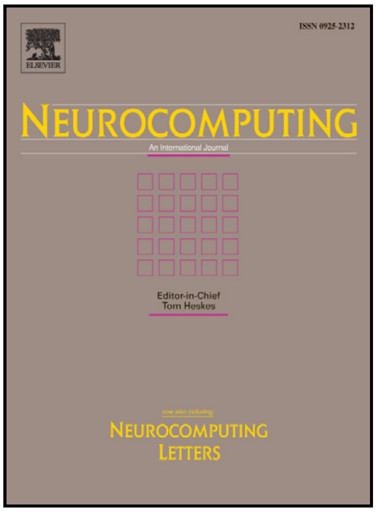
Author's Accepted Manuscript



www.elsevier.com/locate/neucom

Robust visual tracking using structural region hierarchy and graph matching

Yi-Zhe Song, Chuan Li, Liang Wang, Peter Hall, Peiyi Shen

PII: S0925-2312(12)00195-6
DOI: doi:10.1016/j.neucom.2011.11.030
Reference: NEUCOM12640

To appear in: *Neurocomputing*

Received date: 23 June 2011
Revised date: 15 November 2011
Accepted date: 19 November 2011

Robust Visual Tracking using Structural Region Hierarchy and Graph Matching

Yi-Zhe Song[a,b], Chuan Li[b], Liang Wang[c], Peter Hall[b], Peiyi Shen[d]

[a]*School of Electronic Engineering and Computer Science, Queen Mary, University of London, London E1 4NS*
[b]*Department of Computer Science, University of Bath, Bath, BA2 7AY, UK*
[c]*National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences*
[d]*XiDian University, ShaanXi, P.R.China, 710071*

Abstract

Visual tracking aims to match objects of interest in consecutive video frames. This paper proposes a novel and robust algorithm to address the problem of object tracking. To this end, we investigate the fusion of state-of-the-art image segmentation hierarchies and graph matching. More specifically: (i) we represent the object to be tracked using a hierarchy of regions, each of which is described with a combined feature set of SIFT descriptors and color histograms; (ii) we formulate the tracking process as a graph matching problem, which is solved by minimizing an energy function incorporating appearance and geometry contexts; and (iii) more importantly, an effective graph updating mechanism is proposed to adapt to the object changes over time for ensuring the tracking robustness. Experiments are carried out on several challenging sequences and results show that our method performs well in terms of object tracking, even in the presence of variations of scale and illumination, moving camera, occlusion, and background clutter.

Keywords: Tracking, Image Hierarchy, Graph Matching

1. Introduction

Fast and reliable object tracking is a prerequisite for a variety of vision applications such as monitoring and surveillance, robotic navigation, human computer interaction, etc. Although it has been one of the most active research topics over the past three decades in the field of computer vision, visual tracking remains a difficult problem due to a number of challenging factors, e.g., noise, occlusion, varying viewpoints, background clutter, illumination changes, and object appearance changes due to motion and articulation. There are numerous frameworks for tracking in the recent literature. For a comprehensive survey, see [1, 2].

In this paper, we are interested in investigating the effective fusion of structural region hierarchy and graph matching for the object representation and tracking task. Our work is motivated by recent developments in hierarchical image representations [3, 4, 5, 6] and their success in computer vision. In particular, a very latest study on filtering hierarchical image description [7] produces hierarchies that are not only accurate in terms of structural decompositions, but also manageable in terms of computational complexity. In summary, we treat the object to be tracked as a structural hierarchy of regions, and represent it as a relational graph. Each node (region) in the graph is augmented with an associated feature set consisting of local invariant features and color histograms, which is expected to be stable to cope with significant amount of image variability (such as appearance changes due to viewpoint and lighting). The tracking problem can be accordingly cast as a graph correspondence process. An adaptive graph updating scheme is also proposed to account for object appearance variation and deformation. Experimental results on several challenging sequences demonstrate the effectiveness of our method.

Major contributions of this paper can be summarized as follows: i) We exploit the use of structural region hierarchy for the tracking problem. Although the representation of hierarchical regions has been widely studied and used in image segmentation, to the best of our knowledge, it seems that object tracking using structural region hierarchy has not yet been explored. ii) We propose an effective combination of image hierarchy description and graph matching, which leads to a robust object tracking algorithm. We show how this general principle can be effectively implemented in a graph matching and updating framework. iii) We perform a series of experiments on real-world videos, from which it is demonstrated that our method obtains both robustness to handle different challenging factors and flexibility to track different types of objects.

The remainder of this paper is organized as follows. Section 2 briefly reviews related work. The proposed tracking method is described in Section 3. Experiments and results are provided and analyzed in Section 4, prior to discussion and conclusion in Section 5.

2. Related Work

In this section, we restrict to three categories of recent methods related to our research, which are briefly reviewed as follows, respectively.

Object Representation and Tracking: Visual tracking is to continuously determine the position of the object in an image sequence against dynamic scenes [1]. For example, tracking is cast as a sparse approximation problem in a particle filter framework, through a set of trivial templates [8]. In [9], a probabilistic framework is proposed for object tracking using a bag-of-pixels representation and a combination of a rigid registration between frames, a segmentation and online appearance learning.

There are two general ways to represent the objects: appearance-based (e.g., color histogram [10]) and shape-based (e.g., contour [11]). In [10], object appearance is modeled with a mixture of a fixed number of color-spatial Gaussians. In contour tracking, the object is represented as a contour, whose shape and location's evolution over time can be recovered by level sets [11] for example. Appearance-based representations often ignore structural information of the object, while shape-based ones neglect the object appearance. Neither of them is consistently satisfactory in the presence of dramatic intensity or color changes, object deformation and background clutter.

Feature Detectors and Local Descriptors: Detection of interest points and local invariant features constitute the basis for some vision tasks such as object recognition [12] and stereo matching [13]. Most algorithms start with interest point detection, followed by the local descriptor computation.

There have been several methods for interest point detection, e.g., difference-of-Gaussian [12], maximally stable extremal regions (MSER) [13], and Harris-Affine and Hessian-Affine corners [14]. The descriptors are usually designed to be invariant to lighting, scale and rotation changes, e.g., very popular SIFT [12]. See [15] for a comparative study on different descriptors. Recently, Grabner et al. [16] propose to speed up the computation of SIFT using integral images. As a variant of SIFT, SURF [17] shares its distinctiveness and robustness, but has much faster computation speed.

Image Representation via Structure Hierarchy: Hierarchical image descriptions have been recognized as being valuable. As a well established example, the work of [4] uses morphological operators to generate a tree rooted around gray level extrema. Paris and Durand [6] use the notion of stability in feature space to produce a hierarchal description. The connected segmentation tree (CST) [5] takes into account the photometric properties, spatial organization and object structure to yield semantically meaningful hierarchies. The relatively successful boundary detector to date is the probability of boundary (Pb) maps [18]. The latter, i.e., global-Pb, leads to the state-of-the-art region hierarchies [3].

Ideally, hierarchies[1] reflect assemblies that comprise real objects, but in practice they can often be very large and complex. Some studies are carried out to simplify hierarchies, e.g., MSER simplifies a hierarchy in which thresholds make levels [13]. Recently, Song et al. [7] find semantic structures in image hierarchies using Laplacian graph energy as the complexity measure, which not only retains the semantics of the hierarchies but reduces their complexities by an order of magnitude.

Tracking using Local Feature Descriptors: Detection of interest points and local invariant features constitute the basis for some vision tasks

[1]Please note that the structure hierarchy herein means a representation based on hierarchical tree of image regions, which should be differentiated from the hierarchical representation of human body structure (e.g., body parts) in [19] and the hierarchy of multi-scale or multi-stage data representation and analysis strategy [20].

such as object recognition [12] and stereo matching [13]. The descriptors are usually designed to be invariant to lighting, scale and rotation changes, e.g., very popular SIFT [12]. See [15] for a comparative study on different descriptors. Recently, several papers exploit the use of local invariant features in the tracking problem. Ta et al. [21] propose an algorithm for continuous image recognition and feature descriptor tracking. An integrated SIFT-based mean shift algorithm is presented in [22]. In [23], tracking is based on motion analysis of regional affine invariant features, and the object occupancy map is updated according to the pixel motion consistency.

In addition to the use of local features, several algorithms explore their relationships for the tracking task. Tang and Tao [24] present an attributed relational graph (ARG) that incorporates distinctive SIFT features and their relations for object tracking. In [25], a generative model consisting of consistent and random components is used to depict the relationship between local feature motions and object global motion. The latter is estimated in term of maximum likelihood of local SURF feature observations.

Our work also benefits from modeling objects in terms of attributed relational graphs. However, our graphs are built directly from structural region hierarchies other than that made from local features themselves [24, 25], and hence are able to capture regional photometric and geometric properties to facilitate robust tracking via a novel graph matching and updating mechanism. More specifically, 1) Our object representation is a hierarchy of regions, each of which is denoted as a combined feature set of SIFT descriptors and color histograms, and our relational graph is constructed directly from such hierarchy. However, in [24], observations in each frame are only the extracted SIFT features, which are used as vertices to generate attributed relational graph. In addition to different graph formation and updating manners, correlation-based ARG matching in [24] could be far from robust compared with our graph correspondence method. 2) The component tree is used in [26], but it is much different from our hierarchical tree of regions, in terms of both construction manners and semantics of leaves. Our tracking is implemented via graph matching of hierarchy-derived graphs. In [26], to identify the best fit to the input MSER, feature vectors (that are built for each of the extremal regions of the component tree) are compared to choose the region with the smallest weighted Euclidean distance as the tracked MSER. 3) For [25], individual SURF features based graph matching is used to estimate affine motion for constraining the search region. However, ours is based on feature set correspondences which results directly in tracking.

3. The Method

Our tracking method consists of two major components: building hierarchical object descriptions and tracking via graph matching and updating. Figure 2 offers a diagram of the proposed tracker, whose sub-components will be discussed in detail in Sections 3.1 and 3.2.

3.1. Object Representation via Structural Region Hierarchy

From Contours to Structural Hierarchies: At first, the method described in [3] is chosen to construct a hierarchy of regions which works on the output of any contour detector. In this paper, we use the state-of-the-art 'global-Pb' contour detector [18] as its input. Briefly, Arbelaez et al. [3] used oriented watershed transform (OWT) on the topographic surface

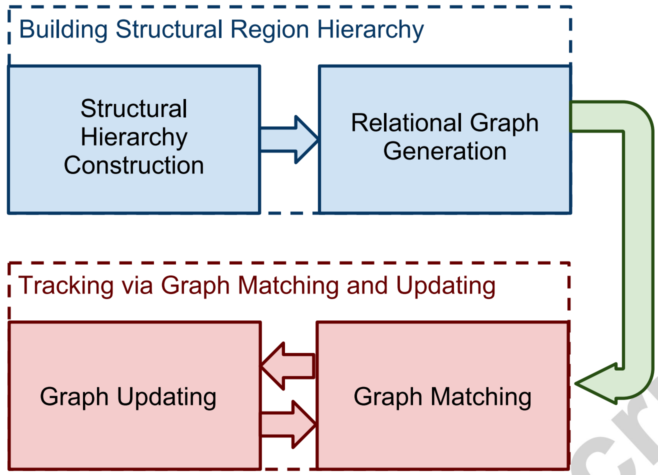


Figure 1: A system diagram of the proposed tracking framework

defined by $O(x, y) = \max_\theta O(x, y, \theta)$, where $O(x, y, \theta)$ is the 'global-Pb' contour response. The catchment basins of the minima $\mathcal{P}_0$ provide the region of the finest partition and corresponding watershed arcs $\mathcal{K}_0$. The strength of the boundaries $O(x, y, \theta)$ is then transferred to the locations $\mathcal{K}_0$ by approximating the watershed arcs with line segments and weighting each point in $\mathcal{K}_0$ by the value of $O(x, y, \theta)$ in direction θ determined by the orientation of the corresponding line segments. Afterwards, the region hierarchy is built by a greedy graph-based region merging algorithm which merges the most similar regions at one time. Figure 2(c) shows an example of the hierarchy, which is visualized as an untrametric contour map (UCM) image obtained by weighting each boundary between two regions by its scale of disappearance. Locally, regional features describe the object parts' details; globally, the relations between regions encode the object structure. It is anticipated that such a hierarchical representation allows parts of the object to have different motions, hence is more flexible to represent objects and to handle

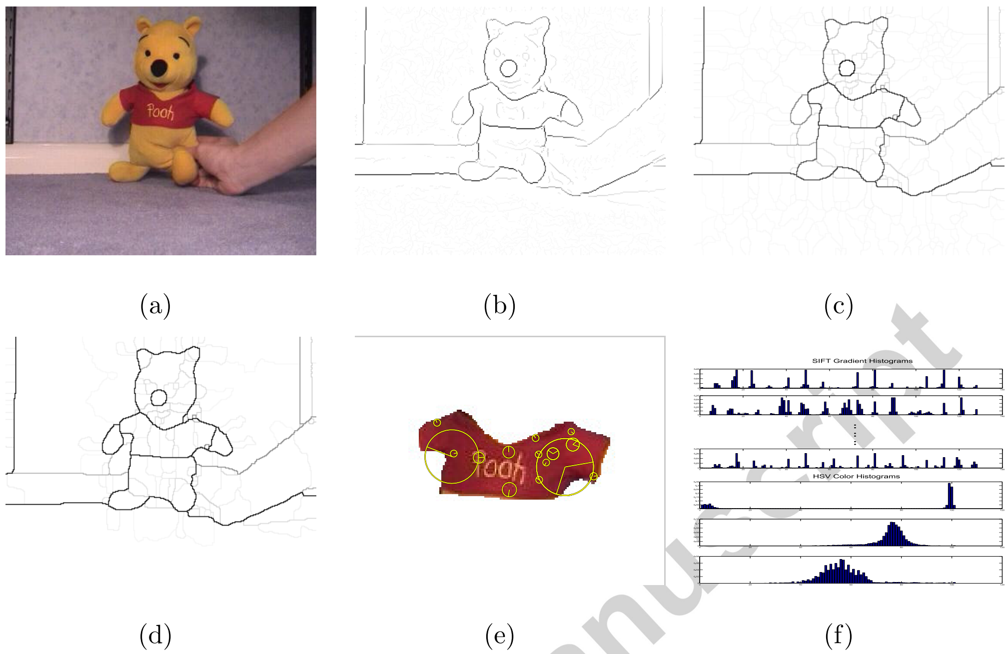


Figure 2: An example illustrating the process of ‘from image to hierarchy of regions’: a) Original image, b) Pb contours, c) original hierarchy of regions, d) filtered hierarchy of regions, e) an example region imposed by the detected SIFT points, and f) its SIFT and HSV histogram features.

object deformation and appearance changes.

Unfortunately, the hierarchy produced by [3] contains far more data than is required for an efficient description, which makes it impractical in real applications such as object tracking. There are significant practical advantages to simplify hierarchical descriptions, e.g., gains in memory efficiency and computational speed, which are all important factors for fast tracking. This work uses the method described in [7] to filter complex hierarchies into simpler ones. The reduced hierarchies preserve the semantic interpretation in terms of objects and object parts; the number of levels of the reduces hierarchies are typically an order of magnitude less than the original.

The principle in solving the problem of filtering hierarchy is to choose those levels that are lower in complexity than their neighbors. The Laplacian graph energy [27] is used to measure the complexity. Let G be a graph of n vertices and m edges, i.e., a (n, m)-graph. Let A and D be its adjacency matrix and corresponding degree matrix. Then $L = D - A$ is the graph Laplacian. The Laplacian graph energy is defined by

$$\mathcal{LE}(G) = \sum_{j=1}^{n} |\mu_j - 2m/n|$$

where μ_j is the eigenvalues of L and $2m/n$ is the average vertex degree. In [7], the affinity matrix is specially defined as $A = \{a_{ij} | a_{ij} = \exp(-w_{ij}/w_{max})\}$ where w_{ij} is the average boundary strength between regions i and j, and w_{max} is a decay factor set to the maximum over all w_{ij}. Another extension, called the *component-wise* Laplacian graph energy (cLGE) is introduced in [7]. For a graph with k disconnected components, the cLGE is defined as

$$c\mathcal{LE}(G) = K \sum_{i=1}^{k} \frac{\mathcal{LE}(G_i)}{|n_i|}$$

in which G_i is ith connected component (or sub-graph) of $|n_i|$ nodes, and K is the number of nodes in the whole graph. The cLGE at every level in the hierarchy is computed independently using graphs built from the primitives at the lowest level. At the bottom level of the hierarchy, each primitive is an 1-node sub-graph on its own, whereas the top level forms a single connected graph. At intermediate levels, as segmentations become coarser, subgraphs are merged to create larger ones, and so the number of disconnected components will fall. cLGE for the level as a whole can rise or fall, depending on the way these primitives are connected. So only those levels, at which cLGE is locally minimal, are kept in the filtered hierarchy [7]. See Figure 2(d) for an example.

From Structural Hierarchy to Relational Graph: Graphs are a versatile and flexible representation formalism useful for a range of problems in information processing. We finish building structural regional hierarchies by turning them into attributed relational graphs G'. We first remove possibly repetitive nodes residing on different levels from the filtered hierarchy via active search, and the resulting tree is still a rooted connected tree, whose nodes represent connected regions within the input and edges define an inclusion relationship between connected regions. Then, for the links between a parent and multiple children, we assign the weights with the corresponding size ratios between the parent and children nodes. It is also assumed that only children of a common parent can be adjacent, similar to the CST [5]. Accordingly, we assign the weight to each link between intra-level nodes as their similarity.

For each node (region) of the graph, we augment it with a feature vector containing its local geometric and photometric information. In particular, we describe each region with a SIFT descriptor [12, 17, 15] which achieved great success in recent applications due to their appealing characteristics. In addition, a color histogram is also used to encode regional photometric content, a notion that has been widely used in object representation [10]. Each SIFT [12] feature is described as $\mathbf{f}_s = \{p, s, o, \mathbf{h}_g\}$, where p is the 2D position of the key point, s is the scale, o is the orientation of the main gradient within the local region, and $\mathbf{h}_g$ is the gradient orientation distribution quantized into 128 bins. For color histograms, we use the HSV color space and quantize each channel into 100 bins, leading to a concatenated 300-dimensional feature vector $\mathbf{f}_c = \{\mathbf{h}_h, \mathbf{h}_s, \mathbf{h}_v\}$. For the sake of efficiency, all SIFT features and color histograms are computed incrementally during creation of the hierarchy.

3.2. Tracking via Graph Matching and Updating

Graph Matching: We formulate the tracking process into a graph matching and updating scheme. Given the graph representation G' of the object to be tracked in current frame t and the (potentially larger) candidate graph representation G'' in next frame $t-1$, the task of object tracking is naturally cast as a graph matching problem, i.e., $\mathcal{M} : G' \rightarrow G''$. In this paper, we borrow the idea from the method in [28]. For description convenience, let P' and P'' be the sets of features to be matched (in our case, 'features' mean regional nodes in G' and G''). Let $R \subseteq P' \times P''$ the set of potential assignments. A matching configuration between the two sets can be denoted as a binary vector $\mathbf{x} \in \{0,1\}^R$. A correspondence $a \in R$ indexes an entry x_a in $\mathbf{x}$, and it is active if $x_a = 1$ and inactive otherwise. An energy function $E(\mathbf{x})$ is defined to formulate the matching task as minimization of $E(\mathbf{x})$. The uniqueness constraint is enforced via the set M

$$M = \{\mathbf{x} \in \{0,1\}^R | \sum_{a \in R(p)} x_a \leq 1, \forall p \in P\} \tag{1}$$

where $P = P' \cup P''$ and $R(p)$ is the set of correspondences involving feature p. The problem is to find a configuration $\mathbf{x}$ that can minimize $E(\mathbf{x})$. In our work, we define our energy as a weighted sum of only two energy terms corresponding to appearance and geometry properties:

$$E(\mathbf{x}) = \lambda^{app} E^{app}(\mathbf{x}) + \lambda^{geo} E^{geo}(\mathbf{x}). \tag{2}$$

The term $E^{app}(\mathbf{x})$ favors correspondences between features having similar appearance, which is defined as the sum of unary terms: $E^{app}(\mathbf{x}) = \sum_{a \in R} \theta_a^{app} x_a$. For an assignment $a = (p', p'') \in R$, θ_a^{app} is the distance between appearance descriptors computed at p' and p'' in the respective sets. In this work, Chamfer distance [29] is used to compute the dissimilarity between SIFT-features sets. Assume that two regional feature sets are $\mathcal{U} = \{\mathbf{u}_i\}_{i=1}^{N_f}$ and $\mathcal{V} = \{\mathbf{v}_j\}_{j=1}^{M_f}$, where N_f and M_f denote the number of SIFT features in the two associated regions, the symmetric Chamfer distance is defined as

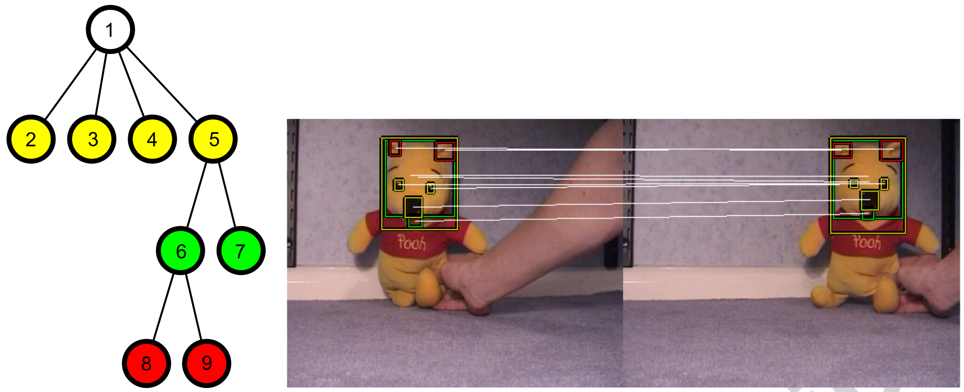
$$d_{cham}(\mathcal{U}, \mathcal{V}) = \frac{1}{N_f} \sum_{\mathbf{u}_i \in \mathcal{U}} \min_{\mathbf{v}_j \in \mathcal{V}} \|\mathbf{u}_i - \mathbf{v}_j\| + \frac{1}{M_f} \sum_{\mathbf{v}_j \in \mathcal{V}} \min_{\mathbf{u}_i \in \mathcal{U}} \|\mathbf{v}_j - \mathbf{u}_i\|.$$

For color histogram features of the two regions, we compute their dissimilarity using the χ^2 distance, i.e.,

$$d_{\chi^2}(\mathbf{h}_1, \mathbf{h}_2) = \frac{1}{2} \sum_i \frac{(\mathbf{h}_1(i) - \mathbf{h}_2(i))^2}{\mathbf{h}_1(i) + \mathbf{h}_2(i) + \epsilon} \tag{3}$$

where the introduction of a non-zero constant ϵ is just to avoid "division by zero" in practice. A weighted sum of normalized d_{cham} and d_{χ^2} is used as the appearance energy function.

The term $E^{geo}(\mathbf{x})$ measures geometric compatibility of correspondences only for neighboring features, which is defined as $E^{geo}(\mathbf{x}) = \sum_{(a,b) \in \mathrm{N}} \theta_{ab}^{geo} x_a x_b$, where the constraint set N is a neighbor system consisting of all correspondence pairs defined over neighboring features. This geometry energy term is computed using the same method described [28]. Feature correspondence is



(a) (b)

Figure 3: An example of graph matching: a) illustration to the hierarchy tree of regions, and b) the matched regions between two frames, where the colored nodes in (a) correspond to the rectangle regions with the same color.

finally written as

$$\min_{\mathbf{x}} E(\mathbf{x}|\bar{\theta}) = \sum_{a \in R} \bar{\theta}_a x_a + \sum_{(a,b) \in \mathrm{N}} \bar{\theta}_{ab} x_a x_b \tag{4}$$

which is naturally referred to as graph matching if features P' and P'' are viewed as vertices of the two graphs. Pairwise term $\bar{\theta}_{ab} x_a x_b$ with $a = (p', p'')$, $b = (q', q'')$ encodes compatibility between edges (p', q'), (p'', q'') of the first and second graph, respectively, while unary term $\bar{\theta}_a x_a$ measures similarity between vertices p', p''. We adopt the dual decomposition approach for this optimization problem. See [28] for more details. Figure 3 shows an example of graph matching of the hierarchy tree of regions, where each region is plotted as its corresponding bounding box for simplicity.

Graph Updating Mechanism: One of the key problems in tracking is how to dynamically update the object model to accommodate the object's appearance and structure changes over time. We update the object graph

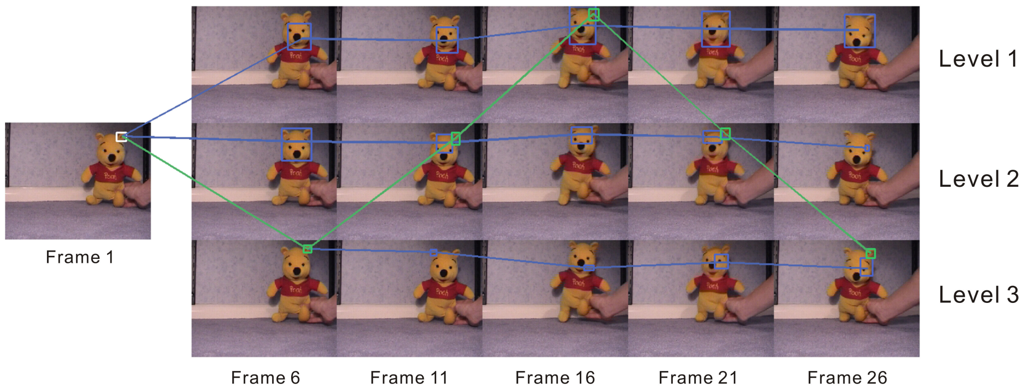


Figure 4: Tracking independent levels in the structural hierarchy (blue line) produces inaccurate results; robustness of tracking is improved by tracking structural region hierarchy instead (green line).

model using a collection of hierarchy of regions and their regional features. Regions with incompatible motion are discarded, while newly appearing object regions are incorporated into the graph representation. In particular, forward and backward matching are combined to improve the tracking robustness.

To track a video, we first build a graph G for each frame. The target object, g_1, is then initialized as a subgraph of G_1 in the first frame. We set 'key frames' for every k frame, and name the other frames the "intermediate history frames". Given g_t for key frame t, its updated graph in the next key frame contains two parts: g^f_{t+k}, a forward graph, g_{t+k}, computed from matching g_t to G_{t+k}; and g^b_{t+k}, a backward graph computed from a sequence of matchings from G_{t+k} to the intermediate history frames $[G_{t+1}, G_{t+2}, ..., G_{t+k-1}]$.

In details, we first perform a forward matching from g_t to G_{t+k} and use

RANSAC to reject outliers. The resulting forward graph g^f_{t+k} only contains region features that already exist in frame t. To incorporate potentially new object regions, we first define a candidate graph g^n_{t+k} that is connected by regions within a short distance from g^f_{t+k}:

$$g^n_{t+k} = \forall r \in G_{t+k} : min(|r - g^f_{t+k}|) < \gamma \tag{5}$$

A sequence of backward matchings is then performed between g^n_{t+k} and $[G_{t+1}, G_{t+2}, ..., G_{t+k-1}]$. The confidence of a candidate region to be a true new object region in g^n_{t+k} is evaluated by the number of times it is successfully matched. Regions of high matching score are incorporated into the final backward graph g^b_{t+k}:

$$g^b_{t+k} = \forall r \in g^n_{t+k} : (\sum_{i=1}^{k-1} Score(r, G_{t+i})) > \lambda \tag{6}$$

The final updated graph g_{t+k} is the union of g^f_{t+k} and g^b_{t+k}. In practice, γ and λ are parameters that control the generosity of introducing new object regions into the graph. We set $k = 5$ to select the key frames. This value works well for most videos and can be tuned for particularly slowly or fast moving objects. In the updated graph, an edge is only kept if both the parent and the child nodes are successfully matched. This means that g_{t+k} is not necessarily restricted to a complete subgraph of the original filtered hierarchy and allows the existence of "isolated" nodes. In order to handle occlusion, we also temporarily cease the updating procedure when forward mapping rate falls below a threshold. Our global graph matching paradigm naturally discovers the object when it reappears.

We use default parameters as in [28], except for the λ (Equation 6), which is a new parameter introduced here and was set to be 2 in our experiment, which means a new feature can only be added into the template if it is matched to at least two of the five intermediate historic frames. For RANSAC, we use the affine model as it provides efficient versatility in transformation without introducing unnecessary complexity.

4. Experiments and Results

In this section, we first provide a simple example illustrating the benefits and effects of tracking using structural region hierarchies. We then concentrate on evaluating our tracker against general challenging tracking problems such as illumination change, occlusion, moving background and so on — a setup that was also used in the two state-of-the-art trackers we compare with.

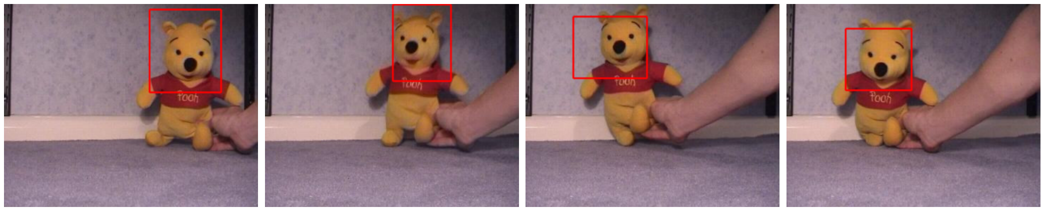
In Figure 4, we illustrate different tracking results of the highlighted region (left ear of the bear) in frame 1: when it is tracked as a node in three separate planar relational graphs (blue trajectories, one for each level) and when it is tracked within a structural region hierarchy as proposed. It shows that tracking independent levels of the structural hierarchy yields incorrect results. First, higher level graphs, such as level 1 and 2, are usually composed of large regions. The consequence is small objects, such as the ear of the bear, will be lost of track because of not being detected in the first place. Second, even on the lower level graphs (level 3) tracking can still fail because the structural information is not sufficiently used to distinguish regions of similar appearance. For example, on level 3 the tracker lost track from frame 11 onwards, where interestingly the left ear was matched to the right one instead. However, the structural hierarchy encodes richer information and it is able to pick out the correct matches across different levels.

In order to evaluate the general tracking performance of our method, several challenging sequences were chosen to carry out our experiments. These sequences encapsulate moving objects of different types such as rigid cars and articulated humans. In particular, they were recorded in indoor and outdoor environments, including significant variations with respect to scale, illumination, moving camera, occlusion, pose and cluttered background. For each sequence, we manually specified the tracking area, and plotted the global object tracking results and local region tracking results using yellow and white bounding boxes respectively.

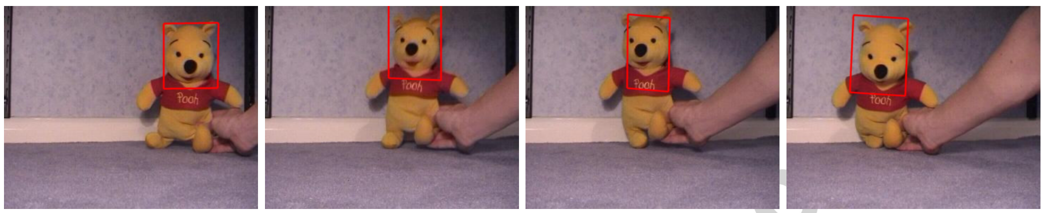
We also compared our method with two state-of-the-art algorithms described in [30] and [8]. The former [30] models the constantly changing foreground shape as a small number of rectangular blocks with histogram-based appearance representation, whose positions within the tracking window are adaptively determined. The l_1 tracker [8] describes each target candidate as sparse representation in the space spanned by target and trivial templates, and tracking is continued using a Bayesian state inference framework. Apart from the first (the example Winne video used throughout), all sequences are taken from VISOR[2]. For both of the two algorithms, we directly used their published codes with default parameter settings.

The first sequence includes a hand-held moving Winne toy. It exhibits variations in terms of scale, rotation and perspective effects. The tracking

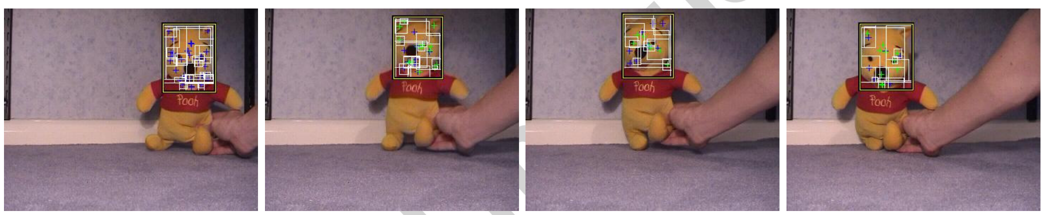
[2]http://imagelab.ing.unimore.it/visor/video_categories.asp



(a) Nejhum et al. 2008 [30]



(b) Mei and Ling 2009 [8]

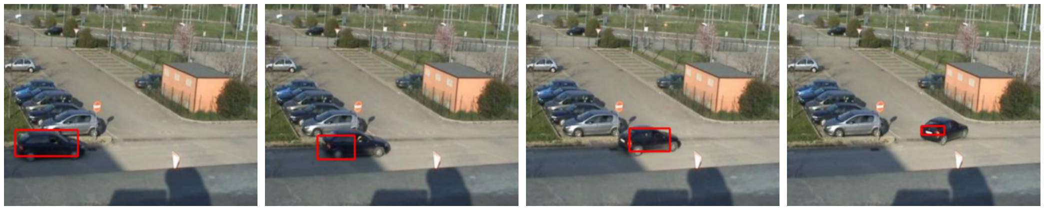


(c) Our method

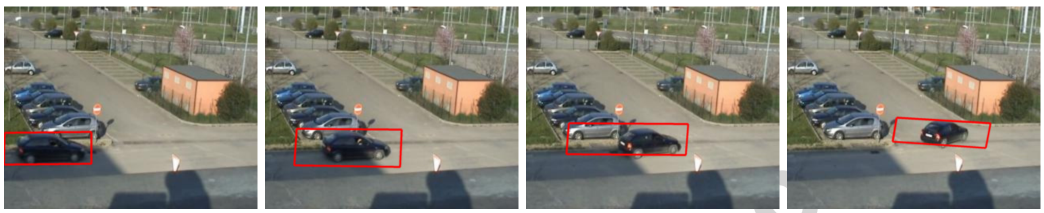
Figure 5: Tracking results on the 'Winne' sequence

results of several important frames are shown in Figure 5, together with corresponding results of the two compared methods. For this relatively simple sequence, all three tracking results are satisfactory, however both the l_1 algorithm and the method of Nejhum et al. [30] tend to include more background regions over time.

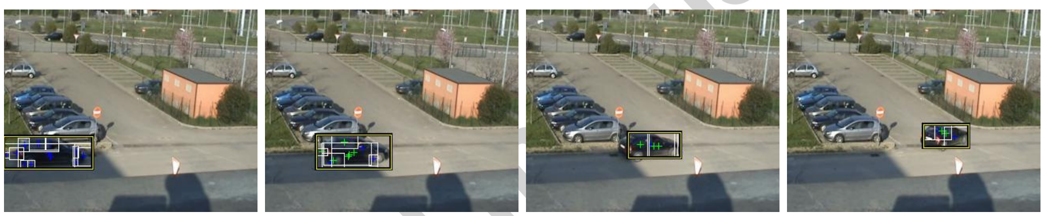
The second sequence exhibits a car driving away from a shadowed area. It mainly includes variations in terms of scale (due to distance traveled) and illumination (from shadow to shiny). Similarly, tracking results of four key frames are provided in Figure 6, together with the corresponding tracking



(a) Nejhum et al. 2008 [30]



(b) Mei and Ling 2009 [8]



(c) Our method

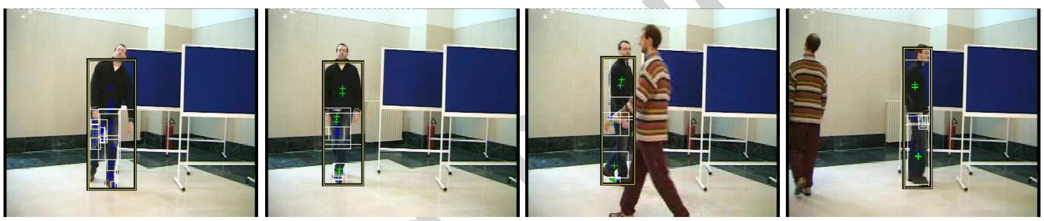
Figure 6: Tracking results on the 'Car' sequence

results using the other two trackers. In this case, all three trackers follows the general motion of the tracked object. However, the Nejhum et al. algorithm and the l_1 method tend to drift away, reflected by either shrunk or enlarged templates. Clearly, our method performs best in terms of its tracking accuracy.

The third sequence includes two walking subjects, whose walking patterns trigger occlusion. It is mainly used to examine the influence of occlusion and pose changes on the tracking performance. The tracking results of several important frames are shown in Figure 7, together with corresponding results

(a) Nejhum et al. 2008 [30]

(b) Mei and Ling 2009 [8]



(c) Our method

Figure 7: Tracking results on the 'Human' sequence with occlusion

using the other two methods. Similarly, our method is still best in this case, in terms of its accuracy and robustness.

The final sequence is the most challenging. It includes several people performing different motions in a cluttered background, filmed by a moving camera. It can be used to examine the effects of moving background, clutter and occlusions on tracking. The results of representative frames are shown in Figure 8, together with corresponding results using the two compared methods. For both the l_1 and Nejhum et al. algorithms, either more backgrounds

(a) Nejhum et al. 2008 [30]

(b) Mei and Ling 2009 [8]

(c) Our method

Figure 8: Tracking results on the 'Human' sequence with moving camera

or the lost of tracking occur. However, our method does not suffer from this at all.

In terms of computational complexity of our system, the total cost K is $N_{frames} * (Cost_R + Cost_M)$, where N_{frame} is the number of frame, $Cost_R$ is the cost for eliminating outliers, and $Cost_M$ is the cost for the actual graph matching. Due to our online graph updating mechanism, $1/K$ of the cost is spent on forward matching and the rest is spent on backward matching. Since we use RANSAC to eliminate outlier matches, the associated complexity is $O(T_{iter}(m^2 * N + 2 * N))$. Here T_{iter} is the number of RANSAC iterations, m

is the degrees of freedom in the estimated motion model and N is the number of nodes in the graph. It is worth noting that graph matching is generally a NP-hard problem and the decomposition process described in [28] is used to for fast matching. Moreover, the Laplacian graph complexity analysis further reduces the number of nodes in the graph by an order of magnitude, which importantly enables our system to matching two frames within a second in practice.

In summary, our method performs best on all these four sequences, in terms of accuracy and robustness. In particular, our method can handle different challenging factors such as changes of scale and illumination, moving camera, background clutter, and occlusion. On one hand, this benefits from our hierarchical tree representation, which reflects structure and appearance information of global object and its local parts (regions); on the other hand, because the hierarchy description can be performed rapidly and reliably online, the model of the object of interest can be adaptively updated over time using our forward/background matching scheme, without suffering from the problem of drifting that causes the loss of tracked objects.

5. Discussion and Conclusion

This paper has presented a novel and robust algorithm for visual tracking. The key contribution of our method (and reason of its superior performance) is the use of structural regional hierarchies to represent objects and a novel graph matching and updating framework as the tracking mechanism.

Our tracker is robust owning to the rich underlying object descriptions encoded as RAGs, which are tracked within a graph theoretic paradigm.

In terms of efficiency, building the RAG involves the use of several commonly practiced state-of-the-art techniques, where efficient implementations are publicly available. Importantly, sizes of RAGs are reduced to a manageable level due to the hierarchy filtering technique used, which in turn largely reduced the burden of the graph matching and updating mechanism. Tracking accuracy is also improved because of the hierarchical nature of our object representations (Figure 4).

We have demonstrated the benefits of our method through a series of qualitative results on real challenging sequences. The tracker is evaluated against major challenging tracking problems such as illumination change, occlusion, moving background and so on — a setup that was also used in the two state-of-the-art trackers we compare with.

Finally, we also believe the idea of tracking RAG has important potential benefits in higher-level tasks such as object matching and recognition across videos.

Acknowledgment

This work is in part supported by the Open Projects Program of National Laboratory of Pattern Recognition, the Natural Science Basic Research Plan in Shaanxi Province of China (2010JM8005) and the NSFC Grant (61072105).

References

[1] W. M. Hu, T. N. Tan, L. Wang, S. Maybank, A survey on visual surveillance of object motion and behaviors, IEEE Transactions on Systems, Man and Cybernetics - Part C 34 (3) (2004) 334–352.

[2] H. Yang, L. Shao, F. Zheng, L. Wang, Z. Song, Recent advances and trends in visual tracking: A review, Neurocomput. 74 (2011) 3823–3831.

[3] P. Arbeláez, M. Maire, C. Fowlkes, J. Malik, From contours to regions: An empirical evaluation, in: International Conference on Computer Vision and Pattern Recognition, 2009, pp. 2294–2301.

[4] J. A. Bangham, R. W. Harvey, P. D. Ling, R. V. Aldridge, Morphological scale-space preserving transforms in many dimensions, Journal of Electronic Imaging 5 (1996) 283–299.

[5] N. Ahuja, S. Todorovic, Connected segmentation tree - a joint representation of region layout and hierarchy, in: International Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–8.

[6] S. Paris, F. Durand, A topological approach to hierarchical segmentation using mean shift, in: International Conference on Computer Vision and Pattern Recognition, 2007, pp. 1–8.

[7] Y. Z. Song, P. Arbelaez, P. M. Hall, C. Li, A. Balikai, Finding semantic structures in image hierarchies using Laplacian graph energy, in: European Conference on Computer Vision, Vol. 4, 2010, pp. 694–707.

[8] X. Mei, H. B. Ling, Robust visual tracking using l_1 minimization, in: International Conference on Computer Vision, 2009.

[9] C. Bibby, I. Reid, Robust real-time visual tracking using pixel-wise posteriors, in: European Conference on Computer Vision, 2008, pp. 831–844.

[10] H. Wang, D. Suter, K. Schindler, Effective appearance model and similarity measure for particle filtering and visual tracking, in: European Conference on Computer Vision, Vol. 3, 2006, pp. 606–618.

[11] Y. Shi, W. C. Karl, Real-time tracking using level sets, in: International Conference on Computer Vision and Pattern Recognition, Vol. 2, 2005, pp. 34–41.

[12] D. Lowe, Distinctive image features from scale-invariant key points, International Journal of Computer Vision 60 (2) (2004) 91–110.

[13] J. Matas, O. Chum, U. Martin, T. Pajdla, Robust wide baseline stereo from maximally stable extremal regions, in: British Machine Vision Conference, Vol. 1, 2002, pp. 384–393.

[14] K. Mikolajczyk, C. Schmid, Scale and affine invariant interest point detectors, International Journal of Computer Vision 60 (1) (2004) 63–86.

[15] K. Mikolajczyk, C. Schmid, Performance evaluation of local descriptors, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (10) (2005) 1615–1630.

[16] M. Grabner, H. Grabner, H. Bischof, Fast approximated SIFT, in: Asian Conference on Computer Vision, 2006, pp. 918–927.

[17] H. Bay, T. Tuytelaars, L. V. Gool, SURF: Speed up robust features, in: European Conference on Computer Vision, Vol. 1, 2006, pp. 404–417.

[18] M. Maire, P. Arbeláez, C. Fowlkes, J. Malik, Using contours to detect and localize junctions in natural images, in: International Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–8.

[19] L. Raskin, M. Rudzsky, E. Rivlin, 3D human body-part tracking and action classification using a hierarchical body model, in: British Machine Vision Conference, 2009.

[20] B. Stenger, A. Thayananthan, P. Torr, R. Cipolla, Model-based hand tracking using a hierarchical Bayesian filter, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (9) (2006) 1372–1384.

[21] D.-N. Ta, W.-C. Chen, N. Gelfand, K. Pulli, SURFTrac: Efficient tracking and continuous object recognition using local feature descriptors, in: International Conference on Computer Vision and Pattern Recognition, 2009, pp. 2937–2944.

[22] H. Zhou, Y. Yuan, C. Shi, Object tracking using SIFT features and mean shift, Computer Vision and Image Understanding 113 (3) (2009) 345–352.

[23] S. Tran, L. Davis, Robust object tracking with regional affine invariant features, in: International Conference on Computer Vision, 2007, pp. 1–8.

[24] F. Tang, H. Tao, Object tracking with dynamic feature graph, in: Joint International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005, pp. 25–32.

[25] W. He, T. Yakayoshi, H. Lu, S. Lao, SURF tracking, in: International Conference on Computer Vision, 2009, pp. 1586–1592.

[26] M. Donoser, H. Bischof, Efficient maximally stable extremal region (mser) tracking, in: International Conference on Computer Vision and Pattern Recognition, Vol. 1, 2004, pp. 553–560.

[27] I. Gutman, B. Zhou, Laplacian energy of a graph, Linear Algebra and its applications 414 (2006) 29–37.

[28] L. Torresani, V. Kolmogorov, C. Rother, Feature correspondence via graph matching: Models and global optimization, in: European Conference on Computer Vision, 2008, pp. 596–609.

[29] H. G. Barrow, J. M. tenenbaum, R. C. Bolles, H. C. Wolf, Parametric correspondence and Chamfer matching: Two new techniques for image matching, in: International Joint Conference on Artificial Intelligence, Vol. 2, 1977, pp. 659–663.

[30] S. M. S. Nejhum, J. Ho, M.-H. Yang, Visual tracking with histograms and articulating blocks, in: International Conference on Computer Vision and Pattern Recognition, 2008.

Chuan Li is a research officer in the Department of Computer Science, University of Bath. He holds a Bachelor Degree in Software Engineering (Zhejiang University, 2005) and a PhD Degree in Computer Science (University of Bath, 2010). He works at the Media Technology Research Center and the Center of Digital Entertainment, both at University of Bath.

Liang Wang received the PhD degree from the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, in 2004. From July 2004 to January 2007, he was with the Department of Computing, Imperial College London, and the Institute for Vision Systems Engineering, Monash University, respectively. Currently, he is with the Department of Computer Science, University of Bath. He has widely published in major international journals such as IEEE Transactions in Pattern Analysis and Machine Intelligence, IEEE Transactions Image Processing, and Pattern Recognition and conferences such as ICDM, ICCV, and CVPR. His main research interests include pattern recognition, computer vision, image and video processing, machine learning, data mining, etc.

*Biography of the author(s)

Peiyi Shen is a Professor in National School of Software at the XiDian Univ. He is also a member of IEEE. He was a research officer and a PhD student in the MTRC, Computer Science at the Univ. of Bath, and a research fellow in CVSSP at the Univ. of Surrey, studying and working under Prof. Philip Willis and Prof. Adrian Hilton.
He was also with Agilent Technologies in the USA, UK, Malaysia and Singapore from 2000 to 2003. He was also a postdoctoral research fellow in the School of Computing at the National Univ. of Singapore in 2000 after he took his first PhD in XIDIAN Univ. in 1999.
His research interests are in Computer Vision, Volume Visualization and ITS applications.

Peter Hall is Associate Professor in the Department of Computer Science at the University of Bath. He is also director of the Media Technology Research Centre, also at Bath. He founded to vision, video and graphics network of excellence in the UK, and has served on the executive committee of the British Machine Vision Conference since 2003. He has published extensively in Computer Vision, especially where it interfaces with Computer Graphics. More recently he is developing an interest in robotics.

*Biography of the author(s)

Yi-Zhe Song received both the B.Sc. (first class) and Ph.D. degrees in Computer Science from the Department of Computer Science, University of Bath, UK, in 2003 and 2008 respectively; prior to his doctoral studies, he obtained a Diploma (M.Sc.) degree in Computer Science from the Computer Laboratory, University of Cambridge, UK, in 2004. After his Ph.D., he continued in the same department to become a research and teaching fellow. Since 2011, he became a lecturer (assistant professor) at School of Electronic Engineering and Computer Science, Queen Mary, University of London.

*Photo of the author(s)

*Photo of the author(s)

*Photo of the author(s)

*Photo of the author(s)

*Photo of the author(s)