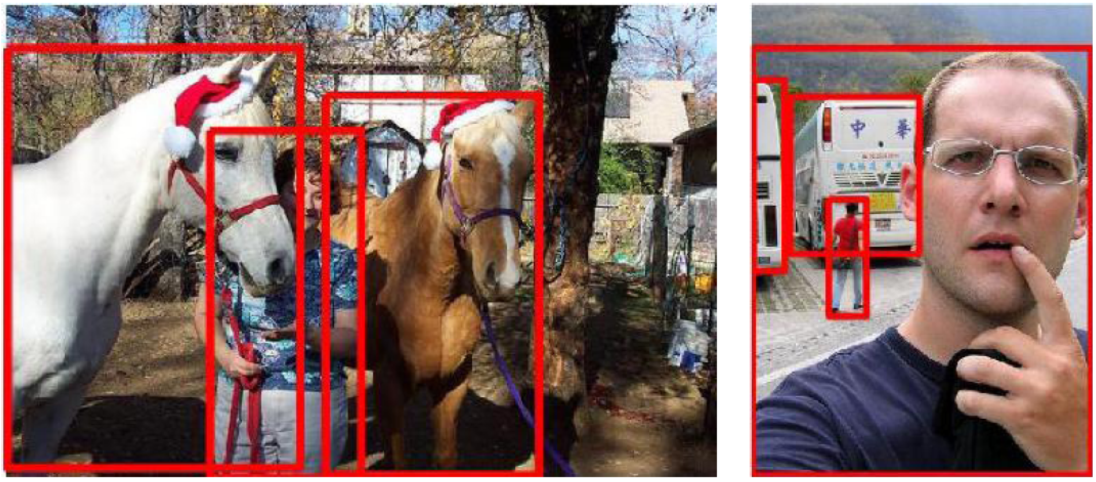


Figure 3.1: Figure taken from [ME07]. Examples from Pascal dataset where up to half of the bounding box pixels do not belong to the object of interest.

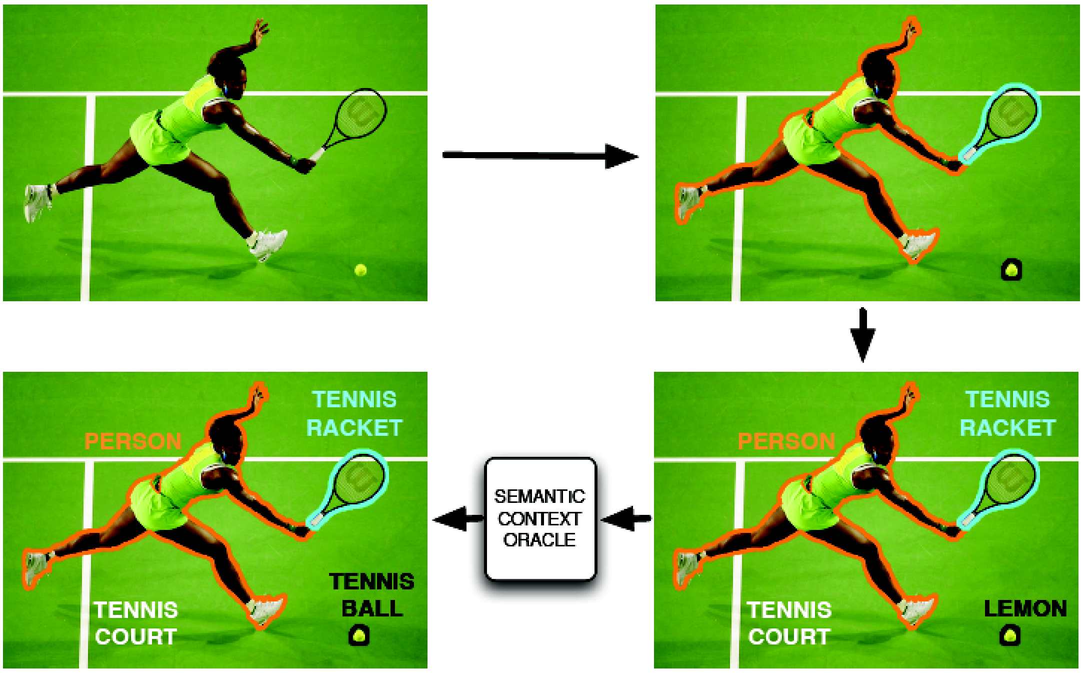


Figure 3.2: Figure taken from [RVG^{+}07]. This image shows how a mis-labeled object is detected and changed by enforcing semantic context.

incorporated by clustering features in segments. However, once each region has been labeled as an object category, semantic context is used as post-processing to assign objects' category labels with respect to other objects in the scene. Figure 3.2 shows an image where objects' category labels 'Tennis court', 'Tennis racket', 'Person' and 'Lemon' have been initially assigned. Then, using semantic context, 'Lemon' label changes to 'Tennis ball' since this label fits in context with the other labels more precisely.

According to [Ram07], approaches based on scanning-window template classifiers can be

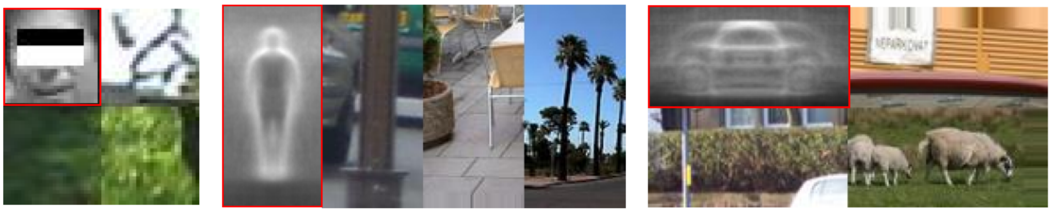


Figure 3.3: Figure taken from [Ram07]. Examples of false positives for a face detector (left), a pedestrian detector (middle), and a car detector (right) using scanning-window template classifiers.

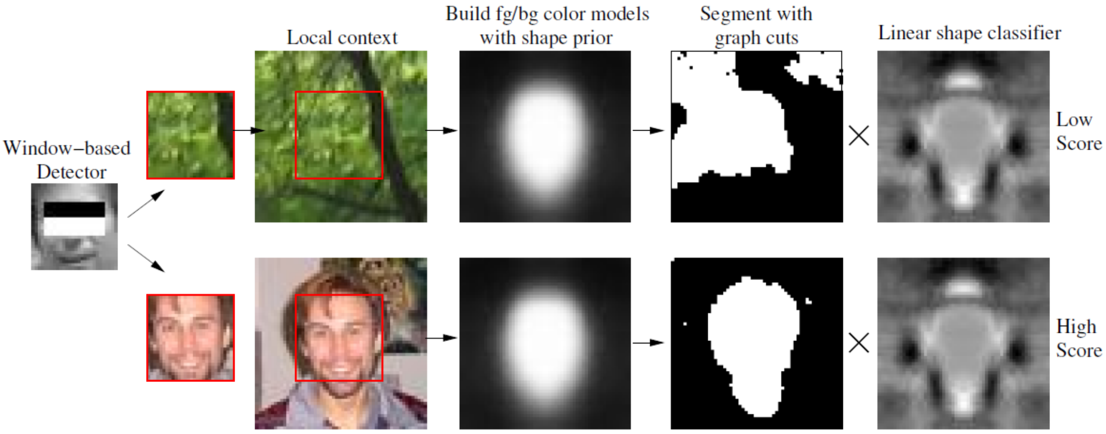


Figure 3.4: Figure taken from [Ram07]. Local figure-ground segmentation for the foliage is detected as a false positive when it is fed into a linear shape classifier. On the other hand, face is correctly detected as a true positive.

hindered by their lack of explicit encoding of object shape, resulting in high false-positives. Figure 3.3 shows examples where a face detector becomes confused by edges in foliage, a pedestrian detector mistakens strong vertical edges for a person, whereas a car detector mistakens strong horizontal edges. The authors propose to use the scanning-window template classifiers to generate possible object locations and compute a local figure-ground segmentation at each hypothesized detection to prune away those hypotheses with bad segmentations (see Figure 3.4).

[DHH$^+$09] presents an empirical evaluation of the role of context in object detection, evaluating several sources of different context and ways to utilize it.

3.2 Proposed approach

Considering each region as a stand-alone image by masking and zero padding the original image as in [RVG$^+$07], the local features computed in each region do not depend on the context. Therefore, if two images contain the same object but with different backgrounds and assuming that the object is rightly segmented, the local features for both objects will be the same. However, experimental results in [RVG$^+$07] did not estimate which was the influence of masking the region for feature extraction, since a more general scheme including masking was evaluated.

Therefore, we propose to analyze which is the influence of the context in keypoint detection. We expect that for different instances of the same object in different contexts, the locations of the keypoints will be more unstable if the object is not masked. In addition to this, context is also expected to have an impact on the description of the detected keypoints. In [MS05], local descriptors robustness to image transformations such as rotation, scale and viewpoint changes, image blur, JPEG compression, and illumination is evaluated. The number of correct correspondences is determined with the overlap error, which measures how well the regions correspond under a transformation. It is defined by the ratio of the intersection and union of the regions. With regard to the keypoints description, the evaluation criterion is based on the number of correct matches and the number of false matches obtained for an image pair. Based on [MS05], overlap error will be used to objectively measure the context influence on the keypoints location stability. Since we are measuring the context influence, the image transformation will consist in changing the background as in [ZMLS07]. Therefore, no affine transformation is applied to keypoint regions when compared by the overlap error. Overlap error will be minimum if keypoint location has not changed due to the background change. On the other hand, keypoint descriptors robustness will be measured as in [MS05]. Furthermore, since our final application will be object retrieval, we will also measure the impact of context in object retrieval benchmarks such as the Instance Search Task of Trecvid.

On the other hand, the idea of zero padding proposed in [RVG$^+$07] seems not to be the best way to get rid of the context. They expected that the masking and zero padding process made features along boundaries more shape-informative. However, this claim depends clearly on the colour of the object. In an extreme case, no local features should be detected in an object totally black, independently of the object shape. Therefore, we think that other strategies for background should be employed to keep the shape information.

3.3 Experiments

We plan to evaluate the impact of the context in object retrieval using the TRECVID benchmark dataset provided for the Instance Search task. The goal of this task is to retrieve the videos which contain a particular object. Some instances of the object as well as their masks are given as visual examples. The task is performed for a set of objects to be retrieved. The decision of evaluate the use of context in object retrieval instead of object recognition has also been motivated by the fact that objects to be retrieved are instances of the same query object, whereas in object recognition datasets there exist a large intra-class variance, which would blur the real impact of the context.

For both query and target images we propose the following set ups:

- Feature detection. Whether regions are masked for keypoints detection or not.
- Feature description. Whether regions are masked for keypoints description or not.

Keypoint detection and description invariance to context will be evaluated in an image dataset on a benchmark based on [MS05]. For each image, a set of images with different backgrounds but with the same object will be generated. Then, overlap error will be used to measure the keypoint detection invariance, whereas keypoint description invariance will be evaluated through a matching process.

Finally, impact of context will be also evaluated in object recognition. We plan to evaluate whether object recognition performance improves when no context is used neither for learning objects' classifiers nor for object detection.

Chapter 4

Bundling interest points with regions

4.1 Related work

Classic visual vocabularies are generated from single image local descriptors. However, these vocabularies are not able to capture the rich spatial contextual information among the local features. In fact, several works have verified that modeling these visual contexts could greatly improve the performance of many visual matching and recognition algorithms. Basically, two differents ways of considering such spatial context can be distinguish: (*i*) algorithms that apply a post geometric verification step [SZ06], and (*ii*) algorithms which consider combination of local features [WKIS09][ZHH$^+$10]. As it will be proposed in Section 4.2, we want to analyze the benefits from using regions as spatial supports for bundling interest points. This is the reason why we briefly describe some previous works based on combination of local features. Generally, considering visual words in groups rather than single visual word could effectively capture the spatial configuration among them. On the other hand, geometric verification is computationally expensive and, therefore, in practice, it is only applied to a subset of the top-ranked candidates.

In [WKIS09], a scheme where local features were bundled into local groups was presented based on the ideas that (*i*) each group of bundled features becomes much more discriminative than a single feature, and (*ii*) within each group simple and robust geometric constraints can be efficiently enforced. The idea of bundling features arises from the decrease of the discriminative power of quantized local features when a single local feature needs matching to billions of local feature in large scale databases, resulting in many false positive matches between individual features. Although a straightforward way to increase the discriminative power is to increase its region size and/or the dimensionality of the descriptor, a larger feature has a lower localization accuracy and it is more sensitive to occlusion and image variations. Unlike a single large feature, a bundled feature provides a flexible representation that allows to partially match two groups of local features, which can have different number of local features with only a subset of them matched. In [WKIS09], SIFT features that falls inside a MSER detection (support region over which SIFT descriptor is computed) are bundled. In fact, ellipses of the

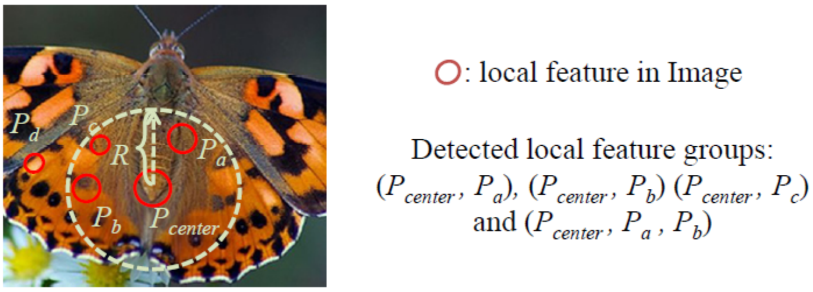

P_d
P
R
P_a
P_b
P_center
○: local feature in Image
Detected local feature groups:
(P_center, P_a), (P_center, P_b) (P_center, P_c)
and (P_center, P_a, P_b)

Figure 4.1: Figure taken from [ZHH$^+$10]. At the centered local feature P_{center}, three groups containing two local features are detected. Furthermore, one group of three local features containing the two closest ones is also detected.

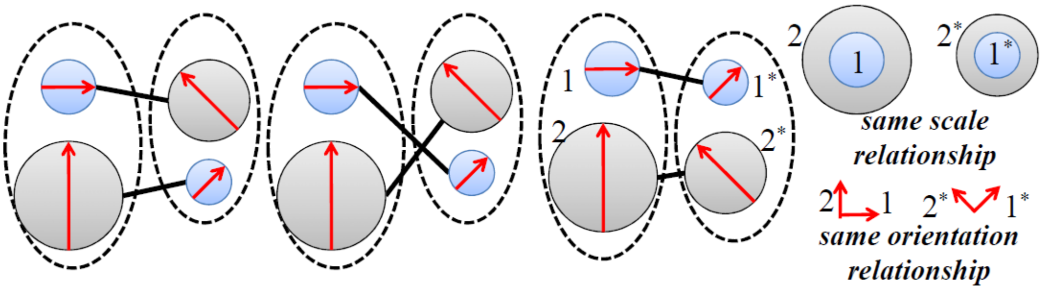
bundling MSER are slightly enlarged. The authors define a matching score between two bundled features that consists of two terms: (i) a membership term, and (ii) a geometric term. The former uses the number of common visual words. The latter performs a weak geometric verification using relative ordering, measuring the consistency between the order before matching and the order after matching. Experiments in web image search, with a database of more than one million images, show that this scheme achieves a 49% improvement in average precision over the baseline bag-of-words approach.

In [ZHH$^+$10], two or three local features are considered in each local feature group since if too many local features are combined, the repeatability of the combination will decrease. In addition, if more local features are contained in each group, there would be more possible feature-to-feature matches between two groups. Each local feature group is formed by a centered local feature and one or two other local features within a circle of radius proportional to the scale of the centered local feature. For local feature groups containing two local features, each centered local feature provides as many local feature groups as the number of local features within the circle. On the other hand, local feature groups containing three local features are built considering only the two nearest local features within the circle. Figure 4.1 shows an example where three groups containing two local features and one group of three local features are detected on a centered local feature. The distance between two local feature groups take into account the spatial context of each local feature group, which is defined as the orientation and scale relationships between the local features inside the group. They further learn a discriminant distance metric between local feature groups by collapsing local features with same semantic labels. Therefore, the metric learning puts more efforts on those local feature groups with same semantic label but small spatial contextual similarities. Figure 4.2 shows an example of two groups of local features that have spatial consistency according to the orientation and scale relationships among their local features.

4.2 Proposed approach

Discriminative visual phrases were proposed in [ZTH$^+$09] to refer to the frequently co-occurring visual word pairs. If two visual words frequently co-occur within short spatial distance in images containing the same object but different backgrounds, spatially consistent visual words are more likely to be located on the object.

Assuming this, we propose to work with regions resulting from a robust contour detector,



(a): The two match orders if each local feature group contains 2 local features

(b): The best match order based on the spatial similarity

Figure 4.2: Figure taken from [ZHH+10]. When comparing spatially two groups of two local features, there are two possible matchings (showed on the left). When first matching is considered, local matched features have the same orientation and scale relationships.

such as gPb, since they can provide a good suport for bundling local features, ensuring that local features being groupped belong to a support area with minimal boundary strength. Furthermore, any boundary strength threshold on the gPb-owt-ucm segmentation algorithm generates an image partition whose regions contain internal boundaries which are smoother than the given threshold. Therefore, using regions provided by gPb-owt-ucm we can guarantee that local features being bundled have not boundaries between them that are stronger than the given threshold. In this way, the probability of local features from the object and from the background being groupped together decreases. In contrast, previous works [ZHH+10][WKIS09] do not consider contour information when bundling local features to the same group.

Once segmented regions are adopted as spatial supports for bundling local features, the question of how to describe the region from the set of local features arises. Therefore, we plan to study if support regions are enough to encode the spatial context and, therefore, classic BoF could represent the region content, or geometric relationships between the local features within the same group should be also encoded. As pointed in [ZHH+10], scale and orientation of keypoints can also be considered to encode spatial context.

4.3 Experiments

We propose to evaluate the performance for different set ups which consider: (i) combination of local features, and (ii) the representation form.

With regard to the combination of local features, we propose three different cases:

- Individual local features. This is the classic approach for both object retrieval and recognition.

- Bundling local features within the support region given by the interest point detector [ZHH+10] [WKIS09].

- Bundling local features within gPb-owt-ucm regions from different hierarchical levels.

With regard to the representation form for a group of local features, the options can be mainly divided in two categories:

- No spatial information. In such case, region is consider to inherently convey enough spatial information. Classic BoF approach can be used to describe the region.
- With spatial information. Spatial coding techniques are used to enforce some geometric constraints when matching two group of local features.

Chapter 5

Work in progress

We have carried out some preliminary experiments on using regions for bundling interest points. The images showed in Figure 5.1 belong to the aeroplane object category of the Pascal VOC 2012 Dataset. The goal of the experiment is to detect and localize the object (aeroplane) in the target image (right image of Figure 5.1) by using a visual example (left image of Figure 5.1). Keypoints and local descriptors have been obtained on each image by using the default implementation of SIFT descriptor available in OpenCV. Figure 5.1 shows the results of matching individually the keypoints and keeping the best matches. On the other hand, Figure 5.2 shows the results of region matching. For this, both query and target images have been first segmented by the gPb-owt algorithm which gives a finest partition. Then, SIFT features belonging to the each region are bundled and matched. For these preliminary experiments, the distance between two local feature groups G_1 and G_2 has been computed by averaging the distance values from each local feature of G_1 to its nearest feature of G_2. One of the advantages of using segmented regions during the matching process is that provides a hypotheses of the object segmentation. From these preliminary experiments, we also propose to consider spatial consistency among the regions during the matching process. Therefore, if two regions have been matched as well as their neighbour regions, then the probabilty of being a true match increases. In this way, some spatial compactness is enforced, avoiding isolated regions being matched.

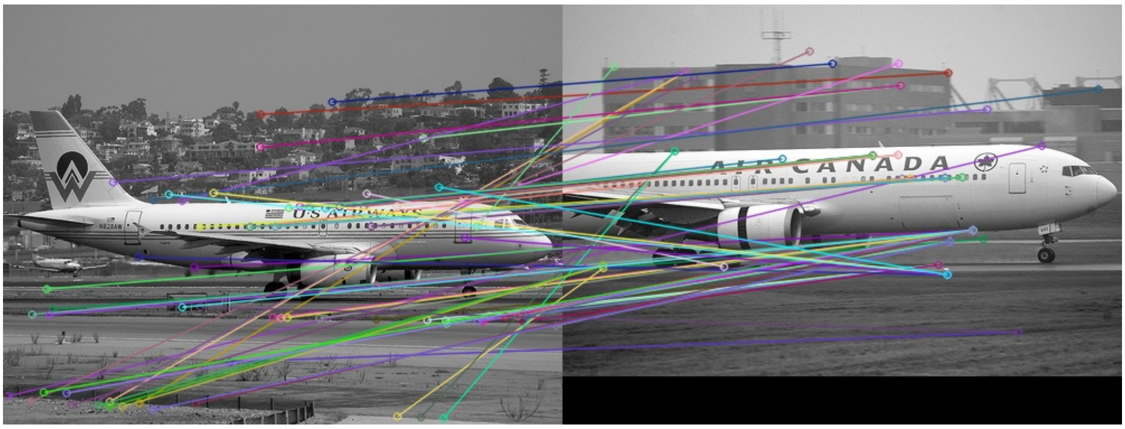


Figure 5.1: Matching individually SIFT descriptors

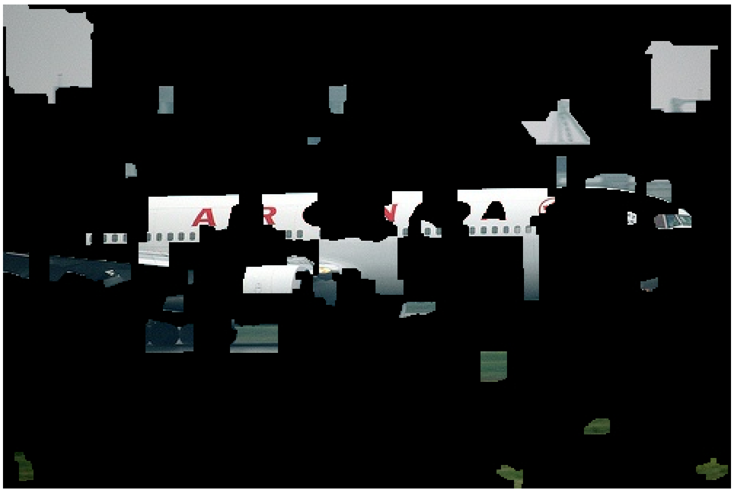
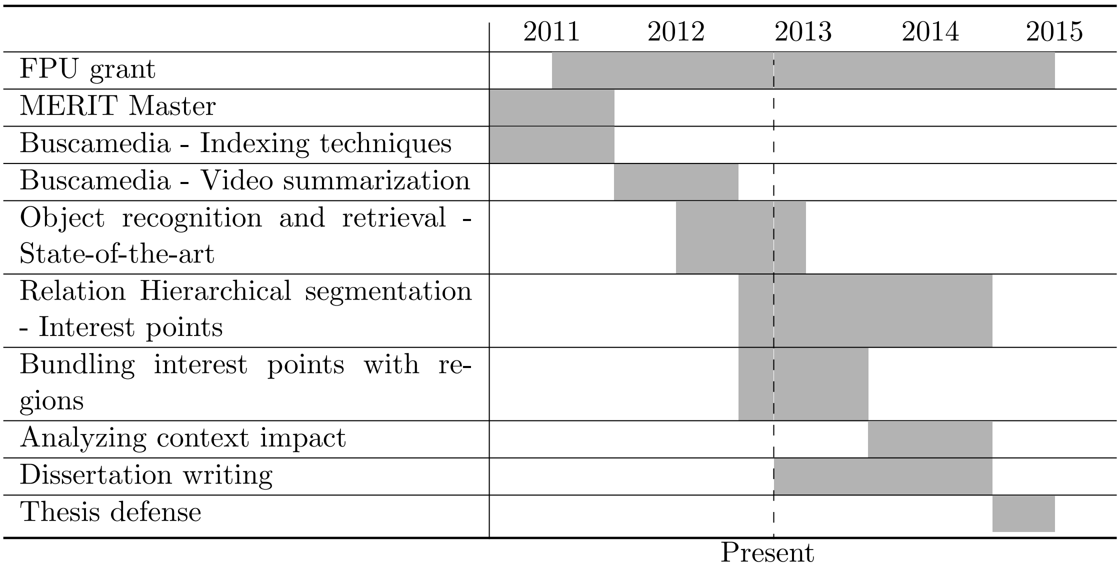


Figure 5.2: Matching region-bundled SIFT descriptors

Chapter 6

Work Plan and Progress

The following table presents the work plan during the 4-year period that lasts the FPU grant from the Spanish Science and Innovation Ministry that the author holds. The progress done in the past is also reflected in the table. The author has been working for two years in other computer vision fiels, such as indexing techniques and video summarization, due to the involvement in a research project named Buscamedia in collaboration with the Corporació Catalana de Mitjans Audiovisuals (CCMA). Previously, the author had also been involved in the i3media project designing an image retrieval system based on MPEG-7 visual descriptors. Once Buscamedia project has already finished, the research work has turned to object retrieval and recognition since these topics are richer and more appealing for research.



The scientific publications done so far are the following:

- [In review process] C. Ventura, V. Vilaplana, X. Giro, F. Marques. *Improving Retrieval Accuracy of Hierarchical Cellular Trees for Generic Metric Spaces.*, Multimedia Tools and Applications 2013.

- [To be published] C. Ventura, X. Giro, V. Vilaplana, D. Giribet, E. Carasusan. *Automatic Keyframe Selection based on Mutual Reinforcement Algorithm*, International Workshop on Content-Based Multimedia Indexing (CBMI) 2013.
- C. Ventura, M. Martos, X. Giro, V. Vilaplana, F. Marques. *Hierarchical Navigation and Visual Search for Video Keyframe Retrieval*, Advances in Multimedia Modeling. Springer Berlin / Heidelberg; 2012 p. 652-654.
- X. Giro, C. Ventura, J. Pont-Tuset, S. Cortes, and F. Marques, *System architecture of a web service for content-based image retrieval*, ACM International Conference On Image And Video Retrieval 2010, 2010, p. 358-365.

The author has also been involved in the following projects:

- Project CENIT-2009-1026 BuscaMedia: Hacia una adaptacin semntica de medios digitales multirred-multiterminal, BUSCAMEDIA (http://www.cenitbuscamedia.es). In collaboration with the Corporació Catalana de Mitjans Audiovisuals (CCMA). Involved from 2011.
- Project CENIT-CENIT-2007-1012 Tecnologas para la creacin y gestin automtica de contenidos audiovisuales inteligentes, I3MEDIA (https://i3media.barcelonamedia.org). In collaboration with the Corporació Catalana de Mitjans Audiovisuals (CCMA) and MediaPro.

Bibliography

[AMFM11] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik, *Contour detection and hierarchical image segmentation*, Pattern Analysis and Machine Intelligence, IEEE Transactions on **33** (2011), no. 5, 898–916.

[Bea78] P. R. Beaudet, *Rotationally invariant image operators*, Proceedings of the 4th International Joint Conference on Pattern Recognition, 1978.

[BETVG08] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool, *Speeded-up robust features (surf)*, Computer vision and image understanding **110** (2008), no. 3, 346–359.

[Bey91] D.J. Beymer, *Finding junctions using the image gradient*, Computer Vision and Pattern Recognition, 1991. Proceedings CVPR '91., IEEE Computer Society Conference on, 1991, pp. 720–721.

[BYRN$^+$99] Ricardo Baeza-Yates, Berthier Ribeiro-Neto, et al., *Modern information retrieval*, vol. 463, ACM press New York, 1999.

[BZM07] Anna Bosch, Andrew Zisserman, and Xavier Munoz, *Representing shape with a spatial pyramid kernel*, Proceedings of the 6th ACM international conference on Image and video retrieval, ACM, 2007, pp. 401–408.

[DHH$^+$09] Santosh K Divvala, Derek Hoiem, James H Hays, Alexei A Efros, and Martial Hebert, *An empirical study of context in object detection*, Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, IEEE, 2009, pp. 1271–1278.

[DT05] Navneet Dalal and Bill Triggs, *Histograms of oriented gradients for human detection*, Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, vol. 1, IEEE, 2005, pp. 886–893.

[EVGW$^+$10] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman, *The pascal visual object classes (voc) challenge*, International journal of computer vision **88** (2010), no. 2, 303–338.

[FGMR10] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan, *Object detection with discriminatively trained part-based models*, Pattern Analysis and Machine Intelligence, IEEE Transactions on **32** (2010), no. 9, 1627–1645.

[FPZ03] Rob Fergus, Pietro Perona, and Andrew Zisserman, *Object class recognition by unsupervised scale-invariant learning*, Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on, vol. 2, IEEE, 2003, pp. II–264.

[Gil98] S. Gilles, *Robust description and matching of image*, PhD thesis, University of Oxford, 1998.

[HJS09] Hedi Harzallah, Frédéric Jurie, and Cordelia Schmid, *Combining efficient object localization and image classification*, Computer Vision, 2009 IEEE 12th International Conference on, IEEE, 2009, pp. 237–244.

[HS88] Chris Harris and Mike Stephens, *A combined corner and edge detector*, Alvey vision conference, vol. 15, Manchester, UK, 1988, p. 50.

[IKN98] Laurent Itti, Christof Koch, and Ernst Niebur, *A model of saliency-based visual attention for rapid scene analysis*, Pattern Analysis and Machine Intelligence, IEEE Transactions on **20** (1998), no. 11, 1254–1259.

[KB01] Timor Kadir and Michael Brady, *Saliency, scale and image description*, International Journal of Computer Vision **45** (2001), no. 2, 83–105.

[KZB04] Timor Kadir, Andrew Zisserman, and Michael Brady, *An affine invariant salient region detector*, Computer Vision-ECCV 2004, Springer, 2004, pp. 228–241.

[Low04] David G Lowe, *Distinctive image features from scale-invariant keypoints*, International journal of computer vision **60** (2004), no. 2, 91–110.

[LSP06] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce, *Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories*, Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on, vol. 2, IEEE, 2006, pp. 2169–2178.

[MAFM08] Michael Maire, Pablo Arbeláez, Charless Fowlkes, and Jitendra Malik, *Using contours to detect and localize junctions in natural images*, Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, IEEE, 2008, pp. 1–8.

[MCUP02] Jiri Matas, Ondrej Chum, Martin Urban, and Tomáš Pajdla, *Robust wide baseline stereo from maximally stable extremal regions*, British machine vision conference, vol. 1, 2002, pp. 384–393.

[ME07] Tomasz Malisiewicz and Alexei A Efros, *Improving spatial support for objects via multiple segmentations.*

[MFM04] David R Martin, Charless C Fowlkes, and Jitendra Malik, *Learning to detect natural image boundaries using local brightness, color, and texture cues*, Pattern Analysis and Machine Intelligence, IEEE Transactions on **26** (2004), no. 5, 530–549.

[MFTM01] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik, *A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics*, Computer Vision,

2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on, vol. 2, IEEE, 2001, pp. 416–423.

[MS04] Krystian Mikolajczyk and Cordelia Schmid, *Scale & affine invariant interest point detectors*, International journal of computer vision **60** (2004), no. 1, 63–86.

[MS05] ———, *A performance evaluation of local descriptors*, Pattern Analysis and Machine Intelligence, IEEE Transactions on **27** (2005), no. 10, 1615–1630.

[Ram07] Deva Ramanan, *Using segmentation to verify object hypotheses*, Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on, IEEE, 2007, pp. 1–8.

[RVG$^+$07] Andrew Rabinovich, Andrea Vedaldi, Carolina Galleguillos, Eric Wiewiora, and Serge Belongie, *Objects in context*, Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on, IEEE, 2007, pp. 1–8.

[SOK06] Alan F. Smeaton, Paul Over, and Wessel Kraaij, *Evaluation campaigns and trecvid*, MIR '06: Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval (New York, NY, USA), ACM Press, 2006, pp. 321–330.

[SWRC06] Jamie Shotton, John Winn, Carsten Rother, and Antonio Criminisi, *Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation*, Computer Vision–ECCV 2006, Springer, 2006, pp. 1–15.

[SZ03] Josef Sivic and Andrew Zisserman, *Video google: A text retrieval approach to object matching in videos*, Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on, IEEE, 2003, pp. 1470–1477.

[SZ06] ———, *Video google: Efficient visual search of videos*, Toward Category-Level Object Recognition, Springer, 2006, pp. 127–144.

[TM08] Tinne Tuytelaars and Krystian Mikolajczyk, *Local invariant feature detectors: a survey*, Foundations and Trends® in Computer Graphics and Vision **3** (2008), no. 3, 177–280.

[vdSUGS11] Koen EA van de Sande, Jasper RR Uijlings, Theo Gevers, and Arnold WM Smeulders, *Segmentation as selective search for object recognition*, Computer Vision (ICCV), 2011 IEEE International Conference on, IEEE, 2011, pp. 1879–1886.

[VJ01] Paul Viola and Michael Jones, *Rapid object detection using a boosted cascade of simple features*, Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on, vol. 1, IEEE, 2001, pp. I–511.

[VJ04] Paul Viola and Michael J Jones, *Robust real-time face detection*, International journal of computer vision **57** (2004), no. 2, 137–154.

[WBW$^+$10] Bo Wang, Xiang Bai, Xinggang Wang, Wenyu Liu, and Zhuowen Tu, *Object recognition using junctions*, Computer Vision–ECCV 2010, Springer, 2010, pp. 15–28.

[WKIS09] Zhong Wu, Qifa Ke, Michael Isard, and Jian Sun, *Bundling features for large scale partial-duplicate web image search*, Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, IEEE, 2009, pp. 25–32.

[ZCYF10] Long Zhu, Yuanhao Chen, Alan Yuille, and William Freeman, *Latent hierarchical structural learning for object detection*, Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, IEEE, 2010, pp. 1062–1069.

[ZHH$^+$10] Shiliang Zhang, Qingming Huang, Gang Hua, Shuqiang Jiang, Wen Gao, and Qi Tian, *Building contextual visual vocabulary for large-scale image applications*, Proceedings of the international conference on Multimedia, ACM, 2010, pp. 501–510.

[ZMLS07] Jianguo Zhang, Marcin Marszałek, Svetlana Lazebnik, and Cordelia Schmid, *Local features and kernels for classification of texture and object categories: A comprehensive study*, International journal of computer vision **73** (2007), no. 2, 213–238.

[ZTH$^+$09] Shiliang Zhang, Qi Tian, Gang Hua, Qingming Huang, and Shipeng Li, *Descriptive visual words and visual phrases for image applications*, Proceedings of the 17th ACM international conference on Multimedia, ACM, 2009, pp. 75–84.