Computer Vision and Image Understanding 160 (2017) 36–44

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

Detecting moving regions in CrowdCam images

Adi Dafni[a,*], Yael Moses[b], Shai Avidan[a], Tali Dekel[c]

[a] Tel Aviv university Israel, Israel
[b] Efi Arazi School of Computer Science, The Interdisciplinary Center, Herzliya, Israel
[c] Google Research

ARTICLE INFO

Article history:
Received 3 January 2017
Revised 18 March 2017
Accepted 11 April 2017
Available online 26 April 2017

Keywords:
Motion detection
CrowdCam
Epipolar geometry

ABSTRACT

We address the novel problem of detecting dynamic regions in CrowdCam images – a set of still images captured by a group of people. These regions capture the most interesting parts of the scene, and detecting them plays an important role in the analysis of visual data. Our method is based on the observation that matching static points must satisfy the epipolar geometry constraints, but computing exact matches is challenging. Instead, we compute the probability that a pixel has a match, not necessarily the correct one, along the corresponding epipolar line. The complement of this probability is not necessarily the probability of a dynamic point because of occlusions, noise, and matching errors. Therefore, information from all pairs of images is aggregated to obtain a high quality dynamic probability map, per image. Experiments on challenging datasets demonstrate the effectiveness of the algorithm on a broad range of settings; no prior knowledge about the scene, the camera characteristics or the camera locations is required.

1. Introduction

CrowdCam images are images captured by a crowd of people. These images usually capture some interesting dynamic event, and the dynamic objects are often where the attention should be drawn. It is therefore useful to ask whether the dynamic regions of a scene from CrowdCam images can be detected. A method for detecting these regions can be used to propose image windows that are likely to contain an object of interest. Other computer vision applications that can benefit from such a method include change detection, moving object segmentation, and action recognition.

In this paper, we address the novel problem of detecting the dynamic regions in a scene from CrowdCam images. As these images may be taken with a wide baseline in space and time, significant new challenges arise, such as distinguishing an object that moved from one whose appearance changed due to changes in the camera's viewpoint or occlusions (as demonstrated in Fig. 1).

CrowdCam images violate the assumptions made by existing methods. Background subtraction methods assume the camera is static, or at least that the images can be properly aligned. Motion segmentation algorithms usually work on video, which has a high temporal frame rate and small baseline between successive frames. In CrowdCam images, on the other hand, the images are few and far between, thus they cannot necessarily be aligned, makeing it impossible to preserve the spatial-temporal continuity. Finally, co-segmentation methods assume that the appearance of the background significantly changes from frame to frame. However, as CrowdCam images capture the same event, the background is usually consistent. Moreover, co-segmentation methods do not distinguish between static and dynamic objects.

Detecting dynamic regions in CrowdCam images can also be considered a by-product of running a dense Structure-from-Motion (SFM) procedure. The static regions will be matched and reconstructed in 3D. All the remaining pixels belong, by definition, to the dynamic regions. In practice, dense correspondence in CrowdCam images is not possible for each pixel (hence there are holes in the reconstructions) and is prone to many errors." For the static regions, the wide baseline causes changes of appearance and occlusions. The moving objects cause additional occlusions (see Fig. 2) and may undergo significant deformations, due to non-rigid motion. This makes it very difficult to reliably match them across images (as demonstrated in Section 4). A straightforward use of epipolar constraint for distinguishing between dynamic and static regions (e.g., Luong and Faugeras, 1996; Yuan et al., 2007), will also suffer from matching failures on CrowdCam data.

We propose a novel method for detecting the dynamic regions of a scene from CrowdCam images. Our method avoids 3D reconstruction and does not rely on establishing dense correspondences between the images. We assume that epipolar geometry can be computed between some pairs of images. We treat each image as

* Corresponding author.
E-mail address: adidafni@gmail.com (A. Dafni).

http://dx.doi.org/10.1016/j.cviu.2017.04.004
1077-3142/

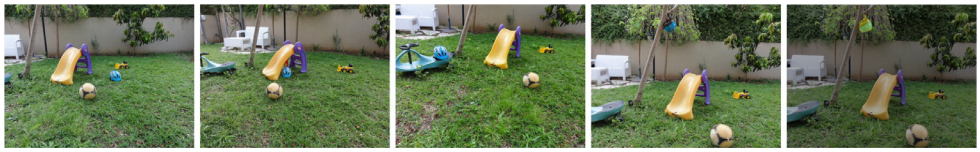


Fig. 1. A set of images captured at different times by different cameras. Which object moved? Right image reveals the answer.

Fig. 2. Types of occlusions in a wide baseline image pair with a moving object: Consider images (a) and (b) – corresponding epipolar lines are marked in yellow, ellipses indicate locations that are occluded by (i) an out of field of view (red), (ii) different viewpoints (purple), (iii) a moving object such as the chicken toy (black). When image (c) is considered as well (corresponding epipolar lines between (b) and (c) are marked in magenta), the red suitcase correspondence occluded in (a) by the moving object is revealed. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Fig. 3. Three images out of a set of eight: (c) was regarded as the reference image; (a) and (b) are two of the seven support images. The dynamic probability map and a its thresholded map are presented in (d) and (e), respectively.

a reference, and compute a *dynamic probability map* that represents the probability of each pixel to be a projection of a dynamic 3D point. An example of a set of three out of eight images, the computed dynamic probability map for one of the images, and a thresholded map are presented in Fig. 3. The maps clearly contain information about the dynamic regions.

The method works as follows. First, the dynamic probability map is computed for a reference image and another (support) image from the set, for which the epipolar geometry can be computed. The probability of a pixel to be a projection of a general moving 3D point depends on the probability that it has a match in the other image along the corresponding epipolar lines; projections of static background must lie on corresponding epipolar lines. We do not try to find the correct match, but instead compute the likelihood that there exists at least one potential match along the epipolar line. In this way we capture the likelihood that a match exists: doing so decreases the probability that the pixel is *dynamic*. This method allows us to deal with matching errors that are due to lack of texture, repeated structure, occlusions, and so on. To reduce ambiguity, the matching is defined on epipolar patches, i.e., patches confined by pairs of matching epipolar lines.

Each image may serve as a reference image associated with a subset of support images (for which epioplar geometry can be computed). We then aggregate, for each reference image, the matching probability maps computed using each of its support images, to obtain the final high quality dynamic probability map. This aggregation is necessary because a single map may be unreliable due to accidental matching, resulting in low dynamic probability, or due to occlusions, resulting in high dynamic probability (see Section 3.1). However, these cases are unlikely to consistently repeat w.r.t. all support images. Hence, the results improve as the number of support images increases.

The main contributions of this paper are (i) the introduction of the new problem of detecting dynamic regions from CrowdCam data; (ii) a voting-based approach that avoids dense correspondence or 3D modeling; (iii) aggregation of information from multiple views for distinguishing between moving objects and occluded regions; (iv) using candidate *epipolar patches* matching while avoiding rectifications between each pair of images (see Section 3.1.1).

2. Related work

We next review existing methods for detecting moving regions under various setups.

Change Detection: Change detection algorithms detect regions of change in images of the same scene taken at different times, where the regions of change often coincide with the moving objects. The change detection algorithms are based on comparing a frame to a learned model of the background. A necessary preprocessing step is accurate alignment of several images into the same coordinate frame (often obtained by using a static camera). Once the images are aligned, the background model can be generated and compared to a new frame. The background model may be at the pixel level (e.g., Stauffer and Grimson, 1999), region level (e.g., Klare and Sarkar, 2009), or the frame level (e.g., Wang and Suter, 2006). More sophisticated algorithms also model the foreground appearance. A comprehensive survey of background subtraction methods is provided in Cristani et al. (2010); Lu et al. (2004); Radke et al. (2005).

Change detection algorithms are not applicable to CrowdCam sets such as those used in our setup, where the images are captured by cameras with a wide baseline, the scenes are not necessarily flat or distant, and the images cannot be aligned (see supplementary material).

Motion-based segmentation: Motion-based segmentation separates regions in the image that correspond to different motion entities. It usually deals with video sequences. The classic approach to motion segmentation is based on two-frame optical flow, while recent approaches consider a set of frames and examine the movement characteristics over time (Ochs et al., 2014; Sun et al., 2012). While early approaches estimate the optical flow and the segmentation independently (Shi and Malik, 1998; Wang and Adelson, 1994), optical flow estimation and segmentation were later considered as a joint optimization problem (Brox et al., 2006; Cremers and Soatto, 2005; Sun et al., 2012). In our case, no video sequence is available, hence optical flow methods are not applicable.

Only two papers introduce methods for segmenting motion fields computed from a wide-baseline pair of still images, which is similar to the setup we considered (Wang et al., 2015),Gullapally et al.. The first method is based on matching feature points, and then minimizing a function that divides the matching into continuous groups of rigid motions. We, on the other hand, do not assume rigid motion, nor do we assume that correspondence between features of moving objects can be computed. The second method is based on computing dense correspondence and segment them into two main motions. The algorithm is limited to regions where dense correspondence can be calculated. We show in Section 4 the limitations of dense correspondence methods on our datasets.

Co-segmentation: Co-segmentation is typically defined as the task of jointly segmenting 'something similar' in a given set of images. Existing co-segmentation approaches cast this problem as a Markov Random Field (MRF) based segmentation of the image pair with a regularized difference of the two histograms, assuming a Gaussian prior on the foreground appearance (Rother et al., 2006) or by calculating the sum of squared differences (Mukherjee et al., 2009). The problem of co-segmentation is different from the one we aim to solve, since it does not distinguish between static and dynamic objects. Moreover, co-segmentation assumes shared appearance models for the foreground but different backgrounds, where in CrowdCam data of dynamic scenes, the background is similar while the appearance of dynamic objects may undergo significant deformations.

Multi-view object segmentation: Algorithms of this family address the task of unsupervised multiple image segmentation of a single physical object, possibly moving, as seen from two or more calibrated cameras. The input may either be still images or video sequences. Gang and Long (2004) coined the problem, and proposed an initial rudimentary silhouette-based algorithm for building segmentations consistent with a single 3D object. Many methods follow this initial trend by building explicit 3D object reconstructions and alternating with image segmentations of the views based on foreground/background appearance models (Campbell et al., 2010; Guillemaut and Hilton, 2011).

Our method avoids the 3D reconstruction. Recovering the 3D structure of a dynamic scene often requires prior knowledge about the 3D structure or the motion of objects, and a very large number of images, which we do not assume to have. The limitations of dense 3D reconstruction on our data are presented in detail in Section 4.

Yet another line of related work is that of objectness proposal, where the goal is to suggest image windows that are likely to contain an object of interest (e.g., Alexe et al., 2012). These methods work with a single image and therefore cannot reason about motion information, which often indicates the interesting regions. Our method can therefore be integrated into objectness proposal algorithms, in addition to other single image cues such as saliency, color contrast and edge density.

3. Method

We are given a set of n images, taken by various uncalibrated cameras. We assume that for each image we can compute its epipolar geometry w.r.t. a subset of images, termed *support set*. This assumption holds when there are sufficient static features in the set of images and the dynamic features are treated as outliers by a RANSAC algorithm which is used to compute the epipolar geometry (e.g., Goshen and Shimshoni, 2008). For each image, we compute a matching probability map based on its epipolar geometry with each of its support images and then merge all those maps into a dynamic probability map for that image. We next describe a method to compute a matching probability map from a pair of images, and then discuss the aggregation of these maps to compute a dynamic probability map.

3.1. Pair of images

Given a reference image I and a single support image, I_s, we compute $P(\mathbf{x}|I_s)$, the probability that a pixel, $\mathbf{x} \in I$, is static and non-occluded. Observe that $P(\mathbf{x}|I_s)$ is low not only for pixels in dynamic regions, but also for pixels in the following regions: (i) Out of field of view; (ii) Occluded due to different viewpoints; (iii) Occluded by the moving object in I_s, e.g., Fig. 4(c) (we refer to these regions as *dynamic object shadows*); (iv) Regions for which the descriptor fails to detect the similarity due to variations in appearance. Such variations exist due to the change of viewpoint and/or illumination, and the difference between the cameras' inner parameters.

It is evident from this list that failure to find a match does not necessarily mean that the pixel belongs to a dynamic region. As we show later, when using many support images, the probability of finding matches for static pixels will be significantly higher than for dynamic pixels (see also Fig. 6).

3.1.1. The set of epipolar patches:

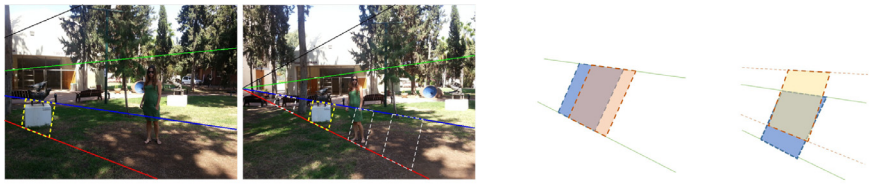
Matching a single pixel is very noisy and we work with patches instead. The probability that a pixel $\mathbf{x}$ has a match is derived from the probability that each of the patches covering $\mathbf{x}$ has a match.

When building a set of candidate pairs of patches for correspondence, the first step in the calculation is to define the patch's shape and size in I and in I_s. Rectangular patches are commonly used, but they are more appropriate for rectified pairs. Since in the general case the epipolar lines are not parallel, each of the possible matches may be of different height. We consider patches that are confined between pairs of epipolar lines – *epipolar patches*. The correspondence of a static epipolar patch in I is an epipolar patch in I_s, confined by the pair of corresponding epipolar lines. This follows directly from epipolar geometry of static regions. The use of epipolar patches determines the height of the candidate patches in I_s, for matching. The ambiguity regarding the candidate patch width remains, since the scale of the object in I_s is unknown (see Fig. 5(a-b).

In practice, we compute a set of epipolar lines in the reference image, and a set of patches between each pair of adjacent lines is defined, with up to 2/3 overlap between them (see Fig. 5(c)). In a similar manner, the candidate set of patches is computed in the support image between the corresponding epipolar lines but with 3 different widths. The epipolar lines are parametrized by the angle of the line where the epipole is taken to be the origin. For obtaining overlap across epipolar lines, additional epipolar lines are considered with 1/3 and 2/3 shift of the angle (see Fig. 5(d)). Thanks to the use of overlapping patches, the confidence that a pixel is static is measured a few times, and the final probability map is smoother and more robust.

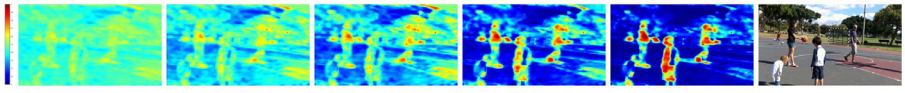
(a) (b) (c)

Fig. 4. Dynamic object shadow: (a) a reference image; (b) a support image; (c) the computed static probability map. Two low probability regions are depicted: the true location of the moving object, marked by a continuous line, and the moving object's shadow, marked by a dashed line.



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

Fig. 5. (a)-(b) The corresponding region of the statue's base (the white object to the left of scene) is located between the corresponding pair of epipolar lines. The width of the corresponding patches differs between the two images. Some additional possible matching patches of varied sizes are depicted for illustration purposes. (c) Overlapped patches defined between a pair of epipolar lines. (d) Overlapped patches across epipolar lines.



(1) (2) (3) (5) (7)

Fig. 6. The improvement of the dynamic probability map as a function of the number of support images. The number of support images is depicted beneath the probability maps; the color bar, which is common to all the probability maps, is depicted to the left.

3.1.2. Patch confidence measure:

Let $C(\mathbf{r}|I_s,\theta)$ be the confidence that the patch $\mathbf{r}\in I$ is a projection of a static scene region not occluded in I_s. This confidence is based on the similarity between $\mathbf{r}$ and its nearest neighbor among $\mathcal{R}'$, the set of candidate matching patches in I_s. Formally,

$$C(\mathbf{r}|I_s,\theta)=\max_{\mathbf{r}'\in\mathcal{R}'}\{sim_\theta(\mathbf{r},\mathbf{r}')\}, \tag{1}$$

where $sim_\theta(\mathbf{r},\mathbf{r}')$ is the similarity between the two patches, using the descriptor θ (e.g., HOG or color histogram). The confidence is normalized to the range (0, 1) for each descriptor by mapping the range of $C(\mathbf{r}|I_s,\theta)$ of all pairs of reference and support images. We denote the normalized value by $\hat{C}(\mathbf{r}|I_s,\theta)$.

Albeit simple, this measure turns out to have important and nontrivial values. Ambiguity often makes it difficult to choose the best candidate. For example, when the background is periodic or uniform, there may be more than a single patch with high correspondence confidence along the epipolar line. As we do not aim to recover the 3D structure, locating the correct match is not important for the success of the algorithm. We merely focus on the question of whether or not a good correspondence exists. Clearly, if the best match has low confidence, the pixel is unlikely to be a projection of a non-occluded static 3D point.

Extensive research exists regarding the difficulties of choosing the best descriptor of a patch and the best method of computing similarity between descriptors of two patches. As expected, we found that the optimal descriptor depends on the image set – the extent to which the image colors change, and the various textures of the captured objects. The algorithm proposed in this paper may be used with any set of descriptors and similarity measures.

3.1.3. Matching probability map:

We treat the confidence as a probability and use it to build a probability map that holds, for each pixel $\mathbf{x}$ in I, the probability that $\mathbf{x}$ is static and not occluded, $P(\mathbf{x}|I_s)$. This probability measure is based on the confidence that a good match of the pixel's region exists along the epipolar line, as described in Eq. 1.

Let $\mathcal{R}_x$ be the set of patches that contain a pixel $\mathbf{x}$ and let $\Theta=\{\theta_1\ldots\theta_m\}$ be the set of descriptors. The matching probability of a pixel, $P(\mathbf{x}|I_s)$, is calculated as the weighted expectation estimation of the set of confidences computed for each of the patches in $\mathcal{R}_x$ with each descriptor in Θ. It is given by:

$$P(\mathbf{x}|I_s)=\sum_{\theta\in\Theta,\mathbf{r}\in\mathcal{R}_x} w_r w_\theta \hat{C}(\mathbf{r}|I_s,\theta). \tag{2}$$

Here w_θ and w_r are the weights of the confidence of a descriptor θ and the location of $\mathbf{x}$ within the patch, respectively. The value w_θ is predefined by the user for each descriptor. We set w_r to be inverse proportional to the distance, $d(\mathbf{x},\mathbf{r}_c)$, of the pixel from the patch center, $\mathbf{r}_c$. In our implementation, $w_r=e^{-d(\mathbf{x},\mathbf{r}_c)^2/2\sigma^2}$ and $\sigma=\max_{x,r_c}\{d(\mathbf{x},\mathbf{r}_c)\}/3$, where the max is taken over all patches in all of the images. All weights are normalized to sum to one for each pixel; therefore $P(\mathbf{x}|I_s)$ is guaranteed to be in the range of zero to one, and we can regard it as probability.

3.2. A set of images

Combining the results obtained from multiple support images is analagous to considering the testimonies of a few witnesses who viewed the same scene from different locations. Regions that are occluded or out of view in one image are expected to be visible in other images (see Fig. 2). Similarly, if the motion coincides with the

epipolar lines in a pair of images, it is unlikely to coincide with the epipolar lines with respect to the other images. Fig. 6 illustrates the effect of using an increasing number of support images, as we next describe.

Our goal is to compute the dynamic probability, $P(\mathbf{x})$, given a set of support images $\{I_S\}$. We compute the matching probability, $P(\mathbf{x}|I_s)$, for each $I_s \in I_S$, and combine the probabilities as follows:

$$P_{static}(\mathbf{x}|I_S) = \frac{\prod_{s\in\mathcal{S}} P(\mathbf{x}|I_s)}{\prod_{s\in\mathcal{S}} P(\mathbf{x}|I_s) + \prod_{s\in\mathcal{S}} (1 - P(\mathbf{x}|I_s))}. \quad (3)$$

$P_{dynamic}(\mathbf{x})$ is the complementary probability. Note that the above aggregation of probabilities has the following characteristics: the aggregated probability of a few probabilities that are higher than 0.5 is higher than each of the input probabilities. Similarly, when all of the probability values are lower than 0.5, the combined probability measure is lower than each of the inputs. An input probability of 0.5 does not influence the combined probability – in this case the combined probability is determined by the rest of the input probabilities. Moreover, high and low probabilities balance each other out and result in a probability that lies in between them. Before combining the probabilities we add a preliminary step of remapping the probability values to the range (0.3,0.7), to avoid the overinfluence of extreme values of $P(\mathbf{x}|I_s)$ (e.g., 0 or 1). The use of multiple images, multiple overlapping patches, and descriptors per pixel, allows our method to handle false correspondences, as we demonstrate in the next section.

4. Results

We implemented the proposed algorithm in MATLAB and tested it on challenging real-world data sets. (Standard datasets for this task are not available.)

Datasets: Three images of each set are depicted in Fig. 1, Fig. 3, and Fig. 7 (the full sets can be found in the supplementary material). The sets capture both indoor and outdoor scenes, single as well as multiple moving objects, and rigid as well as non-rigid (person) objects. The rock-climbing set was captured by Park et al. (2010), the playground, basketball and skateboard sets were captured by Basha et al. (2012), and the other two were captured by us. All images were captured from different viewpoints, without calibration or a controlled setup. We used the same camera in four of the six image sets to focus on the behavior of the algorithm and not the sensitivity of the descriptors to camera change.

Implementation details: We computed the fundamental matrices of the image pairs using the BEEM algorithm (Goshen and Shimshoni, 2008), and used only pairs of images where BEEM succeeded. The sets of patches in the reference image were chosen such that each pixel was covered by nine patches – three overlapping patches along the epipolar line, and three across epipolar lines. We used the same combination of two descriptors for all experiments: a histogram of oriented gradients (HOG) descriptor, and a 2D histogram of the H and S channels of the HSV color representation. The weights of the descriptors were set to 2 and 1, respectively. The similarity of the HOG descriptors was computed using the cosine distance. The similarity of two 2D histograms, B_1 and B_2, was computed using their intersection over union measure (in our implementation we used 10 bins per channel).

4.1. Qualitative results

The dynamic probability maps are presented for each of the datasets as a heat map (blue for static and red for dynamic). We consider independently each image in each dataset as a reference image. Fig. 8 shows an example of a dynamic probability map for one reference image per set, and its thresholded map overlayed on the image. The thresholded dynamic probability maps computed independently for each of the images in the skateboard dataset, are shown in Fig. 9. Overall, we observe that our algorithm successfully assigns high probabilities to the moving regions, in most cases. Hence, it can be used to detect the dynamic regions. Observe that these regions are indeed the interesting parts of the scene and hence our method can be used to direct the attention of higher level algorithms to these regions.

In some places the algorithm struggles. This is usually because some of our underlying assumptions are not met in practice. In the skateboard set, the rider's shirt resembles the color of the right windows, and in the toy-ball set part of the ball is not detected since it resembles parts of the background. In the challenging climbing set, the man wearing the red shirt at the bottom of the images hardly moves; therefore only the edges of his silhouette are detected (Fig. 8, last row). The colors of the climber's shirt resemble the colors of some areas of the climbing wall, and the shirt detection is weak as a result. False positives occur when the descriptor fails to detect similarities. For example, the matching fails on reflective, transparent and narrow objects in the climbing set (with width less than that of a patch).

Table 1
Quantitative Results: for each data set we show the number of images in the set, the average size of the support set for each image, that is, the number of images for which the BEEM algorithm succeed in computing the fundamental matrices per each reference image, and the Jaccard measure (higher is better). We show Jaccard results with a threshold optimized per image and per set.

Image set	Set size	Average size of support sets	Jaccard opt. per image	Jaccard opt. per set
Helmet	4	2	0.53 ± 0.18	0.36 ± 0.28
Skateboard	5	4	0.44 ± 0.1	0.42 ± 0.1
Playground	7	2.6	0.37 ± 0.11	0.32 ± 0.01
Toy Ball	7	4.5	0.63 ± 0.03	0.6 ± 0.05
Basketball	8	7	0.48 ± 0.04	0.47 ± 0.04
Climbing	10	9	0.15 ± 0.05	0.13 ± 0.04

4.2. Quantitative results

We evaluate our method using the Jaccard measure (intersection over union) on manually labeled moving regions. The measure requires a binary map so we threshold the probability map to obtain one. Inspired by the evaluation methodology of the Berkeley Segmentation Data Set Martin et al., we use two thresholds – a threshold that optimizes the Jaccard measure of each image, and one that optimizes the mean Jaccard measure of all of the images in a given set. Examples of manual ground truth masks are shown on the left column of Fig. 8, and examples of masks that resulted from thresholding the dynamic probability map are shown on its right column.

The algorithm was applied to each of the images in the sets and the mean Jaccard measures per dataset are presented in Table 1. The measure is high for the toy ball, basketball and helmet sets. It is low for the challenging climbing set, as discussed earlier.

Comparison to other methods: The results of Gullapally et al. and our method on one of the images of the skateboard dataset are presented in Fig 10. The figure demonstrates that our algorithm not only detects all moving regions in the image, but also creates a smoother and more complete region of the skateboard rider, compared to Gullapally et al.. In order to conduct a fair comparison the method of Gullapally et al., which assumes only a single moving object in the scene, we generated a ground truth mask of only the main moving object.

The reported Jaccard measures by Gullapally et al. on a pair of images from the set were 0.37 and 0.28. Our results were 0.42 and

Fig. 7. Three images of each dataset. From top to bottom: Toy Ball, Playground, Skateboard, Climbing. (The helmet and basketball sets can be viewed in Fig. 1 and Fig. 3, respectively).

0.59, respectively, when only a single support image was used, and 0.45 and 0.6 when four support images were used.

We could not test the method of Gullapally et al. on other images from our datasets, since their code is not available. Instead, we demonstrate that NRDC (HaCohen et al., 2011), which is the first step of their method, does not perform well on most of our datasets. An example of the performance of the NRDC algorithm on an image of our CrowdCam images is demonstrated in Fig. 11, with additional examples in the supplementary material. Quantitative results show that for the skateboard dataset, NRDC found correspondences for 75% of the image, but for the rest of the datasets as few as 33% of the correspondences were found. Moreover, for the moving objects only about 50% of their pixels had some matching points, and in some cases this number was as low as 19%. This is the raw data that is available to the motion segmentation algorithm of Gullapally et al.. We conclude that our algorithm outperforms that of Gullapally et al. by a large margin.

We also tested the applicability of back-projecting dense SFM for detecting moving regions. To this end, we use the SFM algorithm of Wu (2013) on our data sets. However, as can be seen in Fig. 12, the SFM algorithm reconstructs only a small number of scene points, and in the other two datasets it failed completely. Hence, it is impossible to use the back-projecting dense SFM to infer the dynamic regions unless all pixels that were not reconstructed are considered dynamic regions, which is clearly not the case. Further evidence that SFM methods struggle with our data sets is the failure of the SFM-based method of Wang et al. (2015)[1].

We also applied one of the state-of-the-art co-segmentation methods (Faktor and Irani, 2013) to our data set and show the results in Fig. 11(c,d). Again, we see that this algorithm does not cope well with our data. Note that the goal of co-segmentation method is to segment objects rather than detect *moving* regions (see discussion in Section 2).

We could not compare our method to motion segmentation algorithms, because they work on video while we only use a sparse

[1] Personal communication with the authors.

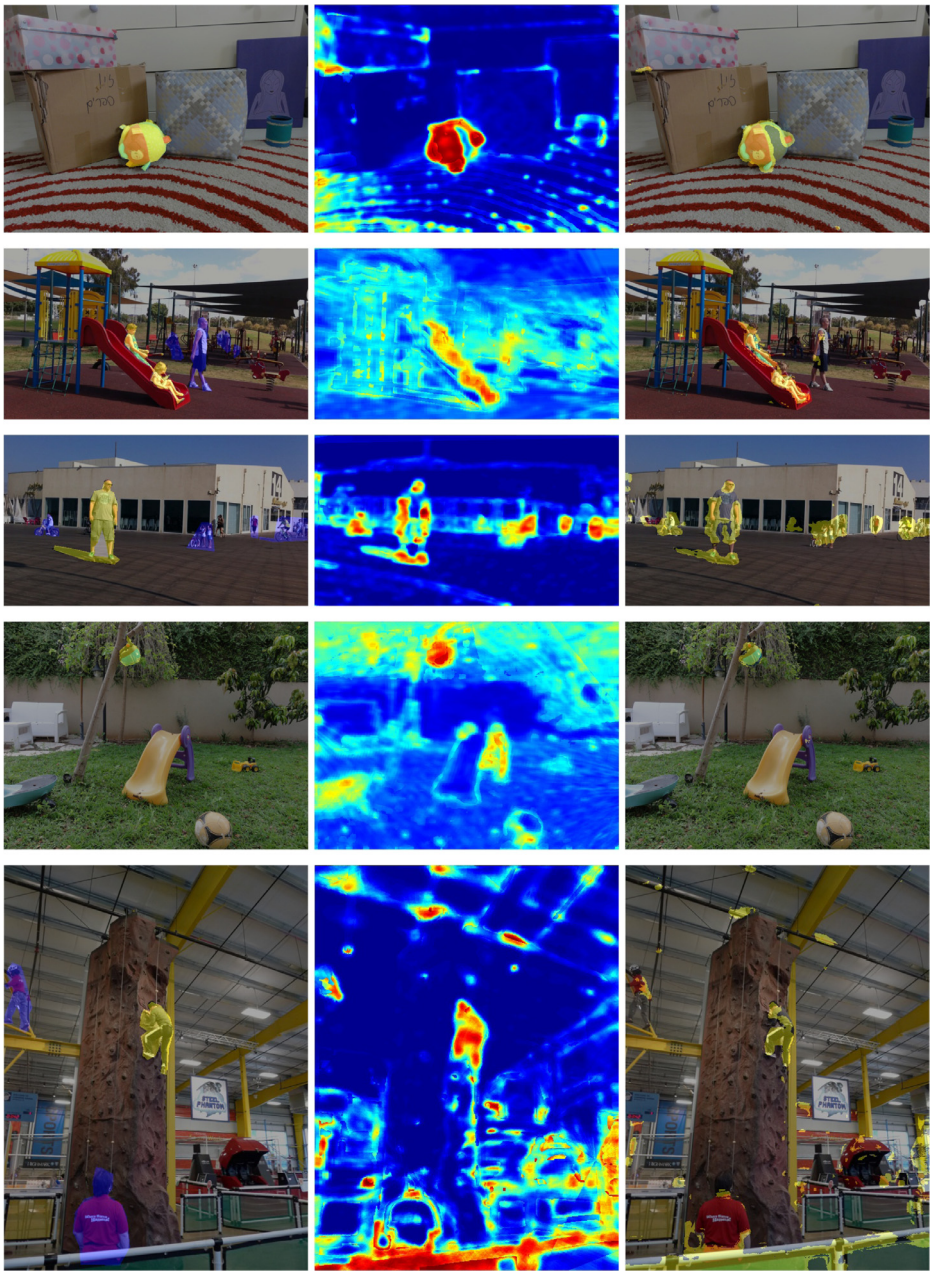


Fig. 8. From left to right: ground truth mask, dynamic probability map, and the thresholded map, for each of the datasets. The 'don't care' areas in the ground truth masks are marked in blue. Note that from the thresholded map, all dynamic regions are detected, only the last row contains many false positive detections.

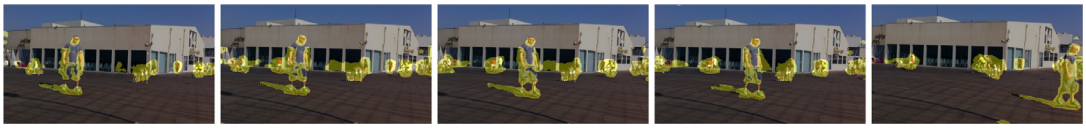


Fig. 9. The result of thresholding the dynamic probability map for each of the images in the skateboard dataset.

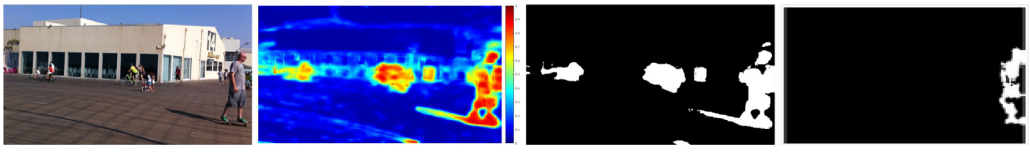
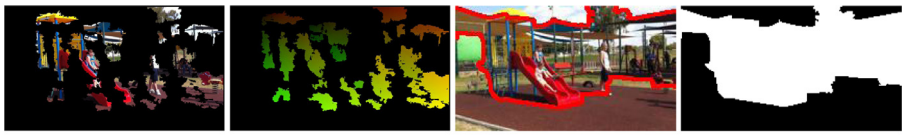


Fig. 10. An image from the skateboard set, the dynamic probability map, a mask obtained from using a simple threshold on it at 0.5, and the mask of Gullapally et al.



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

Fig. 11. Failure cases of other methods: (a) The region for which the NRDC algorithm found matching pixels on a pair of images from the playground dataset, (b) the confidence map of the matching computed by the NRDC algorithm. Black regions indicates regions for which no matching pixels were found. (c)-(d) An example of the failure of co-segmentation (Faktor and Irani, 2013) on the playground dataset.

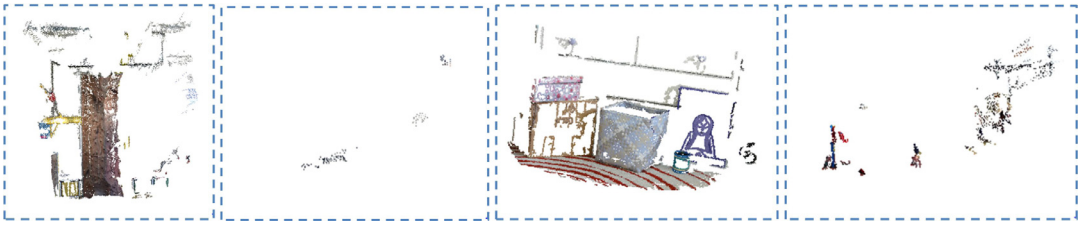


Fig. 12. Visual SFM (Wu, 2013) results on the (left to right) Climbing, basketball, toy-ball and playround sets; the algorithm failed on the skateboard and helmet sets.

set of still images. Similarly, we could not compare our method to change detection algorithms, because our images were not taken from the same viewpoint, nor can they be aligned with a homography.

5. Conclusions and future work

CrowdCam images are an emerging form of photography. Detecting moving regions is a basic step towards analyzing the dynamic content of such data. We proposed an algorithm that computes the probability of a pixel to be a projection of a dynamic 3D point. It does so by finding the probability that an epipolar patch (defined by a pair of matching epipolar lines) has matches consistent with the epipolar geometry. This renders, our algorithm less sensitive to matching errors than alternative algorithms that require precise matching. The aggregation of the results from a set of support images allows us to distinguish dynamic regions from occluded regions and objects which move along epipolar lines. We evaluated our method on a new and challenging data set (that will be made public) and report results better than the alternative.

Our method is sensitive to the quality of the descriptors used for matching patches. We propose to use additional descriptors and set their weight for each pair of images or scene. One way to do it is by considering several descriptors for computing the fundamental matrices, and set the weights according to the number of matched features for each descriptor. In addition, in future research we intend to use the results of our method for various applications including change detection, moving object segmentation, and action recognition. For example, to improve the naive thresholding for moving object segmentation, we propose to use advanced segmentation methods that integrate our maps with other image cues such as color and texture. Moreover, it is of interest to develop a moving object segmentation method that combines the computed probability maps in all images.

Acknowledgment

This work was partially supported by the Israel Science Foundation grant no. 930/12. and Israel Science Foundation grant no. 1917/2015.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at 10.1016/j.cviu.2017.04.004.

References

Alexe, B., Deselaers, T., Ferrari, V., 2012. Measuring the objectness of image windows. IEEE Trans. Pattern Anal. Mach. Intell. 34 (11), 2189–2202. doi:10.1109/TPAMI.2012.28.

Basha, T., Moses, Y., Avidan, S., 2012. Photo sequencing. In: European Conference on Computer Vision (ECCV).

Brox, T., Bruhn, A., Weickert, J., 2006. Variational motion segmentation with level sets. In: European Conference on Computer Vision (ECCV).

Campbell, N.D., Vogiatzis, G., Hernández, C., Cipolla, R., 2010. Automatic 3d object segmentation in multiple views using volumetric graph-cuts. Image Vis. Comput. 28 (1).

Cremers, D., Soatto, S., 2005. Motion competition: a variational approach to piecewise parametric motion segmentation. Int. J. Comput. Vis. (IJCV) 62 (3).

Cristani, M., Farenzena, M., Bloisi, D., Murino, V., 2010. Background subtraction for automated multisensor surveillance: a comprehensive review. EURASIP J. Adv. Signal Process. 2010, 43.

Faktor, A., Irani, M., 2013. Co-segmentation by composition. In: International Conference of Computer Vision (ICCV).

Gang, Z., Long, Q., 2004. Silhouette extraction from multiple images of an unknown background.

Goshen, L., Shimshoni, I., 2008. Balanced exploration and exploitation model search for efficient epipolar geometry estimation. IEEE Trans. Pattern Anal. Mach. Intell. (PAMI) 30 (7).

Guillemaut, J.-Y., Hilton, A., 2011. Joint multi-layer segmentation and reconstruction for free-viewpoint video applications. Int. J. Comput. Vis. (IJCV) 93 (1).

Gullapally, S. C., Malireddi, S. R., Raman, S.,. Dynamic object localization using handheld cameras, in: National Conference on Communications (NCC), IEEE.

HaCohen, Y., Shechtman, E., Goldman, D.B., Lischinski, D., 2011. Non-rigid dense correspondence with applications for image enhancement. ACM Trans. Graphics (TOG) 30 (4), 70.

Klare, B., Sarkar, S., 2009. Background subtraction in varying illuminations using an ensemble based on an enlarged feature set. In: Proc. IEEE Conf. Comp. Vision Patt. Recog.

Lu, D., Mausel, P., Brondizio, E., Moran, E., 2004. Change detection techniques. Int. J. Remote Sens. 25 (12).

Luong, Q.-T., Faugeras, O.D., 1996. The fundamental matrix: theory, algorithms, and stability analysis. Int. J. Comput. Vis. (IJCV) 17 (1), 43–75.

Martin, D., Fowlkes, C., Tal, D., Malik, J.,. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. in: International Conference of Computer Vision (ICCV).

Mukherjee, L., Singh, V., Dyer, C.R., 2009. Half-integrality based algorithms for cosegmentation of images. In: Proc. IEEE Conf. Comp. Vision Patt. Recog.

Ochs, P., Malik, J., Brox, T., 2014. Segmentation of moving objects by long term video analysis. IEEE Trans. Pattern Anal. Mach. Intell. (PAMI) 36 (6).

Park, H.S., Shiratori, T., Matthews, I., Sheikh, Y., 2010. 3d reconstruction of a moving point from a series of 2d projections. In: European Conference on Computer Vision (ECCV).

Radke, R.J., Andra, S., Al-Kofahi, O., Roysam, B., 2005. Image change detection algorithms: a systematic survey. IEEE Trans. Image Process. 14 (3).

Rother, C., Minka, T., Blake, A., Kolmogorov, V., 2006. Cosegmentation of image pairs by histogram matching-incorporating a global constraint into mrfs. In: Proc. IEEE Conf. Comp. Vision Patt. Recog.

Shi, J., Malik, J., 1998. Motion segmentation and tracking using normalized cuts. In: International Conference of Computer Vision (ICCV).

Stauffer, C., Grimson, W.E.L., 1999. Adaptive background mixture models for real–time tracking. In: Proc. IEEE Conf. Comp. Vision Patt. Recog, 2.

Sun, D., Sudderth, E.B., Black, M.J., 2012. Layered segmentation and optical flow estimation over time. In: Proc. IEEE Conf. Comp. Vision Patt. Recog.
Wang, H., Suter, D., 2006. Background subtraction based on a robust consensus method. In: Proc. Int. Conf. Patt. Recog, 1.
Wang, J.Y., Adelson, E.H., 1994. Representing moving images with layers. IEEE Trans. Image Process. 3 (5).
Wang, T.Y., Kohli, P., Mitra, N.J., 2015. Dynamic sfm: detecting scene changes from image pairs. Symp. Geom. Process. 2015.
Wu, C., 2013. Towards linear-time incremental structure from motion. In: 2013 International Conference on 3D Vision-3DV 2013. IEEE.
Yuan, C., Medioni, G., Kang, J., Cohen, I., 2007. Detecting motion regions in the presence of a strong parallax from a moving camera by multiview geometric constraints. IEEE Trans. Pattern Anal. Mach. Intell. (PAMI) 29 (9), 1627–1641.