Learning color and locality cues for moving object detection and segmentation

Feng Liu and Michael Gleicher
Department of Computer Sciences, University of Wisconsin-Madison
1210 West Dayton Street, Madison, WI, 53706
{fliu|gleicher}@cs.wisc.edu

Abstract

This paper presents an algorithm for automatically detecting and segmenting a moving object from a monocular video. Detecting and segmenting a moving object from a video with limited object motion is challenging. Since existing automatic algorithms rely on motion to detect the moving object, they cannot work well when the object motion is sparse and insufficient. In this paper, we present an unsupervised algorithm to learn object color and locality cues from the sparse motion information. We first detect key frames with reliable motion cues and then estimate moving sub-objects based on these motion cues using a Markov Random Field (MRF) framework. From these sub-objects, we learn an appearance model as a color Gaussian Mixture Model. To avoid the false classification of background pixels with similar color to the moving objects, the locations of these sub-objects are propagated to neighboring frames as locality cues. Finally, robust moving object segmentation is achieved by combining these learned color and locality cues with motion cues in a MRF framework. Experiments on videos with a variety of object and camera motion demonstrate the effectiveness of this algorithm.

1. Introduction

Automatically detecting and segmenting a moving object from a monocular video is useful in many applications like video editing, video summarization, video coding, visual surveillance, human computer interaction, etc. Many methods have been presented (c.f. [21, 9, 3, 24, 23]). Many of them aim at a robust algorithm for extracting a moving object from a video with rich object and camera motion. However, extracting a moving object from a video with less object and camera motion is also challenging. Most previous automatic methods rely on object and/or camera motion to detect the moving object. Small motion of the object and/or camera do not provide sufficient information for these methods.

For example, most existing methods use motion to detect moving objects. They assume if a compact region moves differently from the global background motion, it mostly likely belongs to a moving object. Motion-based methods [8, 12, 21, 9, 3] usually take the detected moving pixels as seeds, and cluster pixels into layers with consistent motions (and consistent color and depth). When motion information is sparse and incomplete, they cannot work robustly. For example, Figure 1 shows an example where a boy sits on the floor and moves only in a few frames. And even in these frames, he only moves a part of his body. Methods using object motion information can only detect an incomplete part of the object. For example, if we segment the object in a popular Markov Random Field (MRF) framework, as described in § 2.3, only the moving part of the boy's body is detected in frames where the part moves, and no meaningful region is found in other frames as shown in Figure 1 (b) and (c). This example shows that using object motion alone to infer moving objects is insufficient. Similarly, in this example, since the camera barely moves, it is also difficult for a structure from motion (SFM) algorithm as used in methods like [24] to obtain useful depth information to infer the moving object.

Impressive results have been reported recently for bilayer video segmentation in the scenario of video chatting [4, 23]. These algorithms can robustly segment a major foreground object from a video with dynamic background, however, they are not suitable for videos with complex camera motions.

Instead of building a moving object model, some other methods build a background model to detect and segment a moving object (c.f. [5, 10, 17, 15, 18, 22]). These methods work well for videos with static cameras. When videos have complex camera motions, the background model is hard to build.

This paper presents a solution that learns a moving object model by collecting the sparse and insufficient motion information throughout the video. Specifically, we presented an unsupervised algorithm to learn the color and locality cues of the moving object. We first detect key frames that contain motion cues that can reliably indicate at least some

978-1-4244-3991-1/09/$25.00 ©2009 IEEE

part of the moving object (§ 2.1 and 2.2). From these key frames, we estimate moving sub-objects based on motion cues using a MRF model (§ 2.3). We then learn from these sub-objects a moving object color model characterized by a Gaussian Mixture Model (GMM). To avoid false detection of background pixels with similar color to the moving object, we propagate the location information of sub-objects to non-keyframes as locality cues(§ 2.3). Finally, we extract the moving object by combining these learned color and locality cues with motion cues in a MRF framework(§ 3). Our experiments demonstrate that learning cues about the moving object can significantly help achieve robust moving object segmentation from videos with a variety of object and camera motion (§ 4).

2. Learning moving object cues

We consider moving objects in a video as some compact regions with different apparent motion from the background. Specifically, if a region is moving in a certain frames, we consider it a moving object throughout the video. This is reasonable in practice. For example in a video, a boy walks for a while and stops to swing his hands. It is meaningful to treat his whole body as a moving object instead of his hands only in the late frames.

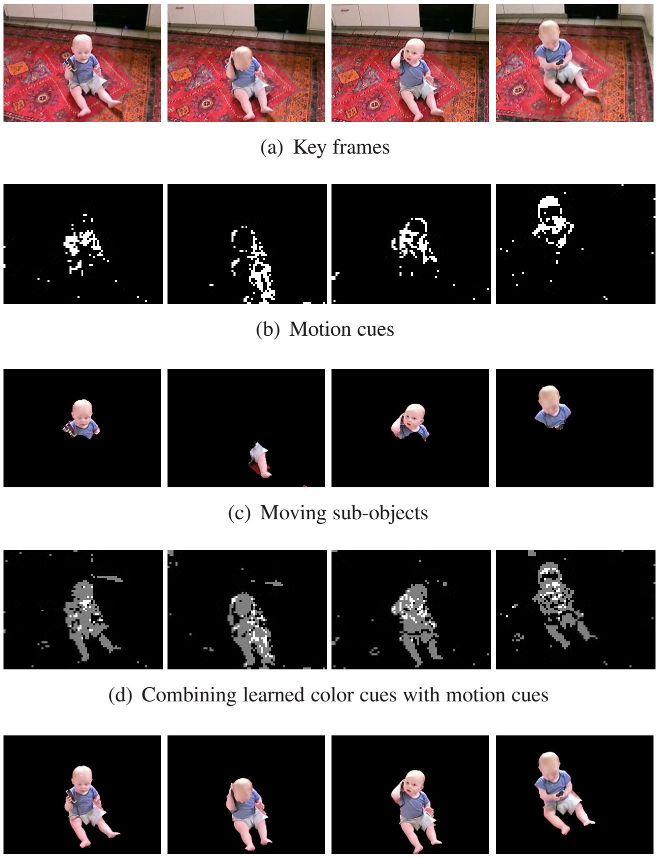
Based on the moving object definition, we consider motion an important cue to identify it. If a pixel/region has significant different apparent motion from the background, it mostly likely belongs to a moving object. Accordingly, motion cues are defined as the difference between the pixel motion and the background motion. We extract key frames with strong, compact motion cues. From these key frames, we segment moving sub-objects using motion cues within a MRF framework. Finally we learn the color and locality cues of moving objects from these moving sub-objects.

2.1. Motion cues

We assume that the background is dominant in the scene. Based on this assumption, motion cues are defined as the discrepancy between the local motion and the global background motion. Estimating the global motion in a video has a rich literature of possible solutions [19]. Because the global motion between consecutive frames is small, we model it using a homography [19]. We use a SIFT [14] feature-based method to estimate the homography since it is robust for processing low-quality videos. Specifically, we extract SIFT features from each frame, establish feature correspondence between neighboring frames, and estimate the homography using the RANSAC [6] algorithm.

With the homography, we calculate the motion cue (mc) at pixel (x, y) as the discrepancy between the optical flow $m_o(x, y)$ and the global motion $m_g(x, y)$as follows:

$$mc(x, y) = \|m_o(x, y) - m_g(x, y)\|_2^2 \qquad (1)$$


(a) Key frames

(b) Motion cues

(c) Moving sub-objects

(d) Combining learned color cues with motion cues

(e) Segmentation results using all cues

Figure 1. A working example of our algorithm. Step 1: Extract key frames (a) with reliable motion cues (b); Step 2: Estimate moving sub-objects (c) in key frames using motion cues; Step 3: Learn color and locality cues from these moving sub-objects. (d) shows the likelihood of each pixel belonging to the moving object based on the color and motion cues. Step 4: Segment the moving object using color, locality and motion cues. The final results are shown in (e).

Directly estimating optical flow suffers from noise, aperture problem, etc. Since we get a robust estimation of the homography, we use the homography as an initial guess to guide a block matching search for optical flow. An example of motion cues is illustrated in Figure 1(b).

2.2. Key frame extraction

We define a key frame where a moving object or its part can be reliably inferred from motion cues. Motion cues are likely reliable when they are strong and compact. Accordingly, we define the following two criteria to extract key frames:

$$\sum (mc(x, y) \geq \delta) > minArea$$
$$\text{Var}((x, y) | mc(x, y) \geq \delta) < maxSpan$$

where $mc(x, y)$ measures the motion cue value at pixel (x, y), and δ is a parameter. The first criterion requires a key frame to have at least $minArea$ pixels with significantly different motion from the global motion. The left part of the second criterion measures the spatial distribution of the potential object pixels by calculating the variance of their positions. By this, the second criterion requires a key frame to have compact motion cues. Some examples of key frames are shown in Figure 1(a).

2.3. Segment moving sub-objects from key frames

Using motion cues alone to identify the moving object is limited. First, not all pixels of the moving object have significant motion cues. As shown in Figure 1(b), motion cues are sparse. Often only the boundary of the moving region can be detected, which is known as the aperture problem. Second, the moving object does not move throughout the whole video, and even when it moves, only parts of it have apparent motion. In this step, we aim to extract some parts of the moving object that move in the current frame, thus ignoring the second issue for the moment. We call these parts moving sub-objects. The first issue can be addressed by considering the interaction between the labels of the neighboring pixels:

1. Neighboring pixels are likely to have the same label.

2. Neighboring pixels with similar colors are more likely to have the same label.

As suggested by previous work (c.f. [9, 17, 15]), we use a MRF prior on labels to model these interactions as follows:

$$p(l_i|i = 1, ..., M) \propto \prod_{i \in \{1,...,M\}} \prod_{j \in N_i} \psi(l_i, l_j) \tag{2}$$

$$\psi(l_i, l_j) \equiv exp(\lambda l_i l_j / (\alpha + d(i, j))) \tag{3}$$

where l_i is the label of pixel i, $l_i = -1$ for the moving object and $l_i = 1$ for the background, and M is the number of pixels in the image. N_i is the 8-connection neighborhood of pixel i. $d(i, j)$ measures the color difference between pixel i and j. α and λ are parameters.

Since for the key frames, motion cues are reliable to predict the moving parts of the object, the likelihood of the image I given a labeling can be modeled as follows:

$$p(I|\{l_i|i = 1, ..., M\}) = \prod_{i \in \{1,...,M\}} p(f_i|l_i) \tag{4}$$

$$p(f_i|l_i) = p_m(mc_i|l_i) \equiv \exp(l_i(mc_i - \delta_m)) \tag{5}$$

where f_i is the feature of pixel i, specifically mc_i here, the motion cue value at pixel i. δ_m is a parameter.

With the above MRF prior and likelihood model, moving object segmentation for a key frame I can be achieved by finding the labeling that maximizes the following posterior:

$$P(L|I) = \frac{1}{Z} \prod_{i \in I} p(f_i|l_i) \prod_{i \in I} \prod_{j \in N_i} \psi(l_i, l_j) \tag{6}$$

The above optimization problem can be solved by graph cut algorithms [2, 13] or a loopy belief propagation algorithm [11]. Comparisons between these algorithms are detailed in [20, 15]. We use a graph cut algorithm to solve the labeling problem in Equation 6 because for the binary MRF in this paper, the graph-cut algorithm can quickly find the global optimum [20]. Some examples of extracted moving sub-objects are shown in Figure 1(c).

2.4. Learning color and locality cues

We assume that the moving sub-objects from all the key frames form a complete sampling of the moving objects in the whole video. By a complete sampling, we mean that each part of the moving objects appears in at least one of the key frames. Based on this assumption, the color distribution of moving objects can be characterized by a Gaussian Mixture Model (GMM). A GMM model can be parameterized by $g_i(p_i, \mu_i, \Sigma_i)$, $i = 1, ..., n$, where n is the number of Gaussian components, μ_i, Σ_i and p_i are the mean color vector, the covariance matrix, and the prior of component g_i respectively. We train a GMM, G^f, using the moving sub-objects extracted from key frames. We use Lab, a perceptually uniform color space [7]. Learning is achieved by using an expectation-maximization algorithm together with an agglomerative clustering strategy to estimate the number of components [1]. The estimation is based on the Rissenen order identification criterion known as minimum description length [16]. The affinity of a pixel with color c to the moving object G^f can be estimated as

$$\text{affi}_c(c) = \max_{g_j \in G^f} g_j(c) \tag{7}$$

$$g_j(c) = \frac{p_j}{\sqrt{|\Sigma_j|}} \exp\left(-\frac{1}{2}(c - \mu_j)^T \Sigma_j^{-1} (c - \mu_j)\right) \tag{8}$$

The likelihood model is built based on this affinity as follows

$$p_c(c_i|l_i) \equiv \exp(l_i(\log(\text{affi}_c(c_i)) - \delta_c)) \tag{9}$$

where δ_c is a parameter.

Since some of the detected moving sub-objects may contain some background pixels, the moving object model G^f will contain some false components. For example, part of the floor is detected as the moving sub-object in the 2nd frame of Figure 1(c). These false components can be detected by checking the affinity between each component and the possible background pixels. If a component g is too close to the background, it is likely to be an outlier, and is removed from G^f. We measure the affinity between g

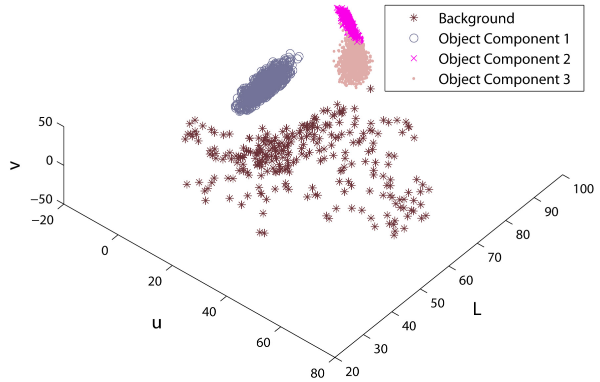


Figure 2. Color GMM model for the moving object. This figure shows 4 Gaussian components. The first one is eliminated for being too close to the background. The *Object 2* component actually has skin color. It is displayed in purple for readability.

and the background by examining its distance to the background pixels adjacent to the moving sub-object boundaries. Specifically, if many pixels around the sub-object boundaries are close to g, g is likely a false component. Following this idea, we calculate the affinity as follows:

$$\text{affi}_b(g) = \frac{\sum_{p_i \in B} [1_{p_i \in g} \sum_{p_j \in N_i, p_j \notin F} 1_{g(p_j) > \gamma} g(p_j)]}{\sum_{p_i \in B} 1_{p_i \in g}} \quad (10)$$

where F is the set of all moving sub-objects, and B is the set of these object boundaries. 1_{expr} is an indicator function that is 1 when *expr* is true and 0 otherwise. The denominator is the total number of pixels in B belonging to component g. N_i is the 8-connected neighborhood of p_i. $g(p_j)$ is the similarity between p_j and g as defined in Equation 8, and γ is a parameter. The numerator is the sum of the similarity between g and all the background pixels adjacent to B.

The color GMM model learned from the video example in Figure 1 is illustrated in Figure 2. Combining color cues with motion cues can well predict the moving object as shown in Figure 1 (d).

Color cues alone are insufficient when the background has similar color to the moving object. We resort to the location of the moving sub-objects in the key frames to provide locality cues to resolve the color ambiguity. Since moving objects can have arbitrary shapes, we use a non-parametric model [12] based on actual data to calculate the spatial affinity of a pixel to the moving object as follows:

$$\text{affi}_s^t(x_i) = \frac{1}{2\pi\sigma^2} \max_{j \in F} \exp(-\frac{(x_i - x_j)^T (x_i - x_j)}{2\sigma^2}) \quad (11)$$

where $\text{affi}_s^t(x_i)$ is the spatial affinity of a pixel i to the moving sub-objects F in key frame t, x_i is position of pixel i, and σ is a parameter. Similar to the color cues, the likelihood model is built based on this affinity as follows

$$p_s^t(x_i|l_i) \equiv \exp(l_i(\log(\text{affi}_s^t(x_i)) - \delta_s) \quad (12)$$

where δ_s is a parameter.

Algorithm 1 Propagate locality cues from key frames and segment the moving objects.

```
For each key frame $k$,
a. Propagate the locality cue of frame $k$, $p_s^k(x_i|l_i)$, forward to frame $k+1$, $k+2$,..., as follows:
  1: Initialize the locality cue at frame $t+1$ using the one at frame $t$ as follows: $p_s^{t+1}(x_i|l_i) = p_s^t(x_i|l_i)$
  2: Estimate the moving objects at frame $t+1$ using a MRF model with likelihood $p^{t+1}(f_i|l_i)$ defined in Equation 13.
  3: Refine the locality cue of frame $t+1$ using the estimated objects.
  4: If frame $t+1$ is a key frame, and the objects estimated in step 2 covers the original moving sub-objects, remove frame $t+1$ from the key frame list.
b. Propagate the locality cue of frame $k$ backward to frame $k-1$, $k-2$,..., in the same way as Step a.
```

3. Moving object segmentation

Given the learned color and locality cues of the moving objects, we could extend the MRF model in Section 2.3 by adding the color and locality cues into the likelihood term in Equation 4 to estimate the full moving objects. A further step could be to extend the MRF model from one key frame to the whole video to achieve temporal consistency. However, since motion cues are sparse and incomplete, and the locality cues are available only for key frames, the color cues will be dominant. This can cause false moving object detection when the background has regions with similar color. We solve this problem by propagating the locality cues from each key frame to others. We develop the following method to propagate the locality cues and segment the moving objects simultaneously.

The first step is to re-estimate the moving objects in the key frames. We extend the MRF model in Section 2.3 by adding the color and locality cues into the likelihood term in Equation 4 as follows:

$$\log p^t(f_i|l_i) = \log p_m^t(mc_i|l_i) + \lambda_c \log p_c^t(c_i|l_i) + \lambda_s \log p_s^t(x_i|l_i) \quad (13)$$

We solve this new MRF model for the refined moving sub-objects in key frames.

The next step is to propagate the locality cues from each key frame to the whole video and segment the objects simultaneously. The algorithm is outlined in Algorithm 1. Finally, we re-estimate the moving object in each frame using the MRF model with the likelihood model defined in Equation 13.

4. Experiments

We implemented our algorithm and tested it on home video clips taken by non-professional users with hand-held consumer digital cameras. The frame size is 320×240 and the frame rate is 30 fps. The current prototype system can process about 40 frames per minute on a 2.2GHz Athlon machine. About 80% of the time is spent on the global motion and optical flow estimation. The testing video clips have a variety of object and camera motions. (**Please examine the background change in the following image sequences to appreciate the camera motion.**) In the following, we discuss the experiments on representative video clips and compare our algorithm to one of the state of the art [24] (HMOE). The HMOE method can robustly extract a moving object from a video taken by a moving hand-held camera based on dense motion and depth estimation.

4.1. Insignificant camera and object motion

Figure 3 shows a video clip where a boy was sitting on the floor playing with a cell-phone. His body barely moved. The hand-held camera shook frequently. Only a few frames have useful motion cues and even in those frames, the motion cues alone are barely enough to segment the boy as shown in Figure 1 and Figure 3(b). As we can see, using these limited motion cues alone is not enough to segment the moving object as shown in Figure 3 (b). Learning color and locality cues from the partial object regions in these few frames helps predict the object well as shown in Figure 3(c). With these cues, the presented algorithm successfully segmented the boy out as shown in Figure 3(d). In this example, since the camera barely moves, it is difficult for a structure from motion (SFM) algorithm to work well and is unsuitable for methods that rely on depth information. For example, HMOE [24] cannot create a meaningful result from this video.

4.2. Significant camera motion and uneven object motion

Figure 4 (a) shows a video clip where a boy played a piano for a while, and walked away. The hand-held camera underwent a rough pan motion plus a zoom-in motion in order to keep the boy in the scene. The motion cues are significant when the boy walked; however they are insufficient when the boy stopped near the piano as shown in Figure 4 (b). Using motion cues alone is insufficient to extract the moving object as shown in Figure 4 (c). The presented algorithm learns the color and locality cues from the key frames, where the boy walks, and uses them to identify the boy. As shown in Figure 4 (d) and (e), the combination of color and locality cues predicts the boy well. The HMOE method fails to create a meaningful result since it is sensitive to the accuracy of its structure from motion result while the SFM algorithm does not work well enough with this video[1].

4.3. Significant camera and object motion

Figure 5 (a) shows a video clip where a boy was running on a road. The hand-held camera moved backwards and rotated to keep the boy in the frame. The segmentation result is shown in Figure 5 (c). (Note, in this example, since the cast shadow moves continuously with the boy, it was considered as part of the moving object. Without semantic understanding, it is difficult to distinguish between the real object and its cast shadow. Our algorithm consistently segments it as the moving object, which shows its robustness.) In this example, the boy ran continuously so the motion cue our algorithm extracted (Figure 5(b)) predicts the moving object well. However, motion alone cannot reliably segment the cast shadow. Combining the motion cue with the learned color and locality cues as shown in Figure 5 (d) and (f), the moving object is predicted well. Although we only use the learned cues in a standard MRF framework for segmentation, we can see that our result is at least comparable to the recent method HMOE [24].

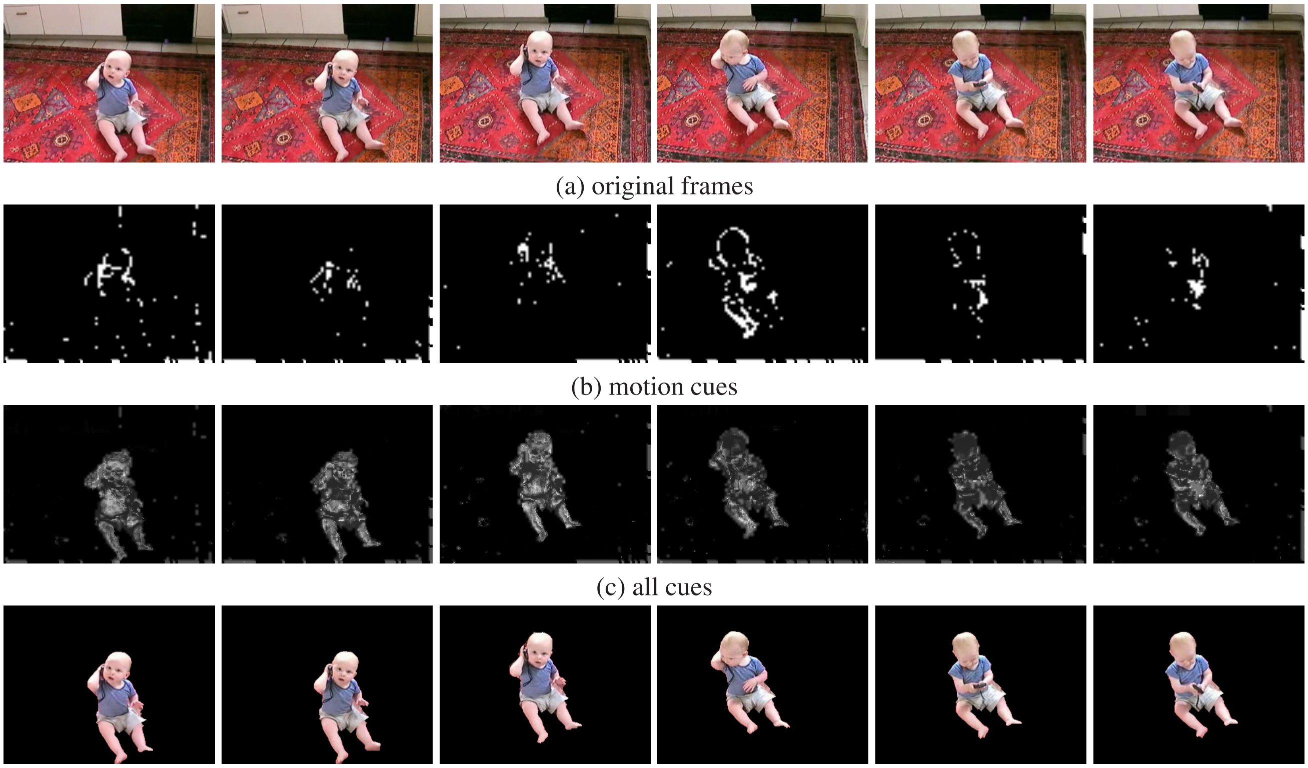
Figure 6(a) shows a video clip where a woman was playing with a boy. The father tried his best to keep them in the scene by zooming and moving the camera. Since the woman moved quickly, there are motion blur artifacts in the video. Also, while her quick motion induced a large amount of motion cues, some of them are not reliable. As indicated in Figure 6(b) and the first frame of (a), some background region adjacent to the moving object is identified as likely to belong to the moving object due to the ambiguity. The learned color cue helps to resolve the ambiguity as shown in Figure 6(d) and (f). Again, we can see that our result is comparable to the recent method HMOE.

5. Discussion

This paper demonstrated that automatically extracting a moving object from a video with less object and camera motion is not necessarily easier than a video with more object and camera motion. Because automatic moving object extraction relies on motion to detect the moving object. Small object motions make the motion contrast between the object and background sparse, ambiguous and insufficient. Small camera motions make depth estimation difficult.

The solution presented in this paper learns a moving object model by collecting the sparse and insufficient motion

[1] The image stabilization was on when this video was taken. Also, the camera focal length was changing for a while to zoom-into the boy, and in other frames, the camera was panned instead of translated. Moreover, the image quality is not good. All these can prevent SFM from obtaining accurate results.



(a) original frames

(b) motion cues

(c) all cues

(d) our segmentation results

Figure 3. Video of a boy sitting on the floor.

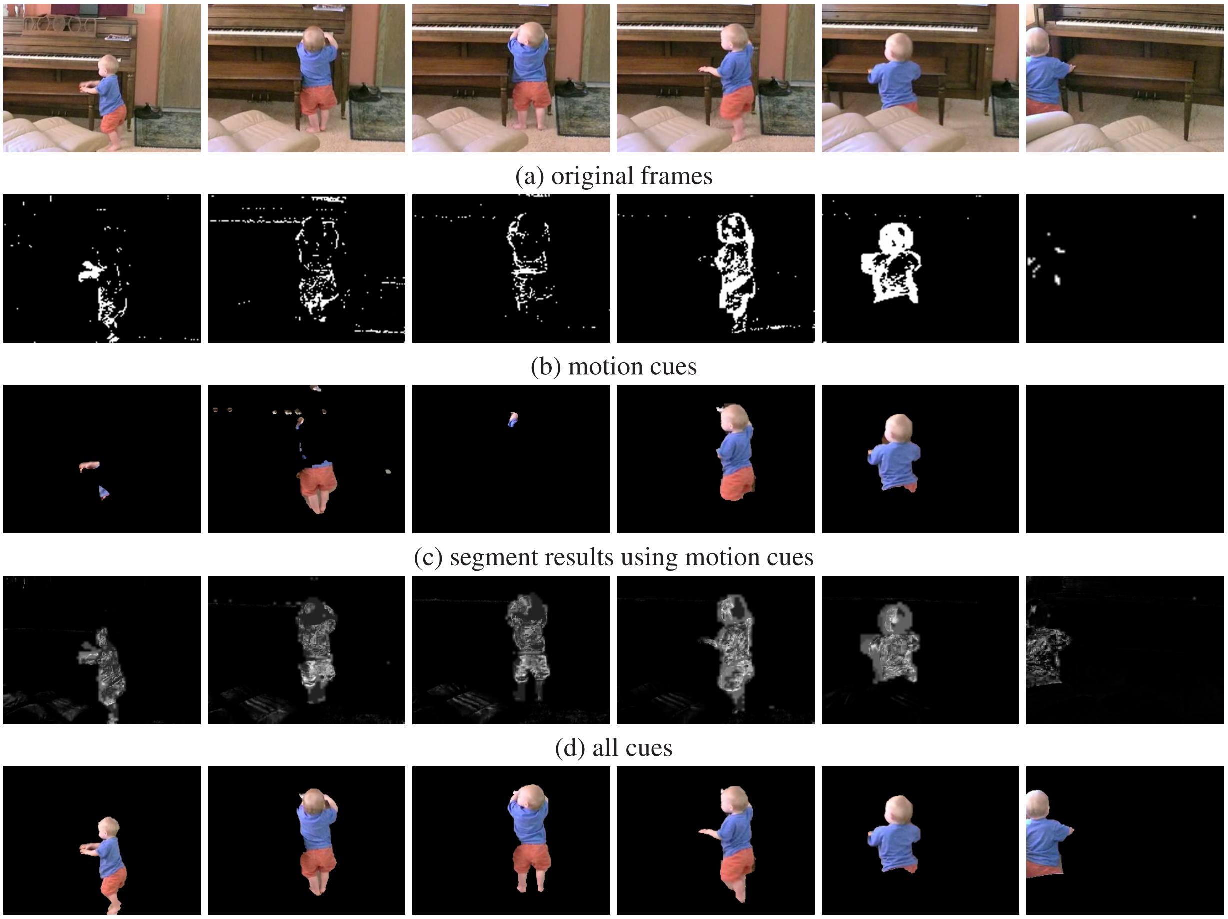
information throughout the video. Specifically, it is an unsupervised algorithm to learn the color and locality cues of the moving object. Using these cues together with motion cues, robust object segmentation is achieved. Our experiments on home videos with different amounts of non-rigid object motions and a variety of camera motions show the initial success of our algorithm.

Currently, our algorithm works off-line because object color and locality cues are learned from the whole video. However, our idea can be extended to online applications. For instance, an object model can be built gradually by incremental learning. Our algorithm provides a robust solution for dealing with camera motion by explicitly estimating the global motion and optical flow. However, explicit motion estimation slows our algorithm down. In fact, 80% of computation is spent on motion estimation. An efficient motion estimation algorithm can speedup our algorithm significantly.

Our algorithm only learns the cues of the moving object. Although our experiments prove its initial success, performance may be improved by learning the background model as well. However, it is difficult to build the background model using only the partial object regions because the regions outside these partial object regions are not necessarily the background. Methods from video surveillance have provided rich solutions for learning background models for videos acquired by static cameras. Since we aim to segment moving object from regular videos with possible complex camera motion, it is much harder.

Our algorithm learns the moving object cues from key frames, which are extracted based on motion cues. When some parts of the object never move, the object cues may not capture their characteristics. In practice, our algorithm can identify most of these parts as moving objects because it incorporates the interaction between neighboring pixels with MRF Prior. But it is still possible our algorithm will miss some of these parts if they are far from other identified moving regions both in color and space. This is a fundamental problem of automatic object segmentation methods (including ours) that only rely on the video itself to infer the object of interest. Incorporating external information can help solving this problem.

We currently set the parameters in our system empirically and the default parameters are robust for most of the videos we experimented on. However, occasionally we need to tune the parameters in Equation 13 to achieve satisfactory results: the weighting parameters for different cues need to be adjusted. This paper focuses on extracting as many information about the moving objects as possible. To examine the effectiveness of the extracted cues, we currently adopt a standard MRF framework to integrate all the cues to extract the moving objects. Although this stan-


(a) original frames

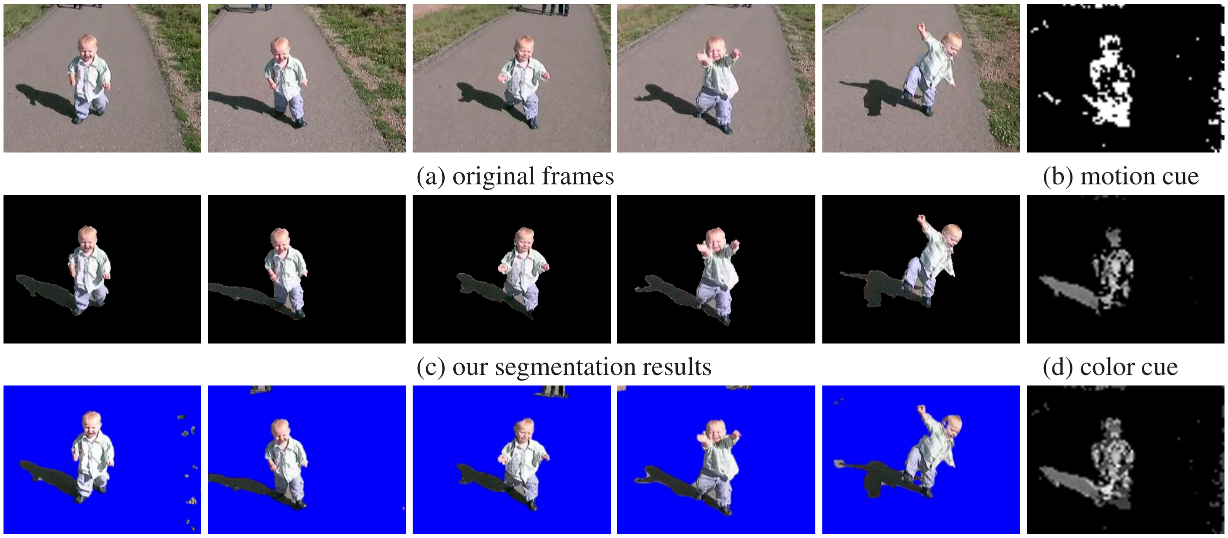
(b) motion cues

(c) segment results using motion cues

(d) all cues

(e) our segmentation results

Figure 4. Video of a boy playing the piano and walking away.


(a) original frames

(b) motion cue

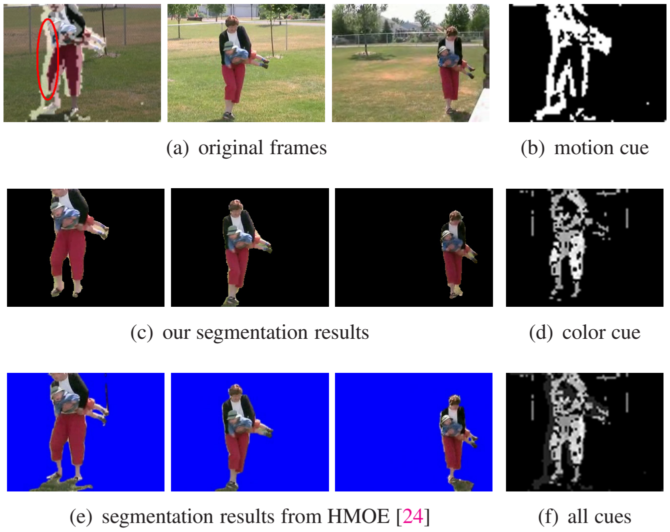
(c) our segmentation results

(d) color cue

(e) segmentation results from HMOE [24]

(f) all cues

Figure 5. Video of a boy running. The hand-held camera moves backwards and rotates to keep the boy in the scene. (b) is motion cue of the first frame, (d) is the color cue, and (f) is the combination of motion, color and locality cues.


(a) original frames (b) motion cue

(c) our segmentation results (d) color cue

(e) segmentation results from HMOE [24] (f) all cues

Figure 6. Video of a woman playing with a boy. The hand-held camera zooms and moves to keep the lady in the scene. (b), (d) and (e) are cues of the first frame. By superimposing the motion cue map of (b) upon the first frame of (a), we can see that motion cues mis-classify the neighboring background pixels of the moving object (indicated by the red ellipse). Color cue (d) can help resolve the ambiguity.

dard framework proves the initial success of our algorithm to extract moving object cues, our experiments reveal that an intelligent mechanism to fuse them into the framework is important.

Acknowledgements We thank Dr. Guofeng Zhang for helping with our experiments running HMOE and Prof. Charles Dyer for helping us improve the presentation of this paper. This research was sponsored in part by NSF grant IIS-0416284.

References

[1] C. A. Bouman. Cluster: An unsupervised algorithm for modeling Gaussian mixtures. Available from http://www.ece.purdue.edu/˜bouman, April 1997. 3

[2] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(11):1222 – 1239, November 2001. 3

[3] A. Bugeau and P. Perez. Detection and segmentation of moving objects in highly dynamic scenes. In *IEEE CVPR*, 2007. 1

[4] A. Criminisi, G. Cross, A. Blake, and V. Kolmogorov. Bilayer segmentation of live video. In *Proc. of IEEE CVPR*, pages 53–60, Jun. 2006. 1

[5] A. Elgammal, R. Duraiswami, D. Harwood, and L. Davis. Background and foreground modeling using nonparametric kernel density estimation for visual surveillance. *Proc. of the IEEE*, 90(7):1151–1163, Jul. 2002. 1

[6] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981. 2

[7] J. Foley, V. D. A., S. Feiner, and J. Hughes. *Computer Graphics: Principles and Practice, 2nd edition*. Addison Wesley, 1997. 3

[8] M. Gelgon and P. Bouthemy. A region-level graph labeling approach to motion-based segmentation. In *Proc. of IEEE CVPR*, pages 514–519, Jun. 1997. 1

[9] M. Han, W. Xu, and Y. Gong. Video object segmentation by motion-based sequential feature clustering. In *ACM Multimedia*, pages 773–782, Oct. 2006. 1, 3

[10] M. Heikkila and M. Pietikainen. A texture-based method for modeling the background and detecting moving objects. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(4):657–662, April 2006. 1

[11] Y. W. Jonathan S. Yedidia, William T. Freeman. Generalized belief propagation. In *NIPS*, pages 689–695, Nov. 2000. 3

[12] S. Khan and M. Shah. Object based segmentation of video using color, motion and spatial information. In *Proc. of IEEE CVPR*, pages 746–751, Dec. 2001. 1, 4

[13] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(2):147–159, Feb. 2004. 3

[14] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004. 2

[15] S. Mahamud. Comparing belief propagation and graph cuts for novelty detection. In *IEEE CVPR*, pages 1154–1159, 2006. 1, 3

[16] J. Rissanen. A universal prior for integers and estimation by minimum description length. *Annals of Statistics*, 11(2):417–431, 1983. 3

[17] Y. Sheikh and M. Shah. Bayesian object detection in dynamic scenes. In *IEEE CVPR*, pages 74–79, 2005. 1, 3

[18] J. Sun, W. Zhang, X. Tang, and H.-Y. Shum. Background cut. In *Proc. of ECCV*, pages 628–641, May 2006. 1

[19] R. Szeliski. Image alignment and stitching: a tutorial. *Found. Trends. Comput. Graph. Vis.*, 2(1):1–104, 2006. 2

[20] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother. A comparative study of energy minimization methods for markov random fields. In *Proc. of ECCV*, pages 16–29, May 2006. 3

[21] Y. Wang and Q. Ji. A dynamic conditional random field model for object segmentation in image sequences. In *Proc. of IEEE CVPR*, pages 264–270, Jun. 2005. 1

[22] X. Xu and T. Huang. A loopy belief propagation approach for robust background estimation. In *IEEE CVPR*, June 2008. 1

[23] P. Yin, A. Criminisi, J. Winn, and I. Essa. Tree-based classifiers for bilayer video segmentation. In *IEEE CVPR*, 2007. 1

[24] G. Zhang, J. Jia, W. Xiong, T.-T. Wong, P.-A. Heng, and H. Bao. Moving object extraction with a hand-held camera. In *IEEE ICCV*, 2007. 1, 5, 7, 8