Spatial selection for attentional visual tracking

Ming Yang, Junsong Yuan, Ying Wu
EECS Dept., Northwestern Univ.
2145 Sheridan Road, Evanston, IL 60208
mya671,jyu410,yingwu@ece.northwestern.edu

Abstract

Long-duration tracking of general targets is quite challenging for computer vision, because in practice target may undergo large uncertainties in its visual appearance and the unconstrained environments may be cluttered and distractive, although tracking has never been a challenge to the human visual system. Psychological and cognitive findings indicate that the human perception is attentional and selective, and both early attentional selection that may be innate and late attentional selection that may be learned are necessary for human visual tracking. This paper proposes a new visual tracking approach by reflecting some aspects of spatial selective attention, and presents a novel attentional visual tracking (AVT) algorithm. In AVT, the early selection process extracts a pool of attentional regions *(ARs) that are defined as the salient image regions which have good localization properties, and the late selection process dynamically identifies a subset of discriminative attentional regions (D-ARs) through a discriminative learning on the historical data on the fly. The computationally demanding process of matching of the AR pool is done in an efficient and innovative way by using the idea in the locality-sensitive hashing (LSH) technique. The proposed AVT algorithm is general, robust and computationally efficient, as shown in extensive experiments on a large variety of real-world video.*

1. Introduction

The rapid growth of computing power allows us to explore video for automatically analyzing, recognizing, interpreting, and understanding the activities and events in video. A fundamental step in this exploration is tracking general targets in unconstrained environments for a long duration in video, *e.g.*, people may want to track an arbitrary designated image region for video analysis. This turns out to be quite challenging, because the target may undergo large uncertainties in its visual appearance due to many factors such as varying lighting conditions and unpredictable occlusions, and because the environments may be cluttered and distractive.

Over several decades, the research of target tracking in computer vision has resulted in many outstanding visual tracking algorithms. The research started as feature point matching and optical flow estimation in consecutive image frames. Since then, visual tracking has been largely formulated in a *match-and-search* framework of motion estimation, and has evolved from simple 2D translation to complex motions. In this framework, great research efforts have been devoted to two important issues, *i.e.*, the matching criteria and the searching methods. Matching criteria (or observation models, or likelihood models) are critical in tracking, as they define the invariants on which the tracking process is based and contribute to the objective functions that the motion estimators need to optimize. An early treatment assumed the constancy in the brightness patterns in images, but it turned out this was too restricted and rarely held in practice. Then, the matching criteria have been extended by considering illuminations, by using more tolerant cues and invariant features [6], by modelling clutter generating processes [15], by integrating observation processes on multiple kernels [13], by involving exemplars [20], by learning the variations in the target's appearance [3], by using generative models [16, 20], by on-line adaptation [16, 5, 2, 11], etc.

When our expectation of visual tracking has rapidly grown from tracking simple image tokens to complex image regions, from simple points to nonrigid targets, from controlled environments to unconstrained environments, the matching criteria become sophisticated and more and more object recognition components are involved because it can be very difficult to find obvious invariants even if they may exist. It seems that this leads to a paradox: efficient tracking should be based on simple and low-level matching criteria that do not involve higher level visual processing, but such low-level criteria may not be able to handle the uncertainties in visual appearances. Complex and high-level criteria may cope with the uncertainties, but they tend to be computationally demanding as they are late stages in visual perception. Therefore, the traditional match-and-search framework for visual tracking seems to be inadequate.

On the contrary to the tremendous challenges we have

encountered in developing tracking algorithms, being able to persistently follow moving objects seems to be a very basic functionality in human visual perception. It is so natural and intuitive that we may not be aware of how complex it is. Although the details in human perception on visual dynamics are still largely mysterious, the studies in psychology, neuroscience and cognitive sciences have obtained substantial evidence and interesting findings, based on which several hypothetical theories have been proposed [17]. For example, evidence shows that human visual perception is inherently selective. Perceiving realistic scenes requires a sequence of many different fixations through the saccadic movements of the eye. Even when the eye is fixated on a particular location, the act of *visual attention* (like the movements of an internal eye or the so-called "mind's eye") selects and determines what subset of the retinal image gets full processing [18]. An interesting question is how we can take advantage of these studies to develop more powerful visual tracking algorithms.

This paper presents a new visual tracking approach that reflects some findings of selective visual attention in human perception. Recent studies in 90s have indicated that *selective attention* may act in both early and late stages of visual processing but under different conditions of perceptual load [17]. *Early selection* may be based on innate principles obtained through evolution, while *late selection* is learned through experiences. By integrating both mechanisms, our new computational model may be able to resolve the paradox of low-level and high-level matching criteria in the traditional match-and-search paradigm: we connect the low-level matching to the early attentional selection and the high-level process to the late selection.

We develop a novel attentional visual tracking (AVT) algorithm based on spatial selective attention. Specifically, the early selection process extracts a pool of *attentional regions* (ARs) that are defined as the salient image regions that have good localization properties, and the late selection process dynamically identifies a subset of discriminative attentional regions (D-ARs) through a discriminative learning on the historical data on the fly. The computationally demanding process of matching of the AR pool is done in an efficient and innovative way by using the idea in the locality-sensitive hashing (LSH) technique.

The proposed AVT algorithm is general, robust and computationally efficient. Representing the target by a pool of attentional regions makes AVT robust to appearance variations due to lighting changes, partial occlusions and small deformation. Spatial attentional selection of ARs allows AVT to focus its computational resources to more informative regions to handle distractive environments and targets with complex shapes. Pre-indexing the features of ARs based on LSH enables fast matching in order to search a large motion parameter space. In addition, AVT can be used as a region tracking tool for tracking general objects without any prior knowledge. These merits have been shown in extensive results on a variety of real-world sequences.

This work is different from some recent work on on-line selection of discriminative features [5] and other adaptive methods [2, 11, 16], in that AVT does not select global features but spatially-distributes local attentional regions so as to enable a broader and a more robust selection. In addition, AVT is also quite different from the fragment-tracking [1] where the target is evenly divided into fragments in a predefined way with no selection.

2. Overview of Attentional Visual Tracking

Selective attention is crucial to visual perception, because the amount of information contained in visual scenes is far more than what we can process at one time and thus the visual system has to sample visual information over time by some inherently selective perceptual acts, including spatial selection that directs the attention to a restricted region of the visual field. Selective attention may be made possible by two kinds of heuristics. One is based on innate principles obtained through evolution, and could be performed in the early stage of visual processing. The other one is learned through experience and might happen later in visual processing. Both are important in the human visual system.

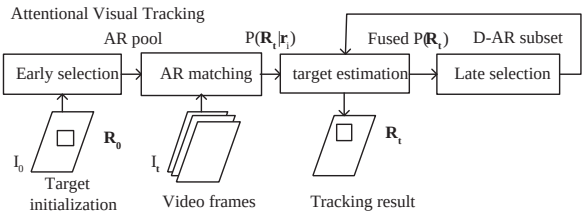


Figure 1. Attentional visual tracking.

As summarized in Fig. 1, the proposed attentional visual tracking reflects these perceptual findings of spatial selection in visual attention. AVT has 4 important processes:

- **Early attentional selection**. As the first step, it extracts informative and salient image regions called *attentional regions* (ARs) from images. This is a low-level process, as it is only concerned on local visual fields. In this paper, we treat those image regions that have good localization properties as ARs, and the AR is characterized by its color histogram;
- **Attentional region matching**. Once a pool of ARs are extracted by the early selection process, they will be used to process an incoming image to localize their matches. An innovative method is proposed to conquer the large computational demands, by pre-indexing the features of ARs. For each frame, the matching set of each AR is obtained and used to estimate a belief of the target location;

- **Attentional fusion and target estimation**. The beliefs of all the ARs are fused to determine the target location. A subset of ARs have larger weights in the fusion process, because they are more discriminative. This subset of ARs are obtained by the late selection process in the previous time frame;
- **Late attentional selection**. This process reflects some higher level processing to learn and adapt to the dynamic environments. Based on the collected history tracks of ARs, a discriminative selection is performed to identify a subset of most discriminative ARs (or D-ARs) that exhibit the distinctive features of the target from the environments. They will have larger weights in the attentional fusion process at the next frame.

3. Components in Attentional Visual Tracking

3.1. Early attentional selection

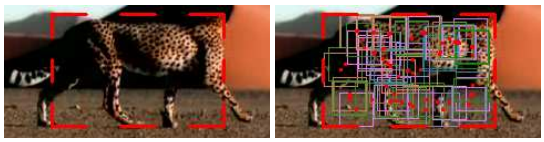
Visual information is so rich that the human visual system has a selective attention mechanism to sample the information over time in processing. Early attentional selection that is believed to act in the very early stage of visual perception performs the initial pre-filtering task, which should not involve much higher level processing such as object recognition. Early selective attention is likely to be based on innate principles of human perception, *e.g.*, to attend certain information that is evolutionary advantageous. For example, moving objects are generally important for survival and appear to play an important role in early attention.

This section describes a spatial selection method for this early attentional process. We call the selected image region as *attentional regions* (ARs). As discussed before, motion detection appears to play an important role in early attention. Therefore, the selection of attentional regions should be sensitive to motion (*i.e.*, informative) but insensitive to noise (*i.e.*, stable). Mathematically, any change in the appearance of such an AR should correspond to a unique motion estimation, and the small differences between two appearance changes should not lead to dramatic different motion estimates (i.e., well-behaved).

In view of this, we choose to use the criterion and the region extraction method described in [9] that views the stability of image region in motion estimation from system theory perspective. The appearance change of an image region is treated as measurements of the motion that is viewed as the system states. For some image regions, *e.g.*, homogeneous regions, the system states (motions) are *unobservable* from the measurements, *i.e.*, the motions of these regions are not fully recoverable from their appearance changes. Thus, they should not be attentional regions. In addition, image regions that lead to unstable systems, *i.e.*, small appearance changes result in dramatically different motion estimates, should not be attentional regions neither. Therefore, attentional regions can be selected by finding those regions that generate observable and stable systems. It was proved [9] that as an image region is characterized by its feature histogram, the stability of the linear motion estimation system can be evaluated by checking the condition number of a matrix that is only related to the properties of the corresponding image region. A more stable system has a lower condition number. Thus, in the proposed AVT algorithm, we select the pool of ARs by locating and extracting those salient image regions.

Specifically, at the first frame I_0, given the target initialization rectangle $\mathbf{R_0}$, we evenly initialize $N_{max} = 100$ tentative ARs inside the target. With an efficient gradient descent search algorithm [9], the tentative ARs converge to positions where the corresponding condition numbers are local minima. By removing the duplicated tentative ARs that have converged to the same location and those that have large condition numbers, the selected AR pool is obtained and denoted by $\{\mathbf{r}_1, \cdots, \mathbf{r}_N\}$. Their relations to the target are recorded for future target estimation in subsequent tracking. The number N of ARs is automatically determined by the early selection process itself, depending on targets, *e.g.*, we have observed $N = 60 \sim 70$ for large and complex objects and $N = 30 \sim 40$ for small and simple objects in our experiments. Then, the color histograms of $\{\mathbf{r}_1, \cdots, \mathbf{r}_N\}$ are obtained as the feature vectors $\{\mathbf{x}_1, \cdots, \mathbf{x}_N\}$ with D bins, *i.e.*, $\mathbf{x}_i = \{x_{i1}, \cdots, x_{iD}\}$.

As the color histograms on various image regions need to be calculated, the integral histogram technique [19] can be applied to save computation. In AVT, we implement a modified version of integral histogram that is able to retrieve histograms at arbitrary locations in constant time, but also consume moderate memory when using high resolution color histograms. Although the sizes and shapes of different ARs are not necessarily identical, to be able to process ARs in a uniform way, we impose all ARs at the same size and shape, *i.e.*, 30×30 squares initially. An example of the early selection of attentional region pool is shown in Fig. 2.



(a) initialization. (b) the pool of ARs.

Figure 2. Early selection of the attentional region pool.

3.2. Attentional region matching

For each frame I_t at time t, to locate the correct target position, all hypotheses in motion parameter space have to be evaluated to find the best matches to the ARs in the AR pool. Because the prior knowledge of the dynamics of the ARs is generally unavailable, exhaustively searching the motion

parameter space can provide the optimal performance. Although this is computational demanding, we have an innovative solution that significantly reduces the computation to allow close to real-time performance. This solution is based on the idea of the locality-sensitive hashing (LSH) [8], a powerful database retrieval algorithm.

Each AR needs to examine a large number of motion hypotheses. In this paper, the motion parameters include location (u, v) and scale s. Each motion hypothesis corresponds to a candidate image region. For all target hypotheses, all image patches $\mathbf{r}_c$ with the same size as ARs within the searching range of one AR constitute the *candidate region set* whose D dimensional color histograms are denoted as $\{\mathbf{y}_1, \cdots, \mathbf{y}_M\}$, where M is the size of the set. Generally the candidate region set has thousands of entries. We employ Bhattacharya coefficient to measure the similarity of two histograms $\mathbf{x}$ and $\mathbf{y}$, which is equivalent to Matusita metric [13] in L_2 distance form

$$d(\mathbf{x}, \mathbf{y}) = \sum_{j}^{D} \|\sqrt{x_j} - \sqrt{y_j}\|^2. \quad (1)$$

Matching a feature vector can be translated to query a database for the nearest neighbor points in the feature space. The worst case complexity is obviously linear, but this is not good enough. A significant speed-up can be achieved if the database can be pre-indexed. Locality-sensitive hashing (LSH) proposed by Indyk and Motwani [14] in 1998 and further developed in [8] aims to solve the approximate Nearest Neighbor (NN) problem in high dimensional Euclidean space. LSH provides a probabilistic approximation to this problem by randomly hashing the database with L locality-sensitive hashing functions, and only the points in the union of the hashing cells that the query point falling in are checked for nearest neighbors. This will lead to computational saves comparing with checking all the entries in the database. The idea is illustrated in Fig. 3. We refer readers to [14, 8] for details.

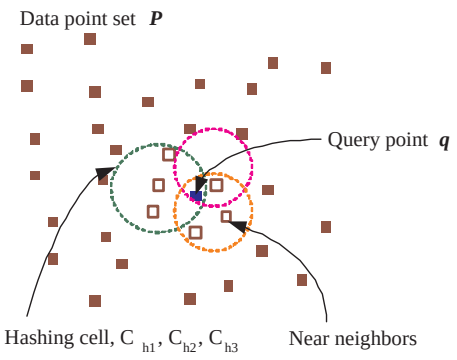


Figure 3. Illustration of query with LSH.

LSH has been applied in texture analysis [10] and fast contour matching [12]. To the best of our knowledge, LSH has not been used for on-line tracking before, although another database technique (K-D Trees) has been used for off-line (non-causal) tracking [4] by hashing the whole video sequence. When incorporating LSH into on-line visual tracking, there is a fundamental difference from database applications. In database applications, the indexing is done off-line and thus the computational overhead of indexing is not a critical issue. In our on-line tracking scenario, on the contrary, the indexing overhead cannot be ignored because both indexing the database and retrieving the database are performed during the tracking process. So computational costs of both indexing and querying are critical. This turns out to be very important in AVT implementation.

Now we have two data sets: one for the AR pool with size N and the other for the *candidate region set* with size M. Typically, N is within one hundred and M is several thousands. The worst case of complexity in matching is $O(N \times M)$. As discussed before, this complexity can be further reduced by applying LSH. Because the overhead of indexing needs to be considered, which data set should be chosen to be the database for LSH? If choosing the candidate set as the database, we find that the indexing overhead is not worth the gain for a limited number of queries from the AR pool. When we treat the AR pool as the LSH database, the computational gain is significant. The detailed complexity analysis will be present in a later section.

After querying all candidate region $\mathbf{r}_c$ with feature vectors $\mathbf{y}_c$ using LSH, the near neighbors within d_t in Matusita distance of each AR $\mathbf{r}_i$ are obtained and denoted as matching set $S_{\mathbf{r}_i} = \{\mathbf{r}_c | d(\mathbf{x}_i, \mathbf{y}_c) \leq d_t\}$.

3.3. Attentional fusion and target estimation

As described in the previous subsection, for each AR $\mathbf{r}_i$, the attentional region matching process outputs a matching set. Based on the recorded geometrical relation between this AR and the target (relative translation and scale in our implementation), the belief of this AR is the probability distribution of target location (u_t, v_t) of $\mathbf{R}_t$ given $\mathbf{r}_i$'s matching set, denoted by $P(\mathbf{R}_t|\mathbf{r}_i)$, which is approximated based on the set of matched candidate $\mathbf{r}_c \in S_{\mathbf{r}_i}$.

To estimate the target location and scale, the beliefs of all the ARs need to be fused. Because some ARs may have a substantial spatial overlap in images, their beliefs may be correlated. This dependency may complicate the exact fusion process. But we can approximate it by clustering the significantly overlapped ARs and treat them as one, so as to reduce the dependency. By doing this, we approximate the estimated distribution of target location $\hat{P}(\mathbf{R}_t)$ by

$$\hat{P}(\mathbf{R}_t) \approx \sum_{i}^{\hat{N}} P(\mathbf{R}_t|\mathbf{r}_i) P(\mathbf{r}_i), \quad (2)$$

where $\hat{N}$ is the number of AR clusters, and $P(\mathbf{r}_i)$ represents the prior distribution of $\mathbf{r}_i$ in I_t which is regarded as

uniform. The mode of the $\hat{P}(\mathbf{R}_t)$ determines the tracking result of $\mathbf{R}_t$. This is a voting process, as shown in Fig. 4.

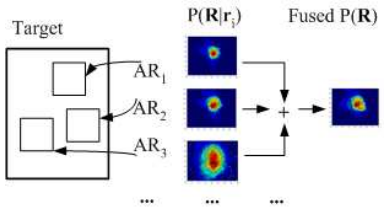


Figure 4. Estimation of target location.

It can be proved that this approximation only holds when $\hat{N}$ is large, because in this case the matching likelihoods of the ARs tend to dominate while the spatial correlations tend to be less critical. But this approximation is questionable when $\hat{N}$ is actually small. This is the limitation of our current implementation, as it is not quite suitable for tracking very small targets when only very few ARs are available and are largely correlated. Study on partial correlated information fusion is out of the scope of this paper.

3.4. Late attentional selection

As described in previous sections, attentional selection is indispensable to the human perception of visual dynamics. For long duration tracking, the human visual tracking system is able to adapt to changing environments and to discriminate the small differences of the target from the distractions. Tremendous psychological evidence [18] indicates that visual tracking involves both early selection and late selection. Late selection may be a serial of focused attention processes that are more proactive and involve higher level processing. For instance, the camouflage objects in background around the target may have similar appearances, *e.g.*, people in a crowd as shown in Fig. 9. When tracking objects with non-convex shapes, it is inevitable to include some background regions in target initialization as shown in Fig. 11 and 12.

Some ARs may be more distinctive and have a large discriminative power, so that they should play a more important role in tracking. Thus, during the tracking, a subset of discriminative attentional regions (or D-ARs) are selected through ranking their abilities of discerning target motion from the background motion. We select the subset of D-ARs based on the Principle of Minimum Cross-Entropy (MCE) [7], also called Maximum Discrimination Information (MDI). This is tantamount to measuring discrimination information between the case of using $P(\mathbf{R}_t|\mathbf{r}_i)$ to approximate $\hat{P}(\mathbf{R}_t)$, and the case of using it to approximate the distribution of background motion:

$$KL(P(\mathbf{R}_t|\mathbf{r}_i)||\hat{P}(\mathbf{R}_t)) - KL(P(\mathbf{R}_t|\mathbf{r}_i)||P(B)), \quad (3)$$

where $P(B)$ is the distribution of nearby background motion. Assume $P(B)$ to be uniform, this reduces to cross-entropy between $P(\mathbf{R}_t|\mathbf{r}_i)$ and $\hat{P}(\mathbf{R}_t)$:

$$\begin{aligned} H(\mathbf{r}_i, \mathbf{R}_t) &= H(P(\mathbf{R}_t|\mathbf{r}_i), \hat{P}(\mathbf{R}_t)) \quad (4) \\ &= H(P(\mathbf{R}_t|\mathbf{r}_i)) + KL(P(\mathbf{R}_t|\mathbf{r}_i)||\hat{P}(\mathbf{R}_t)) \\ &= E_{P(\mathbf{R}_t|\mathbf{r}_i)}(-\log(\hat{P}(\mathbf{R}_t))), \end{aligned}$$

where $H(\cdot,\cdot)$ stands for the cross-entropy of two distributions and $H(\cdot)$ is the entropy.

For each AR, the cross-entropy in a sliding temporal window of $\Delta t = 10$ frames are averaged with forgetting factor $\beta = 0.95$. The average cross-entropy $\tilde{H}(\mathbf{r}_i, \mathbf{R}_t)$ of all ARs are sorted to rank their discriminative abilities:

$$\tilde{H}(\mathbf{r}_i, \mathbf{R}_t) = \sum_{j=0}^{\Delta t} \beta^j H(P(\mathbf{R}_{t-j}|\mathbf{r}_i), \hat{P}(\mathbf{R}_{t-j})). \quad (5)$$

The top-ranked ARs are identified as D-ARs and have larger weights in fusion. In our implementation, we choose the top 75%. They will be used to estimate $\hat{P}(\mathbf{R}_{t+1})$ in the next frame. The D-ARs are not fixed but dynamically changing with respect to the changes of the environment. Fig. 5 shows the top 10 D-ARs (as red rectangles) for two sequences at 3 different frames.

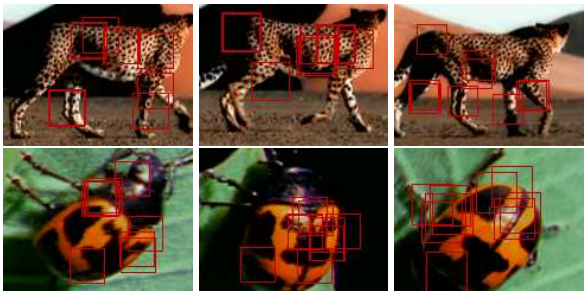


Figure 5. Examples of late selection of discriminative ARs.

3.5. Complexity analysis

In our AVT algorithm, the computation costs for integral histogram calculation, fusion of $P(\mathbf{R}_t|\mathbf{r}_i)$, mode seek of $\hat{P}(\mathbf{R}_t)$ are constant and relatively inexpensive. The most computational intensive module is attentional region matching. Exhaustive matching will involve $O(MN)$ times of D-dimensional vector comparison which is the basic computational unit in our analysis.

When the data set is hashed by LSH with L hashing functions, consider both indexing and query costs, the complexity is $O(ML + NL)$, where one hashing function is a D dimensional inner product calculation [8]. Therefore, the complexity ratio is approximately

$$\tau \approx \frac{O(ML+NL)}{O(MN)} \approx \frac{ML+NL}{MN}. \quad (6)$$

In the tracking scenario, the number of entries M in candidate set is much larger than the number of ARs N. Usually, M is several thousands and N is less than a hundred. Then, if we choose to hash the candidate set, L could be larger than N which means no speedup since we need to do indexing for every frame. So we hash AR pool with N elements, the complexity ratio $\tau \approx (L/N + L/M) \approx L/N$. Suppose there are $N = 100$ ARs, empirically $L = 20$ hashing functions are sufficient for querying the near neighbors within $d_t = 0.1$ at 0.9 probability. The computation reduces to approximately $\tau = 1/5$, if $N = 36$ and $L = 10$, $\tau = 0.28$. With this efficient matching, we can search a larger portion of the motion parameter space, *e.g.*, in our implementation, $[-20, +20]$ for (u, v) respectively and 3 scales ranging from 0.95, 1.0, and 1.05. For large targets, we down-sample the candidate region set to ensure $M \leq 3000$. The algorithm is implemented in C++ and tested on a Pentium-IV 3GHz PC. With moderate code optimization, the program runs at $10 - 15$ fps on average.

4. Experimental results

4.1. Settings

We test the proposed AVT algorithm for a variety of challenging real-world sequences including 3 primary types: quick motion with occlusion, camouflage environments, and objects with complex shapes. Note that in these tests, there are also scale and lighting changes. The targets include pedestrian, people in crowd, wild animals, bicycle and boat *et al*. The AVT tracker is compared with the Mean-shift tracker [6] in the same enhanced YCbCr space with 1040 bins (32×32 for Cb and Cr and 16 bins for Y when the pixel is too dark or too bright). Most of the video clips are downloaded from *Google Video*.

4.2. Quantitative comparison

For the quantitative comparison, the evaluation criteria of tracking error are based on the relative position error between the center of the tracking result and that of the ground truth, and the relative scale normalized by the ground truth scale. A perfect tracking expects the position differences to be around 0 and the relative scales close to 1.

We manually labeled the ground truth of the sequence `Walking` for 650 frames. The walking person, as shown in Fig. 7, is subjected to irregular severe occlusion when passing behind the bush. As indicated in quantitative comparison in Fig. 6, AVT performs extremely well, but mean-shift loses track at frame 164 and never recovers.

4.3. Quick motion with occlusion

As shown in Fig. 8, sequence `Horse Ride` involves very quick motion with occasional severe occlusions. The top row shows AVT tracking results where the first frame displays the attentional region pool. The second row shows

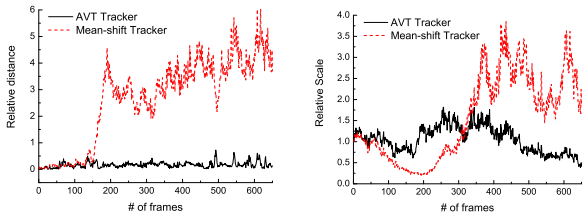


Figure 6. Quantitative comparison of relative position error and relative scale for tracking results of sequence [`Walking`].

Mean-shift tracker's results. For AVT tracker, the target is displayed as red dash rectangle, and the pixels covered by more than one D-AR are highlighted by increasing the luminance and the D-AR regions are surrounded by solid red lines. When there are too few matches for ARs, occlusion is detected and displayed with a white dash bounding box. Mean-shift tracker drifts after a serious occlusion is present at frame 54, while AVT tracker is able to keep the track by a few attentional regions.

4.4. Tracking in camouflage environments

Camouflage environments, *i.e.*, similar or even identical objects around the target, is very challenging for tracking. We demonstrate AVT's advantages by tracking one people in crowd (Fig. 9), and a zebra with similar texture nearby (Fig. 10). The scale of Mean-shift tracker becomes unstable when nearby background presents similar color histograms, while AVT is quite robust in camouflage environments due to the selection of D-ARs.

4.5. Objects with complex shapes

Tracking objects with complex shapes is difficult in practice. Since it is not reasonable to require initialization to give the accurate boundary of the target, some background image regions will be inevitably included in the target. As illustrated in Fig. 11 and Fig. 12, the ground and some water are cropped in the targets. The ARs on the background are not correlated to the target's motion, thus they have high cross-entropy and are excluded from the D-AR subset. On the contrary, Mean-shift tracker tries to match the holistic color histogram which is likely to be distracted by the background regions. More tracking results on a variety of general objects are shown in Fig. 13.

5. Conclusion

In this paper, we propose a novel and promising tracking algorithm inspired by findings of human visual perception. It is suitable for tracking general objects without any prior knowledge. Target is represented by an attentional region pool which brings robustness against appearance variations. Dynamically spatial selection of discriminative attentional regions on the fly enables the tracker to handle camouflage

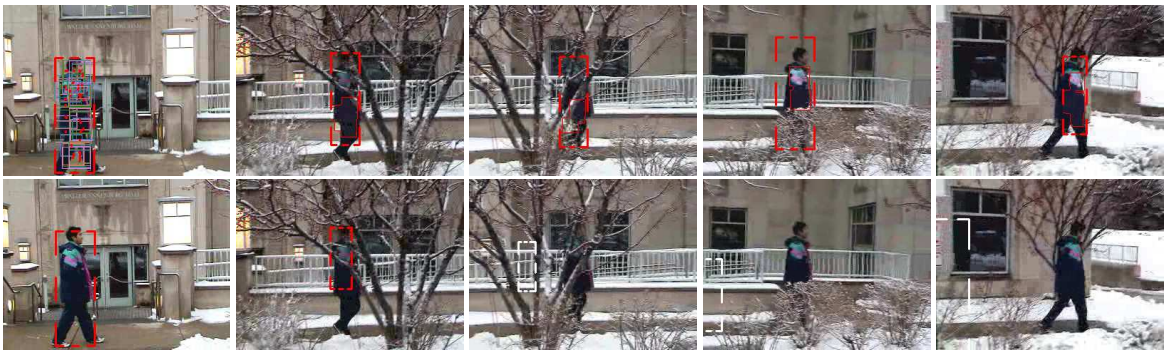


Figure 7. Tracking [`Walking`] for frame #1, 130, 164, 254 and 650, (1st row) AVT tracker (N=55), and (2nd row) Mean-shift tracker.

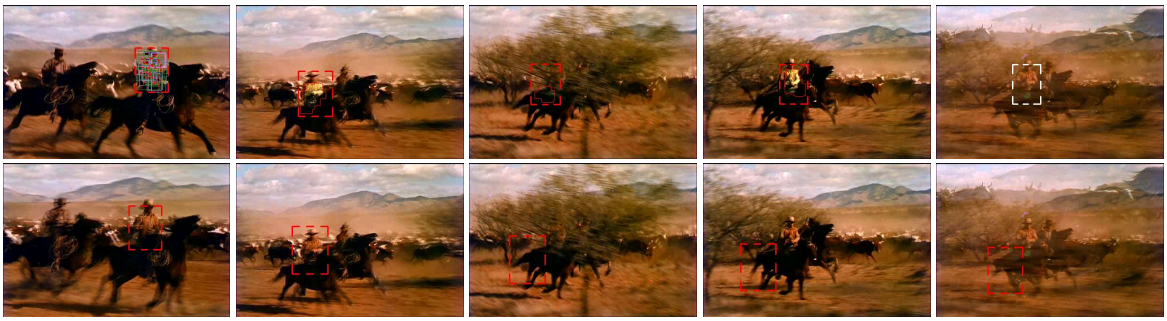


Figure 8. Tracking [`Horse Ride`] for frame #1, 40, 54, 58 and 60, (1st row) AVT tracker (N=45), and (2nd row) Mean-shift tracker.

environments and objects with complex shapes. In addition, by introducing LSH to on-line tracking, the proposed AVT is computationally feasible. Our future work includes 3 aspects: 1) extending our current AVT tracker to a general region tracking tool by taking more motion parameters into consideration, 2) instantiating AVT to particular objects by building extensive attentional region pool for different views, and 3) exploring property selection, *e.g.*, color, shape, and size, of attentional regions.

Acknowledgments

This work was supported in part by National Science Foundation Grants IIS-0347877 and IIS-0308222.

References

[1] A. Adam, E. Rivlin, and I. Shimshoni. Robust fragments-based tracking using the integral histogram. In *CVPR'06*, volume 1, pages 798 – 805, 2006. 2

[2] S. Avidan. Ensemble tracking. In *CVPR'05*, volume 2, pages 494 – 501, 2005. 1, 2

[3] M. J. Black and A. D. Jepson. Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. In *ECCV'96*, pages 329–342, 1996. 1

[4] A. Buchanan and A. Fitzgibbon. Interactive feature tracking using K-D tress and dynamic programming. In *CVPR'06*, volume 1, pages 626 – 633, 2006. 4

[5] R. T. Collins and Y. Liu. On-line selection of discriminative tracking features. In *ICCV'03*, volume 1, pages 346–352, 2003. 1, 2

[6] D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. In *CVPR'00*, volume 2, pages 142–149, 2000. 1, 6

[7] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley-Interscience, 1991. 5

[8] M. Datar, P. Indyk, N. Immorlica, and V. S. Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *SoCG'04*, 2004. 4, 5

[9] Z. Fan, M. Yang, Y. Wu, G. Hua, and T. Yu. Efficient optimal kernel placement for reliable visual tracking. In *CVPR'06*, volume 1, pages 658 – 665, 2006. 3

[10] B. Georgescu, I. Schimshoni, and P. Meer. Mean shift based clustering in high dimensions: a texture classification example. In *ICCV'03*, volume 1, pages 456–463, 2003. 4

[11] H. Grabner and H. Bischof. On-line boosting and vision. In *CVPR'06*, volume 1, pages 260 – 267, 2006. 1, 2

[12] K. Grauman and T. Darrell. Fast contour matching using approximate earth mover's distance. In *CVPR'04*, volume 1, pages 220 – 227, 2004. 4

[13] G. D. Hager, M. Dewan, and C. V. Stewart. Multiple kernel tracking with SSD. In *CVPR'04*, volume 1, pages 790 – 797, 2004. 1, 4

[14] P. Indyk and R. Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *STOC'98*, pages 604 – 613, 1998. 4

[15] M. Isard and A. Blake. Contour tracking by stochastic propagation of conditional density. In *ECCV'96*, pages 343–356, 1996. 1

[16] A. Jepson, D. Fleet, and T. El-Maraghi. Robust online appearance models for visual tracking. In *CVPR'01*, volume 1, pages 415–422, 2001. 1, 2

[17] S. E. Palmer. *Vision Science: Photons to Phenomenology*. The MIT Press, Cambridge, Massachusetts, 1999. 2

[18] H. E. Pashler. *The Psychology of Attention*. The MIT Press, Cambridge, Massachusetts, 1998. 2, 5

[19] F. Porikli. Integral histogram: a fast way to extract histograms in cartesian spaces. In *CVPR'05*, volume 1, pages 829 – 836, 2005. 3

[20] K. Toyama and A. Blake. Probabilistic tracking in a metric space. In *ICCV'01*, volume 2, pages 50–57, 2001. 1

Figure 9. Tracking [Marathon] for frame #1, 33, 48, 75 and 84, (1st row) AVT tracker (N=40), and (2nd row) Mean-shift tracker.

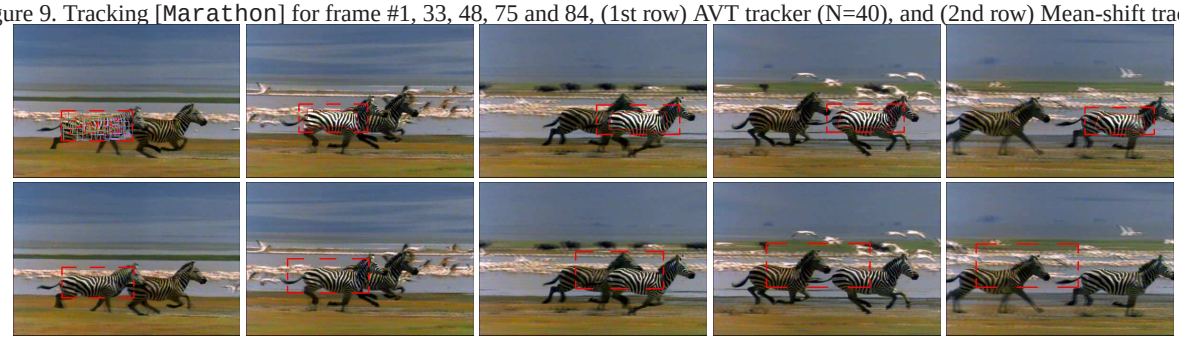


Figure 10. Tracking [Zebra] for frame #1, 63, 118, 136 and 160, (1st) AVT tracker (N=57), and (2nd row) Mean-shift tracker.

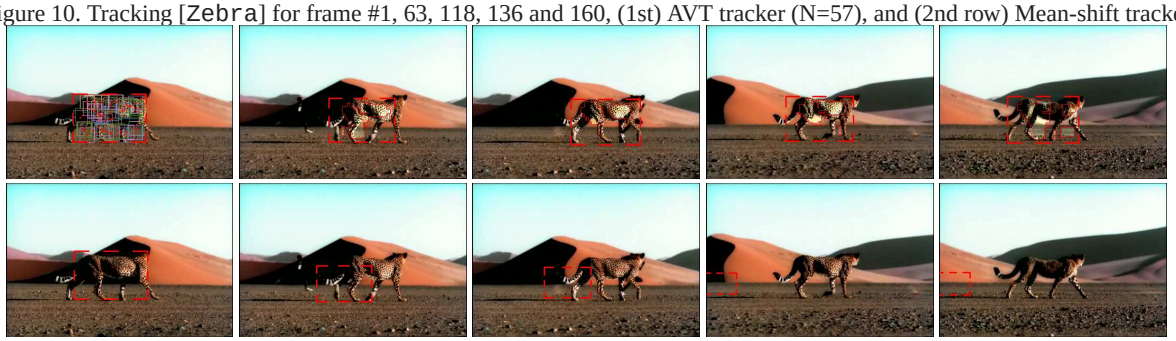


Figure 11. Tracking [Cheetah] for frame #1, 50, 80, 130, and 185, (1st) AVT tracker (N=57), and (2nd row) Mean-shift tracker.

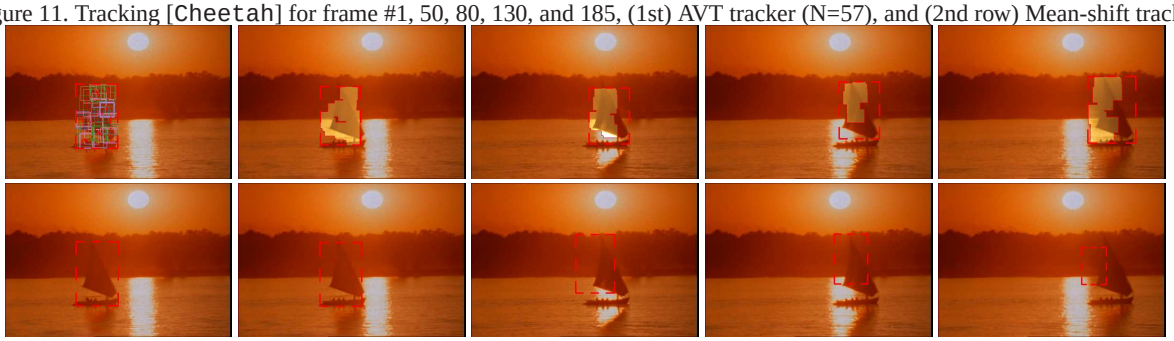


Figure 12. Tracking [Boat] for frame #1, 20, 60, 80 and 110 (1st), AVT tracker (N=56), and (2nd row) Mean-shift tracker.

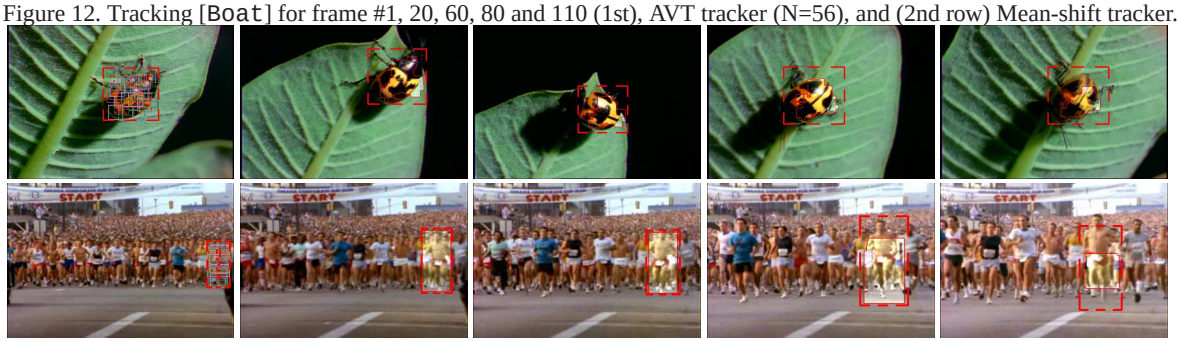


Figure 13. More AVT results: [Bug] for frame #1, 50, 86, 112 and 140 (N=59); [Marathon2] for frame #1, 64, 90, 121 and 150 (N=21) .