The Interdisciplinary Center, Herzliya
Efi Arazi School of Computer Science

Temporal Epipolar Regions and Correspondence

M.Sc. Dissertation

Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science (M.Sc.) Research Track in Computer Science

Submitted by Mor Dar
Under the supervision of Prof. Yael Moses

May 2015

Acknowledgements

I would not have been able to accomplish this work were it not for Professor Yael Moses. The hours she spent teaching and advising me were invaluable. I appreciate all her insights in finding just the right way to approach problems, her direction when I started veering off coarse, and her patience while we hunted for the best way to simply and concisely write an article on what was surprisingly hard to explain. Through her guidance, I learned much about computer vision, the ins and outs of computer science research, and in turn, what I am looking for in my career. I am forever grateful for her efforts over the past year.

I would also like to thank Efrat, my soon to be wife, and the rest of my family for supporting, encouraging, and believing in me through occasional late nights, a few rough days, and a couple of pre-deadline panics.

Abstract

Dynamic events are often photographed by a number of people from different viewpoints at different times. Finding the corresponding moving features in each of the resulting images allows us to extract information about objects of interest in the scene. Computing correspondence of moving features in such a set of images is considerably more challenging than computing correspondence in video, as the prediction methods used in video are not applicable to an unconstrained set of still images. In this thesis we propose a novel method which improves the accuracy of feature matching algorithms by predicting valid locations of a feature point, given a small subset of correspondences. Our method utilizes epipolar geometry to divide images into valid and invalid regions, termed *Temporal Epipolar Regions*. These regions determine which features are consistent with the order of image captures and an approximately linear trajectory of the point. We prove that none of the feature points in invalid regions can be candidate correspondences. We demonstrate the effectiveness of our method to reduce the search space for correspondence on both synthetic and challenging real world data, and show the improved matching.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Given is a set of still images which capture a moving object from multiple viewpoints at different times. Can we predict the location of this object in each of the images if we only know its location in three? The general answer to this is no. In this thesis, we propose a solution to this problem when the object is assumed to have approximately linear, unidirectional motion. Before describing our method, we first explain the motivation for solving this problem.

The popularity of smartphones today ensures that most of us always have a camera within reach, and therefore, interesting dynamic events may be captured by many cameras. The collection of such cameras, is termed a *crowd-based camera* (or *CrowdCam* for short). CrowdCam captured scenes do not require any coordination among the photographers, and therefore the viewpoints and timing between images differs. However as all the photographers capture the same event, the dynamic object is captured in all the images over a relatively short time interval. Tagging in social media facilitates quick retrieval of CrowdCam captured data from exciting events. To extract information about what happened in CrowdCam captured scenes, we need to look beyond data retrieval and analyze the scene itself. One such analysis involves examining the motion of the dynamic objects in the scene. Tracking an object through a set of frames in video is considered a fundamental problem of computer vision. However, videos of dynamic scenes are not always available. In such cases, we can analyze the motion of an object through still images by finding the corresponding object in each image.

Computing correspondence of points in images of a static scene can benefit from the use of epipolar

constraints. Given the epipolar geometry and the location of a matching static point in at least two images, it is a relatively simple task to identify the location of said point in the rest of the images. In this case, epipolar constraints guarantee that in each image for which the correspondence is unknown, the point must lie on epipolar lines calculated from the location of the static point in other images. Therefore, the location of the point is the intersection of these epipolar lines.

However, if the scene is dynamic and the point moves, the location of the point is no longer limited to just the epipolar lines, and the epipolar constraint is no longer applicable. As such, analysis of moving objects in CrowdCam image sets, as in tracking in video, often requires computing correspondence through direct matching of features. Finding corresponding dynamic features in a set of CrowdCam images is challenging, as strategies which directly match features rely entirely on the features themselves and do not take into consideration the geometry between cameras or the timing of image captures. CrowdCam images are captured without any coordination between the photographers, and as such each image may be shot from a different viewpoint and the time between image captures may be inconsistent. As a result, feature matching must be able to overcome viewpoint variations, obstructions, noise, or changes in illumination while still differentiating between features and selecting the correct match.

In this thesis we proposed a new geometric constraint for reducing the search space for candidate correspondence of the moving feature point through the set of images. For a general set of images with an object moving arbitrarily, the feature point can be projected to any location in each image, depending on the camera position at capturing time. Furthermore, each projected feature is independent of the location of corresponding features in the other images. As such, additional information (3D trajectory, speed, precise timing between image captures, etc.) is required in order to predict its location. Our work does not require this type of knowledge, which may be difficult to obtain. Instead, we assume that each dynamic point is moving in an approximately linear trajectory and that the temporal order in which the images were captured is known (if it is unknown, photo sequencing [11] may be used to calculate it). Similar assumptions are often used in video for feature or object prediction in tracking algorithms.

Image 2

Image 4

Image 6

Image 3

Figure 1.1: An example (from dataset g1) of limiting the search space for correspondence using temporal epipolar regions. The epipolar lines on image 3 are calculated using the set of correspondences from images 2, 4, and 6 (yellow stars) in the set. These lines are used to define temporal epipolar regions which limit the search space for correspondence in image 3 (only areas not colored in red should be considered for correspondence). The valid regions are consistent with the order of image captures and the unidirectional linear motion of the point.

Our method: Given the correspondence of the moving feature in a small set of images, we utilize epipolar geometry in order to predict the location of the point in the remainder of the images . To do so, we calculate epipolar lines from all known corresponding points in an image for which correspondence is unknown. As shown by Dekel *et al.* [11], the order of the intersections of these epipolar lines with the projection of the 3D trajectory of the point onto the image, must match the order of image capture times. While we do not know the trajectory of the point, we can utilize this concept in order to define *Temporal Epipolar Regions*, TERs, bounded by a set of epipolar lines and their parallels (Figure 1.1).

We prove, using geometric considerations, that all points in each TER are either consistent or inconsistent with the order of capture timings of the set of images. All points in consistent regions are valid candidates for correspondence, while all points in inconsistent regions are invalid. Therefore,

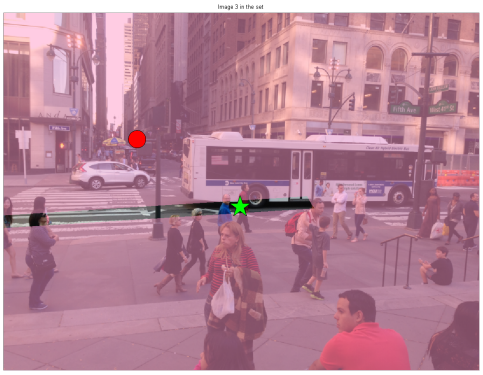


Figure 1.2: An example (from dataset d1) of finding correspondence with and without the use of TERs. In this image we see that the match found without using TERs (the red circle) is incorrect while the same matching algorithm with use of TERs successfully finds the corresponding point (green star), given a set of 4 correct correspondences in other images.

TERs define an area in the images in which the point must reside. TERs can be computed efficiently and limit the search space for candidate points, thus improving matching results of feature matching algorithms. We show an example of prediction using TERs in Figure 1.2 (best viewed on a computer screen). We demonstrate, through experimental results on a variety of datasets, that TERs restrict the search space within images and thus improve the matching of feature points in a number of image sets. Any matched feature increases the set of known corresponding points, thus decreasing the size of the valid TERs in the rest of the images.

The main contributions of our work are (i) the introduction of the novel problem of predicting the location of moving points in a CrowdCam setting, (ii) the proposed extension of epipolar geometry in order to constrain the location of moving feature points in a set of images.

The remainder of this work is organized as follows: We begin by discussing works related to our research (Chapter 2). We then define spatial-temporal consistency and explain how it is used to define valid and invalid temporal epipolar regions in an image (Chapter 3). Subsequently we propose an algorithm to utilize TERs in order to predict the location of a corresponding point in a set of CrowdCam images (Chapter 4). Finally, we evaluate the effectiveness of our method on both simulated and real data (Chapter 5).

Chapter 2

Related Work

Crowd-based cameras capturing a dynamic scene have recently become the focus of a number of studies. Some studies focus on CrowdCam video sequences, such as Ballan *et al.* [4], who presented an algorithm for navigating around a dynamic performer using background reconstruction, camera pose calculation, and segmentation of the performer. Other studies utilize features found in still images captured by CrowdCam to extract information about the cameras and images. For instance, Dekel *et al.* [10, 11], provided a way to determine the time order between image captures using photo sequencing. Kanojia *et al.* [17] offered an alternative approach to determining the time order of the images using homography and identified which camera captured which image by recovering camera matrices using shared static points. Instead of ordering images in time, a recent work by Averbuch-Elor *et al.* [1] assumes that all images were captured at approximately the same time and uses the dissimilarity of objects to find the order of the images in space. While these works extract much information regarding the capturing of the scene, our work focuses on extracting information about the points captured in the scene, specifically how to predict their corresponding locations in the set of images.

Finding corresponding features among images is a topic which has been well studied and varies in its approaches based upon the specific goal of the work. Different types of feature descriptors may be used for different tasks, and the algorithms for finding matching features differ accordingly. For instance, while SIFT [20] or SURF [5] features work well when online calculation is not required, they are considered computationally expensive when compared to binary feature descriptors such as BRISK

[18] or ORB (rBRIEF) [30]. Each descriptor offers its own benefits and drawbacks and comparisons of many popular descriptors are available [23]. A common strategy for finding corresponding features involves searching for the nearest neighboring feature. For high dimensional features such as SIFT, libraries, such as FLANN [25], which select the best algorithm and parameters for efficient search for approximate nearest neighbors, have been developed. Algorithms which have been found to be efficient for approximate nearest neighbors search include randomized k-d forest or the priority k-means tree [24]. Binary descriptor nearest neighbor search is often done using variants of locality sensitive hashing, using Hamming distance for matching [30, 18, 8]. As described in the introduction, matching this way is descriptor dependent and does not take into account timing between image captures or the geometry between cameras. Our proposed method is orthogonal to direct feature matching as we utilize geometry to limit the search space. As such, each of these feature descriptors and matching strategies may be used in conjunction with our method for improved accuracy in matching.

There are approaches which consider more than the descriptors themselves in order to match features. One such approach is using bounded deformation models, as in Lipman *et al.* [19], to ensure that matched features are geometrically consistent between pairs of images. This work selects the candidate correspondences only if they can be aligned without too great a distortion of a triangular mesh. This method may be applied for matching features in CrowdCam captured data assuming that the dynamic objects captured do not deform greatly. As it does not take into consideration the timing between image captures, using our method to first remove temporally inconsistent candidate correspondences, may improve the accuracy of matched features, even in cases which look beyond the descriptors themselves for matching.

Another well known approach for finding correspondence is utilizing the geometry between cameras in order to limit the search space for matching points. This is not a new concept, in fact works dating back to the 1970's [14] detailed such procedures. Over the years many works have utilized epipolar constraints in order to determine correspondence of points in images for a number of applications [34, 13, 7]. A recent work by Shah *et al.* [31], uses epipolar geometry between images to limit the search space for correspondence in order to better find matches in the presence of repetitive structures. However, works which utilize epipolar constraints in order to search on or around epipolar lines for

correspondence, only apply to static points.

Finding correspondence in still images may also be accomplished by triangulating the 3D trajectory of moving points and then using this trajectory to project the point to each of the images. Avidan *et al.*, [2], described a method for trajectory triangulation for points moving along straight lines or conic sections. Kaminski and Teicher [16] extended this concept by building a general framework for trajectory triangulation of points which may move along more complex trajectories. However, both these methods require a full calibration of all the cameras and that a minimum of five correspondences (in the case of a precisely linear trajectory) be given. Park *et al.*, [28, 29] reconstruct the 3D trajectory of a moving point using a linear combination of compact trajectory basis vectors. This work assumes that camera positions, camera matrices and the precise timing of image captures are given. Our method does not require precise timing of image captures, only the order between the image capturing times. Additionally, our work requires a weaker calibration than these works (only fundamental matrices), and only three corresponding points as initialization. Note that using a forgiveness parameter (described in Chapter 4), our method relaxes the precisely linear trajectory assumption, allowing for predicting points moving approximately linearly.

In video, limiting the search space in order to find correspondence of a moving point or object has been studied in approaches for tracking utilizing a prediction of motion [32]. One prediction strategy which has helped trackers limit the search space for correspondence in successive frames is to search a window around the location of the point or object in a previous frame (e.g., [6, 3, 15, 33]). Another approach is to utilize a motion model in order to predict where an object will be based on its locations in previous frames. This approach often utilizes Kalman or Particle filters (e.g., [26, 9, 21, 22, 27]) in order to build the motion models. However, as these strategies for prediction assume short, consistent, time intervals between frames as well as no significant viewpoint change between frames, such strategies do not apply to CrowdCam image sets.

Chapter 3

Spatial-Temporal Consistent Regions

Assume a 3D point Q travels along a unidirectional linear trajectory and is projected to the set of images $\mathcal{I} = \{I_j\}$ at a set of unknown times $T = \{t(I_j)\}$. Let, $\hat{S}_q = \{q_j\}$ be the unknown projection of Q onto the set $\mathcal{I}$ such that q_j is the projection of Q onto image I_j at time $t(I_j)$. Given a known corresponding subset $S_q \subset \hat{S}_q$, our goal is to find the remainder of the set. Finding q_u in I_u requires overcoming possible ambiguities and may be computationally intensive. To narrow the search we propose a method for defining valid and invalid image regions where q_u can or cannot be located. We assume the fundamental matrices F_{ij} between pairs of image I_i and I_j can be calculated.

In order to define the said regions, we use the epipolar geometry between pairs of images and the temporal order of the set of images given by the permutation σ of the indices of $\mathcal{I}$. As such, $\sigma : \{1 \dots N\} \rightarrow \{1 \dots N\}$, such that $t(I_{\sigma(1)}) \leq t(I_{\sigma(2)}) \dots \leq t(I_{\sigma(N)})$.

3.1 Consistency Definitions

To best describe our method, we begin by defining the spatial-temporal consistency, STC, of a set of points. We then propose a method to determine valid and invalid regions using STC.

Definition 1: The set of points S_q is *spatial-temporally consistent* (STC) with a linear motion and σ *iff* the following conditions hold:

1. There exists a set of capturing times $T = \{t(I_j)\}$ which is consistent with the temporal order, σ.

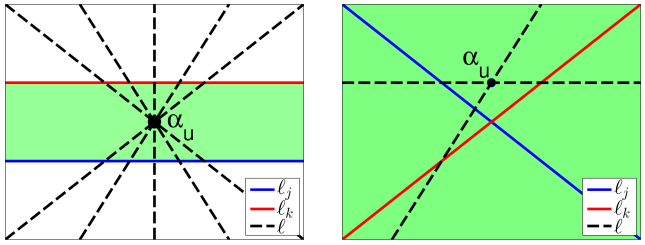


Figure 3.1: (a) As ℓ is unknown, candidate locations of α_u are limited to a region between the parallel lines. (b) With two nonparallel lines, there always exists an ℓ crossing α_u, which preserves σ (e.g., $\sigma = (j, u, k)$ or $\sigma = (u, j, k)$).

2. There exists a set of 3D points $S_Q = \{Q_j\}$ along a 3D line, L, such that q_j is the projection of Q_j onto image I_j at time $t(I_j)$.

3. The relative spatial locations of each point in S_Q along L correspond to the temporal order defined by permutation σ.

Note that if $t(I_i) < t(I_j) < t(I_k)$, then $Q_j \in L\langle Q_i, Q_k\rangle$. That is, Q_j is located on the interval of L between Q_i and Q_k. Therefore, (1) can be verified using the relative spatial locations of S_Q along L.

Assume that we are given S_q and we search for its unknown correspondence q_u in an additional image I_u. A point $\alpha_u \in I_u$ is a candidate location for q_u only if $S_q \cup \{\alpha_u\}$ is a STC set. Such a α_u is named a *valid spatial-temporally consistent point* with respect to S_q and σ or, for short, a *valid point*. We would like to consider the validity of a point while avoiding direct computation of the 3D set S_Q (and therefore L) and the timing set T. Instead, we propose a way to determine the validity of a point without computing L.

We note that it is possible to compute L through trajectory reconstruction, when at least five correspondences are known, and a full calibration of all cameras is available (e.g., [2, 16]). Our method requires a weaker calibration (only fundamental matrices), and only three corresponding points as initialization.

Consider the set of 2D points, $S^u = \{p_j\}$, the projections of S_Q onto a single image, I_u. That is, p_j is the projection of Q_j onto I_u. Note that the spatial order of S_Q along L is identical to the spatial order of the corresponding set S^u along ℓ^u, the projection of L onto I_u. Therefore, the temporal consistency

of S_q can be verified by the spatial order of S^u along ℓ^u. However, S^u and ℓ^u are both unknown.

That being said, as we know that $p_j \in I_u$ corresponds to the given $q_j \in S_q$, we can limit the location of $p_j \in I_u$ to the epipolar line ℓ_j on I_u, given by $\tilde{\ell}_j = F_{uj}\tilde{q}_j$ (where $\tilde{k}$ are the homogeneous coordinates of k). As such, let us consider the order of the intersections of a line ℓ with the epipolar lines defined by S_q and the point α_u. If the order of the crossings matches σ, we say that ℓ conserves σ. The point α_u is valid if there exists a line ℓ which conserves σ. Conversely, α_u is invalid if no such line exists.

Definition 2: (i) A *valid temporal epipolar region* is the set of all valid points in an image I_u with respect to S_q and σ. (ii) An *invalid temporal epipolar region* is the set of all invalid points in an image I_u with respect to S_q and σ.

In order to demonstrate how TERs and their validity are calculated efficiently, we take a closer look at the epipolar lines corresponding to S_q on I_u.

3.2 Two Epipolar Lines

First, let us discuss the degenerate case where we are given two parallel epipolar lines, ℓ_j and ℓ_k, on I_u and $\sigma = (j, u, k)$. Consider a point α_u in the section between ℓ_j and ℓ_k and any line ℓ, not parallel to ℓ_j, passing through it. Let α_j and α_k be the intersections of ℓ with the lines ℓ_j and ℓ_k. Clearly, α_u is on the interval between α_j and α_k on ℓ; hence the order of σ is preserved by ℓ and therefore α_u is a valid point (Figure 3.1a). This is true for all points in this section. However, this is not true for any points outside this section, and thus these two sections are invalid.

We note that in the more general case, in which the epipolar lines are not parallel, all candidate points in the image are valid (Figure 3.1b). We next show that, given more than 2 epipolar lines, it is possible to limit the valid regions.

3.3 Three Epipolar Lines

Given three epipolar lines, ℓ_i, ℓ_j, and ℓ_k, and a temporal order defined by σ, we can split the image plane of I_u into sixteen distinct temporal epipolar regions of five types (Figure 3.2). To do this, we

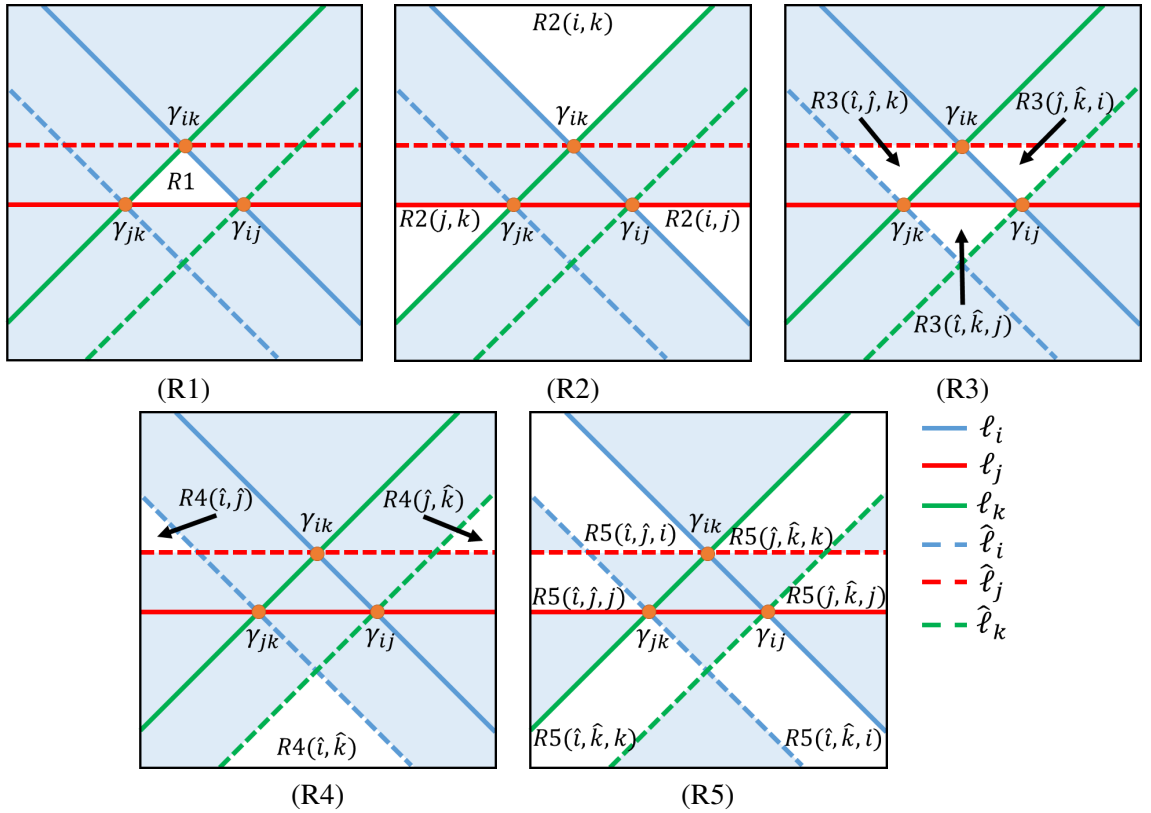


Figure 3.2: Three epipolar lines (ℓ_i, ℓ_j, ℓ_k) and their parallels ($\hat{\ell}_i$, $\hat{\ell}_j$, $\hat{\ell}_k$) split an image space into sixteen distinct regions of five types. Each region type is shown in white. The labels for each region are either contained within the region or point to the region using arrows.

consider an additional set of lines parallel to the epipolar lines. We define the line $\hat{\ell}_j$ as a line parallel to ℓ_j and passing through the intersection of ℓ_i and ℓ_k. The lines $\hat{\ell}_i$ and $\hat{\ell}_k$ are defined similarly. We will prove that in a given region either all points are valid or all points are invalid and that this classification is dependent not on the epipolar lines but on σ. We will show that for a given σ half of the 16 regions are valid TERs, while the other half are invalid. Table 3.1 summarizes the classifications of TERs as valid or invalid given σ, defined without loss of generality as having the suborder $\sigma' = (i, j, k)$ and four possible locations of u.

In practice, much of the efficiency of our method is derived from this classification scheme. As we know which TERs will be valid given a certain σ, there is no need to search for ℓ for every point α_u in each region. Instead, we can simply label regions as valid or invalid using Table 3.1.

We formally define all five region types here and prove their validity given different permutations.

The proofs consist of case analysis of standard geometric considerations. The reader may skip to Section 3.4 in a first reading, and check Table 3.1 and Figure 3.2 for region type reference.

Given epipolar lines ℓ_i, ℓ_j, and ℓ_k, let us label the intersection of ℓ_i and ℓ_j as γ_{ij} and label points γ_{ik} and γ_{jk} similarly. (Note that $\gamma_{ik} = \gamma_{ki}$.) Furthermore, for each epipolar line ℓ_j, let $\ell_j\langle\gamma_{ij}, \gamma_{ik}\rangle$ be the interval along ℓ_j from γ_{ij} and γ_{ik}. There are $4!/2$ possible orders up to direction that must be considered. We will show that 6 of them are valid and the other 6 are invalid for each region type.

Definition 3: *R1* is defined as the triangular region whose points are γ_{ij}, γ_{ik}, and γ_{jk} (Figure 3.2).

Claim 1: $R1$ is a valid TER *iff* u is not at an endpoint of σ.

Proof of Claim 1: Assume without loss of generality that $\sigma = (i, u, j, k)$. Consider the line ℓ which connects any point $\alpha_u \in R1$ and a point α_k on $\ell_k\langle\gamma_{jk}, \infty\rangle$ that does not contain γ_{ik} (i.e., $\alpha_k \notin R1$). Let α_i and α_j be the intersection of ℓ with ℓ_i and ℓ_j respectively. As $R1$ is defined as a closed convex shape (triangle), line ℓ passing through it must cross two of the lines which border $R1$. Additionally, the lines which are crossed at the borders of $R1$ must be ℓ_i and ℓ_j, as ℓ intersects ℓ_k at $\alpha_k \notin R1$. Furthermore, α_u must be on the interval defined by α_i and α_j, as α_i and α_j lie on borders of the region. Finally, as α_k is on the interval $\ell_k\langle\gamma_{jk}, \infty\rangle$, beginning at γ_{jk}, the line intersected between α_u and α_k must be ℓ_j. As such, ℓ conserves the order defined by σ and thus $R1$ is considered valid.

Assume w.l.g. that σ defines an order which has u at an endpoint. Consider the line ℓ passing through $\alpha_u \in R1$. As $R1$ is a closed convex shape, any line which passes through it must intersect two of its borders. Assume without loss of generality, it intersects with ℓ_i and ℓ_j at the intervals $\ell_i\langle\gamma_{ij}, \gamma_{ik}\rangle$ and $\ell_j\langle\gamma_{ij}, \gamma_{jk}\rangle$. However, such a line must have α_u between α_i and α_j, contradicting the order defined by σ, and thus this region must be invalid. QED.

We next consider another region type. Without loss of generality, two non-parallel lines dividing a 2D space split that space into four sections. A third line which is not parallel to either of the first two and does not intersect their intersection point must pass through three of these sections. The *unintersected section* is defined as the section through which the third line does not pass.

Definition 4: $R2(i, j)$ is defined as the unintersected section comprised of ℓ_i and ℓ_j through which ℓ_k does not pass. (See Figure 3.3).

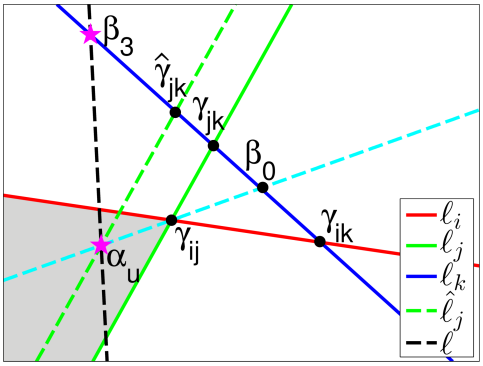


Figure 3.3: This figure shows the notations used in the proof of claim 2, $R2(i,j)$ marked in grey. The black line corresponds to an ℓ consistent with $\sigma = (j,u,i,k)$.

Claim 2: $R2(i,j)$ is a valid TER *iff* u is not adjacent to k in σ.

Proof of Claim 2: Let us first consider the 6 orders for which $R2(i,j)$ is invalid. Each of these orders has u adjacent to k. Assume without loss of generality that $\sigma = (i,u,k,j)$. Consider a line ℓ connecting any point α_u in $R2(i,j)$ and a point α_k. As α_k is outside the region $R2(i,j)$, any $\ell\langle\alpha_k,\alpha_u\rangle$ must cross either ℓ_i or ℓ_j. Hence, α_i or α_j must be between α_u and α_k, which contradicts orders having u adjacent to k.

We next consider orders in which $R2$ is valid (see Figure 3.3). Given a point $\alpha_u \in R2(i,j)$, define $\beta_0 \in \ell_k$ to be the intersection of the line connecting α_u and γ_{ij}. For all $\alpha_k \in \ell_k\langle\beta_0,\infty\rangle$ that do not contain γ_{ik}, α_i is between α_u and α_k, as α_i is on the border of $R2$. Hence α_u is consistent with the suborder $\sigma' = (u,i,k)$. We next show that there exists a line ℓ for every σ containing the suborder σ'.

Consider a line ℓ that connects α_u with any point $\beta_1 \in \ell\langle\beta_0,\gamma_{jk}\rangle$. As β_1 resides on a border of the triangle ($R1$), the interval $\ell\langle\beta_1,\alpha_i\rangle$ must cross ℓ_j at a point bordering $R1$ (as α_i is on the border of $R2$, it does not reside on the border of $R1$). Therefore, α_j must be between α_i and β_1. This line is therefore consistent with the full order $\sigma = (u,i,j,k)$.

Consider a line $\hat{\ell}_j$, parallel to ℓ_j and passing through α_u. Let $\hat{\gamma}_{jk}$ be the intersection of $\hat{\ell}_j$ with ℓ_k. Consider a line ℓ that connects α_u with any point $\beta_2 \in \ell\langle\gamma_{jk},\hat{\gamma}_{jk}\rangle$. As β_2 is on an interval between the two parallel lines, ℓ_j and $\hat{\ell}_j$, it follows that $\beta_2 \in \ell\langle\alpha_u,\alpha_j\rangle$.

Finally, consider a line ℓ that connects α_u with any point $\beta_3 \in \ell\langle\hat{\gamma}_{jk},\infty\rangle$ that does not contain γ_{jk}. As α_u resides on the intersection of ℓ and $\hat{\ell}_j$ and β_3 is at a point on $\ell\langle\hat{\gamma}_{jk},\infty\rangle$, α_j must be on the border

σ	**Valid TERs**
u, i, j, k	$R_2(i,j), R_2(i,k), R_3(i,\hat{j},\hat{k}), R_4(\hat{i},\hat{k}),$ $R_4(\hat{j},\hat{k}), R_5(i,\hat{i},\hat{k}), R_5(j,\hat{j},\hat{k}), R_5(k,\hat{k},\hat{j})$
i, u, j, k	$R_1(i,j,k), R_2(i,j), R_3(j,\hat{i},\hat{k}), R_3(k,\hat{i},\hat{j}),$ $R_4(\hat{i},\hat{j}), R_5(i,\hat{i},\hat{j}), R_5(i,\hat{i},\hat{k}), R_5(j,\hat{j},\hat{i})$
i, j, u, k	$R_1(i,j,k), R_2(j,k), R_3(j,\hat{i},\hat{k}), R_3(i,\hat{j},\hat{k}),$ $R_4(\hat{j},\hat{k}), R_5(j,\hat{j},\hat{k}), R_5(k,\hat{k},\hat{i}), R_5(k,\hat{k},\hat{j})$
i, j, k, u	$R_2(i,k), R_2(j,k), R_3(k,\hat{i},\hat{j}), R_4(\hat{i},\hat{j}),$ $R_4(\hat{i},\hat{k}), R_5(i,\hat{i},\hat{j}), R_5(j,\hat{j},\hat{i}), R_5(k,\hat{k},\hat{i})$

Table 3.1: This table defines, without loss of generality, the valid regions given a suborder of σ, $\sigma' = (i,j,k)$, and the location of $u \in \sigma$, all other regions are invalid.

of $R2$. As α_i and α_j must be on borders of R_2, the interval $\ell\langle\alpha_i, \alpha_j\rangle$ must cross α_u. Therefore, ℓ must be consistent with the full order $\sigma = (j,u,i,k)$. This case is shown in Figure 3.3. The symmetric cases where we switch i and j are proven similarly. QED.

Note that in the above proofs, we define α_u as any point within each region, and thus these proofs apply to all points within each region. Region types 3, 4, and 5 are proven in similar ways in Appendix A.

3.4 Beyond Three Epipolar Lines

When $|S_q| > 3$ (more than three images with known correspondences), we calculate valid TERs on I_u for each subset of three images with known correspondences, then find the intersection of all the valid TERs. Note that for each subset we use a subpermutation of σ that conserves the relative order between S_q and u. The intersection of all the valid TERs defines an overall valid region. Note that this overall valid region does not guarantee that there exists a line which conserves σ. Instead, it guarantees the correctness of the invalid regions. For an optimal computation of the valid region, it is necessary to define additional region types for each size of S_q. This is left for future work. While our method is not optimal, it yields good results, as can be seen in the experiments (Chapter. 5).

Chapter 4

TERs and Feature Matching

Now that we have defined TERs, we focus in this section on how to utilize them to improve feature matching algorithms, and to propose correspondence verification using prediction. As a first step of our method, image features are extracted from all images (we use SIFT features [20]) and the fundamental matrices F_{ij} between images $I_i \in \mathcal{I}$ and $I_j \in \mathcal{I}$ are computed (we use the BEEM algorithm [12]). Furthermore, if the temporal order of the images is unknown, it is possible to utilize photosequencing [11, 10] as an additional preprocessing step to calculate it. We also assume that a set of at least three correspondences is given, otherwise we can run a simple initialization to calculate it.

Matching using prediction: In each image, we extract feature points and run an initialization to calculate S_q. Given S_q and the set of fundamental matrices F_{ij}, we define TERs in each image for which correspondence remains unknown. For $|S_q| = 3$, the valid regions in each image are defined using Table 3.1. For larger $|S_q|$, the method described in Section 3.4 should be implemented to label the regions. The corresponding point, $\alpha_u \in I_u$, is chosen as the nearest neighbor to all features in S_q from valid regions in all images (note that any other matching criteria may be used). Then, S_q is updated to $S_q = S_q \cup \{\alpha_u\}$. The valid TERs in the remaining images are updated according to the additional constraints defined by α_u. This reduces the size of the valid regions in the remainder of the images. This process is repeated until all images are assigned correspondence. We present an example of matching using prediction in Figure 4.1.

Image 4 Image 8 Image 7 Image 6 Image 3

Figure 4.1: Given correspondences in the first, second, and fifth images of a scene (dataset c5), we run our method and find corresponding points in the remaining five images of the set using simple matching (red circles) and matching with TERs (green stars). Note: we labeled the images based on their order in time but presented them in the order, from left to right, in which best nearest neighboring matches were selected by the TER algorithm. Also, as matching with TERs is run independently of matching without TERs and each finds matches based on previously found correspondences, incorrect selections by standard matching may still fall within valid regions (as in image 7).

Correspondence verification: In some cases no valid regions exist in the image. This can be regarded as a *dead end*, since S_q cannot be extended. If we encounter such a dead end, we deduce either that one of our assumptions does not hold (linear motion, correct fundamental matrices and temporal order) and we can do no better than matching without TERs, or that one of the candidate correspondences is incorrect. This is demonstrated in test 3 in Chapter. 5. If we successfully assign points to all images without hitting a dead end, it is likely that points were selected correctly. Note that the more images we have, the higher the confidence that the matching is correct.

Unreliable TERs: The valid regions may be unreliable when the motion deviates from linearity or when the fundamental matrices are not accurate. To compensate for this, a simple *forgiveness parameter* may be used. Such a parameter allows for the consideration of points in invalid TERs as long as they are within a short distance to the border of a valid TER. The larger the forgiveness parameter, the more robust our method is in handling deviations from our assumptions, but the weaker the prediction. A demonstration of the effects of various forgiveness parameters on matching accuracy may be found in test 6 in Chapter 5.

Other cases of unreliability occur when the TERs are within the noise level due to the small sep-

aration between epipolar lines, for example, when the maximal distance between a pair of epipolar lines within the image is smaller than ϵ or when the distance between intersections of epipolar lines is less than ϵ'. In these cases, we may declare the TERs to be *unreliable*, and thus match without using them (by any standard matching algorithm). We propose the following workaround for these cases, only when there are $n > 3$ epipolar lines (otherwise, we declare *unreliability*). Given n epipolar lines, every valid point in the image is also valid in TERs built from every subset of $n - 1$ epipolar lines. We consider all subsets of epipolar lines in which no intersections are closer than ϵ' and the distance between each pair of lines is greater than ϵ. We calculate TERs for each such subset and use the union instead of the intersection of the resulting valid TERs. While this method may not restrict the overall valid region as with well separated epipolar lines, it does allow us to reliably restrict the search space.

Chapter 5

Experiments

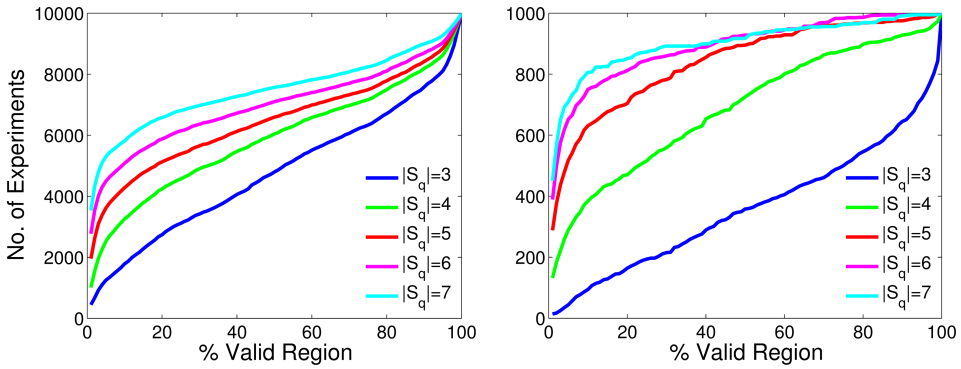
To study how our method performs in practice, we implemented it in MATLAB and tested it on synthetic and real data. Quantitative results were obtained by evaluating the reduction in search space. We also compared our method to one that does not use prediction.

5.1 Simulated Data

The simulated scenario consisted of a set of 4-8 cameras capturing a moving point. The cameras were positioned along a semisphere, all pointed approximately at the origin, through which a randomly generated line passes. Images were generated by projecting onto each image a randomly selected 3D point along the line, thus generating $\hat{S}_q$. The FOV of each camera was set to 30° in both x and y and the distance to the origin was approximately 550 units. Each image was 512 by 512 pixels.

Test 1: We tested the effectiveness of our method in reducing the percent of the image which is valid (PV). We ran 50,000 simulations, split into five groups of 10,000. Each group had a different number of cameras generated (between 4 and 8), all but one of which was given initial correspondences, such that $|S_q| = |\hat{S}_q| - 1$. The image I_u, which was not assigned a correspondence, was selected at random. In each simulation we computed valid TERs in I_u using S_q and calculated the PV.

The results are summarized in Figure 5.1(a) as cumulative histograms of PV, one for each number of initial correspondences, $|S_q|$. Our data shows that for $|S_q| = 3$, approximately 27% of the 10000



(a) (b)

Figure 5.1: (a) and (b) show, given different numbers of initial correspondences, the cumulative percentages of the area of valid TERs in each image, for simulated and real data, respectively.

simulations have images of which up to 20% is valid. When considering $|S_q| = 4$, approximately 42% of the simulations fall into this category. When $|S_q| = 7$, over 65% of our simulations restrict the search space to 20 % of the image or less. Thus, as expected, for a larger $|S_q|$, it is more likely that the search space for an additional correspondence will be smaller. Indeed, when more information about the moving point is available through a larger number of correspondences, the more accurate the prediction is. We do not present the cases in which $|S_q| >> 7$, as each additional image added to S_q can only further restrict the search space, therefore the trend presented is expected to continue.

Test 2: We tested the robustness of our method to noise in pixel locations, which may be caused by deviation from linear movement. We ran 5,000 simulations, using 6 generated cameras, three of which were given initial correspondences. We built TERs in the remaining images and selected one of them at random. If the point projected onto this image was in a valid region, we added it to S_q for the next iteration. Otherwise, if all the images had points in invalid regions, we stopped, as no match could be used for further iterations. Similarly, if an image reached a dead end, we stopped.

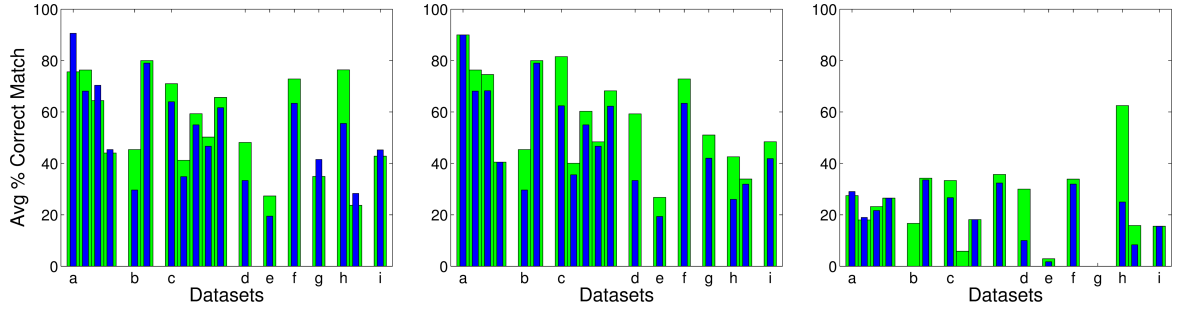
In each simulation we tested 4 noise levels per point on each image for which correspondence was unknown. The first was the original projected location, and the rest were the original projection shifted by a small random amount selected from a normal distribution with a mean of 0 and standard deviations of 1, 3, and 5 respectively. We set an upper bound on the noise such that no shift could be more than twice the standard deviation.

For each noise level used, we considered the number of simulations which successfully assigned a point to each image, the number of simulations which stopped as no further valid matches could be made, and the number of simulations in which a dead end occurred. We found that dead ends only occurred in 2 trials when the standard deviation was 5. At standard deviations of 0 and 1, all simulations were successful in assigning points to every image. When the standard deviation rose to 3, 0.54% of simulations had no candidate matches in valid regions and when the standard deviation was 5, 3.34% of simulations had no valid matches. Note that for this test no forgiveness parameter was used.

Test 3: We tested whether detections of dead ends can be used to identify incorrect correspondence. We ran 30,000 simulations, using between 4 and 6 cameras (10,000 simulations each) in which we initialized S_q with projections from the generated line (as in test 2), but selected random points for the rest of the images. For this test we used a forgiveness parameter (as described in Chapter. 4) of 2 pixels. We calculated valid TERs and proceeded through the images as in Test 2. The results are as follows: using 4 cameras we found 46.2% had no points in valid regions and 0.0% dead ends in the remaining simulations. With 5 cameras 74.81% had no matches, and of the remaining 26.38% were dead ends. With 6 cameras 70.98% had no matches, and of the remaining 66.37% were dead ends. As such, we note that given enough images, it is increasingly likely that incorrect correspondences will yield a dead end case.

5.2 Real Data

We evaluated our method on novel datasets captured at eight locations, by up to six smartphone cameras (e.g., Samsung Galaxy S4, Apple iPhone 5S). At each location between 1 and 7 datasets were captured from different viewpoints and at different times. Each dataset consists of between 5 and 15 images. In each dataset, we searched for correspondence for between 1 and 9 dynamic points, which were visible in all images. The locations, datasets, and points varied greatly in our experiments. Scenes were captured indoors and outdoors; some had many moving objects, while others had just one; and features considered for correspondence were on rigid (e.g., cars or soda cans) and non-rigid (e.g., people or dogs) objects. In addition to these novel datasets, we include in our results the rock climbing



(a) (b) (c)

Figure 5.2: The three graphs presented show the average percent correct matching per dataset with and without using TERs. In (a) no forgiveness is used, while in (b) and (c) the forgiveness parameter used is 25 pixels. In (a) and (b), S_q is given by ground-truth points, while (c) calculates S_q using an initialization. We see that using a forgiveness parameter improves the correctness of matching. Note that in these graphs, datasets are grouped by location, and each location is assigned a letter id (a-i). Each dataset in each location is also assigned a number, such that a1-a4 are presented in the first leftmost bars of each chart.

dataset supplied by [28]. From this dataset, we selected a partial set in which we consider only the unidirectional movement of the climber. Examples of finding correspondence in different datasets are presented in Figs. 5.5 - 5.11. Note that the examples shown were selected in part based on the size of their valid regions, as in many cases, the size of valid TERs is considerably smaller than can be easily viewed in a figure. In all datasets, the ground-truth correspondences ($\hat{S}_q$) and the time order among the images were identified manually.

Test 4: We repeated Test 1 in order to examine the percent of valid regions in natural still images as opposed to simulated data. The initial corresponding set of size $|S_q| = 3$ was given by the ground-truth. We ran our method and for each experiment selected at random a fourth image for which we calculated the valid TERs and PV. We followed a similar procedure given an initial set of $|S_q| = 4$, $|S_q| = 5$, $|S_q| = 6$, and $|S_q| = 7$. We selected at random 1000 samples of TERs calculated given each number of correspondences. The results (Figure 5.1(b)) show that on average, as in the simulated data, we see smaller valid regions when the number of correspondences increases. However, in the real data, for more than four correspondences, we found that over 85% of our samples had a PV of less than 40%. In general, natural image datasets restricted the search space better than simulated data, with the exception of when $|S_q| = 3$.

Test 5: We tested the robustness of our algorithm in finding correspondence and compared the results to the same matching algorithm without prediction by TERs. Matching was done by finding the nearest neighboring SIFT to those in S_q, using the scalar product between feature vectors.

Our method cannot restrict the search space of an image if dead ends and unreliable TERs occur. When these cases were detected, we stopped, having concluded that we could do no better than standard matching. We only present the cases in which no dead-ends or unreliable TERs were detected as those cases offer a fair comparison. However, we present statistics relating to dead ends in Appendix B, Table B.1. Note that for this test no forgiveness parameter (as described in Chapter. 4) was used.

As TERs are dependent on epipolar geometry, their effectiveness in restricting the search space is very dependent on the geometry between the images captured. Thus, we present the results of this test per dataset. Figure 5.2(a) shows the average percent of correct matchings using standard matching with and without our prediction. Each dataset consists of a number of experiments; the exact number depends on the number of points to be tracked in the scene and the number of combinations of initial correspondences. There are between 9 and 221 experiments in each dataset. In all datasets, we see that our TER assisted matching equals or outperforms standard matching on average, with an average improvement of 7.1% over all datasets. More detailed results are presented in Appendix B, Table B.1.

Test 6: We tested the effect of utilizing different forgiveness parameters (as described in Chapter. 4) using a similar method as in test 5 (given $|S_q| = 3$). We ran this test using four forgiveness parameter values: 10 pixels, 25 pixels, 50 pixels, and 100 pixels. Any SIFT feature which was within the number of pixels specified by the forgiveness parameter of a valid region is treated as valid and, therefore, considered for correspondence. In Figure 5.2(b) we present a comparison of matching with and without the use of TERs similarly to test 5, this time using a forgiveness parameter of 25 pixels. An example of such a case in which forgiveness was required for finding a correct match is presented in Figure 5.7 (Image 5).

It is important to note that forgiveness does not always improve matching correctness. For instance, if all our assumptions hold and the point moves precisely linearly, there is no need to use forgiveness. In such a case forgiveness may actually hurt matching accuracy when invalid points located close to valid regions are nearer neighbors to S_q than those within the valid regions. The effect of the tradeoff

between relaxing the assumptions (eg. accounting for slightly nonlinear movements) by increasing the forgiveness and the correctness of matching is shown for three datasets in Figure 5.3.

In our data, when comparing the results of a using forgiveness parameter to those of test 5 (no forgiveness), we found that adding even a small 10 pixel forgiveness improved the overall performance of our method. When considering the average percent of improvement (comparing correct matching with and without using TERs) over all trials in all datasets the we found that given no forgiveness, the average improvement was 2.77%. Using a 10 pixel forgiveness the average improvement was 5.18%, with a 25 pixel forgiveness 6.75%, and with a 50 pixel forgiveness 6.96%. Further increasing the forgiveness to 100 pixels results in a significant drop of the average improvement to 2.07%, highlighting the tradeoff discussed above.

We present a side by side comparison of the correct matching results for no forgiveness versus 25 pixel forgiveness per dataset in Figure 5.4. In this graph we see that in most of our datasets using forgiveness increases the matching correctness when comparing the use of TERs to standard matching). We note that as the number of correct matches changes, the number of dead ends encountered when using forgiveness changes as well. Therefore the results presented with and without forgiveness may compare different numbers of trials. In Appendix B, Tables B.2, B.3, B.4, and B.5, we present detailed results per dataset for each forgiveness parameter tested.

Test 7: We tested the effectiveness of our method similarly to test 5, given $|S_q| < 3$ and a forgiveness parameter of 25 pixels. To initialize S_q, we selected a point and found its two nearest-neighbors in the image set.

As the initial correspondences were built using matching alone, we expected the results to show more errors in the initial set. This expectation held true, as 222 of the 404 attempted experiments resulted in detection of dead ends. The remaining results are shown in Figure 5.2(c). Note that in this figure we do not consider the correctness of the S_q, but the correctness of the correspondences found from it. For each dataset represented in this figure, between 1 and 24 experiments were performed. Here we see that whether or not TERs are used, the percent of correct matchings drops significantly due to the lack of correct initial matches.

While our initialization method is not the state-of-the-art for matching, this test highlights that given

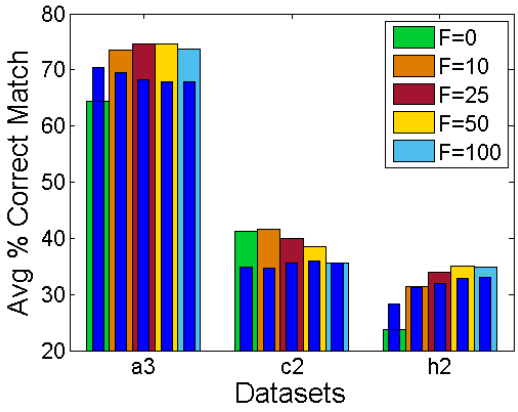


Figure 5.3: This graph shows compares the average percentage of correct matches with (wide bars) and without (narrow blue bars) using TERs for three datasets, when using different amounts of forgiveness (0, 10, 25, 50, 100 pixel forgiveness). In these cases we see that while a small forgiveness increases the accuracy of matching with TERs, the trade-off between forgiveness size and correct matching causes a drop in accuracy given larger forgiveness parameters.

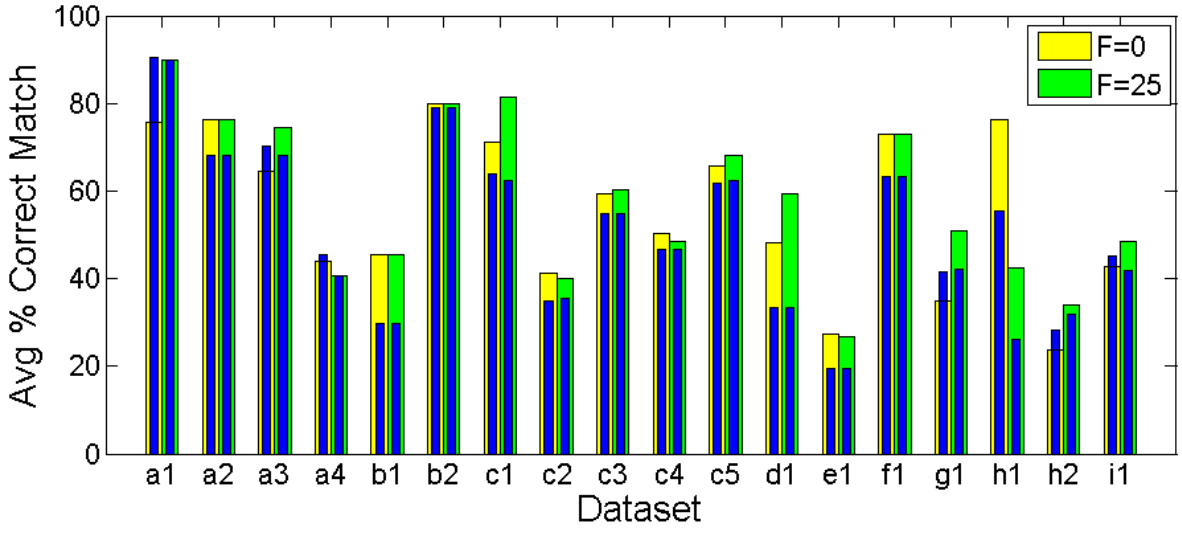


Figure 5.4: This graph shows compares the average percentage of correct matches with and without using TERs per dataset, when using no forgiveness and when using a forgiveness parameter of 25 pixels. Note that the percent of correct matches is calculated after removing dead ends (as those cannot be utilized for TER calculation). We see that in most datasets using forgiveness improves upon the percent correct matches.

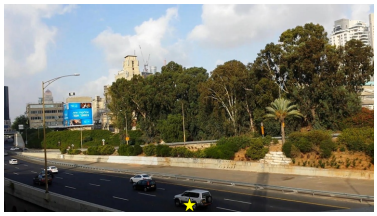


Image 2

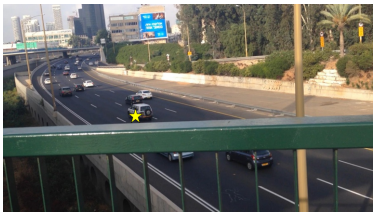


Image 4

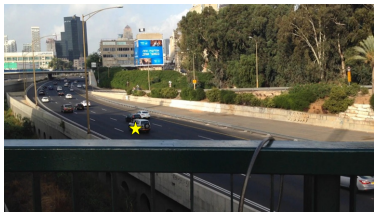


Image 5

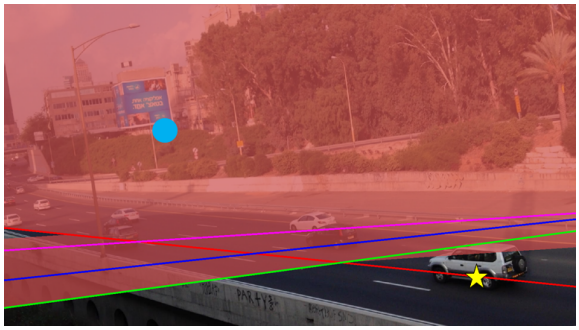


Image 1

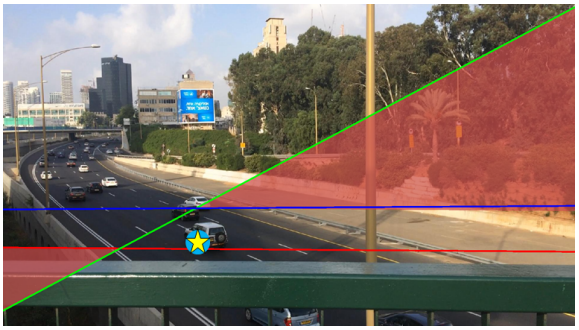


Image 3

Figure 5.5: Example results of finding correspondence in dataset a2. The top row of images are the given correspondences, the remainder of the images show the chosen correspondences. Correspondences selected using TERs are shown as yellow stars while those selected without are cyan circles. Note: images are labeled based on their order in time but presented in order, of best nearest neighboring matches as selected by the TER algorithm.

any initialization, we can still calculate TERs in order to improve matching. The average improvement over all the datasets is 1.8%. We present the detailed results of test 7 in Table B.6.

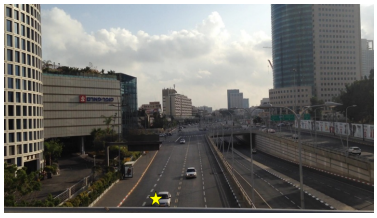


Image 1

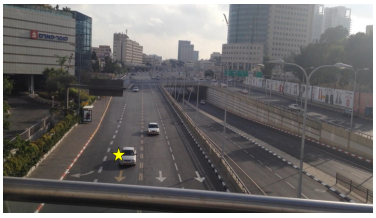


Image 2

Image 4

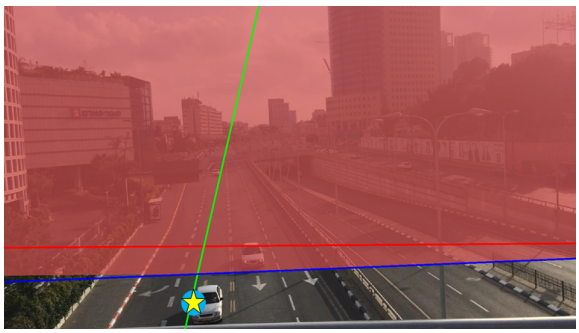


Image 5

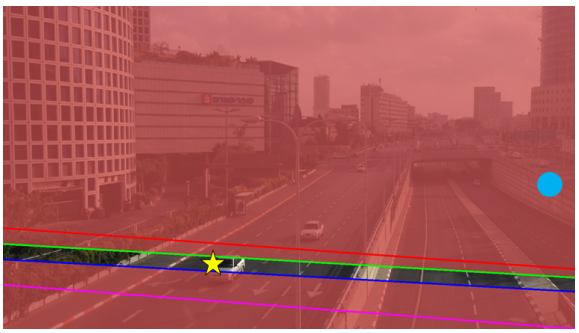


Image 3

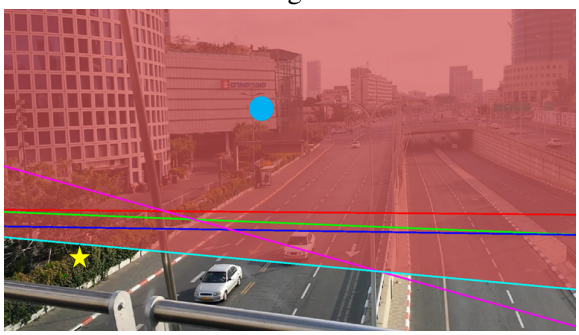


Image 6

Figure 5.6: Example results of finding correspondence in dataset b2. The top row of images are the given correspondences (the side mirror of the car), the remainder of the images show the chosen correspondences. Correspondences selected using TERs are shown as yellow stars while those selected without are cyan circles. In this example we see that an improvement over matching without TERs in image 3 and an incorrectly chosen matches both with and without TERs in image 6.

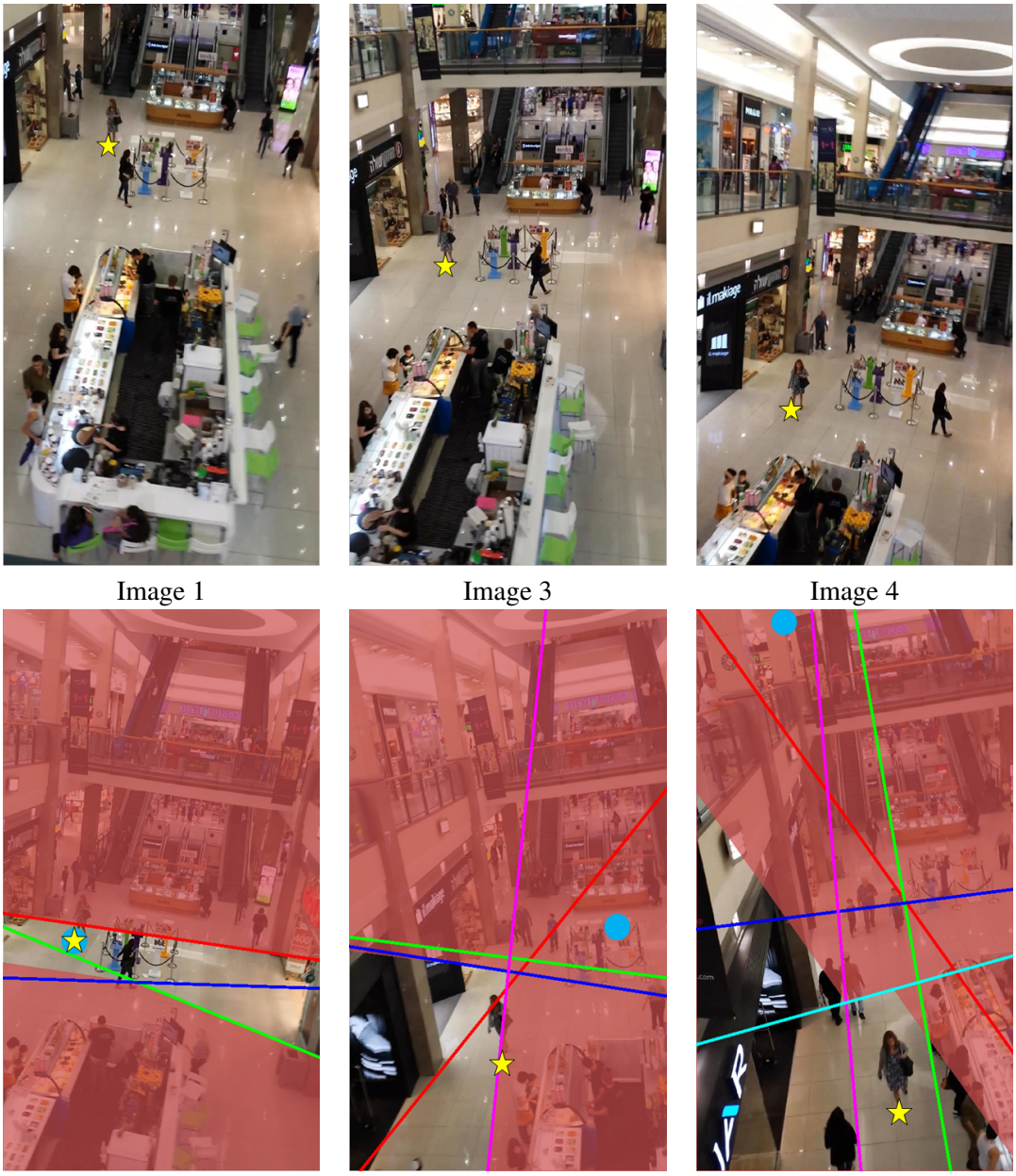


Figure 5.7: Example results of finding correspondence in dataset c1. This example demonstrates the utilization of a forgiveness parameter. The woman in the scene takes an awkward step, and thus her foot lands just outside of the valid region in image 5. However, as forgiveness is used, a correct match is still found. The top row of images are the given correspondences, the remainder of the images show the chosen correspondences. Correspondences selected using TERs are shown as yellow stars while those selected without are cyan circles.

Image 1

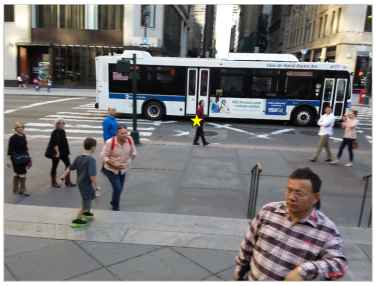


Image 2

Image 5

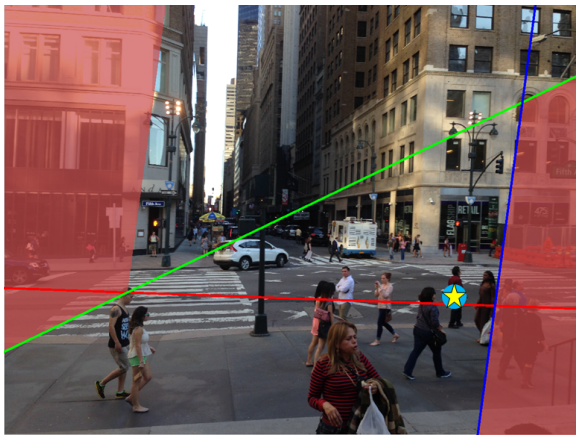


Image 4

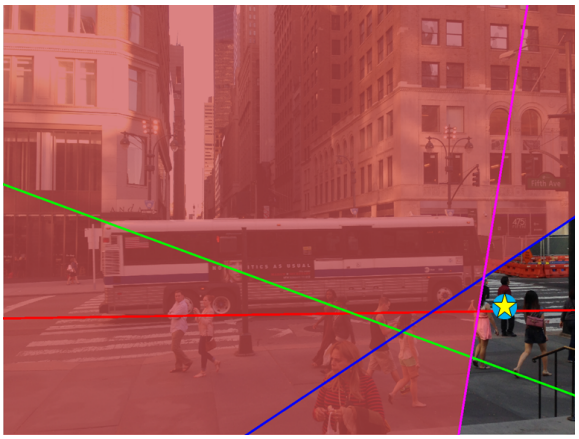


Image 6

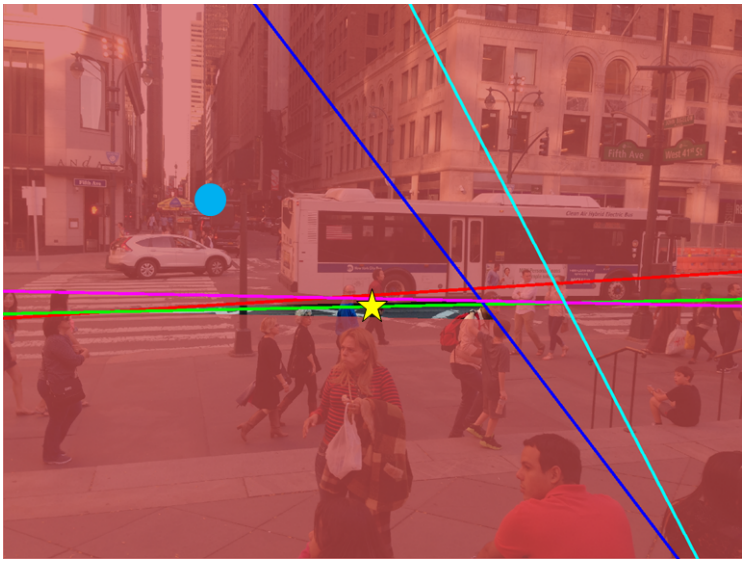


Image 3

Figure 5.8: Example results of finding correspondence in dataset d1. The top row of images are the given correspondences, the remainder of the images show the chosen correspondences. Correspondences selected using TERs are shown as yellow stars while those selected without are cyan circles.

Figure 5.9: Example results of finding correspondence in dataset e1. This example uses data from the rock climber dataset supplied by [28]. The top row of images are the given correspondences, the remainder of the images show the chosen correspondences. Correspondences selected using TERs are shown as yellow stars while those selected without are cyan circles.

Image 2

Image 4

Image 5

Image 3

Image 6

Image 7

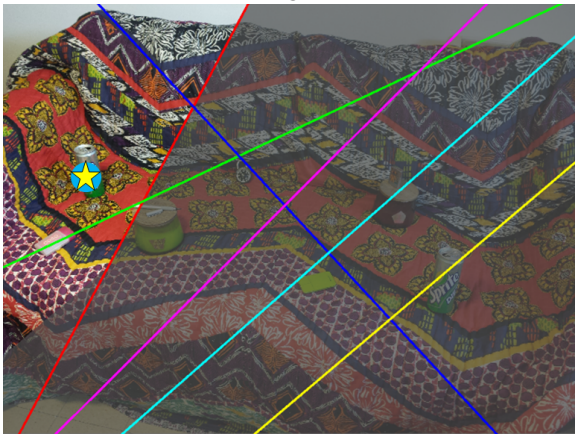


Image 1

Figure 5.10: Example results of finding correspondence in dataset f1. This data has manually manipulated, nearly identical objects (Sprite Cans), and highlights the improvement of using TERs over naive matching given similar features. The top row of images are the given correspondences, the remainder of the images show the chosen correspondences. Correspondences selected using TERs are shown as yellow stars while those selected without are cyan circles.

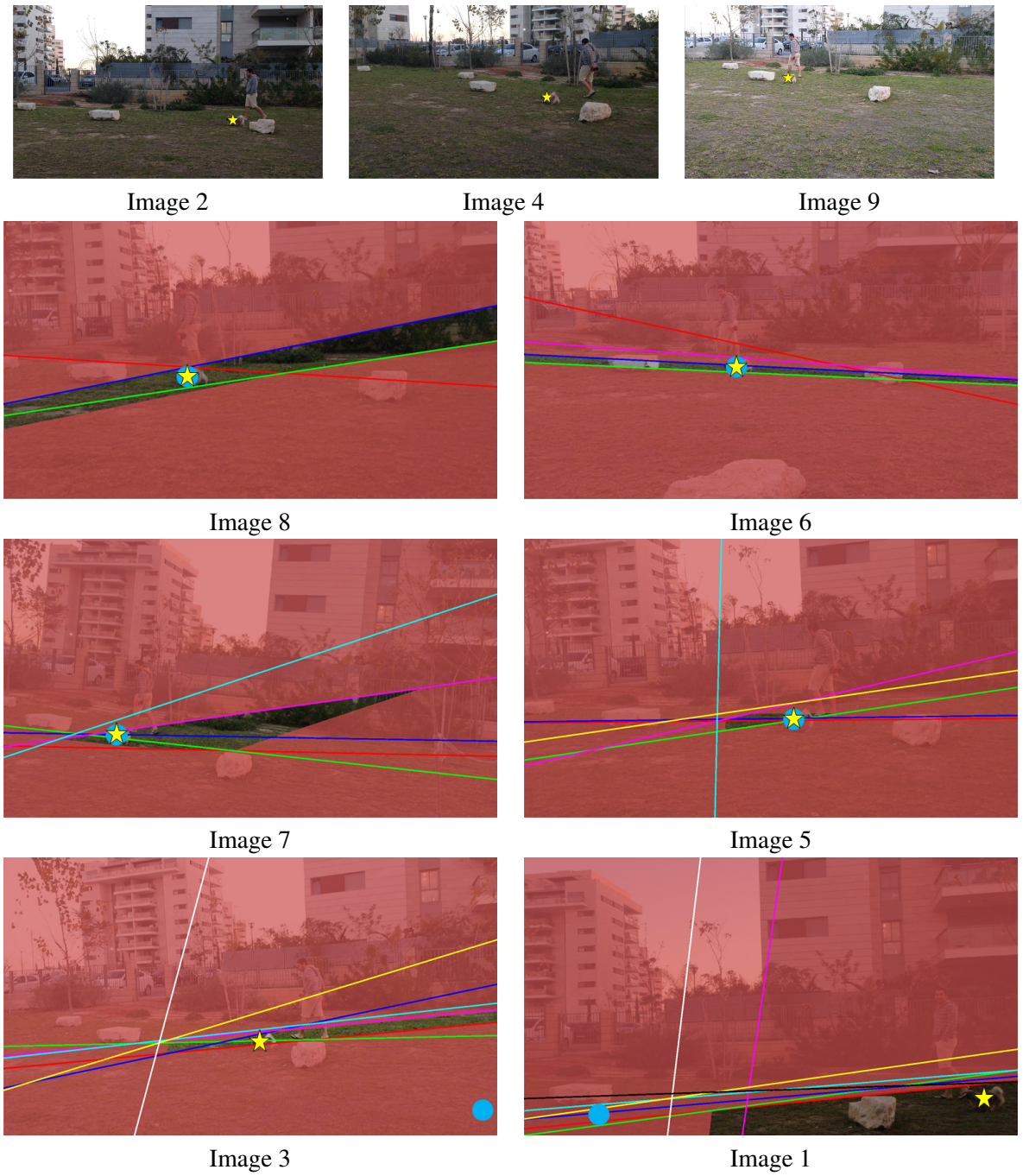


Figure 5.11: Example results of finding correspondence on a non-rigid object (dog) in dataset h1. The top row of images are the given correspondences, the remainder of the images show the chosen correspondences.

Chapter 6

Conclusions and Future Work

We proposed a method for improving feature matching accuracy when searching for correspondence of moving points in CrowdCam image sets. To that end, we introduced temporal epipolar regions which are used in order to predict correspondence of moving points by limiting the search space. We demonstrated on both simulated and varied real world data that this prediction improves the results of matching algorithms. Additionally, we showed how our method may be used to verify the accuracy of our assumptions and correspondence using dead ends. We note that the specific matching algorithm we used is relatively simple and by no means state-of-the-art, however, the proposed correspondence prediction can benefit any feature matching algorithm.

While this thesis introduces the basic concepts of TERs and utilizes them for finding correspondence, we note that there are many possible improvements to our method which we leave for future work, some of which are listed below.

- While our method limits the search space for correspondence well when $|S_q| = 3$, we do not define TERs for every $|S_q| > 3$ and instead consider all combinations of subsets of size 3. As discussed in Section 3.4, this yields regions in which every invalid point is inconsistent with σ, but there may be valid points which are inconsistent as well. Finding optimal regions for $|S_q| > 3$ is expected to yield better search space reduction, and therefore, better matching and dead end detection.

- The method we introduced builds TERs in an image I_u for which correspondence is unknown using information in all images for which correspondence is known. We note that it is possible to further restrict the valid region of I_u given another image I_v for which correspondence is unknown (and TERs are calculated). As we know that one of the valid points in I_v is the corresponding point, we can check the effect of all the valid points on I_v on the TERs of I_u. If we consider all the valid points in I_v, treating each as if it were the correct correspondence and thus use it to generate TERs on I_u, we can check if there are any points in I_u which are invalid for all possible candidate points from I_v. Such points can therefore be labeled as invalid as they cannot be consistent with any point from I_v. Similarly, we can consider all valid points in I_v to further restrict the valid TERs in I_u. This process is computationally expensive, and we leave finding an efficient implementation of this additional restriction for future work.

- In our implementation, we assumed that there were no occlusions in any of our images and always selected the best match regardless of how poor a match it was. However, as described above, we may be able to utilize the information in images for which correspondence is unknown to further restrict the search space in the remainder of the images. As such, removing poor matches which may reflect occlusions by using thresholding can prove very useful in combination with the previous item. Implementation improvements such as poor match removal are left for future work.

In addition to the proposed future work to enhance and improve our method, there are a number of additional uses for TERs which are left to future work, examples of which are listed below.

- In this thesis, we applied our method to CrowdCam image data, however, we can extend the usage of TERs to both static and dynamic video data. We note that it is fairly straightforward to extend our method to video, however, we note that when the cameras are static as the geometry between cameras does not change, and thus, the preprocessing is significantly more efficient. Assume we are given four static video cameras and correspondences in at least one frame in three of the cameras. As in still images, we calculate and use TERs to predict correspondence in a frame of the fourth video. To proceed in predicting the location of correspondence, we would

consider sets of frames from three videos at a time to predict the location of the point in a frame from the fourth. Note that we may use multiple correspondences from each camera to restrict the search space, however, a maximum of two will suffice as the object moves linearly. Using TERs in this way may improve upon tracking methods in video but is beyond the scope of this thesis and is thus left for future work.

- In this thesis we only consider finding correspondence of single points. An important extension to our method involves predicting the location of whole objects, superpixels, or patches, as opposed to individual feature points. Note that a simple implementation to accomplish this task is to find the union of valid TERs from all feature points of the object. Nevertheless, such an implementation often results in a very large valid regions.

- In addition to predicting correspondence, TERs may be used for a number of other tasks. For instance, in calculating segmentation of a dynamic object in the scene, TERs may be used in order to remove noise. All areas which are invalid to all points on the foreground object may be ruled out for segmentation and thus may be removed.

- Using dead ends for additional tasks is also left to future work. For instance, developing algorithms that may utilize the dead end detection for improving correspondence.

Finally, we leave a number of open questions relating to overcoming the assumptions which our method relies on to future work. For instance, how to detect non-unidirectional motion of the point using spatial-temporal inconsistency.

To summarize, the method we introduced in this thesis provides a foundation for an expanded usage of epipolar geometry to analyze the motion of dynamic objects through scenes. While this foundation lays the groundwork for improvements and optimizations of TERs, the method presented to predict correspondence already improves upon any feature matching algorithm. We feel that this work is the start of an exciting new chapter in motion analysis.

Appendix A

Proofs Continued

Here we complete the formal definition and proofs of region types R3, R4, and R5.

Definition 5: $R3(k, \hat{i}, \hat{j})$ is defined by the triangle whose points are γ_{ik}, γ_{jk}, and $\hat{\gamma}_{ij}$ (see Figure 3.2).

Definition 6: $R4(\hat{i}, \hat{j})$ is defined as the unintersected section comprised of $\hat{\ell}_i$ and $\hat{\ell}_j$ through which no other line passes (see Figure 3.2).

Definition 7: $R5(i, \hat{i}, \hat{j})$ is defined as the open section bordered by ℓ_i, $\hat{\ell}_i$, and $\hat{\ell}_j$ through which no other line passes (see Figure 3.2).

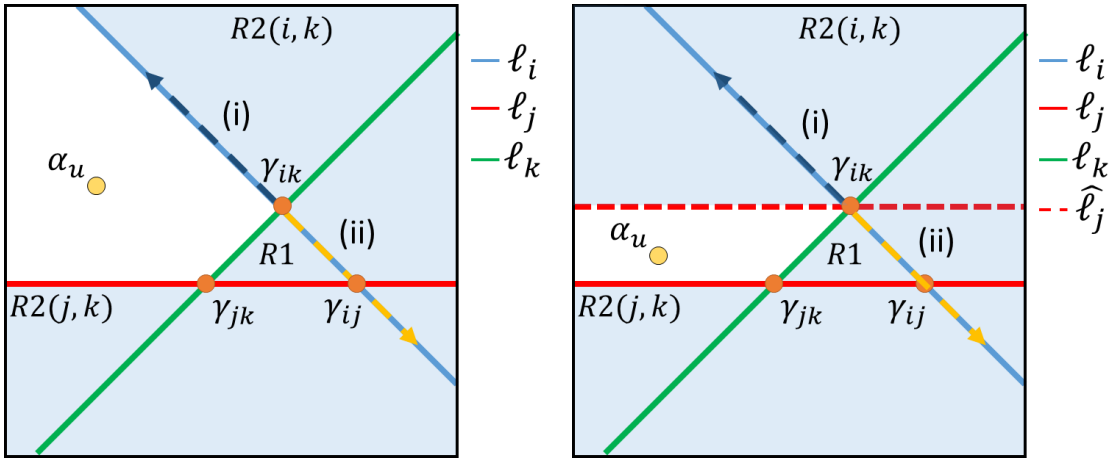
We now prove the validity of each region given different permutations of σ. We begin these proofs by considering the cases in which the regions of $R3(k, \hat{i}, \hat{j})$, $R4(\hat{i}, \hat{j})$, $R5(i, \hat{i}, \hat{j})$, and $R5(j, \hat{j}, \hat{i})$ behave similarly. We term the union of these regions $R345$.

Note that given any set of three nonparallel lines which do not intersect at the same point, there are three regions which can be defined similarly to $R345$. Here we prove only one of the three as the others are be proven similarly.

Claim 3: All points in $R345$ are invalid given $\sigma = (u, i, j, k)$.

Proof of Claim 3:

We have to show that there exists no ℓ which crosses all the epipolar lines through any point $\alpha_u \in R345$ in the order defined by σ. For reference, see Figure A.1.



(Claims 3, 5) (Claim 8)

Figure A.1: The invalid cases. $R345$ is shown here in white and the two cases for choosing α_i, described in the proof of claims 3, 5, and 8 are shown in blue and yellow dashed arrows.

(i) Consider a line ℓ which connects any α_u in $R345$ and $\alpha_i \in \ell_i\langle\gamma_{ik}, \infty\rangle$ not containing γ_{ij}. As α_i is on the border of R2(i,k), α_i must either be the last (or first) point on ℓ, or it must be adjacent to α_k as ℓ_j does not pass through R2(i,k). Both cases are not consistent with the given σ.

(ii) Consider a line ℓ connecting α_u in $R345$ and $\alpha_i \in \ell_i\langle\gamma_{ik}, \infty\rangle$ containing γ_{ij}. Given such a α_i, α_k must be between α_u and α_i along ℓ as α_i and α_u are defined on opposite sides of ℓ_k.

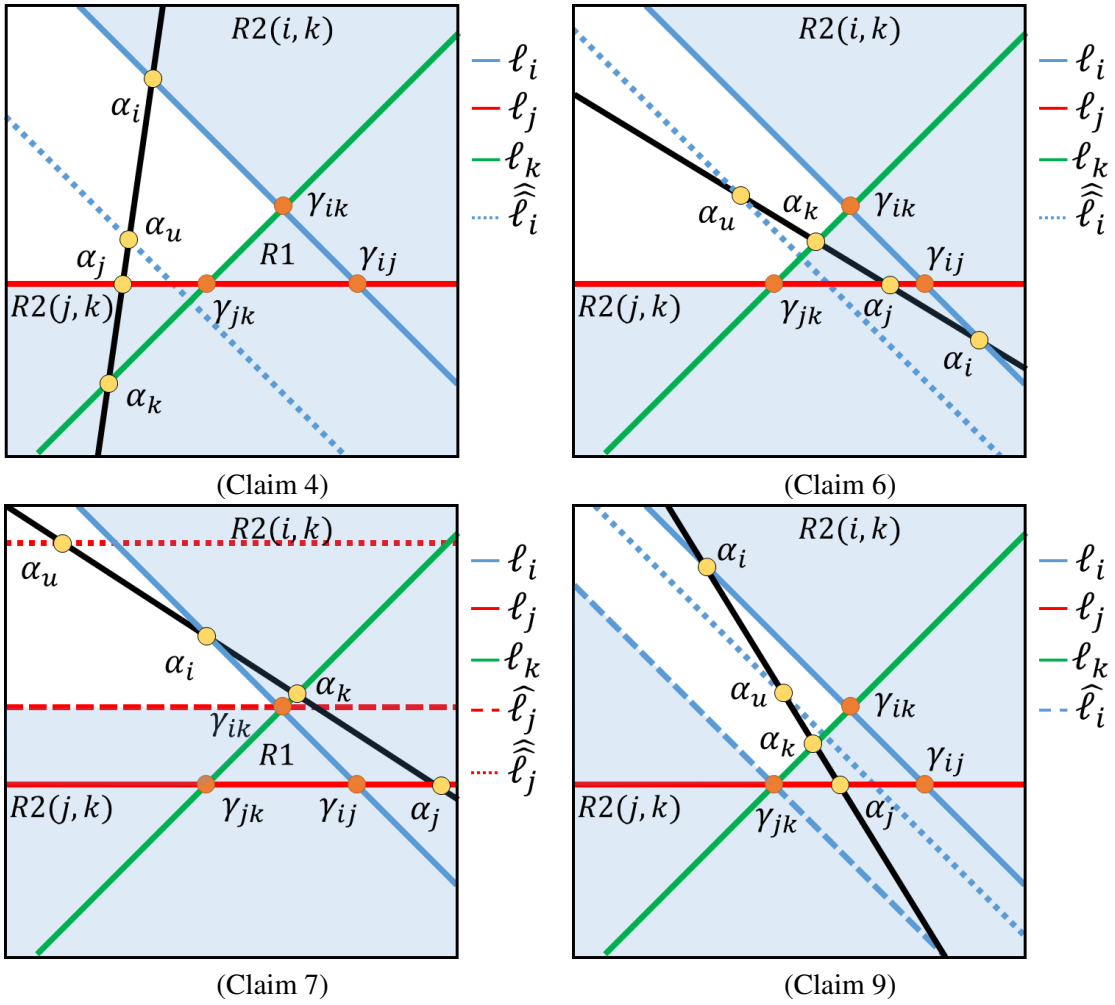
These two considerations show there is no point on ℓ_i which does not contradict σ. QED

Claim 4: All points in $R345$ are valid given $\sigma = (i, u, j, k)$.

Proof of Claim 4:

We have to show that for any point $\alpha_u \in R345$, there exists a line ℓ which crosses the epipolar lines and α_u in the order defined by σ. For reference, see Figure A.2.

Define the line $\hat{\hat{\ell}}_i$ to be a line parallel to ℓ_i passing through any point $\alpha_u \in R345$. If the intersection of $\hat{\hat{\ell}}_i$ with ℓ_k is on a border of $R1$, define the point $\alpha_k \in \ell_k\langle\gamma_{jk}, \infty\rangle$ not containing γ_{ik}. Otherwise define the point $\alpha_k \in \ell_k\langle\hat{\hat{\ell}}_i \times \ell_k, \infty\rangle$ not containing γ_{jk}. Consider the line ℓ connecting α_k and α_u. As α_k is on the border of $R2(j, k)$ but not on the border of $R345$, α_j must be between α_k and α_u along ℓ. Furthermore as α_j must be outside of the strip defined by $\hat{\hat{\ell}}_i$ and ℓ_i and α_j borders $R345$, α_u must be between α_j and α_i.



(Claim 4) (Claim 6)

(Claim 7) (Claim 9)

Figure A.2: $R345$ is shown here in white, and the black line is one possible ℓ which demonstrates that the point α_u is valid as described in proof of claims 4, 6, 7, and 9, respectively.

Claim 5: All points in $R345$ are invalid given $\sigma = (i, j, u, k)$.

Proof of Claim 5:

We have to show that there exists no ℓ which crosses all the epipolar lines through any point $\alpha_u \in R345$ in the order defined by σ. For reference, see Figure A.1.

(i) Consider a line ℓ which connects any α_u in $R345$ and $\alpha_i \in \ell_i\langle\gamma_{ik}, \infty\rangle$ not containing γ_{ij}. As α_i is on the border of R2(i,k) and $R345$, α_i cannot be adjacent to α_j along ℓ as ℓ_j does not pass through R2(i,k), and α_u must be adjacent to α_i as $\alpha_u \in R345$.

(ii) Consider a line ℓ connecting α_u in $R345$ and $\alpha_i \in \ell_i\langle\gamma_{ik}, \infty\rangle$ containing γ_{ij}. As in proof 3, α_k must be between α_u and α_i.

These two considerations show there is no point on ℓ_i which does not contradict σ. QED

Claim 6: All points in $R345$ are valid given $\sigma = (i, j, k, u)$.

Proof of Claim 6:

We have to show that for any point $\alpha_u \in R345$, there exists a line ℓ which crosses the epipolar lines and α_u in the order defined by σ. For reference, see Figure A.2.

Define the line $\hat{\hat{\ell}}_i$ to be a line parallel to ℓ_i passing through any point α_u in $R345$.

If $\hat{\hat{\ell}}_i$ intersects ℓ_j in $\ell_j\langle\gamma_{jk}, \gamma_{ij}\rangle$, then define the point $\alpha_j \in \ell_j\langle\hat{\hat{\ell}}_i \times \ell_j, \gamma_{ij}\rangle$. Otherwise $\alpha_j \in \ell_j\langle\gamma_{jk}, \gamma_{ij}\rangle$.

Consider a line ℓ which connects any α_u in $R345$ and α_j. As $\alpha_u \in \hat{\hat{\ell}}_i$ and α_j is within the strip defined by $\hat{\hat{\ell}}_i$ and ℓ_i, α_j must be between α_i and α_u. Furthermore α_j is on a border of $R1$ and α_i is outside of $R1$, α_k must be on the border of $R1$. Furthermore, as α_i and α_j are outside $R345$, α_k must be on the border of $R345$, between α_u and α_j. Such an ℓ is consistent with the order defined by σ.

For each of the above $R345$ proofs, the symmetrical cases (in which i and j are swapped in σ) are proven similarly. We now proceed to define cases in which not all regions of $R345$ are labeled the same.

Claim 7: The region $R4(\hat{i}, \hat{j}) \cup R5(i, \hat{i}, \hat{j})$ is valid given $\sigma = (u, i, k, j)$.

Proof of Claim 7:

We have to show that for any point $\alpha_u \in R4(\hat{i}, \hat{j}) \cup R5(i, \hat{i}, \hat{j})$, there exists a line ℓ which crosses the epipolar lines and α_u in the order defined by σ. For reference, see Figure A.2.

Define the line $\hat{\hat{\ell}}_j$ to be a line parallel to ℓ_j passing through any point $\alpha_u \in R4(\hat{i}, \hat{j}) \cup R5(i, \hat{i}, \hat{j})$.

Consider the line ℓ which connects α_u and $\alpha_i \in \ell_i\langle\gamma_{ik}, \hat{\hat{\ell}}_j \times \ell_i\rangle$. As α_i is on a border of the triangle whose points are γ_{ik}, $\gamma_{i\hat{j}}$, and $\gamma_{\hat{j}k}$, and α_i is in the strip defined by ℓ_j and $\hat{\hat{\ell}}_j$, α_k must also be on the border of the triangle. As both α_i and α_k are within the strip they must also be between α_u and α_j. Furthermore, α_i must be between α_u and α_k as α_i resides on a border of $R345$ and $R2(i, k)$ while α_k is only on the border of $R2(i, k)$.

Claim 8: The region $R3(k,\hat{i},\hat{j}) \cup R5(j,\hat{j},\hat{i})$ is invalid given $\sigma = (u,i,k,j)$.

Proof of Claim 8:

We have to show that there exists no ℓ which crosses all the epipolar lines through any point $\alpha_u \in R3(k,\hat{i},\hat{j}) \cup R5(j,\hat{j},\hat{i})$ in the order defined by σ. For reference, see Figure A.1.

(i) Consider a line ℓ connecting any α_u in $R3(k,\hat{i},\hat{j}) \cup R5(j,\hat{j},\hat{i})$ and a point $\alpha_i \in \ell_i\langle\gamma_{ik},\infty\rangle$ not containing γ_{ij}. As α_u is within the strip defined by ℓ_j and $\hat{\ell}_j$ and α_i is outside of this strip, α_u must be between α_j and α_i.

(ii) Consider a line ℓ connecting α_u and $\alpha_i \in \ell_i\langle\gamma_{ik},\infty\rangle$ containing γ_{ij}. Given such a α_i, α_k must be between α_u and α_i along ℓ as α_i is defined on ℓ_i from the intersection point γ_{ik} therefore, α_u and α_i are on opposite sides of ℓ_k.

These two considerations show there is no point on ℓ_i which does not contradict σ. QED

Note that the case in which $\sigma = (i,k,j,u)$ is symmetrical to $\sigma = (u,i,k,j)$, and thus is proven similarly.

Claim 9: The region $R3(k,\hat{i},\hat{j}) \cup R5(i,\hat{i},\hat{j})$ is valid given $\sigma = (i,u,k,j)$.

Proof of Claim 9:

We have to show that for any point $\alpha_u \in R3(k,\hat{i},\hat{j}) \cup R5(i,\hat{i},\hat{j})$, there exists a line ℓ which crosses the epipolar lines and α_u in the order defined by σ. For reference, see Figure A.2.

Define the line $\hat{\hat{\ell}}_i$ to be a line parallel to ℓ_i passing through any point α_u in $R3(k,\hat{i},\hat{j}) \cup R5(i,\hat{i},\hat{j})$.

Consider the line ℓ connecting $\alpha_u \in R3(k,\hat{i},\hat{j}) \cup R5(i,\hat{i},\hat{j})$ and a point $\alpha_k \in \ell_k\langle\hat{\hat{\ell}}_i \times \hat{\ell}_k, \gamma_{kj}\rangle$. As α_k is at a point outside of the strip bordered by $\hat{\hat{\ell}}_i$ and ℓ_i and α_u is on $\hat{\hat{\ell}}_i$, α_u must be between α_i and α_k. Furthermore, as α_k is on the border of $R1$ and α_i is not, α_j must also be on the border of $R1$ and α_k must be between α_u and α_j. Such an ℓ therefore preserves the order defined by σ.

Claim 10: The region $R4(\hat{i},\hat{j}) \cup R5(j,\hat{j},\hat{i})$ is invalid given $\sigma = (i,u,k,j)$.

Proof of Claim 10:

We have to show that there exists no ℓ which crosses all the epipolar lines through any point $\alpha_u \in R4(\hat{i},\hat{j}) \cup R5(j,\hat{j},\hat{i})$ in the order defined by σ. For reference, see Figure A.3.

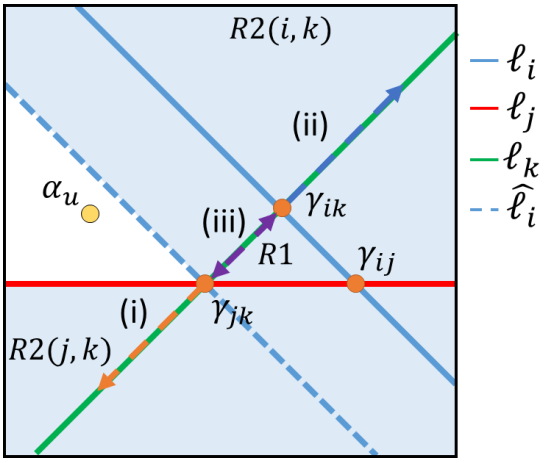


Figure A.3: $R4(\hat{i},\hat{j}) \cup R5(j,\hat{j},\hat{i})$ is shown here in white and the three cases described in the proof of claim 10 are shown as dashed arrows.

(i) Consider a line ℓ connecting $\alpha_u \in R4(\hat{i},\hat{j}) \cup R5(j,\hat{j},\hat{i})$ and a point $\alpha_k \in \ell_k\langle\gamma_{jk},\infty\rangle$ not containing γ_{ik}. As α_k borders $R2(j,k)$, α_j must be between α_u and α_k, thus ℓ does not the preserve the order of σ.

(ii) Similarly, consider ℓ connecting $\alpha_u \in R4(\hat{i},\hat{j}) \cup R5(j,\hat{j},\hat{i})$ and a point $\alpha_k \in \ell_k\langle\gamma_{ik},\infty\rangle$ not containing γ_{jk}. As α_k borders $R2(i,k)$, α_i must be between α_u and α_k, thus ℓ does not the preserve the order of σ.

(iii) Consider ℓ connecting $\alpha_u \in R4(\hat{i},\hat{j}) \cup R5(j,\hat{j},\hat{i})$ and a point $\alpha_k \in \ell_k\langle\gamma_{jk},\gamma_{ik}\rangle$. As α_u is outside the strip defined by $\hat{\ell}_i$ and ℓ_i and α_k is within this strip, α_k must be between α_u and α_i, thus ℓ does not the preserve the order of σ.

As there is no point on ℓ_k through which an ℓ can pass and preserve the order of σ, this region must be invalid.

Note that the case in which $\sigma = (j,u,k,i)$ is symmetrical to $\sigma = (i,u,k,j)$, and thus is proven similarly.

Appendix B

Detailed Results

This appendix presents tables which contain more detailed results than were shown in Chapter 5. All the tables have the same structure. Each dataset in the first column corresponds to a location, given by the letters a-i, and different image sets taken at each location at different times, given by the numbers 1-5. The number of images and the number of points to be matched vary between datasets and are thus listed in the table as well. We tested our method a number of times on each dataset, each with a different combination of initial points summarized in the no. of combinations column. For each dataset we also present the percent of the tests in which dead ends were encountered. Note that, as described in Chapter 4, when encountering dead ends, we deduce that we do no better than matching without TERs. As such, the rest of the columns present results from the remainder of the data, which has no dead ends. We present the percent correct matching with and without the use of TERs over all data in each dataset. We also present the difference in the percent of correct matching between matching with and without TERs for the points in which for better or worse the use of TERs changed the results. This difference is presented in the average % diff. column.

Dataset	No. of Images	No. of Points	No. of Combinations	% Dead End	% Matches Without TERs	% Matches With TER	Average Diff. %
a1	5	4	40	5.0	90.6	75.6	-20.0
a2	5	5	50	2.0	68.1	76.3	13.7
a3	6	9	124	31.5	70.4	64.5	-7.6
a4	10	2	125	93.6	45.4	44.1	-2.9
b1	7	1	34	20.6	29.6	45.4	15.7
b2	6	5	100	2.0	79.0	80.0	5.0
c1	6	3	59	23.7	64.0	71.0	7.0
c2	7	4	127	38.6	34.8	41.2	6.4
c3	6	4	74	2.7	55.0	59.3	4.3
c4	6	2	38	2.6	46.7	50.2	3.6
c5	8	5	242	23.6	61.7	65.7	4.0
d1	6	1	18	50.0	33.3	48.1	14.8
e1	6	4	80	15.0	19.5	27.3	15.7
f1	7	6	208	16.3	63.4	72.9	11.3
g1	8	4	135	86.7	41.5	34.9	-6.6
h1	9	2	122	90.2	55.6	76.4	20.8
h2	7	3	98	50.0	28.3	23.7	-4.6
i1	6	4	63	33.3	45.2	42.8	-2.4

Table B.1: This table presents the results of test 5, in which three initial correspondences are given manually.

Dataset	No. of Images	No. of Points	No. of Combinations	% Dead End	% Matches Without TERs	% Matches With TER	Average Diff. %
a1	5	4	40	0	90.0	90.0	
a2	5	5	50	2.0	68.1	76.3	13.7
a3	6	9	124	19.4	69.5	73.6	9.0
a4	10	2	125	71.2	41.3	41.3	
b1	7	1	34	20.6	29.6	45.4	15.7
b2	6	5	100	2.0	79.0	80.0	5.0
c1	6	3	59	28.8	62.4	78.0	15.6
c2	7	4	127	37.8	34.6	41.6	9.2
c3	6	4	74	2.7	55.0	59.8	4.8
c4	6	2	38	2.6	46.7	48.4	1.7
c5	8	5	242	19.0	62.3	64.3	1.9
d1	6	1	18	50.0	33.3	48.1	14.8
e1	6	4	80	15.0	19.4	26.8	29.6
f1	7	6	208	16.3	63.4	72.9	11.3
g1	8	4	135	88.9	43.3	46.9	5.0
h1	9	2	122	67.2	26.1	46.8	20.7
h2	7	3	98	49.0	31.2	31.3	0.2
i1	6	4	63	25.4	43.0	49.4	13.5

Table B.2: This table presents the results of test 6, in which 3 correspondences were given and a forgiveness parameter of 10 pixels was set.

Dataset	No. of Images	No. of Points	No. of Combinations	% Dead End	% Matches Without TERs	% Matches With TER	Average Diff. %
a1	5	4	40	0	90.0	90.0	
a2	5	5	50	2.0	68.1	76.3	13.7
a3	6	9	124	15.3	68.3	74.6	17.7
a4	10	2	125	68.8	40.5	40.5	
b1	7	1	34	20.6	29.6	45.4	15.7
b2	6	5	100	2.0	79.0	80.0	5.0
c1	6	3	59	28.8	62.4	81.5	19.1
c2	7	4	127	33.9	35.5	40.1	6.0
c3	6	4	74	2.7	55.0	60.3	5.3
c4	6	2	38	2.6	46.7	48.4	1.7
c5	8	5	242	15.7	62.3	68.2	6.0
d1	6	1	18	50.0	33.3	59.3	25.9
e1	6	4	80	16.3	19.4	26.8	29.6
f1	7	6	208	16.3	63.4	72.9	11.3
g1	8	4	135	87.4	42.0	51.1	9.0
h1	9	2	122	63.9	26.0	42.6	16.6
h2	7	3	98	51.0	31.9	33.9	3.1
i1	6	4	63	25.4	41.9	48.4	13.8

Table B.3: This table presents the results of test 6, in which 3 correspondences were given and a forgiveness parameter of 25 pixels was set.

Dataset	No. of Images	No. of Points	No. of Combinations	% Dead End	% Matches Without TERs	% Matches With TER	Average Diff. %
a1	5	4	40	0	90	90	
a2	5	5	50	2.0	68.1	76.3	13.7
a3	6	9	124	16.1	67.9	74.7	19.0
a4	10	2	125	70.4	40.7	40.7	
b1	7	1	34	2.9	28.8	43.9	15.2
b2	6	5	100	2.0	79.0	80	5.0
c1	6	3	59	28.8	62.4	80.9	18.5
c2	7	4	127	33.1	35.9	38.4	3.3
c3	6	4	74	1.4	54.4	60.8	8.7
c4	6	2	38	2.6	46.7	51.2	4.5
c5	8	5	242	19.4	63.2	69.4	6.1
d1	6	1	18	44.4	30	53.3	23.3
e1	6	4	80	15.0	19.5	26.9	29.6
f1	7	6	208	16.3	63.4	72.9	11.3
g1	8	4	135	88.1	42.2	49.2	7.0
h1	9	2	122	63.1	26.3	44.1	17.8
h2	7	3	98	50	32.9	35.0	3.2
i1	6	4	63	22.2	42.1	49.0	14.5

Table B.4: This table presents the results of test 6, in which 3 correspondences were given and a forgiveness parameter of 50 pixels was set.

Dataset	No. of Images	No. of Points	No. of Combinations	% Dead End	% Matches Without TERs	% Matches With TER	Average Diff. %
a1	5	4	40	0	90	90	
a2	5	5	50	2.0	68.1	72.2	10.3
a3	6	9	124	14.5	67.9	73.7	16.1
a4	10	2	125	70.4	40.7	40.7	
b1	7	1	34	2.9	28.8	43.9	15.2
b2	6	5	100	2.0	79.0	80	5.0
c1	6	3	59	22.0	61.1	86.6	25.4
c2	7	4	127	32.3	35.7	35.6	-0.1
c3	6	4	74	2.7	55.1	61.1	12.3
c4	6	2	38	0	45.6	50	4.4
c5	8	5	242	23.6	64.9	71.4	10.9
d1	6	1	18	44.4	30	53.3	23.3
e1	6	4	80	16.3	19.4	26.8	29.6
f1	7	6	208	14.9	63.8	71.8	9.6
g1	8	4	135	91.9	44.4	50.4	11.7
h1	9	2	122	68.0	26.1	46.3	20.2
h2	7	3	98	48.0	33.0	34.9	3.0
i1	6	4	63	25.4	42.8	49.0	13.0

Table B.5: This table presents the results of test 6, in which 3 correspondences were given and a forgiveness parameter of 100 pixels was set.

Dataset	No. of Images	No. of Points	No. of Combinations	% Dead End	% Matches Without TERs	% Matches With TER	Average Diff. %
a1	5	4	20	10.0	29.1	27.5	-6.3
a2	5	5	25	8.0	19.0	18.0	-5.0
a3	6	9	42	40.5	21.7	23.2	5.9
a4	10	2	13	69.2	26.5	26.5	
b1	7	1	7	85.7	0	16.7	16.7
b2	6	5	30	20.0	33.6	34.3	3.3
c1	6	3	18	38.9	26.7	33.3	6.7
c2	7	4	23	73.9	0	5.8	16.7
c3	6	4	24	12.5	18.2	18.2	
c4	6	2	12	16.7	0	0	
c5	8	5	34	52.9	32.4	35.7	6.6
d1	6	1	6	66.7	10.0	30.0	20.0
e1	6	4	23	47.8	1.7	2.9	2.4
f1	7	6	42	47.6	31.9	33.9	2.9
g1	8	4	27	88.9	0	0	
h1	9	2	14	85.7	25.0	62.5	37.5
h2	7	3	20	50.0	8.3	15.8	25.0
i1	6	4	24	50.0	15.6	15.6	

Table B.6: This table presents the results of test 7, in which nearest neighbors to a single given point are used to select the three initial correspondences.

Bibliography

[1] H. Averbuch-Elor and D. Cohen-Or. Ringit: Ring-ordering casual photos of a temporal event. *ACM Transactions on Graphics*, 35, 2015.

[2] S. Avidan and A. Shashua. Trajectory triangulation: 3d reconstruction of moving points from a monocular image sequence. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(4):348–357, 2000.

[3] S. Baker and I. Matthews. Lucas-kanade 20 years on: A unifying framework. *International Journal of Computer Vision*, 56(3):221–255, 2004.

[4] L. Ballan, G. J. Brostow, J. Puwein, and M. Pollefeys. Unstructured video-based rendering: Interactive exploration of casually captured videos. volume 29, 2010.

[5] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (surf). *Computer vision and image understanding*, 110(3):346–359, 2008.

[6] K. Briechle and U. D. Hanebeck. Template matching using fast normalized cross correlation. In *Aerospace/Defense Sensing, Simulation, and Controls*. International Society for Optics and Photonics, 2001.

[7] M. Z. Brown, D. Burschka, and G. D. Hager. Advances in computational stereo. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(8):993–1008, 2003.

[8] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. Brief: Binary robust independent elementary features. In *European Conference on Computer Vision*. 2010.

[9] L. Cehovin, M. Kristan, and A. Leonardis. An adaptive coupled-layer visual model for robust visual tracking. In *International Conference on Computer Vision*, 2011.

[10] T. Dekel, Y. Moses, and S. Avidan. Space-time tradeoffs in photo sequencing. In *International Conference on Computer Vision*, 2013.

[11] T. Dekel (Basha), Y. Moses, and S. Avidan. Photo sequencing. *International Journal of Computer Vision*, pages 1–15, 2014.

[12] L. Goshen and I. Shimshoni. Balanced exploration and exploitation model search for efficient epipolar geometry estimation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 30(7):1230–1242, 2008.

[13] J. H. Han and J. S. Park. Contour matching using epipolar geometry. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(4):358–370, 2000.

[14] M. J. Hannah. Computer matching of areas in stereo images. Technical report, DTIC Document, 1974.

[15] S. Hare, A. Saffari, and P. H. Torr. Struck: Structured output tracking with kernels. In *International Conference on Computer Vision*, 2011.

[16] J. Y. Kaminski and M. Teicher. A general framework for trajectory triangulation. *Journal of Mathematical Imaging and Vision*, 21(1-2):27–41, 2004.

[17] G. Kanojia, S. R. Malireddi, S. C. Gullapally, and S. Raman. Who shot the picture and when? In *Advances in Visual Computing*, pages 438–447. Springer, 2014.

[18] S. Leutenegger, M. Chli, and R. Y. Siegwart. Brisk: Binary robust invariant scalable keypoints. In *International Conference on Computer Vision*, 2011.

[19] Y. Lipman, S. Yagev, R. Poranne, D. W. Jacobs, and R. Basri. Feature matching with bounded distortion. *TOG*, 33, 2014.

[20] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[21] X. Mei and H. Ling. Robust visual tracking using ℓ_1 minimization. In *International Conference on Computer Vision*, 2009.

[22] X. Mei, H. Ling, Y. Wu, E. Blasch, and L. Bai. Minimum error bounded efficient ℓ_1 tracker with occlusion detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[23] O. Miksik and K. Mikolajczyk. Evaluation of local detectors and descriptors for fast feature matching. In *IEEE International Conference on Pattern Recognition*, 2012.

[24] M. Muja and D. Lowe. Scalable nearest neighbour algorithms for high dimensional data. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 36(11):2227–2240, 2014.

[25] M. Muja and D. G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. *International Conference on Computer Vision Theory and Applications*, 2009.

[26] H. T. Nguyen and A. W. Smeulders. Fast occluded object tracking by a robust appearance filter. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(8):1099–1104, 2004.

[27] S. Oron, A. Bar-Hillel, D. Levi, and S. Avidan. Locally orderless tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[28] H. S. Park, T. Shiratori, I. Matthews, and Y. Sheikh. 3d reconstruction of a moving point from a series of 2d projections. In *European Conference on Computer Vision*. 2010.

[29] H. S. Park, T. Shiratori, I. Matthews, and Y. Sheikh. 3d trajectory reconstruction under perspective projection. *International Journal of Computer Vision*, pages 1–21, 2015.

[30] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. Orb: an efficient alternative to sift or surf. In *International Conference on Computer Vision*, 2011.

[31] R. Shah, V. Srivastava, and P. Narayanan. Geometry-aware feature matching for structure from motion applications. In *IEEE Winter Conference on Applications of Computer Vision*.

[32] A. W. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah. Visual tracking: An experimental survey. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 36(7):1442–1468, 2014.

[33] S. Wang, H. Lu, F. Yang, and M.-H. Yang. Superpixel tracking. In *International Conference on Computer Vision*, 2011.

[34] Z. Zhang, R. Deriche, O. Faugeras, and Q.-T. Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial intelligence*, 78(1):87–119, 1995.

תקציר

אירועים דינמיים מצולמים לעתים קרובות על ידי מספר אנשים מנקודות מבט שונות בזמנים שונים. מציאת התאמה של נקודות עניין זזות בכל התמונות מאפשר לנו לחלץ מידע על האובייקטים המעניינים בסצנה. חישוב התאמה של נקודות עניין נעות בסט כזה של תמונות יותר מאתגר מאשר חישוב התאמה בוידאו, מכיוון שלא ניתן להשתמש בשיטות חיזוי בוידאו בסט תמונות סטילס כלליות. בתיזה הזאת אנו מציעים מתודה חדשה לשיפור דיוק התאמת נקודות עניין על ידי חיזוי אזורים בתמונה שבהם הנקודה יכולה להימצא. האזורים נקבעים על סמך מיקומה של נקודת העניין בקבוצה קטנה של תמונות, כך שמקום הנקודה יהיה עיקבי עם סדר הזמנים שבהם נילקחו התמונות, תנועה לינארית (בקירוב) של הנקודה במרחב, והפרמטרים של המצלמות המיוצגים על ידי הגיאומטריה האפיפולרית. אנו מכנים אזורים אלו בשם *אזורים אפיפולריים תלויי-זמן*. אנו מוכיחים שכל הנקודות באזורים הלא ולידיים לא יכולות להיות מועמדות להתאמה. אנו מדגימים את האפקטיביות של המתודה בהקטנת מרחב החיפוש להתאמה, על מידע סינטטי ואמיתי, ומראים שיפור בהתאמת נקודות העניין.

המרכז הבינתחומי, הרצליה

בית ספר אפי ארזי למדעי המחשב

אזורים אפיפולרים תלויי-זמן והתאמת נקודות עניין

מוגש כחיבור מסכם לפרויקט מחקר לתואר M.Sc.

מגיש: מור דר

מנחה: פרופ' יעל מוזס

מאי 2015