Using Dominant Sets for Object Tracking with Freely Moving Camera

Giovanni Gualdi*, Andrea Albarelli°, Andrea Prati**, Andrea Torsello°, Marcello Pelillo°, Rita Cucchiara*
*D.I.I. - Univ. of Modena and Reggio Emilia, Italy °D.I. - Univ. of Venice, Italy
**DI.S.M.I. - Univ. of Modena and Reggio Emilia, Italy
{giovanni.gualdi,andrea.prati,rita.cucchiara}@unimore.it, {albarelli,torsello,pelillo}@unive.it

Abstract

Object tracking with freely moving cameras is an open issue, since background information cannot be exploited for foreground segmentation, and plain feature tracking is not robust enough for target tracking, due to occlusions, distractors and object deformations. In order to deal with such challenging conditions a traditional approach, based on Camshift-like color-based features, is augmented by introducing a structural model of the object to be tracked incorporating previous knowledge about the spatial relations between the parts. Hence, an attributed graph is built on top of the features extracted from each frame and a graph matching technique based on Dominant Set clustering is used to find the optimal match with the model. Pixel-wise and object-wise comparison with other tracking techniques with respect to manually-obtained ground truth are presented.

1. Introduction

In recent years, object tracking has been recognized by the scientific community as a fundamental task in several applications of video analysis. Tracking rigid objects in simple, uncluttered scenes acquired from static cameras is an almost solved problem [26]. Conversely, in complex scenarios where objects camouflage with the background, have severe shape variations and are strongly occluded, tracking can be really challenging. In addition, when either the background is not fixed or the camera is moving, no statistical or geometrical model can be exploited to segment the foreground objects and predictive models (such as Kalman filters) are ineffective.

In point tracking, objects are usually represented by single or multiple points and the correspondences between two consecutive frames is established by either deterministic [25] or statistical methods [3] to provide tracking without object segmentation. An alternative is to represent the data using kernel primitives such as rectangles or ellipses. These kernel methods can be used to estimate a density-based appearance model of the object [7]. Other approaches encompass silhouette tracking, estimating the object contour evolution by means of state-space models [13] or variational methods [4].

These proposals are robust and efficient when the object can be represented by a single feature, such as the color histogram, but in the case of complex articulated objects represented by parts which are often partially or completely overlapped they are likely to fail. To deal with such challenging scenarios structural information expressing spatial constraint among features might be used. This is the case of the pictorial structures of [9] that have been proposed for object recognition and has been further developed for people tracking by [19]. Similarly [12, 23] are based on inference in a graphical model and can be applied again to people tracking [22]. All these approaches tend to be specifically focused on the articulated structure of the human body or human face (whereas our framework tackles generic-shape object tracking), and rely on Bayesian probabilistic frameworks; on the other hand tracking can be brought to a problem of graph matching through a graph based representation based on Region Adjacency Graph (RAG), where vertices represent image regions and edges encode adjacency. This is the case of [10, 8, 2, 11]. A notable exception is [14] where RAGs are tracked by fitting independent Kalman filters to both regions and adjacency relations. [21] uses graphs and Kalman filter for insects tracking.

Structural methods based on point features are less used than region-based ones. This is due primarily to the fact that is more difficult to define relations between point features. In [24] SIFT features are extracted from the tracked object and a nearest-neighbor graph is built on top of them. Relaxation labelling is used for matching and the object graph itself is updated by removing disappearing features and adding new ones. In [5] the features tracked are the linear borders of geometric objects and edges connect parallel or perpendicular borders.

The definition of the structural model can be inferred from the image data. This approach is very general but might suffer from the instability of the model inference, both in terms of detection of regions/features and with

respect to the invariance of the relational structure to be tracked. In addition an inferred model is inherently unable to capture detailed information about the intrinsic articulation and deformability of non-rigid objects. By contrast, our approach requires an a priori structural model of the target object (not necessarily bound to the human body figure), that is then enriched with attributes extracted from real data. This way we are able to search for a match that not only maximizes the coherence between attributes, but that also accounts for the coherence of the structural relations in a way invariant to variations in scale, rotations and translations, and even blurring due to camera motions; the search for the best coherent match with the provided model is made through Dominant Set extraction.

2. Overview of the Framework

Fig. 1 shows the conceptual scheme of our framework. An initial *Graph-Based Model Definition* provides the framework with both a model of the features to be tracked and a structural representation of their spatial arrangement. In this work color features are used, but different or more descriptive features (e.g. textures, edges) can be exploited. Moreover, an initial image can be used as reference for the extraction of the feature model (as in our current implementation), the structural model or both (Fig. 2a). Each new frame I_t is provided to the *Feature Cluster Extraction* component (Sect. 3) that applies the feature model and produces the mask of the probability of each feature class onto the current image (called *back-projection* [6]). Each back-projection is then clustered using meanshift and, for each cluster, attributes are extracted. Most of the extracted feature clusters represent erroneous detections of the tracked object feature (see Figs. 2b-f) and the correct candidates must be extracted using global consistency information.

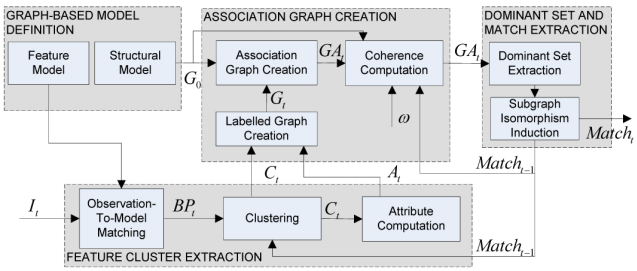

Figure 1. Scheme of the proposed framework.

A labelling function maps each feature cluster on the originating model feature. Each pair of clusters whose features are rigidly joined together in the structural model, are connected by edges to form the *labelled graph* G_t (Sect. 4.1). Then an edge weighted *association graph* GA_t is created between the structural model G_0 and the labelled graph

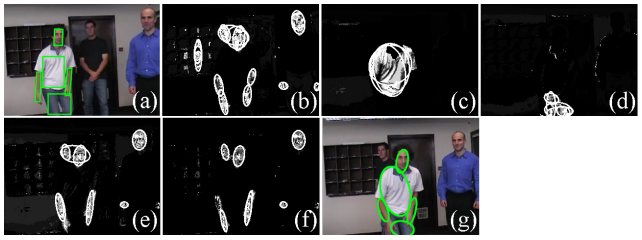

Figure 2. Framework steps example: (a) model to track, (b-f) feature back-projections and clusters, (g) best coherence match

G_t (Sect. 4.2) and each edge is weighted according to a global *coherence measure* (Sect. 4.4) in such a way that each maximal edge weight clique in GA_t corresponds to a maximal coherence subgraph isomorphism and vice versa. Finally the Dominant Set framework is used to search for the maximum coherence $Match_t$ (Sect. 4.3, Fig. 2g).

3. Extraction of Feature Clusters

For each feature of the model, the *Feature Cluster Extraction* component extracts all the possible clusters of features which might represent a part of the tracked object according to the feature model. In our work, the *Feature Cluster Extraction* operates on simple color features using a modification of the Camshift algorithm [6], but different cluster extraction algorithms can be used.

The standard Camshift tracking algorithm uses a model of the object, consisting of a color histogram, and requires a region of interest to initialize the search. For each input image a probability mask of the model is produced, evaluating each pixel according to the color histogram as if it were a pdf. The resulting value is then scaled on 256 gray levels, producing the so called back-projection. Then, iteratively alternating the meanshift gradient ascend algorithm and a size-adaptation of the region of interest, the region estimate converges to encompass the extracted features and then provides the initial location for the next frame.

For the extraction of the feature clusters, the Camshift is modified as follows. *First*, the object to be tracked is modelled with multiple color histograms, corresponding to different areas of the objects (e.g. Fig. 2a); therefore, for each input image, multiple back-projections (BP_t) are obtained (Figs. 2b-f) and the cluster extraction proceeds independently for each BP_t. *Second*, the back-projections are obtained on the following color space:

$$(h,s,v) = \begin{cases} \left(\left\lfloor \frac{H}{16} \right\rfloor, \left\lfloor \frac{S}{16} \right\rfloor, \max_V\right) & \text{if } S > \tau_S \wedge V > \tau_V \\ \left(0, 0, \left\lfloor \frac{V}{16} \right\rfloor\right) & \text{otherwise} \end{cases} \quad (1)$$

The addition of value and saturation components to the standard Camshift color space allows us to deal better with

low-saturation (considering only the V component) and provides an enriched color description. *Third*, our approach scatters particles over the BP_t from which to start the cluster extraction, producing therefore several clusters. The particles are spatially scattered over the BP_t with Gaussian or uniform distribution, depending on the object tracking status at the previous frame.

For each cluster C_t^i of the set C_t, the set of attributes $A_t^i = \left(D\left(C_t^i\right), M\left(C_t^i\right), P\left(C_t^i\right), R\left(C_t^i\right)\right)$ are computed, where D is the density, M the mass, $P = (x, y)$ the coordinates of the cluster's centroid and R the area:

$$M\left(C_t^i\right) = \sum_{p \in C_t^i} BP_t\left(p\right) \; ; R\left(C_t^i\right) = \left\|\left\{p \in C_t^i\right\}\right\|$$
$$D\left(C_t^i\right) = M\left(C_t^i\right) / R\left(C_t^i\right) \tag{2}$$

4. Tracking using relational information

Regardless of the robustness of the extraction step several factors could lead to a wrong assignment between clusters. In fact, distractors, noise, deformation or pose and illumination changes can easily lower the coherence between correct correspondences or make unrelated features more similar. For this reason any approach that is based only on the similarity between features is inherently sensitive to noise. To overcome this limitation we add contextual information, thus casting the feature matching into a more robust subgraph matching problem.

4.1. From feature clusters to labelled graphs

In order to obtain a graph from a set of feature clusters we exploit the previous knowledge about the physical structure of the object. To this end, we define a structural model where each part of the object is associated to a feature class which is known to be rigidly joined to some other parts, but can move freely from the rest. This is the case with any articulated object, while totally-rigid objects can be modeled by joining all the parts.

A *structural model* of an object is a connected graph $G_m = (P, S)$ where P is the set of distinct parts we want to use to represent the object and $S \subseteq P \times P$ are their structural relations, where $(p_a, p_b) \in S$ iff p_a and p_b are joined in the object. This model embeds our previous knowledge about the structure of the object to be tracked in terms of its parts. In Fig. 3 some examples of structural models are presented.

Given a structural model $G_m = (P, S)$, a set of features clusters C assigned to $|P|$ classes by a surjective labelling function $l : C \rightarrow P$ and their attributes A, we define the *labelled graph* as the $|P|$-partite graph $G = (C, E, A, l)$ where C is the vertex set, $E = \{(u, v) \in C \times C | (l(u), l(v)) \in S\}$ the edge set, A the vertex attributes and l the vertex labelling function. In this graph each edge represents a structural relation between a pair of feature clusters. The automatic extraction of feature cluster candidates from a frame I_t yields a graph with many nodes and edges. The supervised selection of the ground truth from a reference frame will result in a simpler graph with just one cluster for each part of the object to be tracked: we call this graph the model graph. Our goal is to find within each labelled graph extracted from a frame I_t the subgraph which is the most coherent with the model graph we are tracking. In other words we are looking for a maximum coherence subgraph isomorphism.

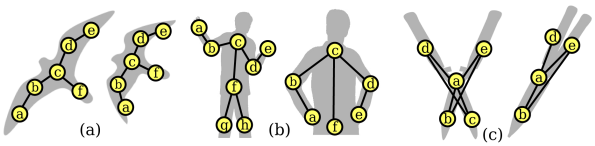

Figure 3. Example of structural models and labelled graphs. The model is subject to deformation (a) and also to scaling and occlusion (b); in (c) it comprises two totally-rigid submodels partially occluding each other

Given labelled graphs $G_1 = (C_1, E_1, A_1, l_1)$ and $G_2 = (C_2, E_2, A_2, l_2)$ a *labelled isomorphism* between them is a relation $M \subseteq C_1 \times C_2$ such that for each $(u_1, u_2), (v_1, v_2) \in M$ the following properties hold:

$$l_1(u_1) = l_2(u_2) \text{ and } l_1(v_1) = l_2(v_2) \tag{3}$$
$$u_1 = v_1 \Leftrightarrow u_2 = v_2 \tag{4}$$

The first condition ensures that M does not map feature cluster of incompatible classes. The second condition forces M to be a partial injective function. It is easy to see that any labelled isomorphism is a special case of subgraph isomorphism which enforces label consistency.

We still need to define a measure of the global coherence of a labelled isomorphism M. In our context limiting the measure to a similarity between vertex attributes would be not enough, as this way we would be unable to take into account structural relations among vertices. Unfortunately, even measuring coherence between edges would not be general enough, as it would not be possible to account for invariants that depends on more than one edge, such as length ratios or angle differences. For this reason we defined a coherence measure between pairs of edge matches as this allows us to deal with variations in scale and articulation throughout the video sequence. To this end we define the set of edge matches as:

$$e(M) = \{((u_1, v_1), (u_2, v_2)) \in E_1 \times E_2 | \\ (u_1, u_2) \in M \wedge (v_1, v_2) \in M\} \tag{5}$$

and let $\omega : (E_1 \times E_2) \times (E_1 \times E_2) \rightarrow \mathbb{R}^+$ be a measure of coherence between pairs of edges matches, then the total weight of M is defined as:

$$\Omega(M) = \sum_{a \in e(M)} \sum_{b \in e(M) \setminus \{a\}} \omega(a, b)\,. \tag{6}$$

4.2. From graph matching to clique search

In order to search for a match of maximum compatibility between two labelled graphs we choose a two-step approach which first casts the matching problem into a clique search problem and then solves it using continuous optimization.

Given labelled graphs $G_1 = (C_1, E_1, A_1, l_1)$ and $G_2 = (C_2, E_2, A_2, l_2)$ and a function $\omega : (E_1 \times E_2) \times (E_1 \times E_2) \rightarrow \mathbb{R}^+$ that measures the coherence between pairs of edge associations, we define an association graph between them as an edge weighted graph $Ga = (Va, Ea, \omega)$ where $Va = E_1 \times E_2$, $Ea \subset Va \times Va$ with $(((u_1, v_1), (u_2, v_2)), ((w_1, z_1), (w_2, z_2))) \in Ea$ iff:

$$l_1(u_1) = l_2(u_2), l_1(v_1) = l_2(v_2),$$
$$l_1(w_1) = l_2(w_2) \text{ and } l_1(z_1) = l_2(z_2) \tag{7}$$
$$u_1 = w_1 \Leftrightarrow u_2 = w_2, v_1 = z_1 \Leftrightarrow v_2 = z_2 \tag{8}$$

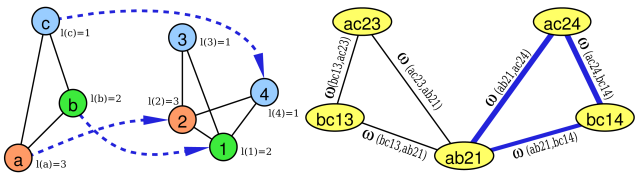

Figure 4. Labelled isomorphism between two labelled graphs and the clique associated to it in the edge weighted association graph.

With this definition we are able to show some useful connections between labelled isomorphisms and complete subgraphs (cliques) in this association graph.

To this end, note that each $X \subseteq V_a$ represents a relation between edges in E_1 and E_2. In order to obtain a relation between vertices in V_1 and V_2 we define a natural map $v : \mathcal{P}(V_a) \rightarrow \mathcal{P}(V_1 \times V_2)$ as:

$$v(X) = \{(u_1, u_2) \in V_1 \times V_2 | \\ ((u_1, v_1), (u_2, v_2)) \in X \vee ((v_1, u_1), (v_2, u_2)) \in X\} \tag{9}$$

That is, a match between vertices is induced by X if they are mapped by any edge match in X. It is easy to see that v is not injective, nevertheless it has a proper right partial inverse, namely the function $e(M)$ defined by (5).

We now formulate the following lemmas (proofs in [1]):

Lemma 1 *Given labelled graphs G_1, G_2 and their association graph G_a, $X \subseteq V_a$ is a clique iff $v(X)$ is a labelled isomorphism between G_1 and G_2.*

Lemma 2 *If $X \subseteq V_a$ is a maximal clique in G_a, then $v(X)$ is a maximal labelled isomorphism between G_1 and G_2. Conversely, if M is a maximal labelled isomorphism between G_1 and G_2 then $e(M)$ is a maximal clique in G_a.*

From the previous lemmas and the definition of the weight of a labelled isomorphism M, derives the following:

Theorem 1 *Given two feature graphs G_1 and G_2, each maximal(maximum) weight labelled isomorphism M between them induces a maximal(maximum) edge weight clique in $Ga(G_1, G_2)$ and vice versa.*

Fig. 4 shows an example of a labelled isomorphism and the correspondent clique in a labelled association graph.

4.3. An effective heuristic for the weighted clique problem

Theorem 1 casts our tracking problem into a search for a maximal edge weighted clique in a novel type of association graph. In order to perform this search we use the Dominant Set framework [17]. Given an edge weighted graph $G = (V, E, \omega)$, a subset of vertices $S \subseteq V$ and two vertices $i \in S$ and $j \notin S$, the following function measures the coherence between nodes j and i, with respect to the average coherence between node i and its neighbors in S:

$$\phi_S(i, j) = \omega(ij) - \frac{1}{|S|} \sum_{k \in S} \omega(ik) \tag{10}$$

While overall weighted coherence between i and all the nodes in S is defined as:

$$w_S(i) = \begin{cases} 1 & \text{if } |S| = 1 \\ \sum_{j \in S \setminus \{i\}} \phi_{S \setminus \{i\}}(i, j) w_{S \setminus \{i\}}(j) & \text{otherwise} \end{cases} \tag{11}$$

Intuitively, $w_S(i)$ will be high if i is highly coherent with vertices in S. Given this measure $S \subseteq V$ is said to be *dominant* if the following conditions hold:

$$w_S(i) > 0, \forall i \in S \text{ and } w_{S \cup \{i\}}(i) < 0, \forall i \notin S \tag{12}$$

The conditions above correspond to the two main properties of a cluster: namely internal homogeneity and external inhomogeneity. In the literature this framework has been associated to clustering, nevertheless its use as an heuristic for the edge weighted clique problem is justified by the fact that, when applied to unweighted graphs, the notion of a dominant set is equivalent to the notion of a clique. Hence, a dominant set can be seen as a generalization of cliques

to graphs with weighted edges. Moreover there is another compelling reason to prefer dominant sets over traditional techniques of clique search: in fact their clustering property allows us to discard automatically nodes that are less coherent with respect to the others. This is the case when a part of the model is missing or occluded. For instance in Fig. 5 the face is out of the frame border, but candidates for it are generated anyway by the back projection: in this situation an exact graph matching would wrongly include in the result also the best of those candidates (green ellipse), whereas dominant sets leave it out as its coherence is very low with respect to the other parts in the result (red ellipses). It is worth noting that this selection does not require the user to choose a threshold as it is implicit in the cluster properties.

Pavan and Pelillo [17] have shown that dominant sets correspond to local maximizer over the standard simplex of the quadratic function $f(\mathbf{x}) = \mathbf{x}^t A\mathbf{x}$ where A is the weighted adjacency matrix of the graph (thus $A_{ij} = \omega(i,j)$). These maximizers can be found by exploiting the convergence properties of the payoff monotonic replicator dynamic $x_i(t+1) = (Ax(t))_i/(x(t)^t Ax(t))$ which is guaranteed to converge to a local maximum when the association graph is undirected and, thus, the matrix A is symmetric [18]. At convergence the value of the function f is a measure of the coherence of the extracted set. This property is used to detect the absence of the object from the scene and suspend the tracking. Finally, as the local maximizer found by the replicator dynamic is not guaranteed to be the global maximum, we used an enumeration strategy similar to the one presented in [20].

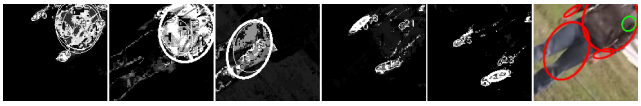

Figure 5. Example of the failing of an exact graph matching

4.4. Coherence Computation

Given the association graph $Ga_{t,0}$ between G_t and G_0, our goal is to assign to each of its edges $(((u_t, v_t), (u_0, v_0)), ((w_t, z_t), (w_0, z_0))) \in Ea_{t,0}$ a weight in the interval $[0, 1]$ which reflects the coherence between the two connected edge associations (see Fig. 4). This measure $\omega : Ea_{t,0} \rightarrow [0, 1]$ is the sum of several components, each referring to a specific property of the tracked object that should be consistent along the video sequence. Since different and independent properties are considered, the mis-detection of any of them (for example, due to occlusion or deformation) does not compromise the overall coherence evaluation. In the present work we define three properties that are expected to be consistent along the video sequence: *color* and *structure* w.r.t. the initial model, and *spatial similarity* w.r.t. the previous frame.

Color-based consistency measured through cluster density ω_d and mass ω_m: let us define the normalized density and the normalized mass respectively as:

$$ND(u_t) = \frac{D(u_t)}{\max\limits_{\forall v_t \in C_t \mid l(u_t)=l(v_t)} D(v_t)} \tag{13}$$

$$NM(u_t) = \frac{M(u_t)}{\max\limits_{\forall v_t \in C_t \mid l(u_t)=l(v_t)} M(v_t)} \tag{14}$$

then:

$$\omega_d = \sqrt[4]{ND(u_t) \cdot ND(v_t) \cdot ND(w_t) \cdot ND(z_t)} \tag{15}$$

$$\omega_m = \sqrt[4]{NM(u_t) \cdot NM(v_t) \cdot NM(w_t) \cdot NM(z_t)} \tag{16}$$

The clusters are defined over the back-projection that measures the color similarity of the image I_t compared to a color feature of the model: therefore the higher the density of a cluster, the higher its color similarity to the model. The densities of the four clusters are multiplied and not summed up in order to reinforce the overall $Ea_{t,0}$ color similarity. Since small clusters might show very high ω_d, the ω_m component reinforces only the $Ea_{t,0}$ that have strong masses.

Structure consistency measured through cluster sizes and inter-cluster distances ω_{sd}: this component reinforces the $Ea_{t,0}$ that shows structural similarity with the model, i.e. cluster size variations which are supported by consistent inter-cluster distance variations. Fig. 6 depicts three different cases. (a) is a typical structure size reduction (for example, due to camera zoom out) that maintains consistency between area and distance variations. On the other hand, (b) and (c) depict a structure deformation that is penalized by ω_{sd}: in both cases the distance variation between top and middle clusters is not supported by a similar variation in the size of the cluster; ω_{sd} is formalized introducing the *linear area ratio* and the *distance ratio* respectively as: $lar : C_t \times C_0 \rightarrow [0, \infty)$, $lar(u_t, u_0) = \sqrt{\frac{R(u_t)}{R(u_0)}}$ $dr : E_t \times E_0 \rightarrow [0, \infty)$, $dr((u_t, v_t), (u_0, v_0)) = \frac{|\overline{P(u_t)P(v_t)}|}{|\overline{P(u_0)P(v_0)}|}$. Structure consistency of $Ea_{t,0}$ is obtained when lar measures are similar to the respective dr measures, i.e. their ratio is close to 1; the consistency measure can then be obtained modelling the deviation with a Gaussian. To evenly stretch the ratio codomain from $[0, \infty)$ to $(-\infty, \infty)$, it is appropriate to compute the logarithm. Therefore, ω_{sd} is defined as follows:

$$\omega_{sd} = e^{\frac{-(Q(u)-\Delta(u,v))^2}{2\sigma^2}} \cdot e^{\frac{-(Q(v)-\Delta(u,v))^2}{2\sigma^2}} \cdot e^{\frac{-(Q(w)-\Delta(w,z))^2}{2\sigma^2}} \cdot e^{\frac{-(Q(z)-\Delta(w,z))^2}{2\sigma^2}} \tag{17}$$

where $Q(a) = \log(lar(a_t, a_0))$ and $\Delta(b,c) = \log(dr((b_t, c_t),(b_0, c_0)))$. In analogy to what is done with ω_d and ω_m, the four contributes of ω_{sd} are multiplied together and not summed up.

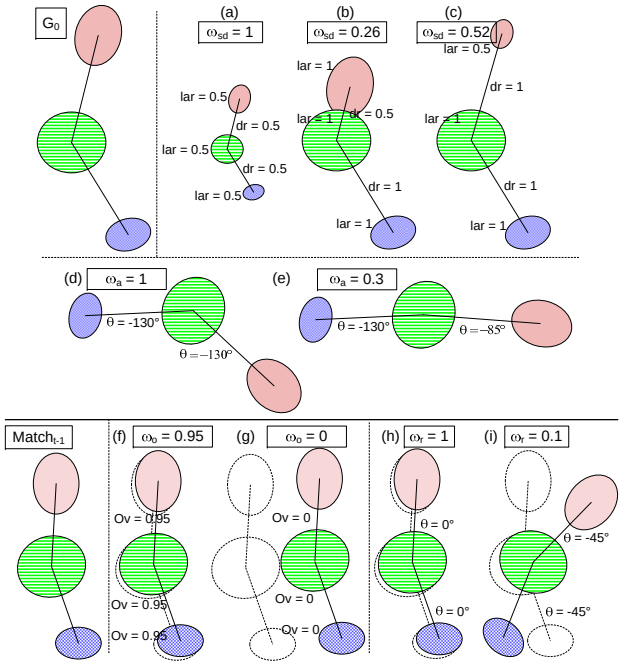

Figure 6. Structure consistency measure with ω_{sd} (a, b, c) and ω_a (d, e), and spatial similarity measure with ω_o (f, g) and ω_r (h, i). For the only sake of clarity and without loss of generality, the G_0 and $Match_{t-1}$ are made of only three nodes.

Structure consistency measured through cluster relative orientations ω_a : this component favors the maintenance of angular consistency of the $Ea_{t,0}$. Fig. 6 depicts two cases: regardless of the *overall* rotation of one graph compared to the other, (d) maintains the consistency of reciprocal angles of the segments, while (e) does not and is therefore penalized by ω_a. Let $\vartheta((u_t, v_t),(u_0, v_0)) = \arccos \frac{\overline{P(u_t)P(v_t)} \times \overline{P(u_0)P(v_0)}}{\|\overline{P(u_t)P(v_t)}\| \cdot \|\overline{P(u_0)P(v_0)}\|}$ as the angle between the two segments connecting the centroids of the clusters, the value ω_a is defined as: $\omega_a =$

$$\frac{\exp\{m \cdot \cos[\vartheta((u_t, v_t),(u_0, v_0)) - \vartheta((w_t, z_t),(w_0, z_0))]\}}{\exp\{m\}} \tag{18}$$

This resembles the Von Mises distribution [15], that is often used to model angular distributions.

Spatial similarity with the object at previous frame measured through overlap ω_o *and rotation* ω_r: let us consider the graph $Match_{t-1}$, which represents the tracked object at the previous frame, and the projection of its attributes over the graph G_0; the similarity components ω_o and ω_r favor the edges in $Ea_{t,0}$ that respectively maximize the area of overlap and minimize the overall graph rotations with respect to $Match_{t-1}$. In case $Match_{t-1}$ is partial/missing, these components will provide the contribution for the detected portion of the object only. Fig. 6 (f,g,h,i) depicts some explanatory examples. By defining $Ov(u_t, u_{t-1}) = \frac{2 \cdot R(u_t \cap u_{t-1})}{R(u_t) + R(u_{t-1})}$, we have:

$$\omega_o = \sqrt[4]{Ov(u_t, u_{t-1})\, Ov(v_t, v_{t-1})\, Ov(w_t, w_{t-1})\, Ov(z_t, z_{t-1})} \tag{19}$$

ω_r is defined to favor the minimization of the rotation of each single segment:

$$\omega_r = \frac{\exp\{m \cdot \cos[\vartheta((u_t, v_t),(u_{t-1}, v_{t-1}))]\}}{\exp\{m\}} \cdot \frac{\exp\{m \cdot \cos[\vartheta((w_t, z_t),(w_{t-1}, z_{t-1}))]\}}{\exp\{m\}} \tag{20}$$

5. Experimental Results

To demonstrate the advantage that the proposed graph based tracking (GB from now) offers with respect to existing techniques, we selected two tracking algorithms to compare with: Camshift (CS from now) and a *particle filtering* tracking based on color features (PF from now) similar to that proposed in [16]. For the sake of a fair comparison all three approaches are applied to the same color space (1).

In contrast to our approach, CS and PF do not correlate the results on the different feature models, that is, they do not exploit structure model. Therefore, we issue several independent instances of the algorithm on each single feature of the same object model used for the GB. They work well in standard conditions, but for the intrinsic limitation due to the lack of a structure model, they are likely to fail in challenging conditions, especially in the case of occlusions.

Our test bed consists of 3 videos and in one of them (Video 3) the tracking is applied twice, on two different target objects[1]. Table 1 summarizes the main characteristics of the benchmark videos.

In order to evaluate the performance of the approaches, we manually extracted the ground truth with the help of the VIPER-GT tool [2], consisting of several oriented bounding boxes, each containing a single part of the object to be tracked. Given the ground truth and the output of the tracking algorithms, it is possible to automatically compute the performance based on a set of metrics. Specifically, using the VIPER-PE tool [2], we obtained true positives, false negatives, false positives and, from them, *recall* and *precision*; these measures were extracted both at object and pixel level.

[1] Downloaded by AVSS 2007 dataset: ftp://motinas.elec.qmul.ac.uk/pub/multi_face

[2] http://viper-toolkit.sourceforge.net/

	Video 1	Video 2	Video 3 - a and b	
Generic info	Outdoor, moving cam, 1 person	Outdoor, moving cam, 2 persons	Indoor, static cam, 3 occluding people	
Model	F,T,P, LA,RA	F,T,P	F,T, H,P	F,T, LA,RA
Challenges	Severe scale vars and rotations, camouflaging background	Scene cuts, total obj. disapp, scale vars, rotations, camouflaging bkg	severe occlusions, several color distractors	

Table 1. Benchmark (F=face, T=torso, H=hands, P=pants, LA,RA= left/right arm).

The pixel-wise evaluation is shown in Fig. 7. In this case, we directly plot the frame-by-frame *F-measure* defined as an aggregation of recall R and precision P: $F = \frac{2 \cdot R \cdot P}{R+P}$. Pixel-wise recall and precision aggregate together the pixel measures performed separately on each single model class. $F = 1$ could reveal either a perfect matching (never happened in our tests), or the correct tracking suspension when the whole object is absent from the scene. Conversely, $F = 0$ reveals either a total failure or the detection of an object when this is not present.

Table 2 reports the summary of the pixel-wise and object-wise performance on the benchmark videos. Differently from the pixel-wise evaluation that merges together the pixel evaluations of all the tracked model classes, the object-wise evaluation gives a fairer evaluation on the tracking of the single classes, regardless of their pixel areas (e.g. it equally weights the tracking of a small part like a hand as the tracking of a bigger part like a torso). The three original video sequences, the four post-processed videos with our graph based tracking, the four ground truths in VIPER XML meta-data and the graph of the object-wise evaluation are web published [1]. Since video 1 does not contain severe occlusions, scene cuts or object disappearances, the structural model in our approach does not significantly increases the performance compared to CS or PF, with exception of frames 199 and 231, when the face exits from the view: in fact, our approach correctly suspends the face tracking to resume it when the face reappears, whereas the other approaches fail. On the other hand, the sharp scene cuts (frames 156 and 231) and the full object disappearance (frame 156) of video 2 make the performance of CS and PF drop severely. Our approach instead is not affected at all, suspending the tracking when necessary and resuming it as soon as the structure of the model is found again.

Conversely, Video 3 is a static camera sequence but the persons occlude each other several times and the scene is full of color distractors (e.g. the several skin-colored regions of faces and arms, the two blue jeans, the dark t-shirt of the person on the right and the dark cupboard on the back wall). In such conditions the use of a structural model is determinant to have a successful tracking. As can be seen in sequence 3-a, our approach takes a few frames to resume the tracking since it needs to locate the structure first.

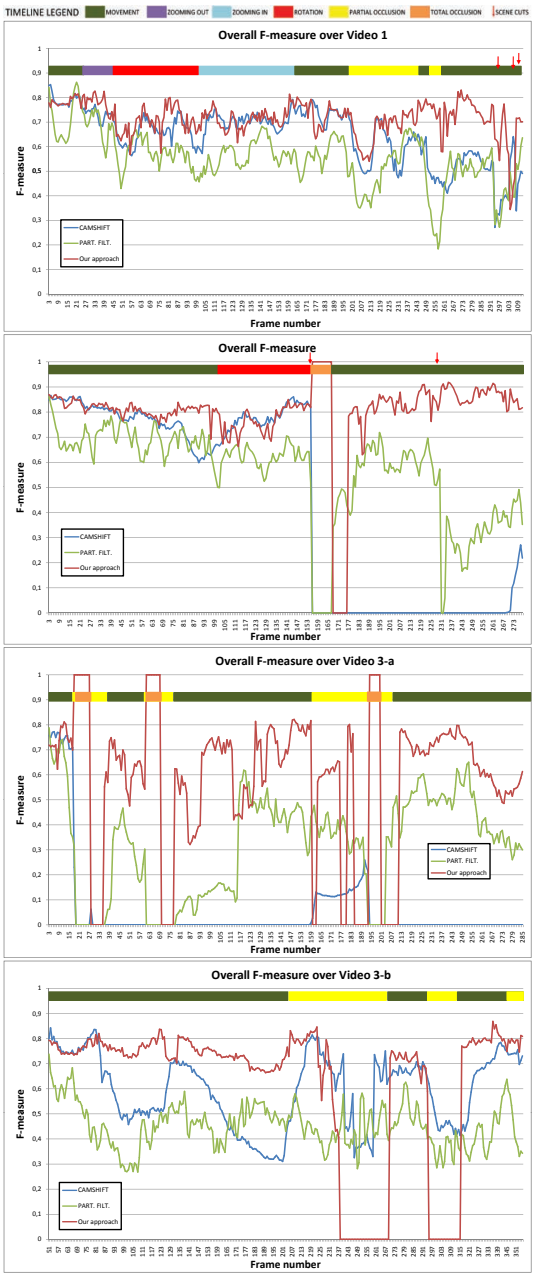

Figure 7. Pixel-level measure of performance. On top of each graph, a time line represents the different challenges on the tracking. The legend is on the top of the figure.

6. Conclusions and Future Work

The proposed joint feature-structure approach to object tracking in freely moving camera scenarios has shown inter-

Object level	Recall			Precision			F-measure		
	CS	PF	GB	CS	PF	GB	CS	PF	GB
Video 1	96,72%	96,41%	99,24%	87,66%	74,52%	95,31%	91,97%	84,07%	97,24%
Video 2	92,83%	95,99%	100,00%	66,86%	89,00%	97,20%	77,73%	92,36%	98,58%
Video 3-a	30,25%	72,46%	97,87%	12,94%	69,89%	89,58%	18,13%	71,15%	93,54%
Video 3-b	88,59%	86,28%	98,13%	91,55%	79,39%	87,65%	90,05%	82,69%	92,59%
Pixel level	**Recall**			**Precision**			**F-measure**		
	CS	PF	GB	CS	PF	GB	CS	PF	GB
Video 1	84,54%	66,92%	85,71%	52,77%	49,80%	62,14%	64,09%	55,41%	71,16%
Video 2	53,20%	49,42%	84,87%	36,85%	64,64%	76,20%	43,34%	55,50%	79,78%
Video 3-a	7,93%	30,68%	65,26%	4,76%	36,93%	53,42%	5,67%	32,07%	57,67%
Video 3-b	65,06%	44,20%	71,95%	69,67%	74,30%	63,65%	64,66%	52,74%	66,90%

Table 2. Summary of the performance.

esting and promising results. In particular the experimental results show that the use of the structural approach give robustness to the tracking in the presence of severe occlusions and distractors. The coherence measure, used for weighting the association graph, is also used as a metric for the reliability of the tracking, allowing it to be suspended in case the object is not found in the scene. Moreover, the exclusion of low-coherence nodes from the extracted dominant set allows to reject false positive detections, often due to distractors. It is worth noting that the use of color features presented in this work is not a limitation, since the framework is flexible and open to be extended to different types of features. In fact, in future work we aim to include texture and edge features as well. Regarding the graph matching step, other search heuristics can be plugged into the framework in substitution of dominant sets. An extensive evaluation of the results obtained using different algorithms could be useful to choose the best performing technique in a general scenario. It should be noted that the current implementation of the dominant sets does not fit a real-time tracking as their extraction time for a single frame can span from one to several seconds, whereas discrete matching techniques could be much faster.

References

[1] imagelab.ing.unimore.it/imagelab/∼gualdi/dst.asp.

[2] E. L. Andrade Neto, E. Khan, J. C. Woods, and M. Ghanbari. Segmentation and tracking for interactive sport scenes using region adjacency graphs, picture trees and prior information. In *Picture Coding Symposium*, pages 353–358, 2003.

[3] Y. Bar-Shalom and T. Foreman. *Tracking and Data Association*. Acad.Pr.Inc, 1988.

[4] M. Bartalmio, G. Sapiro, and G. Randall. Morphing active contours. *IEEE Trans. on PAMI*, 22(7):733–737, 2000.

[5] H. Borotschnig, D. Sinclair, and A. Pinz. Fuzzy graph tracking. In *5th Int'l Symp. on Intelligent Robotic Systems*, 1997.

[6] G. Bradski. Real time face and object tracking as a component of a perceptual user interface. In *Proc. of WACV*, 1998.

[7] D. Comaniciu, V. Ramesh, and P. Andmeer. Kernel-based object tracking. *IEEE Trans. on PAMI*, 25:564–575, 2003.

[8] D. Conte, P. Foggia, J.-M. Jolion, and M. Vento. A graph-based, multi-resolution algorithm for tracking objects in presence of occlusions. *Pattern Recognition*, 39(4), 2006.

[9] P. Felzenszwalb and D. Huttenlocher. Pictorial structures for object recognition. *Int. J. Comput. Vision*, 61:55–79, 2005.

[10] C. Gomila and F. Meyer. Graph-based object tracking. *Proc. ICIP*, 2:41–44, 2003.

[11] A. Graciano, R. Cesar-Jr, and I. Bloch. Graph-based object tracking using structural pattern recognition. *Proc. of SIBGRAPI*, pages 179–186, 2007.

[12] M. Isard. Pampas: real-valued graphical models for computer vision. *Proc. of IEEE Int'l Conference on Computer Vision and Pattern Recognition*, 1:613–620, 2003.

[13] M. Isard and A. Blake. Condensation - conditional density propagation for visual tracking. *Int'l Journal of Computer Vision*, 29(1):5–28, 1998.

[14] D.-S. Jang, S.-W. Jang, and H.-I. Choi. 2d human body tracking with structural kalman filter. *Pattern Recognition*, 35(10):2041–2049, 2002.

[15] K. Mardia and P. Jupp. *Directional Statistics*. Wiley, 2000.

[16] K. Nummiaro, E. Koller-Meier, and L. Van Gool. An adaptive color-based particle filter. *Image and Vision Computing*, 21(1):99–110, 2003.

[17] M. Pavan and M. Pelillo. Dominant sets and pairwise clustering. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(1):167–172, 2007.

[18] M. Pelillo. Replicator equations, maximal cliques, and graph isomorphism. *Neural Comput.*, 11(8), 1999.

[19] D. Ramanan, D. Forsyth, and A. Zisserman. Tracking people by learning their appearance. *IEEE Trans. on PAMI*, 29(1):65–81, Jan. 2007.

[20] S. Rota Bulò, A. Torsello, and M. Pelillo. A continuous-based approach for partial clique enumeration. In *GbRPR*, pages 61–70, 2007.

[21] G. Schindler and F. Dellaert. A rao-blackwellized parts-constellation tracker. In *WDV*, pages 178–189, 2006.

[22] L. Sigal, S. Bhatia, S. Roth, M. Black, and M. Isard. Tracking loose-limbed people. *Proc. of IEEE Int'l Conference on Computer Vision and Pattern Recognition*, 1:421–428, 2004.

[23] E. Sudderth, A. Ihler, W. Freeman, and A. Willsky. Nonparametric belief propagation. *Proc. of IEEE Int'l Conference on Computer Vision and Pattern Recognition*, 1:605–612, 2003.

[24] F. Tang and H. Tao. Object tracking with dynamic feature graph. *2nd Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pages 25–32, 15-16 Oct. 2005.

[25] C. Veenman, M. Reinders, and E. Backer. Resolving motion correspondence for densely moving points. *IEEE Trans. on PAMI*, 23(1):54–72, 2001.

[26] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Journal of Computing Surveys*, 38(4), 2006.