Temporal Epipolar Regions

Mor Dar and Yael Moses
Efi Arazi School of Computer Science
The Interdisciplinary Center, Herzliya 46150, Israel
mor.dar@post.idc.ac.il and yael@idc.ac.il

Abstract

Dynamic events are often photographed by a number of people from different viewpoints at different times, resulting in an unconstrained set of images. Finding the corresponding moving features in each of the 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 due to possible significant differences in viewpoints and inconsistent timing between image captures. The prediction methods used in video for improving robustness and efficiency are not applicable to a set of still images. In this paper we propose a novel method to predict locations of an approximately linear moving feature point, given a small subset of correspondences and the temporal order of image captures. Our method extends the use of epipolar geometry to divide images into valid and invalid regions, termed Temporal Epipolar Regions *(TERs). We formally prove that the location of a feature in a new image is restricted to valid TERs. We demonstrate the effectiveness of our method in reducing the search space for correspondence on both synthetic and challenging real world data, and show the improved matching.*

1. Introduction

While most moving object analysis is based on video data, videos of dynamic scenes are not always available. Instead of videos, still images captured by observers of a dynamic event may be considered for analyzing a moving object. Such a set of images, termed *CrowdCam* ([1, 10]), is taken from multiple viewpoints at different times. As in videos, feature matching may serve as a basic component for analysis of moving objects. A common strategy to improve efficiency and robustness of feature matching in a video sequence is to limit the search space of a feature location using prediction. Such methods are not applicable to CrowdCam data, as video based approaches assume short, consistent time intervals between frames as well as no significant viewpoint change. In this paper, we propose a method to predict the possible location of a candidate match in CrowdCam images.

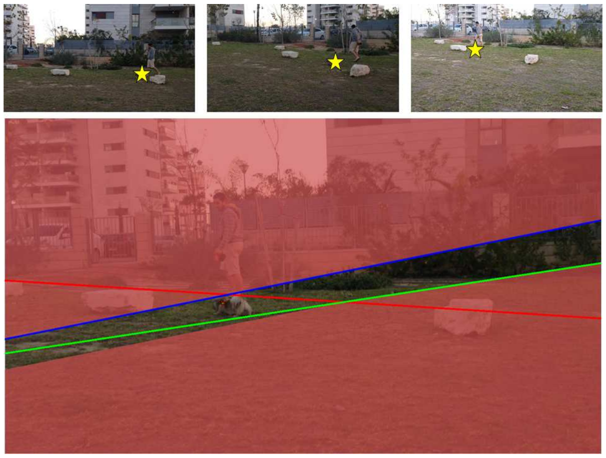


Figure 1: An example (dataset h1) of limiting the search space for correspondence using TERs. Epipolar lines calculated from three correspondences (yellow stars) are used in conjunction with a known temporal order to define valid regions in a fourth image, greatly limiting the search space for correspondence. (Best viewed on a computer screen.)

To predict the location of moving features, assumptions must be made relating to the movement of the object, the positions of the cameras, or the timing of the frames. Our work assumes the following: (i) The features to be matched are moving in an approximately linear 3D trajectory. This is a common assumption in many video-based tracking works (e.g., [30]). (ii) The epipolar geometry between pairs of cameras can be computed using static features from a common background. Our method does not require epipolar geometry for every pair, but at least three fundamental matrices for each image in which we search for correspondence. (iii) The temporal order of the image captures is given or can be computed directly from the data. When reliable camera clocks are unavailable, the temporal order of CrowdCam images can be computed using photo sequencing[9, 10]. These studies use a similar setup and the two aforementioned assumptions. (iv) The feature location in at least three images is given. If not, we can use standard matching techniques as an initialization in order to find them. Note that prediction methods in videos also assume that some initial correspondence can be computed.

In order to predict the possible locations of a moving feature in a new image, I, we utilize assumptions (i-iv) to define valid and invalid regions for correspondence in I. Assumptions (ii) and (iv) are used to compute epipolar lines in I from the known correspondences. In contrast to static points, a moving point is not restricted to lie on the epipolar line. However, using the approximate linear motion and known temporal order, we show how to use epipolar lines to restrict the possible locations of the point. To that end, we define regions in I which are bounded by the epipolar lines and their parallels. We prove that the given temporal order of the set of images determines whether each region is valid or invalid for the feature location. As such, we name these regions *Temporal Epipolar Regions* (TERs). The validity of each TER depends only on the temporal order, and therefore for any set of images and features, the same lookup table may be used to assign validity to TERs (e.g., Table 1). We discuss relaxing our assumptions in Sec. 4.

We demonstrate, through experimental results on a variety of datasets, that TERs considerably restrict the possible location of features within the images. We further show that additional matched features increase the set of known corresponding points and, in turn, decrease the size of the valid TERs in the remainder of the images. As TERs reduce the search space for a corresponding feature, they may increase the efficiency and accuracy of any feature matching algorithm. The overhead required for our method consists of preprocessing to calculate the fundamental matrices and computing the temporal order, if not available. The calculation of the valid regions is computationally inexpensive as it involves computing epipolar lines and utilizing a lookup table.

The main contributions of our work are (i) the introduction of the novel problem of predicting the location of moving image points in a CrowdCam setting and (ii) the proposed extension of epipolar geometry to constrain the location of moving feature points in a set of images, as opposed to the classic use of epipolar lines for restricting the location of static features.

2. Related Work

Prediction of feature (or object) location in successive frames is a basic building block in video based tracking methods (see review by [30]). A naïve prediction is a window around the location of the feature (or object) in a previous frame (e.g., [6, 4, 13, 33]). Another approach is to utilize a motion model learned from previous frames. This approach often utilizes Kalman or Particle filters (e.g., [24, 8, 20, 21, 25]). As these strategies for prediction assume short, consistent, time intervals between frames as well as no significant viewpoint change between frames, they do not apply to CrowdCam image sets.

The epipolar constraints between a pair of images are used to restrict the possible set of correspondences for static scenes. For example, they can be used to find dense correspondence for pairs of images (e.g., [28]) or rejecting incorrect matches when searching for sparse correspondence (e.g., [12, 35, 19, 32, 14]. A recent work by Shah *et al.* [29] uses epipolar geometry between wide-baseline images to predict correspondence to improve matching in the presence of repetitive structures. When a set of images is available, Structure From Motion can also be used to improve correspondences ([34, 31]). However, these studies, which use geometric constraints, only apply to static features.

Finding correspondence between still images has been very well studied and varies in its approaches according to the goal of the work. Direct comparison of descriptors [23, 22, 27, 17, 7] may be the most common approach for matching. Our proposed method is complementary to direct feature matching, as TERs are used to limit the search space. Therefore, matching strategies may be used in conjunction with our method for improved matching accuracy.

The CrowdCam is becoming increasingly popular as various devices such as smartphones are used for capturing dynamic events. Therefore, novel problems are being addressed in a number of recent studies. Some focus on visualization of CrowdCam video sequences (e.g., [5, 1]). Others order the images in time [9, 10, 16] or space (e.g., [2]). Our work is another step forward in extracting information available in CrowdCam images.

3. Spatial-Temporal Consistent Regions

In this section, we define TERs, the validity of regions, and describe how to determine which of the TERs are valid. 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 $q_u \in I_u$ can or cannot be located. To do so, we use the fundamental matrices $F_{u,j}$ between images I_u and I_j, where $q_j \in I_j$ and $q_j \in S_q$.

In order to define the said regions, we use 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 $t(I_{\sigma(1)}) < t(I_{\sigma(2)}) \dots < t(I_{\sigma(N)})$.

3.1. Consistency Definitions

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

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.

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 $t(I_j)$.

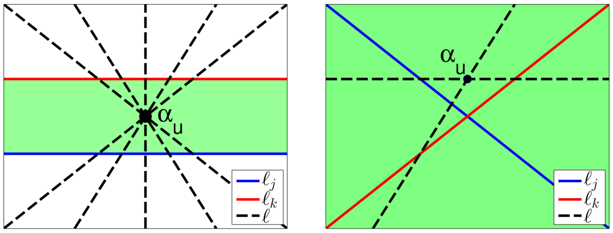
3. The relative spatial locations of each point in S_Q along L correspond to the temporal order, σ.

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 termed a *valid spatial-temporally consistent point* with respect to S_q and σ or, for short, a *valid point*.

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., [3, 15]). 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. Our method requires a weaker calibration (only fundamental matrices between partial set of images) than those required for L recovery, and only three corresponding points as initialization. Furthermore, our method is less sensitive to deviation from linear motion.

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, passing through α_u, 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.



(a) (b)

Figure 2: (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 any σ.

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.

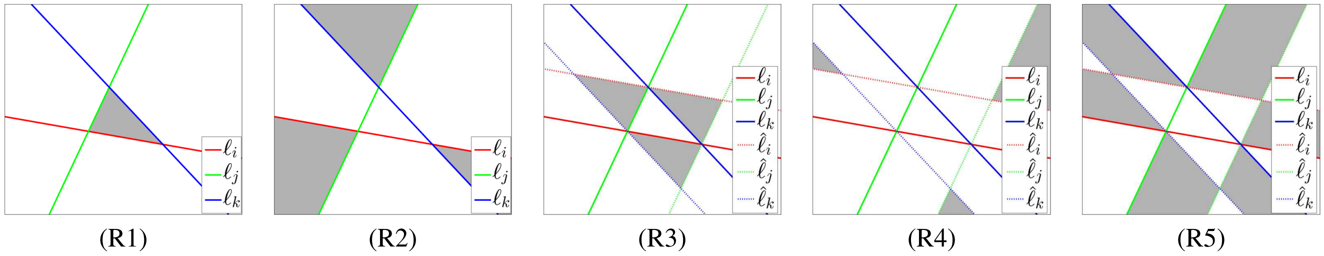
3.2. Two Epipolar Lines

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 area 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 (Fig. 2a). Note that all points within this area are similarly valid, while all points outside the area are invalid. As such this image is split into one valid and two invalid temporal epipolar regions.

In the more general case, in which the epipolar lines are not parallel, all candidate points in the image are valid (Fig. 2b). 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 TERs of five types (Fig. 3). TERs are defined by ℓ_i, ℓ_j, and ℓ_k, and their parallels. 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 ($\hat{\ell}_i$ and $\hat{\ell}_k$ are defined similarly). The following claims are proved in the supplementary material:



(R1) (R2) (R3) (R4) (R5)

Figure 3: Three epipolar lines (ℓ_i, ℓ_j, ℓ_k) and their parallels ($\hat{\ell}_i$, $\hat{\ell}_j$, $\hat{\ell}_k$) split an image plane into sixteen distinct regions (in gray) of five types.

(i) In each region, all points are valid or all points are invalid. This classification is dependent not on the epipolar lines but on σ. (ii) There are $4!/2$ possible orders up to direction that must be considered. For each region, 6 of these orders are valid while the remaining 6 are invalid. (iii) For a given σ, 8 of the 16 regions are valid TERs, while the other 8 are invalid.

Table 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 1.

We next outline the proof for the above classifications. Inspired by [9], we define for a point, α_u, six sections. Each section defines a unique order in which an ℓ passes through α_u and the set of epipolar lines. The sections are defined using two types of critical lines. The first is the line, c_{ij}, connecting α_u with the intersections of a pair of epipolar lines. Formally, let $\gamma_{ij} = \tilde{\ell}_j \times \tilde{\ell}_i$ and $c_{ij} = \tilde{\alpha}_u \times \tilde{\gamma}_{ij}$. The second type of critical line is parallel to the epipolar lines and passes through α_u (i.e., c_i is parallel to ℓ_i and passes through α_u). Fig. 4 gives an example of sections for a point in $R2(i, j)$. Note that $\alpha_i = \tilde{\ell} \times \tilde{\ell}_i$ (in homogenoues coordinates) and that α_j and α_k are similarly defined.

The following observations can be easily verified geometrically.

A1 The order of α_i and α_j is swapped in neighboring sections which share the border c_{ij}.

A2 The location of α_i in the order moves from first to last (or vice versa) in neighboring sections which share the border c_i.

A3 Within each section, the order of intersections of all lines is preserved.

Using these observations about sections and the order of critical lines, we use one valid order to calculate the rest. Let us consider as an example the region R2. Without loss of generality, two non-parallel lines divide a 2D space into four areas. In the general case, a third line will pass through three of these areas. The remaining area is defined to be R2.

Definition 3: $R2(i, j)$ is defined as the area comprised of ℓ_i and ℓ_j through which ℓ_k does not pass.

Claim 1: $R2(i, j)$ is a valid TER for all orders in which u is not adjacent to k in σ.

Proof Outline of Claim 1: A formal proof can be found in the supplementary material. Consider a line ℓ in the section bordered by the pair (c_{ij}, c_{ik}) (see Fig. 4). By definition it passes through α_u and $R1$ (the triangular region defined by three non-parallel epipolar lines). It is easy to visually verify that the intersection order is given by $\sigma = (u, j, i, k)$. The intuition is as follows. As c_{ij} and c_{ik} both intersect ℓ_i on the border of $R1$, ℓ must also intersect ℓ_i in this interval. As this intersection is outside of $R2(i, j)$, ℓ must intersect ℓ_j between α_u and ℓ_i. As $R1$ is a closed convex shape (triangle), any line passing through it must cross two of its borders. As such, ℓ_k and ℓ_i are adjacent in the intersection order. Finally, as ℓ_k does not pass through $R2(i, j)$, it cannot be between ℓ_j and ℓ_i in the order. Therefore, we conclude that the order of intersections of ℓ must be $\sigma = (u, j, i, k)$.

To find the remainder of the valid orders of $R2(i, j)$, we utilize observations **A1**-**A2** and the order of the critical lines, given by $(c_{jk}, c_{ij}, c_{ik}, c_i, c_k, c_j)$ (a formal proof is given in the supplementary material). Critical line c_{ik} is on the border of neighboring sections (c_{ij}, c_{ik}) and (c_{ik}, c_i). To find the order in section (c_{ik}, c_i), **A1** dictates that a swap be made between i and k in the order, such that $\sigma = (u, j, k, i)$. Using **A2**, the order in section (c_i, c_k), which shares the border c_i with (c_{ik}, c_i), is given by moving i to the opposite end of the order. As such, for this section we have that $\sigma = (i, u, j, k)$. The remaining valid orders, $\sigma = (k, i, u, j)$, $\sigma = (j, k, i, u)$, and $\sigma = (u, i, j, k)$, can be computed similarly. These six orders represent the 6 distinct valid orders of $R2(i, j)$ out of the 12 possible orders. Hence, the remaining 6 are invalid.

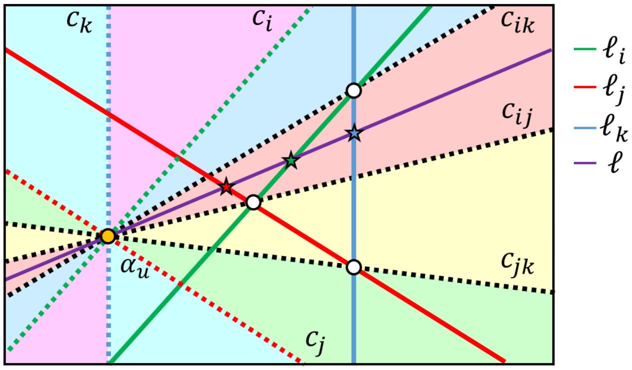


Figure 4: The different sections defined by critical lines, given $\alpha_u \in R2(i, j)$ and the order along the line ℓ in the section bordered by (c_{ij}, c_{ik}).

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

Table 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.

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. 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. This region does not guarantee that there exists a line which conserves σ. Instead, it guarantees the correctness of the invalid regions. Even though this method is not optimal, a larger $|S_q|$ can only further limit valid TERs, and therefore the search space. 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.

4. TERs and matching

Now that we have defined the validity of TERs, we focus on how to utilize them to improve feature matching algorithms, and to propose correspondence verification. As a first step of our method, features are extracted from all the images (we use SIFT features [18]) and the fundamental matrices F_{ij} between required pairs of images $I_i, I_j \in \mathcal{I}$ are computed (we use the BEEM algorithm [11]). If the temporal order of the images is unknown, it is possible to utilize photosequencing [10, 9] as an additional preprocessing step to calculate it. A set S_q of at least three correspondences is either given or computed by any standard matching algorithm.

Matching using prediction: 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 1. For larger $|S_q|$, the method described in Sec. 3.4 is used. Only the features in the valid regions are considered as candidates for correspondence. From these features, the nearest neighbor to the features in S_q is chosen as the corresponding point, α_u (note that other matching criteria may be used). Then, S_q is updated to $S_q = S_q \cup \{\alpha_u\}$. The additional constraints defined by α_u are used to update and reduce the size of the valid regions in the remainder of the images. This process is repeated until all the images are assigned correspondence. We present an example of matching using prediction in Fig. 5. An alternative approach which may be more efficient, is to compute features only in valid regions.

Relaxed TERs: The valid regions may be unreliable when the motion deviates from linearity or when the fundamental matrices are inaccurate. To compensate for this, a simple *forgiveness parameter* may be used. The borders of the valid TERs are extended by this parameter. Note that larger forgiveness parameters make our method more robust in handling deviations from our assumptions, but also increase the valid set of candidate correspondences.

Unreliable TERs: As there may be noise in the computation of epipolar lines, TERs may be unreliable when epipolar lines or their intersections are too close together. In these cases we may declare the TERs *unreliable*, and thus match without them (by any standard matching algorithm). Alternatively, in cases in which we have $n > 3$ epipolar lines, we propose the following workaround. Instead of building TERs using epipolar lines which are close together, we find TERs using every subset of epipolar lines which are sufficiently far apart. Then we find the union of the valid TERs to determine an overall valid area. This allows us to still utilize the constraints given by each known correspondence while avoiding errors from close epipolar lines.

Correspondence verification: In some cases no valid regions exist in the image. This can be regarded as a *dead end*, as S_q cannot be extended. If we encounter such a dead end, we deduce that either the linear motion assumption does not

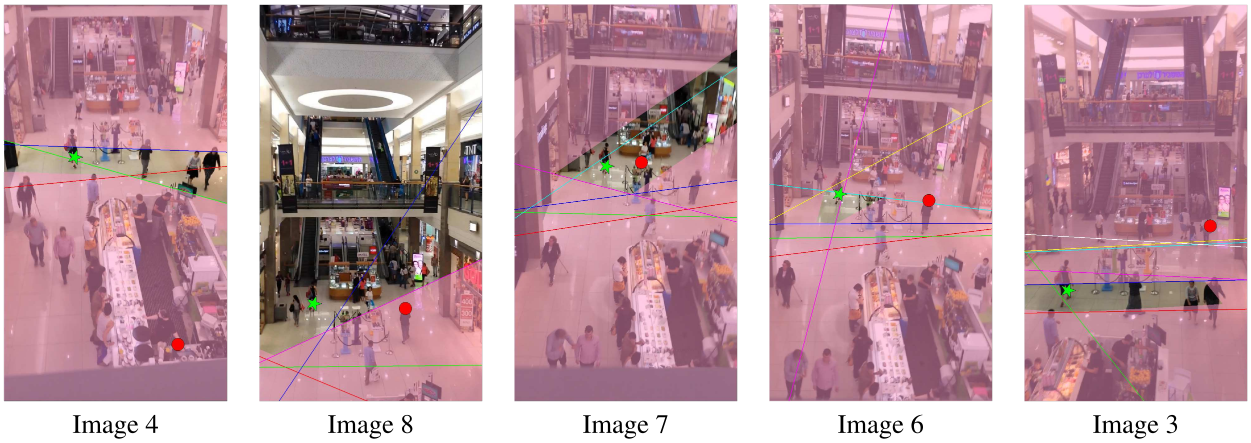


Image 4 Image 8 Image 7 Image 6 Image 3

Figure 5: 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 matching with TERs (green stars) and without TERs (red circles).Note: images are labeled based on their order in time but presented in the order, from left to right, in which best nearest neighboring matches were selected by the TER algorithm. 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).

hold or that there are errors in the computed fundamental matrices, temporal order, or set S_q. In this case, we can do no better than matching without TERs. This is demonstrated in tests 3 and 6 in Sec. 5. If we 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.

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 a direct nearest neighbor matching method with and without the use of TERs.

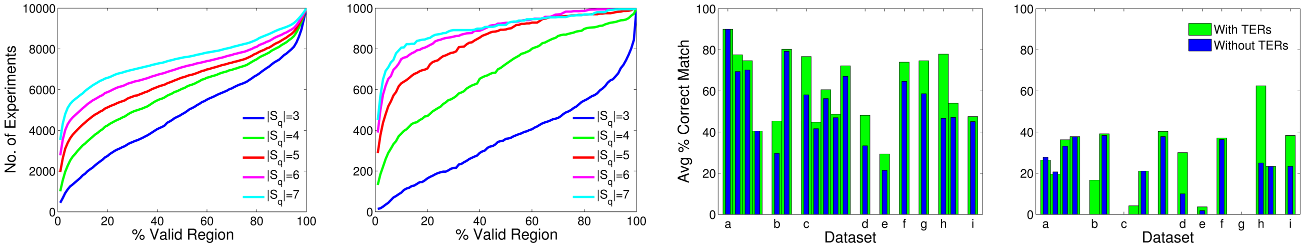
5.1. Simulated Data

The simulated scenario consisted of a set of 4-8 cameras capturing a moving point. The cameras were positioned randomly along a semisphere, all pointed approximately at the origin, through which a randomly generated line passes. Images were created 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) for different values of initial correspondences. This test is required as we have no closed analytic computation of the size of the valid region; it depends on the camera configurations and the location of the moving point. 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 Fig. 6(a) as cumulative histograms of PV, one for each number of initial correspondences, $|S_q|$. Ideally, a large number of experiments (the y axis) should have as small as possible percent of the image valid (the x axis). Our data shows that for $|S_q| = 3$, approximately 27% of the 10000 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 valid region 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 size of the valid regions decreases, as desired. 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, and therefore the trend 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.



(a) (b) (c) (d)

Figure 6: (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. (c) and (d) show the average percent correct matching per dataset with and without TERs. In (c), S_q is given by ground-truth points, while in (d) S_q was initialized using nearest neighbor matching. We see that, with the exception of two datasets in which initialization failed, TERs improve the percent correct matching. Note that in (c) and (d) 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.

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, we stop if an image has reached a dead end.

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

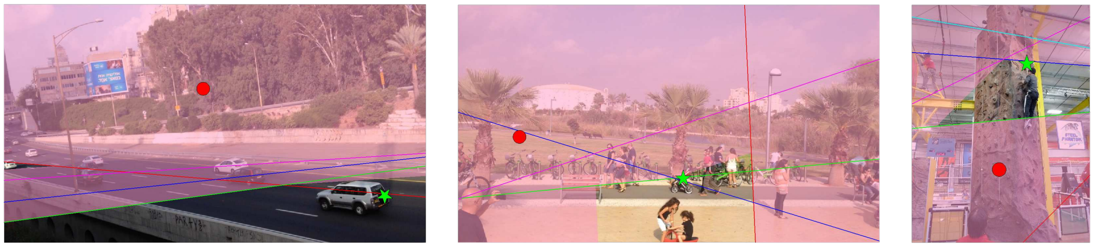
For each noise level used, we find the number of simulations in which: (i) a point in a valid region exists in each image; (ii) no point was in a valid region in at least one of the images; and (iii) a dead end occurred. For STD of 0 and 1, all simulations fell into category (i). When the STD rose to 3, 0.54% of simulations were of category (ii) and the rest were category (i). Finally, when the STD was 5, 3.34% of the simulations were category (ii), 2 simulations were category (iii) and the remainder were category (i).

Test 3: We tested whether 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). However, in each of the remaining images a random point was selected. For this test we used a forgiveness parameter (as described in Sec. 4) of 2 pixels. We proceeded through the images as in Test 2. The results are as follows: using four cameras, no points were found in valid regions in 46.2% of the simulations, which reflects the probability of having a random chosen point within an invalid region (no dead ends can occur with only 4 cameras). With 5 cameras 74.81% of the simulations had no matches, and of the remaining simulations 26.38% were dead ends. With 6 cameras the results were 70.98% with no matches, and 66.37% were dead ends. As such, 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 up to 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, 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. The 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 dataset supplied by [26]. From this dataset, we selected a partial set in which we consider only the unidirectional movement of the climber. Examples taken from datasets we considered are presented in Fig. 7 and additional examples may be found in the supplementary materials. The selection of examples was in part based on the size of the valid regions, as in many cases, valid TERs are are too small to 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. The ground-truth correspondence was used to initialize a set of size $|S_q| = 3$. 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 $|S_q| \in \{4, 5, 6, 7\}$. We selected at random 1000 sam-



Dataset a2

Dataset g1

Dataset e1

Figure 7: The application of our method to a variety of scenes. Correspondences found using TERs (green stars) are correct and those without TERs (red circles) are incorrect. Additional examples may be found in the supplementary materials.

ples of the set of experiments for each size of $|S_q|$. This allows to compare the PV for different values of $|S_q|$. The results (Fig. 6(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 cosine similarity between feature vectors.

When dead ends occurred, or in the case of unreliable TERs, our method could not be applied; it can do no better than standard matching. We only present cases in which no dead-ends or unreliable TERs were detected, as those cases offer a fair comparison. Statistics relating to dead ends and unreliable TERs in the supplementary materials.

As TERs are dependent on epipolar geometry, their effectiveness in restricting the search space dipends on the geometry between the images captured. Thus, we present the results of this test per dataset. Fig. 6(c) shows the average percent of correct matchings using standard matching with and without our prediction. There are between 9 and 221 experiments in each dataset. The number of experiments is defined by the number of points to be matched and the number of combinations of initial correspondences. 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.

Test 6: We tested the effectiveness of our method similarly to test 5 when S_q is not given. Given a point, S_q was initialized by finding its two nearest-neighbors in the image set. As expected, the results show errors in the initial set. In 294 of the 496 experiments, at least one point in S_q was incorrectly matched. Experiments using these incorrect initializations yielded dead ends more often in datasets containing more images. Datasets with 5 images, yielded 21.1% dead ends, datasets with 7 images yielded 80.0% dead ends, and datasets having 10 or more images always resulted in dead ends. Additional details are provided in the supplementary materials. This trend demonstrates that with high probability errors in correspondence, in particular with the initial correspondence, can be detected using only 10 images of the moving feature.

In Fig. 6(d) we present the percent correct matching when experiments did not reach a dead end. For each dataset this figure, between 1 and 24 experiments were performed. We see that with or without TERs, the percent of correct matchings drops significantly due to the lack of correct initial matches. However, this test highlights that given any initialization, TERs may still be used to improve matching. The average improvement over all the datasets is 2.7%.

6. Conclusions

In this paper, we introduced a method for utilizing epipolar geometry to predict correspondence of moving points through CrowdCam images. We demonstrated on both simulated and real world data that this prediction reduces the possible locations of a moving feature within an image. As a result, it may improve the efficiency and accuracy of matching algorithms. Additionally, we showed how our method may be used to verify the accuracy of our assumptions and correspondence using dead ends. Developing algorithms that utilize dead end detection in order to improve correspondence, is left for future research. Another possible extension is to directly determine the optimal TERs for $|S_q| > 3$, rather than using intersections of valid regions. We expect that any state-of-the-art algorithm for matching dynamic features can be improved using our method.

Acknowledgements

This work was partially supported by the Israel Science Foundation grant no. 930/12 and by the Israeli Ministry of Science, Grant 3-8744.

References

[1] A. Arpa, L. Ballan, R. Sukthankar, G. Taubin, M. Pollefeys, and R. Raskar. Crowdcam: Instantaneous navigation of crowd images using angled graph. In *IEEE International Conference on 3D Vision*, 2013.

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

[3] 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.

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

[5] L. Ballan, G. J. Brostow, J. Puwein, and M. Pollefeys. Unstructured video-based rendering: Interactive exploration of casually captured videos. *ACM Transactions on Graphics*, 29(4):87, 2010.

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

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

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

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

[11] 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.

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

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

[14] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, 2003.

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

[16] 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.

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

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

[19] J. Maciel and J. P. Costeira. A global solution to sparse correspondence problems. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(2):187–199, 2003.

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

[21] 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.

[22] 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.

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

[24] 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.

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

[26] 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.

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

[28] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1-3):7–42, 2002.

[29] 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*, pages 278–285, 2015.

[30] 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.

[31] N. Snavely, S. M. Seitz, and R. Szeliski. Modeling the world from internet photo collections. *International Journal of Computer Vision*, 80(2):189–210, 2008.

[32] R. Szeliski. *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.

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

[34] C. Wu. Towards linear-time incremental structure from motion. In *IEEE International Conference on 3D Vision*, 2013.

[35] 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.