SURF Tracking

Wei He[1], Takayoshi Yamashita[2], Hongtao Lu[1], and Shihong Lao[2]
[1]Department of Computer Science and Engineering, Shanghai Jiao Tong University, China
[2] OMRON Corporatoin, Japan
hewei1986@gmail.com, takayosi@omm.ncl.omron.co.jp, lu-ht@cs.sjtu.edu.cn

Abstract

Most motion-based tracking algorithms assume that objects undergo rigid motion, which is most likely disobeyed in real world. In this paper, we present a novel motion-based tracking framework which makes no such assumptions. Object is represented by a set of local invariant features, whose motions are observed by a feature correspondence process. A generative model is proposed to depict the relationship between local feature motions and object global motion, whose parameters are learned efficiently by an on-line EM algorithm. And the object global motion is estimated in term of maximum likelihood of observations. Then an updating mechanism is employed to adapt object representation. Experiments show that our framework is flexible and robust in dealing with appearance changes, background clutter, illumination changes and occlusion.

1. Introduction

Most motion-based tracking algorithms assume that objects undergo rigid motion. They constrain object pixels to keep constant relative positions in object movement, which is most likely disobeyed in real world.

Local invariant features achieve great success in pattern recognition problems due to their appealing characteristics. A detailed study about the performance of variety of local features is given in [14]. SIFT [13] performs best among different types of local features such as GLOH [14], shape context [3], PCA-SIFT [11], etc. Recently developed SURF feature [2] is a variant of SIFT and shares equal repeatability, distinctiveness and robustness, but has much faster computing speed.

In this paper, we present a novel motion-based tracking framework which makes no assumption of rigid motion. Object is represented by a set of SURF features of interest. Such a localized representation allows different parts of object have different motions, therefore it is more flexible to deal with object deformation and appearance changes. Feature motions are observed exactly by a feature correspondence process. Object structure information is embedded into this process to guarantee sound observation results.

A generative model is proposed to depict the relationship between local feature motions and object global motion. The model has two components: consistent component and random-walk component. A feature's motion belonging to consistent component implies that it is compatible with object global motion, otherwise they are irrelevant. Parameters of the model are learned efficiently with an online version of EM algorithm.

Object global motion is estimated in term of maximum likelihood of feature motion observations, then an updating mechanism is employed to adapt object representation. While features with incompatible motion to object global motion are discarded, newly appeared features are incorporated into the representation to learn the appearance changes.

The rest of the paper is organized as follows. Section 2 analyzes related work. Section 3 describes the generative model of feature motion. Section 4 gives the tracking algorithm. Experiment results are shown in section 5. Section 6 draws the conclusion on this paper.

2. Related work

Many algorithms have been proposed to estimate object affine motion. Shi and Tomasi [15] extends Newton-Raphson style search methods to work under affine image transformations. They monitor the quality of image features during tracking by using a measure of feature dissimilarity that quantifies the change of appearance of a feature between the first and the current frame. $\mathcal{WLS}$ tracker proposed by Allan et al. [10] combines an adaptive appearance model with motion estimation. They model each pixel's wavelet response, and search object affine motion parameters to satisfy a rigid correspondence between previous pixels and current ones. Zhang et al. [20] uses a joint position-color representation, and designs a kernel-based similarity measure to describ the relationship between image regions with respect to affine transformation parameters. Similarly, Yu and Wu [19] build a spatial-appearance model(SAM) to

2009 IEEE 12th International Conference on Computer Vision (ICCV)
978-1-4244-4419-9/09/$25.00 ©2009 IEEE

represent object and define a maximum likelihood matching criterion to obtain object motion parameters. Bearing some similarities with these algorithms, Our method differs from them in several aspects. We represent object by a set of local features, and allow flexible feature motions in feature correspondence process. Also, we judge the quality of feature by its compatibility of motion with object global motion. Bad features with incompatible movement would be discarded from the representation, While good features with consistent movement would be highly *weighted* in the next frames. Thus, our algorithm can make robust tracking in dealing with appearance changes and background clutters.

Several papers also employ local invariant feature in their tracking algorithms. Tang and Tao [16] present a SIFT-based attributed relational graph(ARG) for object represent. Features which persistently appear in several consecutive frames are considered as stable ones and used to construct the graph, while features which never matched in several consecutive frames are considered as inactive ones and are deleted from the graph. However, such stable features are always rare in real cases due to object appearance deformation or illumination changes, which has been illustrated in our experiment. Tran and Davis [18] model object motion and background motion in pixel-level by feature matching on a frame-to-frame basis. An occupancy map is maintained to stand for where the object is. More simply, Donoser and Bischof [7] tracks a single maximally stable extremal region(MSER) feature. Compared with these algorithm, our tracking framework works on a sophisticated feature motion generative model, which focuses on building the relationship between local feature motions and object global motion. Experiment results demonstrate a better performance of our tracker.

3. Feature Motion

In the paper, SURF feature is represented as $f = \{p, s, cl, ht\}$ where p is the 2-D position of the feature in the image coordinate, s is the feature scale, cl is the average r, g, b value in feature area, ht is the 128-bin oriented Haar response histogram.

3.1. Generative Model

we first introduce the generative model of a single feature motion observation, $v_t = (v_{x,t}, v_{y,t})$, at time t. Since we should not expect that object undergoes no deformation and follows a rigid motion during tracking, our generative model consists of two components. The first one is the consistent component, which aims to capture the agreement relationship between local feature's motion and object global motion. In particular, assume that a feature motion observation v_t generated by the consistent component, we model the probability density for v_t by the Gaussian density $p_c(v_t \mid \mathbf{c}_t, \Sigma_c)$. Here $\mathbf{c}_t$ denotes the object motion parameter at time t, and Σ_c is a fixed empirical covariance matrix.

The second component of the model is to express the irrelevance of some feature motion observations with object global movement. Such irrelevance would be brought to the system by observation noise, or by the case that certain parts move incompatibly with the whole object, such as arm swings backward when body moves forward. We refer to this component as a random walk process, and the probability density for a observation v_t generated by this component, $p_r(v_t)$, is taken to be a uniform distribution over the observation domain.

These two components are combined in a probabilistic mixture model for a feature motion v_t,

$$p(v_t \mid \mathbf{c}_t, \mathbf{m}_t) = m_{c,t} p_c(v_t \mid \mathbf{c}_t) + m_{r,t} p_r(v_t), \quad (1)$$

where $\mathbf{m}_t = (m_{c,t}, m_{r,t})$ are the mixing probabilities for this feature at time t. It is important to notice that the mixing probabilities reveals how likely a feature's motion is consistent with the object global motion.

3.2. On-line Learning

In this section, an on-line EM algorithm is developed to learn mixture model parameters $\mathbf{m}_t$.

We first introduce a temporal window function, which gives an exponential envelope located at the current time, $S_t(k) = \alpha e^{-(t-k)/\tau}$, for $k \leq t$. Here, $\tau = n_s / \log 2$, where n_s is the half-life of the envelope in frames, and $\alpha = 1 - e^{-1/\tau}$, so the envelope $S_t(k)$ sum to 1. Under such an envelope, recent observations are given more consideration than remote ones. The log-likelihood of the observation history, $\mathbf{v}_t = \{v_k\}_{k=0}^{t}$, weighted by the envelope function is:

$$L(\mathbf{v}_t \mid \mathbf{m}_t, \mathbf{c}_t) = \sum_{k=0}^{t} S_t(k) \log p(v_k \mid \mathbf{m}_t, \mathbf{c}_k) \quad (2)$$

where $\mathbf{m}_t$ denote the mixture model parameters relevant to the observations under the temporal support envelope $S_t(k)$.

In the standard EM algorithm, which intend to maximize the log-likelihood $L(\{v_k\}_{k=0}^{t})$, given a current value for $\mathbf{m}_t$, the E-step calculates the ownership probabilities to mixture components for each observation v_k:

$$o_{i,t}(v_k) = \frac{m_{i,t} p_i(v_k \mid \mathbf{m}_t, \mathbf{c}_k)}{p(v_k \mid \mathbf{m}_t, \mathbf{c}_k)} \quad (3)$$

for $i \in \{c, r\}$. Conditioned on these ownerships, the M-step then provides new maximum likelihood estimates for the parameters $\mathbf{m}_t$ by:

$$m_{i,t} = \sum_{k=0}^{t} S_t(k) o_{i,t}(v_k) \quad (4)$$

for $i \in \{c, r\}$. Here a normalization constant, which makes the mixing probabilities sum to one, is omitted.

This EM algorithm works under the assumption that all observations from previous time be stored to compute $o_{i,t}(v_k)$, which is impractical for an on-line approach. As mentioned above, the mixing probabilities stand for the consistency of this feature's motion to object global motion, and thus change slowly through time. In the temporal window, we could make approximation and exploit a recursive formula to execute EM algorithm on-line:

$$\begin{aligned} m_{i,t} &= \sum_{k=0}^{t} S_t(k) o_{i,t}(v_k) \\ &\approx S_t(t) o_{i,t}(v_t) + \sum_{k=0}^{t-1} S_t(k) o_{i,k}(v_k) \\ &\approx \alpha o_{i,t}(v_t) + (1-\alpha) m_{i,t-1} \end{aligned} \tag{5}$$

where $\alpha = 1 - e^{\frac{1}{\tau}}$ denotes a learning rate. In the last step, an approximation $1 - e^{\frac{1}{\tau}} \approx \frac{1}{\tau}$ is used. With this online EM algorithm, mixture model parameters could be updated efficiently.

3.3. Feature Correspondence by Graph Matching

Observing Features' motions could be formulated as feature correspondence in two feature sets. Let F' be the set of object features of interest in frame $t-1$, and F'' be the set of features detected from frame t. If a matching between them, $M : F' \to F''$, is determined, motion of a feature f in the set F' could be obtained by $v_f = p_{M(f)} - p_f$, where p is the position part in the feature descriptor, and $M(f)$ denote the corresponding feature of f in feature set F''.

Since the distinctiveness of SURF descriptor would be inadequate to provide sound matching result in complicated situation, object structure information should be taken into account in feature corresponding process, which lead us to graph matching technique. Graph matching is a challenging optimization problem which received considerable attention in the literature [12, 8, 5, 6, 17]. In this paper, feature correspondence is formulated as a matching energy minimization problem, and a dual decomposition approach is employed to search for a global optimal matching result. Technical details is referred to [17].

4. Object Tracking

Object global motion is modeled as a 2-D affine transform, $\mathbf{c} = (u_x, u_y, \varphi, \rho)$, where u_x, u_y is the spatial translation, φ and ρ describe the rotation and scale changes, respectively. Given the transform $\mathbf{c}_t$, the expectation of the motion of feature f located at p_f is a vector, $v^E_{f,t} = \mathbf{w}(\mathbf{c}_t, p_f) - p_f$, pointing from its original position to new position. Here, $\mathbf{w}$ is the warp function. And the probability density of observing its motion $v_{f,t}$ under the consistent model is Gaussian function which centered at $v^E_{f,t}$ with covariance matrix Σ_c. To find an optimal transform $\mathbf{c}_t$ we maximize the sum of the observation log likelihood on the matched feature pair set M_c:

$$L(\{v_{f,t}\}_{(f,e)\in M_c}) = \sum_{(f,e)\in M_c} \log p(v_{f,t} \mid \mathbf{c}_t, \mathbf{m}_{f,t-1}) \tag{6}$$

Here we impose on $\mathbf{c}_t$ no priors such as slow motion, small acceleration, since we found in our experiments that any such prior would turn into a nuisance in videos shot by a shaky camera. To evaluate $\mathbf{c}_t$ we apply the extension of EM algorithm described in [9], which suggests an iterative way and guarantee to increase the log likelihood for Gaussian distributions at each iteration. Excluding the full details for brevity, we just outline the E-step and M-step. In E-step, the "ownership probabilities" $o_{c,t}$ is calculated for the motion $v_{f,t}$ of each feature f in $\{f \mid (f,e) \in M_c\}$, as in Equation 3.

Algorithm 1 SURF Tracking Algorithm

```
Input:    $n$ video frames $I_1, \dots, I_n$
          Ellipse $e_1$ of object in first frame
Output:   Ellipses $e_2, \dots, e_n$

Initialization(for frame $I_1$):
  • Initialize list $ob\_lst$ by features extracted in area of $e_1$
    with initial mixture parameter $\mathbf{m}_{ini}$.

For each new frame $I_i$ do:
  • Extract features in and around the area of $e_{i-1}$, and
    preserve in list $dt\_lst$.
  • Do feature corresponding between sets $ob\_lst$ and
    $dt\_lst$ to detect features' motions $\{v_{f,i}\}$.
  • Evaluate object motion $\mathbf{c}_i$ with $\{v_{f,i}\}$ by Equation 7.
  • Object new position $e_i \leftarrow \mathbf{w}(\mathbf{c}_i, e_{i-1})$.
  • Update each feature $f$ in list $ob\_lst$:
      – Update $f$'s descriptor $\{p, s, cl, ht\}$ and model parameter
        $\mathbf{m}_{f,i}$ by Equation 5, if $v_{f,i}$ detected.
      – Update $f$'s position $p$ by motion $v^E_{f,i}$, downweight its
        $m^c_{f,i}$ with factor $f_{dw}$ (keep $m^c_{f,i} + m^r_{f,i} = 1$),
        if $v_{f,i}$ not detected.
      – Add newly appeared features(from list $dt\_lst$ and
        in the area of $e_i$ but not matched).
      – Check all features and abandon those with $m^c_{f,i}$
        lower than an empirical threshold $m_{thr}$.
```

In the M-step, we use these ownership probabilities to build a linear system for the update $\delta \mathbf{c}_t$:

$$\sum_{(f,e)\in M_c} o_{c,t}(v_{f,t}) U^T \Sigma_c^{-1} (U\delta \mathbf{c}_t - C) = 0 \qquad (7)$$

where U is the derivative of $\mathbf{w}(\mathbf{c}_t, p)$ with respect to $\mathbf{c}_t$, and C equals to $v_{f,t} - v_{f,t}^E$. The details of our method is given in Algorithm 1.

4.1. Occlusion

Previously proposed tracking algorithm would be challenged in the case of occlusion. When object is partially or completely occluded, updating mechanism would admit background SURF features in object region to be added into object representation as newly appeared object features. In next frames, motion observations of these background features would disturb the tracker to make correct estimation on object global motion, and result in tracking failure.

A direct solution is that we monitor background features as well as object features. Specifically, a set of SURF features extracted from object surrounding is maintained to model the background. As object feature set, this background feature set also takes part in feature correspondence process, and is updated frame by frame. Thus, when object is occluded, background features detected in object region will be matched with features in background feature set, since they has been previously added into background feature set in several frames before. This strategy helps our tracker overcome short term occlusion, also enable us to detect occlusion actively: when newly detected SURF features, which matched with previous background features, are found in object region now, an occlusion is most likely occurring. Experiments were conducted to approve our solution to occlusion.

5. Experiment

We implement our algorithm with the OpenCV library. On Pentium-4 3.0GHZ machine, averagely, the computation time is less than 120ms for single frame of size 640×480. The empirical parameters in the algorithm are well selected and all fixed in our following experiment. Typically, the initial mixture model parameters are set to $\mathbf{m}_{ini} = (0.2, 0.8)$, and the abandon threshold m_{thr} is 0.15. We demonstrate the performance of our algorithms through a number of video sequences, and receive promising results.

In the first experiment, we track a calendar moved by human hand. The calendar undergoes translation, scaling and rotation in the sequence. our tracker collects the motions of SURF features of the calendar, and then estimates its global motion. Figure 1 gives tracking results for some frames in this 180-frame sequence, and it is shown that our tracker catches exactly the calender's motion.

In the second sequence, we track a pedestrian walking on the playground. When this pedestrian walks out of the building shadow, the illumination on her has greatly changed. Still, our tracker successfully follows the pedestrian through the whole sequence. Figure 2 shows several tracking results of this 67-frame sequence. Note that, even though illumination on the object has steeply increased, some SURF features, such as feature on head, remain detected and make tracking stable.

The third experiment is to illustrate the capability of our tracker to deal with background clutter. Most of appearance-based tracking algorithms fail to follow the object in such case because the complexity of background would lead the appearance model to drift away from the true one . In this sequence, a girl in white cloth runs in front of her schoolmates who are also wearing white school suits, drastic appearance changes and motion blurs occur due to both object moving and camera floating. Figure 3 shows several tracking results of our algorithm and ensemble tracking [1] for comparison. The ensemble tracker tries to adapt classifiers to catch up appearance changes, but it turns to wrong target, the girl's trousers, as the confidence maps shown. One reason for its failure is that pixels with similar color in the background is hard to distinguish from pixels in object, which makes its classifiers drift away. Another reason is that ensemble tracker drops object structure information during classification, so it could not adapt well to the dramatic appearance changes. In contrast, our tracker works well in this situation. Intuitively, we display features' *life-span* distribution of this sequence in Figure 4. The phenomenon that most features suffer from short *life-span*, reveals the fact that most features become demoded quickly in fast appearance changes. Even so, feature corresponding process encapsulates object structure information in matching to guarantee most of correct motion observations, and generative model exactly depicts the relationship between local features' motions and object global motion. We still have many long-lived features due to the high repeatability and robustness of SURF feature, which contribute a lot to the stableness of our tracker.

The fourth experiment aims to illustrate our tracker's ability to handle several challenging situations together. In this sequence, a school girl is running in playground. The background is crowded by many school students with similar appearances. In this sequence, drastic motion blur and appearance deformation also occur, and the girl is occluded by her teacher for several frames. Figure 5 shows some tracking results of this 100-frame sequence, which illustrate that our tracker still works stably and accurately when these challenging situations happens together. Here, the sum of area of background features in object region is calculated and its ratio to the whole object region area is used as an occlusion indicator. When object is occluded, background

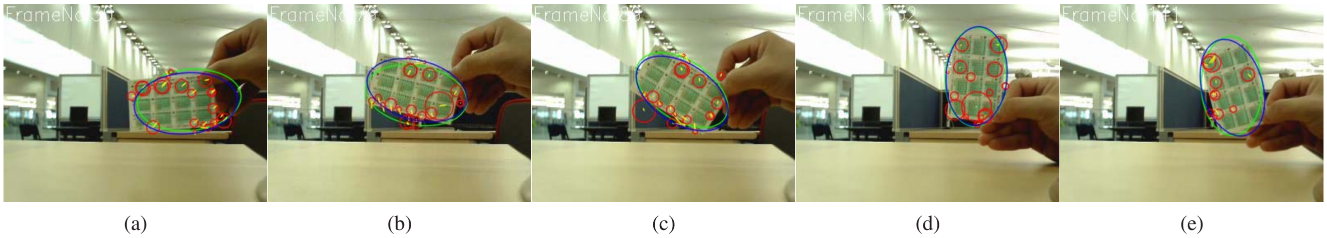
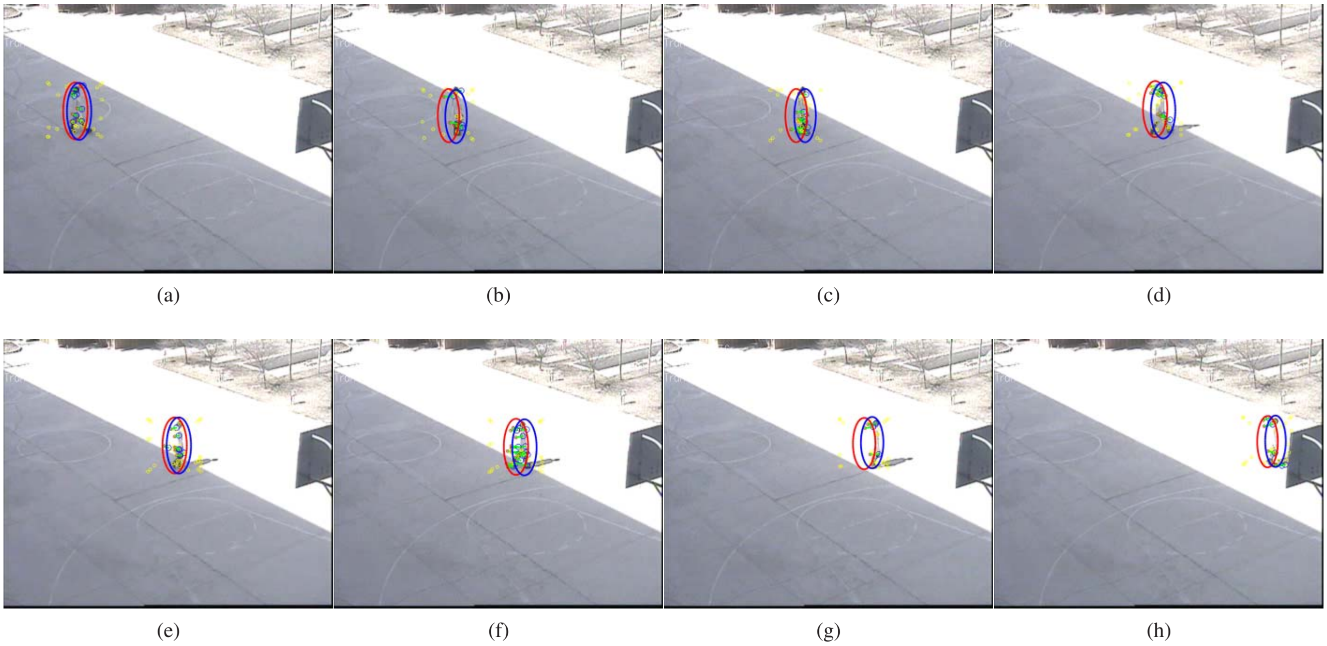

(a) (b) (c) (d) (e)

Figure 1: Tracking calendar. By monitoring its SURF features, our tracker catches exactly the motion of the calendar, which undergoes translation, rotation and scaling. Tracking results of frames 30, 79, 89, 132 and 141 are shown. Green and blue ellipses stand for object position in previous frame and current frame, respectively.


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

(e) (f) (g) (h)

Figure 2: Tracking human in illumination changes. When the pedestrian walks out of the building shadow, the illumination on her has steeply increased. Still, our tracker successfully follows the pedestrian through the whole sequence. Frame 22, 37, 39, 43, 46, 48, 51 and 58 are shown. Red and blue ellipses stand for object position in previous frame and current frame, respectively. Green lines denote feature motion observations.

features are found in object region. Tracking results of frames 89, 91, 92 show that such an indicator exactly responses with occlusion.

6. Conclusion

we present a novel motion-based tracking framework. Object is represented by a set of SURF feature of interest. Feature motions are observed exactly by a feature correspondence process. A generative model is proposed to depict the relationship between local feature motions and object global motion. And object affine motion parameter is estimated in term of maximum likelihood of feature motion observations. Then, an updating mechanism is employed to adapt object representation. Experiments show that our framework can get reliable tracking under dramatic appearance deformation, background clutter, illumination changes and occlusion.

However, the performance of our framework is degraded if too few features are detected on object. This happens when the object is too small or when its appearance is almost homogeneous(a single-color blob). In such cases, we can simply substitute a blob tracker [4] to keep tracking the object. Our tracker also fails in some cases that object moves in a way much different from a 2-D affine transform,

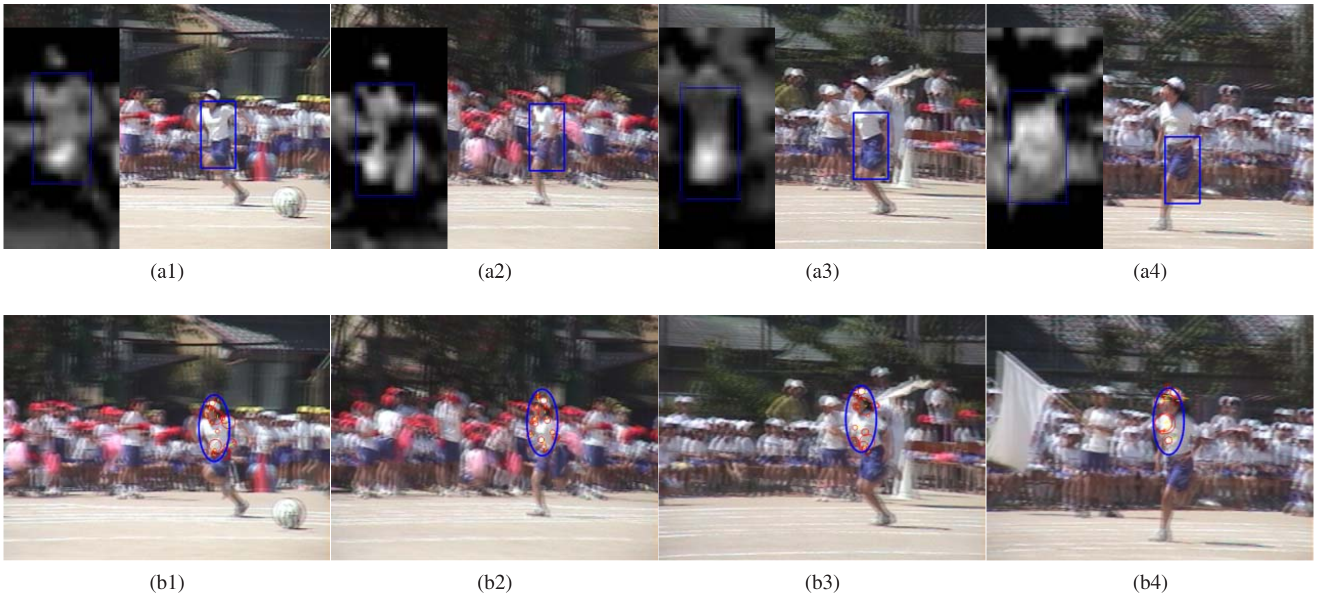


Figure 3: Tracking human in background clutter. The upper row is the result of ensemble tracker. the confidence map for each frame is attached at the left-bottom corner, which reveals that ensemble tracker fails to capture the dynamic object appearance model in background clutter. The lower row is the result of our method. graph based feature correspondence encapsulates object structure information in matching to guarantee correct feature motion observations in background clutter.

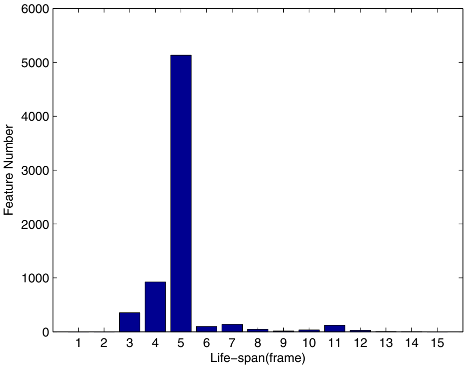


Figure 4: Feauture life-span distribution in tracking. Most features suffer from short life-span due to the dynamic change of object appearance. Still, some long-lived features account for the high repeatability and robustness of SURF feature.

such as out-plane rotation. These direct the future research to the following aspect: (1) Since features from object and background always have different motions, motion classifiers could be used to distinguish object features from background ones. Such classifiers should also be trained on-line like that in [1]. (2) Complicated motion of object, such as out-plane rotation, would result in dynamic shape of object. level-set based contour evolution could be introduced into tracking framework to handle such dynamic changes.

7. Acknowledgements

The authors would like to thank colleagues in OMRON Corp for helpful discussions. Wei He and Hongtao Lu are partially supported by the NLPR under grant 09-4-1, 863 (NO. 2008AA02Z310) and 973 Program 2009CB320900.

References

[1] S. Avidan. Ensemble tracking. In *CVPR*, pages 494–501, 2005.

[2] H. Bay, T. Tuytelaars, and L. V. Gool. Surf: Speeded up robust features. In *ECCV*, pages 404–417, 2006.

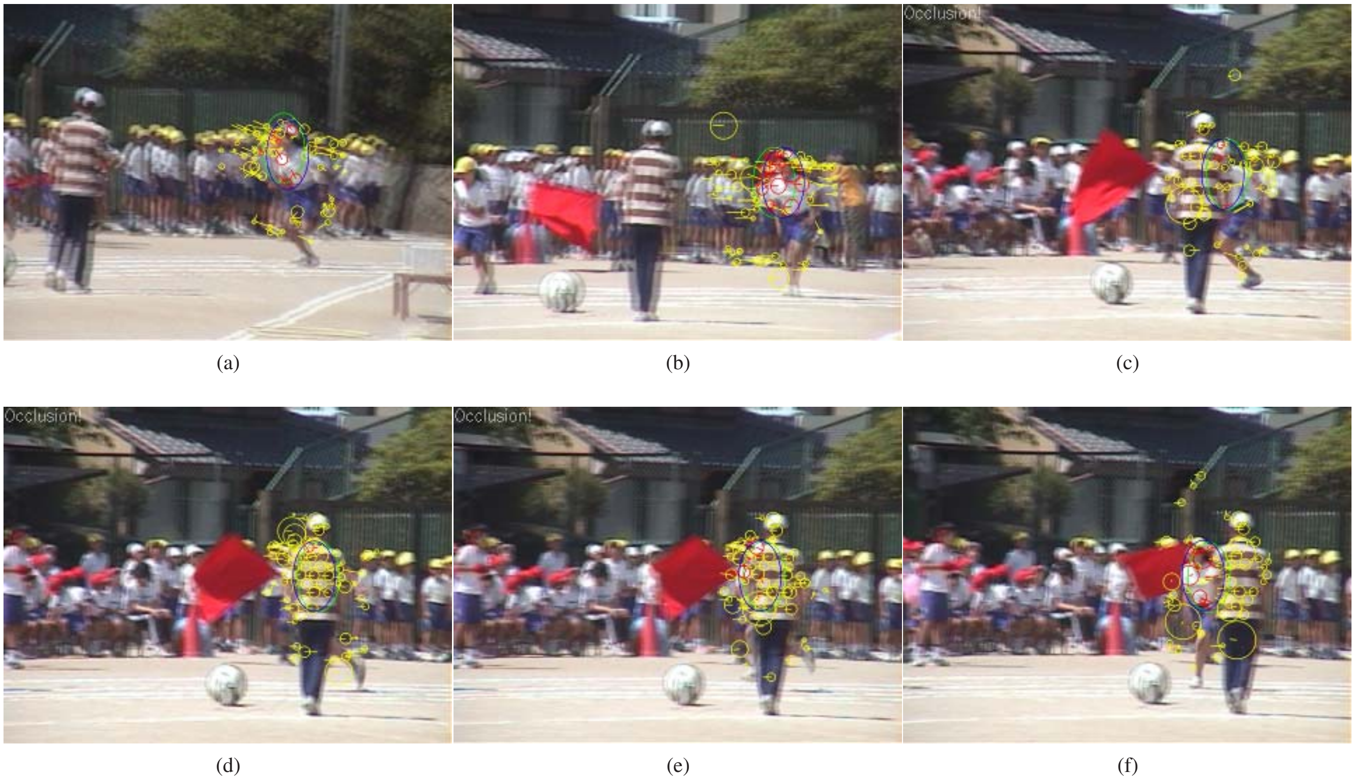
[3] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *PAMI*, 24:509–522, 2002.

[4] R. T. Collins. Mean-shift blob tracking through scale space. In *CVPR*, volume 2, pages II–234–40 vol.2, 2003.

[5] D. Conte, P. Foggia, C. Sansone, and M. Vento. Thirty years of graph matching in pattern recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 2004.

[6] T. Cour, P. Srinivasan, and J. Shi. Balanced graph matching. In *NIPS*, 2006.

[7] M. Donoser and H. Bischof. Efficient maximally stable extremal region (mser) tracking. In *CVPR*, pages 553–560, 2006.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 5: Challenging situations including appearance changes, background clutter and occlusion. Tracking results of frames 77, 81, 89, 91, 92 and 93 are shown. In each sub-figure, background features are denoted by yellow circles, and object features are red circles. Green and blue ellipses stand for object position in previous frame and current frame, respectively. Here, An area ratio based occlusion indicator is used to response with occlusion, as shown in frames 89, 91, 92.

[8] S. Gold and A. Rangarajan. A graduated assigment algorithm for graph matching. In *PAMI*, pages 377–388, 1996.

[9] A. Jepson and M. J. Black. Mixture models for optical flow computation. In *CVPR*, pages 760–761, 1993.

[10] A. D. Jepson, D. J. Fleet, and T. F. El-maraghi. Robust online appearance models for visual tracking. In *PAMI*, pages 415–422, 2001.

[11] Y. Ke, R. Sukthankar, and R. Sukthankar. Pca-sift: A more distinctive representation for local image descriptors. In *CVPR*, pages 506–513, 2004.

[12] M. Leordeanu and M. Hebert. A spectral technique for correspondence problems using pairwise constraints. In *ICCV*, volume 2, pages 1482 – 1489, October 2005.

[13] D. G. Lowe. Object recognition from local scale-invariant features. In *ICCV*, pages 1150–1157, 1999.

[14] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. In *PAMI*, pages 257–263, 2003.

[15] J. Shi and C. Tomasi. Good features to track. In *CVPR*, pages 593–600, 1994.

[16] F. Tang and H. Tao. Object tracking with dynamic feature graph. In *ICCV*, pages 25–32, 2005.

[17] L. Torresani, V. Kolmogorov, and C. Rother. Feature correspondence via graph matching: Models and global optimization. In *ECCV*, 2008.

[18] S. Tran and L. Davis. Robust object tracking with regional affine invariant features. In *ICCV*, 2007.

[19] T. Yu and Y. Wu. Differential tracking based on spatial-appearance model (sam). In *CVPR*, 2006.

[20] H. H. Zhang, W. M. Huang, Z. Y. Huang, and L. Y. Li. Affine object tracking with kernel-based spatial-color representation. In *CVPR*, 2005.