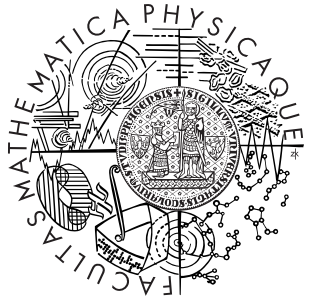
Charles University in Prague
Faculty of Mathematics and Physics

DIPLOMA THESIS



Andrej Mikulík

Methods for precise local affine frame constructions on MSERs

Department of Software Engineering

Supervisor: Doc. Dr. Ing. Jiří Matas
Study program: Informatics

2009

On this place I would like to thank to my supervisor Jiří Matas for the time spent at the consultations, for relevant comments, and ideas. My thanks go to Ondřej Chum, Michal Perďoch and Štepán Obdžálek, who provided me with valuable inputs and ideas and for many correction they did in the course of the work an in the thesis itself.

Prohlašuji, že jsem svou diplomovou práci napsal samostatně a výhradně s použitím citovaných pramenů. Souhlasím se zapůjčováním práce.

In Prague 11th December 2009 Andrej Mikulík

Contents

Title: Methods for precise local affine frame constructions on MSERs
Author: Andrej Mikulík
Department: Department of Software Engineering
Supervisor: Doc. Dr. Ing. Jiří Matas
Supervisor's e-mail address: matas@cmp.felk.cvut.cz

Abstract: Feature detection and matching is a fundamental problem in many applications in computer vision. We propose a novel approach that improves repeatability and precision of Local Affine Frames (LAFs) constructed on discretized contours detected by Maximally Stable Extremal Regions (MSERs) detector. Proposed method reconstructs a discretized contour of extremal region by taking into account the intensity function in local neighborhood of the contour points. Additionally we propose a new method for detection of local curvature extrema, based on the refined contour. The extensive experimental evaluation on publicly available datasets showed higher number of correspondences and higher inlier ratio in more than 80% of the image pairs. Since the procesing time of the contour refinement is negligible, there is no reason not to include the proposed algorithms as a standard extension of MSER detector.

Keywords: MSER, LAF, Local affine frame, feature detector, contour refinement

Název práce: Metódy pro přesnou konstrukci lokálních afinních rámců na MSER
Autor: Andrej Mikulík
Katedra (ústav): Katedra softwarového inženýrství
Vedoucí diplomové práce: Doc. Dr. Ing. Jiří Matas
e-mail vedoucího: matas@cmp.felk.cvut.cz

Abstrakt: Detekce příznaků a hledání korespondencí je jeden ze základních problémů v mnohých aplikacích počítačového vidění. V této práci navrhujeme novou metodu pro zlepšení opakovatelnosti a přesnosti lokálních afinních rámců (LAFs) konstruovaných na diskretizované kontuře maximálně stabilních extremálních regionů (MSER). Navrhovaná metoda rekonstruuje diskretizovanou konturu extremálního regionu na základě intenzitní funkce v blízkém okolí bodů kontury. Dále představujeme novou metodu pro detekci lokálních extrému křivosti na rekonstruované kontuře. Navrhované metody jsou implementovány a vyhodnoceny na veřejně dostupných databázích. Rozsáhlé experimenty ukazují vyšší počet správných korespondencí a jejich lepší poměr k tentativním korespondencím ve více jako 80% párech obrázků. Jelikož čas zpracování je vůči detekci regiónů zanedbatelný, není důvod nezahrnout navrhovaný algoritmus jako standardní rozšíření MSER detektoru.

Klíčová slova: MSER, LAF, Lokálne afinní rámce, feature detektor

Chapter 1

Introduction

1.1 Problem Formulation

Feature detection and matching is one of the fundamental problems in computer vision. It is a key ingredient to a wide range of applications including object recognition [FPZ03, SZ03], wide-baseline matching [TVG00, MCUP02, MOC02], 3D reconstruction [SSS06, SZ02] (Figure 1.3), mosaicing [BL03] (Figure 1.2), and tracking [DB06].

Consider the two images of in Figure 1.1. The same object will never appear identical in images from different viewpoints. There is a number of aspects that have impact on the final appearance. Position and rotation of the camera related to the object, scene configuration, camera parameters, lighting conditions, and so on.

A human brain is easily capable to recognize the same bus, to find correspondences in two or more images. However, in computer recognition systems this is a non-trivial task. Naive approaches such as simple correlations are insufficient.

One way is to use a representation, which is invariant, or at least to a large degree robust to all possible variations of the appearance, but still it has to be discriminative enough to distinguish between two different objects. In

Figure 1.1: Two different views of the same object. Note the different viewpoint, background (also visible trough the windows), different lighting conditions, different tone of color (specular reflections in the left image), etc.

past eight years significant research have been conducted to solve matching problem.

Nowadays approaches with the best results in matching problems follow the same pattern:

1. detection of local features
2. selection of measurement regions
3. description
4. matching
5. spatial verification

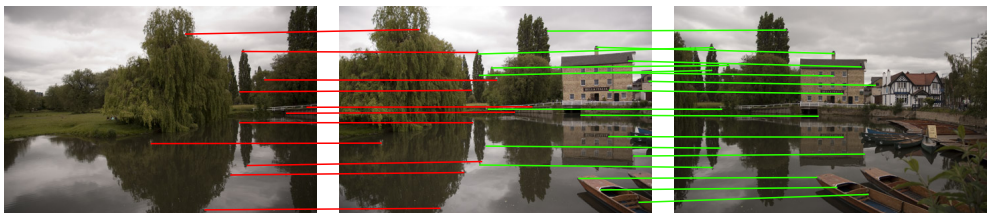
Feature detection refers to a method that aims at finding regions in the image with, in some sense, interesting property. The output of the feature detector are interest points, which have form of ellipses, blobs, or curves with high stability and reproducibility.

State of the art of features detectors includes Hessian-affine [MS02] and Harris-affine [MS02, SZ02] detectors, maximally stable extremal regions detector (MSER) [MCUP02], edge-based region detector (EBR) [TVG99], detector based on intensity extrema [TVG00], and detector of 'salient regions' [KZB04]. The performance between different detectors is usually measured in terms of repeatability and stability of detected features under various geometric and photometric transformations.

This thesis is focused on the MSER detector. MSER achieved the highest score among many tests, proving it to be a reliable region detector [MTS+05]. Especially, in viewpoint change or lighting change tests, it outperforms other five competitive detectors. In the scale change test, MSER detector comes in second following Hessian-affine detector. The only area that this type of detection is not suitable for are blurred scenes, to which the MSER detector was the most sensitive. However, according to work of Forssén and Lowe [FL07], a simple multi-resolution extension of the MSER detector improves the results also in blurred scenes.

While other detectors return elliptical regions, MSER detect contiguous regions of arbitrary shape. This allows to extract additional points on regions and its contour, and construct the so called Local Affine Frames (LAFs) [Obd07]. The LAFs are constructed by exploiting multiple affine-covariant procedures that take the detected regions as input. Assuming locally planar approximation of object shape, any image measurement expressed in LAF coordinates is viewpoint-invariant. Appearance of the objects is thus represented by local patches with shapes and locations given by the object-defined affine coordinate system. The need for further transformation of image measurements to obtain invariant description, such as rotational or differential invariants, is eliminated.

Main property of detected feature is repeatability. Patches repeatably detectable in multiple images do not necessarily provide good discriminative power. Commonly, an affine covariant constructions are used to obtain so-called measurement region [MCUP02]. A measurement region typically in-



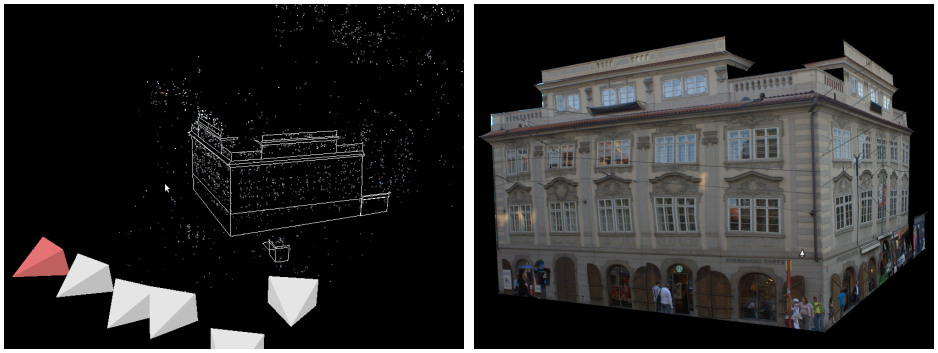
1. Detect feature points, find correspondences between images.

2. Estimate geometric transformations, put all images into one frame.

3. Estimate photometric transformations, blend.

Figure 1.2: Mosaicing – an example of an application that requires finding correspondences between images.

1. Detect feature points, find correspondences between images.



2. Estimate epipolar geometry, put all images into one scene.
3. Connect faces, create a 3D model.

Figure 1.3: 3D reconstruction. Courtesy of Lukáš Mach [Mac09].

cludes some neighborhood of the detected region and is used for the feature description.

Descriptors either compute a transformation invariant description directly or by pre-normalization of the patch to a canonical form. The state-of-the-art descriptor which is most commonly used is SIFT by Lowe [Low99]. This 128 dimensional descriptor (applied to affine normalized patches) gave the best matching results in an exhaustive evaluation of different descriptors computed on scale and affine invariant regions [MS03, MS05]. Discrete cosine transform (DCT) is other descriptor suggested by Obdržálek [Obd07] as fast and memory efficient choice for describing LAFs. Pure shape descriptor is proposed by Chum and Matas [CM06]. These will be described in Section 2.3.

Finally, feature matching is provided by examining distances in descriptor space. Once tentative correspondences are established, they are spatially verified. This is done by fitting a appropriate geometric model. The RANSAC [FB81] algorithm is used to achieve a robust estimation of model parameters [HZ03].

1.2 Goals of the Thesis

In this thesis, our goal is to find out how the post-processing of the MSER detector can be improved and how to extract better distinguished points from MSER to create LAFs.

There are several ways how the MSER detector could be improved. Some of the approaches will be described in Chapter 2. In this thesis, we will look closer on the MSER contour. We assume that even simple local processing of the MSER contour can lead to improvement in stability and repeatability of the region.

The output of the MSER algorithm are sets of connected image pixels, creating regions and holes at some intensity level – threshold, which locally maximizes stability of the region. The contours of these regions are aliased because of discretization effects. In real scenes, the edges are very often smooth with only few sharp corners. The effect of discretization can be seen on the image pair in Figure 1.2, where the detected MSER is compared with the same MSER, but in under-sampled image. In this example, we can see that discretization affect smaller regions more severely.

If we want to have better approximation of the real regions, we need to use some post-processing method to smooth the curve (MSER contour) to reduce artifacts of image rasterization. After this, we will be able to find the important points of the contour in sub-pixel accuracy with higher stability and repeatability. One approach how to suppress discretization effect was proposed in work of Obdržálek [Obd07], which shows that such post-processing can improve the results significantly.

The proposed approach should improve repeatability and stability of the distinguished points, which in consequence should improve extraction of the same points in scenes from largely different viewpoints and in different illumination conditions. This will be measured in terms of repeatability and

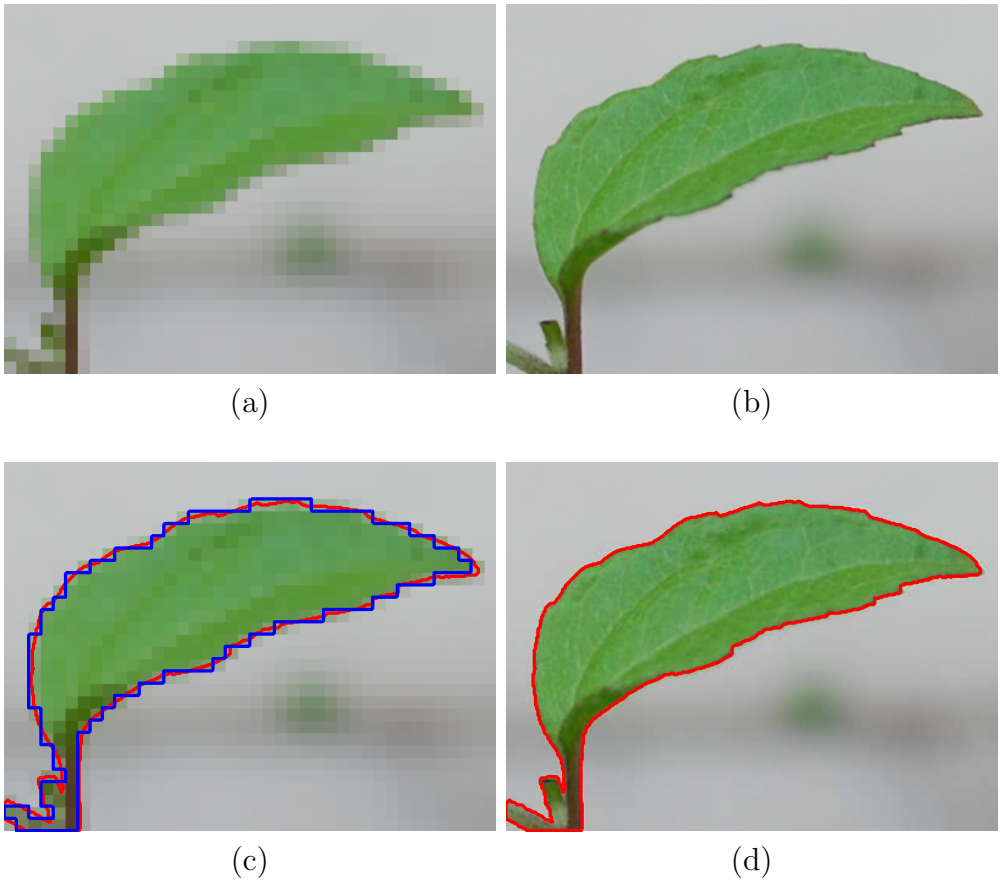


Figure 1.4: Discretization effects on a MSER region. (a) Input image for the MSER detector. (b) The same image, but with 10 times higher resolution used as ground truth. (c) Detected MSER (blue line) on input image in comparison to the desired MSER contour. (d) Detected MSER on hi-resolution image.

precision, and demonstrated on wide-baseline matching problem and object recognition.

1.3 Thesis Contribution

- A new type of contour reconstruction based on the image intensity function was proposed. This allows to extract primitives on contour with subpixel accuracy.
- Exploiting the reconstructed contour a novel approach for detecting local curvature extrama was introduced.
- An extensive experimental evaluation conducted on two publicly available datasets (ZuBuD and Mikolajczyk's) proved that the novel approach leads to more precise and more repetitive LAF constructions and more stable SIFT descriptor.
- The proposed methods were implemented and included in source repository of Center for Machine Perception at Czech Technical University in Prague, the inventor of MSER.
- There is no trade-off between the accuracy and the running-time, because the processing time of the proposed contour refinement is negligible compared to region detection. The proposed algorithm improves the state-of-the-art detector and should become a standard extension to it.

1.4 Structure of the Thesis

The thesis is structured as follows.

- **Chapter 2** gives an overview of the state-of-the-art detectors, measurement region selection and descriptors. A special attention is paid to MSER detector [MCUP02] and LAFs constructions [Obd07].
- **Chapter 3** proposes enhancements to the base MSER detector — reconstruction of region contour based on an image intensity function.
- **Chapter 4** introduce a novel method for detection local curvature extrema on refined MSER contour.
- **Chapter 5** lays out how the experiments were made and shows the results.
- **Chapter 6** discusses the achievements.

Chapter 2

Overview and Related Work

This chapter takes a closer look at the state-of-the-art of affine covariant region detectors, measurement region selection and descriptors. A special attention is paid to MSER detector [MCUP02] and LAFs constructions [Obd07].

2.1 Feature Detectors

As mentioned in Chapter 1, the feature detectors have many practical applications and they have been well described in computer vision literature in recent years.

In this place, the most important properties of each feature detector are discussed, which is followed by a brief description of individual detectors. The detailed comparison with experimental results of selected detectors can be found in Mikolajczyk et al. [MTS+05].

Desirable properties of regions detected by feature detectors are:

- High repeatability (same regions are detected in different images of same scene) (Figure 2.2)
  - Invariant to geometric transformations (change of viewpoint) (Figure 2.1)
  - Invariant to illumination changes (distinguishable under different conditions)
  - Well localized in their spatial neighborhood (*i.e.* precision)
- Discriminative neighborhood
- Robust to occlusion (regions are local)

Features are modeled as projections of small, locally planar surface patches of the 3D scene. Such projections are well approximated by an affine transformations, hence invariance to affine transformations commonly required.

The shape of the detected regions can vary depending on the detector. The most detectors return sets of ellipses – Harris-affine, Hessian-affine, and Intensity-Extrema Based Region Detector (IBR). Output of the Edge-Based

Figure 2.1: Four ellipses as an example of the output of Hessian-affine detector. Despite change of viewpoint and significant change in scale, features covering the same surface are detected.

Region Detector (EBR) is the set of parallelograms and the Maximally Stable Extrema Region Detector (MSER) returns the set of arbitrary shaped boundaries of the regions.

If ellipses representation is required, any region can be transformed to elliptical in an affine covariant manner.

Depending on the input parameters and scene character, detectors usually extract from a few hundred to several thousands of features.

2.1.1 Affine covariant blob detectors

This class of detectors looks for affine covariant blob features in the image. They are based on scale-space properties of second order derivative operators (Hessian and Laplace) [Lin94, Lin09]. Lindeberg observed that scale-space maxima of Hessian and Laplace operators are preserved for given feature in images of different scale. Blob detectors Hessian-Laplace, Harris-Laplace, and Difference of Gaussians (DoG) [MS04, Low99] use this observations to detect rotationally and scale invariant points. To extend invariance of detected features Baumberg [Bau00, LG97] proposed a shape adaptation procedure. There are couple of blob detectors based on this procedures. Two best performing according to Mikolajczyk et al. [MTS$^+$05] are Hessian-Affine and Harris-Affine.

The Hessian-Affine detector selects maxima of determinant of Hessian matrix at multiple levels of scale-space pyramid.

$$H(\mathbf{x}) = \begin{pmatrix} L_{xx}(\mathbf{x}) & L_{xy}(\mathbf{x}) \\ L_{xy}(\mathbf{x}) & L_{yy}(\mathbf{x}) \end{pmatrix} \tag{2.1}$$

where $L_{xx}(\mathbf{x})$ is second partial derivative in the a direction x and $L_{xy}(\mathbf{x})$ is mixed partial second derivative in the x and y directions. First, rotationally invariant interest points are located at each scale as extrema of determinant of Hessian matrix.

$$\det(H) = \sigma_I^2(L_{xx}L_{yy}(\mathbf{x}) - L_{xy}^2(\mathbf{x}))$$

The scale of each extremum is then selected as a scale maximum of Laplacian of Gaussians (LoG):

$$\text{trace}(H) = \sigma_I(L_{xx} + L_{yy})$$

As discussed in Mikolajczyk et al. [MS05], choosing points that maximize the determinant of the Hessian penalizes longer structures that have small signal changes (second derivatives) in a single direction. Finally, the affine shape adaptation procedure as proposed by Baumberg [Bau00] is used to obtain affine covariant interest points. Example of the output from Hessian-affine detector is shown in Figure 2.1.

The Harris-Affine detector based on the properties of an autocorrelation matrix of image gradients:

$$A(\mathbf{x}) = \sum_{x,y} w(x,y) \begin{pmatrix} I_x^2(\mathbf{x}) & I_xI_y(\mathbf{x}) \\ I_xI_y(\mathbf{x}) & I_y^2(\mathbf{x}) \end{pmatrix} \tag{2.2}$$

where I_x and I_y are first order derivatives of intensity function in x and y directions and w is weighting function. The weighting function can be either uniform (intensity is measured precisely in local window), or better Gaussian (Equation 2.3) to achieve better robustness.

$$w(x,y) = g(x,y,\sigma) = \frac{1}{2\pi\sigma} e^{\left(-\frac{x^2+y^2}{2\sigma}\right)} \tag{2.3}$$

Harris and Stephens [HS88] observed that the autocorrelation matrix centered at well localized interest points has two high and positive eigenvalues, *i.e.* gradient changes significantly in two orthogonal directions. They proposed to select these points by computing maxima of the following response function:

$$R = \det(A) - \alpha\text{trace}^2(A) = \lambda_1\lambda_2 - \alpha(\lambda_1 + \lambda_2)^2$$

The Harris-Affine detector finds the maxima of Harris measure at each scale and selects scale similarly to Hessian-Affine detector by finding maxima of Laplacian of Gaussians over scales. In second step, the shape of the elliptical region is determined by the second-moment matrix of the intensity gradient [Bau00].

2.1.2 An Edge-Based Region Detector (EBR)

Edges are typically the most stable and most descriptive features of a large domain of objects. This is also the reason why a human brain has developed the capability of recognizing objects from a set of objects contours. The EBR detector [TVG99, TVG04] is designed on this base too. Edges are stable over a range of viewpoints, illumination, and scale change.

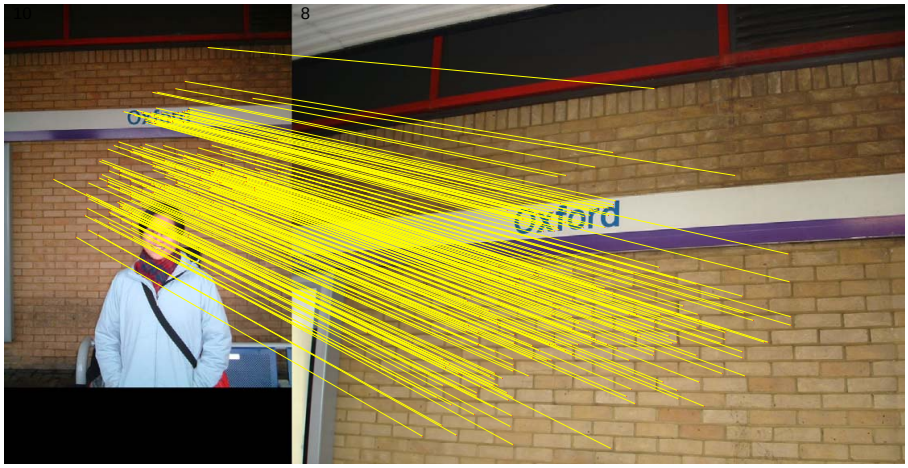


Figure 2.2: Repeated features detected with the Hessian-affine detector. For clarity, only every other correspondence is shown.

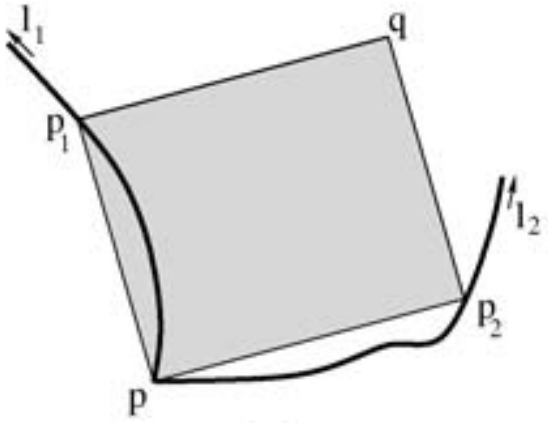


Figure 2.3: The edge-based region detector starts from an Harris corner p. Exploits the edges detected with Canny detector and sets points p_1 and p_2. p, p_2, and p_2 defines parallelogram. Figure taken from [TVG04].

The detected region is a parallelogram, one corner being the corner point p two other corners p_1 and p_2 are located on the detected edges l_1 and l_2 respective. To avoid a 2D search space for locations of p_1 and p_2, identical distance to p along the edges is required.

In the first phase, the corner points and two nearby-located edges are extracted. The Harris corner detector is utilized for detection of the corner points — this has been already described in Section 2.1.1 — and the Canny detector [Can86] is used for the edges.

In next phase, points p_1 and p_2, equally distant from corner point, are set on these edges l_1 and l_2. The distance is chosen by maximizing value of specific function to assure covariance with geometric transformation. This function and the definition of distance are discussed in detail by Tuytelaars and Van Gool [TVG99, TVG04].

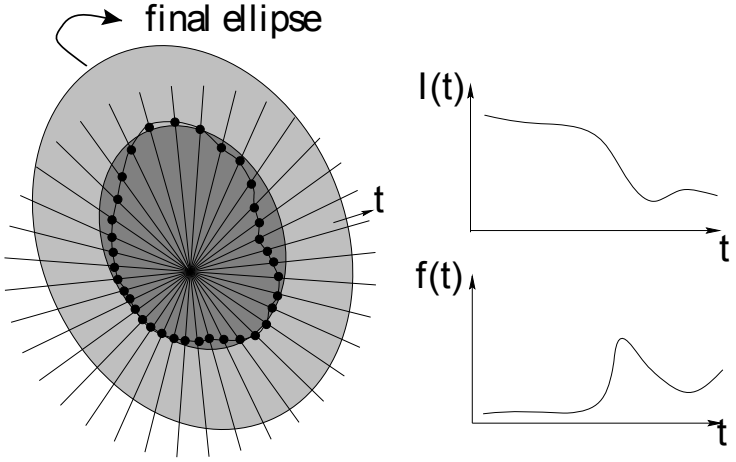


Figure 2.4: The intensity extrema-based region detector starts from an intensity extrema and examine the intensity function along rays emanating from this point. Figure taken from [TVG00].

Like the other feature detectors, also this one uses standard technique to increase the robustness to scale changes. In the first phase, the corner points with the edges are extracted from an image at multiple scales.

2.1.3 Intensity Extrema-Based Region Detector (IBR)

In the first step of the IBR detector [TVG00], intensity extrema with non-maximum suppression are localized in image. These are called anchor points. In second step, the image intensities are explored in radial way from the anchor points (Figure 2.1.3). On each ray, one point, which maximizes value of function

$$f_I(t) = \frac{\text{abs}(I(t) - I_0)}{\max\left(\frac{\int_0^t \text{abs}(I(t)-I_0)\text{d}t}{t}, d\right)} \tag{2.4}$$

is selected. In 2.4 t is the distance from the anchor point, $I(t)$ is the intensity at the distance t, I_0 is the intensity value at the extrema, and d is a small number, which is added just to avoid a division by zero.

The point on the ray selected this way is invariant under affine geometric and linear photometric transformations, given the ray. Linked points on all rays create region boundary. The directions of rays cannot be chosen in covariant manner in general. This can be alleviated by sufficiently dense sampling. For better robustness the authors suggest to replace the regions by an ellipses having the same shape moments up to the second order. This ensures covariance with affine geometric transformation.

To increase scale change robustness of detector, detection of the intensity extrema and all following steps are done at multiple scales.

IBR detector has been the first affine covariant detector used in wide-baseline matching problems. This detector has been outperform by MSER in survey [MTS$^+$05] and is rarely used in practice.

2.1.4 Maximally Stable Extremal Regions (MSERs)

In this section, we recall Maximally Stable Extremal Regions [MCUP02] and its detector. This will be done in greater detail, as the algorithm proposed in this thesis aims primary as improvement of MSER detector. In spite of the fact that the improvements can work in similar manner for all detectors, which returns region boundary as output, for now, this is the only one of this type from state-of-the-art detector set.

In the first place, we introduce formal definitions as given in the original paper on MSER by Matas et al. [MCUP02]:

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

1. $\mathcal{S}$ is totally ordered, i.e. reflexive, antisymmetric and transitive binary relation $\leq$ exists. $S = \{0, 1, \ldots, 255\}$ for the one byte representation of the pixel intensity.
2. An adjacency (neighborhood) relation $A \subset \mathcal{D} \times \mathcal{D}$ is defined. In this paper 4-neighborhoods are used, i.e. $p, q \in \mathcal{D}$ are adjacent (pAq) iff $\sum_{i=1}^{d} |p_i - q_i| \leq 1$.

Region $\mathcal{Q}$ is a contiguous subset of $\mathcal{D}$, i.e. for each $p, q \in \mathcal{Q}$ there is a sequence $p, a_1, a_2, \ldots, a_n, q$ and $pAa_1, a_iAa_{i+1}, a_nAq$.

(Outer) Region Boundary $\partial\mathcal{Q} = \{q \in \mathcal{D} \setminus \mathcal{Q} : \exists p \in \mathcal{Q} : qAp\}$, i.e. the boundary $\partial\mathcal{Q}$ of $\mathcal{Q}$ is the set of pixels being adjacent to at least one pixel of $\mathcal{Q}$ but not belonging to $\mathcal{Q}$.

Extremal Region $\mathcal{Q} \subset D$ is a region such that for all $p \in \mathcal{Q}, q \in \partial\mathcal{Q} : I(p) > I(q)$ (maximum intensity region) or $I(p) < I(q)$ (minimum intensity region).

Maximally Stable Extremal Region (MSER). Let $\mathcal{Q}_1, \ldots, \mathcal{Q}_{i-1}, \mathcal{Q}_i, \ldots$ be a sequence of nested extremal regions, i.e. $\mathcal{Q}_i \subset \mathcal{Q}_{i+1}$. Extremal region $\mathcal{Q}_{i^*}$ is maximally stable iff $q(i) = |\mathcal{Q}_{i+\Delta} \setminus \mathcal{Q}_{i-\Delta}|/|\mathcal{Q}_i|$ has a local minimum at i^* ($|.|$ denotes cardinality). $\Delta \in \mathcal{S}$ is a parameter of the method.

The concept can be explained informally as follows: We choose an intensity threshold t and divide the set of pixels into two groups; B (black) and W (white). If the pixel has intensity below t it belongs to set B else to set W.

When changing the threshold from maximum to minimum intensity, the cardinality of the two sets changes. In the first step, all pixels will be contained in B and W is empty (we see completely black image). As the threshold t is lowered, white spots corresponding to local intensity maxima start to appear and grow. At some point, regions corresponding to two local maxima would merge. Eventually all of them merge when the threshold reaches near minimum intensity and the whole image will be white (all pixels are in W

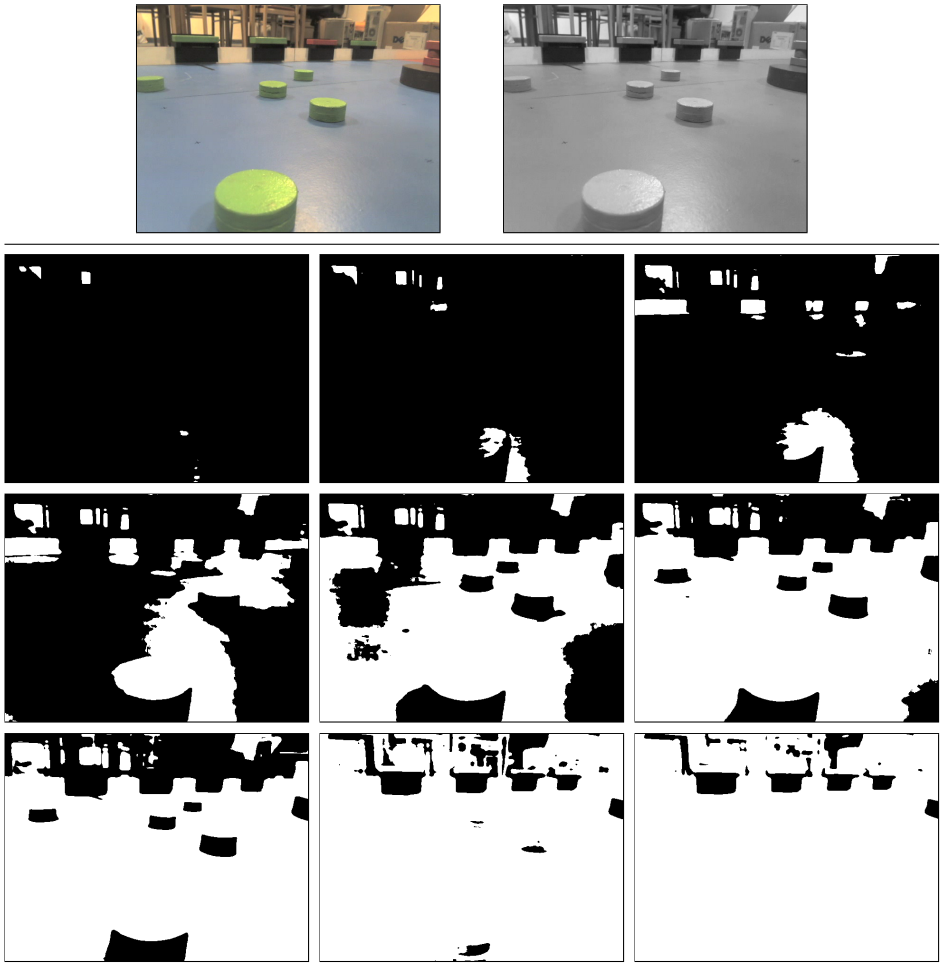


Figure 2.5: MSER evolution. Input image, image converted to grayscale and results of 9 different thresholding levels are displayed, each time for lower intensity threshold t.

and B is empty). Figure 2.5 demonstrates the evolution process with different threshold levels.

Connected components in these images are called *extremal regions* – in this case *minimal regions* or MSER$^-$. If the image is inverted and whole process repeated *maximal regions* or MSER$^+$ are obtained.

Maximally stable regions are those that have changed in size only a little across at least several intensity threshold levels. The number of levels needed and a tolerance of change are parameters of the algorithm.

According to the definition given above, totally ordered values of image pixels are required. The question is: how to design the mapping I for color image? The common choice is to convert the image into grayscale *i.e.* to order the pixels by intensity. Depending on the actual problem being solved, different orderings can lead to better results. For instance for traffic sign recognition problem, where red, blue, and white regions are the most common, the projection onto the red-blue axis of the RGB space is recommended [Obd07]. If there is no a priori knowledge about which of the orderings facilitate the region detection best, multiple orderings can be used simultaneously.

2.2 Measurement region

Measurement region is the part of the image whose appearance, after appropriate description (see Section 2.3), is used to determine local correspondences. The choice of measurement shape is arbitrary and depends on used detector and descriptor.

Interest points detected by feature detector has to be repeatable but not necessarily describing. Any function, which create a measurement region in affine covariant way out of detected region can be used. Mikolajczyk et al. [MTS$^+$05] examined the effect of rescaling the detected regions on the matching problem and show that enlargement of detected region typically leads to more discriminative power – certainly for the small regions. However the repeatability of such region can be smaller. Large scale factor can be also more detrimental, due to higher risk of occlusions or non-planarities.

Rescaling is only one option to create measurement region from detected region. If output of a detector is more complex than just an ellipse — arbitrary shaped region for instance — more complicated constructions can be made. Convex hull computed on region is another affine covariant construction, which can be useful for its robustness. These have been one of the first attempts to create an affine covariant regions out of contours [LSW90].

In work of Obdržálek [Obd07] Local Affine Frames (LAFs) are presented as set of affine covariant constructions on distinguished regions – in our case MSERs. These are described in detail in next section.

2.2.1 Local Affine Frames (LAFs)

There is several ways how can be MSER described. One way is to approximate MSER with an ellipse with the same shape moments up to the second order

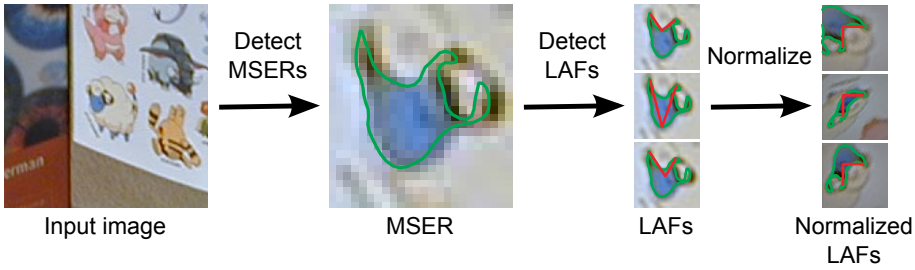


Figure 2.6: Creation of affine invariant frames with MSER detector and LAFs.

and proceed as in the case of other affine covariant detectors. Other possibility is to create convex-hull of the MSER if the convex region is required.

In this section we recall local affine frames (LAFs) [Obd07]. For further describing of the objects appearance is represented by sets of measurements defined in local coordinate systems that are established on MSER (Figure 2.6).

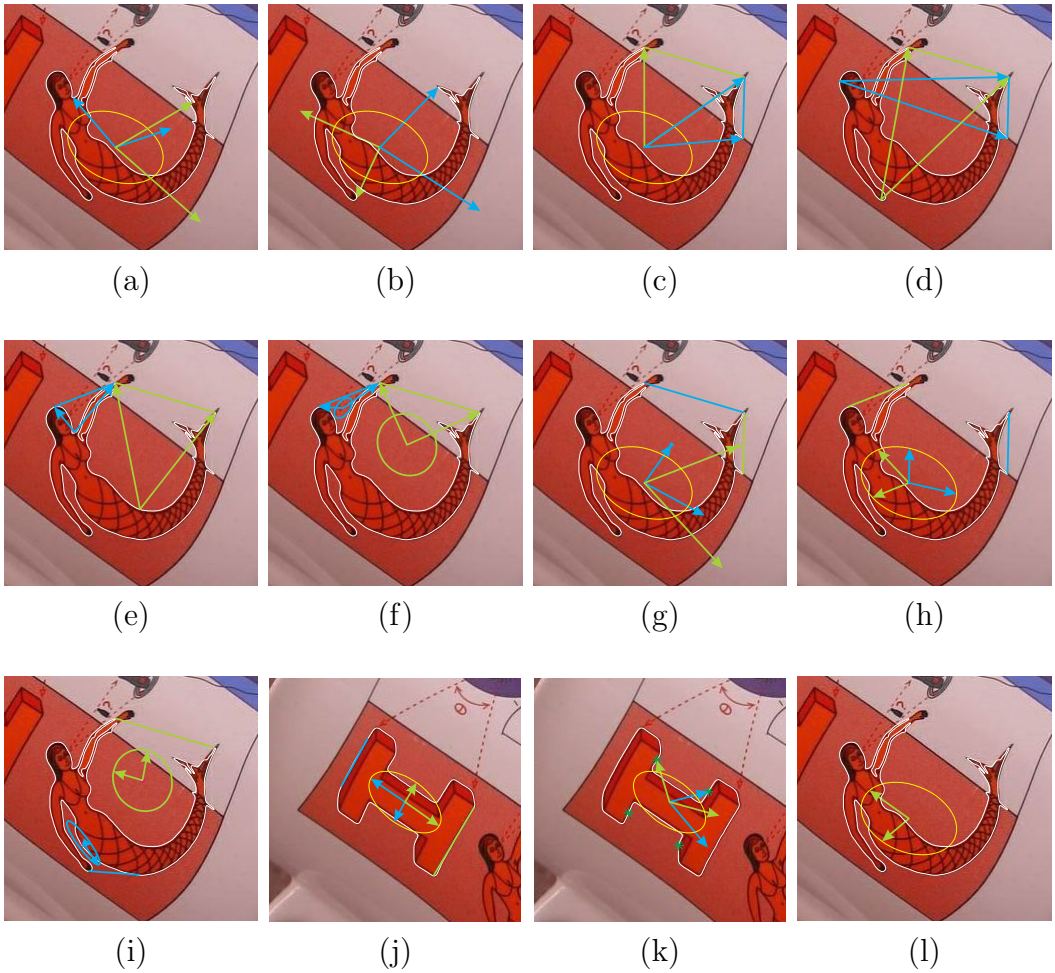
A local affine frame is constructed by combining affine-covariant primitives detected on MSER, which constrain all of its six degrees of freedom. The primitives, which can be extracted from MSER, are for instance points of extreme curvature on MSER contour, inflection points, or centers of gravity of MSER. Full list of primitives gives the Table 2.7, and theirs precise definition can be found in [Obd07].

Number of different LAFs constructions was designed. Overview with examples are shown in Figure 2.7. The images show basis vectors of the frames along with the primitives — ellipses representing covariance matrices, linear segments (e.g. bitangents), and points (e.g. curvature extrema points). Figure includes a table listing, for each of the frame type, the combination of primitives that define it. For detailed descriptions and definitions, see [Obd07].

2.3 Descriptors

Description is next step in matching problems after feature detection. A descriptor is suitable representation of a local image patch and it is associated with a appropriate similarity measure – often Euclidean distance, but others are used as well. Desired properties are:

- The descriptor has to be discriminative, *i.e.* to be able to distinguish between a large number of patches.
- The value of the similarity measure should well separate corresponding and not-corresponding regions (Figure 2.8).
- The descriptor should be insensitive to localization errors of the detector, *i.e.* to misalignment of corresponding patches.
- The descriptor should be efficiently computable.
- Evaluation of similarity measure should be efficiently computable.
- Compact representation.



Geometric primitive	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)	(j)	(k)	(l)
Centre of gravity of region (i)	×	×	×				×	×		×	×	×
Covariance matrix of region (ii)	×	×					×	×		×	×	×
Curvature minima* (iv)	×											
Curvature maxima* (iv)		×										
Tangent points of concavity (v)			×	×	×	×						
Farthest point on the contour (vi)				×								
Farthest point on the concavity (vi)					×							
Centre of gravity of concavity (i)						×	×		×			
Covariance matrix of concavity (ii)									×			
Direction of bitangent (v)								×	×			
Inflection point (vii)											×	
Direction of linear segment (viii)										×		
Third-order moments direction (ix)												×

Figure 2.7: Examples of local affine frames of different types. The table indicates which affine-covariant primitives were combined to obtain the frames. Figure credits Š. Obdržálek [Obd07].

* Affine-covariant localization of curvature extrema requires prior shape normalization by covariance matrix.

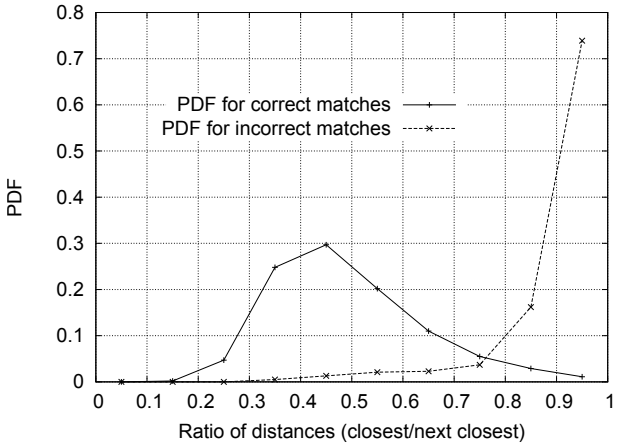


Figure 2.8: The probability that a match is correct can be determined by taking the ratio of distance from the closest neighbor to the distance of the second closest. Using a database of 40,000 keypoints, the solid line shows the PDF of this ratio for correct matches, while the dotted line is for matches that were incorrect. Image taken from [Low99].

(a) (b) (c)

Figure 2.9: (a) A mug is an example of an object from class that requires a shape descriptor. (b) An example of scene where the textural descriptor is appropriate. (c) A giraffe is an example of object class, where the information is carried in shape, texture, and color simultaneously.

There is a number of approaches for describing features. The most suitable method depends on the task and classes of objects to be described. The descriptors into two groups: shape descriptors and texture descriptors. Consider the scenes in Figure 2.9. If we want to describe a mug, descriptor cannot use textural information, as the picture on the mug can be nearly anything. Also there are object classes, in which all the information is carried in texture or color. For some classes, combination of both types of descriptors is advantageous.

Shape descriptors include descriptors that compute various edge or contour signatures [MHYS04], hierarchical or active shape models [FS07, CTCG95], and others.

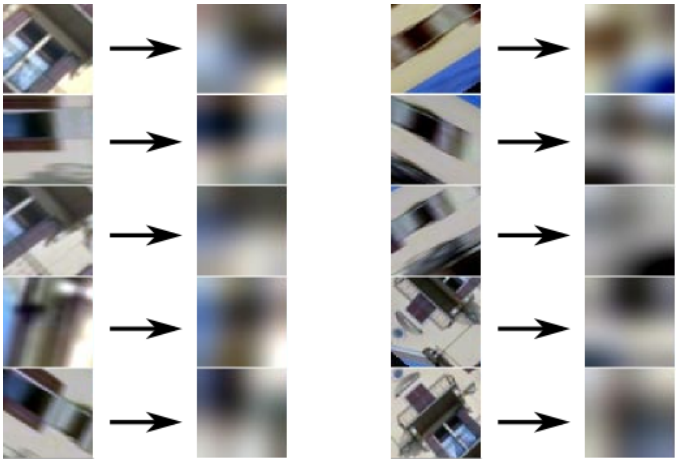


Figure 2.10: DCT representations of ten patches. 10 coefficients per color channel were used.

2.3.1 Discrete Cosine Transformation (DCT)

One possibility is to represent the local appearance by low-frequency coefficients of the discrete cosine transformation (DCT). DCT has the desirable properties of a descriptor. It is computationally eficient, fast algorithms exist that computes DCT with $\mathcal{O}(n \log n)$ time complexity. Thanks to widespread use in image and video compression (JPEG and MPEG), hardware implementations of DCT are widely available.

Definition of two-dimensional DCT for an input normalized patch I an output matrix of coefficients D is:

$$D_{p,q} = \alpha_p \alpha_q \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} I_{m,n} \cos \frac{\pi(2m+1)p}{2N} \cos \frac{\pi(2n+1)q}{2N},$$

where N is the patch resolution in pixels, $p : 0 \leq p \leq N$ and $q : 0 \leq q \leq N$ are coefficient indices, and

$$\alpha_p = \begin{cases} 1/\sqrt{N} & \text{if } p = 0 \\ \sqrt{2/N} & \text{if } 1 \leq p \leq N-1 \end{cases}, \qquad \alpha_q = \begin{cases} 1/\sqrt{N} & \text{if } q = 0 \\ \sqrt{2/N} & \text{if } 1 \leq q \leq N-1 \end{cases}.$$

Figure 2.10 shows ten different patches and their DCT representation.

Number of used coefficients is the trade of robustness to frame misalignment and discriminativity. Meanwhile the low-frequencies are less sensitive to the misalignment the higher frequencies introduce additional information to descriptor and therefor increase discriminativity. For patch with resolution 21×21 pixels, use of first 14 coefficients are suggested by Obdržálek [Obd07].

2.3.2 SIFT

The state-of-the-art of textural descriptor class is descriptor SIFT of Lowe [Low99], which have been used in several experiments in this thesis. The image patches, which are achieved after normalization of detected features, are

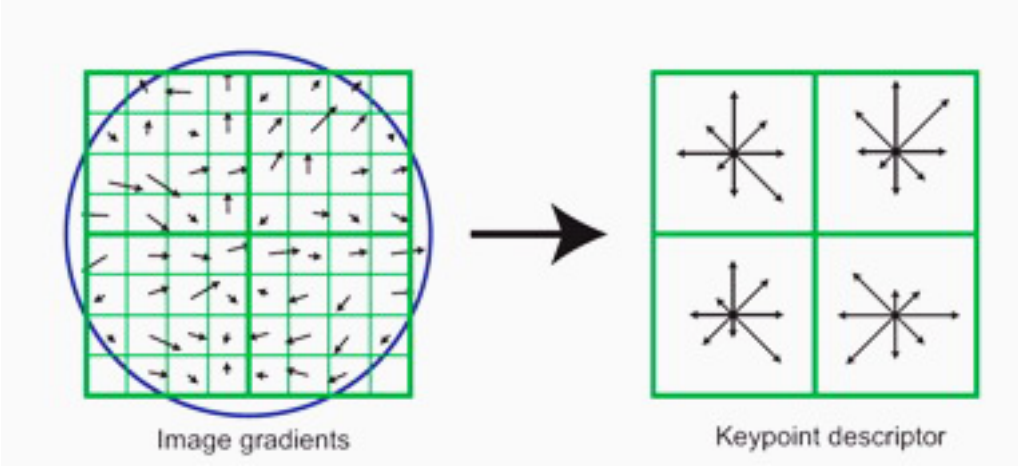


Figure 2.11: SIFT generation of a keypoint descriptor. Image taken from [Low99]

described by m dimensional descriptor. This is based on gradient histogram in patch.

Each patch is divided into a $n \times n$ grid and histograms of the direction of intensity gradients (d bins) are computed for each cell. After various weighting and normalizations the histograms are serialized into $d \times n^2$ dimensional vector (Figure 2.11). Detailed description of the algorithm can be found in [Low99]. Most common values are $n = 4$ and $d = 8$, which creates an 128 dimensional descriptor.

2.3.3 Geometric hashing with Local Affine Frames

Geometric hashing with Local Affine Frames proposed by Chum and Matas [CM06] is the representation is a collection of local affine frames that are constructed on outer boundaries of maximally stable extremal regions (MSERs) in an affine-covariant way. Each LAF is described by relative poses of other LAF in its affine neighborhood. The image is thus represented by quantities that depend only on the location of the boundaries of MSERs. Inter-image correspondences between all local affine frames are formed in a linear time by geometric hashing. Local affine frames, which are also the quantities represented in the hash table, occupy a 6D space (Figure 2.12).

This representation is insensitive to a wide range of photometric changes, robust to local occlusions and computationally efficient, but sensitive to precision of LAFs.

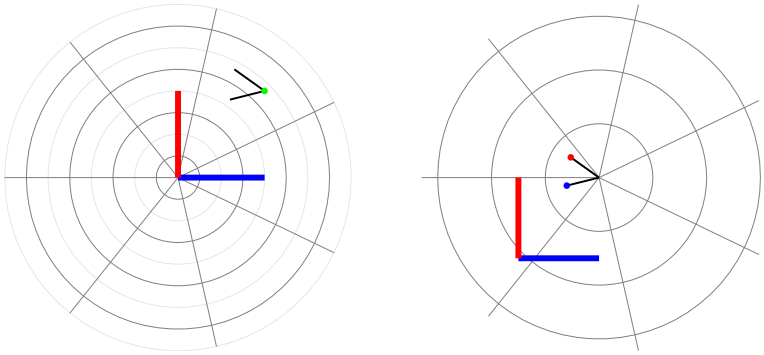


Figure 2.12: Parameterization of the 6D space of affine invariant descriptors. First two dimensions are polar coordinates of the central point of the description frame (left). The other four are polar coordinates of the remaining two points of the DF (right). Courtesy of Ondřej Chum [CM06].

Chapter 3

Discrete Contour Refinement

Regions returned by MSER detector are connected sets of image pixels. The outer contour

$$C = (\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_n), \quad \mathbf{a}_i \in \mathbb{Z}^2$$

is Jordan curve (a closed curve that does not intersect itself), where $\mathbf{a}_i = (x, y)$ and $\forall i : \left\|\mathbf{a}_i - \mathbf{a}_{(i \mod n)+1}\right\| = 1$. The contour is defined by the outer pixels and represented as a polygon — a cyclic list of (x, y) coordinates of contour vertices (the first and the last point are treated as neighbors). The coordinates of these polygons are integers — corners of the boundary pixels $\partial\mathcal{Q}$ — and edges are axis aligned. If the region contains holes, these are processed separately using the same algorithm.

As can be seen in Figure 3.1, contours are severely affected by image rasterization. Especially contours of smaller regions.

Further refinement of the region contour is important for subsequent LAF construction. Some of the constructions use primitives defined on a region contour, which is affected by the discretization effects. Reducing the discretization

(a) (b)

Figure 3.1: Examples of detected MSER regions. (a) detected regions represented by a polygon consisting of pixel boundary segments, (b) the same regions after contour smoothing with Gaussian kernel.

artifacts helps to localize such primitives as contour curvature extrema, inflection points or bitangents with higher precision. Other primitives like center of gravity or the matrix of second moments remains more or less the same.

As a reference, approach suggested by Obdržálek [Obd07] is briefly reviewed. Second, a novel approach for discretized contour refinement is proposed. Unlike [Obd07], the proposed approach uses additional information — pixels intensities from underlying image — to study local properties.

The output of contour refinement algorithm is contour C':

- $C' = (\mathbf{a'}_1, \mathbf{a'}_2, ..., \mathbf{a'}_n), \quad \mathbf{a}_i \in \mathbb{R}^2,$
- C' is Jordan curve.
- $|C'| = |C|$

Contour refinement is applicable to any discretized curve either directly or with minor changes. The novel contour refinement algorithm proposed in this chapter is a base for further improvements of the local curvature extrema extraction. This will be described in Chapter 4.

Experimental evaluation and comparison are given in Chapter 5.

3.1 Contour Smoothing – the Reference Approach

Contour pre-processing in the reference approach [MM92, Obd07] means to smooth boundary contour with Gaussian filter. This can be done with weigthed mean:

$$\mathbf{a}'_i = \frac{\sum_t \mathbf{a}_t G_{0,\sigma}(d(\mathbf{a}_i - \mathbf{a}_t))}{\sum_t G_{0,\sigma}(d(\mathbf{a}_i - \mathbf{a}_t))}, \tag{3.1}$$

where $\mathbf{a}'_i$ is position of contour vertex i, $d(\mathbf{a}_i, \mathbf{a}_t)$ is distance on contour between vertices i and t, and $G_{0,\sigma}(\bullet)$ is the Gauss PDF with standard deviation σ. Since $\forall i : \left\|\mathbf{a}_i - \mathbf{a}_{(i \mod n)+1}\right\| = 1$, the smoothing is efficiently implemented as a convolution of a normalized 1D Gaussian kernel with a sequences of x and y coordinates separately.

This smoothing method has a parameter σ — standard deviation of Gaussian distribution. The main issue of this method is to choose how strong the smoothing should be. To preserve scale invariance of regions, σ must be proportional to the square root of the region's area. In the work of Obdržálek [Obd07] this is implemented as

$$\sigma = \max\left(\frac{\sqrt{|\Omega|}}{k}, 1\right), \tag{3.2}$$

where $|\Omega|$ is size of the region (number of region pixels), and k is parameter, which controls the amount of smoothing.

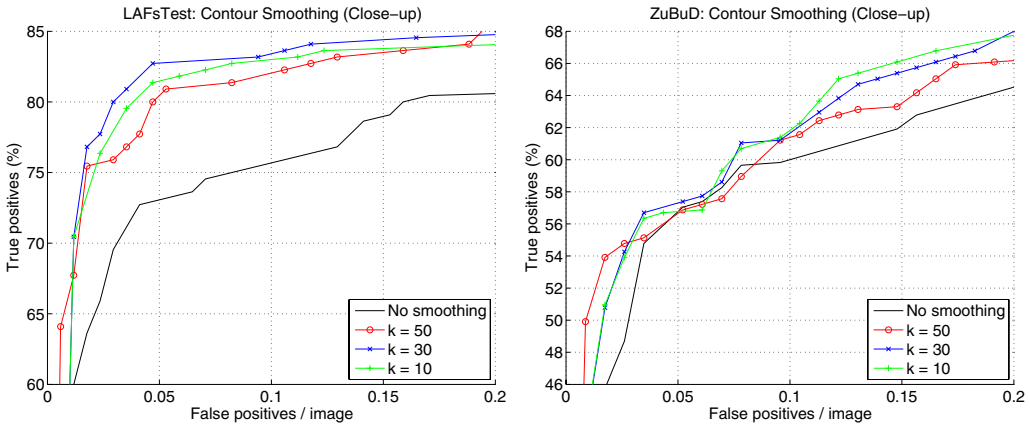


Figure 3.2: Impact of MSER boundary smoothing on recognition rate. Courtesy of Obdržálek [Obd07].

LAFsTest: Region boundary smoothing		
Configuration	Avg count of frames	Avg representation build time
No smoothing	3766	370 ms
k = 10	2514	317 ms
k = 30	2490	311 ms
k = 50	2302	282 ms

Table 3.1: LAFsTest dataset: Number of frames and time needed to build local representation according to boundary smoothing. Courtesy of Obdržálek [Obd07].

This approach is not affine-invariant but is fast and sufficiently suppresses the rasterization effect. As shown in Figure 3.2, contour smoothing has a good impact on the recognition rate between true positive and false positive detections in the object recognition problem. Furthermore, this results in faster processing in later stages since only relevant and interesting primitives are detected.

In Table 3.1 we can see the impact of smoothing on object recognition rate while varying the parameter k. It is clear that smoothing improves true positive, false positive rate but choosing parameter k is trade-off between quality of suppressing rasterization effect and preservation of local structures. If the amount of smoothing is too high, the local structures are suppressed, contour points are shifted towards the center of region and therefor the localization of contour primitives suffer. On the other hand if the smoothing is too faint, rasterization effects remains visible and false positive detections of contour primitives may appear. To avoid this trade-off a novel approach is proposed (Section 3.2).

Best performance for original approach has been observed with setting $k = 30$. Experiments have been done in object recognition problem [Obd07]. This setting have been preserved during comparison with new approach.

3.2 Contour Reconstruction with 4-point Regression

To avoid the oversmoothing caused by the method [Obd07], we propose to use the pixel intensities on and around the boundary. We show that this way, even fine details on large regions are preserved.

We refer to this procedure as contour reconstruction instead of contour smoothing. Furthermore, because this procedure is useful not only as preprocessing for extracting region primitives, but treated rather as last stage of MSER detector we would refer to it as region post-processing.

As shown in Section 3.1 σ has to be bigger for smoothing MSERs with larger areas. The problem of such approach is therefore more visible for larger regions with small structures (relative to the region area). If there is no another interesting information on such MSER contour precision resulting from this method will suffer. After smoothing out all interesting structures the MSER became nearly useless. Only small number of LAF construction (those which depends only on global information) can be build in such case.

Another problem of the contour smoothing is in the local curvature extremes — sharp corners of the contour. Under the influence of the contour smoothing, the extremal points lose their sharpness and are strongly shifted from their original position. Constructions based on these points are therefore geometrically inaccurate.

In the new approach, we want to look at structures on MSER curves independently and locally. There is no reason why the small structures should suffer from being part of larger MSER as it is in the case of convolution with Gaussian kernel. Instead of this, curve will be refined with assistance of additional information — the source image.

In the phase of the contour reconstruction, each vertex of the contour is processed independently of its neighbors. The aim of this method is to shift each vertex to a 'better' position. The resolution of contour (*i.e.* number of contour vertices) and its discrete representation is preserved. There are some other possibilities for representation of curve. We choose this approach to achieve important property — out-of-the-box usability. If the output of MSER remains unchanged after application of post-processing, it can be used immediately in all existing applications.

The MSER detector outputs regions which are darker (resp. lighter) than their neighborhood. This is separated with a threshold intensity. The boundary of the MSER region is an isophote corrupted by the discretization effects. With this knowledge and the assumption that the gradient is constant in a small area around examined vertex — four underlying pixels (Figure 3.3), which have this vertex as one of theirs corner, are used. Using intensities of these pixels, we can estimate the point lying on the isophote nearest to the original vertex. After this, we can shift original polygon vertex to new position. This way we can approximate the isophote contour with minimal changes of MSER output.

The first step is determining the gradient orientation $\nabla f_{(x_0,y_0)} = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right)$

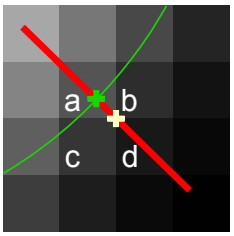


Figure 3.3: Shifting the vertices on the original contour to a new position. Yellow cross — the original vertex position, red line — a line with direction of the gradient, green line — the isophote, green cross — the new vertex position. The new position is estimated from pixels marked as a, b, c and d.

of the contour vertex, where f is image intensity function and (x_0, y_0) is vertex coordination[1].

Partial derivations in discrete world can be determine by convolving the four neighbor pixels with Roberts operator [Rob63]. This is accomplished with two 2×2 kernels

$$\begin{pmatrix} 1 & -1 \\ 1 & -1 \end{pmatrix}, \text{ and } \begin{pmatrix} 1 & 1 \\ -1 & -1 \end{pmatrix}. \tag{3.3}$$

This method is reasonably fast and good enough.

In the second step we want to shift our vertex to the nearest point on the desired region contour. As we mention before the desired contour is an isophote on the intensity level, which the MSER detector have used to obtain our contour.

The nearest point on this contour lies on a line passing through our vertex with direction given by obtained gradient (Figure 3.3). New vertex cannot be closer to another possible vertex than to original one. In that case MSER would choose another vertex. The new position of the vertex is then estimated by linear regression. The input variables are the positions of four center points of underlying pixels projected to the gradient direction, the dependent variables are the intensities of those points (Figure 5.7).

If the assumption about constant gradient is not entirely met, occasionally an error in linear regression can induce an greater shift. In this case we force the property

$$\forall i : \|\mathbf{a}_i - \mathbf{a'}_i\|_\infty < 0.5,$$

where $\|\bullet\|_\infty$ denotes L_∞ norm, which ensures Jordan contour.

On straight edges, the proposed method gives result similar to [Obd07]. However, significantly better alignment with the original curve is achieved in areas with greater curvature. Example of this we can see in Figure 3.4. The precision of the contour restoration is well demonstrated in Figure 3.5, where the original MSER contour is detected form the same image but with much better resolution.

[1]Note that coordinate system of region contours are shifted in comparison to image coordinates. For example, contour vertex $(0, 0)$ is upper left corner of $(0, 0)$ pixel.

The contour reconstruction procedure is summarized in following Algorithm:

Algorithm 1 Contour Reconstruction with 4-point Regression

Input: region contour $C = (\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_n)$, image I, intensity threshold t

Output: refined contour $C' = (\mathbf{a'}_1, \mathbf{a'}_2, ..., \mathbf{a'}_n)$

For each contour vertex $\mathbf{a} \in C$ estimate refined position $\mathbf{a}' \in C'$ in the following way:

1. $\nabla I(\mathbf{a}) = \left(\frac{\partial I(\mathbf{a})}{\partial x}, \frac{\partial I(\mathbf{a})}{\partial y}\right)$
   Approximate the gradient $\nabla I(\mathbf{a})$ of the image intensity function from four underlying pixels $p_1, \ldots, p_4$ of vertex $\mathbf{a}$ using Roberts operator [Rob63] (Figure 3.3).

2. Project center points of pixels $p_1, \ldots, p_4$ to $p'_1, \ldots p'_4$ with orthogonal projection to line passing trough vertex $\mathbf{a}$ in the gradient direction.

3. Estimate the new position $\mathbf{a}'$ of the vertex $\mathbf{a}$ for threshold t by linear regression. The regressors are $x_i = \|p'_i - \mathbf{a}\|$, and pixel intensities $I(p_i)$ are the regressands.

LAF constructions derived from the restored contour are more repetitive and more geometrically precise. Experimental comparison of these methods is evaluated on wide-baseline matching problem and described in Section 5.2.1. One of the experiments 5.6 also shows smaller distances between corresponding SIFT descriptors created from these contours than from the reference ones.

Note that the given assumption about constant gradient on the four adjacent pixels is not very limiting. Even in sharp or complicated images the area spanned by four pixels is small enough for such condition. Situation is a bit more difficult on very sharp corners when the isophote is curved within the bounds of one pixel. In such case it is impossible to extract the exact position of the peak without further assumptions, since the information is lost in aliasing. A few examples demonstration the validity of the asumption about the gradient are given in Section 5.2.3.

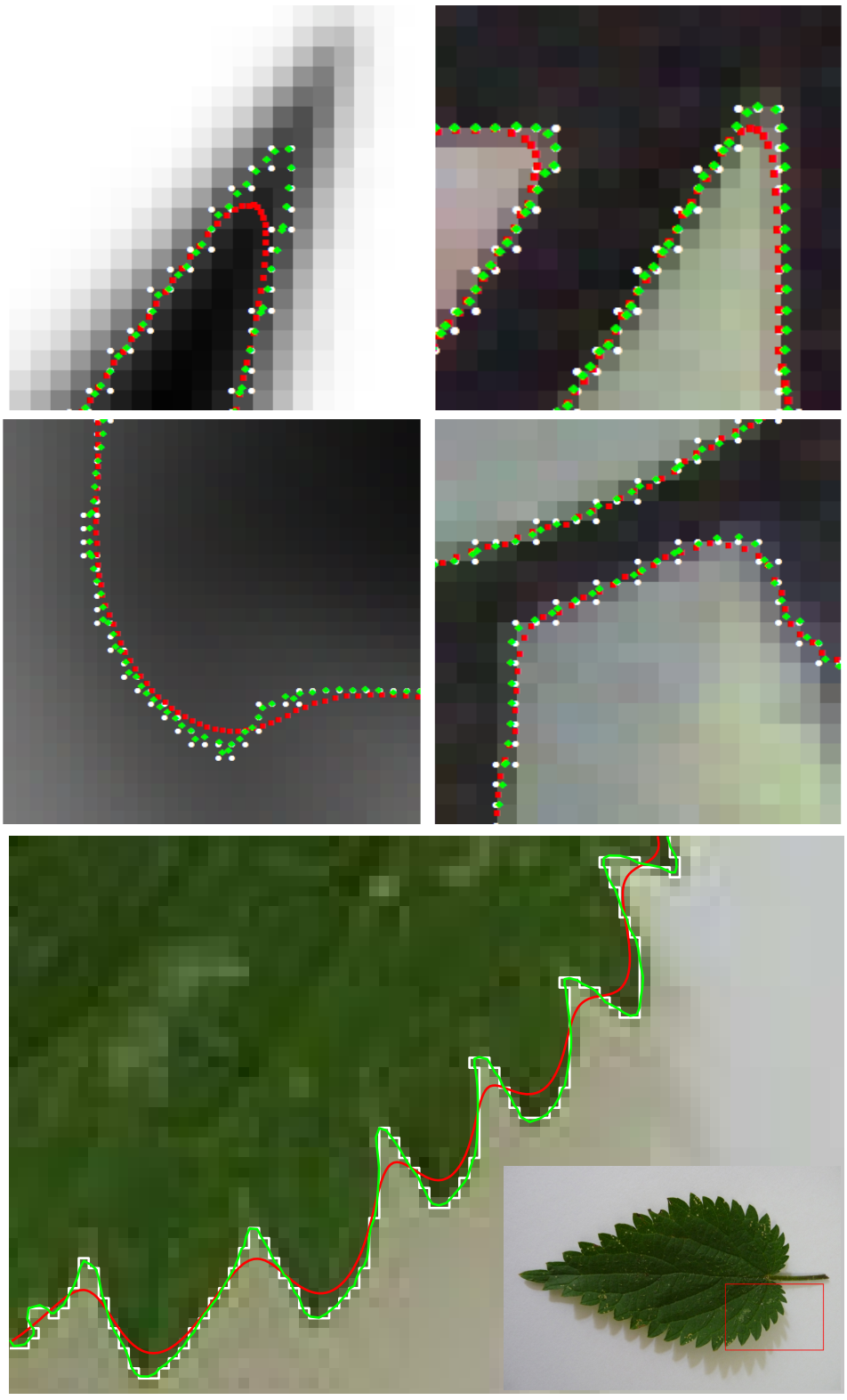


Figure 3.4: The comparison of three region contours. Yellow — the original MSER boundary, Red — smoothed with the Gaussian filter [Obd07], Green — contour reconstruction (the proposed approach)

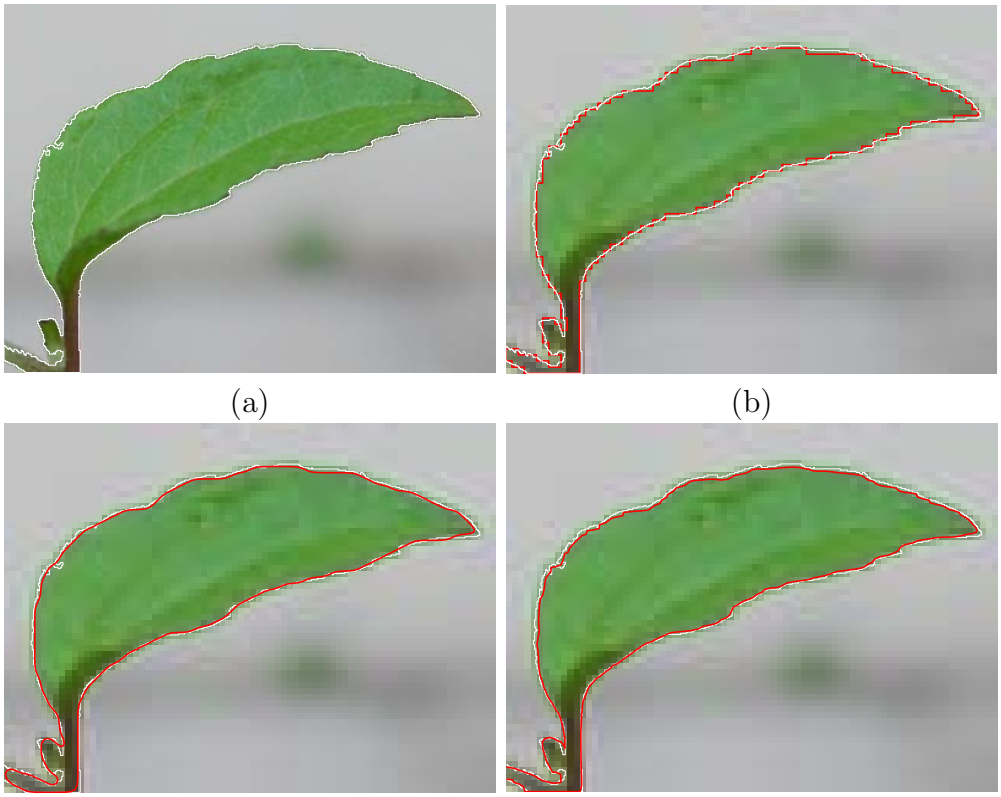


Figure 3.5: Comparison of detected contour (red) with contour detected on ten times higher resolution (white). (a) High resolution image with MSER contour without smoothing or reconstruction. (b) Red contour without post-processing. (c) Red contour after smoothing with Gaussian filter. (d) Red contour after reconstruction (new approach).

Chapter 4

LAF constructions using curvature extrema

As another contribution of this thesis, we propose a novel approach to curvature extrema detection. Unlike in [Obd07], the curvature is not computed at every point of the contour, but the extrema are detected directly.

4.1 Reference Contour Curvature Definition

Two of the LAF constructions are dependent on contour curvature. These are curvature minima and maxima used with center of gravity and covariance matrix of the region (first and second algebraic moments). Curvatures of MSER contour is not affine invariant unless the region is normalized.

In [Obd07], first of all the covariance matrix is computed. Then the region is normalized so that the covariance matrix of the resulting shape equals to the identity matrix. Shape normalization together with the position of the center of gravity of the region, fixes the affine transformation up to a rotation. If the rotation is fixed with point of curvature extrema computed on normalized contour, achieved patch will be affine-invariant.

Approximate 'curvature' is computed in [Obd07] as follows: For each vertex $\mathbf{x}$, two segments $l = \overline{\mathbf{x}l}$ and $r = \overline{\mathbf{x}r}$ of defined length a are spanned in opposite directions along the polygon boundary (see Figure 4.2). The cosine of the angle ϕ is:

$$\cos\phi = \frac{l_x r_x + l_y r_y}{|l||r|}$$

from which the curvature is estimated as:

$$\text{curvature } \kappa = s\frac{1+\cos\phi}{2}, \qquad \text{where } s = \begin{cases} 1 & \text{if } l_x r_y - l_y r_x > 0 \\ -1 & \text{otherwise} \end{cases} \tag{4.1}$$

The curvature ranges from -1 to 1, equals to 0 for straight segments, and is negative for concave and positive for convex curvatures. An example of the curvature values is shown in Figure 4.4. The important property of this definition is preserving local curvature extrema and inflection points. Histogram of curvatures on MSER contours is shown in Figure 4.1.

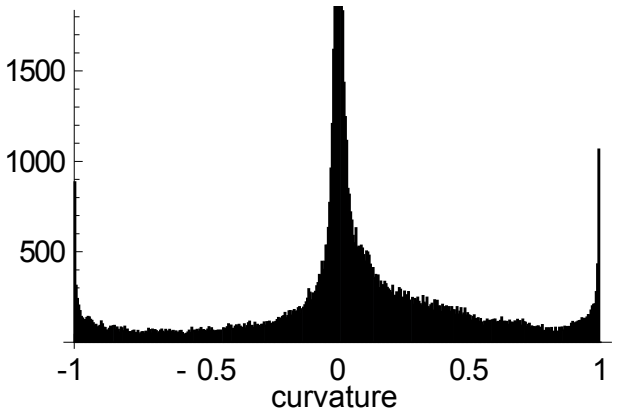
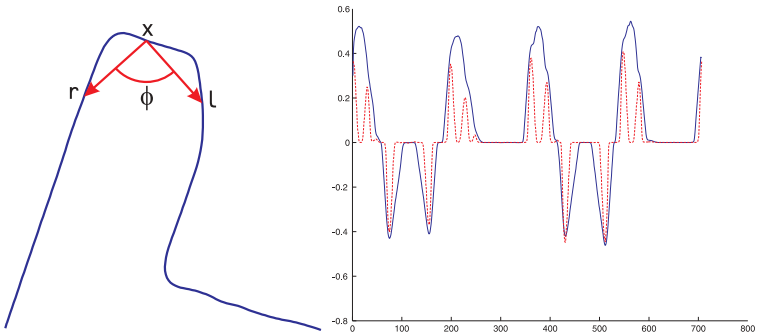


Figure 4.1: Histogram of curvatures on MSER contours.



(a) (b)

Figure 4.2: (a) Curvature estimation. (b) Curvature computed for two different values of a, $a = 0.5$ (blue line) and $a = 0.2$ (red line). Courtesy of Obdržálek [Obd07].

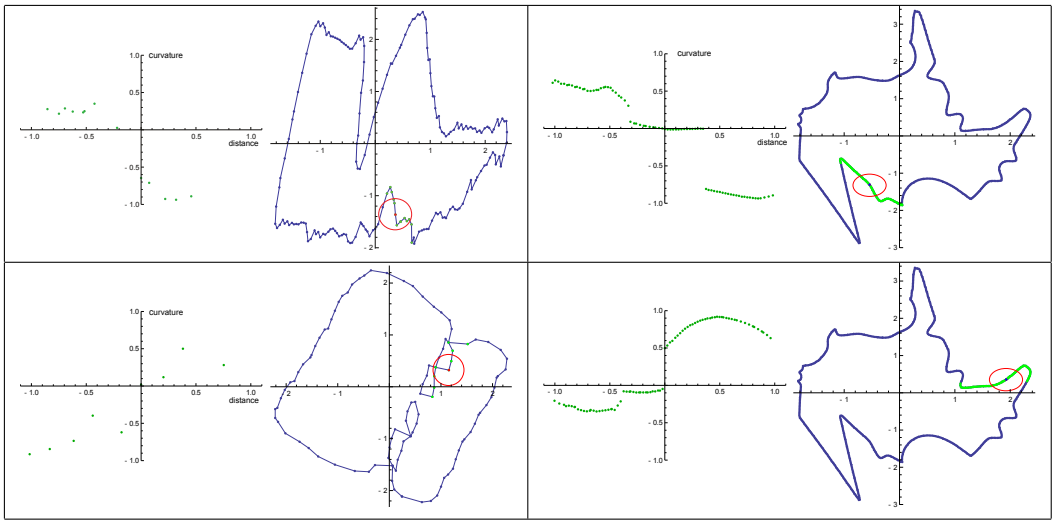


Figure 4.3: Four examples of inflection points localization. On the right side of each image is the contour of the normalized region. Red dots highlight the location of the inflection point. Green dots show the concave/convex part of the curve consecutive to the point of inflect. Curvatures of the marked points is shown on the left side of each plot. The distance from inflection point is shown on the horizontal axis.

This approach has one parameter — length of spanned arms. If the arm is shorter, more extrema are retrieved and process is easily affected by noise. On the other hand long arms ignore small structures.

The curvature definition used in [Obd07] is quite problematic. Curvature of neighboring vertices is affected by the choice of the length of the arms, which creates undesirable artifacts mainly on smaller regions with lower resolution. This is demonstrated on selected regions in Figure 4.3. There are several approaches for the curvature estimation on discretized contour.

4.2 Proposed Contour Curvature Extrema Detection

If the contour has low resolution it will be problematic to estimate curvature with any method. It is not necessary to estimate curvature for LAF constructions using curvature extrema. It is sufficient to detect the extremal points in affine-invariant way. We treat curvature extrema as originating from two types:

- the first is caused by the noise on the contour, which includes inprecission in the image aquiring, rasterization artefacts, etc. This type of curvature extrema is not repeatable over images and hence should be suppressed.
- repeatable, curvature extrema defined by the shape of the boundary.

We model the first type as having curvature Gaussian distributed from an ideal contour curvature before discretization, the second as deviations from

this distribution. The smoothing in [Obd07] provides good smoothed contour. Note that curvature extrema are shifted more from original position. However, once the curvature extrema are detected, we use their positions on contour before smoothing. Since this approach avoid the trade-off between suppressing discretization effects and preservation of local structures, we use stronger smoothing to complete suppression of discretization effects. The k in Equation 3.2 was set to 60.

In the thesis, we propose following algorithm:

Algorithm 2 Local curvature extrema detection

Input: region contour C, $P = \text{trace}(C)$

Output: curvature extrema $E \subset P$

1. $C' = \text{smooth}(C, \sigma)$
   Smooth the contour with a Gaussian kernel with σ proportional to the root of region area. (Same as in approach described in Section 3.1, but with $k = 60$.)

2. $N' = AC'$, where $A = (\text{chol}(\Sigma))^{-1}$
   Normalize the region shape by the inverse of Cholesky decomposition of th covariance matrix Σ, so that the covariance matrix of the resulting shape equals to the identity matrix. (Same as in approach described in Section 3.1.)

3. At each contour point $p'_i \in \text{trace}(N')$, the curvature κ_i is estimated from the neighboring contour points using Equations 4.1.

4. $\mu = E(K), \quad \sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\mu - \kappa_i)^2}$
   Mean value μ and standard deviation σ of estimated curvatures κ_i is computed.

5. $K = \{p'_i \in \text{trace}(N') | \quad |\kappa_i - \mu| > 3\sigma\}$
   Set of local curvature extrema candidates K is composed of points p' with curvature exceeding 3σ are candidates for local curvature extrema.

6. The local extrema $E' \subseteq K$ are selected from the candidates K by non-maxima suppression.

7. Points $e \in E \subset P$ – the pre-images of $e' \in E'$ – are the locations of the local curvature extrema on the original curve.

Detection of local curvature extrema in comparison with reference method is shown in Figure 4.4.

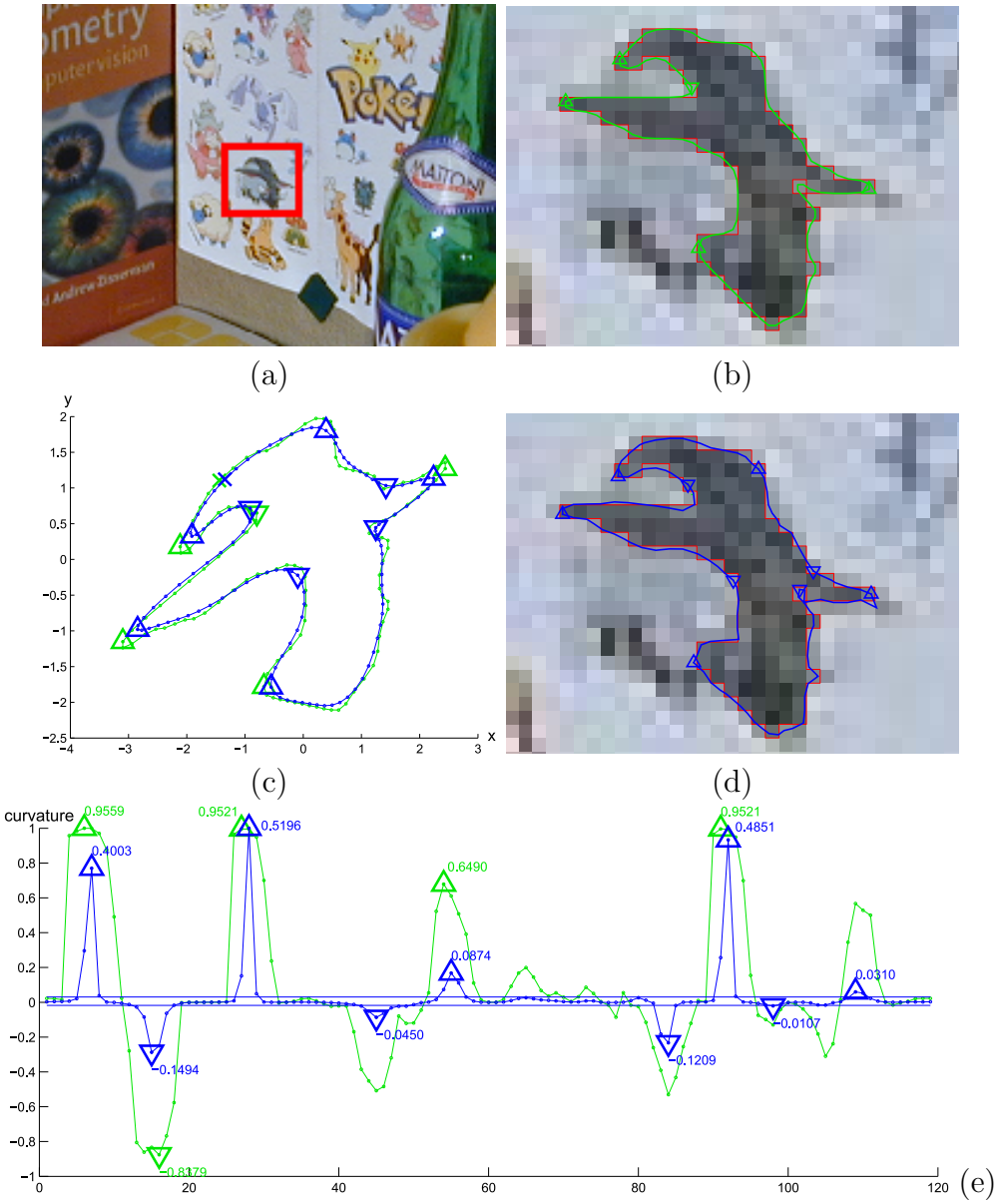


Figure 4.4: The comparison of local curvature extrema detection on MSER contour for two different post-processing method. (a) An input image. (b) and (d) detected MSER contours. Original contour is red. Contour smoothed with Gaussian kernel is green and reconstructed contour with the algorithm proposed in this thesis is blue. (c) Normalized contours. (e) Chart with vertices curvatures. Detected extrema is marked with triangle. Note better localization (narrow peaks) of extrema on blue contour. Straight blue lines around zero curvature are detected curvature outliers thresholds.

Chapter 5

Experimental Validation

In this chapter the performance of the proposed approach is evaluated on wide-baseline matching problem. In next section two datasets which have been used in the experiments are introduced. Section 5.2 then evaluates important aspects of the recognition system: repeatability of MSERs, precision of the detected features, and the number of correctly detected correspondences.

5.1 Datasets

The following two standard datasets have been used in the experimental evaluation.

5.1.1 ZuBuD Dataset

The ZuBuD dataset represents a larger, real-world problem, with images taken outdoor, with occluded objects, varying background, and mild illumination changes. ZuBuD contains images of 201 buildings in Zurich, Switzerland, and is publicly available [SSVG03]. The database consists of five photographs of each of the 201 buildings, 1005 images in total. Image resolution is 320×240 pixels. The photographs are taken from different viewpoints but under approximately constant illumination conditions.

Figure 5.1: ZuBuD dataset [SSVG03]: Examples of corresponding database images. Five images are present for every of the 201 buildings.

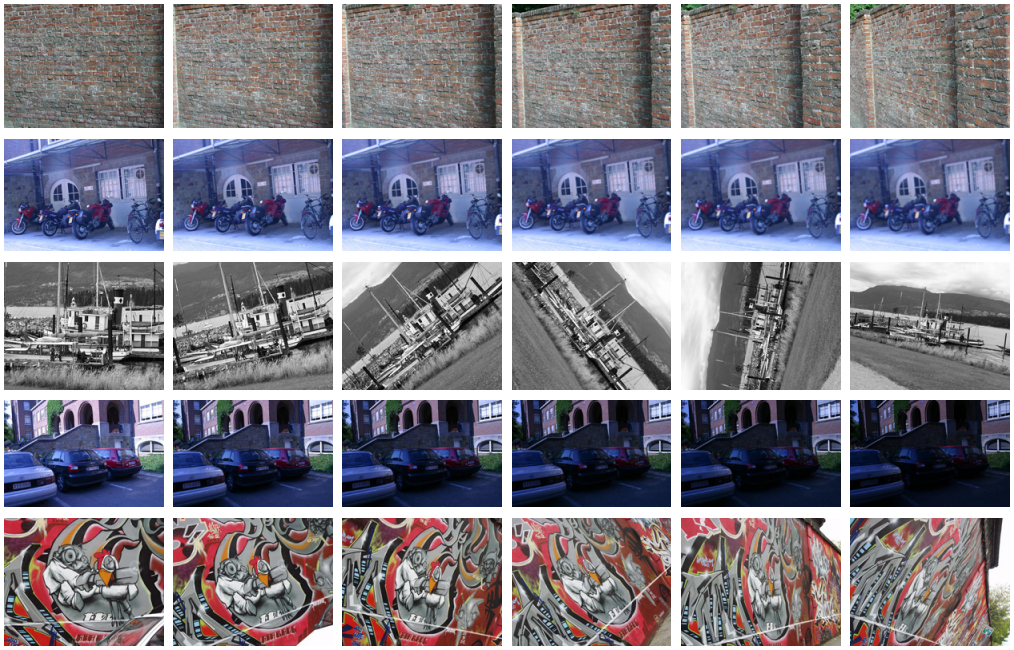


Figure 5.2: Subset of images from Mikolajczyk dataset. The homography between the images is known.

5.1.2 Mikolajczyk's Dataset

Mikolajczyk et al. [MTS+05] studied the repeatability of various affine invariant detectors. A database of images with increasing effect of different distortions — viewpoint, orientation, and scale change — is provided for the purposes of comparison. The images depict planar objects, thus homographies, which are known for the data, describe geometric transformations. Localization of detected primitives can be therefore directly compared across different views. We use the same subset of this dataset (shown in Figure 5.2) similarly as in reference work [Obd07] to compare the proposed method reference one.

5.2 Repeatability and Precision Comparison

In this section we compare three approaches to MSER boundary processing: plain – rough MSER contour without further processing, smooth – Gaussian filtering from the reference approach [Obd07], refine – the proposed method described in Chapters 3 and 4.

5.2.1 Repeatability Comparison

It has been shown [Obd07] that the elimination of discretization effects improves repeatability (see Table 3.1). However, the reference approach has also some disadvantages, which have been discussed in Chapter 3. One of them is misplacing of some contour primitives. The largest displacement is introduced to local curvature extrema (Figure 3.4), concavities and bitangents. Other

primitives such as inflection points, or linear boundary segments gain mainly by improved curvature definition. At last, there are some primitives that are virtually inaffected by boundary processing, *e.g.* center of gravity, matrix of second moments, or orientation of gradients. The affected constructions of LAFs (*i.e.* constructions (a), (b), (e), (h), (i), and (k) from Table 2.7, denoted in the following as *selected set*) were selected for the comparison.

In this section we analyze repeatability of LAF constructions before and after contour reconstruction, and compare proposed approach with reference one. The analysis is done on ZuBuD dataset and wide-baseline matching problem. For all pairs formed out of five images each building (2010 pairs in total), we run the process of finding correspondences.

Algorithm proceeds as follows:

1. MSERs are detected in both images.
2. LAFs are constructed on MSERs with (without) contour refinement, photometrically normalized and described by descriptors.
3. Tentative correspondences are established by finding mutually nearest descriptions.
4. RANSAC algorithm is used to find inliers to a global model of geometric transformation – epipolar geometry or homography. (Inliers are tentative correspondences that are consistent with the global model of geometric transformation).

Since the ground truth transformations for the pairs of images are not provided with the ZuBud dataset, the epipolar geometry with the highest number of inliers was used to evaluate the quality of the tentative correspondences in RANSAC.

Similar analysis have been done for Mikolajczyk's dataset. In this case ground truth of homography transformations provided with dataset have been used. Each of 5 scenes contain 6 images with homography transformation from first image to the others. This results to evaluation for 25 pairs.

First, a detailed table of comparison is shown for one example image pair (image 1 and image 3 from Figure 5.3), one by one primitive from *selected set* of local affine frames constructions (Table 5.1). Overall gain from the proposed approach is shown in Figure 5.4 for ZuBuD dataset and in Figure 5.5 for Mikolajczyk's dataset. We observe that proposed method increases the number of inliers for most constructions while preserving or improving the inlier ratio. Note that proposed method gives better inlier ratio in 80% of all image pairs and in 95% of all image pairs porposed method provides more inliers.

The inlier ratio is the number of inliers divided by the number of tentative correspondences. The higher inlier ratio, the faster the geometric verification by RANSAC [HZ03]. With increasing number of inliers the precision of the geometric model is increased. Higher number of inliers also guarantees higher insensivity to occlusion, since certain absolute minimal number of inliers is required to distinguish correct solution from random constellation of outliers.

image 1 from Figure 5.3, 378 MSERs						
approach:	smooth			refined		
LAF construction	desc	tc	inl	desc	tc	inl
CG+CURV_MIN	545	230	70	607	285	120
CG+CURV_MAX	825	397	218	1074	528	283
2TP+CONC	342	150	40	305	138	42
CG+BT	781	311	128	370	165	90
CCG+BT	210	90	37	186	78	35
CG+INFLECT	248	102	48	730	313	139
TOTAL	2951	1280	541	3272	1507	**709**
LAF construction	tc/desc	inl/desc	inl/tc	tc/desc	inl/desc	inl/tc
CG+CURV_MIN	42.2%	30.4%	12.8%	47.0%	42.1%	19.8%
CG+CURV_MAX	48.1%	54.9%	26.4%	49.2%	53.6%	26.4%
2TP+CONC	43.9%	26.7%	11.7%	45.2%	30.4%	13.8%
CG+BT	39.8%	41.2%	16.4%	44.6%	54.5%	24.3%
CCG+BT	42.9%	41.1%	17.6%	41.9%	44.9%	18.8%
CG+INFLECT	41.1%	47.1%	19.4%	42.9%	44.4%	19.0%
TOTAL	43.4%	42.3%	18.3%	**46.1%**	**47.0%**	**21.7%**

image 2 from Figure 5.3, 485 MSERs						
approach:	smooth			refined		
LAF construction	desc	tc	inl	desc	tc	inl
CG+CURV_MIN	753	230	70	770	285	120
CG+CURV_MAX	965	397	218	1315	528	283
2TP+CONC	449	150	40	385	138	42
CG+BT	1031	311	128	471	165	90
CCG+BT	277	90	37	249	78	35
CG+INFLECT	283	102	48	854	313	139
TOTAL	3758	1280	541	4044	1507	**709**
LAF construction	tc/desc	inl/desc	inl/tc	tc/desc	inl/desc	inl/tc
CG+CURV_MIN	30.5%	30.4%	9.3%	37.0%	42.1%	15.6%
CG+CURV_MAX	41.1%	54.9%	22.6%	40.2%	53.6%	21.5%
2TP+CONC	33.4%	26.7%	8.9%	35.8%	30.4%	10.9%
CG+BT	30.2%	41.2%	12.4%	35.0%	54.5%	19.1%
CCG+BT	32.5%	41.1%	13.4%	31.3%	44.9%	14.1%
CG+INFLECT	36.0%	47.1%	17.0%	36.7%	44.4%	16.3%
TOTAL	34.1%	42.3%	14.4%	**37.3%**	**47.0%**	**17.5%**

Table 5.1: Repeatability comparison. CG = center of gravity of region, CURV_MIN/MAX = curvature extrema, 2TP = Tangent points of concavity, CONC = Farthest point on the concavity, BT = Direction of bitangent, CCG = center of gravity of concavity, INFLECT = inflection point. desc = number of LAFs constructions, tc = number of tentative correspondences, inl = number of inliers

	ZuBuD		Mikolajczyk's	
	mean difference in number of inliers	mean difference in inlier ratio	mean difference in number of inliers	mean difference in inlier ratio
refine - plain	65.1	3.56%	117.2	4.15%
refine - smooth	62.5	2.79%	91.9	2.04%

Table 5.2: Overall gain from proposed approach evaluated on ZuBuD and Mikolajczyk's dataset.

image 1 image 2 image 3

Figure 5.3: Images from comparison in Table 5.1 and Table 5.3

5.2.2 Precision comparison

In this section, two types of experiments are conducted to comapre the presion of the geometric localization of the LAFs.

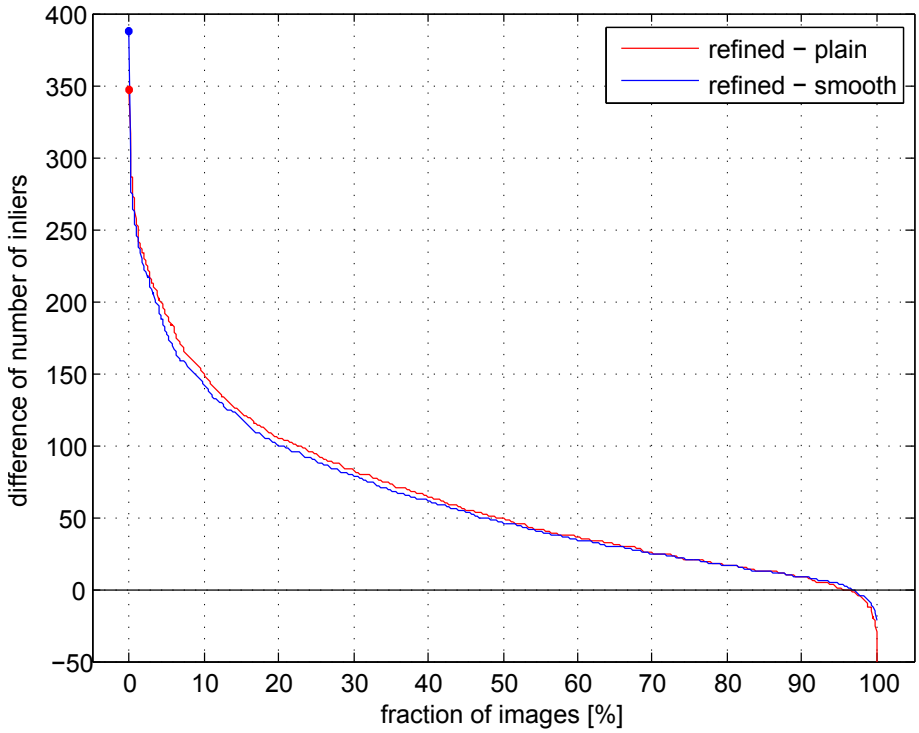
SIFT descriptor stability. Precise geometric location of the measurement region (LAF) is important for the stability of the descriptor. If the local affine frame is located over different physical surface in different images, then the descriptors are computed from different signals. Small deviations are typically handled by robustness build in the descriptor. However, higher precision of geometric location brings higher stability of the descriptor and finally results in better matching results. The precision of the geometric localization of LAFs is first compared through the comparison of stability of the SIFT descriptor.

Figure 5.6 compares SIFT distances of inliers and outliers as well as distances ratio of between first closest and second closest neighbor for different contour refinement methods. Note that distances and distances ratio of inliers are the smallest one for the method proposed in this thesis.

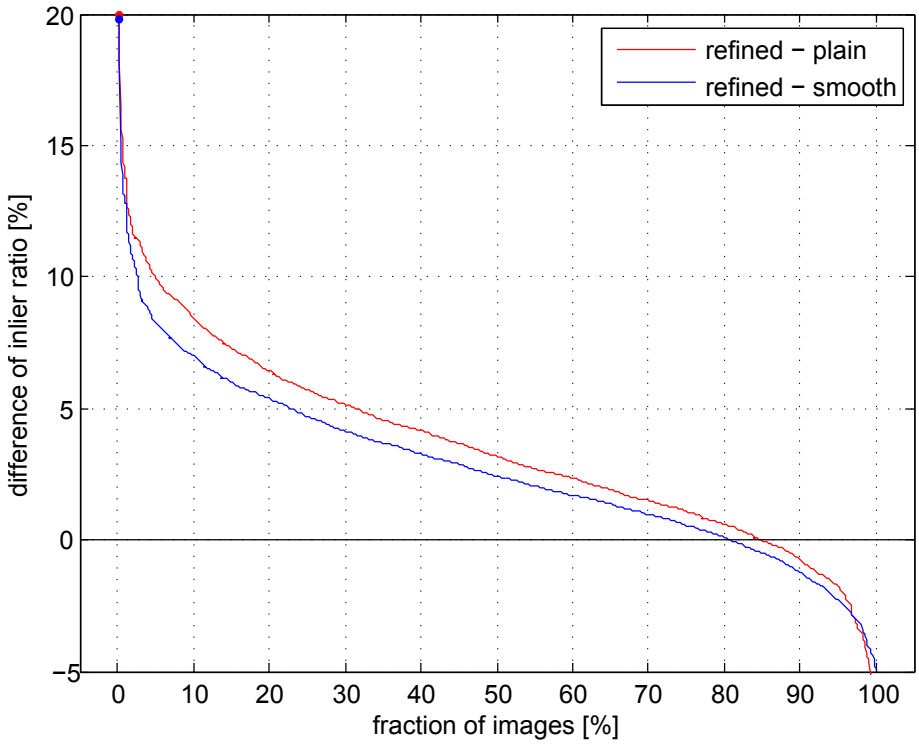
Geometry based matching. The matching results of geometric hashing with LAFs [CM06] are compared. Unlike the SIFT descriptor, the geometric hashing is not using the intensity function to compute the descriptor. The descriptor is based on mutual geometric positions of pairs of LAFs in each image. Hence, such an approach is extremely sensitive to the precision of the geometric localization.

Influence of precision of the LAF constructions are the most significant in this experiment. Table 5.3 shows results from matching of two pairs from Mikolajczyk's dataset.

The tentative correspondences obtained by geometric hashing are naturally ordered by their quality (*i.e.* probability of being a correct match) [CM06]. This ordering can be exploited in the geometric verification step using PROSAC [MC05] instead of RANSAC. To highlight the improvement in the matching results inlier ratio at 500 best tentative correspondences is also considered, see Table 5.3.

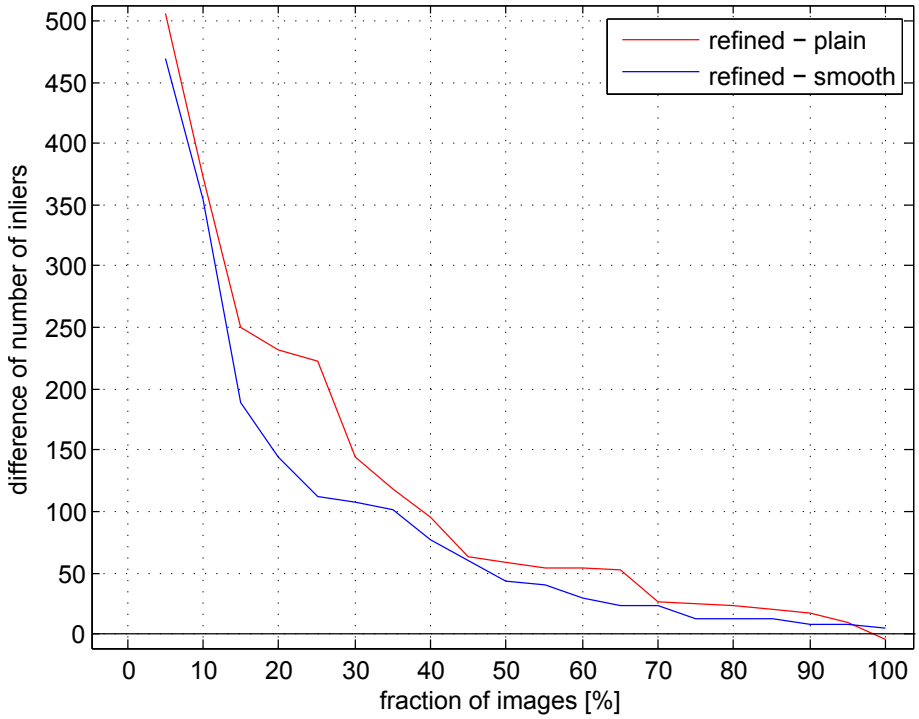


(a)

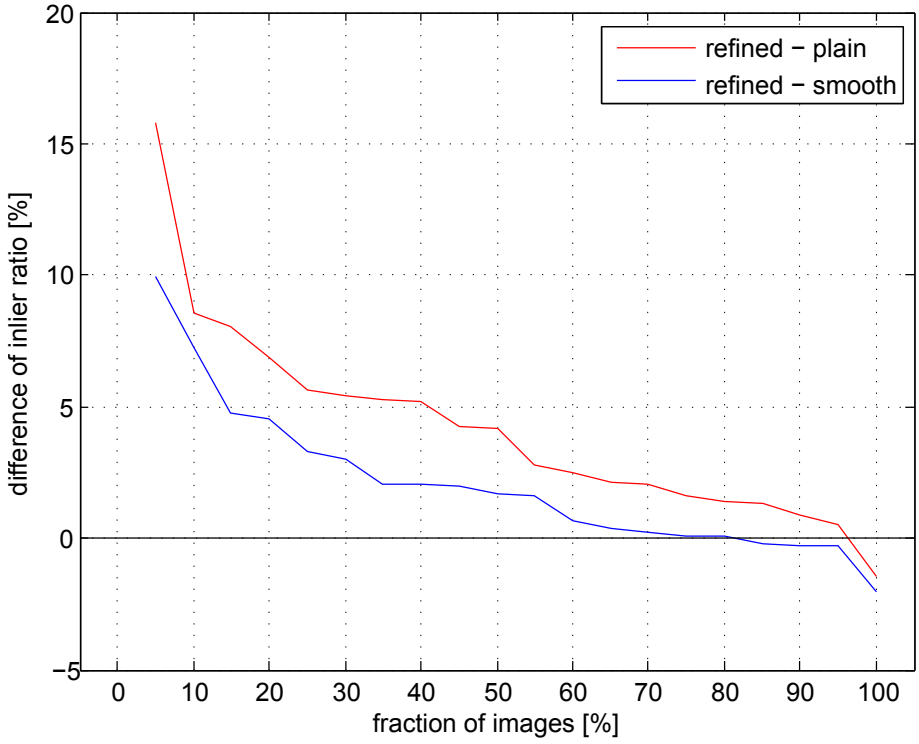


(b)

Figure 5.4: Repeatability comparison on ZuBuD dataset. (a) Difference between inlier ratios. (b) Difference between number of inliers.

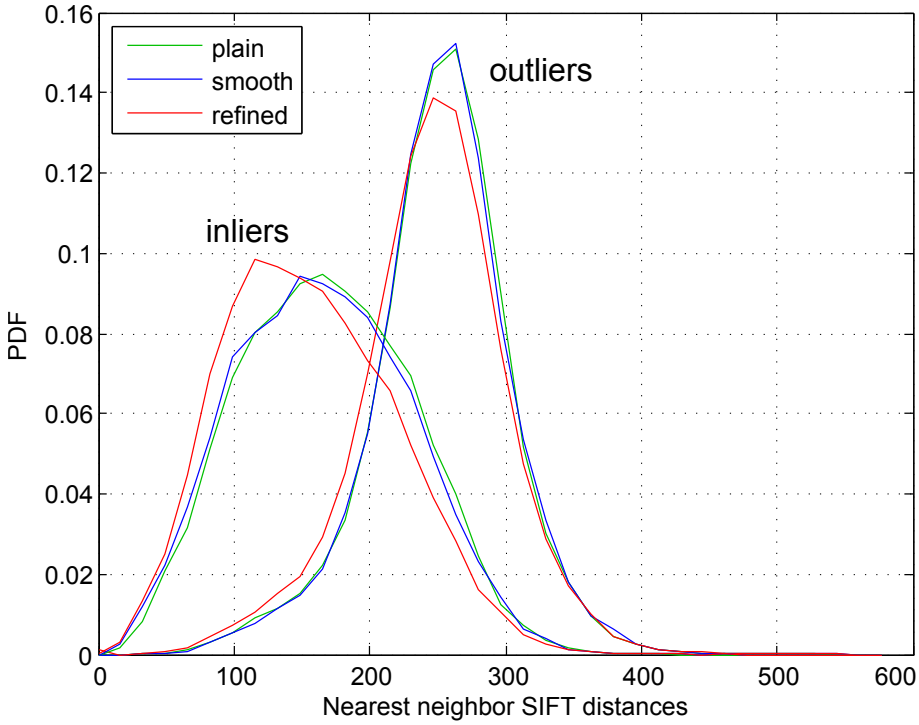


(a)

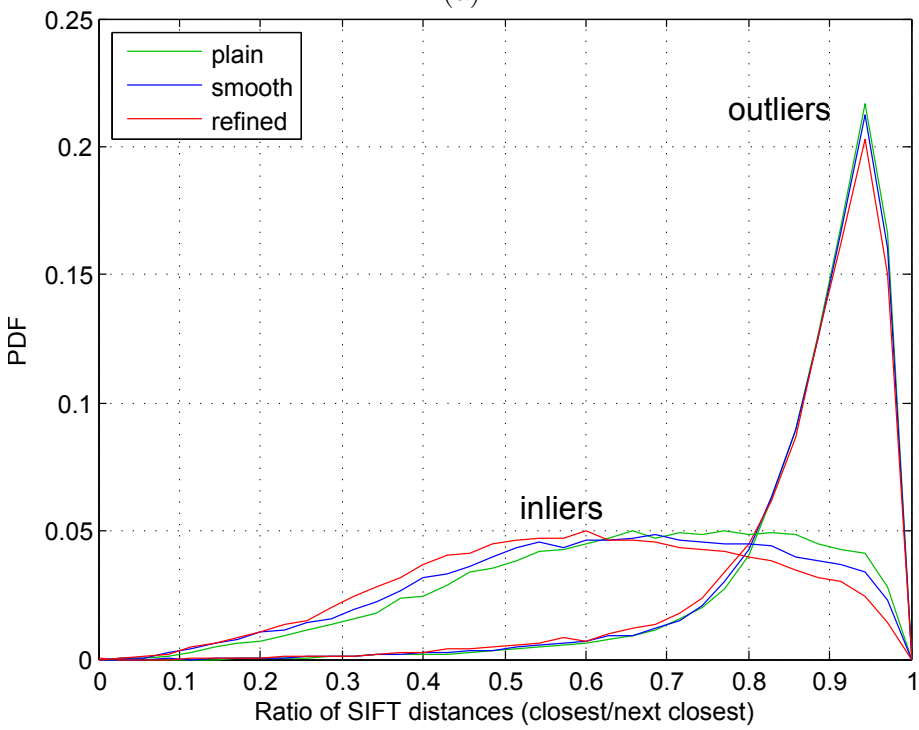


(b)

Figure 5.5: Repeatability comparison on Mikolajczyk's dataset. (a) Difference between inlier ratios. (b) Difference between number of inliers.



(a)



(b)

Figure 5.6: (a) PDF of distances for inlier and outlier pairs for different contour refinement methods. (b) PDF of distances ratio between first closest and second closest neighbor for different contour refinement methods.

image 1 ↔ image 2 from Figure 5.3					
	all			best 500 tc	
method	tc	inl	tc/inl	inl	tc/inl
plain	622	425	68.3%	418	83.6%
smooth	678	516	76.1%	458	91.6%
refine	921	**702**	**76.2%**	**486**	**97.2%**

image 1 ↔ image 3 from Figure 5.3					
	all			best 500 tc	
method	tc	inl	tc/inl	inl	tc/inl
plain	500	284	56.8%	284	56.8%
smooth	566	337	**59.5%**	337	67.4%
refine	801	**470**	58.7%	**409**	**81.8%**

Table 5.3: Matching results for geometric hashing with LAFs. tc = number of tentative correspondences, inl = number of inliers

5.2.3 Position Estimation Error

In this section, we will look closer on estimation of new positions for contour polygon vertices. As we mentioned in Section 3.2 there is one assumption about the gradient around processed vertex. We assume that gradient on the four underlying pixels is constant. This is not necessary true in general. The condition is not entirely satisfied in noisy images or sharp corners, but we can still estimate new position with the maximum likelihood (Figure 5.7). Furthermore, we can compute error of position estimation as mean square error of regression and see how good this model fits.

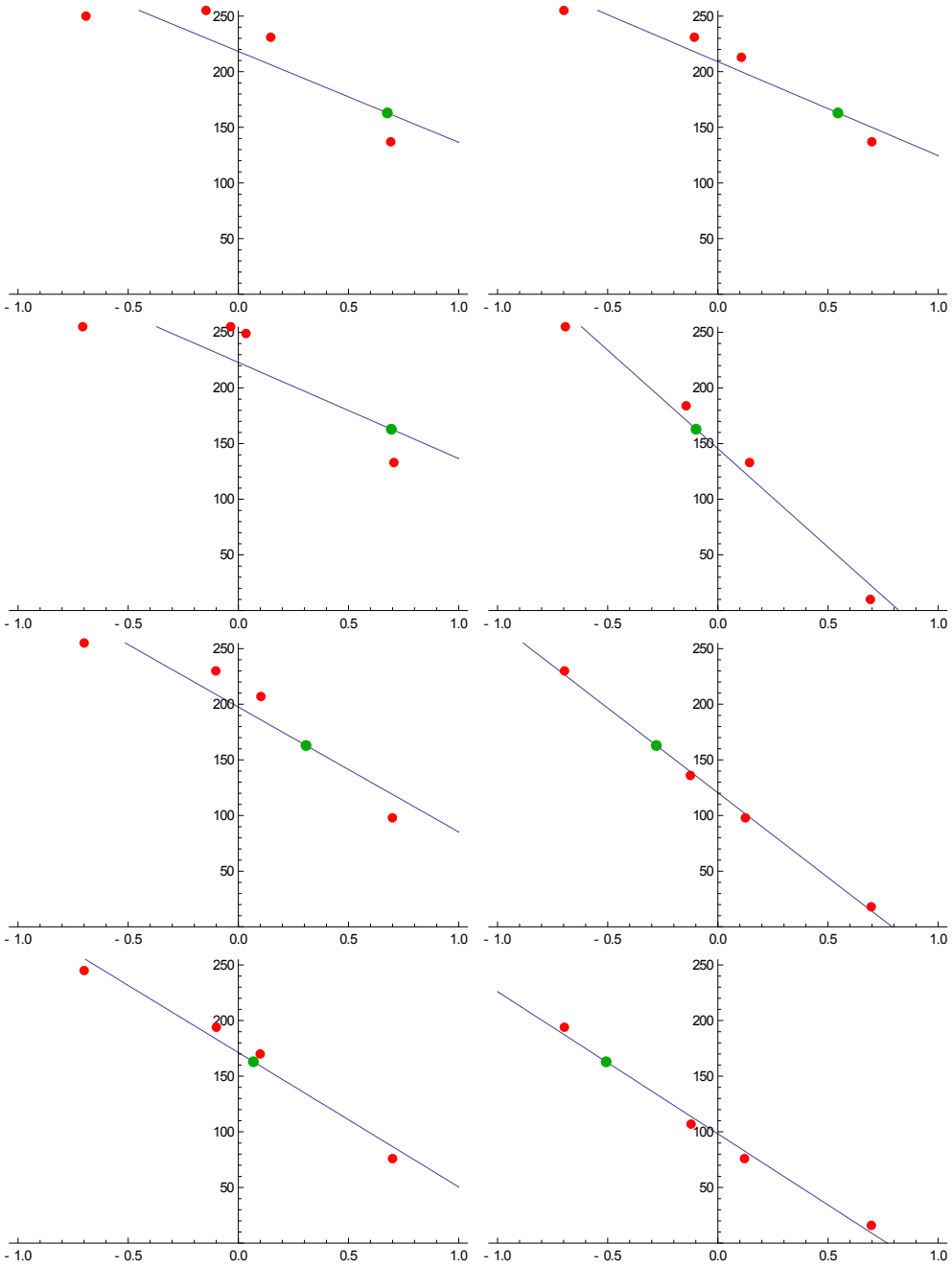


Figure 5.7: Examples of new vertex position estimation. Red dots represent centers of the underlying pixels, where the value on x-axis means distance from the original vertex position and y-axis represents the pixel intensity. Blue line is the linear model fitted to the intensity function, and green dot is the new estimate of the vertex position. In this example an isophote on intensity level 162 is considered.

Chapter 6

Conclusions

Maximally Stable Extremal Region (MSER) detector extracts a comprehensive number of image features, which are the base for many applications in computer vision. Detected regions have good properties and in many types of scenes outperforms the other affine covariant region detectors, proving MSER to be a reliable state-of-the-art region detector.

Local Affine Frames (LAFs) construct measurement regions on primitives extracted from the detected regions. A number of these primitives are defined on region contour. Since the boundary of region is defined by set of pixels, the contour is affected by rasterisation, which negatively affects the precision of created LAFs. The more precise LAF constructions are, the more stable and distinguishable are the descriptions. This leads to faster spatial verification and higher number of correspondences established between images.

Methods for suppressing rasterisation effects were studied in this thesis and a new type of contour reconstruction based on the image intensity function was proposed. This allows to extract primitives on contour with subpixel accuracy. On the base of reconstructed contour the novel approach for detecting local curvature extrama was introduced.

The methods were implemented and extensive experimental evaluation was conducted on two publicly available datasets (ZuBuD and Mikolajczyk's). Three approaches to MSER contour processing were compared. Rough MSER contour without further processing, Gaussian filtering from the reference approach, and the proposed method. The evaluation included the following experiments. To assess the repeatability, wide-baseline matching of 2010 image pairs from the ZuBuD database was conducted. The precision of the geometric localization of the detected features was tested in two complementary ways. First, the stability of the SIFT descriptor was measured. Second, the results of image matching based on geometric hash were compared. The results of all the experiments proved that the novel approach leads to more precise and more repetitive LAF constructions and more stable SIFT descriptor.

Since the processing time is insignificant to the time of region detection, there is no reason not to include the proposed algorithms as a standard extension and improvement of popular state-of-the-art MSER detector. The source codes are already included in source repository of Center for Machine Perception at Czech Technical University in Prague, the inventor of MSER.

Appendix A

Demostration software

To show the performance of proposed contour refinement method a simple demonstration software is provided on attached CD. It presents the solution of the correspondence problem in wide baseline setup. It is built on the state of the art method that uses MSER detector [MCUP02], LAFs [Obd07] and finally SIFT descriptors [Low99].

A.1 Structure of the attached CD

My modifications replace the original contour smoothing algorithm and detection of distinguished points on a contour of extremal region (extrema of curvature, inflection points...). They inherit and overwrite the original classes and methods. The source codes of my modifications are available in subdirectory `refinement/refinement/`, files `distinguishedregionsng.*, lafsng.*`. Finally, the source code of the demonstration program is in `refinement.cpp`. Demonstration software depends on a several libraries kindly provided by the Center of Machine Perception and publicly available library `Lapack`, all located in `refinement/lib/win32` directory. To build the sources on Win32 platform a Visual Studio 2008 solution is provided in the `refinement` directory. Additionally, a set of images is provided in directory `refinement/pics`.

A.2 Functionality

Attached software demonstrates the process of finding correspondences between two wide baseline images of the same scene. The image names are provided using commandline parameters `-i1 img1.ppm`, `-i2 img2.ppm`, only `.ppm` image file-format is supported.

In the first phase of the algorithm, MSER regions are detected in a both images independently. Then contours are processed by one of the three methods selected by a commandline parameter `-method`:

- 0 - no contour smoothing,
- 1 - contour smoothing proposed by [Obd07],
- 2 - proposed contour reconstruction method.

Afterwards, a set of local affine frames is computed (from the computed distinguished points), photometrically normalized and described using SIFT descriptors. Finally, a set of tentative correspondences is found using nearest neighbours in the space of SIFT descriptors. Other matching methods and parameters of the local affine frame construction can be set in `lafs.cfg` that have to be located in the binary working directory.

If requested with parameter `-tc tcfilename`, a set of tentative correspondences is output to a file in form of 3xN corresponding points in homogenous coordinates (x_1,y_1,1,x'_1, y'_1,1,x_2,y_2,1,$x'_2 \ldots$, y'_3,1) resulting in 18 numbers for each pair for each matching LAF.

Optionally a globally consistent model of epipolar geometry or homography and a set of inliers to the model is sought by RANSAC [FB81, Chu05] and inliers output into file set by `-inl inlfilename`.

Demonstration program also allows to set the parameters `-ms`, `-mm` and `-per` of the MSER detector, that controls the size of smallest detected region, stability (measured in number of intensities) and percentage of the image covered by the largest detected region. These are tuned to work well on images of approximately 1 Mpixel. Other parameters of the local affine frames construction and matching method are available in `lafs.cfg` configuration file and described in [Obd07]. It is safe to keep them untouched.

Summary of the usage

```
Running with proposed regression contour refinement.

Usage: refinement.exe [options]
    -i1 (null) [null] input image1 (ppm, pgm)
    -i2 (null) [null] input image2 (ppm, pgm)
    -tc (null) [null] output file for tentative correspondences
    -inl (null) [null] output file for inliers to the model
    -method (2) [2] contour refinement method (0 - none,
            1 - smoothing,2 - proprosed regression)
    -model (2) [2] global transformation model (0 - none,
            1 - homography, 2 - epipolar geometry)
    -ms (30) [30] minimum size of output region
    -mm (10) [10] minimum margin
    -per (0.010) [0.010] maximum relative area
    -help (0) [0] print out usage info
Dependencies:
    Option -i1 is compulsory
    Option -i2 is compulsory
Errors detected during  option parsing:
    Missing compulsory option -i1
    Missing compulsory option -i2
```

Example results using reference method

```
<CDROOT>\refinement\bin\refinement.exe -i1 ../pics/graffA.ppm
-i2 ../pics/graffB.ppm -method 1 -model 1
Running gaussian smoothing of contour.
```

```
Processing image ../pics/graffA.ppm
  Detected 369 MSER+ and 421 MSER- regions in 0.095 sec.
  Computing DRs...790 distinguished regions in 0.244 sec.
  Generating LAFs...11141 local affine frames in 5.026 sec.

Processing image ../pics/graffB.ppm
  Detected 158 MSER+ and 341 MSER- regions in 0.083 sec.
  Computing DRs...499 distinguished regions in 0.170 sec.
  Generating LAFs...6191 local affine frames in 2.786 sec.

Finding tentative correspondences... 1670 pairs in 9.270 sec.
Removing inconsistent correspondences ...
Got 730 consistent correspondences in 0.035 sec.
Running LO-RANSAC(H)
313 inliers, inlier ratio: 42.877%
```

Example results using proposed method

```
<CDROOT>\refinement\bin\refinement.exe -i1 ../pics/graffA.ppm
-i2 ../pics/graffB.ppm -method 2 -model 1
Running with proposed regression contour refinement.

Processing image ../pics/graffA.ppm
  Detected 369 MSER+ and 421 MSER- regions in 0.095 sec.
  Computing DRs...790 distinguished regions in 0.314 sec.
  Generating LAFs...10626 local affine frames in 4.752 sec.

Processing image ../pics/graffB.ppm
  Detected 158 MSER+ and 341 MSER- regions in 0.085 sec.
  Computing DRs...499 distinguished regions in 0.221 sec.
  Generating LAFs...5781 local affine frames in 2.563 sec.

Finding tentative correspondences... 1658 pairs in 7.620 sec.
Removing inconsistent correspondences ...
Got 821 consistent correspondences in 0.032 sec.
Running LO-RANSAC(H)
394 inliers, inlier ratio: 47.990%
```

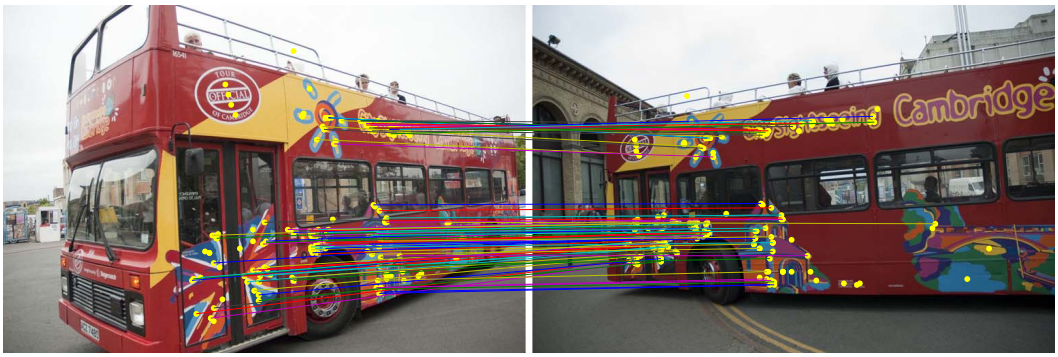
A.3 Visualization

A simple matlab script `refinement/bin/refinement.m` is provided for results visualization. Note that the script writes the output files into current directory and therefore write permission is required. Figure A.1 shows the visualization.

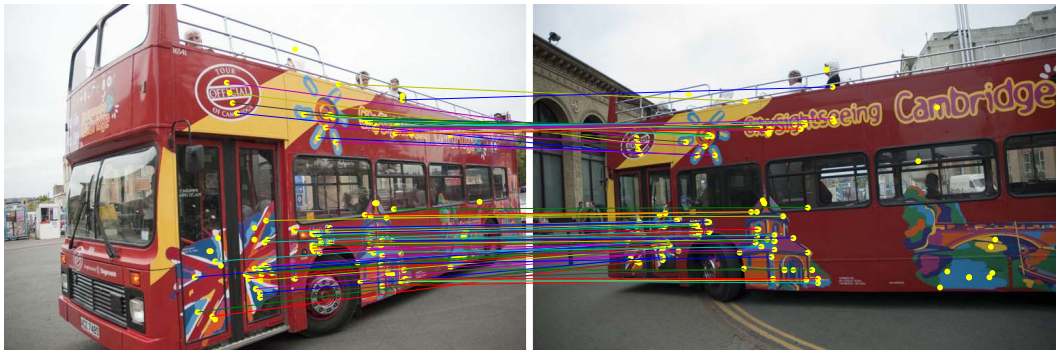
Example results of visualization script

```
Running with proposed regression contour refinement.
Processing image ..\pics\busA.ppm
  Detected 188 MSER+ and 313 MSER- regions in 0.050 sec.
  Computing DRs...501 distinguished regions in 0.132 sec.
  Generating LAFs...6167 local affine frames in 2.255 sec.

Processing image ..\pics\busB.ppm
```

```
  Detected 131 MSER+ and 239 MSER- regions in 0.048 sec.
  Computing DRs...370 distinguished regions in 0.110 sec.
  Generating LAFs...4333 local affine frames in 1.595 sec.

Finding tentative correspondences... 1063 pairs in 2.974 sec.
Removing inconsistent correspondences ...
Got 422 consistent correspondences in 0.014 sec.
Running LO-RANSAC(F)
310 inliers, inlier ratio: 73.460%

Running gaussian smoothing of contour.
Processing image ..\pics\busA.ppm
  Detected 188 MSER+ and 313 MSER- regions in 0.050 sec.
  Computing DRs...501 distinguished regions in 0.111 sec.
  Generating LAFs...6070 local affine frames in 2.210 sec.

Processing image ..\pics\busB.ppm
  Detected 131 MSER+ and 239 MSER- regions in 0.049 sec.
  Computing DRs...370 distinguished regions in 0.084 sec.
  Generating LAFs...4205 local affine frames in 1.556 sec.

Finding tentative correspondences... 929 pairs in 2.881 sec.
Removing inconsistent correspondences ...
Got 286 consistent correspondences in 0.012 sec.
Running LO-RANSAC(F)
230 inliers, inlier ratio: 80.420%
```



(a)



(b)

Figure A.1: Example output of wide-baseline matching. Yellow dots are tentative correspondences. Inliers are connected with lines. (a) The proposed method. (b) The reference method.

Bibliography

[Bau00] Adam Baumberg. Reliable feature matching across widely separated views. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 1:1774, 2000.

[BL03] Matthew Brown and David G. Lowe. Recognising panoramas. In *Proc. ICCV*, pages 1218–1225, 2003.

[Can86] John Canny. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(6):679–698, November 1986.

[Chu05] Ondřej Chum. *Two-view Geometry Estimation by Random Sample and Consensus*. Phd thesis, Center for Machine Perception, K13133 FEE Czech Technical University, Prague, Czech Republic, September 2005.

[CM06] Ondřej Chum and Jiří Matas. Geometric hashing with local affine frames. In Andrew Fitzgibbon, Camillo Taylor, and Yan LeCun, editors, *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 879–884, Los Alamitos, USA, June 2006. IEEE Computer Society.

[CTCG95] Timothy. F. Cootes, Christopher J. Taylor, David H. Cooper, and Jim Graham. Active shape models-their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, January 1995.

[DB06] Michael Donoser and Horst Bischof. Efficient maximally stable extremal region (mser) tracking. In *Computer Vision and Pattern Recognition*, volume 1, pages 553–560, 2006.

[FB81] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, June 1981.

[FL07] Per-Erik Forssén and David Lowe. Shape descriptors for maximally stable extremal regions. In *IEEE International Conference on Computer Vision*, volume CFP07198-CDR, Rio de Janeiro, Brazil, October 2007. IEEE Computer Society.

[FPZ03] Robert Fergus, Pietro Perona, and Andrew Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Proc. CVPR*, 2003.

[FS07] Pedro F. Felzenszwalb and Joshua D. Schwartz. Hierarchical matching of deformable shapes. In *CVPR*, 2007.

[HS88] Chris Harris and Mike Stephens. A combined corner and edge detector. In *Proceedings of the 4th Alvey Vision Conference*, 1988.

[HZ03] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge University, Cambridge, 2nd edition, 2003.

[KZB04] Timor Kadir, Andrew Zisserman, and Michael Brady. An affine invariant salient region detector. In *Proceedings of the 8th European Conference on Computer Vision, Prague, Czech Republic*. Springer, May 2004.

[LG97] Tony Lindeberg and Jonas Gårding. Shape-adapted smoothing in estimation of 3-d cues from affine deformations of local 2-d brightness structure. *ICV*, 1997.

[Lin94] Tony Lindeberg. Scale-space theory in computer vision, 1994.

[Lin09] Tony Lindeberg. *Encyclopedia of Computer Science and Engineering*, chapter Scale-space, pages 2495–2504. Hoboken, 2009.

[Low99] David G. Lowe. Object recognition from local scale-invariant features. *Computer Vision, IEEE International Conference on*, 2:1150–1157 vol.2, August 1999.

[LSW90] Yehezkel Landan, Jacob T. Schwartz, and Haim J. Wolfson. Affine invariant model-based object recognition. *IEEE Transactions On Robotics And Automation*, 6:578–589, 1990.

[Mac09] Lukáš Mach. Semi-automatic system for reconstruction of 3d scenes. Master's thesis, Faculty of Mathematics and Physics, Charles University in Prague, 2009.

[MC05] Jiří Matas and Ondřej Chum. Randomized ransac with sequential probability ratio test. In Songde Ma and Heung-Yeung Shum, editors, *Proc. IEEE International Conference on Computer Vision (ICCV)*, volume II, pages 1727–1732, New York, USA, October 2005. IEEE Computer Society Press.

[MCUP02] Jiří Matas, Ondřej Chum, Martin Urban, and Tomáš Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In Paul L. Rosin and David Marshall, editors, *Proceedings of the British Machine Vision Conference*, volume 1, pages 384–393, London, UK, September 2002. BMVA.

[MHYS04] Siddharth Manay, Byung-Woo Hong, Anthony J. Yezzi, and Stefano Soatto. *Integral Invariant Signatures*, pages 87–99. Springer, 2004.

[MM92] Farzin Mokhtarian and Alan K. Mackworth. A theory of multiscale, curvature-based shape representation for planar curves. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(8):789–805, 1992.

[MOC02] Jiří Matas, Štěpán Obdržálek, and Ondřej Chum. Local affine frames for wide-baseline stereo. In R. Kasturi, D. Laurendeau, and Suen C., editors, *ICPR 02: Proceedings 16th International Conference on PatternRecognition*, volume 4, pages 363–366, CA 90720-1314, Los Alamitos, US, August 2002. IEEE Computer Society.

[MS02] Krystian Mikolajczyk and Cordelia Schmid. An affine invariant interest point detector. In *European Conference on Computer Vision*, pages 128–142, 2002.

[MS03] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 257–264, 2003.

[MS04] Krystian Mikolajczyk and Cordelia Schmid. Scale and affine invariant interest point detectors. *IJC*, 60(1):63–86, 2004.

[MS05] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 27(10):1615–1630, 2005.

[MTS+05] Krystian Mikolajczyk, Tinne Tuytelaars, Cordelia Schmid, Andrew Zisserman, Jiří Matas, Frederik Schaffalitzky, Timor Kadir, and Luc Van Gool. A comparison of affine region detectors. *Int. J. Comput. Vision*, 65(1-2):43–72, 2005.

[Obd07] Štěpán Obdržálek. *Object Recognition Using Local Affine Frames*. Phd thesis, Center for Machine Perception, K13133 FEE, Czech Technical University, Prague, Czech Republic, April 2007.

[Rob63] Lawrence Roberts. Machine perception of three dimensional solids. Technical report, MIT Lincoln Lab., May 1963.

[SSS06] Noah Snavely, Steven M. Seitz, and Richard Szeliski. Photo Tourism: exploring photo collections in 3D. In *Proc. ACM SIGGRAPH*, pages 835–846, 2006.

[SSVG03] Hao Shao, Tomáš Svoboda, and Luc Van Gool. ZuBuD — Zurich Buildings Database for Image Based Recognition. Technical Report 260, Computer Vision Laboratory, Swiss Federal Institute of Technology, March 2003. http://www.vision.ee.ethz.ch/showroom/zubud.

[SZ02] Frederik Schaffalitzky and Andrew Zisserman. Multi-view matching for unordered image sets, or "How do I organize my holiday snaps?". In *Proceedings of the 7th European Conference on Computer Vision, Copenhagen, Denmark*, volume 1, pages 414–431. Springer, 2002.

[SZ03] Josef Sivic and Andrew Zisserman. Video Google: A text retrieval approach to object matching in videos. In *Proc. ICCV*, 2003.

[TVG99] Tinne Tuytelaars and Luc Van Gool. Content-based image retrieval based on local affinely invariant regions. In *In Int. Conf. on Visual Information Systems*, page 493–500, 1999.

[TVG00] Tinne Tuytelaars and Luc Van Gool. Wide baseline stereo matching based on local, affinely invariant regions. In *Proc. 11th BMVC*, 2000.

[TVG04] Tinne Tuytelaars and Luc Van Gool. Matching widely separated views based on affine invariant regions. *Int. J. Comput. Vision*, 59(1):61–85, 2004.