Who Shot the Picture and When?

Gagan Kanojia, Sri Raghu Malireddi, Sai Chowdary Gullapally, and Shanmuganathan Raman

Electrical Engineering, Indian Institute of Technology Gandhinagar, India
{gagan.kanojia, sriraghu_malireddi, sai_gullapally, shanmuga}@iitgn.ac.in

Abstract. Consider a set of images corresponding to a dynamic scene captured using multiple hand-held cameras. Assuming that we do not have any other information about the camera settings and the dynamic scene, we would like to identify the cameras which captured each of these images. Further, we would like to estimate the order in which these images were captured by each of the cameras. We address this challenging problem using principles derived from multiple view geometry and unsupervised learning techniques. We show that the camera identification problem can be modelled as clustering of the affine camera matrices estimated from the images. We show that homography estimation from the static regions of the scene enables us to order the images captured by each camera individually. Apart from discussing the advantages of the proposed approach, we conclude the paper providing the limitations of the approach and future directions.

1 Introduction

Analysing a dynamic scene captured using a hand-held camera is of great interest to computer vision and computer graphics community recently. This interest is due to the emergence of mobile phone cameras with high spatial and temporal resolutions. A natural scene is often dynamic as objects can change their positions when one captures multiple images of the scene at different time instants. The motion of the objects can be estimated using traditional optical flow techniques if the camera is placed on a tripod and the object motion is not too large [1]. This scenario does not arise when hand-held cameras in the mobile phones and other digital devices are used to capture the dynamic scene over time.

Suppose a dynamic scene is captured by multiple persons with hand-held cameras, inferring the camera which shot each of the images is a challenging problem. Recent algorithms such as photo sequencing [2] have assumed that both this information and the order in which the images were shot by each camera are known. This is a very strong constraint which restricts the use of these algorithms in practical situations. Also, given a sequence of ordered images from a single camera the ascending or descending order can only be determined with additional learning [3].

In this work we provide an unsupervised algorithm, to not only predict the source of each of the images but also order them in the captured sequence. This

solution requires us to understand and develop an algorithm with principles from the multiple view geometry. The proposed approach is extended to localize the changes caused due to moving objects in each of the images corresponding to a camera with respect to a reference image. The major contributions of the work are listed below.

1. Given m images of a dynamic scene captured using n hand-held cameras we are able to assign these images correctly to each of the n cameras.
2. Suppose α_i be the number of images assigned to the i^{th} camera. We developed an algorithm to arrange these α_i images in the order in which they were shot by the i^{th} camera.

The rest of the paper is organized into the following sections. Section 2 discusses the work related to the proposed approach and the techniques used therein. Section 3 explains the proposed approach in detail(Figure 1). Section 4 presents the results and discussions for various dynamic scenes captured using varying number of cameras. Section 6 presents the summary of the proposed approach and possible directions to improve it in future.

2 Related Work

The works very closely related to the application we address are photo sequencing [2] and seeing the arrow of time [3]. The work by Basha *et al.* on photo sequencing introduces an approach to time sequence a set of images of a dynamic scene captured using multiple cameras. This work focuses on ordering the images captured by multiple cameras in a single sequence in the common time frame in which they were shot. Apart from assuming information regarding which camera shot which picture, this work also assumes that we have at least two images captured by the same camera from a fixed location. A more recent work addresses this problem by just assuming that the images shot by each camera and their ordering in which they were shot are known [4]. Both these works do not predict whether the arrangement of images on a common time frame is ascending or descending. This can be estimated using supervised learning on various video sequences as proposed in [3].

One of the commonly used algorithms developed for the purpose of interest point detection and description is the scale invariant feature transform (SIFT) algorithm [5]. After SIFT, there have been other related algorithms developed for addressing the interest point detection and description task [6]. Another commonly used feature descriptor is the speeded-up robust feature (SURF) algorithm [7]. The objects which are non-rigid can be matched across 2 images using the non-rigid dense correspondence (NRDC) algorithm better than SIFT and SURF descriptors [8]. In the present work, we use NRDC to match features between the images.

Alignment of successive frames in a video sequence by matching spatial features over time is a common task in different computer vision applications [9].

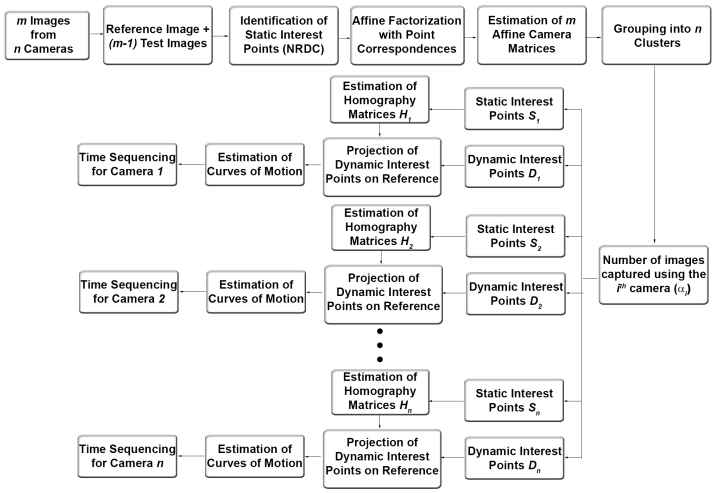

Fig. 1. The Proposed Approach - Who shot the picture and when?

This problem has been extended to videos captured using different unsynchronized, non-stationary cameras. The registration process is challenging and has been addressed by a recent work [10]. In the present work, we address the problem of alignment of images captured using multiple non-stationary cameras.

Consider a static scene captured by multiple cameras separated in space. The position of the camera and the structure of the objects in the scene can be estimated jointly using factorization methods [11]. The cameras can be assumed to be either projective or affine. The affine assumption simplifies the factorization algorithm [12]. Images captured using hand-held cameras can be used to model 3D scenes by using factorization methods [13].

3 Proposed Approach

In this work, we deal with the images captured using multiple hand-held cameras from different positions. We assume that we know neither the sequence in which the images are captured nor the cameras using which they are captured. Global registration of these images is not feasible as the scene is dynamic [14]. In the case of a dynamic scene, the interest points in the images of the scene can be classified into static and dynamic. If we can achieve this task, the static and dynamic interest points present can be processed independent of each other.

3.1 Identification of Static Interest points

We observe that there will be two kinds of regions in any given image from the set of images - static and dynamic. The interest points computed in the image

(a) (b) (c)

Fig. 2. (a,b) Two images of a dynamic scene, (c) Features matched between images in (a) and (b).

should fall into one of these regions. Consider any two images of the scene and the feature correspondence for all the interest points in the two images. It can be observed that we can predict that all the points in the images corresponding to static regions in the scene will be shifted in the same direction between these two images (see Figure 2(c)). Our objective is to segregate those interest points which exhibit motion in a single direction from the rest. To achieve this, we further assume that the number of dynamic interest points in a given image is considerably less compared to the static interest points.

Consider a dataset of m images of a dynamic scene captured using n hand-held cameras. To start with, consider one image from this set of images as the reference image. We shall compare the remaining $(m-1)$ images with respect to the reference image. For each of these $(m-1)$ images we estimate NRDC correspondence with respect to the reference image (see Figure 2) [8].

Consider that there are k_i $(i = 1, 2, \ldots, m-1)$ matched points for each of the $(m-1)$ images with respect to the reference image. For each of these k_i matched points, we calculate the displacement vector and store them in matrices A_i of dimension $k_i \times 2$, where $(i = 1, 2, \ldots, m-1)$. We perform singular value decomposition (SVD) on these matrices A_i [15]. SVD of matrices A_i are given by the equation $A_i = U_i \Sigma_i {V_i}^T$. The singular values σ_{i1} and σ_{i2} represent the variation in two different directions. As we have initially assumed that the number of dynamic interest points in the scene are much less compared to the static interest points, so the static interest points should represent the most significant variation in the unit displacement vectors. So we replace σ_{i2} with zero and form a set of new diagonal matrices Σ'_i. We reconstruct the matrices A'_i using only the dominant singular values as given by the equation, $A'_i = U_i \Sigma'_i {V_i}^T$.

These matrices capture information only about the most prominent displacement direction (assumed to be static). From the normalized estimated matrix A''_i we compute the magnitudes of the displacement in a vector $|{A_i}''|$ of size k_i. From $|A''_i|$, we calculate the average displacement magnitude using equation 1. Let $a'_{i,j}$ be the j^{th} element in the vector A'_i.

$$a''_{i,j} = \frac{a'_{i,j}}{|a'_{i,j}|}, \quad \overline{|A''_i|} = \frac{\sum_{j=1}^{k_i} |a''_{i,j}|}{k_i} \tag{1}$$

where $a''_{i,j}$ is the j^{th} element in the vector A''_i. We consider only those displacement vectors from A'_i whose magnitude of difference with the average displacement vector $\overline{|A''_i|}$ is less than a small real number ϵ. We form new matrices corresponding to static interest points (S_i) using equation 2.

$$S_{i,j} = \begin{cases} A_{i,j} & ||a'_{i,j}| - \overline{|A''_i|}| < \epsilon \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where $S_{i,j}$ and $A_{i,j}$ correspond to the j^{th} rows of the matrices S_i and A_i respectively. We choose the parameter ϵ empirically to be a very small real number. In the matrices obtained through equation 2, all the non-zero rows correspond to the interest points present in the static region of the image. We will exploit these static interest points in order to recover the n camera matrices.

3.2 Who took the picture?

Having found out the static interest points in each of the m images, we need to recover the camera matrices corresponding to each of these images. We shall assume that the underlying camera matrices for these n cameras are affine. The first step is to find the static points which are common to all the m images which are estimated iteratively. Let the number of common static interest points estimated be β. For recovering the n affine camera matrices we require that the number of common interest points β to be greater than or equal to 4, as it is the minimum requirement to perform affine factorization. We then calculate the centroids of all the points in each of the images using the equation 3.

$$t_i = \frac{\sum_{j=1}^{\beta} \mathrm{x}_{i,j}}{\beta}, \mathrm{x}_{i,j} = (x_{i,j}, y_{i,j})^T \tag{3}$$

where t_i are the centroids of all the static interest points estimated for the i^{th} image.

We subtract these centroids from the corresponding static interest points extracted from each of the m images using the equation $\widetilde{\mathrm{x}}_{i,j} = \mathrm{x}_{i,j} - t_i$, where $\widetilde{\mathrm{x}}_{i,j}$ are the normalized interest points in the i^{th} image and $j = 1, 2, \ldots, \beta$.

With the normalized coordinates corresponding to β matched points in each of the m images, we generate a matrix W with dimensions $2\beta \times m$. We use SVD to decompose this matrix W as shown in equation 4.

$$W = U_{\mathrm{W}} \Sigma_{\mathrm{W}} V_{\mathrm{W}}^T \tag{4}$$

where U_{W} is of dimension $2\beta \times 2\beta$, Σ_{W} has dimensions $2\beta \times m$ and V_{W}^T has dimensions $m \times m$.

Part of the affine camera matrices (M_i) are generated by multiplying the first three columns of U_{W} with the first three singular values of Σ_{W} [11]. Now M_i and t_i represent the affine camera matrix corresponding to the i^{th} image with the third row being $(0, 0, 0, 1)$. We form a 8 dimensional vector for each image by

augmenting the first two rows of estimated the affine camera matrix to form a feature descriptor. Therefore, we have an eight dimensional feature descriptor for each of the m images. We will apply clustering algorithm to group these feature descriptors into n clusters. These n clusters represent n different cameras. In this work, we use K-means clustering to achieve this task assuming that the number of cameras n is known [16]. This enables us to identify the camera which shot each of the m images. After clustering we obtain α_i which represent the set of images captured from the i^{th} camera ($i = 1, 2, \ldots, n$).

3.3 Identification of Dynamic Interest Points

There are sets of coordinates corresponding to all the interest points stored in the matrices A_i. The matrices S_i are extracted from matrices A_i by selecting the coordinates of the interest points present in the static regions. Those points in the matrices A_i which have not been accounted for in matrices S_i will represent the interest points corresponding to dynamic regions of the scene. This is due to the fact that we consider only the matched interest points which have been given high confidence value by the NRDC algorithm. This also enables us to get rid of any false matches or outliers. We form matrices D_i which capture all the points

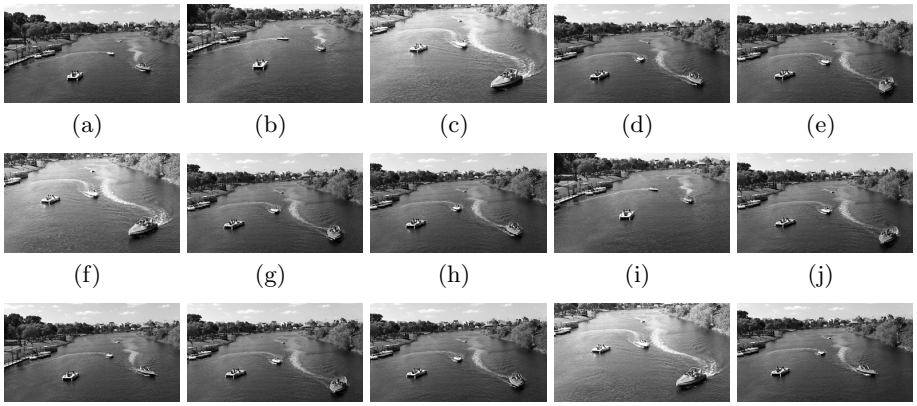

Fig. 3. (a-o) Images of a dynamic scene captured using two cameras.

which are present in A_i but not in S_i. From the set of displacement vectors in D_i we retain those displacement vectors whose slopes differ significantly from the average slopes computed from the matrices S_i. This process enables us to retain only those displacement vectors in D_i which most probably correspond to dynamic regions of the scene.

3.4 When?

Having classified the images captured by each of the n cameras, our next task is to arrange them in time sequences. In the absence of accurate depth information of the scene, two images of a static scene captured using the same camera at different positions can be related using a homography matrix as shown in the equation $\mathrm{x}_2 = H_{3\times3}\mathrm{x}_1$, where x_1, x_2 are the homogeneous representations of the points in the first and second images respectively and $H_{3\times3}$ is the homography matrix or 2D projective transformation matrix relating the two images.

We pick one out of the α_i images captured from each camera as the reference image. We compute the homography matrices H_i by using the static interest points S_i of rest of the $(\alpha_i - 1)$ images with the corresponding static interest points of the reference image [11]. This process is repeated for all the cameras. Now we will have a set of $(\alpha_i - 1)$ homography matrices H_i for the images captured by the i^{th} camera. We employ these homography matrices on the dynamic points present in the matrices D_i to bring them to the same coordinate frame. We plot the homography transformed dynamic points on a separate image grid for each of the cameras. If we plot either a line or a polynomial curve through the

Fig. 4. Estimated curve through dynamic points which depict the sequence of captured images by camera 1.

homography transformed dynamic points, we can estimate the dynamic points which are closest to this fitted curve in the least squares sense. By tracing along the fitted curve, we will be able to order the projected dynamic points obtained from the various images. Classifying the motion as either ascending or descending (arrow of time) is a challenging task which requires us to use supervised learning techniques [3].

We shall present the various experiments we conducted using the proposed approach and justify the effectiveness of the proposed approach using results obtained by processing different datasets.

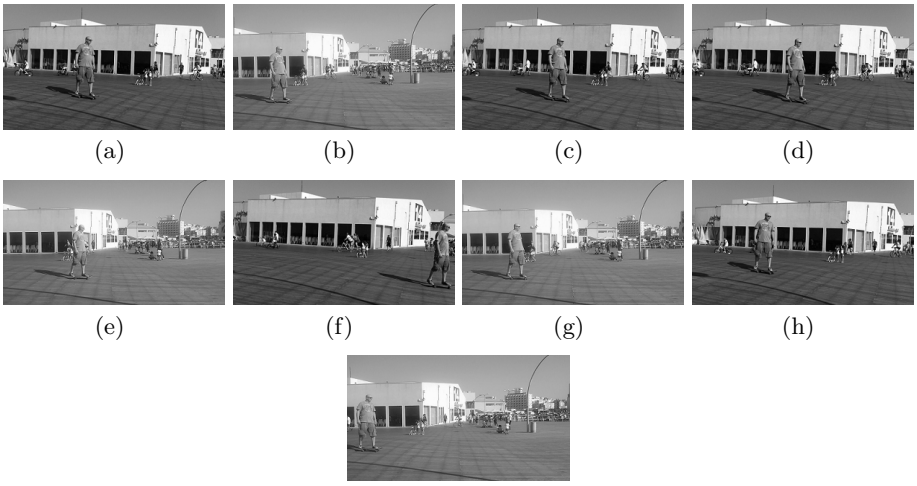

Fig. 5. (a-i) Images of a dynamic scene captured using two cameras.

4 Results and Discussion

Consider the dataset consisting of 15 photos captured by two cameras as shown in Figure 3 [2]. Here $m = 15$ and $n = 2$. We need to determine the images belonging to two different clusters corresponding to the two cameras. We require at least four point correspondences to perform affine factorization as discussed earlier. We would now like to process these 15 images using the proposed approach mentioned in sections 3.1 and 3.2. We cluster these images into two cluster corresponding to the two cameras and verify the proposed approach with the prior knowledge regarding the cameras. We are able to identify all the images captured using first camera - Figure 3 (a, d, e, g, h, j, k, l, m, o) and second camera - Figure 3 (b, c, f, i, n) respectively.

We select one reference image from each of the clustered image sets of camera 1 and camera 2 and follow the steps in sections 3.3 and 3.4. The resultant dynamic points projected using the homography matrices (H_i) captured using camera 1 are shown in Figure 4. Figure 4 shows the curves through different dynamic objects in the reference image corresponding to camera 1. As we already know the origin of each dynamic point projected on the reference image we can order the rest of the images sequentially. We assume that the ordering is ascending along the curve of motion (in this case, line). The ordered images we got from camera 1 with respect to a reference are Figure 3 (a, o, d, h, m, i, g, e, l, j). The resulting ordered images captured using camera 2 with respect to a reference are Figure 3(b, i, f, n, c) (not shown).

Consider the dataset of 9 images taken using 2 cameras shown in Figure 5. Applying the proposed approach on this dataset we got the images captured using camera 1 as Figure 5(a, c, d, f, h) and camera 2 as Figure 5(b, e, g, i). The

Fig. 6. Estimated curve through dynamic points which depict the sequence of captured images by camera 2.

curve of motion obtained is shown in Figure 6. The ordered images for camera 1 are obtained as Figure 6 (h, a, c, d, f) and camera 2 as Figure 6 (i, b, e, g). The third dataset consists of 7 images taken using 5 cameras as shown in Figure 7. After applying the proposed approach the result is as follows, Figure 7(a,d) - camera 1, Figure 7(c,e) - camera 2, Figure 7(b) - camera 3, Figure 7(f) - camera 4 and Figure 7(g) - camera 5 respectively.

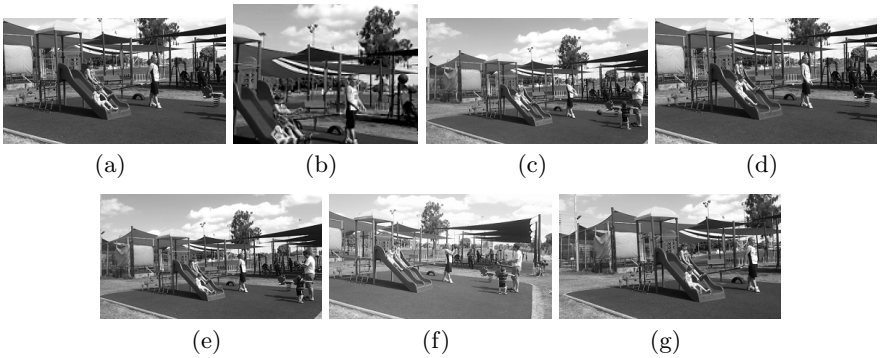

Fig. 7. (a-g) Images of a dynamic scene captured using five cameras.

All the cameras involved in the capture of a dynamic scene should preserve the same camera settings such as focus, exposure time, and aperture. If the number of cameras is large or the perspective views of the cameras are too different, the relative view points of the cameras may be different from each other. We may not be able to find sufficient static interest points corresponding to the same scene points across all the images in the given dataset. This would make the process of affine factorization challenging. Further, the homography matrix based dynamic interest point projection may be erroneous due to depth variation.

5 Conclusion

We have developed a novel approach to identify the cameras which captured a set of images of a dynamic scene. We could also determine the sequence in which the images were captured by each camera. The most salient feature of our approach is that we do not assume any knowledge of camera parameters and dynamics of objects present in the scene. The approach is fully automated and could serve as a precursor to existing applications such as photo sequencing. Further all the cameras are assumed to be hand-held and be avoided performing any type of pixel-wise registration. We would like to extend the proposed approach to deal with scenes exhibiting complex motions such as crowd movement and repetitive motion. We would also like to extend the proposed approach to process images captured by more number of cameras arranged in different spatial locations.

References

1. Horn, B.K., Schunck, B.G.: Determining optical flow. In: 1981 Technical Symposium East, International Society for Optics and Photonics (1981) 319–331
2. Basha, T., Moses, Y., Avidan, S.: Photo sequencing. In: ECCV. Springer (2012) 654–667
3. Pickup, L.C., Pan, Z., Wei, D., Shih, Y., Zhang, C., Zisserman, A., Schölkopf, B., Freeman, W.T.: Seeing the arrow of time. In: CVPR, IEEE (2014)
4. Dekel, T., Moses, Y., Avidan, S.: Space-time tradeoffs in photo sequencing. In: ICCV, IEEE (2013)
5. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International journal of computer vision **60** (2004) 91–110
6. Tuytelaars, T., Mikolajczyk, K.: Local invariant feature detectors: a survey. Foundations and Trends® in Computer Graphics and Vision **3** (2008) 177–280
7. Bay, H., Tuytelaars, T., Van Gool, L.: SURF: Speeded up robust features. In: ECCV. Springer (2006) 404–417
8. HaCohen, Y., Shechtman, E., Goldman, D.B., Lischinski, D.: Non-rigid dense correspondence with applications for image enhancement. In: ACM Transactions on Graphics (TOG). Volume 30., ACM (2011) 70
9. Caspi, Y., Irani, M.: Spatio-temporal alignment of sequences. Pattern Analysis and Machine Intelligence, IEEE Transactions on **24** (2002) 1409–1424
10. Meyer, B., Stich, T., Magnor, M.A., Pollefeys, M.: Subframe temporal alignment of non-stationary cameras. In: BMVC. (2008) 1–10
11. Hartley, R., Zisserman, A.: Multiple view geometry in computer vision. Cambridge university press (2003)
12. Triggs, B.: Factorization methods for projective structure and motion. In: CVPR, IEEE (1996) 845–851
13. Pollefeys, M., Van Gool, L., Vergauwen, M., Verbiest, F., Cornelis, K., Tops, J., Koch, R.: Visual modeling with a hand-held camera. International Journal of Computer Vision **59** (2004) 207–232
14. Zitova, B., Flusser, J.: Image registration methods: a survey. Image and vision computing **21** (2003) 977–1000
15. Trefethen, L.N., Bau III, D.: Numerical linear algebra. Volume 50. SIAM (1997)
16. Flach, P.: Machine learning: the art and science of algorithms that make sense of data. Cambridge University Press (2012)