Space-Time Tradeoffs in Photo Sequencing

Tali Dekel (Basha)
Tel Aviv University
talib@eng.tau.ac.il

Yael Moses
The Interdisciplinary Center
yael@idc.ac.il

Shai Avidan
Tel Aviv University
avidan@eng.tau.ac.il

Abstract

Photo-sequencing is the problem of recovering the temporal order of a set of still images of a dynamic event, taken asynchronously by a set of uncalibrated cameras. Solving this problem is a first, crucial step for analyzing (or visualizing) the dynamic content of the scene captured by a large number of freely moving spectators. We propose a geometric based solution, followed by rank aggregation to the photo-sequencing problem. Our algorithm trades spatial certainty for temporal certainty. Whereas the previous solution proposed by [4] relies on two images taken from the same static camera to eliminate uncertainty in space, we drop the static-camera assumption and replace it with temporal information available from images taken from the same (moving) camera. Our method thus overcomes the limitation of the static-camera assumption, and scales much better with the duration of the event and the spread of cameras in space. We present successful results on challenging real data sets and large scale synthetic data (250 images).

1. Introduction

Group photography is emerging as one of the most popular means of photography today. One can often see a group of people, armed with smartphones, huddling together to grab pictures of some exciting dynamic event. The set of still images taken this way can be regarded as the output of a new type of extended camera, which we call a *crowd-based camera*, or *CrowdCam* for short. Traditionally, such image sets are used for analyzing or visualizing the static regions of the scene (e.g. [17]). However, such data also contains rich information about the dynamic content of the scene.

We are interested in developing tools that analyze, explore and visualize the *dynamic* regions of the scene given images taken by CrowdCam (e.g., see Fig.1). A preliminary step in solving this problem is to recover the temporal order of the still images taken asynchronously by a set of uncalibrated cameras. It is of interest to develop vision based methods to solve this problem rather than assume that an arbitrary set of smartphones is synchronized, to frame level precision, ahead of time.

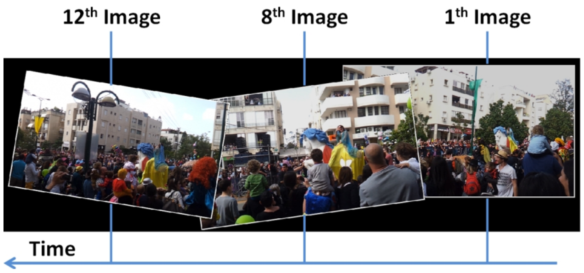

Figure 1. Visualization of a dynamic event from a set of still images; each image was captured from a different location at a different time. This visualization was generated automatically.

Basha *et al.* [4] were the first to propose a vision based solution to the photo-sequencing problem, and we follow their general framework. However, we use different assumptions on the data, which requires developing new tools. First, we compute corresponding static and dynamic feature points across images. The static features are used to determine the epipolar geometry between pairs of images. Each set of corresponding dynamic features vote for the temporal order of the images in which it appears. The partial orders provided by the dynamic feature sets are aggregated into a globally consistent temporal order of images using rank aggregation.

One of the non-trivial problems that must be solved is how a set of corresponding dynamic features can be used to determine the partial order of the images in which it was found. What makes this problem so challenging is the uncertainty both in time and space. That is, each feature set contains the projections of a 3D dynamic point onto different viewpoints at a different time instance. Basha *et al.* proposed a 2D geometric based solution that requires that two of the input images be captured by a static camera. Under linear motion of each of the dynamic features, this assumption allows them to compute a unique ordering by mapping all the features to the same reference image. However, there are a number of problems. First, such a pair must be detected automatically, which is not a trivial task. More disturbing is the fact that all feature points must appear and be matched to features in the static pair. This complicates the correspondence problem and limits the spatio-temporal extent of the event that can be captured.

We consider a somewhat different scenario, where a static-pair is not needed, but each camera takes more than a single photo. This scenario increases the uncertainty in space, because the features cannot be mapped to the same reference image, but decreases the uncertainty in time (since the temporal order of images taken by the same camera is known). We show that, using both the spatial and the temporal constraints, a small number of temporal orders can be determined from each feature set. In addition, the temporal information is also integrated as a confidence vote into the rank aggregation,which improves the robustness to errors and noise.

Our novel geometric analysis allows us to sidestep the need to establish correspondence w.r.t. a specific image, i.e., each input image can be matched to a different subset of images. As a result, our approach scales better than [4] when the distance between the cameras, or the time interval within which the images are taken, grows. This demonstrates the advantages of the tradeoff between spatial and temporal cues.

2. Related Work

Basha *et al.* [4] are the only ones to address the photo-sequencing problem. We elaborate on their work later in the paper. Related problems include Dynamic Structure-From-Motion (D-SFM), Non-Rigid Structure-From-Motion (NRSFM), and Video Synchronization.

In D-SFM and NRSFM the goal is to recover a 3D model of a dynamic world from a set of images taken at different time instances (e.g., monocular video). This is an under-constrained problem because we do not have the projection of a moving 3D point on two or more images and therefore cannot use triangulation to recover its 3D location. To overcome this limitation one needs to add priors on the motion of the 3D point.

D-SFM relies on trajectory priors, under the assumption that points move along a parametric trajectory. For example, Avidan & Shashua [2] showed how to recover the 3D coordinates of a point moving along a straight line or a conic section. This was later extended by Kaminski & Teicher [10] to general trajectories using polynomial representation. Wolf & Shashua [19] gave a catalog of different trajectory types and their corresponding solution by framing the problem as one of projection from $\mathcal{P}^N$ to $\mathcal{P}^2$.

NRSFM methods rely on shape priors to constrain the problem. For example, Bregler *et al.* [5] extended the factorization-based method of Tomasi & Kanade [18] to model nonrigid 3D objects observed by a monocular camera. That is, the nonrigid shape is taken to be a linear combination of some basis shapes. This, however, assumes an orthographic projection. Hartley & Vidal [9] proposed a closed-form solution to nonrigid shape and motion recovery from multiple perspective views, under the assumption that the nonrigid object deforms as a linear combination of K rigid shapes. These methods usually assume only a single nonrigid object of interest. Instead of assuming the nonrigid shape to be a linear combination of basis shapes, Akhter *et al.* [1] took the dual approach and described each trajectory as a linear combination of basis trajectories.

Algorithm 1 Photo Sequencing Alg.

Input: A set of still images, $\mathcal{I}$, taken by a set of moving cameras. The temporal order of images taken by the same camera is known.
Output: A permutation, σ, of $\mathcal{I}$.

```
Match features between all input image pairs.
Compute the fundamental matrices between the image pairs.
Classify the features into static and dynamic feature sets, $S^i$.
for each set of dynamic features, $S^i$ do
  for each image, $I_j \in \mathcal{I}_{S^i}$ do
    Compute the order set, $\Gamma^i_j$, using $I_j$ as reference (see Sec. 3.1).
  end for
  Compute final order set, $\Gamma^i = \Gamma^i_1 \cap \cdots \cap \Gamma^i_n$ .
end for
Compute the transitivity matrix, $M$ from $\Gamma^1, \Gamma^2, \ldots$.
Integrate the temporal constraints into $M$.
Find the full order, $\sigma$, from $M$ using rank-aggregation [6]
```

Our problem is also related to video synchronization where the goal is to temporally align two or more video sequences. However, only a small number of parameters need be estimated, typically, just shift and scale are enough. Such methods are inapplicable for solving the photo sequencing problem because there are many more degrees of freedom. The video synchronization techniques most relevant to our case are those that employ geometric constraints to align multiple video sequences (e.g., [12, 14]) or to achieve sub frame accuracy (e.g., [11]). In both cases the intersections of the trajectory of dynamic features with the epipolar lines of corresponding features in the other images were used to define order. None of these methods consider the inconsistent ordering that may be caused by different choices of features.

Other related studies are [3, 15]. Ballan *et al.* [3] proposed a method for navigating in a collection of videos of a dynamic scene, e.g., a music performance, but they use a collection of casually captured videos rather than still images as we do. Schindler *et al.* [15] proposed a method for constructing 4D city models from images that span many years. Their method is based on analyzing long-term changes in the 3D static scene. Our method deals with a dynamic scene captured in a short time interval and is based on motion.

3. Method

A group of cameras moves in space and captures a set of still images, $\mathcal{I}$, of a dynamic event. Each image is captured from a different location at a different time step. Our goal is to recover the *photo-sequencing*) of $\mathcal{I}$. We assume that the relative order of images captured by the same camera is given, and use it to impose temporal constraints on those images. However, we do not assume the actual time stamp of the images is known.

Following Basha $et\ al.$ [4], our method can be roughly divided into three main steps (pseudo-code given in Alg. 1).

Preprocessing: Feature points are detected and matched between image pairs. The fundamental matrix is computed between image pairs for which enough inlier features are found. In this case, feature points that obey the epipolar constraint are labeled as *static* points, and those that do not are labeled as *dynamic* points.

Order from a Single Feature Set: Let S^i be a set of corresponding dynamic features, which are the projections of a dynamic 3D point P^i onto a subset of images, $\mathcal{I}_{S^i} \subseteq \mathcal{I}$. The set S^i is used to compute a set of possible temporal orders (permutations), Γ^i, of its set of images, $\mathcal{I}_{S^i}$. To do so, we make use of all available spatial-temporal information, and perform a 2D-based analysis. This is the key that allows us to consider a wider spread of images in space and time. We elaborate on this step in Sec. 3.1.

Rank Aggregate: Rank Aggregation is used to compute a unique global order that is as consistent as possible with the computed partial temporal orders from all sets, $\bigcup \Gamma^i$, as well as with all available ordering constraints. See Sec. 3.2.

3.1. Temporal Order from a Single Feature Set

The question we are facing is how to compute the order set Γ of the images $\mathcal{I}_S$ given the feature set S (the superscript i is dropped for simplicity). Possible solutions to this problem include recovering the trajectory of P in 3D or in 2D. However, these solutions make several limiting assumptions on the data, which we would like to avoid. We next briefly review these approaches and then describe in detail our solution.

3.1.1 Previous Solutions – Trajectory Recovery

Assume we can recover the 3D trajectory of P and sample it at the time the images were captured. Then, the temporal order of the image set, $\mathcal{I}_S$, is determined (up to time-flip) by the spatial order of the 3D locations of P along its 3D trajectory. Under the assumption of linear motion, the 3D trajectory of P can be recovered using [2]. However, their solution requires 5 or more fully calibrated images ($|S| \geq 5$), which is hard to obtain in reality and prone to errors due to deviation from the linear motion assumption and sensitivity to noise, as demonstrated in our experiments. Therefore, we avoid 3D reconstruction and propose a 2D-based solution.

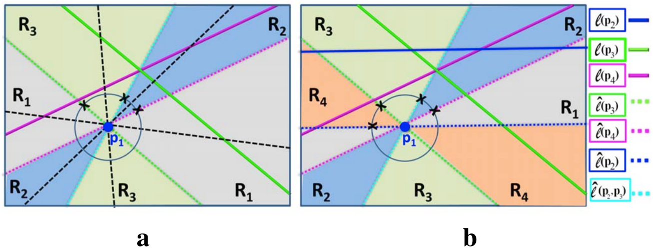

Figure 2. **Critical points**: **(a)**, a point p_1 and two epipolar lines (green and purple); each black dashed line, ℓ, is in a different sector and induces different orders; the 3 sectors are define by 3 critical points that are marked on the unit circle centered in p_1. **(b)**, a point and 3 epipolar lines; here not all critical points are marked; see proof in Sec. 3.1.2.

Basha $et\ al.$ [4] suggested recovering the 2D linear trajectory of the point P in one reference image. Then, the 2D projections of P at time $\{t(I_i)|\ I_i \in \mathcal{I}_S\}$ could be easily computed, and their 2D spatial order along the 2D trajectory induce a unique temporal order of $\mathcal{I}_S$. The main drawback of their solution is that it relies on a *static camera*, which requires establishing reliable correspondence between features in each of the input images and the reference one. This assumption limits the spatial and temporal configurations of cameras that can be considered by their method. Moreover, they do not exploit all the available spatio-temporal information. That is, only feature sets that involve the reference image are used to induce the temporal order, and valuable temporal information, such as known temporal orders of images taken by the same camera, is ignored. We suggest a 2D geometric-based solution that extends the solution of Basha $et\ al.$ [4], but overcomes its main limitations.

3.1.2 Order Without Trajectory Recovery

We drop the static camera assumption, which means that a unique order of $\mathcal{I}_S$, based on the recovered 2D trajectory cannot be obtained as in [4]. Instead, we treat each image $I_j \in \mathcal{I}_S$ as a reference, and use it to compute a set of ordering $\Gamma_j = \{\sigma_1, \sigma_2 \cdots\}$. As a result, we obtain $n = |S|$ sets of temporal orders, $\Gamma_1 \cdots \Gamma_n$. The intersection $\Gamma = \Gamma_1 \cap \cdots \cap \Gamma_n$ is the final set of ordering consistent with the set S.

The main challenge is how to efficiently compute the set Γ_j and we show that geometric and temporal constraints can drastically reduce the size of Γ_j, compared to pure combinatorial considerations. For example, from a combinatorial point of view there are $\sim 10^{43}$ possible ways to order a set obtained by 10 cameras that take 5 images each. This is reduced by geometric constraints to only 10^3. In practice, it can be further reduced to $\sim$4 using temporal constraints.

A Single Reference Analysis: W.l.o.g. let $S = \{p_j\}_{j=1}^n$ such that $p_j \in I_j$, and let I_1 serve as a reference image. Given that the fundamental matrix between image I_1 and I_j can be computed (i.e., enough inlier features were found), the corresponding epipolar line $\ell(p_j) \in I_1$ can be determined. Therefore, the available information in I_1 is a feature point, p_1, and a set of epipolar lines, $\{\ell(p_j)\}_{j=1}^{n}$.

Assuming that P moves along a straight line L, the projection of L onto I_1 is a line ℓ that intersects the set $\{\ell(p_j)\}_{j=1}^{n}$. The spatial order of the intersection points along ℓ induce the temporal order of the images, up to order reversing. Unfortunately, the line ℓ is unknown since we dropped the static-pair assumption of [4]. In our case, any line passing through p_1 is a valid solution to ℓ. Thus, a line ℓ is defined by its orientation. Since different lines may induce different temporal orders (see example in Fig. 2(a)), a unique order cannot be recovered in the general case. However, thanks to geometric and temporal constraints, we can recover a small bounded number of valid orders.

Geometric Analysis: The key observation is that there are ranges of orientations for which the temporal order induced by ℓ is fixed (up to order reversing). This is shown in Fig. 2(a) for a particular configuration of one point and two epipolar lines; the order defined by all lines in sector R1 will be the same, and p_1 will be between the pink and the green lines, while in sector R2 the green line will be in the center. With this observation in mind, we divide the image plane into sectors, such that all lines within a sector give rise to the same (up to reversing) temporal order.

We define the image sectors by *critical points*, which are points on the unit circle centered at p_1. Critical points are formed by the intersection of two types of lines that incidence at p_1 with the unit circle. The first type are lines connecting the point p_1 and the intersection of a pair of epipolar lines $\hat{\ell}(p_i, p_j) = p_1 \times (\ell(p_i) \times \ell(p_j))$. Each such line intersects the unit circle at a critical point, $c_{i,j}$. The second type are lines passing through p_1 and parallel to an epipolar line; we denote these lines by $\hat{\ell}(p_i)$. Each such line intersects the unit circle at a critical point, c_i. This is shown again in Fig. 2(a), where there are three critical points, defining three sectors, R_1, R_2 and R_3.

The number of possible temporal orderings, $|\Gamma_j|$, is bounded by the number of sectors (or critical points). In fact, it can be further reduced by eliminating sectors that do not fulfill the known temporal orders of images taken from the same camera. We next use a toy example to illustrate these bounds.

Two Ordered Image Pairs: Consider the case of two cameras that move and capture two images each. That is, $S = \{p_i\}_{i=1}^{4}$, such that the temporal order of each of the pairs $\{I_1, I_2\}$ and $\{I_3, I_4\}$ is known. In this case there are 6 critical points given by $\{c_i | \; 2 \leq i \leq 4\}$, and $\{c_{i,j} | \; i \neq j, 2 \leq i, j \leq 4\}$.

Fig. 2(b) shows an example of four of these points and the resulting sectors (R1-R4), while ignoring for the sake of clarity the critical points $c_{2,3}$ and $c_{2,4}$. We make the following claim:

Claim 1: *There are at most 4 possible orders that are both temporally and geometrically consistent.*

Before we prove this case, let's have a look at several general observations that can be verified geometrically.

1. **Time-Direction:** All lines in the same sector induce the same order, up to time-direction ambiguity. The known temporal order between two images, w.l.o.g. I_1 and I_2, is used to resolve this ambiguity. Hence, a single order is induced in each sector.

2. **Temporal Consistency:** The order induced in each sector may be either consistent or inconsistent with all the known temporal orders. In our example, if we are given that $t(I_1) < t(I_2)$ and $t(I_3) < t(I_4)$ (the green before the pink), then the sectors R_2 and R_4 are consistent, while R_1 and R_3 are not.

3. **Adjacent sectors:** The orders induced in adjacent sectors are different. In particular, when crossing a critical point, $c_{i,j}$, the order of $\ell(p_i)$ and $\ell(p_j)$ alternates. When crossing a critical point, c_i, the rank of $\ell(p_i)$ in the induced order is changed from the first to the last or vice versa. A special case is the critical point c_2, since $\ell(p_2)$ is used to define the time-direction. When crossing c_2, the time-direction flips to preserve the order of I_1 and I_2, and the rank of $\ell(p_2)$ in the induced order does not change, while the order of all others is reversed.

Proof Claim 1: We prove that at most 4 sectors are consistent with the known temporal orders. We assume that the time-direction is determined by the order of I_1 and I_2. Hence, the question is how many sectors are consistent with the temporal order of I_3 and I_4?

Let's consider only the critical points that affect the order of I_3 and I_4: c_2, c_3, $c_{3,4}$ and c_4. These points define 4 sectors (see Fig.2). From observation (3), the order of $\ell(p_3)$ and $\ell(p_4)$ alternates in these sectors. Hence, only two of them are consistent with the known order of I_3 and I_4.

Now consider the additional critical points, $c_{2,3}$ and $c_{2,4}$, that add two sectors. Each of them can split either an inconsistent sector or a consistent one. In the first case, the number of consistent sectors remains 2. In the second case, a consistent sector is replaced by two consistent ones. Thus, it follows that the maximum number of consistent sectors is 4 and is obtained if each of $c_{2,3}$ and $c_{2,4}$ splits a consistent sector. In this case, we end up with 4 valid permutations of the 4 images. QED.

Note that the proof is independent of the specific order of the critical points shown in Fig 2.

The General Case: Consider a set S such that $\mathcal{I}_S$ consists of images taken from k cameras, n_j images from camera j. The number of temporal permutations of $\mathcal{I}_S$ from a combinatorial point of view is given by:

$$\pi = n!/\prod_{j=1}^{k}(n_j!).$$

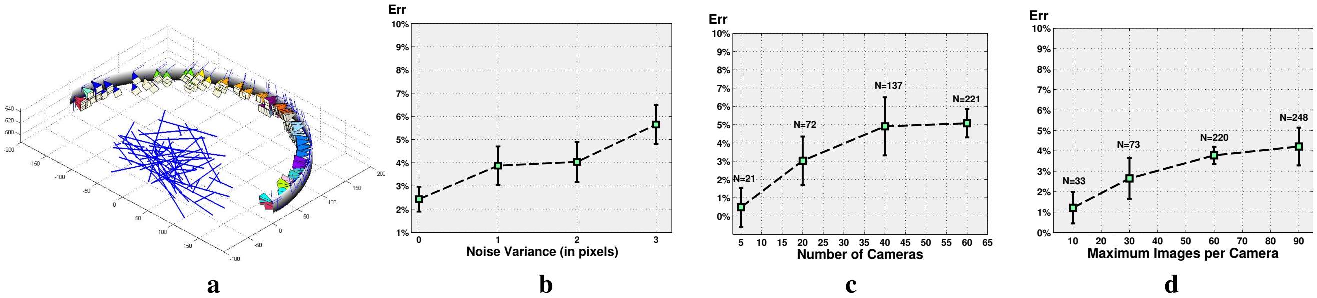

Figure 3. **Synthetic Experiment**: **(a)**, 3D visualization of the camera setup; the camera color encode the same camera at different locations; **(b)**, the computed mean error and its variance when adding Gaussian noise with increasing variance; **(c-d)**, the mean error and its variance when increasing the number of cameras, or increasing the maximum images taken by each camera, respectively; the number of images is shown with each bullet.

Note that π is smaller than $n!$ but still can be very large. For example, if we have 10 cameras taking 5 images each, then $\pi = 50!/(5!)^{10} = 4.9 \cdot 10^{43}$. Using the geometric constraint, the number of critical points is an upper bound of the number of orders: $\pi_{\text{CP}} \leq \binom{n-1}{2} + n - 1$. For the example above, $\pi_{\text{CP}} \leq 1225$.

As in the two ordered pairs case, we can further reduce the number of valid orders by determining the sectors that are temporally consistent. However, obtaining a complete characterization of the general case remains open.

Computing Γ in Practice: Given a feature set S we compute the set of $n(n-1)/2$ critical points. For each sector, R_j, a single representative line $\ell \in R_j$ is selected, and the temporal order induced by this sector is computed. In case all available temporal constraints for S are satisfied, the computed order is added to the order list, Γ. This process is repeated for all images in $\mathcal{I}_S$. The intersection $\Gamma = \Gamma_1 \cap \cdots \cap \Gamma_n$ is the final list of valid ordering consistent with the set S.

3.2. Rank Aggregation

We follow [4] and adopt the Markov Chain based solution of [6] to the rank-aggregation problem. Due to space limitation, we describe only the modifications with respect to [4]. In our case, each feature generally votes for more than a single order, and we set the weight of the vote to be inverse proportional to the number of orders it votes for ($|\Gamma^i|$). These votes are accumulated in the auxiliary matrix, which is used to compute the transitivity matrix, M. In addition, the global order should be consistent with the known temporal orders (of images taken by the same camera) in addition to the computed partial orders $\bigcup \Gamma^i$. Thus, we set the entries of M that correspond to the pairwise known temporal orders to have probability of 1 (maximum) and then normalize the rows of M again to sum to 1.

4. Results

We tested our method on both synthetic and real data. To quantitatively evaluate the results, we measured the percentage of incorrect pairwise orders out of the total number of image pairs, known as the *Kendall distance*. That is, the error ranges from 0% (the order is perfectly correct) to 100% (all pairwise orders are incorrect). For all the real datasets, the fundamental matrices between the image pairs were computed using the BEEM algorithm [7]. Feature points were detected and matched across images as in [4], using corners and NRDC matching [8].

4.1. Synthetic Data

We evaluated in a controlled manner two important properties of our method: robustness to noise and scalability. To this end, we generated synthetic data by simulating a 3D scene that is captured by freely moving cameras. The cameras were placed in a half circle, facing the center. The 3D scene consisted of multiple 3D lines that were chosen inside a bounding box located in the center of the circle (see Fig. 3).

The number of cameras as well as the number of 3D lines were manually set, whereas several other parameters were randomly generated, including the locations of the cameras, the 3D lines, and the number of images captured by each camera. Each 3D line was associated with a random subset of the cameras (in the range of 2% - 40% of the total number of cameras). For each chosen camera, the 3D line was projected to a random subset of its images.

Theoretically, there is no limitation to the scalability of our method as long as the motion of 3D points is not periodic, and provided that dynamic features are correctly matched in as many images as needed. Clearly, these conditions are not likely to hold in practice. To simulate a realistic scenario, we computed the fundamental matrices only for relatively proximate image pairs (i.e., the angle between their centers of projection and the center of the circle is below 60°). We repeat each test five times, and report the mean error and standard deviation. The following experiments were conducted:

Robustness to Noise: To assess robustness to noise, we added Gaussian noise with zero mean and increasing vari-

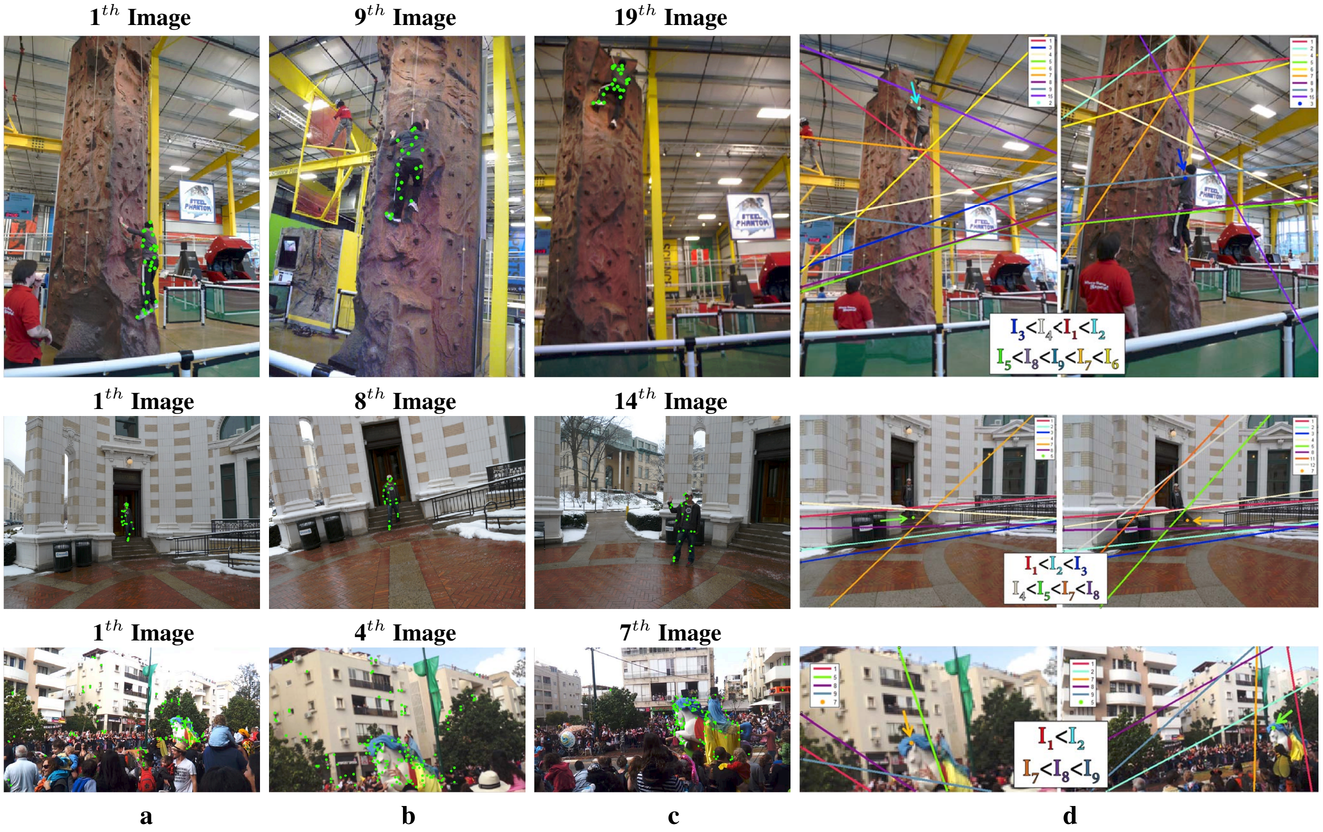

Figure 4. **RockClimbing, HandWave and Carnival: (a-c)**, the detected dynamic features are marked in green over the images, and **(d)**, the same feature set as seen in two different reference images; in each image, an arrow marks the feature point; the epipolar lines are marked in different colors according to their index (see legend); the available temporal constraints are shown on the bottom.

ance (measured in pixels) to the 2D features, while fixing the rest of the parameters. In particular, we used 54 images taken by 10 freely moving cameras (each camera provided a maximum of 10 images). The 3D scene consisted of 100 3D lines, where each line was projected to only 3 images on average; the fundamental matrices were computed between 58% of the image pairs. Fig. 3(b) shows the computed mean error and standard deviation. As expected, the mean error increases with the noise level. Yet, the maximum mean error is below 6% incorrect pairwise orders out of a total of 1431 image pairs.

Scalability: In this experiment, we evaluated the scalability of our method in the total number of images and considered two cases: increasing the number of cameras, or increasing the number of images captured by each camera. In both cases, we again used 100 lines, and added a fixed level of Gaussian noise with variance equal to one. The average set size (i.e., the number of images in which each dynamic feature appears) in all experiments was in the range of 3 to 7, and the fundamental matrices were computed for 60% of the image pairs.

Increasing the Number of Cameras: In each experiment we increased the number of cameras (and therefore the total number of images). The maximum number of images provided by each camera was fixed to 5 in all trials. As can be seen in Fig. 3(c), our method scales up with the number of images; the computed mean error increases at first but then stabilizes.

Increasing the Number of Images per Camera: Increasing the number of images per camera increases the number of ordering constraints, which should lead to a better solution. However, since there are more images to order, the number of possible permutations of all images is always increasing, making the problem more challenging. In each trial, we increased the maximum number of images provided by each camera (the actual number of images was randomly chosen for each camera in the range of 1 to a predefined maximum value). As can be seen in Fig. 3(d), the error indeed increases with the number of images, but the errors are lower than in the previous case.

4.2. Real Data – RockClimbing & HandWave

We tested our method on the real datasets, *RockClimbing* and *HandWave*, provided by [13]. In each of the datasets, the scene consists of one dominant non-rigidly moving object, captured by multiple, freely moving, photographers.

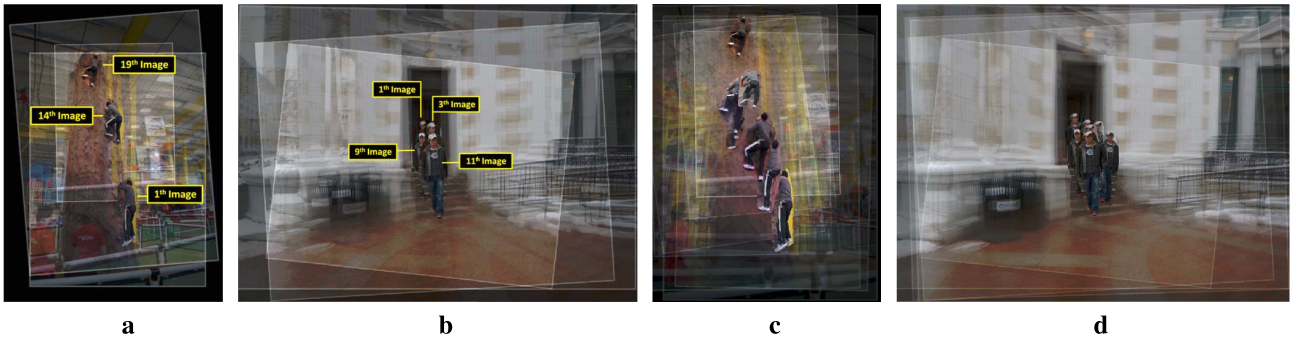

Figure 5. **Stroboscopic images:** the results of aligning the static regions of the images and overlaying them with proper masking (see Sec.4.5). **(a),(b)**, stroboscopic images that consist of 3 and 4 images, respectively; the images are nicely distributed in time; **(c),(d)**, stroboscopic images that consist of 6 and 7 images, respectively; the dynamic objects intersect.

Since our method assumes that the motion is not periodic, we extracted the relevant images for each of the datasets. For example, in the original RockClimbing dataset, the man climbs up and down the wall, but we extracted only the images up to the time he reaches the top.

The correspondences of dynamic features across the input images and the ground truth order were given by [13]. It is important to note that due to major differences in the viewpoints, the fundamental matrices could not be computed between all image pairs. 54% of the fundamental matrices were computed for *RockClimbing* and 70% for *HandWave*. Thus, although the dynamic features were manually extracted and matched across all images by [13], the actual size of the feature sets was much smaller.

Our method successfully recovered the correct temporal order of 19 images of *RockClimbing* taken by five cameras, and 14 images of *HandWave* taken by four cameras. The details of these experiments are summarized in the first two columns of Table 1. Fig.4(a)-(c) shows three of the input images for each dataset, each taken by a different camera. The dynamic features are marked in green over the images. In addition, Fig.4(d) shows the information available from a single feature set in two different reference images. For the *RockClimbing*, the feature set consists of 10 images, taken by 3 cameras, whereas the *HandWave* feature set consists of 7 images taken by 4 cameras.

Voting by 3D Reconstruction: Here, we tested the 3D approach to photo-sequencing (see Sec. 3.1) on *RockClimbing* and *HandWave* by replacing our 2D based voting scheme with a 3D reconstruction of lines. In particular, we used the method of Avidan & Shashua [16] to reconstruct the 3D linear trajectory of each 3D dynamic feature and the 3D locations along it. The temporal order is then given by the spatial order of the 3D locations along the line. Except for the voting, the rest of the framework remained the same. The calibration was obtained from [13] using structure from motion. For the HandWave dataset, 28% of the 91 pairwise orders were incorrect, and for the *RockClimbing* dataset 43% out of the 171 pairwise orders were incorrect. This demonstrates that 3D reconstruction is indeed highly sensitive to noise and deviations from the linear motion assumption, and is far more error-prone than our 2D voting approach.

4.3. Real Data – Carnival

This dataset consists of ten images taken by three hand-held mobile phones: iPhone 5 & two Galaxy SII. Two of the mobile phones provided four images each, while the third one provided two images. This dataset is very challenging because the scene consists of many transient objects (people in a crowd) that move inconsistently. In addition, the viewpoints are significantly different, making the preprocessing stage even more challenging and noisy. As can be seen in the last row of Fig. 4, some of the detected dynamic features are incorrect and belong to the background. The change in viewpoints can be seen by the varying background in each of the input images in the last row of Fig. 4.

For this dataset, 53% of the fundamental matrix pairs were computed. Note that [4] could not be applied here since each image was taken from a different viewpoint. More importantly, none of the images could be reliably matched to all other input images. We successfully obtained the correct order of all ten images.

4.4. Real Data – Boats

We tested our method on the *Boats* dataset provided by [4]. The original dataset consists of 15 images, taken by two hand-held mobile phones (iPhone 4), where ten of the images were taken by the first camera, and five by the other one. In [4], two reference images were manually selected, i.e., the image pair taken by the static camera. Our method was not provided with any prior information except for the known temporal constraints. We successfully recovered the correct temporal order with no error. We then added 9 more images to the original data (24 images in total) and still managed to have zero error.

The main challenge in this dataset was matching the dynamic features due to the change in appearance of the boats (avg. feature size was only 3.4). Since the viewpoints are

	R.C.	H.W.	Carnival	Boats
# Cameras	5	4	3	2
# Images	19	14	10	24
# Dyn. Features	45	65	147	514
Avg. Set Size	12.7	11	3.7	3.4
Avg. # Orders per Ref.	5.6	6.4	3.5	2.3
Avg. # Orders All Ref.	3.9	3.7	1.8	1.5

Table 1. For each of the datasets the table shows: number of cameras, number of images, average feature set size ($|S|$), average size of the order set per reference ($|\Gamma^i_j|$), and average size of the order set after intersection from all references, ($|\Gamma^i|$).

relatively close, the fundamental matrices were obtained for all image pairs. The details of this test are shown in the last column of Table 1.

4.5. Visualization & Alignment

We visualize our results by aligning the static regions of the images and overlaying them according to their computed order. Then the warped images can be played, in the computed order, as a single video sequence, or can be chained into one image (see Fig. 1). To do so, we compute the optimal similarity transformation of all images to the first image in the sequence. For images that were successfully matched to the first image, the transformation is directly computed. Otherwise, we find the shortest path in a graph in which each image is a node, and an edge exists between node i and j, if a matching was found between images I_i and I_j. If there is only one dynamic object in the scene, the aligned images can be integrated into a single "stroboscopic-like" image (See Fig. 5(a-b)). The temporal order can, in some cases, be determined by the spatial order of the features in the stroboscopic image but this will not work in the general case as, Fig. 5(c-d) shows.

5. Limitations & Conclusions

Our method relies on point matching to work properly. Since all we care about is the order of images, we can tolerate inaccuracy and partial information in both the computed geometry and the matching of dynamic features. However, we may fail when such errors lead to wrong voting, or in the degenerate case where the motion of an object is parallel to some of the epipolar lines. For example, when we added 6 images that caused wrong voting to the *RockClimbing* dataset, and two more to *HandWave*, the error increased to 4%, and 2.5%, respectively.

To conclude, we extended photo-sequencing to handle more general space-time camera configurations by dealing with space-time tradeoffs. In particular, we dropped the static camera assumption of [4] and compensate for the uncertainty in space by adding temporal certainty that stems from our knowing the order of images taken by each camera. Experiments on synthetic and real data demonstrate the advantages of our method.

Acknowledgments: This work was supported in part by Israel Science Foundation grant no. 1556/10 and 930/12, and European Community grant PIRG05-GA-2009-248527.

References

[1] I. Akhter, Y. Sheikh, S. Khan, and T. Kanade. Trajectory space: A dual representation for nonrigid structure from motion. *PAMI*, 2011.

[2] S. Avidan and A. Shashua. Trajectory triangulation: 3d reconstruction of moving points from a monocular image sequence. *PAMI*, 2000.

[3] L. Ballan, G. Brostow, J. Puwein, and M. Pollefeys. Unstructured video-based rendering: Interactive exploration of casually captured videos. *ACM (TOG)*, 2010.

[4] T. Basha, Y. Moses, and S. Avidan. Photo sequencing. *ECCV*, 2012.

[5] C. Bregler, A. Hertzmann, and H. Biermann. Recovering non-rigid 3d shape from image streams. In *CVPR*, 2000.

[6] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the web. In *IW3C2*, 2001.

[7] L. Goshen and I. Shimshoni. Balanced exploration and exploitation model search for efficient epipolar geometry estimation. In *ECCV*, 2006.

[8] Y. HaCohen, E. Shechtman, D. Goldman, and D. Lischinski. Non-rigid dense correspondence with applications for image enhancement. *SIGGRAPH*, 2011.

[9] R. Hartley and R. Vidal. Perspective nonrigid shape and motion recovery. In *ECCV*, pages 276–289, 2008.

[10] J.-Y. Kaminski and M. Teicher. A general framework for trajectory triangulation. *JMIV*, 2004.

[11] B. Meyer, T. Stich, M. Magnor, and M. Pollefeys. Subframe temporal alignment of non-stationary cameras. In *BMVC*, 2008.

[12] F. L. C. Pádua, R. L. Carceroni, G. A. M. R. Santos, and K. N. Kutulakos. Linear sequence-to-sequence alignment. *PAMI*, 2010.

[13] H. Park, T. Shiratori, I. Matthews, and Y. Sheikh. 3d reconstruction of a moving point from a series of 2d projections. *ECCV*, 2010.

[14] D. Pundik and Y. Moses. Video synchronization using temporal signals from epipolar lines. In *ECCV*, 2010.

[15] G. Schindler and F. Dellaert. Probabilistic temporal inference on reconstructed 3d scenes. In *CVPR*, 2010.

[16] A. Shashua and L. Wolf. Homography tensors: On algebraic entities that represent three views of static or moving planar points. *ECCV*, 2000.

[17] N. Snavely, S. Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3d. In *SIGGRAPH*, 2006.

[18] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *IJCV*, 1992.

[19] L. Wolf and A. Shashua. On projection matrices $p^k \rightarrow p^2$, k=3,...,6 and their applications in computer vision. *IJCV*, 2002.