Segmentation, Ordering and Multi-Object Tracking using Graphical Models

Chaohui Wang, Martin de La Gorce
Laboratoire MAS, Ecole Centrale Paris
{chaohui.wang, martin.de-la-gorce}@ecp.fr

Nikos Paragios
Laboratoire MAS, Ecole Centrale Paris
Equipe GALEN, INRIA Saclay - Ile-de-France
nikos.paragios@ecp.fr

Abstract

In this paper, we propose a unified graphical-model framework to interpret a scene composed of multiple objects in monocular video sequences. Using a single pairwise Markov random field (MRF), all the observed and hidden variables of interest such as image intensities, pixels' states (associated object's index and relative depth), objects' states (model motion parameters and relative depth) are jointly considered. Particular attention is given to occlusion handling by introducing a rigorous visibility modeling within the MRF formulation. Through minimizing the MRF's energy, we simultaneously segment, track and sort by depth the objects. Promising experimental results demonstrate the potential of this framework and its robustness to image noise, cluttered background, moving camera and background, and even complete occlusions.

1. Introduction

Segmentation and tracking in video sequences are among the most active research topics in *Computer Vision* and often serve as low and mid-level cues in numerous applications. One can cite for example, video surveillance, action recognition, robot navigation, medical imaging and human-machine interaction.

Segmentation aims at delineating object contours and is either edge-based or region-based. In the first case, one seeks the object boundaries based on visual discontinuities, while in the second case pixels are grouped together according to their visual properties and spatial relationships. Active contours [11] are the most popular methods in the first case, while recent MRF-based techniques are the current state of the art in the second case [1], with the advantage over active contours that they are less susceptible to local minima.

Tracking aims at locating moving objects in time and is either patch/appearance based over pair of images or performed using dynamical systems which also provide some information on the object trajectory properties. In both cases, similarities of the object appearance in time are used as metric. The mean-shift algorithm [3] and the condensation [9] are the most popular methods.

Segmentation and tracking are two complementary tasks and several previous methods aim at combining MRF segmentation with object tracking/pose estimation [2, 17]. These approaches relate to other methods that aim at introducing object shape priors within MRF segmentation [5, 7, 16]. In comparison with methods that first perform segmentation without any use of the available knowledge about the tracked object shape and then estimate the object pose from segmented regions [8, 25], these combined methods are able to cope with more challenging conditions such as image noise and cluttered background.

In [2], articulated object tracking and MRF segmentation are combined. A gradient-free local search is performed on an objective function which is defined as a function of the object articulation parameter vector. For each tested pose, a shape prior is defined using a stickman model and the image is segmented using binary graph-cuts. While being effective for single object tracking, this approach is not suited to multiple-object tracking, as it does not provide treatment of occlusions between objects. In [17], the poses of the tracked objects are predicted from previous frames, template shapes of the objects at the predicted locations are used as shape priors to perform multi-label MRF image segmentation with graph-cuts, and then the object locations are re-estimated using the segmented regions. The use of multi-label segmentation helps in avoiding *evidence over-counting* (*i.e.*, associating a pixel to more than one object) but is still insufficient to ensure robustness to sever occlusions that would require some occlusion reasoning.

Better performance can be expected by: (i) integrating occlusion reasoning using depth ordering between objects; (ii) coupling and simultaneously estimating all variables of interest (depth, object motion parameters and pixel segmentation labels), furthermore, if such an integration can be done within a single MRF, then one can also take advantage of existing/generic MRF inference techniques which are less prone to be trapped in local minima than local

search or expectation-maximization techniques, and thus can deal with more challenging cases. While depth notion and layered models were widely used in the literature [4, 10, 15, 18, 20, 23, 24], our method extends them to a unified MRF framework which performs scene understanding through the simultaneous estimation of the corresponding parameters.

However, taking into account the occlusion process between objects in a graphical-model formulation without introducing high order cliques is not straightforward. In [19, 22], for example, occlusions are partially considered towards avoiding over-counting image support, but the formulations do not intrinsically guarantee that at least one object or the background has to be associated to a given pixel.

In this paper, we propose a unified pairwise MRF to address the challenge of combining the segmentation, multi-object tracking with a rigorous visibility modeling (*i.e.*, depth ordering). The unknown pixels' states (associated object's index, relative depth) and objects' states (model motion parameters, relative depth) are integrated along with a principled way in the MRF. By minimizing the MRF's energy, we simultaneously segment the image, track and estimate the objects' motion parameters, and sort by depth the objects.

The main contribution of our approach is a single-shot optimization MRF framework for joint segmentation, depth ordering and multi-object tracking, where all the variables of interest do interact. To this end, we introduce a rigorous visibility modeling, which is achieved by introducing visibility constraints that involve only pairs of variables through a pairwise MRF. The resulting formulation is modular with respect to the data terms and independent from the inference algorithm.

The remainder of this paper is organized as follows: we present the generative scene modeling in section 2, and then transport it into the MRF formulation of the integrated multi-object tracking, ordering and image segmentation in section 3. Experimental validation and some discussion compose section 4. Finally, we conclude the paper with some future directions in section 5.

2. Generative Scene Modeling

Let us consider a sequence of images with K objects to be tracked and each image composed of N pixels. Furthermore, let $\mathcal{V}_o = \{1, 2, \ldots, K\}$ denote the index set of the objects and $\mathcal{V}_p = \{K+1, K+2, \ldots, K+N\}$ the index set of the pixels[1]. Let us assume that there is no mutual occlusion (*e.g.*, object 1 partially occludes object 2 and is partially occluded by object 2) between the objects. Thus each object can be considered to be flat, especially for the purpose of visibility modeling. We consider "background" as a special object which occupies all the pixels in the image and which lies behind (is occluded by) all the objects to be tracked[2]. We assign it an index "0" and use $\mathcal{V}_s$ to denote the extended object set which contains the indices of the objects and the background, *i.e.*, $\mathcal{V}_s = \mathcal{V}_o \cup \{0\}$.

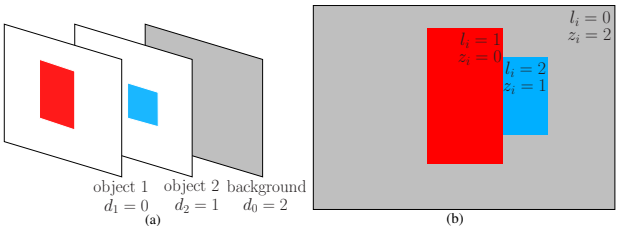


Figure 1. Sketch Map of Scene Generative Modeling

In such a setting, the spatial property of each object/background can be described by its 2d shape, *i.e.*, the projection in the image. Let us associate each object with a geometric shape prior and an appearance prior (with respect to the visual distribution of the intensities of the object). Let $\mathcal{M}_k(\theta_k)$ $(k \in \mathcal{V}_s)$ denote the index set of the pixels which are occupied by the shape model of object/background k with parametrization[3] θ_k. And let $\mathbf{H}_k$ $(k \in \mathcal{V}_s)$ denote the appearance prior of object/background k. Then we can generate the image, if we have a depth ordering between the objects and background. However, this ordering is unknown when we estimate the motion of the objects in a video sequence. Therefore, an explicit modeling of the visibility in the segmentation/motion estimation framework is to be considered.

We first introduce a *relative depth index*[4] d_k $(k \in \mathcal{V}_s)$ to each object/background, and only an object having smaller depth can occlude another object having bigger depth. Let us define the range of the depth. Assuming that in a video sequence, at most D $(D \leq K)$ objects (not including the background) may overlap at a point in time[5], $D+1$ depths are sufficient to model the depth ordering between the objects and the background (Note that two objects may have the same depth). Thus, we define $\mathcal{D} = \{0, 1, 2, \ldots, D-1\}$

[1] The pixels are indexed from $K+1$ in order to be coherent with their corresponding nodes' indices in the MRF formulation (see section 3).

[2] The floating background, *i.e.*, those objects which are not tracked but may occlude the objects to be tracked will be discussed in section 4.2. However, it is not a limitation with regard to the proposed generative framework.

[3] θ_0 is an abuse of notation (this variable is not needed), since the background's shape $\mathcal{M}_0$ always is the support of the image. It is considered for clarity and consistency purposes.

[4] Towards simplifying the presentation of the framework, the term *relative depth index* will be replaced by *depth*.

[5] A precise definition of D: Let us describe *depth ordering* of the objects/background in an image using a *Ordering Graph* $\mathcal{G}_v$ [4], which is a directed graph. Each object/background is represented by a node. The object corresponding to node k_1 occludes (partially or completely) the one corresponding to the node k_2 for an arc (k_1, k_2) of $\mathcal{G}_v$. Thus, D is the length of the longest (directed) path of $\mathcal{G}_v$.

as the set of all the possible depths for the objects, and "D" being the depth of the background, *i.e.*, $d_k \in \mathcal{D}$ $(k \in \mathcal{V}_o)$ and $d_0 = D$.

This depth is associated with the image through the *Pixel Label Consistency*. It imposes that for a given pixel i, we consider the objects whose shapes occupy this pixel and then associate this pixel to the one having the smallest depth. Let l_i $(l_i \in \mathcal{V}_s)$ denote the index of the object/background to which the pixel i associates, the above constraint can be formulated mathematically as follows:

$$l_i = \underset{\{k|i\in\mathcal{M}_k(\theta_k),k\in\mathcal{V}_s\}}{\arg\min} d_k \qquad (\forall i \in \mathcal{V}_p) \tag{1}$$

For *Pixel Label Consistency* being well defined, *i.e.*, $\arg\min_{\{k|i\in\mathcal{M}_k(\theta_k),k\in\mathcal{V}_s\}} d_k$ being singleton, we introduce a constraint on the assignment of d_k $(k \in \mathcal{V}_s)$, namely, *Object Depth Consistency*, which imposes the constraint that for a given pixel i, there is one and only one object which has the smallest depth among the objects whose shapes occupy it. We can formalize that as follows:

$$\begin{aligned} &\forall i \in \mathcal{V}_p \,, \exists \tilde{k} \in \{k|i \in \mathcal{M}_k(\theta_k) \,, k \in \mathcal{V}_s\} \\ &\text{s.t. } \forall k' \in \{k|i \in \mathcal{M}_k(\theta_k) \,, k \in \mathcal{V}_s\} \backslash \{\tilde{k}\} \,, d_{\tilde{k}} < d_{k'} \end{aligned} \tag{2}$$

A depth assignment which verifies *Object Depth Consistency* composes a depth ordering hypothesis between the objects/background.

In order to model the visibility in a distributed way, we also assign a *depth* z_i $(z_i \in \mathcal{D} \cup \{D\})$ to each pixel i. It represents the depth of the object to which the pixel associates, *i.e.*, $z_i = d_{l_i}$. Thus, we define *Pixel Depth Consistency* as:

$$z_i = \min_{\{k|i\in\mathcal{M}_k(\theta_k),k\in\mathcal{V}_s\}} d_k \qquad (\forall i \in \mathcal{V}_p) \tag{3}$$

Pixel Label Consistency, *Object Depth Consistency* and *Pixel Depth Consistency* (Formulas 1, 2 and 3) compose the constraints that ensure the values z_i and l_i of each pixel i to be those produced from the generative process of image with occlusions between the objects (Fig. 1).

While for a pixel i, its depth z_i can be retrieved using the pixel label l_i and the depth information of the associated object (*i.e.*, $z_i = d_{l_i}$), there is a necessity of modeling depth also at the pixel level. In our framework, we model the visibility in a distributed manner in the MRF and thus the depth ordering can be automatically estimated with other latent variables of interest during the inference process. To this end, introducing the depth z_i to each pixel is crucial because it allows to model the visibility using constraints that involve only pairs of variables through a pairwise MRF. Therefore, we reformulate the above mentioned constraints in a distributed manner as follows (see proof in *Appendix*):

$$\forall i \in \mathcal{V}_p, \quad \mathcal{A}_1 \wedge \mathcal{A}_2 \wedge \mathcal{A}_3 \Leftrightarrow \bigwedge_{k\in\mathcal{V}_s} (\mathcal{C}_{1k} \wedge \mathcal{C}_{2k} \wedge \mathcal{C}_{3k}) \tag{4}$$

with:

$$\begin{cases} \mathcal{A}_1 & : & l_i = \arg\min_{\{k|i\in\mathcal{M}_k(\theta_k),k\in\mathcal{V}_s\}} d_k \\ \mathcal{A}_2 & : & z_i = \min_{\{k|i\in\mathcal{M}_k(\theta_k),k\in\mathcal{V}_s\}} d_k \\ \mathcal{A}_3 & : & \exists \tilde{k} \in \{k|i \in \mathcal{M}_k(\theta_k) \,, k \in \mathcal{V}_s\} \text{ s.t.} \\ & & \forall k' \in \{k|i \in \mathcal{M}_k(\theta_k) \,, k \in \mathcal{V}_s\} \backslash \{\tilde{k}\} \,, d_{\tilde{k}} < d_{k'} \\ \mathcal{C}_{1k} & : & \neg((l_i = k) \wedge (z_i \neq d_k)) \\ \mathcal{C}_{2k} & : & \neg((l_i = k) \wedge (i \notin \mathcal{M}_k(\theta_k))) \\ \mathcal{C}_{3k} & : & \neg((l_i \neq k) \wedge (z_i \geq d_k) \wedge (i \in \mathcal{M}_k(\theta_k))) \end{cases} \tag{5}$$

1. Keeping $\mathcal{C}_{1k}$ true imposes that: the depth of pixel i should be equal to the depth of object/background k if it is associated to the object k.

2. Keeping $\mathcal{C}_{2k}$ true imposes that: a pixel i can be associated to object/background k only when it is occupied by the shape of object/background k.

3. Keeping $\mathcal{C}_{3k}$ true imposes that: if a pixel i is occupied by the shape of object/background k, it can be associated to an object other than k only when the depth of pixel i is strictly smaller than the depth of object/background k.

With such an equivalence, the satisfaction of the above mentioned conditions on the right-side in formula (4) for each pixel ensures that a pixel i will be explained once and only once by the object which is supposed to be visible at pixel i. One can now integrate these constraints/conditions with support coming from the images towards segmentation, ordering and multi-object tracking. We adopt the use of a pairwise MRF, since conditions of the visibility satisfaction can be mapped to pairwise interactions, while image support can be encoded through singleton potentials.

3. Markov Random Field Formulation

The proposed MRF is composed of two types of nodes (Fig. 2). The first is *object* nodes corresponding to the objects to be tracked, and the second is *pixel* nodes corresponding to the image pixels. The index set of the nodes is denoted by $\mathcal{V} = \mathcal{V}_o \cup \mathcal{V}_p$ (V_o and $\mathcal{V}_p$ correspond respectively to the two types of nodes, see section 2 for detail).

The MRF comprises a discrete latent random variable vector $\mathbf{X} = (X_i)_{i\in\mathcal{V}}$ such that each variable X_i takes a value x_i from its label set $\mathcal{X}_i$ containing all possible labels. We use $\mathbf{x} = (x_i)_{i\in\mathcal{V}}$ to denote the MRF's configuration and $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2 \times \cdots \times \mathcal{X}_{|\mathcal{V}|}$ the MRF's configuration space, *i.e.*, $\mathbf{x} \in \mathcal{X}$. The latent variable X_i for the two types of nodes will be defined with their singleton potentials in section 3.1.

In order to introduce the geometric prior and the visibility satisfaction constraints, all the object nodes are connected with all the pixel nodes. These edges compose the

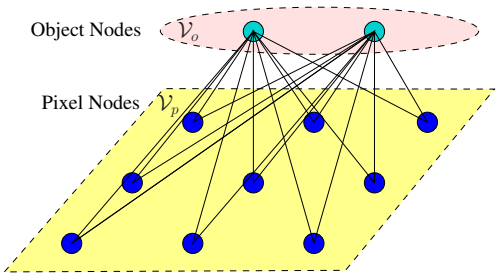


Figure 2. MRF Model (Example for Two Tracked Objects)

edge set $\mathcal{E}$ of the MRF, *i.e.*, $\mathcal{E} = \{(k,i)|k \in \mathcal{V}_o, i \in \mathcal{V}_p\}$. We can also introduce interactions/constraints on the labels of the pixel nodes (in particular with respect to the segmentation) through conventional 4-neighborhood or 8-neighborhood systems [1], which will be discussed in section 4.3.

The total energy of the MRF with a configuration $\mathbf{x}$ is defined as:

$$E(\mathbf{x}) = \sum_{i \in \mathcal{V}} \phi_i(x_i) + \sum_{(k,i) \in \mathcal{E}} \psi_{k,i}(x_k, x_i) \qquad (6)$$

3.1. Singleton Potential

There are two types of singleton potentials, one referring to the pixel nodes and the other referring to the object nodes. They mostly encode the intensity evidence coming from the image and object motion priors from one frame to the next.

Pixel node singleton term: The latent random variable X_i ($i \in \mathcal{V}_p$) is composed of the associated object's index and the depth, *i.e.*, $x_i = (l_i, z_i)$. We define the configuration space of pixel node i as: $\mathcal{X}_i = (\mathcal{V}_o \times \mathcal{D}) \cup \{(0, D)\}$ (see section 2 for the definitions of these symbols). Note that if a pixel is labeled as "background" (*i.e.*, $l_i = 0$), its depth is deterministic (*i.e.*, $z_i = D$). Like [1, 2], we use singleton potential $\phi_i(x_i)$ ($i \in \mathcal{V}_p$) to encode the data likelihood, which imposes penalties for assigning l_i to pixel i and is defined as:

$$\phi_i(x_i) = -\log \Pr(\mathbf{I}_i | \mathbf{H}_{l_i}) \qquad (7)$$

where $\mathbf{I}_i$ denotes the intensity/color (*e.g.*, RGB value) of pixel i, and $\mathbf{H}_k$ ($k \in \mathcal{V}_s$) denotes the intensity/color distribution for object/background k. We can model the color distribution for each object/background using existing approaches such as a Gaussian mixture, a kernel-based approximation (*e.g.*, Parzen windows) of the distribution, *etc.*

Object node singleton term: The latent random variable X_k ($k \in \mathcal{V}_o$) is composed of the motion parameters of the shape model (*e.g.*, position, rotation, scale, other deformation parameters) and the depth, *i.e.*, $x_k = (\theta_k, d_k)$. We use $\mathcal{X}_k = \Theta_k \times \mathcal{D}$ to denote the configuration space of object node k, where Θ_k denotes the motion parameter space. The singleton potential function for object node k encodes the prior preference on x_k and can be defined as:

$$\phi_k^{(t)}(x_k^{(t)}) = \alpha_1 \cdot \left\| \theta_k^{(t)} - \hat{\theta}_k^{(t)} \right\|^2 + \alpha_2 \cdot \left\| d_k^{(t)} \right\| \qquad (8)$$

where $\alpha_1 > 0$ and $\alpha_2 > 0$ are the weights for the corresponding terms, $\hat{\theta}_k^{(t)}$ is the predicted configuration of θ_k for the instant t, and $\|\cdot\|$ denotes the Euclidean norm.

The first term is defined to model the temporal constraint. Actually, it can be neglected when the image evidence is quite informative for estimating the motion of objects. We explicitly define it in the theoretical framework in order to keep the general preciseness of the model. For example, when an object is completely occluded by another object during a period, there is not enough visual information to determine the motion of this object and this prior term can be used to determine the motion. The choice of the predictor for $\hat{\theta}_k^{(t)}$ is independent from this framework and one can choose an off-the-shelf predictor.

The second term is used to avoid an arbitrary choice of depth in case of depth ambiguities by favoring the depth variable to be the smallest possible one because, without this term, different depths may produce the same MRF's energy. One obvious example is the case of an object having no occlusion with any other object, since it can take any possible depth. However, the neglect of this term will not impact the performance of the method.

3.2. Pairwise Potential

The pairwise potential between an object and a pixel is used to model the shape prior and the visibility constraint in the MRF. For an edge (k,i) ($k \in \mathcal{V}_o$ and $i \in \mathcal{V}_p$), the pairwise potential $\psi_{k,i}(x_k, x_i)$ is defined as:

$$\psi_{k,i}(x_k, x_i) = \psi_{k,i}^{(1)}(x_k, x_i) + \psi_{k,i}^{(2)}(x_k, x_i) + \psi_{k,i}^{(3)}(x_k, x_i) \qquad (9)$$

where $\psi_{k,i}^{(1)}$, $\psi_{k,i}^{(2)}$ and $\psi_{k,i}^{(3)}$ are the penalties respectively corresponding to the cases $\mathcal{C}_{1k}$ = false, $\mathcal{C}_{2k}$ = false and $\mathcal{C}_{3k}$ = false (see formula 5 for $\mathcal{C}_{1k}$, $\mathcal{C}_{2k}$ and $\mathcal{C}_{3k}$):

$$\begin{cases} \psi_{ki}^{(1)}(x_k, x_i) = \gamma_1 \cdot [\neg \mathcal{C}_{1k}] \\ \psi_{ki}^{(2)}(x_k, x_i) = \gamma_2 \cdot dist(i, \mathcal{M}_k(\theta_k)) \cdot [\neg \mathcal{C}_{2k}] \\ \psi_{ki}^{(3)}(x_k, x_i) = \gamma_3 \cdot dist(i, \mathcal{M}_k^c(\theta_k)) \cdot [\neg \mathcal{C}_{3k}] \end{cases} \qquad (10)$$

where $\gamma_1 > 0$, $\gamma_2 > 0$ and $\gamma_3 > 0$ are the weights for the corresponding penalties, $\mathcal{M}_k^c(\theta_k)$ denotes the complement of $\mathcal{M}_k(\theta_k)$, Iverson Bracket $[\cdot]$ is defined as: for a statement S, $[S] = 1$ if S is true and 0 otherwise, and $dist(i, \mathcal{M})$ denotes the distance function which is defined as the minimum Euclidean distance between the geometric shape corresponding to $\mathcal{M}$ and the considered pixel's position:

$$dist(i, \mathcal{M}) = \min_{j \in \mathcal{M}} \|loc(i) - loc(j)\| \qquad (11)$$

where $loc(i)$ denotes the spatial coordinates of pixel i in the image.

Instead of giving an infinite penalty to any case where a statement in formula 5 is false, we set $\psi_{ki}^{(1)}$ to be constant, $\psi_{ki}^{(2)}$ and $\psi_{ki}^{(3)}$ to be distance penalties. This is motivated by the fact that, in general, shape models are not exact: when we get closer to the center of shape, the degree of certainty of being in the projection increases. Such a penalty yields an elastic force and can guide both object tracking and image segmentation.

Using the MRF model defined above, we can now simultaneously perform segmentation, ordering and multi-object tracking, which is formulated as the inference of those latent random variables through a minimization problem over the MRF's total energy:

$$\mathbf{x}^{opt} = \arg\min_{\mathbf{x}} E(\mathbf{x}) \tag{12}$$

This MRF can be optimized using standard inference methods. We adopt the *sequential tree-reweighted message passing* (TRW-S) [12], since it offers a good compromise between the quality of the obtained minimum, the ability to model complex interactions between the MRF's nodes and reasonable computational complexity.

4. Experimental Results

In order to validate the proposed framework, we have considered several video sequences of increasing difficulties.

4.1. Experimental Setting

A weak geometric prior is considered, which is a bounding box (except for *Shell Game* sequence, where the geometric prior is the manually delineated contour of each object in the first frame.). The motion parameters θ_k of each object correspond to the position, scale and rotation angle around the shape's center of mass. The position space is defined as the support (or: lattice) of the image. The rotation angle space is defined as $\{r|r \in \mathbb{Z} \text{ and } 0 \leq r < 360\}$, where $\mathbb{Z}$ is the set of all integers. The scale factor space is defined as $\{s|s = 1.05^n, n \in \mathbb{Z}\}$. In practice, the search space is in the vicinity of the previous motion parameter vector, due to the fact that the motion between two successive frames is limited. This setting is combined with a linear predictor where the estimated motion parameter vector for the current frame is used to predict that of the next frame, *i.e.*, $\hat{\theta}_k^{(t)} = \theta_k^{opt,(t-1)}$ $(k \in \mathcal{V}_o)$.

For the visual appearance term, we distinguish the case of static background from that of dynamic background. In the first case, using the manual delineation of the objects in the first frame, a Gaussian mixture is considered towards modeling the color distribution of each object. The background's color, either is globally modeled as a Gaussian mixture (*Box* and *Shell Game* sequences), or is modeled using a pixelwise model (*Pedestrian Sequence 1*), *i.e.*, each pixel's color is modeled using a Gaussian distribution whose mean and variance are learned from a sequence of background images [21]. The case of dynamic background (*Pedestrian Sequences 2 and 3*) is treated differently. Given the manual segmentation of the first frame, a non-parametric Parzen windows approximation is used to model the color distribution of each object/background. The color model for the background is updated for the next frame using the segmentation result of the current frame, while those of the objects are kept constant.

There are two components still to be addressed, the motion parameter sampling and the parameter setting for the weights of the MRF's energy. We adopt a sparse sampling strategy [6], where $\theta_k^{(t)}$ is sampled uniformly along each main axis plus the two diagonal directions of the translation centered at the predicted value $\hat{\theta}_k^{(t)}$, plus $\hat{\theta}_k^{(t)}$ itself to get the motion parameter candidates. In order to mitigate inaccuracy of the solution due to the fact that the sampling is sparse, we iterate by re-sampling at each iteration around the solution found in the previous iteration. According to the roles of the energy terms, we set the parameters as follows: we adjust and fix γ_2 by trial and error on the first few frames. It is different from one sequence to another since the color statistics and/or the color model may be different. The rest are set as: $\gamma_1 = 50\gamma_2$ and $\alpha_1 = \alpha_2 = \gamma_3 = \gamma_2$.

4.2. Results

We show the results on two sequences with rigid objects and three sequences with deformable objects. The test sequences involve varying degree of image quality (severe noise has been added to some of them), varying complexity both with respect to the objects and background visual properties, varying degree of occlusions and last, but not least both static and moving observers.

Box Sequence: In the original sequence, two boxes move such that significant occlusions (even complete occlusions) occur between them. Our algorithm has well tracked the objects, segmented the image, and estimated the depths of the objects. Furthermore, in order to test the robustness to noise, we independently added Gaussian white noise of mean 0 and variance 0.8 (the range of RGB value is $[0, 1]^3$) to each frame. Figure 3(a) shows the results for this very degraded video.

Shell Game Sequence: In order to test the robustness of our algorithm to both temporally and spatially significant occlusions, we have tested *Shell Game* sequence [8]. In this video, there are three identical cups facing downwards and two chips of different colors. The operator begins the game by placing two cups such that each cup covers one of

the two chips, then he/she quickly shuffles the three cups around and finally uncovering the chips. While being occluded, each chip keeps sliding with the cup that covers it. This video is quite challenging mainly due to the long-term complete occlusions of the two occluded chips (Fig. 3(b)).

Note that we previously assumed that the background was always behind all the objects. However, one can also imagine floating background, *i.e.*, those objects which are not to be tracked but may occlude the tracked objects (*e.g.* the hands in the video). In our experiments, we dealt with this by adding another possible depth "-1" for the background (*i.e.*, add $(0, -1)$ into $\mathcal{X}_i$ $(i \in \mathcal{V}_p)$) and giving a prior penalty to the case where a pixel is labeled as "background" and has depth "-1".

Pedestrian Sequences: Sever occlusions have also been considered in a real setting, with deformable objects, image noise, changes of illumination and moving camera. We have considered three sequences: (i) the first one consists of a static background with five people, severe occlusions between the objects and the maximum level of occlusions being three (Fig. 3(c)); (ii) the second one consists of a moving background with five people, severe noise and changes of illumination (Fig. 3(d)); (iii) the last one consists of a moving background with four people and significant changes in texture (Fig. 3(e)).

For these pedestrian sequences, a rectangle is used to model the shape of a person. Since, in the shape prior, the torso is more reliable than the limbs due to limb motions, we manually set an area inside the shape model (*i.e.*, including the majority of the torso), and it has the same motion as the shape model. When computing $\psi_{ki}^{(3)}(x_k, x_i)$ using formula 10, if pixel i is inside this area with the configuration θ_k, we multiply $\psi_{ki}^{(3)}(x_k, x_i)$ by a factor 10 to increase the confidence to this area, and otherwise we divide $\psi_{ki}^{(3)}(x_k, x_i)$ by a factor 10 to decrease the confidence.

For these test sequences, despite of different difficulties, our algorithm has successfully segmented, tracked and ordered by depth all the objects. The main limitation of the method is the computational complexity. Running times vary from a few seconds to several minutes per frame. It is shown that, with presence of occlusions in the observed image, TRW-S needs much more iterations to converge to a satisfactory solution than the cases without occlusions.

4.3. Discussion

Algorithm Acceleration: Since the object motion is bounded in a finite speed, given the motion configuration at an instant $t-1$, there is no need to model the relationship between an object and all the pixels for instant t. Based on this observation, we propose an approach to simplifying the MRF model in section 3. Once we get the estimation of the motion parameters $\theta_k^{opt,(t-1)}$ $(k \in \mathcal{V}_o)$, we calculate the distance function $\mathcal{M}_k(\theta_k^{opt,(t-1)})$ for object k. Using this distance function, we prune the connections between the object k and those pixels i with $dist(i, \mathcal{M}_k(\theta_k^{opt,(t-1)})) > b$ (where b is a tolerance coefficient). And for these pixels, the label k is excluded from their configuration spaces. In this way, the algorithm can be sped up by more than 15 times on average, which was observed during experiments (with $b = 20$).

Introducing Interactions between Pixels: As we said previously, we can also introduce interactions/constraints on the labels of the pixel nodes through conventional 4-neighborhood or 8-neighborhood systems. To this end, we add the edges between those neighbor pixels into the edge set $\mathcal{E}$. Thus, we can smooth the segmentation result by defining the corresponding potential as:

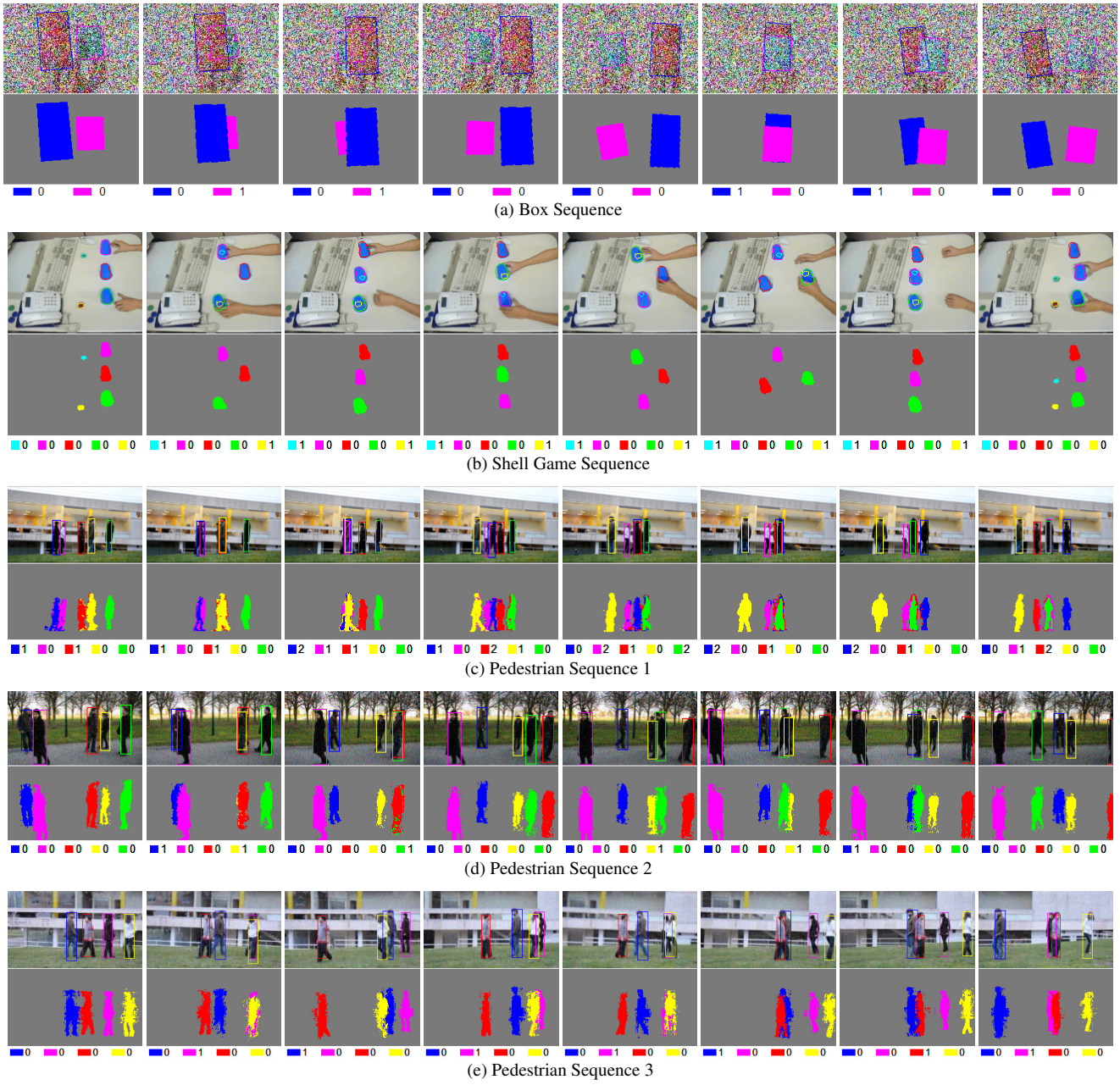
$$\psi(x_i, x_j) = \begin{cases} \eta\,(\eta > 0) & \text{if } l_i \neq l_j \\ 0 & \text{if } l_i = l_j \end{cases} \quad (i, j \in \mathcal{V}_p, (i, j) \in \mathcal{E}) \tag{13}$$

which favors neighbor pixels having the same label. We can also define other forms of potentials (*e.g.*, by considering the contrast). We have tested the cases both with and without this smoothness term. It is shown that the inclusion of this term does not improve the tracking performance but can smooth and improve the segmentation to some extent. However, the running-time significantly increases with the use of this term and the choice of η complicates the parameter setting.

5. Conclusion and Future work

In this paper, we have proposed a novel approach for segmentation, depth ordering and tracking with occlusion handling. Our approach introduces a distributed way to deal with visibility satisfaction where individual pixel modeling contributes to the depth ordering of objects through condition preservation constraints. The above constraints are expressed as cost terms in an MRF and are integrated with image support towards scene understanding. To the best of our knowledge, this is the first approach that combines low-level image support with high-level object representation along with proper occlusion handling in a single modular MRF where image data terms as well as priors can be easily replaced with more advanced models. Promising experimental results demonstrate the potentials of the method.

Improving the object representation towards more accurate tracking is one of the most promising directions of our approach. Opposite to simple rectangle representations, we can imagine more complex object representations that are able to cope with important deformations such as point distribution models. Another possible direction is to use this framework to deal with articulated objects such as hand



(a) Box Sequence

(b) Shell Game Sequence

(c) Pedestrian Sequence 1

(d) Pedestrian Sequence 2

(e) Pedestrian Sequence 3

Figure 3. Experimental Results. The first line of each sub-figure is the tracking result, where we draw the shape contours of the objects with the estimated motion parameters. The second line is the segmentation result. The third line presents the estimated depths of the objects. We use different colors to distinguish the objects.

pose estimation, where the rigorous handling of visibility/occlusion could greatly impact the quality of the obtained results. Also, the use of higher order MRFs [13] could introduce significant modeling flexibility and better understanding of the scene. Last, but not least accelerating the convergence of the method is necessary to make the method applicable to other scenarios using either more efficient optimization algorithms [14] or a GPU implementation.

Acknowledgments

The authors would like to thank Prof. Dimitris Samaras for fruitful discussions, and Loic Simon, Rola Harmouche, Fabrice Michel and Prof. Iasonas Kokkinos for their valuable comments towards improving the clarity and the understanding of the paper.

Appendix: Proof of Equivalence in Formula 4

Let $\mathcal{A} = \mathcal{A}_1 \wedge \mathcal{A}_2 \wedge \mathcal{A}_3$, $\mathcal{C} = \bigwedge_{k\in\mathcal{V}_s}(\mathcal{C}_{1k} \wedge \mathcal{C}_{2k} \wedge \mathcal{C}_{3k})$ and $\mathcal{O}_i = \{k | i \in \mathcal{M}_k(\theta_k), k \in \mathcal{V}_s\}$.

"$\Rightarrow$": We first prove that for a pixel i ($\forall i \in \mathcal{V}_p$), "$\mathcal{A}$ is true" then "$\mathcal{C}$ is true" using *Reduction to the absurd*:

1. Assuming that $\exists \tilde{k} \in \mathcal{V}_s$ s.t. $\mathcal{C}_{1\tilde{k}}$ is false, then $l_i = \tilde{k}$ and $z_i \neq d_{\tilde{k}}$. But according to $\mathcal{A}_1$, $\mathcal{A}_2$, $z_i = d_{l_i} = d_{\tilde{k}}$. So the assumption is wrong, *i.e.*, $\forall k \in \mathcal{V}_s$, $\mathcal{C}_{1k}$ is true.
2. Assuming that $\exists \tilde{k} \in \mathcal{V}_s$ s.t. $\mathcal{C}_{2\tilde{k}}$ is false, then $l_i = \tilde{k}$ and $i \notin \mathcal{M}_{\tilde{k}}(\theta_{\tilde{k}})$. But according to $\mathcal{A}_1$, $l_i \in \mathcal{O}_i$ then $\tilde{k} \in \mathcal{O}_i$, *i.e.*, $i \in \mathcal{M}_{\tilde{k}}(\theta_{\tilde{k}})$. So the assumption is wrong, *i.e.*, $\forall k \in \mathcal{V}_s$, $\mathcal{C}_{2k}$ is true.
3. Assuming that $\exists \tilde{k} \in \mathcal{V}_s$ s.t. $\mathcal{C}_{3\tilde{k}}$ is false, then $l_i \neq \tilde{k}$, $z_i \geq d_{\tilde{k}}$ and $i \in \mathcal{M}_{\tilde{k}}(\theta_{\tilde{k}})$. So $\tilde{k} \in \mathcal{O}_i$. And according to $\mathcal{A}_1$ and $\mathcal{A}_2$, $d_{l_i} = z_i \geq d_{\tilde{k}}$. But according to $\mathcal{A}_1$ and $\mathcal{A}_3$, $d_{l_i} < d_{k'}$ ($\forall k' \in \mathcal{O}_i \backslash \{l_i\}$). So the assumption is wrong, *i.e.*, $\forall k \in \mathcal{V}_s$, $\mathcal{C}_{3k}$ is true.

"$\Leftarrow$": Now we prove that for a pixel i ($\forall i \in \mathcal{V}_p$), "$\mathcal{C}$ is true" then "$\mathcal{A}$ is true":

$$\begin{aligned}\mathcal{C} =& (\bigwedge_{k\in\mathcal{V}_s} \mathcal{C}_{1k}) \wedge (\bigwedge_{k\in\mathcal{V}_s} \mathcal{C}_{2k}) \wedge (\bigwedge_{k\in\mathcal{V}_s} \mathcal{C}_{3k}) \\ =& \underbrace{(\neg \bigvee_{k\in\mathcal{V}_s} (\neg\mathcal{C}_{1k}))}_{\mathcal{C}_1} \wedge \underbrace{(\neg \bigvee_{k\in\mathcal{V}_s} (\neg\mathcal{C}_{2k}))}_{\mathcal{C}_2} \wedge \underbrace{(\neg \bigvee_{k\in\mathcal{V}_s} (\neg\mathcal{C}_{3k}))}_{\mathcal{C}_3}\end{aligned} \tag{14}$$

$$\begin{aligned}\mathcal{C}_1 \quad &\Leftrightarrow \quad \nexists k \in \mathcal{V}_s, (l_i = k) \wedge (z_i \neq d_k) \\ &\Rightarrow \quad d_{l_i} = z_i \end{aligned} \tag{15}$$

$$\begin{aligned}\mathcal{C}_2 \quad &\Leftrightarrow \quad \nexists k \in \mathcal{V}_s, (l_i = k) \wedge (i \notin \mathcal{M}_k(\theta_k)) \\ &\Rightarrow \quad l_i \in \mathcal{O}_i \end{aligned} \tag{16}$$

$$\begin{aligned}\mathcal{C}_3 \quad &\Leftrightarrow \quad \nexists k \in \mathcal{V}_s, (l_i \neq k) \wedge (z_i \geq d_k) \wedge (i \in \mathcal{M}_k(\theta_k)) \\ &\Rightarrow \quad \forall k' \in \mathcal{O}_i \backslash \{l_i\}, z_i < d_{k'} \end{aligned} \tag{17}$$

1. (15) and (17) $\Rightarrow \forall k' \in \mathcal{O}_i \backslash \{l_i\}, d_{l_i} < d_{k'}$. And according to (16), $l_i \in \mathcal{O}_i$. So $\mathcal{A}_1$ and $\mathcal{A}_3$ are true.
2. (15) and $\mathcal{A}_1$ (has been proved to be true) $\Rightarrow z_i = d_{l_i} = d_{\arg\min_{k\in\mathcal{O}_i} d_k} = \min_{k\in\mathcal{O}_i} d_k$, *i.e.*, $\mathcal{A}_2$ is true.

References

[1] Y. Boykov and G. F. Lea. Graph cuts and efficient n-d image segmentation. *IJCV*, 70(2):109–131, November 2006.

[2] M. Bray, P. Kohli, and P. Torr. Posecut: Simultaneous segmentation and 3d pose estimation of humans using dynamic graph cuts. In *ECCV*, 2006.

[3] D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. In *CVPR*, 2000.

[4] T. Darrell and D. Fleet. Second-order method for occlusion relationships in motion layers. Technical Report 314, MIT Media Lab, 1995.

[5] D. Freedman and T. Zhang. Interactive graph cut based segmentation with shape priors. In *CVPR*, 2005.

[6] B. Glocker, N. Komodakis, G. Tziritas, N. Navab, and N. Paragios. Dense image registration through mrfs and efficient linear programming. *Medical Image Analysis*, 12(6):731–741, December 2008.

[7] R. Huang, V. Pavlovic, and D. N. Metaxas. A graphical model framework for coupling mrfs and deformable models. In *CVPR*, 2004.

[8] Y. Huang and I. Essa. Tracking multiple objects through occlusions. In *CVPR*, 2005.

[9] M. Isard and A. Blake. Condensation - conditional density propagation for visual tracking. *IJCV*, 29(1):5–28, August 1998.

[10] A. D. Jepson, D. J. Fleet, and M. J. Black. A layered motion representation with occlusion and compact spatial support. In *ECCV*, 2002.

[11] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *IJCV*, V1(4):321–331, January 1988.

[12] V. Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *PAMI*, 28(10):1568–1583, October 2006.

[13] N. Komodakis and N. Paragios. Beyond pairwise energies: Efficient optimization for higher-order mrfs. In *CVPR*, 2009.

[14] N. Komodakis, G. Tziritas, and N. Paragios. Performance vs computational efficiency for optimizing single and dynamic mrfs: Setting the state of the art with primal-dual strategies. *CVIU*, 112(1):14–29, 2008.

[15] M. P. Kumar, P. H. S. Torr, and A. Zisserman. Learning layered motion segmentation of video. In *ICCV*, 2005.

[16] M. P. Kumar, P. H. S. Torr, and A. Zisserman. Obj cut. In *CVPR*, 2005.

[17] J. Malcolm, Y. Rathi, and A. Tannenbaum. Multi-object tracking through clutter using graph cuts. In *ICCV*, 2007.

[18] M. Nitzberg and D. Mumford. The 2.1-d sketch. In *ICCV*, 1990.

[19] L. Sigal and M. J. Black. Measure locally, reason globally: Occlusion-sensitive articulated pose estimation. In *CVPR*, 2006.

[20] P. Smith, T. Drummond, and R. Cipolla. Layered motion segmentation and depth ordering by tracking edges. *PAMI*, 26(4):479–494, April 2004.

[21] C. Stauffer and W. Grimson. Adaptive background mixture models for real-time tracking. In *CVPR*, 1999.

[22] E. B. Sudderth, M. I. Mandel, W. T. Freeman, and A. S. Willsky. Distributed occlusion reasoning for tracking with nonparametric belief propagation. In *NIPS*, 2004.

[23] H. Tao, H. S. Sawhney, and R. Kumar. Dynamic layer representation with applications to tracking. In *CVPR*, 2000.

[24] J. Winn and A. Blake. Generative affine localisation and tracking. In *NIPS*, 2004.

[25] T. Yang, S. Z. Li, Q. Pan, and J. Li. Real-time multiple objects tracking with occlusion handling in dynamic scenes. In *CVPR*, 2005.