GEOMETRIC GROUPING OF PLANAR PATTERNS IN A PERSPECTIVE VIEW

Thesis submitted in partial fulfillment
of the requirements for the degree of

Master of Science (by Research)
in
Computer Science

by

Kiran Varanasi
200399008
kiran@research.iiit.ac.in

International Institute of Information Technology
Hyderabad, INDIA
Nov. 2006

Abstract

When geometric primitives such as curves and patterns appear in repetition over a perspective image, they offer key information for recovering the real metric structure of the scene. This happens because multiplicity is equivalent to motion - a single image of such a scene is equivalent to multiple images taken from varying camera viewpoints. Multiplicity can manifest in the image through several forms - tiling (*translational symmetry*), reflection (*bilateral symmetry*), or rotation (*point symmetry*). In all these cases, it pays dividends to group these patterns into geometrically meaningful sets, i.e, into sets of patterns which produce a uniform geometric constraint. For example, a unique vanishing point. Each of these patterns is defined in terms of some interest points and contour segments. It is conceivable that these points and contours are ill-identified, and this makes the task of geometric grouping all the more challenging. However, if only the patterns are grouped robustly, the later tasks of 3D reconstruction and the estimation of pose can be handled with high accuracy.

Symmetrical patterns are commonplace in natural and man-made environments. Especially in architectural scenes, these patterns abound in all types of variety. Several attempts have been being made by the research community to exploit this information. Success has been reported in several areas, particularly in the area of image based modeling and rendering (IBMR). In this thesis, we study the problem of geometric grouping of patterns, from the context of an interactive IBMR application. We will demonstrate how geometric grouping with minimal user interaction is useful towards improving the robustness of structure recovery. We handle the problem of geometric grouping of planar patterns in all its generality - we do not presume a known period of repetition or a known template for the patterns. The only properties which we use for grouping the patterns are the geometric constraints (such as the *vanishing line* and the *circular points*) - properties that would be estimated by the patterns themselves. Our algorithms facilitate the aggregation of geometric information from multiple sources in the image, and thus make robust rectification possible.

The principal contributions of our work are on three fronts - (1) A method for identifying interest points on a set of poorly identified and badly fragmented image contours (2) A greedy optimization approach for computing point correspondences through preserving

the coherence of spatial information (3) Ways to incorporate the information provided by user-input into the optimization process. Below, we discuss each of these issues in brief.

In order to have a generic mechanism for describing patterns, we need to have a method for detecting interest points on the patterns reliably. The color / intensity information may not hold important cues for corner detection if the saliency of the required points is primarily geometric. A pattern in our case is represented using a collection of image contours, which could be badly fragmented and erroneous. We note that local shape properties such as derivatives at a point would be damaged badly by the operation of perspective projection. However, global shape properties such as the relative distance of points from the center of mass of the contour would not be badly affected, though not guaranteed to be preserved. We shall use these global properties and guardedly compute a set of interest points. We apply a neighborhood of a given size and suppress interest points which are not maximal in their neighborhoods. In the thesis, we demonstrate through experiments that this method identifies interest points reliably on a perspectively distorted image of the pattern. We compare our results with those provided by the method of Shi-Tomasi.

After the detection of interest points, we study the problem of pairing two sets of points uniquely with each other. We provide an effective solution for this problem by an optimization approach which tries to maximize the spatial coherence subject to a consensus on geometric constraints. Our method stands in contrast to classical methods which try to match points through the use of geometric invariants. Instead, we try to satisfy spatial coherence between the point matches - which means the conservation of Euclidean properties such as angle and ratio of lengths. Though it is true that these Euclidean properties are no longer valid after a projective transform, we show that they can be preserved in an approximate sense. Unlike invariant-based methods, our method has the added advantage of being robust to noise and outliers. We frame the optimization problem in a greedy setting and thus obtain a locally maximal solution. This produces a matching between the two point sets. When the replication (multiplicity) of the patterns is more than two, they are handled through generalizing our solution to the entire set of points. The model is solved using Levenberg Marquardt optimization. This is similar to the bundle-adjustment algorithm in stereo.

We study the problem of geometric grouping in the context of an interactive application. One of the major concerns for such an application would be to minimize the level of user-interaction and also to be able to tolerate errors at the micro-level. Previous methods of IBMR through plane-based rectification have required the user to provide information at the pixel level, in the form of parallel and perpendicular line segments. This method is not only error-prone but also extremely taxing for the user to provide. In our application, we provide new ways for the user to interact with the system and new means of incorporating

this user-input into the optimization process. We demonstrate that the user input is indeed useful in reducing the combinatorial complexity of the algorithm by a large factor.

The method of geometric grouping of patterns has several applications. We discuss the direct application of an interactive image-based modeler. We provide references to the other applications - *tracking, recognition, stereo etc* in the future work section.

Dedicated to the city of Hyderabad

Acknowledgments

It is my pleasure to thank all the people who have been part of my CVIT experience. Firstly, I wish to thank my adviser Dr. P. J. Narayanan. He has been very supportive during the ups and downs of my research, and has been a great friend, mentor and guide. I would also like to thank all the members of my lab for all the good times, valuable discussions and dialog. Particularly, I wish to thank Visesh and Paresh for the discussions on structure from motion, and Vardhman for the discussions on image segmentation.

I wish to thank the researchers who have corresponded with me over email and helped me understand key ideas - Marc Proesmans, Tinne Tuytelaars and Allen Yang. I wish to thank my friends at INRIA, Rhône-Alpes for the fruitful dialog they have given me - Srikumar, Pao and Andrei. I wish to thank my friend and senior Sujit for valuable discussions on planar object recognition and for sharing software.

Finally, I wish to thank my family - *Amma, Nánna, Rénu and Stalin* for always having faith in me.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Layered Computer Vision

Computer Vision deals with the understanding and processing of visual information from images. The several tasks of computer vision can been grouped into 3 levels.

- Low-level Vision : edge detection, segmentation, image restoration
- Mid-level Vision : grouping
- High-level Vision : object recognition

For handling low-level vision tasks, efficient algorithms have been developed based on advances in machine learning [1] and optimization theory [2]. High-level vision tasks prove much harder to handle - because of two reasons. Firstly, they require far more intelligent algorithms which should perform on par with human perceptive skills. Secondly, they rely on the effectiveness of the prior two levels. The errors in the lower level processing get propagated through the pipeline.

In contrast to low-level tasks, handling mid-level tasks in a robust manner has not been attempted thoroughly by the research community. All the same, mid-level tasks tend to be very important - they are crucial in converting the 2D information of the images into the 3D information of the real world. Without good mid-level processing, high-level problems become extremely ill constrained.

When geometric primitives such as curves and patterns appear in repetition over a perspective image, they offer key information in recovering the real metric structure of the scene. Mid-level vision deals with the task of grouping these patterns into sets of geometric relevance, i.e, into sets of patterns which produce a uniform geometric constraint. For example, a unique vanishing point. We need effective methods for harnessing this information in a robust manner.

1.2 Mid-level Vision : Symmetry as a Generic Notion

The field of photogrammetry [3] has long been investigating several geometric constraints to aid in the process of 3D reconstruction. These constraints occur in the form of parallel lines, perpendicular lines, regular curves such as conics and so forth. The geometric tool which effectively handles all these constraints is called the absolute conic. However, each of these constraints is a specific instance of a more generic notion of symmetry. Symmetry occurs in the world due to the repeated occurrence of a pattern. It occurs in several forms such as

- Bilateral Symmetry - when a pattern is reflected onto a mirror image, over an axis of reflection.
- Translational Symmetry - when a pattern is tiled periodically along a specific direction.
- Point Symmetry - when a pattern is rotated around an axial point to yield a symmetric shape.

These cues are very helpful in constraining the absolute conic of a given image, and thus obtaining the corresponding 3D reconstruction of the scene. Sometimes, an object exhibits multiple forms of symmetry at the same time. The patterns that get repeated in an object can be thought of as a grammar which generates the 3D shape of the object. These patterns translate the problem of computer vision into a symbolic learning task.

1.3 Patterns as sets of Interest Points

We define a pattern as a complex curve or a collection of curves that can be detected multiple times in an image. This definition is relevant to our discussion in the previous section - a pattern is a geometric primitive that can be used for grouping.

However, in this thesis, we do not make any assumption that the patterns are detected exactly a priori. Instead, we take as input an image over which several primitive image contours have been detected. This can be done, for example, through edge detection followed by linking edgels which are in close proximity. The image contours thus identified would be poorly defined and badly fragmented, i.e, they suffer from the following twin problems

- A single pattern is split into Several image contours
- An image contour might belong to more than one pattern

In this scenario, we deal with the problem of detecting patterns and grouping them together into geometrically relevant bins. However, we bypass the problem of exactly detecting each pixel which belongs to a pattern. Instead, we try to detect each pattern as a set of a few

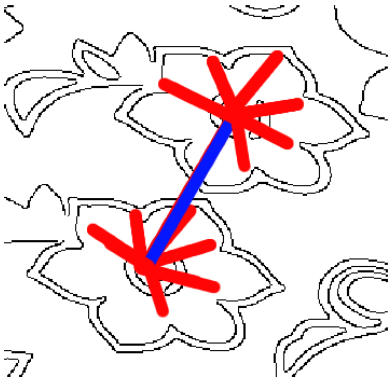


Figure 1.1: Use of spatial coherence for matching interest points

interest points.

It becomes possible to do this, because we show that the set of candidate interest points can be computed directly from the fragmented contours. The patterns are later detected (and simultaneously grouped) as an outcome of the geometric grouping task, which is performed on the entire set of candidate interest points.

1.4 Geometric Grouping

Geometric grouping is the task of grouping interest points into sets which produce a uniform geometric constraint. For example, the interest points which produce a unique vanishing point are grouped together into a single set. This grouping needs to be performed in a robust manner, because of noise in the estimation of the interest point positions. The grouping also needs to be robust against the presence of outliers.

The task of geometric grouping is defined as a discrete optimization problem which tries to maximize a geometric constraint consensus (such as vanishing point consensus). The optimization is constrained by the principle of *spatial coherence*. That is, if two points P_i, P_j are grouped together, we expect the neighborhood of P_i to be similar to that of P_j. Spatial coherence, as used here, is similar to the principle of temporal coherence which is used in tracking. However, the neighborhood is defined not in terms of the local color/intensity information, but in terms of the relative positioning of the other interest points. In the thesis, we shall demonstrate how this information is more useful in the context of geometric grouping.

We address the problem of geometric grouping from the context of an interactive image-based modeling and rendering (IBMR) application. We will demonstrate how geometric

grouping with minimal user interaction can be used for improving the robustness of structure recovery. The user-input is incorporated into the optimization process, producing substantial gains in accuracy and computational time.

1.5 Overview of the System

The following are the components of our system.

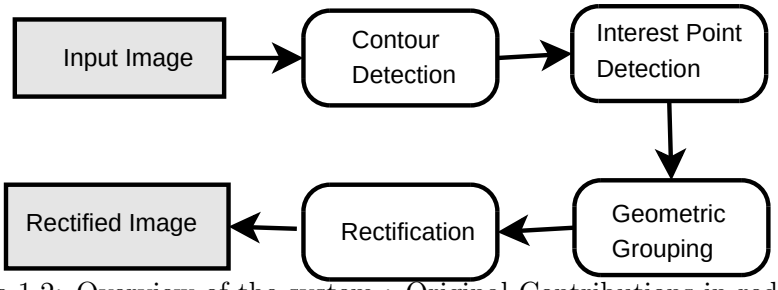


Figure 1.2: Overview of the system : Original Contributions in red color

The image is pre-processed to detect edges and link the edgels to form image contours. This is accomplished by using the OpenCV software library. A set of candidate interest points are identified using the algorithm described in Chapter 3. The user then interacts with the system to select the region of interest, to specify the direction of match or to directly identify the fixed structure of the match. The ways of this interaction are described in Chapter 5. The algorithm takes over and groups the interest points together based on the discrete optimization process described in Chapter 4. When the vanishing line of a plane is identified, the user can select to obtain the affine rectification of the scene. If the circular points are also identified, the user can select to obtain the metric rectification of the scene. These rectified images can be saved for later processing, for example, as textures for a 3D model.

In figure (1.3), these steps are demonstrated on a sample image.

1.5.1 Principal Contributions

Our principal contributions are in the following areas.

1. **A novel interest point detection :** We have developed a new method for detecting interest points using macro shape properties. This method is purely geometric and is not dependent on appearance except for computing a set of image contours.

Figure 1.3: Different stages of the algorithm on the sample image of a design on a cloth (a) Input Image (b) Detected Contours (c) Interest Points (d) Vanishing point estimation through grouping (e) Feature correspondence across patterns (f) Rectified image

2. **Using spatial coherence for geometric grouping :** We have formulated the problem of grouping patterns according to geometry consensus, as one that is hinged on the spatial coherence of local neighborhoods. This formulation is akin to the smoothness constraint in stereo, albeit at a bigger neighborhood. We discuss several formulations of spatial coherence, based on pure geometry based analysis.

3. **Interactive geometric grouping :** We formulated the geometric grouping as an interactive problem which utilizes user-input to perform the job in a more efficient manner.

1.6 Motivation - Applications

Geometric grouping is a generic tool which is useful in several applications of computer vision. We discuss one direct application - that of an interactive image based modeler. This application could be used in several areas such as

- Modeling architectural scenes
- Virtual tourism of patrimonial buildings
- Crime scene investigation
- Accident scene reconstruction
- Creating background environments for blue-screen shots

Currently, there is commercial software available to obtain the 3D models from images [4] [5]. Such software uses classical constraints such as parallel and perpendicular lines. Also, the user input is required to be in the form of marking points, lines and edges on the image. Such input is very laborious to provide. By effective use of geometric grouping, we show how to make this job of the user much easier. Specifically, we do not make any obligation on the user to provide any input with pixel-level accuracy.

Since geometric grouping is a good solution for the mid-level vision problem, this expertise can be beneficial to any high-level vision task - such as object recognition, object tracking, optical character recognition etc. We discuss these applications briefly in the future work section (Chapter 6).

1.7 Spatial Coherence - Related Work

Our method for geometric grouping draws inspiration from the recent techniques of exploiting spatial coherence towards solving vision tasks. We discuss them in this section.

1.7.1 Image Segmentation

Archaic methods for image segmentation have required the user to specify the boundary approximately as an initialization. In the method of Boykov and Jolly [6], the image segmentation problem is defined in a framework analogous to stereo. This method demands only a few pixels from the interior of the object, a few from the background and none from the boundary. These can readily be provided by the user.

$$E = \Sigma_i D(i) + \Sigma_{i,j \in N} S(i,j) \tag{1.1}$$

The image segmentation problem can be understood as the minimization of the energy (1.1). This energy is composed of a *unary* potential $D(i)$ (which depends on each individual pixel i) and a *binary* potential $S(i,j)$ (which depends on each pair of pixels i,j)

The problem of image segmentation is a binary labeling problem - which is handled very intuitively through the framework of graphcuts. The graph is constructed with each pixel in the image constituting a node. There is an edge between each pair of neighboring nodes (between each pixel and its 8-neighbor). The weight of this edge corresponds to the closeness of the intensity values of the two pixels. This weight is called the *smoothness energy* $S(i,j)$ in the context of *MRF* energy minimization. This corresponds to the *binary potential* in the equation (1.1). The *unary potential* $D(i)$ is called the *domain energy* and represents how closely the labeling of a given pixel matches the probability computed from the real data.

There are two additional nodes in the graph which are called foreground (*frg*) and background (*bkg*). These two nodes form the source and the sink vertices of the graph. If the user marks a pixel as foreground, the edge which connects that pixel to *frg* is given a weight of ∞. At the same time, the edge which connects this pixel to *bkg* is given a weight of 0. The converse holds if the user marks a pixel as background instead of foreground. Two probabilistic models (typically histograms) are constructed for *frg* and *bkg* based on the intensity values of the pixels input by the user. The rest of the image pixels are assigned probabilities based on these models. These probabilities are represented in the graph as weights of the edges connecting the pixels to the two nodes *frg* and *bkg*. It is simple to prove [6] that this graph completely encodes the energy in equation (1.1). The *mincut* of this graph can be obtained through the method of Kolmogorov et al [7].

If the user is not satisfied with the output of the labeling, he can provide more input (for instance, if he notices that part of the foreground is labeled incorrectly, he can mark more pixels as foreground) and the graph would be modified in a marginal manner. The value of the *flow* computed from the previous iteration could be used now, and optimization on this

newer graph takes a shorter amount of time than doing it from the beginning.

In our system, we have implemented the graphcut based segmentation tool as a form of user interaction. The user can mark pixels as foreground or background using a *brush*. This tool is used for selecting regions of interest, as will be discussed in Chapter 5.

In *ObjCut* [8], the authors provide an automatic method for image segmentation which is coupled by bayesian learning on a large set of training images. Using this training, the system dispenses with the user-input that is required for the construction of the graph, and automatically segments the object out of the image. Since the system claims to segment 3D objects in their generality, the 3D connectivities have to be handled, which are done by *pictorial structures*. Thus, this method encodes a higher degree of spatial coherence than the one that is provided by the simple 8-neighborhood of pixels. The graphical model is solved using belief-propagation.

Apart from image segmentation, this method of using spatial coherence in the local 8-neighborhood of the image pixels has also been successfully employed in stereo [2], colorization [9] etc.

1.7.2 Geometric Context

Pioneering work in marrying appearance models to geometric reconstruction is recently done by Hoeim et al [10]. Instead of segmenting an image into object and background, Here, the authors try to label each pixel as belonging to a geometric class. Three geometric classes are defined - *ground*, *vertical* and *sky*. The authors employ bayesian learning on a large set of labeled training images, to help the system learn these geometric classes. Given a new image, the appearance model is used to assign a score for each pixel with respect to each label.

Instead of using the pixels directly, the system first clusters them into *superpixels* using simple mean-shift segmentation. These superpixels have similar appearance information - such as color and intensity. The system then tries to label each superpixel into a geometric class using the appearance models that are learnt through training. In doing so, the system tries to preserve spatial coherence - that is neighboring superpixels are required to have similar labels. This provides the binary potential between the nodes in the Markov Random Field. The unary potential is provided through the appearance model. The energy of the MRF is minimized using belief-propagation.

The geometric labels are used to automatically construct primitive 3D models from a single image [11]. These models are termed as *automatic photo-popup models*. When used

in conjunction with the real image, these geometric labels provide key information to improve the performance of several vision tasks. In [12], the authors describe how to employ this information to improve the efficiency of an object detector.

1.7.3 Shape Contexts

Instead of the classical 8-neighbors of pixels, Malik et al [13] try to analyze a greater neighborhood in the image. A histogram is constructed with regard to the variation of shape around the several direction-bins for each pixel. This histogram is termed as the *shape context*. Shape contexts have been employed for several tasks of computer vision - object recognition, recovery of pose [14] etc.

Shape-contexts are similar in principle to an earlier work called *spin images* which works for 3D points [15]. Spin images are more powerful because they do not have to address the problem of the distortion of shape which happens due to perspective projection. Spin images have been successfully used for 3D object registration and object recognition [15]. A 3D version of the shape-context has been proposed as an alternative to the spin-image [16].

Of specific interest to the current thesis is the problem of non-rigid shape alignment in 3D. An interesting algorithm has been proposed by Anguelov et al [17] to solve this problem through energy minimization on an MRF. The unary potentials of the MRF are encoded through local shape properties such as spin-images. The binary potentials are encoded as a means of preservation of geodesic distances between the 3D surface points, and also as a means to minimize elastic deformations. Thus, spatial coherence has been employed in a much stronger fashion.

In the current thesis, we try to enforce spatial coherence in a similar scale and fashion. Though we do not encode this as potentials of an MRF as done by [17], we do penalize the solutions which do not satisfy spatial coherence.

1.8 Shape from Symmetry - Related Work

The utilization of geometric invariants towards scene reconstruction has been a highly popular topic in the computer vision research community. In the recent past, there has been an emerging trend to discuss these invariants under the holistic concept of symmetry. In the current section, we provide a historical overview of this research.

1.8.1 Regular Curves and Polygons

Rectangles In a perspective view, two parallel lines intersect at a vanishing point. So two sets of parallel lines are sufficient to identify the vanishing line of the plane, which yields affine rectification. To obtain metric rectification one needs two sets of perpendicular lines [18], none of which are parallel.

Conics Regular curves such as conics have a fixed number of parameters which can be used to constrain the operation of perspective projection. If two imaged conics are known to be circles, they can be used for the metric rectification of the plane [19]. This follows from the fact that the circular points of the plane can be directly recovered from the points of intersection of both the circles.

1.8.2 Translational Symmetry

Generic Tiling A shape is termed to be translationally symmetric, if there exists a canonical frame of reference, where the shape is unaltered by the application of the translation transformation.

$$T = \begin{bmatrix} 1 & 0 & t_1 \\ 0 & 1 & t_2 \\ 0 & 0 & 1 \end{bmatrix}$$

Traslational symmetry can be observed in man-made structures of several forms - tiles on the floor, windows on a wall, bars on a fence etc. This kind of symmetry provides a strong cue for grouping similar objects, which later aide in the computation of vanishing points. This was first noticed by Schaffalitzsky et al [20]. Until then, the popular method for discovering principal directions in an architectural image was through detecting lines and clustering them together (for example, by applying the Hough transform). It has been noticed by [20] that if the image exhibits translational symmetry, it offers more information than given by an arbitrary positioning of lines. They have modeled the translation operator in a canonical frame and explained the process to compute the homography which maps the image plane to this canonical plane.

In a perspective image, two planar shapes related by a translational symmetry have a restricted image homography. Instead of 8 *d.o.f*, this transformation has only 4 *d.o.f*. This is called an *Elation* and is discussed in detail in Chapter 2.

Equally spaced Lines A special case of translational symmetry is when a set of straight lines are detected in the image which are equally spaced in the real world. For a set of parallel lines, it is possible to identify the vanishing point. But when the lines are also equally spaced, it is possible to obtain more information. From a set of 3 or more such lines, it is

possible to completely estimate the vanishing line l_∞ of the plane. The information required to compute the other vanishing point (and thus, the vanishing line) is obtained through the known ratio of lengths across the parallel lines.

Let l_1, l_2, l_3 be 3 equally spaced lines. Let l be any line which does not pass through their mutual point of intersection (vanishing point). Let $[a, b, c]$ be the determinant of the 3×3 matrix which consists of a,b and c as columns. Schaffalitzsky et al mention that there exists a closed form solution for computing the vanishing line of the plane.

$$l_\infty = [l_1, l_2, l]l_3 \; - \; [l, l_2, l_3]l_1 \tag{1.2}$$

1.8.3 Bilateral Symmetry

This type of symmetry is also known as *reflective symmetry* or *mirror symmetry*.

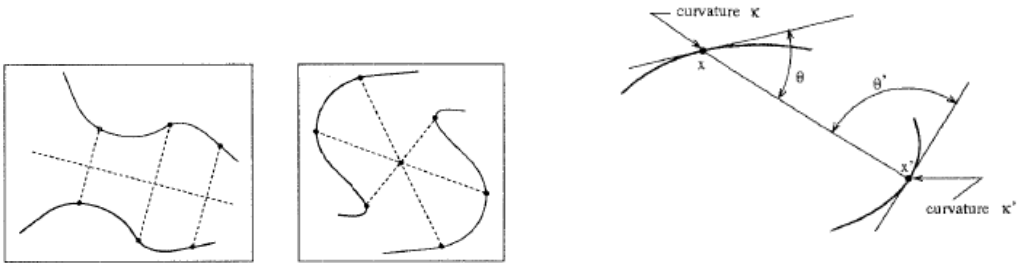
Mirror Symmetry in an Affine View Mukherjee et al [21] have given several geometric invariants which are satisfied by a symmetric shape under an affine camera. They mention that if an uncalibrated image exhibits two kinds of symmetry which are neither parallel nor orthogonal, then the image can be backprojected to obtain a metric rectification of the plane. It is discussed further in Chapter 2 about how to use the known ratios of lengths across two intersecting lines, to uniquely solve for the circular points of the plane. These known ratios of lengths can be constructed from the symmetry of the shape.

In [21], the authors, however, provide an alternative framework for doing the same thing, initially by providing a parametrization for the affine camera matrix and later by solving for this through linear constraints.

The geometric invariants that have been mentioned to check for symmetry under affine camera include

- **Area in the image space** between three points of a shape and between their reflections.
- **Moments in the canonical frame**, the frame as defined by Lamadan et al [22], which consists of an equilateral triangle at vertices (-1,0), (1,0) and $(0,\sqrt{3})$.

Mirror Symmetry in a Perspective View Proesmans et al [23] is the first work to identify algebraic invariants for a symmetric shape under perspective skewing. They introduced the notion of fixed sets for dealing with the homography connecting a planar shape to its symmetrical counterpart. They argued that, for mirror symmetry, the fixed set produces a harmonic homology. This homology, as will be discussed in Chapter 2, has 4 degrees of



(a) (b)

Figure 1.4: Mirror and Point Symmetry under Perspective Skew : (a) example shapes (b) Ponce criterion

freedom and can be identified by specifying two pairs of corresponding points.

Let's say that the points x_i have a symmetrical correspondence with the points x'_i. Proesmans et al [23] have mentioned several invariants which we briefly discuss below.

Fixed pair and curvature This invariant was first introduced by Ponce [24] who has proved it in the context of an affine camera. This is proved to be true even in the perspective case. The invariant depends on the curvature κ and the angle θ as revealed in the figure 1.4

$$\frac{\kappa}{\kappa'} = \frac{sin^3\theta}{sin^3\theta'}$$

Equivalently this can be written as

$$\frac{|x^{(1)}x^{(2)}|}{|x - x'x^{(1)}|^3} = \frac{|x'^{(1)}x'^{(2)}|}{|x' - xx'^{(1)}|^3} \tag{1.3}$$

Starting from a single (hypothesized) correspondence x_1,x'_1, one can find additional pairs of correspondences by using the symmetry specific invariant parameter

$$\int abs\left(\frac{|x^{(1)}\ x^{(2)}|^{1/3}}{|x_1 - x'_1\ x_1 - x|}\right) dt \tag{1.4}$$

To check if the assumption of symmetry is indeed correct, for each new point pair, the Ponce relation 1.3 can be checked. However, the problem with this is that it cannot be used in the local neighborhood of the hypothesized pair x_1, x'_1. Proesmans et al [23] mention a couple of invariants that can be used if one needs to detect symmetry locally.

Fixed pair and tangents If the application of the invariant does not have to be local, we can obtain more robust invariants. This is done by getting rid of the second order derivatives altogether. Only the left hand side of the invariant equation is mentioned. (Upon swapping roles of x with x' and x_1 with x'_1, the invariant should coincide).

As invariant parameter one can use

$$\int \sqrt{abs\left(\frac{|x_1 - x\ x^{(1)}||x'_1 - x\ x^{(1)}|}{|x - x'_1\ x_1 - x|^4}\right)}dt \tag{1.5}$$

and to check if the assumption of symmetry is validated, one can use the invariant

$$\frac{|x - x_1\ x_1^{(1)}||x - x'_1\ x_1'^{(1)}|}{|x'_1 - x_1\ x_1^{(1)}||x_1 - x'_1\ x_1'^{(1)}||x - x_1\ x - x'_1|^2} \tag{1.6}$$

Two Fixed Point Pairs If two pairs of symmetric points are known or hypothesized, one is completely informed about the transformation and this information can therefore be checked directly for matching purposes. If we denote the two selected fixed pairs of points as x_1, x'_1 and x_2, x'_2, the following pair of invariants can be used to validate the assumption of symmetry.

$$\frac{|x - x_1\ x - x'_1|}{|x - x_2\ x - x'_2|} \tag{1.7}$$

$$\frac{|x - x_1\ x - x'_1|^2}{|x - x_1\ x - x_2||x - x'_1\ x - x'_2|} \tag{1.8}$$

There are a few other invariants that are mentioned in [23] and a few more which can be derived from [25]. In the context of image based modeling, these invariants have the utility of detecting symmetrical correspondences after a putative set of points is found, for example, through the method that we propose in Chapter 3.

As we shall discuss in the Chapter 2, reflective symmetry produces a special case of plane projective transformation called the planar harmonic homology. Specifically, the invariant 1.7 is actually related to the geometric invariant 2.32 of Chapter 2.

1.8.4 Rotational Symmetry

Point Symmetry in Perspective View If a curve is reflected across a point instead of an axis of reflection, we obtain point symmetry instead of bilateral symmetry. It is proved in [23], that *point symmetry* is exactly equivalent to the case of *mirror symmetry*. That is, as discussed in section 1.8.3, this type of symmetry leads to a harmonic homology, which has 4 degrees of freedom. There are two fixed structures - a fixed point (*vertex*) and a

fixed line of points (*axis*). The difference between *mirror* and *point* symmetry lies in the kind of structure that needs to be moved to infinity. It is the *vertex* in the case of mirror symmetry, and the *axis* in the case of point symmetry. But after projective distortion, both these symmetries are equivalent. The invariants discussed in section 1.8.3 work for the case of point symmetry as well.

When using this kind of symmetry for grouping, it pays to first discover the *vertex* which is the point of concurrence of all the corresponding pairs. This is very similar to the method of handling perspectively distorted parallel lines which arise due to mirror symmetry (the *vertex* being the vanishing point). After the discovery of the vertex, tighter invariants (discussed in 1.8.3) can be employed for identifying the remaining correspondences.

Generic Rotational Symmetry A shape is termed to be rotationally symmetric, if there exists a canonical frame of reference, where the shape is unaltered by the application of the rotation transformation.

$$R = e^{i\theta} = \begin{bmatrix} cos\theta & -sin\theta & 0 \\ sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Point symmetry is a special case of rotational symmetry where the angle of rotation θ is equal to π.

Van Gool et al [26] have mentioned uses of rotational symmetry towards extracting the projective structure from a plane. They have relied on the theory of fixed sets and groups, which condense the plane projective transformations (homographies) into homologies. The invariants that can be employed for the generic case of planar homologies are mentioned in 2.11.1. Specifically the cross ratio of the areas of two corresponding point pairs will be preserved.

If one knows about the angle of rotation, then stronger invariants can be derived, specifically for the case where the angle of rotation is $\frac{\pi}{n}$ for an integral value of n. Interested reader may refer to [26]. Identifying 3 pairs of corresponding points will be sufficient to fix the transformation.

Rotational Symmetry of 3D Structures Liu et al [27] have extended the ideas of [26] for 3D structures with rotational symmetry. They have proved that it is possible to recover metric properties, such as angles, directly from an uncalibrated image of an object taken from an unknown viewpoint. The images of symmetric counterparts of a scene, taken by an unknown perspective camera, will be related by the fundamental matrix F. For computing the fundamental matrix, it pays to become aware of the fixed points of the rotational

transformation. For such a point x, the epipolar constraint becomes an equation in the quadratic form $x^\top Fx = 0$.

The fundamental matrix is composed of the symmetric (F_S) and anti-symmetric (F_A). Since $F_A = -F_A$ by definition, the anti-symmetric part contributes nothing to the quadratic form. The fundamental matrix is a rank-two matrix, but in general the symmetric part F_S is a matrix of full rank. However, if the rotation axis is perpendicular to the direction of translation, or if the translation is zero, F_S drops rank.

In that case, the conic defined by the quadratic form $x^\top F_S x = 0$ becomes degenerate into a pair of straight lines. One of these lines, l_A is the image projection of the axis of rotation and the other line, l_∞ is the vanishing line of the plane perpendicular to the rotation axis. The fundamental matrix F, in this case, has 6 degrees of freedom, which can be estimated by 6 correspondence pairs. (2 out of the 8 *d.o.f* are reduced by the equations $|F| = 0$ and $|F_S| = 0$.

Unfortunately, this is the best that can be attempted in the uncalibrated case of 3D data.

1.8.5 Symmetry as a Homography Group

When the structures are planar, it is possible to treat each form of symmetry mentioned above through the unified framework of homography groups [28]. A set of points $S \subset \Re^3$ is called a *symmetric structure* if there exists a non-trivial subgroup G of the Euclidean group $E(3)$ that acts on it. That is, for any element $g \in G$, it defines a bijective (one-to-one, onto) map from S to itself.

$$g \in G : S \rightarrow S$$

Each element g of G is a transformation of points. These transformations are explained in the previous sections for reflexive, rotational and translational symmetries. Since an object can exhibit multiple types of symmetry, it could be invariant under a *set* of transformations, called as a *group*. Mathematically, symmetric structures and groups are equivalent ways of representing symmetry. The symmetric structure is invariant under the group action (called as *fixed structure*).

There exists a unique frame of reference for each symmetric object in which these group operations are best described. It is revealed by Yi Ma et al [28] that, provided the object exhibits enough symmetry, it is possible to recover the canonical frame of reference from a single perspective image. This is intuitive because, we have seen how the different forms of symmetry produce fixed structures and metric constraints. What the authors [28] have done is to interpret this in the unified framework of homography groups.

Homography Group: Example of a Rectangle As shown in figure 1.5, the symmetry group of a rectangle is a dihedral group D_2 of order 2, and can be represented in the canonical object frame (x, y, z) using four matrices.

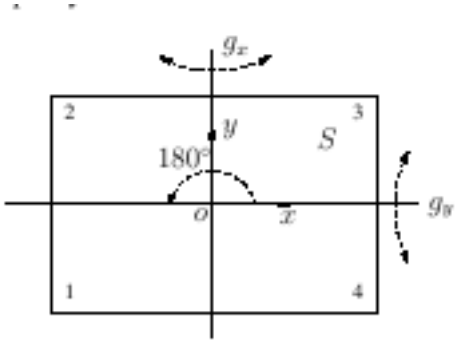


Figure 1.5: Homography group for a rectangle

$$g_e = Ig_x = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} g_y = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} g_z = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Amongst these transformations, g_e stands for the trivial identity transformation, g_x and g_y stand for bilateral reflections across the X and Y axes and g_z stands for rotation of π across the Z axis.

Recovery of Pose If the images are calibrated, Yi Ma et al [28] provide means of obtaining the relative pose from point correspondences. Each point correspondence produces an epipolar constraint, and the essential matrix is solved for. The relative rotation R and translation T can be obtained using SVD, or a factorization algorithm such as the outlined in [29].

The canonical pose of the camera (the rotation R_0 and translation T_0, with respect to the canonical frame of reference) need to be further estimated from these values. This is done by solving a set of simple equations (called the *Lyapunov type* equations) of the form

$$R(g_i)R_0 - R_0R_i = 0 \qquad i = 1, 2 \ldots m$$

with $R(g_i)$ (depending on the type of symmetrical transformation) and R_i (relative rotation computed from factorization) known for each i.

Structure Recovery without Correspondences There could be a case where point correspondences cannot be obtained, but the symmetric structure be decomposed into two

curves which are symmetric counterparts. In this case, if it is possible to obtain the essential matrix using an alternative method, canonical pose of the camera can be estimated as mentioned by [28].

Solving for the essential matrix without point correspondences is a hard task. But there are some techniques currently available. One can directly estimate the homography using fourier domain representation of the discrete contours [30].

With this approach, it would be possible to estimate the canonical pose without any kind of point correspondences. It is in this scenario, that the *homography-group* based reasoning of [28] becomes indispensable, the alternative means of estimation via vanishing line/circular points not being feasible without point correspondences.

However, there is a catch - the camera needs to be calibrated earlier to apply this method.

1.9 Geometric Grouping - Related Work

Geometric grouping of repeated elements in an image has been attempted through several approaches in the vision community. Each of these approaches have taken advantage of the understanding of symmetry as gleaned by the work reported in the previous section. Though they differ in the type of geometric constraints that they employ, there are two unifying strands for the various methods of geometric grouping that have been proposed so far.

1. They tried to apply the geometric invariants exactly. This approach has a problem when the image primitives are not identified exactly due to noise.

2. They diligently avoided any type of combinatorics in the grouping stage. This has been done to prevent the problem from getting into a combinatorial explosion. However, this approach leads only hashing based algorithms such as *Cascading Hough Transforms* [31] which may be too simplistic.

In the current thesis, we diverge from both these strands. Also, the past methods have presented completely automatic methods of geometric grouping which have obvious applications towards object recognition etc. In the current thesis, we emphasize on interactive methods which overcome some of the crucial issues related to combinatorial complexity. Such interactive algorithms of geometric grouping have applications towards interactive 3D reconstruction. However, our algorithms are also amenable for further modifications to be made completely automatic.

The past work on geometric grouping can be broadly divided into two groups based on the image primitives on which they work on. The first group of methods take line segments as primitives where as the second group of methods work on image patches.

1.9.1 Linear Features

The most recent work by Allen Yang et al [32] presents a hierarchical procedure to detect and group symmetrical structures. It is completely automatic procedure which effectively identifies most, but not all, of the regular repetitions such as tiles, books etc. A principal limitation for this approach is that the patterns are assumed to be composed of linear features. The absence of such features makes this approach incapable of dealing with those shapes.Thus the method has reported results only on shapes such as rectangles even though the theoretical basis for this work is strong enough to deal with arbitrarily complex shapes.

Earlier works based on linear features such as that of Schaffalitzsky et al [20] have also reported results on structures such as bars on the windows and fences. A principal absence in such works are results on images such as tiled repetitions on a carpet.

1.9.2 Image Patches

The limitation of linear features is addressed to some extent by considering the texture information of image patches as primitives to be fed into the *Cascading Hough Transform*. Affine invariant image patches [33] are considered for grouping by Turina et al [34]. This approach has been built over the framework of planar homologies [25] and has presented catchy results on images such as the wings of a butterfly, designs on a carpet etc. The framework of affine-invariant image patches has been successfully employed in the problem of wide baseline stereo [33]. However, there is a distinction between the shapes related by symmetry and those by stereo. Neither the interior nor the exterior of a symmetrical shape satisfy the assumption of symmetry. The definition of symmetry is based entirely on the boundary of the shape, so to say. So, the approach of using image patches for the detection of symmetry is not effective in all the cases. Particularly, it is not useful when there is not enough variation in the texture surrounding the shape, or when this texture is misleading.

In the recent past, several low-level vision problems such as object segmentation have successfully been attempted through combinatorial optimization [6] [1]. In our current work, we apply combinatorial optimization for the mid-level task of grouping. We consider the task of geometric grouping in this framework and present a greedy solution to this problem.

1.10 Organization of the Thesis

The thesis is organized into the following chapters.

1. **Introduction** This chapter describes the setting of the thesis and outlines the components of the system. It also provides a detailed overview of the related work.

2. **Hardness of the Plane Rectification Problem** This chapter provides the mathematical background for the problem of 3D reconstruction from a single view. The nonlinearities involved in imaging by a perspective camera are discussed. The types of information required for the affine, metric and Euclidean structure recovery are described. It is also described how this information fits into the unified framework of symmetry. This is further illustrated through certain examples.

3. **Interest Point Detection using Macro-Shape Properties** This chapter describes our method for identifying the set of candidate interest points. It discusses the relative effectiveness of macro-shape properties vs micro-shape properties in identifying the interest points of a pattern from a perspective view. Results of our method are presented on several input images of badly fragmented image contours. These results are compared with those produced by the Shi-Tomasi algorithm [35]. The virtues of using geometric saliency for interest point detection are explained.

4. **Geometric Grouping using Spatial Coherence** This chapter deals with the core contribution of the thesis - the method for geometric grouping of interest points. The grouping problem for a pattern with a multiplicity of two is formulated as a discrete optimization problem - which is solved using a greedy algorithm. The importance of using spatial coherence to constrain the optimization process is explained. When the pattern has a multiplicity of more than two, it is explained how to generalize the solution using Levenberg Marquardt optimization. Results of grouping are presented on a wide array of images.

5. **User Guided Geometric Grouping** This chapter deals with the problem of geometric grouping from the perspective of an interactive application. The issue of user-friendliness is discussed and justified. Examples are presented on the different methods of user-interaction. Ways to incorporate this user-input into the optimization process are detailed.

6. **Conclusion** This chapter summarizes the contributions of the thesis and outlines the prospects for future work.

Chapter 2

Hardness of the Plane Rectification Problem

It is possible to obtain the 3D scene structure and camera pose given a single image. This is intuitive - human beings are capable of visually understanding the scene by looking at a single painting or a photograph. However, it is straightforward to see that there is an information loss with the process of perspective projection - many world points get mapped to a single image point. Then how is it that it is possible to obtain a 3D reconstruction ?

The main reason why this is possible is because **the world we live in is very orderly**. We would find large connected objects instead of dusty clouds of points. We would rarely find objects hanging in mid air, but rested firmly on a flat plane, called the ground. Further, we find several objects to be vertical in their orientation, that is, they stand perpendicular to the ground plane. Beyond these aspects, there is also a great manifestation of symmetry in the world. Sometimes, this symmetry is present on top of planar structures. Each of these cues help us in deriving a probable model of the 3D space which is represented by the image.

2.1 Camera Projection Matrix

In homogeneous coordinates, the operation of perspective projection can be represented by the following equation.

$$x = PX \tag{2.1}$$

where X is a 4-vector representing the scene point, x is a 3-vector representing the image point and P is the 3×4 camera projection matrix.

The elements of the camera matrix can be represented as

$$P = \begin{pmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{pmatrix} \tag{2.2}$$

3D reconstruction is the process of identifying the scene point X given the image point x. This is possible if the parameters of the camera p_{ij} are estimated. This estimation is called *calibration and pose recovery*. After this step, each image point is mapped uniquely to a ray which passes from the camera center. However, the scene point could be located at any place along this ray. This ambiguity is later resolved using further assumptions.

This chapter deals with the hardness of estimating each of the camera parameters P_{ij}. Since the entire matrix can be normalized with respect to a single scale factor, there are 11 degrees of freedom in specifying the camera matrix.

These parameters can be principally decomposed into the internal parameters which deal with focal length, camera skew etc, and the external parameters which strictly deal with the camera pose.

$$P = K[R|T] \tag{2.3}$$

The translation T and the rotation R can be specified by 3 parameters each, accounting for 6 parameters in total. The $3X3$ triangular matrix K contains the internal camera parameters, which can be specified by 5 degrees of freedom.

$$K = \begin{pmatrix} f & s & x_0 \\ 0 & rf & y_0 \\ 0 & 0 & 1 \end{pmatrix} \tag{2.4}$$

(x_0, y_0) is called the *principal point* of the projection. f is called the *focal length*, r is called the *aspect ratio* and s is called the skew of the pixels.

In most cases, the camera satisfies the assumption of unit aspect ratio and zero skew. Such a camera is called a natural camera K_n.

$$K_n = \begin{pmatrix} f & 0 & x_0 \\ 0 & f & y_0 \\ 0 & 0 & 1 \end{pmatrix} \tag{2.5}$$

It is possible to estimate the internal parameters K and the external parameters $[R|T]$ separately. The estimation of K is called *calibration* and the estimation of $[R|T]$ is called *pose recovery*.

2.2 Planar Scenes - Homography

If the world points are completely contained in a plane, they can be indexed by a homogeneous 3 vector x_p instead of the homogeneous 4 vector X. The relation that exists between the world plane and the image plane can be represented by a $3X3$ matrix H which is called *planar homography*.

$$x = H * x_p \tag{2.6}$$

where

$$H = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \tag{2.7}$$

This homography H is specified upto a scale factor, and thus, it has 8 degrees of freedom.

There are interesting relations between a known homography H and the possible values of the camera matrix P. These relations are subject to discussion in later sections.

2.3 Homography Estimation - Stratification based Methods

As first described by Koenderink [36] and Faugeras [37], the *plane* $\rightarrow$ *plane* homography matrix can be decomposed into 3 matrices, each of which deal with several aspects of the imaging process.

$$H = S * A * V \tag{2.8}$$

where V is called the pure projective transform, A is called the pure affine transform and S is called the pure similarity transform. They could be estimated in the order - V, A and finally S.

Pure Projective Transform It has **2** degrees of freedom, and is specified by the *vanishing line* of the plane, which can be represented as a homogeneous 3 vector $l_\infty = (l_1, l_2, l_3)^\top$. Estimation of the vanishing line is termed as the *affine rectification* of the plane.

$$V = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ l_1 & l_2 & l_3 \end{pmatrix} \tag{2.9}$$

Pure Affine Transform It has **2** degrees of freedom and is specified by two values α and β which are related to the location of the *circular points* on the the line at infinity. The circular points are a pair of complex conjugate points which are initially located at $(1, \pm i, 0)^\top$. Due to the action of the affine camera, they get transformed to $(\alpha \mp i\beta, 1, 0)^\top$. The pure affine transform is specified by the following equation.

$$A = \begin{pmatrix} \frac{1}{\beta} & -\frac{\alpha}{\beta} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{2.10}$$

Identification of the circular points is termed as the *metric rectification* of the plane.

Pure Similarity Transform It has **4** degrees of freedom, which are specified by the translation t on the plane (2 *d.o.f*), the rotation R (1 *d.o.f*) and an isotropic scaling s (1 *d.o.f*).

$$S = \begin{pmatrix} sR & t \\ \mathbf{0}^\top & 1 \end{pmatrix} \tag{2.11}$$

Technically speaking, these 4 degrees of freedom cannot be identified without specifying Euclidean scene properties, such as the exact location of a scene point and the exact length of a line segment. Identifying these 4 parameters is termed as the *Euclidean rectification* of the plane.

2.4 Plane Rectification - From Projective to Affine

This section deals with the identification of the vanishing line of the plane $l_\infty = (l_1, l_2, l_3)^T$. The vanishing line can be identified by information of two types - parallelism of lines, a known length ratio on a line.

2.4.1 Parallel Lines

According to Euclid, two parallel lines intersect with each other at infinity. However, due to perspective projection, these parallel lines get imaged as non-parallel lines which have a finite intersection point. This point is called the *vanishing point* of the direction. If it is possible to identify two vanishing points, the vanishing line can be reconstructed by joining both of them. For this, we need two independent pairs of parallel lines.

Man-made scenes are usually composed of cuboidal structures which give rise to several pairs of parallel lines. These lines can be identified by using a line detector coupled with a Hough transform [38].

Sometimes it is not possible to identify a line on the image, but the image, nevertheless, admits some parallelism. This happens due to the presence of either reflective or translational symmetry. If corresponding points can be identified across the similar shapes, they can be joined to obtain parallel lines. This is further discussed in Chapter 5.

2.4.2 Known Length Ratio on a Line

If we know apriori the ratio of the euclidean lengths of two intervals on a line that is imaged, we can recover the vanishing point corresponding to that line. This can be computed due to the estimation of the 1D projective transform $L_{2\times2}$ which relates the imaged line with the real line. Applying $L_{2\times2}$ to the point $(1,0)^\top$ would provide us with the vanishing point (in homogeneous coordinates).

Even though a line is not present on the image, it is sometimes possible, to identify 3 collinear points with a known length ratio. This situation arises due to the presence of translational symmetry. This is again discussed in Chapter 5.

2.5 Plane Rectification - From Affine to Metric

This section deals with the identification of the *circular points* of the plane. If the plane is already affine rectified (vanishing line known), then the circular points can be represented as $(\alpha \mp i\beta, 1, 0)^\top$. The circular points can be estimated through a variety of techniques which capture metric information.

We need to estimate the position of the point (α, β) on the complex plane to obtain the circular points. Each of the following constraints yields a circle on the complex plane. The points of intersection of these circles are computed, which later yield the circular points.

Each of the following constraints can be combined. Two such constraints are sufficient to identify the circular points on the complex plane. Since two circles intersect at two points, there would be an ambiguity of resolving between the two points of intersection, but this is explained by the reflection property within the pure similarity transform. The metric rectification, as such, would be completely performed.

2.5.1 Known Angle

Suppose θ is the angle *on the world plane* between the lines imaged as l_a and l_b. Then it can be shown that (α, β) lies on the circle with the center

$$(c_\alpha, c_\beta) = (\frac{(a+b)}{2}, \frac{(a-b)}{2} cot\theta) \tag{2.12}$$

and radius

$$r = |\frac{(a-b)}{2sin\theta}| \tag{2.13}$$

where $a = -\frac{l_{a2}}{l_{a1}}$ and $b = -\frac{l_{b2}}{l_{b1}}$ are the line directions. If the known angle is $\frac{\pi}{2}$, the circle center is on the α axis.

2.5.2 Equality of Unknown Angles

This situation can arise in the real world due to the presence of rotational symmetry. It can also arise due to the repetition of a shape through translation.

Let's suppose that the angle *on the world plane* between two lines imaged with directions a_1, b_1 is the same as that between two lines imaged with directions a_2, b_2. Then it can be shown that (α, β) lies on the circle with the center

$$(c_\alpha, c_\beta) = (\frac{a_1 b_2 - b_1 a_2}{a_1 - b_1 - a_2 + b_2}, 0) \tag{2.14}$$

and radius

$$r = (\frac{a_1 b_2 - b_1 a_2}{a_1 - b_1 - a_2 + b_2})^2 + \frac{(a_1 - b_1)(a_1 b_2 - b_1 a_2)}{a_1 - b_1 - a_2 + b_2} - a_2 b_1 \tag{2.15}$$

2.5.3 Known Length Ratio across Two Intersecting Lines

Suppose that there are two line segments which are non-parallel on the world-plane. If we know the length ratio of the two line segments to be s *on the world plane*, it is possible to provide a constraint on the position of (α, β).

Let's say the first segment is identified on the image plane between the points (x_{11}, y_{11}) and (x_{12}, y_{12}). Similarly, let's say the second segment is identified between the points (x_{21}, y_{21}) and (x_{22}, y_{22}).

Let

$$\delta x_n = x_{n1} - x_{n2}$$

$$\delta y_n = y_{n1} - y_{n2}$$

It can be shown that (α, β) lies on the circle with the center on the alpha axis.

$$(c_\alpha, c_\beta) = (\frac{\delta x_1 \delta y_1 - s^2 \delta x_2 \delta y_2}{\delta y_1^2 - s^2 \delta y_2^2}, 0) \tag{2.16}$$

and radius

$$r = |\frac{s(\delta x_2 \delta y_1 - \delta x_1 \delta y_2)}{\delta y_1^2 - s^2 \delta y_2^2}| \tag{2.17}$$

2.6 Plane Rectification - From Projective to Metric

It is possible to obtain the circular points directly without solving for the vanishing line of the plane. The circular points will be imaged at $((\alpha \mp i\beta)l_3, l_3, -\alpha l_1 - l_2 \mp i\beta l_2)^\top$. Once the circular points I, J are identified, it is thus possible to estimate all the affine parameters of the homography H from these values.

There is a conic which is dual to the circular points. It is termed as the absolute conic and is defined as $D = IJ^\top + JI^\top$. If it is possible to identify the image of the dual conic (**IAC**), it would be possible to identify the images of the circular points, and hence, to identify all the affine parameters.

In general, direct application of metric constraints will generate a non-linear constraint on the parameters. However, for orthogonal lines, the constraint will become linear. It can be shown that two orthogonal lines l_a, l_b are conjugate with respect to the conic D, that is they satisfy the equation $l_a^\top D l_b = 0$.

Since the $3X3$ matrix D is symmetric and satisfies rank-2 constraint, 4 such constraints will uniquely identify it. However, application of the rank-2 constraint is non-linear. If there are 5 constraints available, D can be solved entirely through linear constraints.

But, it may not be always possible to identify five pairs of perpendicular directions for each plane in the image. Hence, it is better to do plane rectification in the stratified manner, by first solving for the vanishing line.

2.7 Hardness of the Plane Rectification Problem

The constraints discussed in the previous sections are applicable to several types of shapes. Most often, the end-points of the shape are not detected appropriately for the proper utilization of these techniques. However, in principle, these techniques offer a holistic solution to image rectification using different kinds of shapes.

2.7.1 Equivalence of the shape *W* to a rectangle

As an example, we demonstrate the equivalence of the shape ***W*** to a rectangle for the purpose of plane rectification.

A rectangle contains two independent sets of parallel lines. In a perspective view, these two sets can be used to obtain two separate vanishing points. These two can be joined to construct the vanishing line of the plane. Beyond that, there is also a single independent

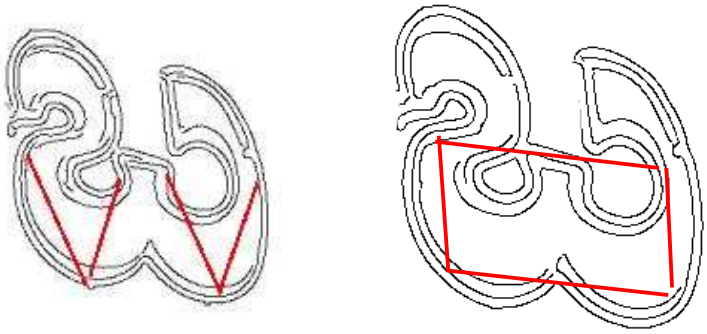


Figure 2.1: Example of a shape which can be used for metric rectification

set (only one pair) of perpendicular lines which yields one constraint on the circular points. Since two constraints are required to solve for the circular points, this is not sufficient by itself. However, if we know the aspect-ratio of the rectangle (the ratio of the length to the breadth), this yields another constraint of the form (2.16) and (2.17). Thus, the plane can be completely metric rectified.

In the shape 'W', we can see two independent sets of parallel lines. These can be used to solve for two vanishing points, and thus the vanishing line. Further, we can impose another constraint that the two angles at the bottom of the troughs in 'W' are equal. This yields a constraint of the form (2.14) and (2.15). To completely solve for the circular points, we need another constraint. If there is further information, for example on the ratio of lengths of the segments 'ánd '/' that make up the 'V' in 'W', we can use this to impose a constraint of the form (2.16) and (2.17). Thus, the plane can be completely metric rectified.

Thus, we notice that the shape 'W' is exactly equivalent to a rectangle. In principle, it can be used for the metric rectification of an arbitrary plane if the 5 end-points of the shape are identified properly.

2.7.2 Information contained in symmetric shapes

Similar information is contained in several shapes which exhibit symmetry. Most often, this information is hard to tap because the interest points could not be detected properly. In 2.1, we show an example of a shape which is identical to the letter 'W' in terms of the constraints that they provide for metric rectification. But these shapes have traditionally been much harder to rectify, due to the lack of proper line segments.

In the table 2.1 we explain the type of constraints that can be used for metric rectification

Nature of Constraint	Type of Structure	Symmetry
Corresponding point pairs yield parallel lines	Vanishing point	Translational, Bilateral
Two pairs of corresponding points yield different parallel lines	Vanishing line	Translational, Point
Corresponding point triads yield equal angle constraints	Circular points	Translational, Rotational
Two pairs of corresponding points yield equal length constraint for non-parallel lines	Circular Points	Rotational, Bilateral

Table 2.1: Nature of constraints for doing metric rectification from symmetric shapes

of different kinds of symmetric shapes.

2.8 Camera Estimation

In this section, we consider the rectification of the world for the generic case where the world is not entirely composed in a single plane. If the scene is piecewise-planar, it may be possible to rectify each of the planes separately. However, all the planes may not contain enough metric information independently, to be processed for rectification. It is helpful to solve initially for the internal and external parameters of the camera, before undertaking the reconstruction of the 3D scene. This section deals with the problem of the estimation of the camera matrix P

2.9 Camera Calibration - Image of the Absolute Conic

We first describe the process of solving for the internal parameters of the camera. These are captured by the 3×3 triangular matrix K as specified in the equation 2.4.

For identifying K, we deal with a very useful entity called the *Image of the Absolute Conic* ω. It is defined as

$$\omega = K^{-\top} K^{-1} \tag{2.18}$$

where ω is a 3×3 symmetric matrix which is defined upto a constant scale factor (5 *d.o.f*).

$$\omega = \begin{pmatrix} \omega_1 & \omega_2 & \omega_4 \\ \omega_2 & \omega_3 & \omega_5 \\ \omega_4 & \omega_5 & \omega_6 \end{pmatrix} \tag{2.19}$$

If we manage to find ω, it is possible to compute the calibration matrix K using a method called *Cholesky Decomposition*. This method decomposes a symmetric square matrix into a product of an upper triangular and a lower triangular matrices. The matrices $K^{-\top}$ and K^{-1} are precisely that. Determining ω in an image, is thus, equivalent to identifying the camera internal parameters.

2.9.1 Rectified Planes

The absolute conic is an esoteric entity which lies on the plane at infinity in 3D. It has the property of containing the circular points of all the planes in 3D. Thus, each rectified plane in the image provides 2 constraints on the absolute conic, corresponding to each of its circular points C_i.

$$C_i^\top \omega C_i = 0 \tag{2.20}$$

It's image IAC (ω) is related to the camera calibration matrix K as mentioned in 2.18.

2.9.2 Orthogonal Directions

IAC is also useful for reasoning about known angle θ in the world between two directions x_1 and x_2.

$$cos(\theta) = \frac{x_1^\top \omega x_2}{\sqrt{x_1^\top \omega x_1}\sqrt{x_2^\top \omega x_2}} \tag{2.21}$$

In particular, when two directions are perpendicular in the world, we have

$$x_1^\top \omega x_2 = 0 \tag{2.22}$$

Thus each *orthogonality* in the world yields a single linear constraint on the IAC.

2.9.3 Reflexive/Translational Symmetry in Orthogonal Directions

As discussed by Yi Ma et al [28], planar symmetry can aid in calibrating the camera. If point correspondences can be identified for translational or reflexive symmetry, they can yield direction vectors v_i which correspond to the vanishing points along those directions. If these directions are orthogonal, they are related using the equation 2.22.

The reason for the equivalence of symmetrical correspondence with the vanishing point is straightforward. It relates to the *known length ratio on a line* method of estimating the vanishing point, described in the section 2.4.2.

2.9.4 Rotational Symmetry

The axis of the rotational symmetry is always perpendicular to the translation T in the external parameters of the camera [28]. Let's say the point correspondences for rotational symmetry are related through the so-called fundamental matrix of symmetry F, which can be identified from the image. Let e be the unit length (left) epipole of F and the scalar λ be one of the two non-zero eigenvalues of the matrix $F^\top \hat{e}$. Then the calibration matrix satisfies the normalized Kruppa's equations

$$FKK^\top F^\top \;=\; \lambda^2 \hat{e} K K^\top \hat{e}^\top \tag{2.23}$$

These equations were first proposed in the literature of autocalibration from stereo. Yi Ma et al [28] have realized that symmetry is equivalent to stereo and thus tried to use these equations. However, (2.23) yield non-linear constraints on the parameters of K. This has been a major limitation for their applicability. But, if the camera parameters are all-identified except for the focal-length f (these cases discussed in more detail in the next section), (2.23) yields a linear constraint in f^2

2.9.5 Camera Assumptions

In the previous section, several possibilities of linear constraints on ω are presented. However, there may be cases when the image can not offer 5 such constraints. In such cases, we may rely on placing more assumptions on K, whenever applicable. The following assumptions can be placed in an increasing order of restrictiveness.

Zero Skew : This is valid for CCD cameras. It is not valid when a photographic negative is enlarged and the paper is not parallel to the plane of the negative. From (2.18) and (2.19), the constraint of zero skew translates into

$$\omega_2 = 0 \tag{2.24}$$

Unit Aspect Ratio : This assumption is also valid for CCD cameras. It is not valid for the photographic negative enlargement case. This constraint can be derived from (2.18) and (2.19) as

$$\omega_1 - \omega_3 = 0 \tag{2.25}$$

It may be sometimes the case that the aspect ratio is not unit, but its value known. Principally, both these cases are equivalent in the amount of information they provide, but

in the later case, a different linear equation results.

As mentioned in 2.5, if the camera satisfies the constraints (2.24) and (2.25), it is called as a *natural camera*. Such camera contains 3 internal parameters to be identified. Thus, only 3 constraints of the form (2.20) and (2.22) are required.

Principal Point at Image Origin : This assumption is valid if the image is taken by a CCD camera and it is not digitally tampered with. Specifically, if the image gets cropped, the principal point no longer remains near the image origin.

This highly restrictive assumption provides us with two more constraints on ω. There is only a single parameter (focal length) that needs to estimated for calibrating the camera, which can be done through a single constraint of the form (2.22) or (2.20).

2.9.6 Camera Calibration - Advantages

When the calibration matrix K is known, the stratified plane rectification method discussed in 2.3 becomes more simplified. An affine rectification (identification of vanishing line) automatically yields the metric rectification (identification of circular points).

This is because the IAC ω intersects any vanishing line in the two circular points of the plane. There would be no need to compute them independently. Since the plane gets metric rectified, the relative orientation of the plane with respect to the camera gets known.

Even if the scene is composed of non-planar shapes, a calibrated camera helps much in the reconstruction process. A calibrated camera also simplifies the task of estimating the pose (external parameters of the camera).

2.10 Pose Recovery - External Parameters of the Camera

When the camera is calibrated, a single rectified plane identifies the pose of the camera. This is because the relative orientation of the plane with respect to the camera gets known. And thus, by fixing the world plane, the pose of the camera can be exactly estimated.

In this section, we deal with the problem of estimating the camera pose without first calibrating the camera.

2.10.1 Estimation of Camera Position

The camera position C is related to the camera projection matrix P mentioned in 2.2 as

$$PC = 0 \tag{2.26}$$

If we can obtain a parametrization of the camera matrix P without solving for the internal parameters K, it is possible to obtain the camera position C as the null vector of P. There are cases where obtaining such parametrization would be possible.

This method was described in the work of Criminisi et al [39]. The metric information needed from the image are the vanishing line of a world-plane (l) and the vanishing point of a direction not on the plane (v). Two points are randomly picked on the vanishing line l and are named as $l_1^{\perp}$ and $l_2^{\perp}$. These two points along with v constitute the first 3 columns of the camera projection matrix (which can be assumed to be the vanishing points of the X, Y and Z directions of the world-coordinate system). The final 4^th column should not lie on the vanishing line to preserve the rank-3 of the matrix P. Such point can be identified by normalizing the vanishing line itself. $\mathbf{o} = \mathbf{l}/||\mathbf{l}|| = \hat{\mathbf{l}}$.

The camera projection matrix can thus be parametrized as

$$P = [l_1^{\perp}\ l_2^{\perp}\ \alpha v\ \hat{l}] \tag{2.27}$$

From 2.27 and 2.26, the camera position can be identified uniquely.

2.10.2 Measurements between Parallel Planes

The above camera parametrization 2.27 can be used for finding measurements between parallel planes. Let two points **b** and **t** be identified as the base and the top points on two planes which are parallel and claim the vanishing line **l**.

$$\mathbf{b} = P \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix}, \quad \mathbf{t} = P \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Substituting the value of P from 2.27, we obtain the equations

$$\mathbf{b} = \rho(Xp_1 + Yp_2 + p_4) \tag{2.28}$$

$$\mathbf{t} = \mu(Xp_1 + Yp_2 + Zp_3 + p_4) \tag{2.29}$$

Taking the scalar product of 2.28 with $\hat{\mathbf{l}}$ yields $\rho = \hat{\mathbf{l}}.\mathbf{b}$, and combining this with the third column of 2.27 and 2.29 we obtain

$$\alpha Z = \frac{-||\mathbf{b} \times \mathbf{t}||}{(\hat{\mathbf{l}} \cdot \mathbf{b})||\mathbf{v} \times \mathbf{t}||} \tag{2.30}$$

where α can be estimated from a known reference world distance Z_0.

2.10.3 Measurements on Parallel Planes

If two planes are parallel in the world, there exists a plane-plane homography for their images. This homography is not a generic homography with 8 *d.o.f* but a restricted homography due to the fact that the planes are parallel in the world. Such a homography is called a homology and discussed in more detail in section **??**. In this case, it can be expressed as

$$\tilde{H} = I + \alpha Z \mathbf{v}\hat{\mathbf{l}}^\top \tag{2.31}$$

Using the above equation, points from one plane can be mapped to another parallel plane.

2.10.4 Estimation of the Camera Pose

In none of the above sub-sections, did we manage to estimate the pose of the camera. This is given by the matrix R in the equation (2.3). We were able to recover part of the external camera parameters in the form of T (which is related to C computed in the section 2.10.1).

In fact this recovery is not possible until we completely rectify a plane and fix its orientation in the world coordinate system. This provides the relative orientation of the camera with respect to that world plane, and thus defines R. We can perform the metric rectification of the plane by computing its *circular points* as mentioned in section 2.5. Since the position of the camera C (or T) can be identified using 2.26, this obtains the external parameters of the camera completely.

An alternate way of computing the camera pose is through the use of symmetry groups and is reviewed in the previous chapter (section 1.8.5). If there is a presence of symmetry in the image, the two curves which make for the symmetric counterparts are related by a fundamental matrix. If the camera is calibrated they would be related by an essential matrix. The essential matrix can be decomposed to obtain the relative rotation R_g and translation T_g of the curves. The camera pose (R,T) can be estimated from (R_g, T_g).

2.11 Special Cases of Planar Homography

In this section, we discuss special cases where the plane-plane homography gets restricted to less than the usual 8 *d.o.f.* Such cases arise quite often due to planar symmetry, and are

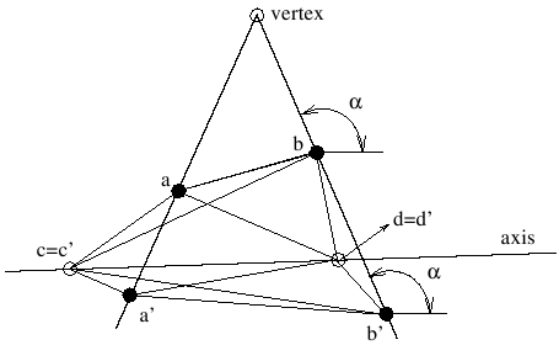


Figure 2.2: Geometric Invariants under Planar Homologies

helpful in performing metric rectification of the planes.

Symmetry is broadly understood through the notion of fixed sets. A symmetry defines a subgroup in the space of plane projective transformations. This subgroup exhibits certain unique properties such as a fixed point, a fixed line etc.

2.11.1 Planar Homologies

As mentioned by Van Gool et al [25], a plane projective transformation is a planar homology if it has a line of fixed points (called the *axis*), together with a fixed point not on the line (called the *vertex*). Algebraically, an equivalent statement is that the 3×3 matrix H, representing the transformation has two equal and one distinct eigenvalues. The *axis* is obtained as the join of the eigenvectors corresponding to the degenerate eigenvalues. The third eigenvector corresponds to the *vertex*. There is another *d.o.f* which is specified by the ratio of the third to the first/second eigenvalue. This crossratio is an invariant specific to the homology. In total, a homology contains 5 *d.o.f* as opposed to the usual 8 *d.o.f* of a generic homography.

Planar homologies can be identified due to the presence of perspectivity in an image. This happens, for example, through the shadows of line-segments due to a point light source.

All the 5 *d.o.f* of a planar homology are over-determined by three pairs of corresponding points, each of which provide 2 constraints. Alternatively, if the fixed point or the line could be identified directly in the image, fewer pairs of corresponding points would suffice.

As illustrated in the figure 2.2, there are two independent geometric invariants that can be observed in a homology. These are

- the angle α as indicated in the figure, between the axis and the line joining the

corresponding points.

- the area ratio for the triangle pairs, also indicated in the figure:

$$\frac{\Delta(a,b,c)}{\Delta(a,b,d)} = \frac{\Delta(\acute{a},\acute{b},\acute{c})}{\Delta(\acute{a},\acute{b},\acute{d})} \tag{2.32}$$

2.11.2 Planar Harmonic Homologies

A harmonic homology is a special kind of homology - which has a unit cross-ratio. That is, all the eigen values of the transformation H are identical. Specifically, the ratio of the areas of the triangles (2.32) becomes equal to 1.

Thus there are only 4 *d.o.f* for specifying this homology. They can be completely identified by specifying two pairs of corresponding points.

These homologies arise when there is a mirror (reflexive) or point symmetry (rotation with angle π) in the image. As discussed by Proesmans et al [23], such cases offer stronger algebraic/geometric invariants.

2.11.3 Elation

An elation is a plane homography with only 4 *d.o.f.* It arises due to the presence of translational symmetry in the world plane. An elation can be completely determined by specifying 2 pairs of corresponding points (along 2 independent directions of translation).

The type of fixed structures that are present in an elation are - a fixed line $\mathbf{l}_\infty$ specified by (2 *d.o.f* - the vanishing line), a fixed point v on $\mathbf{l}_\infty$ (1 *d.o.f*) and the scale of v (1 *d.o.f* which represents the magnitude of the translation).

Similar to mirror and point symmetry [23], this property can be used for geometric grouping [20].

2.12 Conclusion

In this chapter, we have studied in detail the stratified method of metric rectification. We have studied the methods of autocalibration from planes and discussed the importance of calibration for rectification. We have observed the relative hardness of the steps of obtaining the vanishing line and the circular points. We have looked at equivalent ways of obtaining the same information through several fixed structures.

Chapter 3

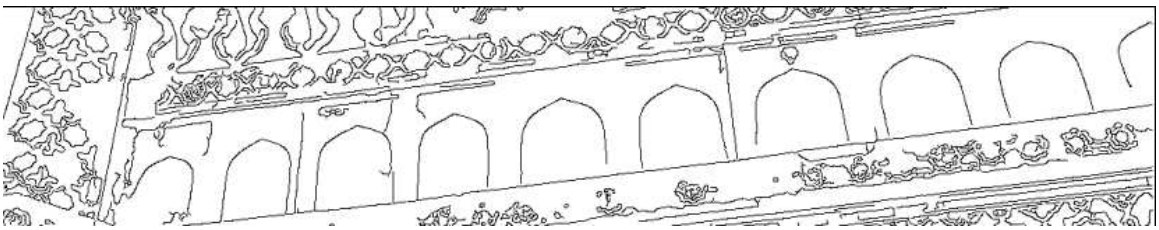
Interest Point Detection using Macro-Shape Properties

3.1 Interest Points of a Pattern

We proceed by a hypothesis that a pattern is completely determined by a set of a few interest points. That is, we assume that any form of repetition of the pattern could be detected solely by observing the interest points of the pattern. This is a valid assumption for most of the patterns that occur in the real world.

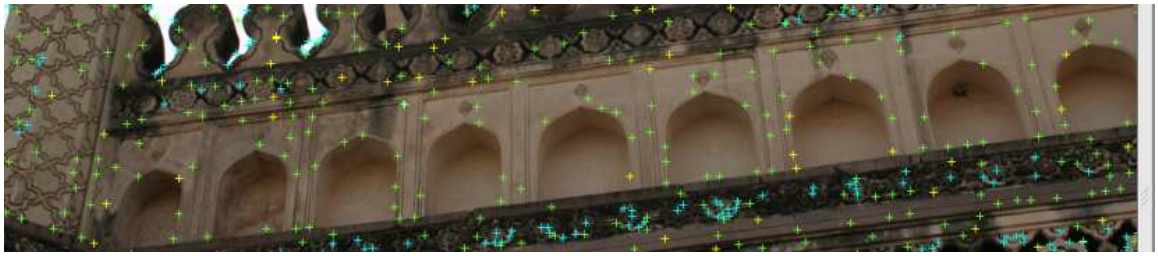
We are not concerned with the problem of how to generate the pattern given the set of interest points. Instead, we pay attention to the problem of how to use the interest points to do geometric grouping. Also, we do not hold any assumption that the interest points which belong to each pattern are strictly segmented and separated out. That would be an impractical assumption for many cases. Our algorithm takes a holistic set of candidate interest points which occur in the image and then proceeds to group them using geometric constraints. This geometric grouping, which will be discussed in Chapter 4, segments the *interest point set* into patterns. In the current chapter, we discuss how to generate the *interest point set* to begin with. We present several methods for doing this. We first discuss the classical appearance based methods. However, these methods are not completely effective, as will be demonstrated in certain examples. We then present a novel method for generating interest points which is based on geometric saliency.

Since we are interested in using these interest points for geometric grouping in a perspective view, we require these points to be robust against the operation of perspective projection. Specifically, our algorithm should continue to identify an interest point even when the image gets taken from a different perspective.



(a)

(b)



(c)

Figure 3.1: Different stages of interest point detection(a) Result of canny edge detection (b) Result of primitive contour detection (c) Interest points detected by geometry based saliency (*green*) and appearance based saliency (*blue*)

3.2 Properties Preserved by Perspective Distortion

When a pattern gets distorted in a perspective view, it is hard to identify all the points which belong to it. However, some properties are preserved even in a perspective view, and it is imperative for us to identify those points which are preserved by these properties.

Collinearity A perspective projection is also called a *collineation* because the *collinearity of three points* is preserved.

Concurrence Conjugately, if three lines are concurrent, they remain concurrent in the perspective view. The *point of concurrence* becomes an important interest point that needs to be detected. If two intersecting line segments can be properly identified in both the views, the *point of intersection* remains preserved for the same reason.

Inflection Similarly, *points of inflection* are also preserved by a perspective transformation. If it is possible to identify the points of inflection on the discrete curve, it makes sense to use them as interest points. However, as we shall discuss in later sections, the discrete curvature can often be very unstable, and thus misleading.

Colour Information The color information of the image does not change irrespective of the perspective projection. So, if it is possible to robustly detect interest points through appearance based information, it should be done by all means. However, it should be noted that this method is not a silver bullet for identifying symmetry in a single view, since there could be several cases where the appearance is not uniform across two patterns.

3.3 Appearance-based Analysis

In this section, we discuss the approach of using the appearance information in terms of colour and intensity values to detect interest points. So far, this has been the popular approach for several tasks in computer vision. The interest points are identified in two steps.

3.3.1 Canny Edge Detection

Canny's edge detector works in a multi-state process. It first smooths the image by convolution with a Gaussian kernel. This removes spurious noise to some extent. Then a discrete 2D derivative operator is applied to the smoothened image to highlight regions with high first order spatial derivatives. In the resultant gradient magnitude image, edges give rise to ridges. The algorithm then tracks along the top of these ridges and performs *non-maximal*

suppression, that is setting to zero all pixels which are not exactly on the top of a ridge.

This produces a thin line per each edge in the output. The tracking process exhibits hysteresis which can be controlled by two thresholds : T_1 and T_2 with $T_1 > T_2$. Tracking can begin only at a point on a ridge higher than T_1. However, once begun, tracking continues in both directions out from that point until the height of the ridge falls below T_2. This hysteresis helps to ensure that noisy edges are not broken up into multiple edge fragments.

There are three parameters that need to be specified to the Canny operator - the variance of the Gaussian kernel σ_C and the thresholds T_1 and T_2.

3.3.2 Shi's Good Features to Track

We now discuss the problem of detecting corners. A corner is defined as a point of intersection of two contours in the gradient image. A pixel which notices sharp falls in the gradients in more than one direction is defined to be a corner. However not all corners are equal. The goodness of a corner can be measured through the eigen values of its spatial neighborhood in the difference image. Pixels with high eigen values make for good corners.

Shi and Tomasi [35] discuss the viability of using corners for the problem of multi view tracking. They claim that corners make for good features since they can be matched easily across images. Also, since a single image of a symmetric object is equivalent to multiple images taken from different perspectives, the logic of [35] holds good even for the case of finding point correspondences in a symmetric object.

Shi et al [35] operate a threshold on the minimum eigen value of the corner for eliminating some of the candidates. They also have methods of *non-maximal suppression*. During the process of detection, they discard the corners which are not maximal in their local neighborhood. Further, they have a higher level process which discards corners which are too near to other corners. Only the corners with the maximal eigen value are kept in the case of proximity. In practice, we have found that this method identifies most of the corners and inflections that are present in symmetric objects.

In our system, we employ this method for detecting interest points thorough appearance. We use the utility function *cvGoodFeaturesToTrack()* from the OpenCV library to implement this method. It has two parameters - *quality level* of the eigen value and *minimum distance* to other corners, which are tuned by our system.

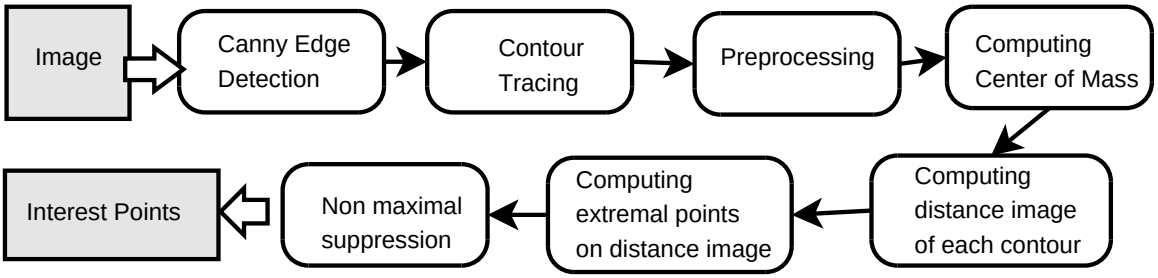


Figure 3.2: Overview of the Process of Detecting Interest Points through Geometric Saliency

3.4 Geometry based Analysis

From now on, we present an alternative method for analyzing the saliency of interest points. This method is purely geometric and we do not refer to the appearance information at all, except for constructing the image contours which is done by observing the edge image.

An overview of our algorithm is shown in the figure 3.2.

3.4.1 Automatic Contour Detection

Contour detection proceeds from the output of the Canny's edge detection algorithm. In spite of the hysteresis, Canny's algorithm fails to find many contours in the image. This is because, each contour gets broken down into *edgels* which are separated by large gaps. These edgels need to be linked using a higher level algorithm.

This is performed using the OpenCV utility function *cvFindContours()* This function has an incentive to find longer and closed contours.

3.4.2 Preprocessing

Some of the contours that are detected are too small to be useful. These contours are usually the result of an improper edge detection step. In the preprocessing stage, we prune several of these contours by looking at their shape properties.

Contour Length We apply a threshold on the minimum length of a contour and discard the contours which are smaller than this. For images with a resolution of 1200×800 pixels, we have discovered that applying a threshold of 20 pixels to the minimum contour length achieves good results.

Bounding Box Area We compute the minimum area rectangle which encloses the contour - this rectangle need not be parallel to the image axes. We employ the utility function *minAreaRect2()* of the OpenCV library to obtain this. This rectangle is termed as the bounding box and is an indicator of the extent of the contour. We apply a threshold on the minimum area for this bounding box, and discard smaller contours. For images with a resolution of 1200×800 pixels, we have discovered that applying a threshold of 200 pixels for the bounding box area achieves good results.

3.4.3 Limitations of Contour Detection

Since the approach is a simplistic process and does not use any high-level information, it is prone to several kinds of errors. These can be observed in the figures 3.4 3.5.

Wiggly Contours The contours tend to become too wiggly instead of smooth. This happens due to the merging of several small foreign edges into the contour from the neighborhood.

Fragmented Contours The contours tend to become fragmented at several places. This happens when there is a big gap between the edges of the contour.

Merged Contours In some cases, several large independent contours get merged into a single compound contour. As we shall see later on, this problem presents the most acute limitation for our method.

In the later sections, we present our methods for detecting interest points on this set of poorly defined image contours.

3.5 Micro-Shape Properties

By micro-shape properties, we mean those properties which deal with the local shape information at each point. This information is usually about the differential properties of the curve. It is a well-known fact that derivatives are not preserved under perspective distortion. However, some properties which can be computed through derivatives, such as inflection points, will be preserved. However, this approach is still prone to several errors because the derivatives tend to be unstable. Moreover, as observed in the previous section, the detected contours are not perfect. This will compound the problems when derivatives are used.

3.5.1 Discrete Curvature

Let the point x_t be located in between the points x_s and x_u. Let the directions of the tangents at these points be given by θ_t, θ_s and θ_u respectively. The discrete curvature at the point x_t is defined as the following

$$\kappa = sin(\frac{\theta_t - \theta_s}{2}) + sin(\frac{\theta_u - \theta_t}{2}) \tag{3.1}$$

We would like to identify points with a high curvature.

3.5.2 Derivatives on Spline Approximation

The discrete contour points can be approximated using an interpolating spline [40]. Instead of using the discrete curvature, the curvature at a point can be computed through analytical means from the approximated spline. This derivative tends to be more robust, primarily due to the aggregation of information from a larger local neighborhood.

The micro-shape properties fail completely when the contours are detected improperly. This is because the local shape information is no longer reliable. In such cases, the appearance based methods (Shi-Tomasi algorithm) tend to produce better results. In the next section, we discuss a method of geometric saliency which overcomes this limitation.

3.6 Macro-Shape Properties

By macro-shape properties, we mean those properties which deal with the shape information at a global level. Each point on the shape is analyzed in the context of all the other points on the shape, and not just with respect to its local neighborhood. This approach is useful when the local shape information is not preserved properly, as in the case of fragmented contours.

However, many global shape properties are not preserved under perspective projection. However, we shall demonstrate that a few properties are affected only slightly. These properties can then be used for the detection of interest points.

3.6.1 Center of Mass

The center of mass is defined as the average of the positions of all the peripheral points on the curve. In a discrete setting, this point is subject to the quantization errors that happen on the image grid. The center of mass is not preserved properly under perspective distortion. If the curve is fragmented, the center of mass for each fragment may vary from the center of mass of the complete curve. Due to these problems, it is not a good idea to consider the center of mass as an interest point.

3.6.2 Distance Image

The Euclidean distance from the center of mass is computed for each peripheral point and these distances are encoded as the distance image of the contour.The distance image is a succinct representation of the global shape properties, because the distance-image value for each point is dependent on the positions of all the contour points. The distance image is not preserved under perspective distortion. So, it cannot be used directly.

3.6.3 Extremal Points

We define a local maximum or minimum in the distance image as an extremal point.. An extremal point signifies a bend in the direction of the contour, and thus it is a point of intersection for two contour-segments, which should be preserved under a collineation. A local maximum in the distance image represents a peak of *convexity* of a shape, where as a local minimum represents the depth of a *concavity*. Both these points are eligible for consideration as interest points.

To compute the extremal points, we first smooth the distance image using a Guassian kernel. This reduces the effects of noise. We then process the distance-image value of each point and output the local maxima/minima.

3.6.4 Points of Intersection

Under a collineation, the point of intersection is preserved exactly. Thus, if the point of intersection of two contours can be identified robustly, it can be considered for use as an interest point.

3.7 Effect of Perspective Distortion

Since the center of mass is not preserved under perspective distortion, the distance-image values would get corrupted. However, the extremal points get distorted only slightly.

An extremal point χ of a contour becomes lost if the distance image is computed from a point on the tangential direction of χ.

For a closed contour, this does not happen as long as the center of the mass remains sufficiently to the interior of the contour. For an open contour, the effects of perspective distortion are more dangerous as the center of mass moves closer in proximity to the contour points. Some extremal points may become completely lost where as some points get shifted by a small amount.

However, as our experiments in section 3.9 shall demonstrate, the interest point detection remains quite robust. The geometric grouping algorithms that will be discussed in the next chapter have efficient outlier rejection mechanisms, which shall eliminate some of the false positives amongst the interest points. Also, these algorithms handle cases where the interest points are not detected exactly. Thus, the damage done by perspective distortion gets rectified.

3.8 Effect of Contour Fragmentation

A more serious problem for interest point detection is the fragmentation of contours. As we have discussed in section 3.4.3, they pose serious problems for the detection of interest points.

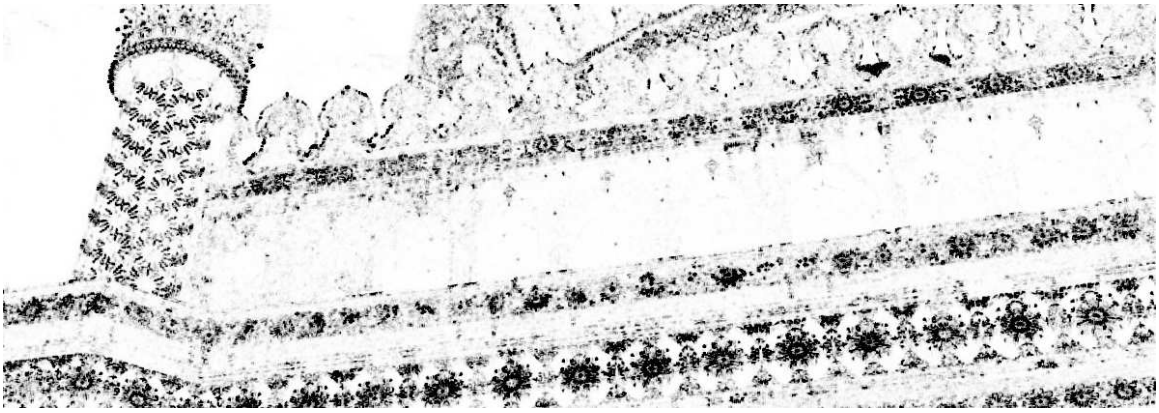
Fragmented Contours : As long as the contour is not fragmented into a series of line-segments, the interest point detection remains quite robust. For instance, when a large closed contour gets fragmented, the new centers of mass usually stay sufficiently away from the tangential directions of the two open contours. Thus, the previous set of interest points will still be identified without any errors.

Merged Contours : When two large image contours get merged into one, the center of mass gets shifted dangerously away from its previous locations. If the new location appears on the tangential directions of several previous interest points, they will be lost. However, this does not happen very often.

The results of the algorithm are demonstrated in section **??**.

3.9 Results

We have conducted experiments on two types of datasets - architectural scenes and patterns on cloth. Our intention is to demonstrate the limitations of the classical appearance based methods in identifying the entire set of interest points. The points marked by light-blue crosses are the points detected using the Shi-Tomasi method. The points marked by green crosses are the points detected as peaks of convexities on the contour. The points marked by yellow crosses are the ones detected as concavities. Additional images are present to show the output of the canny algorithm and the contour detection algorithm respectively.



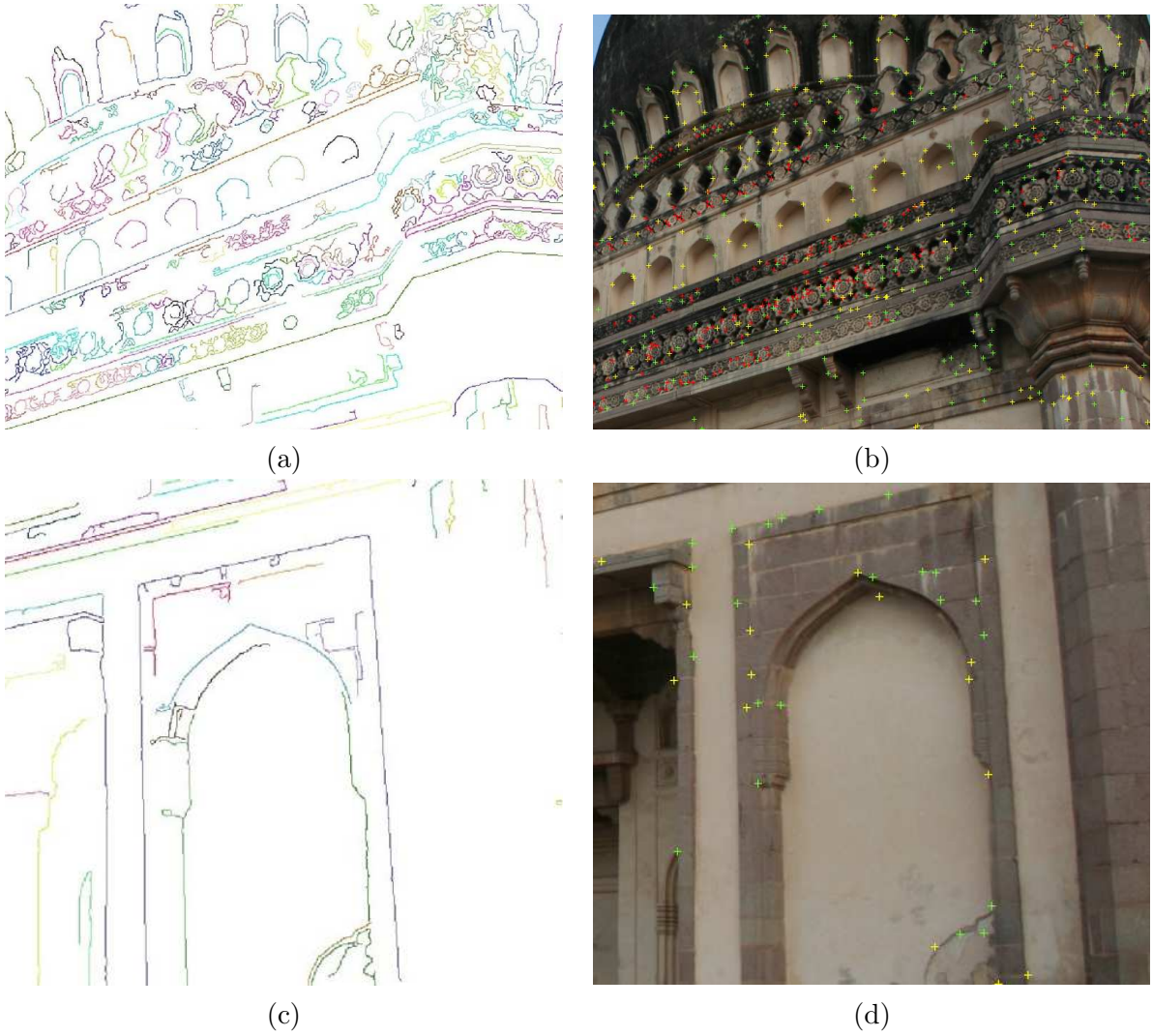
(a)

(b)

Figure 3.3: Image showing the minimum eigen values for the gradient image of an architectural scene - no features can be seen along the arches

3.9.1 Architectural Scenes

We have taken several pictures of the Qutb-Shahi tombs in Hyderabad. These tombs have been constructed by the Qutb-Shahi rulers - medieval emperors of the Deccan region of India. As is evident in the pictures, the feature points of the patterns have little texture information. So appearance based methods of feature point detection do not yield good results. The Shi-Tomasi method selects corners based on the eigen values of the gradient information in the local neighborhood. The minimum eigen value for each pixel is presented in the figure 3.3. It can be seen that the minimum eigen value comes out as very low (undetected) for several feature points in this image.



(a) (b)

(c) (d)

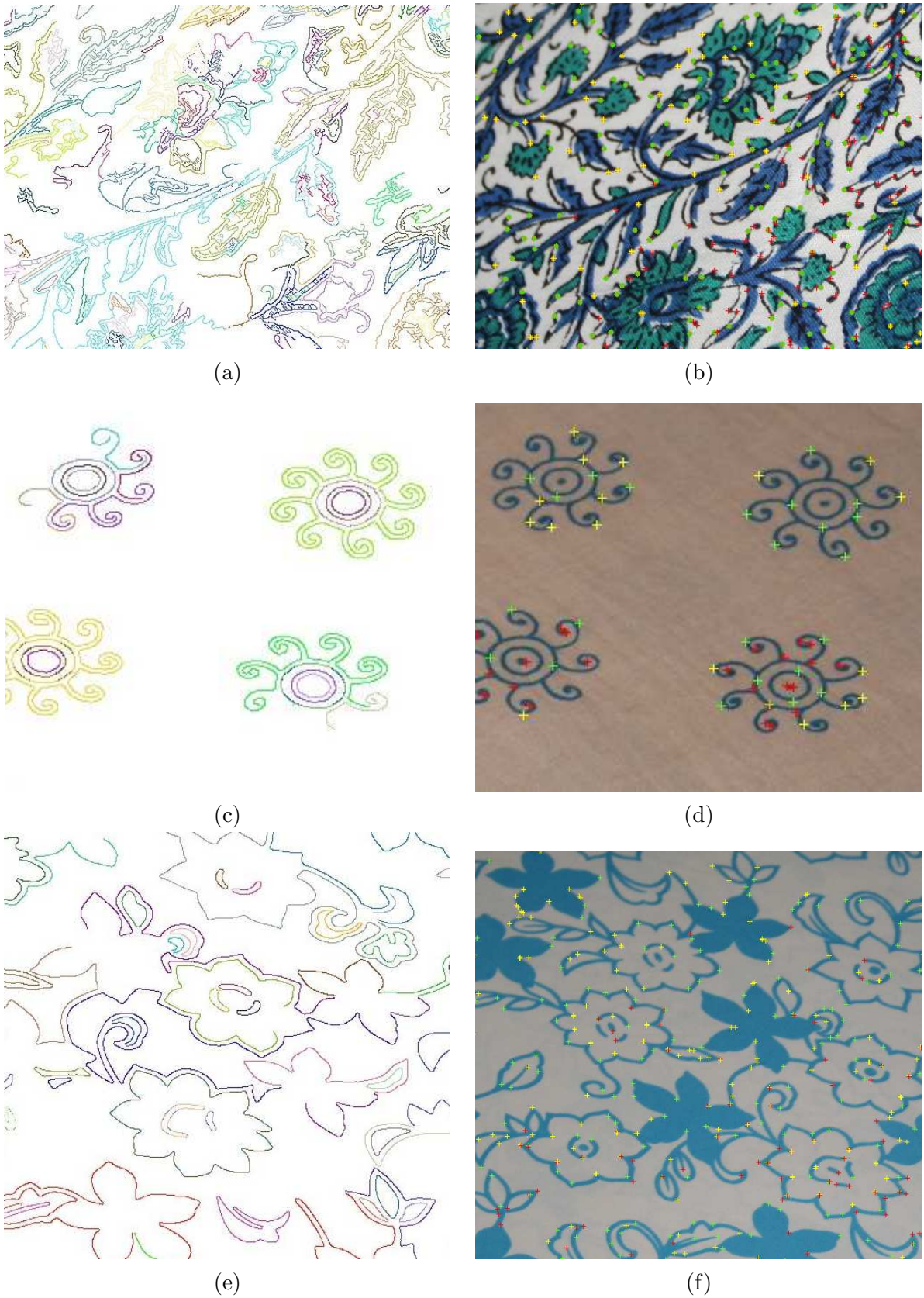
Figure 3.4: Results of Interest Point Detection on Architectural Scenes : (a)(c) detected image contours, (b)(d) detected interest points. Points in green show peaks of convexities, points in yellow show depths of concavities, points in red are detected by the Shi-Tomasi algorithm, and are not replaced. It can be seen that several new and useful features are identified as points in green and yellow.

3.9.2 Designs on Cloth

Several pictures have been taken for design patterns on cloth. These patterns are highly symmetric and usually lack any line-segments that can be directly identified. It can be seen here that several important points that are missed by the appearance based detector are identified by the geometry based detector.

3.10 Conclusion

In this chapter, we have presented a novel approach for interest point detection which is based on geometric saliency. This approach is specifically useful when the texture information is minimal or missing completely. In this chapter, we adopt a hungry approach for detecting feature points and thus liberally accept many outliers. These outliers are later removed by the geometric grouping algorithm which will be discussed in the next chapter.



(a) (b) (c) (d) (e) (f)

Figure 3.5: Results of interest point detection for perspective images of design patterns on cloth : (a)(c) & (e) detected image contours, (b)(d) & (f) detected interest points. Points in green show peaks of convexities, points in yellow show depths of concavities, points in red are detected by the Shi-Tomasi algorithm, and are not replaced

Chapter 4

Geometric Grouping using Spatial Coherence

In this chapter, we discuss the main contribution of the thesis - a method of geometric grouping of patterns through the assumption of spatial coherence. This approach is different from the past approaches which have used highly constraining grouping mechanisms based on geometric invariants. Instead, this approach adopts a novel form of *smoothness constraint*, which says that the local neighborhoods of the matched points should be similar. Unlike approaches such as stereo, the local neighborhood is not constrained to the 8-neighbors of the pixels but instead, is representative of a much larger area.

It is assumed that the patterns are defined solely in terms of a set of interest points, and this chapter presents algorithms which exploit the geometric positions of the interest points. No other information, such as color or appearance, is used in the algorithms presented here.

Geometric constraints and invariants are used to justify the matching between the patterns. In what follows, we use the term *geometric structure* to denote structures such as the vanishing point, vanishing line or the circular points. The grouping of the patterns would be performed through *geometric structure consensus*.

4.1 Overview of the Algorithm

A generic version of the algorithm is shown in figure (4.1). The input is given in terms of specifying two sets of interest points. It is assumed that each set of points belongs to a pattern. These points could be detected using the methods discussed in Chapter 3. The current task is to uniquely match points in the first set (called the *left set*) with those in the second set (called the *right set*). There could be several outliers due to improper detection of interest points and improper specification of the boundaries of the two sets.

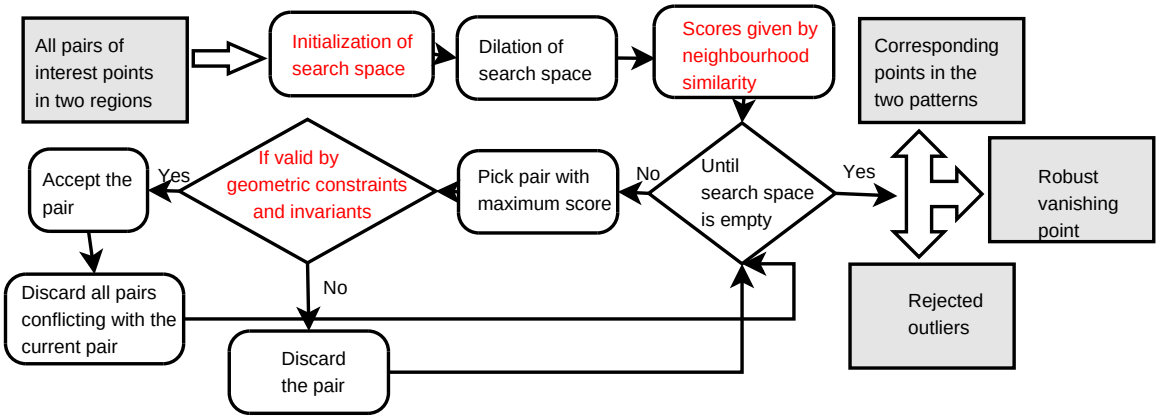


Figure 4.1: Overview of the generic greedy algorithm for matching planar patterns

The algorithm consists of three main stages. The first stage is the initialization stage where the search space is defined though a heuristic. Instead of all the possible pairs between the left and the right sets, only a subset of them are considered. For example, it is possible to constrain the slope of the line joining the left and the right points to be near to a given value. The different forms of initialization are discussed in section 4.2. After the initialization, a crude value of the geometric structure (for example, the vanishing point) can be computed. This value is used to dilate the search space to include other pairs in the marginal neighborhood.

The second stage of the algorithm deals with assigning scores to each pairing of points. The method of scoring exploits principles of spatial coherence and will be discussed in section 4.3.

The third stage of the algorithm is a greedy optimization approach for matching points in the left-set with points in the right-set. The point-pairs are considered in a descending order of the scores which were assigned in the previous stage. Only the pairs which are in agreement with the geometric structure are accepted. Other are discarded. The search space is pruned so that future acceptances do not conflict with the current pair in terms of spatial coherence. This stage is discussed in section 4.4.

There is an optional fourth stage of the algorithm which generalizes the results found across two patterns to an entire stretch of region. This method of generalization is described in section 4.5.

All the stages of the algorithm are demonstrated on a sample image in the figure 4.2.

In this chapter, we adopt a combinatorial approach for discovering point matches. We shall demonstrate in the results section (4.7), that this approach is fast and efficient.

4.2 Initialization

Since there exists a positional ordering in the world plane, several putative pairs can be eliminated from the sample space though the use of simple heuristics. Since we deal with a combinatorial optimization approach for point matching, this stage pays large dividends in the reduction of the search space. This stage is termed as the *initialization* of the algorithm.

The input required for this initialization can be in terms of four forms

1. Specifying the boundaries of the left point set or the right point set
2. Specifying an initial match between two points
3. Specifying the approximate direction of match between the two patterns
4. Specifying the geometric structure approximately

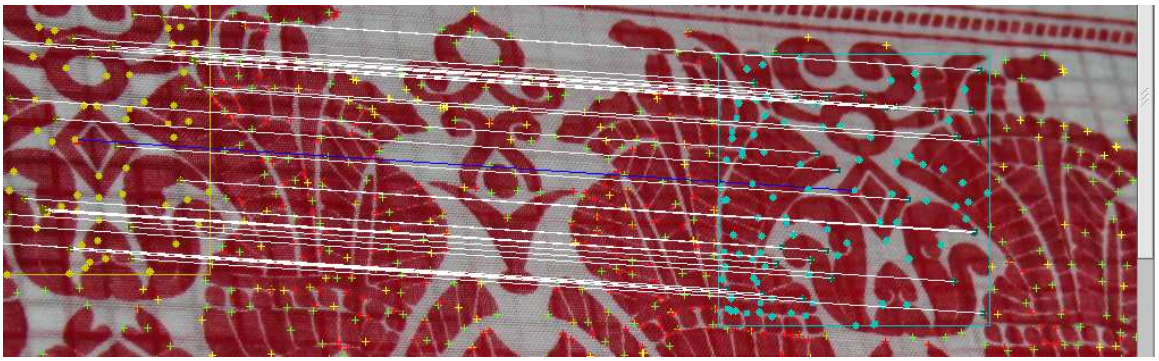
One method of obtaining this required input is through user interaction. We discuss these issues elaborately in chapter 5 from the point of view of a friendly user interface. However, there could be other forms of obtaining this information.

For example, the approximate direction of match between the patterns can be obtained through Hough-transform based analysis. The limits of the left and the right point sets can be set through an intelligent segmentation algorithm. Sometimes, it is also possible to match a few feature points robustly, through the use of appearance based methods (SIFT features). We separate this type of processing from the geometric grouping algorithm because of the combinatorial nature of the optimization.

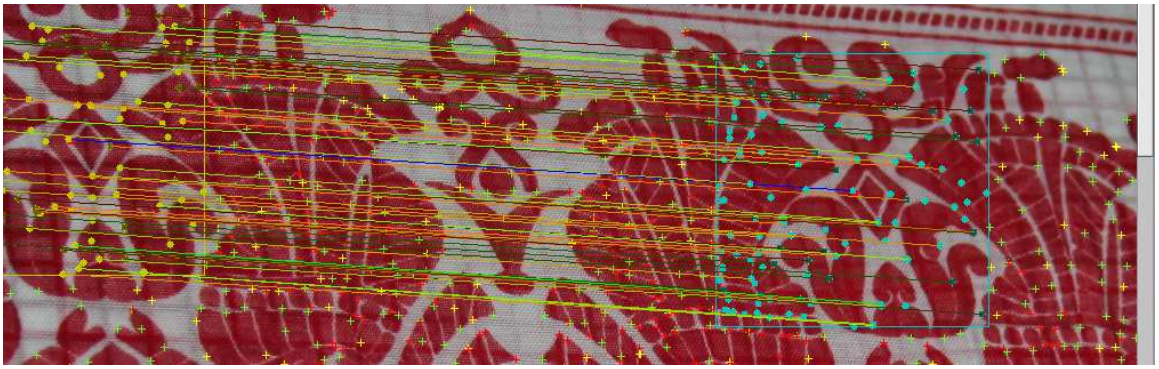
In the following subsections, we discuss how to obtain the search space based on this initialization.

4.2.1 Initialization by Pairing Two Feature Points

Let the two points be χ_l and χ_r. The approximate direction of the match is obtained as the slope μ of the line joining these two points. The search space is initially set to include all possible pairs between the left and the right sets. It is later pruned to obtain the pairs

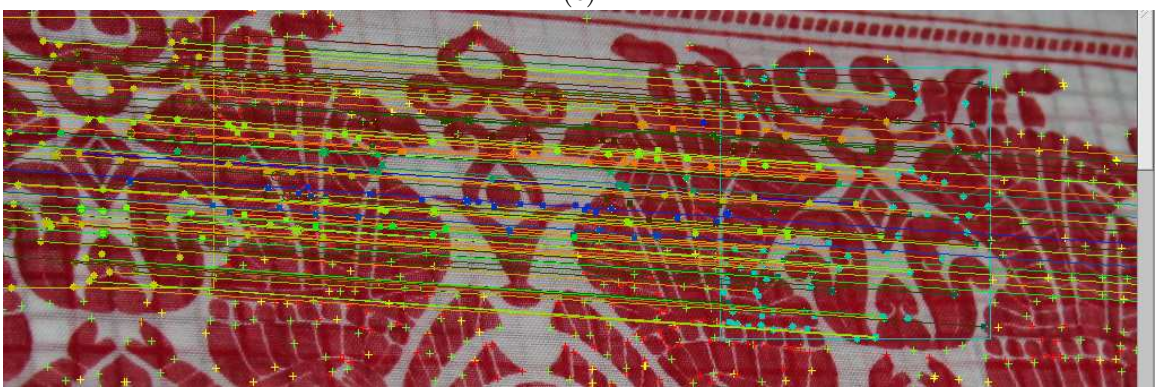


(a)



(b)

(c)



(d)

Figure 4.2: Overview of several stages in the greedy matching algorithm for vanishing point detection : (a) The search space selected through initialization by a sample direction (b) Resulting parallel lines obtained through the greedy algorithm (c) Point correspondences across the two patterns (d) Generalization of the result across an entire stretch

of points which have a slope near to μ within a threshold θ_μ. Also, since the validity of the two given points is known to be absolute, the pairs of points are constrained to be spatially coherent with (χ_l, χ_r). The forms of spatial coherency will be discussed in section 4.3.

4.2.2 Adaptive Threshold on the Direction of Match

Instead of being assigned a fixed value, the threshold θ_μ is selected adaptively by observing the sample space. If the left set contains N_l elements and the right set contains N_r elements, one can obtain $Max(N_l, N_r)$ pairs of correspondences at maximum. This presents a constraint on the threshold θ_μ.

In the initial stages of the grouping, we would like to be very liberal in considering sample pairs. So the threshold θ_μ is set to a value such that the sample space contains at least $1.2 * Max(N_l, N_r)$ elements.

Due to the adaptive threshold, this method of initialization can handle several forms of perspective distortion.

4.2.3 Initialization by Specifying the Geometric Structure

Instead of selecting two feature points, the initialization can be done through specifying the approximate location of the geometric structure (for example, the vanishing point). This type of initialization serves the same purpose, though sometimes it may be more intuitive for the user to provide. However, with this kind of input, the algorithm cannot employ spatial coherence at the stage of initialization.

4.3 Spatial Coherence

We intend to match patterns under perspective distortion. Similar to the Scale Invariant Feature Transform (SIFT), we try to preserve angles in the local neighborhood. The angles are conserved exactly when the distortion is only a similarity (rotation and scaling). Under affine or perspective distortion, the angels will no longer be preserved. However, they can still be employed for matching features. Euclidean properties such as the length of a line segment suffer higher levels of distortion in a perspective view. If these are used for matching patterns, they have to be used with a more liberal threshold.

We define the local neighborhood of a point χ in terms of a set of interest points that are present in a square region around χ. The size of the local neighborhood can be adjusted based on the type of images. For images with a resolution of 1200×800 pixels, the local neighborhood is chosen to be 40 pixels wide.

The neighboring points η are defined in terms of vectors $\overrightarrow{\eta\chi}$ connecting to the candidate interest point χ. Both the direction and magnitude of these neighborhood vectors are used for matching two interest points.

4.3.1 Assignment of Scores

A score is assigned to a match between two interest points based on how well the neighborhoods align with each other. Let the point χ_l be considered for match with another point χ_r. For each point η_l in the left neighborhood, the algorithm attempts to select a point in the right neighborhood η_r such that the following two thresholds are satisfied.

$$\frac{length(\overrightarrow{\eta_l\chi_l})}{length(\overrightarrow{\eta_r\chi_r})} < \alpha_\lambda \tag{4.1}$$

$$abs(angle(\overrightarrow{\eta_l\chi_l}) - angle(\overrightarrow{\eta_r\chi_r})) < \alpha_\theta \tag{4.2}$$

where α_λ is a threshold on the ratio of lengths and α_θ is a threshold on the difference of angles. In our system, we have chosen α_λ to be 1.3 and α_θ to be $\frac{\pi}{10}$.

We count the number of left neighbors that could be successfully paired with the right neighbors and vice versa. Then, we assign a score for each putative match between interest points χ_l, χ_r.

$$Score_{l,r} = \frac{numMatched(\chi_l)}{numNeighbours(\chi_l)} + \frac{numMatched(\chi_r)}{numNeighbours(\chi_r)} \tag{4.3}$$

4.3.2 Checking for validity of positional ordering

Even though Euclidean properties get distorted in a perspective view, the relative ordering of their positions remains unaffected. When a new pair of points (X_l, X_r) is being considered for matching, it has to be checked if the new pair has any conflict with any pair of points (χ_l, χ_r) accepted previously. We perform this test by multiplying the coefficients of the difference vectors.

$$order_x = (X_l.x - \chi_l.x) * (X_r.x - \chi_r.x) \tag{4.4}$$

$$order_y = (X_l.y - \chi_l.y) * (X_r.y - \chi_r.y) \tag{4.5}$$

For translational symmetry (tiling of patterns), we accept the new pair (X_l, X_r) as valid only if both the values of $order_x$ and $order_y$ are positive. The same test is also used in the case of rotational symmetry. However, in the case of bilateral symmetry (patterns reflected

over mirror), we adopt the opposite test. If we know that the axis of symmetry is very much along the y-axis, we check if $order_x$ is negative and if $order_y$ is positive. If the axis of symmetry is tilted towards the x-axis instead, we adopt the opposite test.

This test is used at the initialization stage to accept the pairs which are spatially coherent with the provided input. It is also used in the optimization stage, to be described in the section 4.4.

4.4 Greedy Optimization

In discrete combinatorial optimization, a greedy algorithm yields a local maximum. It is important to avoid taking wrong steps in the initial stages of the optimization. In the current scenario, it is imperative not to accept a wrong pair of points, which may later conflict with the true matches in terms of spatial coherence. Even though this is a severe limitation, we would see that this this can be overcome through the proper use of heuristics, and that the greedy algorithm usually yields valid solutions. This is because of the property of the preservation of positional ordering in projective spaces.

4.4.1 Heuristics to Avoid Wrong Matches

Spatial coherence is encoded *apriori* into an array of scores for each candidate pair of points. These scores are assigned based on the similarity of the local spatial neighborhood. Further, if a pair of points is fixed by the user, and its validity is known to be absolute, the greedy algorithm may first look for matches around the local neighborhood of the given points.

That is the scores are to be modified in the following manner

$$Score_{l,r} = Score_{l,r} * \frac{length(\overrightarrow{X_l\chi_l}) + length(\overrightarrow{X_r\chi_r})}{2} \tag{4.6}$$

This forces the greedy algorithm to first consider matching nearby points, and thus prevents it from making major errors in the beginning.

4.4.2 Vanishing Point Consensus

The validity of each pair is checked by properly utilizing the geometric structure which relates the patterns together. For example, if the two patterns are related by tiling or reflection, they should yield a common vanishing point. The greedy matching algorithm can be considered as a search for the optimal position of the vanishing point. An initial value for this is provided by the methods discussed in section 4.2. The value is updated as more

evidence is presented in the course of optimization.

A unique vanishing point can be computed by two pairs of corresponding points. Each pair of points yields a homogeneous vector for the equation of the line L_i. These vectors are stacked together to form the matrix A and the vanishing point V is computed as the null vector of this matrix through SVD.

$$A = [L_1 L_2 \dots L_n] \tag{4.7}$$

For each new pair of points (X_l, X_r), a line L_l is computed by joining the vanishing point V with the point X_l.

$$L_l = v \times X_l \tag{4.8}$$

The distance between X_r and its perpendicular projection on the line L_l is considered as a measure of agreement with the vanishing point. If this value is high, then the pair (X_l, X_r) is termed as invalid and discarded.

In case of a valid pair, it is accepted. All the pairs inside the sample space are checked for validity in spatial ordering with respect to the new pair. If conflicts are found, the corresponding pairs are discarded as invalid.

4.4.3 Vanishing Line Consensus

If two patterns are related by translational or rotational symmetry, the point correspondences yield more than one vanishing point. This is because the lines joining $X_l \rightarrow \chi_l$ and $X_r \rightarrow \chi_r$ are also parallel in the real world. A new vanishing point can then be computed as following.

$$v_x = (X_l \rightarrow \chi_l) \times (X_r \rightarrow \chi_r) \tag{4.9}$$

These new vanishing points can be grouped together to obtain the vanishing line of the plane. Since there will be more than two vanishing points v_i available, the vanishing line is computed using PCA - as the null vector of the following matrix.

$$A_v = [v_1 v_2 \dots v_n] \tag{4.10}$$

The evidence for each of these vanishing points need not be uniform. Specifically, the main vanishing point which relates $\forall X_l \rightarrow X_r$ has much higher evidence and is thus less prone to error. So, each vanishing point is weighted by the number of points that are supporting it, and the vanishing line is finally computed by weighted PCA.

To check for the validity of a new pair (X_l, X_r), the distance between the new vanishing point (4.9) and its perpendicular projection on the vanishing line is computed. If this distance is less than a threshold, the new pair is accepted. Otherwise, it is rejected as being invalid.

4.4.4 Fixed Point Consensus

Instead of a vanishing point, the two patterns might share another geometric structure in common. For example, in point symmetry, it is the point of concurrence for the lines joining the corresponding points. This fixed point is similar to the vanishing point and is employed in a similar manner by the greedy algorithm. Another example is when the planar patterns are related by a perspectivity - a pattern and its shadow. In this case, the fixed point becomes the position of the light source. The greedy optimization approach can be employed without any modification.

4.4.5 Outlier Removal

The greedy algorithm proceeds until all the pairs have been accounted for. The algorithm terminates by providing a robust solution for the geometric structure along with a set of point correspondences across the two patterns. Some of the points in the *left* and the *right* point sets shall not be matched with any point. This happens because they could not produce a match which is valid according to the geometric structure and which is spatially coherent with the other matches. This can happen because of three reasons.

1. The corresponding interest point is not detected in the other pattern
2. The pair could not be matched because of an error in the greedy optimization
3. The interest point is detected due to an error.

In the third case, specifically, the interest point needs to be removed. However, we treat all the three cases uniformly, and remove all these interest points as outliers. Thus, this method of geometric grouping provides a method for improving the interest point detector described in Chapter 3.

4.5 Generalization of Results

In several cases, the pattern of consideration occurs in more than two repetitions. In such cases, it is simple to generalize the solution obtained above to the entire stretch of points. An overview of this process is presented in figure 4.3.

4.5.1 Generation of *optlines*

We take each pair that has been matched successfully by the greedy algorithm and join the points together to obtain an *optline*. These *optlines* intersect with each other at the principal vanishing point. The area of the region containing further repetitions of the pattern is roughly identified. Specifically, if there are further repetitions in between the left and the right point sets identified before, it is trivial to identify this area. We consider all

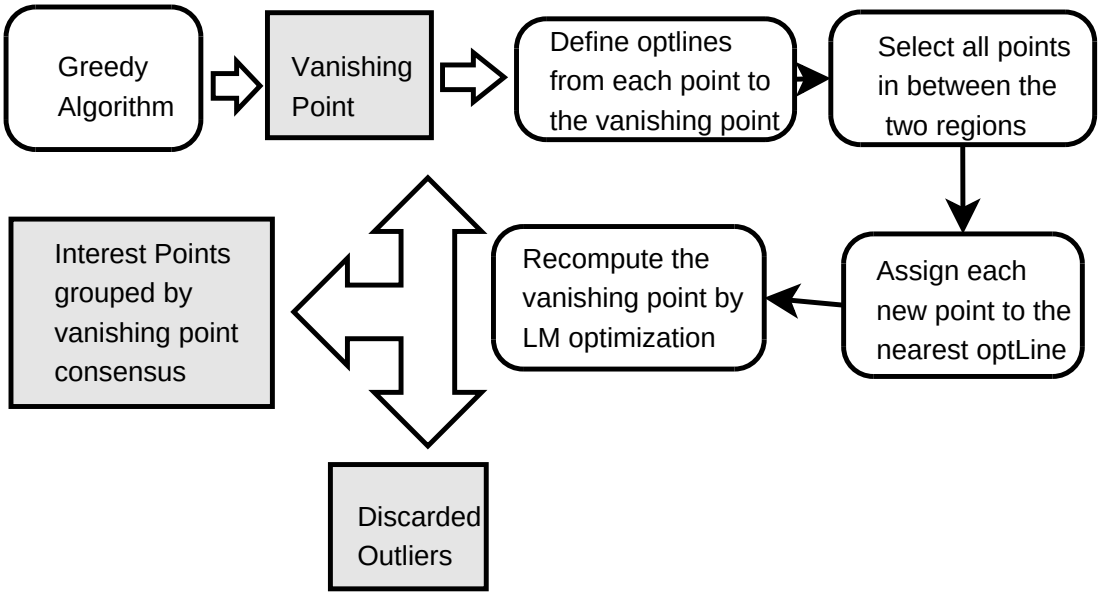


Figure 4.3: Generalization of the solution obtained by the greedy algorithm

the interest points inside the region of interest and compute the nearest *optline* to each one of them. The distance of each point to its perpendicular projection on the *optline* is taken as a measure of nearness. If this distance falls within a threshold, we consider the new point as legitimately belonging to the *optline*. Even though the interest points are not detected exactly, this threshold helps in grouping them together.

4.5.2 New Candidate Lines

After assignment of several interest points to each optline, there would still be several points left out. This may be due to the lack of corresponding points in the left and the right point sets. Each such point C_{a0} is assigned a new optline L_a. If the new optline L_a is detected to pass through points other than C_{a0}, then it is considered a valid optline and added to the set. Otherwise, it is discarded along with the point C_{a0}. This is the method for the detection and handling of outliers.

4.5.3 Optimization

We have a set of optlines $\{L_a\}$ each of which contains a set of points $\{C_{ja}\}$ We would like to obtain the point of concurrence of these lines V. However, due to the redundancy of data, we shall not obtain a unique point of concurrence. It has been argued by [41] that averaging the points of intersection does not yield the optimal result. Instead, a new set of optlines have to be computed which are (i) exactly incident on the point of concurrence

and (ii) as closely resembling the original lines as possible.

It has been discussed in [41] that the error from the model lines to the original lines can be considered as a sum of the perpendicular distances of the endpoints over which the line segments have been detected. This error is only an approximation in their case because they detect line segments using methods other than interest point detection. This error becomes exact in our case. We define the error as follows

$$E = \sum_{L_i} \sum_{C_j \in L_i} Dist(C_j, L_i) \tag{4.11}$$

The model parameters are the vanishing point $V = (v_x, v_y, v_w)$ and a set of N lines each of which is parametrized by a finite point in non-homogeneous coordinates $M_i = (m_{xi}, m_{yi})$. The line is defined as the one passing through the points V and M_i. We proceed to minimize the error E using Levenberg-Marquardt optimization. To do this, we need to obtain the partial derivatives of E with respect to the model parameters.

Let's assume that the corner point $C_j = (x_j, y_j)$.

Now the error is given as

$$E = \sum_{L_i} \sum_{C_j \in L_i} \frac{|N|}{\sqrt{D}} = \sum_{L_i} \sum_{C_j \in L_i} \frac{|(m_{yi}v_w - v_y) * x_j + (v_x - m_{xi}v_w) * y_j + (m_{xi}v_y - m_{yi}v_x)|}{\sqrt{(m_{yi}v_w - v_y)^2 + (m_{xi}v_w - v_x)^2}} \tag{4.12}$$

The derivatives of the function E with respect to the various parameters :

$$\frac{\partial E}{\partial m_{xi}} = (|N| * -\frac{1}{2} D^{-\frac{3}{2}} * (2v_w * (m_{xi}v_w - v_x))) + (D^{-\frac{1}{2}} * \frac{|N|}{N} * (-v_w y_j + v_y)) \tag{4.13}$$

$$\frac{\partial E}{\partial m_{yi}} = (|N| * -\frac{1}{2} D^{-\frac{3}{2}} * (2v_w * (m_{yi}v_w - v_y))) + (D^{-\frac{1}{2}} * \frac{|N|}{N} * (v_w x_j - v_x)) \tag{4.14}$$

$$\frac{\partial E}{\partial v_x} = \sum_i (|N| * -\frac{1}{2} D^{-\frac{3}{2}} * (-2 * (m_{xi}v_w - v_x))) + (D^{-\frac{1}{2}} * \frac{|N|}{N} * (y_j - m_{yi})) \tag{4.15}$$

$$\frac{\partial E}{\partial v_y} = \sum_i (|N| * -\frac{1}{2} D^{-\frac{3}{2}} * (-2 * (m_{yi}v_w - v_y))) + (D^{-\frac{1}{2}} * \frac{|N|}{N} * (-x_j + m_{xi})) \tag{4.16}$$

$$\frac{\partial E}{\partial v_w} = \sum_i (|N| * -\frac{1}{2} D^{-\frac{3}{2}} * (2m_{yi}(m_{yi}v_w - v_y) + 2m_{xi}(m_{xi}v_w - v_x))) + (D^{-\frac{1}{2}} * \frac{|N|}{N} * (m_{yi}x_j - m_{xi}y_j)) \tag{4.17}$$

These partial derivatives constitute a Jacobian vector J of the order $2N + 3$.

Our error function E is a scalar. The optimization algorithm proceeds by the following updates during each iteration.

$$J^\top J\Delta = J^\top E \tag{4.18}$$

The update vector is given by

$$\Delta = (J^\top J)^{-1} * J^\top * E \tag{4.19}$$

When we use LM-optimization, the above equation needs modified to accommodate a certain damping factor λ.

$$\Delta = (J^\top J + \lambda I)^{-1} * J^\top * E \tag{4.20}$$

Here I is the identity matrix, and λ is a parameter which depicts the success rate of the previous updates. If the updates are going well, the parameter λ is given a high value. Otherwise, it is given a low value.

4.5.4 Relation with Bundle Adjustment

The above method of Levenberg Marquardt optimization is related the method of bundle adjustment popular in stereo. When the feature points are sparse and not completely in agreement, the fundamental matrix which relates the two views of stereo is computed using a similar optimization approach. Our method can be understood as inherently similar to this approach, because symmetry can also be understood as a form of motion.

4.6 Discussions on Utility

The hallmark of the approach that we discussed for geometric grouping is that it does not need the patterns to be properly pre-segmented. Since segmentation is a tough problem, holding such an assumption would have been a severe limitation. Appearance based methods are particular prone for this, because it is hard to judge in the neighborhood whether a patch belongs to a pattern or to the background. In our method, we are less bogged by this limitation because the method is dependent on a set of few interest points rather than on the generic patches in the neighborhood. So, segmentation becomes less of an issue. If proper segmentation can be provided to our method, the results will of-course be better.

Our method is also applicable on patterns which do not exhibit strict repetitions. This happens when the two patterns are similar by parts, but not as a whole. An example of this is displayed in figure 4.4. We are able to perform this is because we are grouping the interest points based on vanishing point consensus rather than on strict geometric invariants

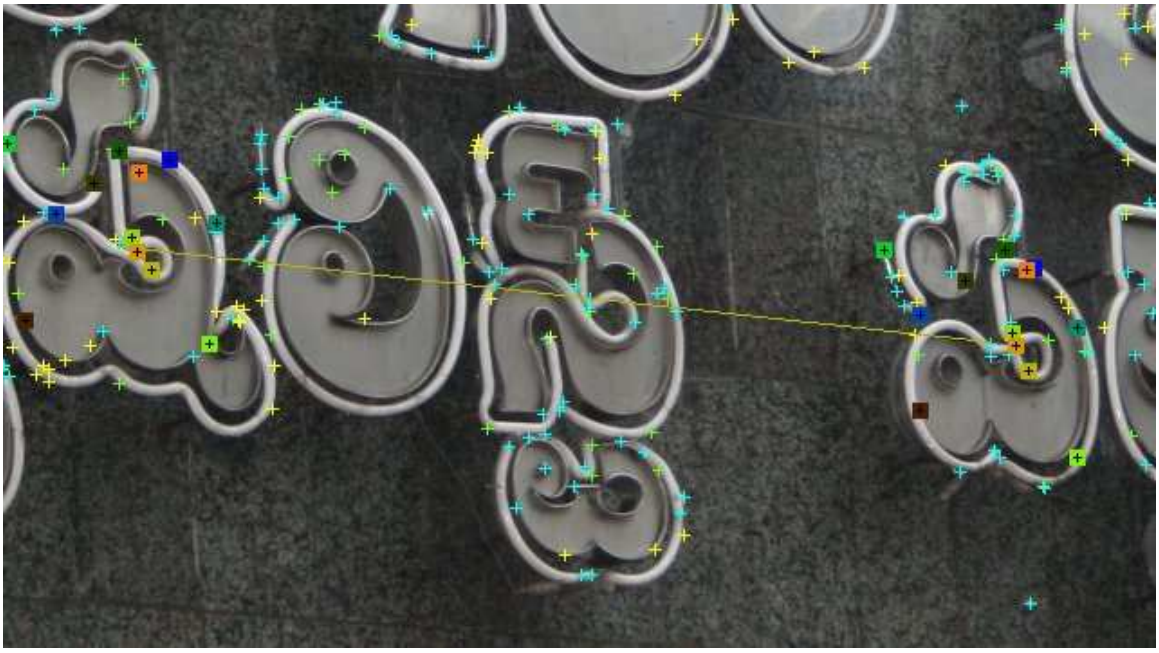


Figure 4.4: Pattern Matching with Partial Repetitions

of repetition.

If the user is not sure about the repetitions, he can turn off the condition which checks for validity on the basis of spatial coherence. By doing so, a more number of feature points can be detected which are valid. An example for this is displayed in the figure 4.5

4.7 Results

We demonstrate results of geometric grouping on three different kinds of images. The first set of images deals with architectural scenes which offer very minimal color information to aid in matching. The second set of images deals with photographs taken for symmetrical patterns on cloth. Most of these patterns do not have any line segments that can be easily detected. The third set of images deals with sign boards which contain characters of human languages. These characters are not strictly repetitive.

4.7.1 Architectural Scenes

We have taken several images of a set of tombs which were constructed by the Qutb Shahis - medieval emperors of the Deccan region of India. The tombs are a fine blend of Persian and Hindu influences. The walls of the tombs are principally flat, with embossed decora-

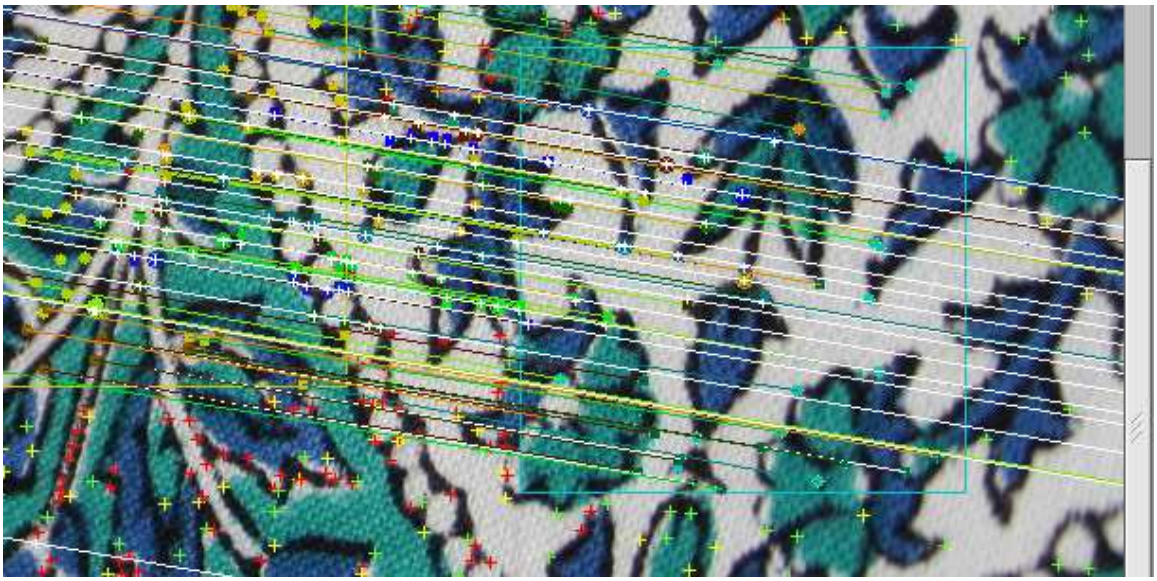


Figure 4.5: Pattern Matching without any Spatial Coherence

tive work as embellishments. This decorative work is highly repetitive - the tiled patterns include faux arches, eight-pointed stars, circular flowery patterns and so on.

The tomb is topped by a circular dome which highlights the symmetry. The circumference of the dome is etched with a sequence of planar decorative work.

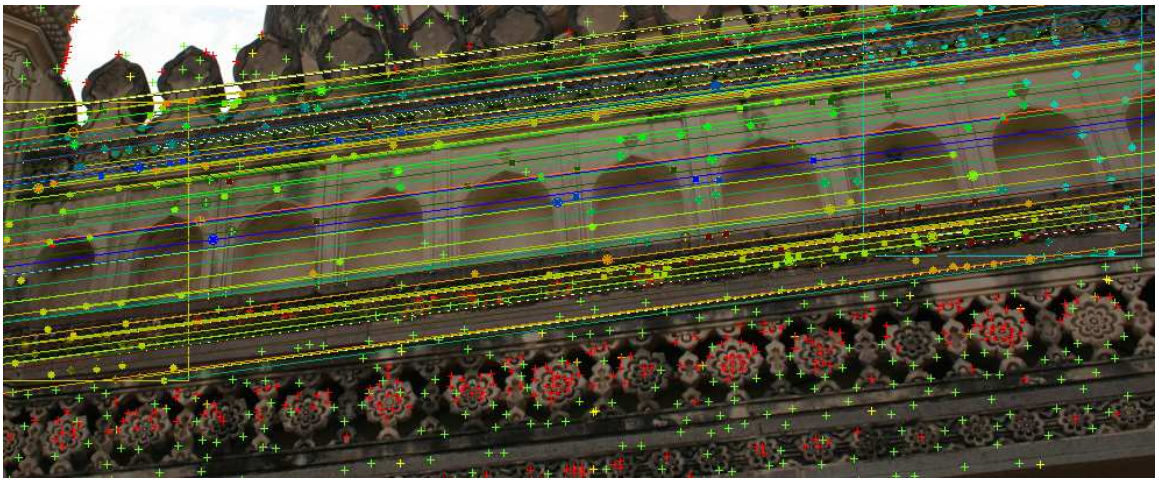
Unlike Hindu architecture which is adorned with a multitude of statues, Islamic architecture is minimalist. Most of the structures are also piecewise-planar. The decorative art work is directly embossed onto the planar walls. Each of these patterns of decoration is a careful choice of symmetry. These patterns are tiled in a very painstaking manner by the artists.

These patterns readily offer us corners and linear features. There are even a couple of rectangular patterns available. However, the majority of the information comes in the form of repetitive and symmetric patterns.

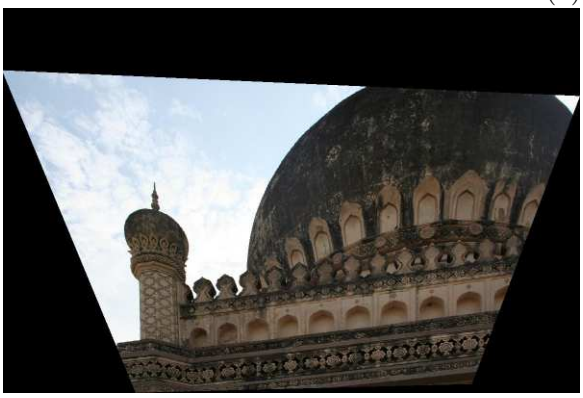
In the figure 4.7.1, the results of the grouping algorithm on these images are demonstrated.

4.7.2 Designs on Cloth

Using a perspective camera, we have taken several pictures of cloth spread on the top of a planar surface. These images exhibit different forms of symmetry due to tiling and mirror

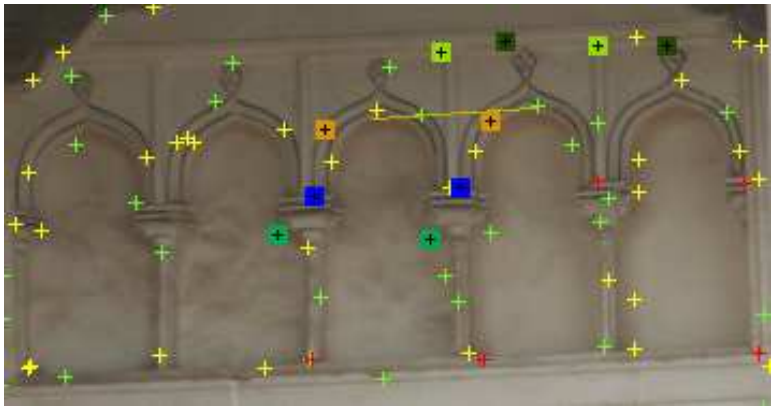


(a)



(b)

(c)



(d)

Figure 4.6: Results of the geometric grouping on Qutb Shahi tombs : (a) Parallel Lines constructed automatically (b) Image Rectified through vanishing line (c) Image rectified through circular points (d) Point correspondences across patterns

reflective patterns. Note that most of the images do not contain any easily identifiable straight line segments.

The results of the geometric grouping are present in the figure 4.7.2

4.7.3 Characters on Sign Boards

We have taken several pictures of characters on sign boards on the streets with a perspective camera. Since scripts of human languages exhibit a lot of symmetry, these characters can be used to remove perspective distortion. The results are shown in the figure 4.7.3. Please note the patterns are matched even in the cases where the repetition is only partial.

4.7.4 Normalization of Image Coordinates

Before conducting any type of geometric processing, the contours in the image have to be normalized. Without normalization, the results of the geometric computation will be lost beyond the precision of floating point variables. To avoid such numerical instabilities, the image contours should be constrained to have certain fixed properties. In this section, we discuss the normalization that we have applied for achieving the results.

In his work on the defense of the 8-point algorithm [42], Hartley discusses the procedure for normalizing the image coordinates one needs to do before attempting the task of projective reconstruction.

This normalization needs to be performed in such a manner that the following two constraints are ensured :

1. The average position of the pixels should be the origin $(0, 0)$.
2. The average distance of the pixels from the origin should be unity (1 unit).

We implement the ideas of [42] in the current problem of matching repetitions inside a single image.

Let the height of the image (along the X direction) be h and the width of the image (along the Y direction) be w. The first constraint is ensured by translating all the pixels by

$$(x, y) = (x - h/2, y - w/2) \tag{4.21}$$

To enable the second constraint, one needs to compute the average distance d of each pixel from the origin.

$$d = \frac{1}{h \times w} \sum_i \sqrt{x_i^2 + y_i^2} \tag{4.22}$$

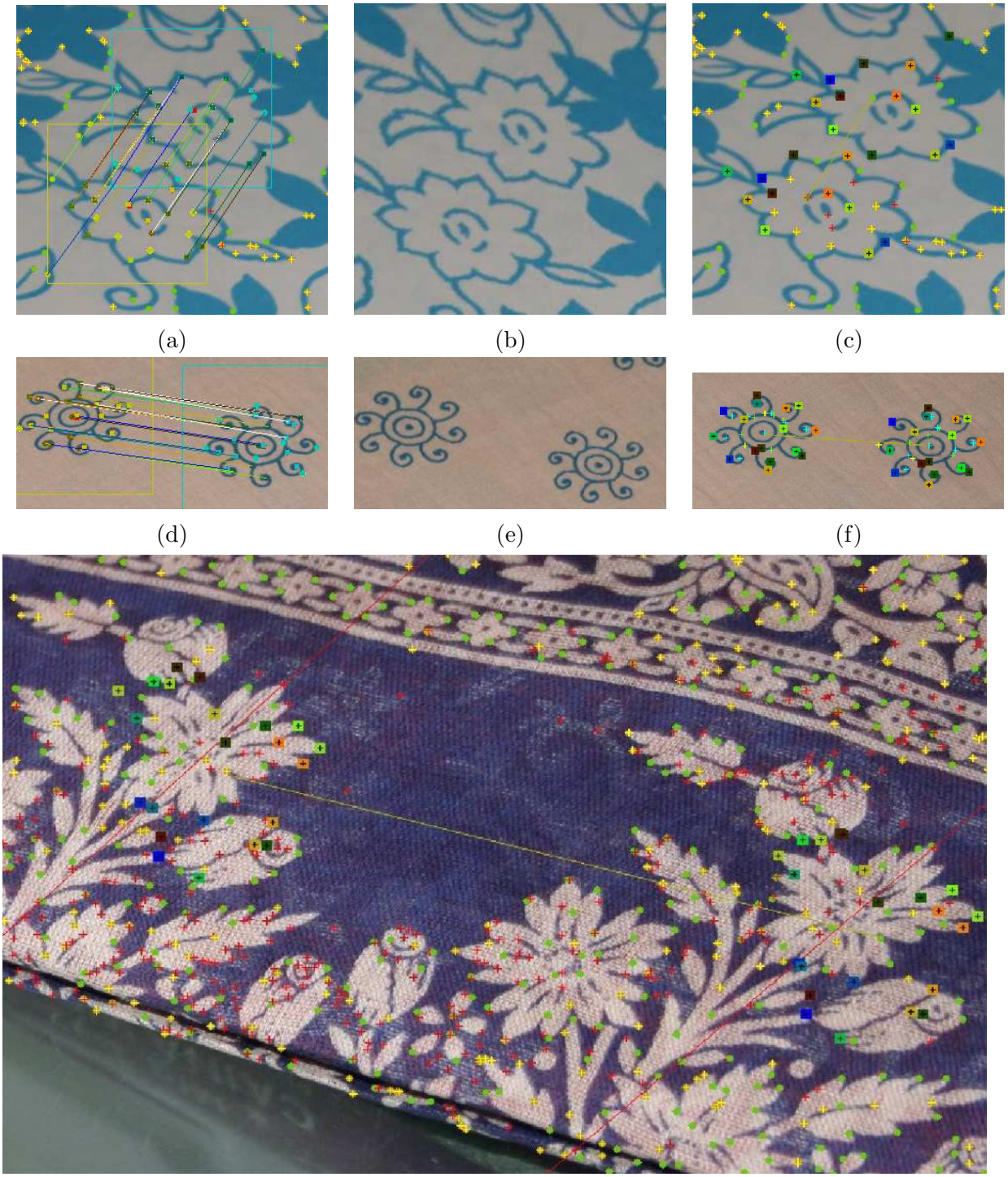
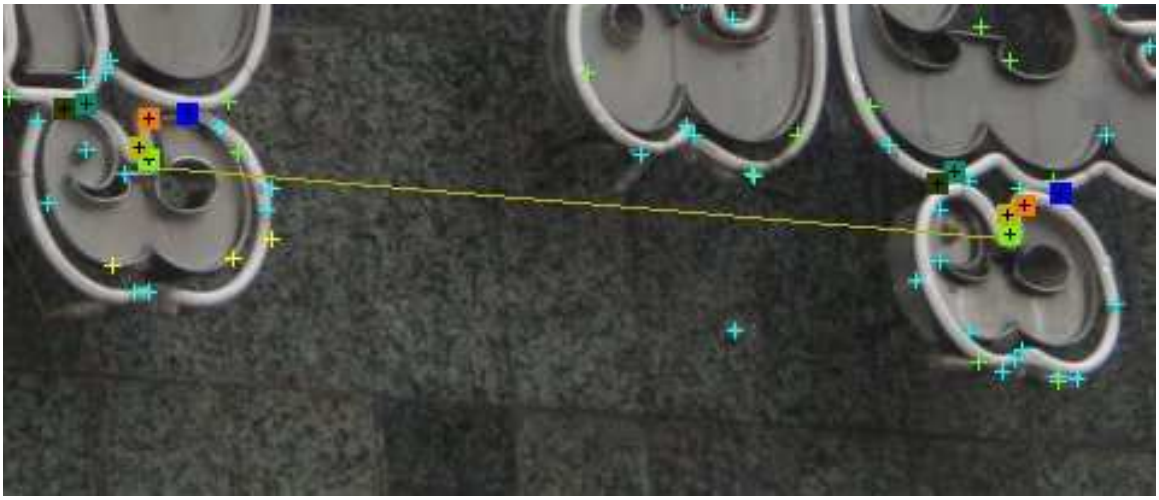
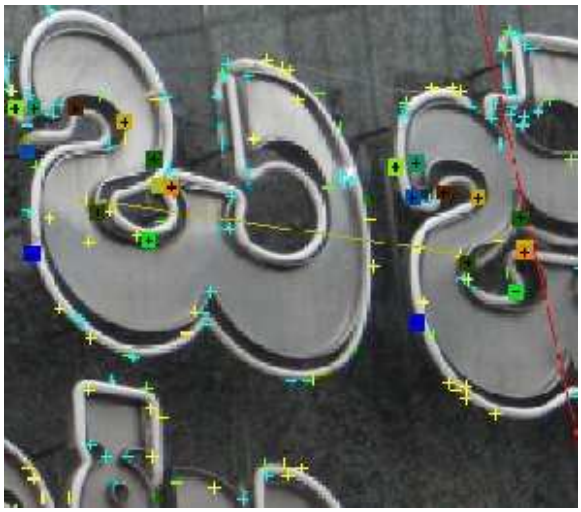


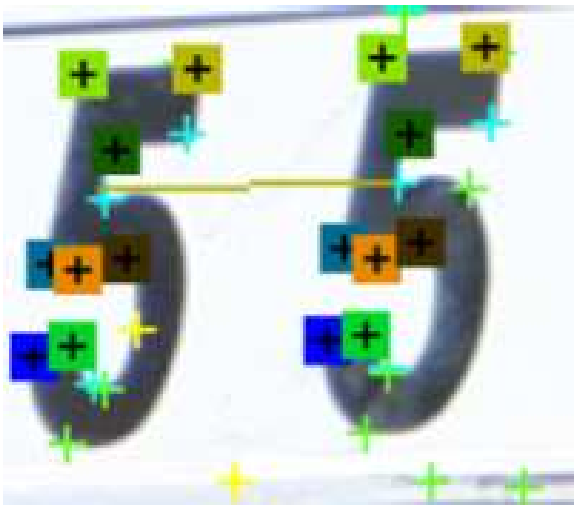
Figure 4.7: Geometric grouping for symmetric patterns on a cloth : (a)(d) - geometric grouping in action, (b)(e) - affinely rectified images, (c)(f)(g) - point correspondences across patterns



(a)

(b) (c)





(d) (e)

Figure 4.8: Results of geometric grouping on images of characters on sign boards : Note that the points are matched even when the repetition is partial

Then the image is scaled by the below factor in the X direction.

$$\frac{d * h}{\sqrt{h^2 + w^2}}$$

The scaling to be done for the Y direction is similar can be obtained by exchanging the position of h with w in the above formula. This scaling ensures the second constraint.

If the user has segmented out a chosen object of interest, and geometric processing needs to be done on its contour, one can choose the normalization factor to better reflect the problem situation. A rectangular *window of operation* is first identified which completely contains the contour. Then, the entire normalization is performed by treating this *window of operation* as a complete image.

4.7.5 Rank Conditioning for Matrix Problems

The reason why normalization works in favor of the geometric computation is because it improves the condition number of the fundamental matrix.

The condition number of a matrix is defined as the ratio of the largest and the smallest eigen values. If this number becomes very high, the inverse of a matrix runs into numerical instabilities. In an *ideal* situation, the condition number has the value of 1. When the matrix is rank-deficient, the condition number becomes ∞. Thus, the magnitude of the condition number needs to be as low as possible.

Hartley [42] argues how the above mentioned normalization scheme improves the conditioning of the fundamental matrix. In the current context, the homography which relates one pattern with its repetition is the equivalent of the fundamental matrix.

4.8 Conclusion

In this chapter, we have presented a method for doing geometric grouping by a combinatorial optimization on strictly geometric information. The classical methods which use geometric invariants have been limited in their utility because they require the interest points to be detected accurately. The methods based on affine invariant appearance models, have suffered from limitations such as being too dependent on object segmentation, requiring a lot of texture information etc. In contrast, in the current algorithm, we perform grouping by comparing only geometric information.

We have demonstrated that this method works on natural images taken from a perspective camera. Though the theory has been well developed for doing metric rectification from

arbitrarily complex symmetric patterns, past methods have not been able to exploit this when there are no simple line-segments that can be detected or when the color information is minimal.

By showing results on images of symmetric designs on cloth, and on architectural scenes, we have demonstrated that these images can be handled with equal amount of ease.

Chapter 5

User Guided Geometric Grouping

This chapter presents the algorithms of geometric grouping from the perspective of an interactive Image based Modeling and Rendering (IBMR) application. Classical approaches for single view image reconstruction have required the user to provide input in the form of identifying points and line segments. This is a pain staking process. Due to the approach of geometric grouping, some of these difficulties can be done away with.

5.1 Motivation for Interactive Methods

Since computer vision is a hard task, sometimes, user interaction cannot be avoided. Typically, user input is needed in the form of high level knowledge - such as recognizing an object. This knowledge is later coupled into applications such as image matting, image inpainting etc. The trend of *putting the human into the loop* has become very popular in computer graphics in the recent past.

An interactive application typically claims to do a much better job than an automatic system. User input facilitates in greatly increasing the scope and the range of operation of the system. We demonstrate how this is true in the context of an IBMR application.

5.2 User Friendliness

In the context of an interactive IBMR application, we term a system as user friendly if it does not require any input to be accurate at the level of each pixel. Human beings are good at identifying high-level features, but they are not very good at providing low-level features with high accuracy. An example can be provided in the context of image segmentation. The classical methods have required the user to specify some pixels on the boundary. The novel methods of interactive segmentation require the user to specify some pixels in the interior of the object. The later input is much easier for the user to provide because it provides

lesser scope for making errors.

5.2.1 Tolerating errors

The classical methods of single view reconstruction have required the user to provide input in the form of identifying the position of points, and there by specify metric constraints on the object. Criminisi et al [39] have studied the effect of errors in the user input through a Gaussian error model. Since the classical approaches compute geometric structures through the basis of a very few seed points, they tend to be very intolerant towards the errors in the input. Methods of geometric grouping aggregate the information from several regions in the image and thus would be more tolerant towards errors.

5.2.2 Taking higher level input

Humans can provide information at a higher level easily. Examples of this include (i) specifying if an object is present within the image or not (ii) specifying if an object inside the image exhibits a type of symmetry (iii) Specifying which one of the two objects is bigger in the real world (iv) Specifying whether the object has a symmetrical correspondence along a direction or not etc.

In contrast, the following information is required at the lower pixel-level and is thus more difficult for humans to provide accurately. This includes tasks such as (i) Specifying the location of intersection of two straight lines (ii) Specifying the exact location of a symmetrical point correspondence (iii) Specifying the boundary of an object (iv) Specifying the exact angle or length etc.

5.2.3 Providing visual feedback

For providing any type of information with accuracy, humans prefer to do it through a dialog with feedback. An interactive application needs to provide a dialog for the user whenever it demands any type of input. This principle is properly utilized in the current system.

5.3 Forms of User Interaction

In this section, we discuss the forms of input needed by the geometric grouping algorithm and how the user may help in providing this.

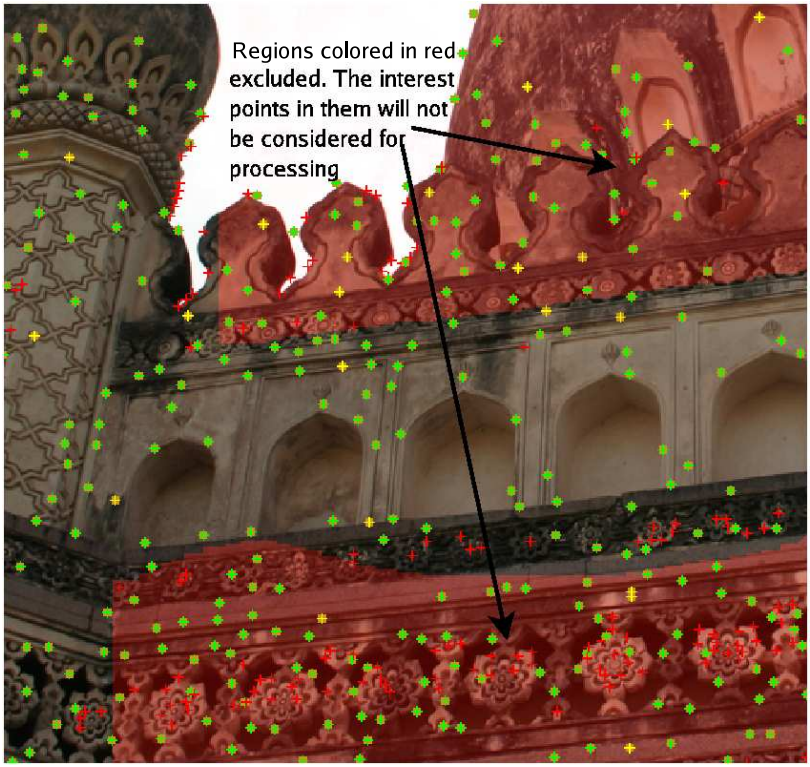


Figure 5.1: User interaction for excluding portions from the region of interest

5.3.1 Selecting the Region of Interest

The region of interest needs to be identified to define the set of left-points and the set of right-points, before they can be provided for the algorithm for matching. Though the geometric grouping algorithm is tolerant towards the errors in this specification (discussed in section 4.2), it requires a rough estimate of the left and the right point sets.

We provide the following set of tools for the user to provide this input.

Excluding a region to one direction of a curve : The user can choose amongst the left, right, top or bottom directions to be eliminated as he draws a rough stroke on the image. This gesture would remove several misleading interest points from the sample space, and thus improve the accuracy. (figure 5.1)

Considering only the neighborhood of a specified point : The user provides the extent of a region by just a single mouse-click. The user can also choose to vary the size of the brush to specify the extent of the surrounding region that is required. (figure 5.2)

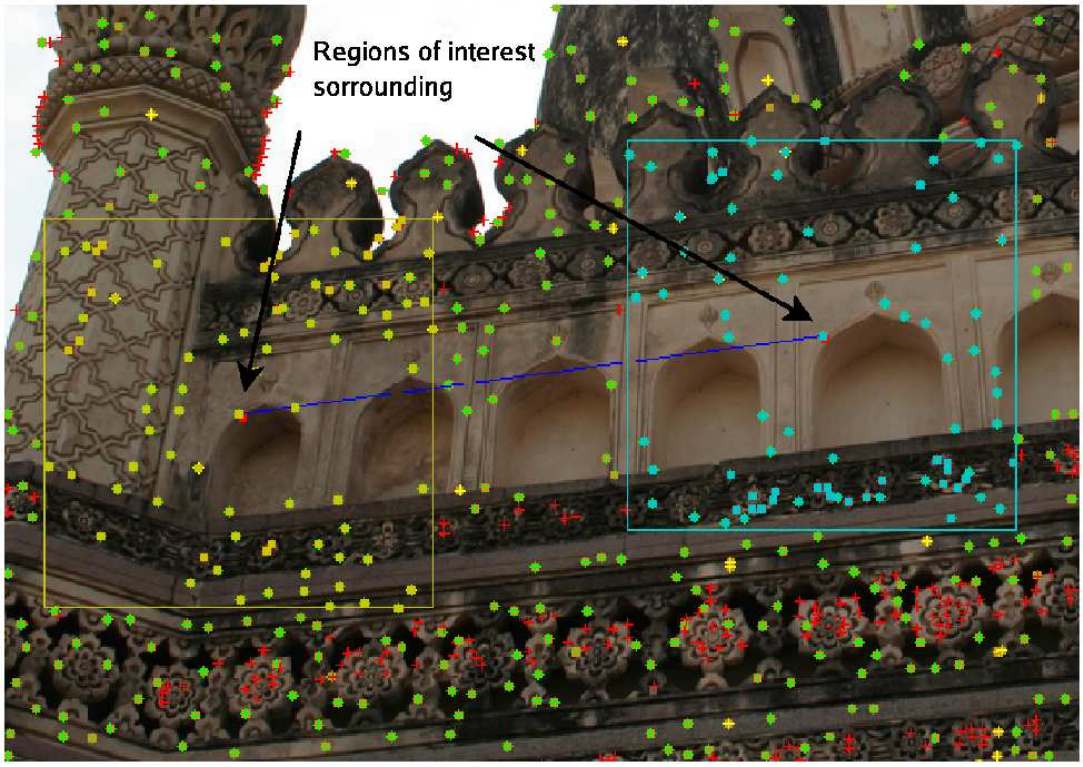


Figure 5.2: User interaction for specifying the region of interest by a mouse-click

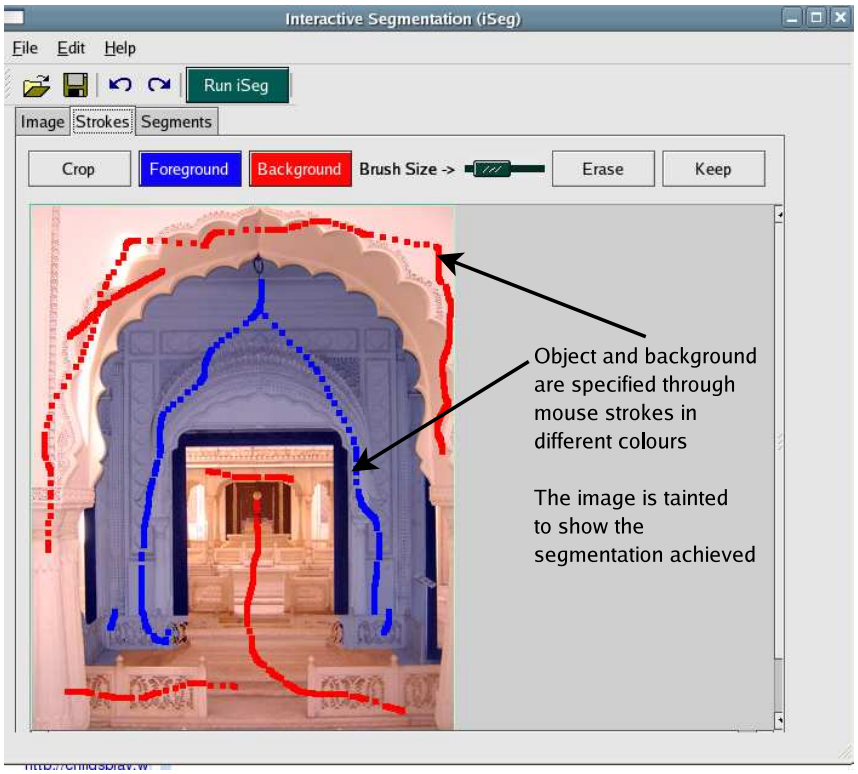


Figure 5.3: User Interface for Image Segmentation by graph-cuts

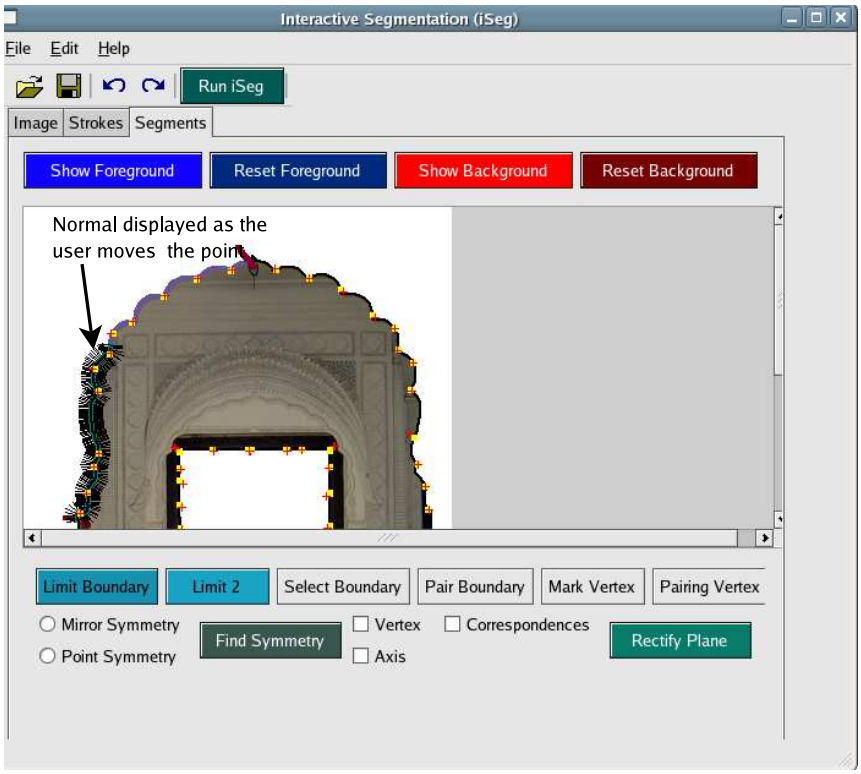


Figure 5.4: User Interface for choosing the region of interest on an object boundary

Through intelligent segmentation by graph cuts : Our system provides the user with the tool of interactive segmentation based on graph cut optimization. With a couple of mouse-strokes, the user may specify the extent of the region that he wants to consider. If the user is not satisfied with the segmented region, he may provide modifications through additional mouse-strokes. (figure 5.3)

Through putting limits on the boundary of an object : Sometimes the user may wish to consider only those points that are present on the boundary of a pre-segmented object. He may further wish to specify portions of the boundary which are truly useful in rectification and exclude the rest. He may do so by the placing markers delimiting the boundary. However, we do not required him to be accurate by clicking with a mouse. Instead we provide him with a button on the key board which he can press to move along a specific direction on the boundary. As he moves along, we provide him with a visual feedback by displaying him the normal to the curve at that position. This kind of interaction is far more user-friendly. (figure 5.4)

5.3.2 Specifying the Direction of Match

The user should specify the approximate direction of the match to initialize the search space. This can be provided as a line-segment by specifying the two positions of the end points. It can also be specified as a single stroke of the mouse. The user is provided with a visual feedback to assist him in making a right choice. (figure 5.5)

5.3.3 Specifying the Interest Points

Sometimes, point features cannot be done away with. These are needed, for example, in identifying symmetrical point correspondences. Since a point in the image is a feature whose specification requires pixel level accuracy, we cannot simplify this task drastically for the user. However, we can still make it more benign, by precomputing the entire set of candidate interest points. We take user input in two forms

- **Demarking a neighborhood** We treat the input of the user in a fuzzy sense. We define a probability distribution around the point and consider each of the neighboring points as a likely candidate for the user input. The neighborhood can be identified by a single mouse-click. Alternatively, it can be identified by drawing an ellipse. In the latter case, the probability distribution will be strictly confined to the interior of the boundary.

- **Providing a direction** If a specific point is identified beforehand, for example the center of mass of a pattern, the user can specify the interest points by giving the direction towards which they are located, from the known point.

Figure 5.5: User interaction for specifying the approximate direction of the match

5.3.4 Specifying the Fixed Structure

The user may also specify the fixed structure of a transformation directly - such as the vanishing point along a direction, in an approximate manner. The vanishing point would then be searched for only in the local neighborhood of the given specification. If some vanishing points are already known, the vanishing line would be constructed to help the user in visualization. The input image will be affine rectified and presented to the user as a visual feedback for interaction. (figure 5.6)

If the fixed structure is a straight line, the user may specify it approximately by drawing a stroke with the mouse. This is helpful, for example, with the case of the axis of reflection in bilateral symmetry.

We also provide the user with an option for specifying the circular points directly. As discussed in the section (2.5), metric constraints such as equal angles and equal ratio of lengths yield circular constraints on the $\alpha\beta$ plane. We provide an interface for the user to visualize these circles and provide circular points directly with their aid. The given image is metric rectified and displayed to the user as a form of visual feedback. (figure 5.7)

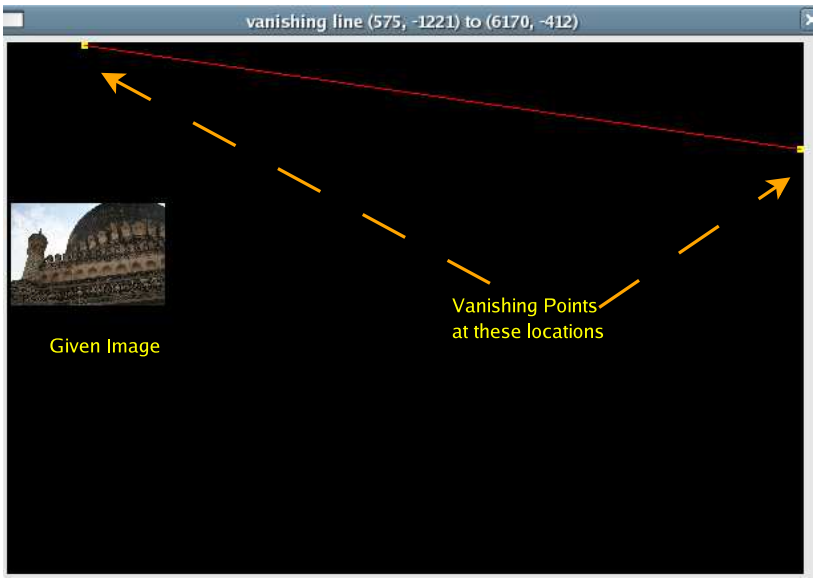


Figure 5.6: User interface for visualizing the vanishing line

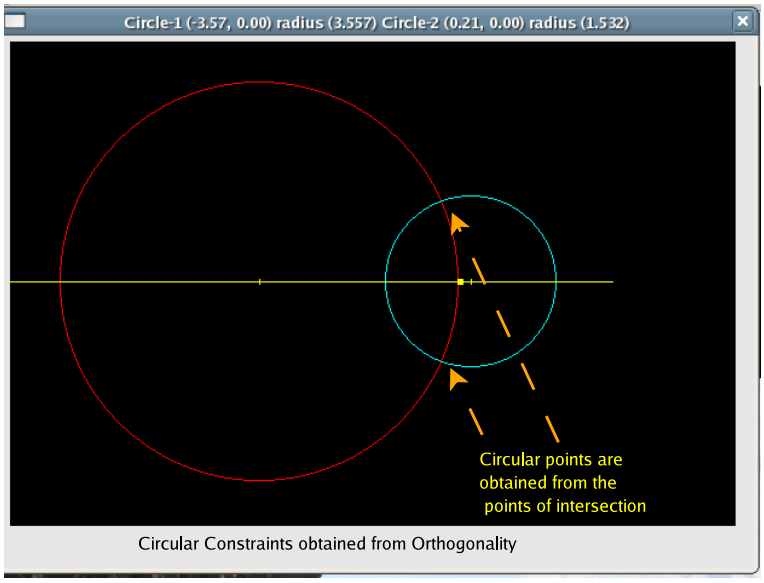


Figure 5.7: User interface for visualizing the circular points

5.4 Conclusion

In this chapter, we have discussed how to take user input for the task of geometric grouping from the perspective of user-friendliness. These forms of interaction may be looked as an alternative to the classical paradigm for single view reconstruction. In this alternative, we make no obligation on part of the user to provide input with pixel-level accuracy. We are more tolerant towards errors in the user input, and provide with a simple way to correct them through visual feedback.

Chapter 6

Conclusion

In this thesis, we have presented a new framework for handling mid-level tasks in computer vision. The primitives for this framework are a set of interest points on the image. An image which has 10^6 pixels may have only about 10^3 interest points, thus bringing the tasks of mid level vision into the fold of combinatorial optimization. We have discussed several ways of further reducing the computational complexity of this optimization - specifically through the use of coherence in geospatial information and through the use of user interactivity. We believe that the inputs being taken from the user are for information at a high conceptual level and will be hard to replace. However, we envisage completely automatic algorithms replacing the human user. These algorithms would use Bayesian learning on a large set of training examples to help in the decision process.

Until that happens, interactive systems would offer the best means for solving high level vision problems. These interactive systems can also be used for quickly labeling the data and thus for providing the training set of training examples, which may be used by the automatic algorithms. We discussed the concept of interactivity from the perspective of user friendliness. We have implemented these principles in an interactive IBMR application, which offers friendlier forms of user interaction through the proper use of geometric grouping.

6.1 Major Contributions

Our major contributions are in the area of using spatial coherence for geometric grouping. Instead of considering the 8-neighbors, we looked at a wider and deeper neighborhood which tags each point with more global information. We have discussed how this information can be used to compute point correspondence based purely on geometric information. Since we do not use appearance based features, we do not encounter limitations such as being too dependent on image and texture segmentation.

In building a purely geometry based grouping system, we are facilitated by the discovery of a new method for detecting interest points. Even for the interest point detection, we discard the classical 8-neighbors and look into a deeper neighborhood. By observing the macro-shape properties of the image contours, we identify interest points which have a more global significance. Indeed, without these interest points, our algorithms for geometric grouping would not have been very successful.

The principle of using spatial coherence at a deeper level is the theme that unifies our entire work. We believe that this form of processing becomes the norm in the future.

6.2 Limitations

Since our algorithms are purely geometric, they suffer from certain handicaps. They are currently not able to exploit the texture information in the cases where it is useful. We need to generalize our methods of analyzing spatial coherence to understand appearance models as well.

Though it is feasible to build a completely automatic system to do geometric grouping, we have built only an interactive system. We have done this in order to distinguish the harder tasks which require higher level of conceptual information from the lower level tasks which can be automatized. Our method for grouping, currently yields its best results when it proceeds from the user input. We would like to work on automating these aspects of our algorithm.

6.3 Applications - Future Work

Geometric grouping is a generic tool which is useful in several problem areas of computer vision. It has been demonstrated by Hoeim et al [10] that the geometric context of each pixel yields a lot of useful information with respect to the tasks of object detection, 3D reconstruction etc. The geometric groups that are recognized by our algorithm can be utilized in a similar manner.

Since the geometric group produces a larger sense of locality, the features computed on the group will have larger scopes of invariant properties. These features can be employed in several high level vision tasks such as object detection and recognition. Much work needs to be done in these directions.

Geometric grouping has applications towards several other problem areas such as projec-

tive OCR, stereo reconstruction, image based retrieval and so on. To make our algorithms suitable for these applications, it is necessary to automatize some of the tasks that are being handled by the user. So, the future work is oriented primarily in this direction.

Bibliography

[1] Yung-Yu Chuang, Brian Curless, David H. Salesin, and Richard Szeliski, "A bayesian approach to digital matting," in *Proceedings of IEEE CVPR 2001*. December 2001, vol. 2, pp. 264–271, IEEE Computer Society.

[2] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother, "A comparative study of energy minimization methods for markov random fields," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2006.

[3] C.Slama, *Manual of Photogrammetry*, American Society of Photographers, 1980.

[4] EOS Systems, "Photomodeler - software for modeling from images," .

[5] RealViz Solutions, "V-tour, image-modeler - software for modeling from images," .

[6] Yuri Boykov and Marie-Pierre Jolly, "Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2001, pp. 105–112.

[7] Yuri Boykov, Olga Veksler, and Ramin Zabih, "Fast approximate energy minimization via graph cuts," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 1999, pp. 377–384.

[8] Pawan Kumar M, Phil Torr, and Andrew Zisserman, "Obj-cut," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[9] Anat Levin, Yair Weiss, and Dani Lischinski, "Colorization using optimization," in *Proceedings of the ACM International Conference on Computer Graphics and Interactivity (SIGGRAPH)*, 2004.

[10] Derek Hoeim, Alexei Efros, and Martial Hebert, "Geometric context from a single image," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2005.

[11] Derek Hoeim, Alexei Efros, and Martial Hebert, "Automatic photo pop-up," in *Proceedings of the ACM International Conference on Computer Graphics and Interactivity (SIGGRAPH)*, 2005.

[12] Derek Hoeim, Alexei Efros, and Martial Hebert, "Putting objects in perspective," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[13] Serge Belongie Jitendra Malik and Jan Puzicha, "Shape context: A new descriptor for shape matching and object recognition," in *Proceedings of the International Conference on Neural and Information Processing Systems (NIPS)*, 2000.

[14] Greg Mori and Jitendra Malik, "Recovering 3d human body configurations using shape contexts," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2006.

[15] Andrew Johnson, *Spin-Images: A Representation for 3-D Surface Matching*, Ph.D. thesis, CMU, Pittsburgh, 1997.

[16] A. Frome, D. Huber, R. Kolluri, T. Buelow, and J. Malik, "Recognizing objects in range data using regional point descriptors," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2004.

[17] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, H. Pang, and J. Davis, "The correlated correspondence algorithm for unsupervised registration of nonrigid surfaces," in *Proceedings of the International Conference on Neural and Information Processing Systems (NIPS)*, 2004.

[18] Richard Hartley and Andrew Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2000.

[19] M.Pawan Kumar, C.V.Jawahar, and P.J.Narayanan, "Geometric structure computation from conics," in *Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing*, 2002.

[20] F. Schaffalitzky and A. Zisserman, "Planar grouping for automatic detection of vanishing lines and points," *Image and Vision Computing*, vol. 18, pp. 647–658, July 2000.

[21] D. P. Mukherjee, A. Zisserman, and J. M. Brady, "Shape from symmetry - detecting and exploiting symmetry in affine images," in *Philosophical Transactions of the Royal Society of London, Series A*, 1995, vol. 351, pp. 77–106.

[22] Y.Lamadan, J.Schwartz, and H.Wolfson, "Object recognition by affine invariant matching," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1988, pp. 335–344.

[23] Luc Van Gool, Theo Moons, and Marc Proesmans, "Mirror and point symmetry under perspective skewing," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1996.

[24] J.Ponce, "On characterizing ribbons and finding skewed symmetries," in *Proceedings of the International Conference on Robotics and Automation*, 1989, pp. 49–54.

[25] L. Van Gool, M. Proesmans, and A. Zisserman, "Planar homologies as a basis for grouping and recognition," *Image and Vision Computing*, vol. 16, pp. 21–26, Jan. 1998.

[26] Luc Van Gool, Theo Moons, Marc Proesmans, and André Oosterlinck, "Groups, fixed sets, symmetries and invariants," in *Proceedings of the International Conference on Image Processing*, 1995.

[27] J. Liu, J. Mundy, and A. Zisserman, "Grouping and structure recovery for images of objects with finite rotational symmetry," in *Proceedings of the Asian Conference on Computer Vision*, 1995, vol. I, pp. 379–382.

[28] Wei Hong, Allen Yang Yang, Kun Huang, and Yi Ma, "On symmetry and multiple-view geometry: structure, pose and calibration from a single image," *International Journal on Computer Vision (IJCV)*, 2004.

[29] Y.Ma, J.Kosecká, and K.Huang, "Rank deficiency condition of the multiple-view matrix for mixed point and line features," in *Proceedings of the Asian Conference on Computer Vision (ACCV)*, 2002.

[30] Paresh Jain, C.V.Jawahar, and P.J.Narayanan, "Computation of projective homography from fourier domain analysis," in *Proceedings of the International Conference on 3D Processing, Visualization and Transmission (3DPVT)*, 2006.

[31] Tinne Tuytelaars, Luc Van Gool, Marc Proesmans, and Theo Moons, "The cascaded hough transform as an aid in aerial image interpretation," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 1998, pp. 67–72.

[32] Allen.Y.Yang, Kun Huang, Shankar Rao, Wei Hong, and Yi Ma, "Symmetry based 3d reconstruction from perspective images," *Computer Vision and Image Understanding (CVIU)*, pp. 210–240, 2005.

[33] Tinne Tuytelaars and Luc Van Gool, "Wide baseline stereo based on local, affinely invariant regions," in *Proceedings of the British Machine Vision Conference*, 2000.

[34] Andreas Turina, Tinne Tuytelaars, and Luc Van Gool, "Efficient grouping under perspective skew," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2001.

[35] Jianbo Shi and Carlo Tomasi, "Good features to track," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1994.

[36] J. Koenderink, J. and J. van Doorn, A., "Affine structure from motion," *J.Opt.Soc.Am.A*, vol. 8(2), pp. 377–385, 1991.

[37] O.D.Faugeras, "Stratification of three-dimensional vision: projective, affine and metric representation," *Journal of Optimization (J.Opt.Soc.Am.A)*, vol. A12, pp. 465–484, 1995.

[38] Rafael C.Gonzalez and Richard E.Woods, *Digital Image Processing*, Pearson Education, 2002.

[39] A. Criminisi, I. Reid, and A. Zisserman, "Single view metrology," *International Journal of Computer Vision*, vol. 40, no. 2, pp. 123–148, Nov. 2000.

[40] W. Press, B. Flannery, S. Teukolsky, and W. Vetterling, *Numerical Recipes in C*, Cambridge University Press, 1988.

[41] D. Liebowitz and A. Zisserman, "Metric rectification for perspective images of planes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 1998, pp. 482–488.

[42] Richard I.Hartley, "In defence of the eight-point algorithm," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 19, June 1997.