Semi-Supervised Video Segmentation using Tree Structured Graphical Models

Vijay Badrinarayanan, Ignas Budvytis, and Roberto Cipolla, *Senior Member, IEEE*

Abstract—We present a novel patch-based probabilistic graphical model for semi-supervised video segmentation. At the heart of our model is a temporal tree structure which links patches in adjacent frames through the video sequence. This permits exact inference of pixel labels without resorting to traditional short time-window based video processing or instantaneous decision making. The input to our algorithm are labelled key frame(s) of a video sequence and the output is pixel-wise labels along with their confidences. We propose an efficient inference scheme that performs exact inference over the temporal tree, and optionally a per frame label smoothing step using loopy BP, to estimate pixel-wise labels and their posteriors. These posteriors are used to learn pixel unaries by training a Random Decision Forest in a semi-supervised manner. These unaries are used in a second iteration of label inference to improve the segmentation quality. We demonstrate the efficacy of our proposed algorithm using several qualitative and quantitative tests on both foreground/background and multi-class video segmentation problems using publicly available and our own datasets.

Index Terms—Semi-Supervised Video Segmentation, Label Propagation, Mixture of Trees Graphical Model, Tree-structured Video Models, Structured Variational Inference.

1 Introduction

Semi-supervised video segmentation has a number of interesting applications, including video editing, harvesting labelled video data for training classifiers and learning shape, actions [1] as well as developing priors for unsupervised video segmentation [2]. In the past, several heuristic systems for semi-automatic video segmentation have been proposed [3], [4] which process a few frames at each step. But, unlike semi-supervised image segmentation [5], [6], rigorous video modelling and inference for semi-supervised video segmentation have not received much attention. This can perhaps be attributed to high cost of inference. In this work, we propose a probabilistic graphical model and an efficient inference method dedicated to semi-supervised video segmentation.

In recent times, unsupervised video segmentation has gained a lot of attention [7], [8], [9], especially as extensions of image super-pixellization to space-time super-pixels. The aim of these methods is to group pixels which are photometrically and motion wise consistent. In simple cases, where there is a clear distinction between foreground and the background, the grouping may appear to be semantically meaningful. However, in more complex videos, the result in general is an over-segmentation, and requires additional knowledge (through user interaction for example) to achieve any object level segmentation. When using unsupervised segmentation as a pre-processing step for class-specific segmentation, has to deal with issues like selecting super-pixels at the right scale for a class. To account for this uncertainty, each super-pixel is usually connected to several others in adjacent frames [10]. This leads to loopy models which require approximate inference schemes which are inefficient over 3D volumes. In contrast, we propose tree structured video models built using patch cross-correlation and which are quite robust, efficient for the task of label propagation. The tree structure automatically permits exact inference of pixel posteriors as opposed to approximate MAP inference.

Another distinction of our algorithm is that robust unaries can be learnt in a semi-supervised manner using only one or two frames of user labelled data and the inferred posteriors of the unlabelled pixels from the tree-structured model. These unaries aid in selecting regions of the right scale for a specific task and improve the quality of the segmentation. In contrast, with unsupervised methods, there is a need for additional mechanisms to select super-pixels at the right scale for each task separately.

One or two notable instances [11], [12] which have tried to extend their image segmentation algorithms directly for n-D sequences/videos have met with only limited success. A few others [13], [10] have tackled the problem of joint tracking and segmentation using unaries learnt at the start frame. We demonstrate via quantitative studies on such problems that our algorithm can achieve better or comparative results without using heuristics such as fixing the labels at each frame successively.

In this work, the semantic objects of interest are defined by the user labelled key frame(s) of a video sequence (see Fig.1). It is also possible to input only a few user mouse strokes in some frames (see supplementary material for an example). Our proposed segmentation algorithm uses this input to label each pixel in the video data into one of the user defined categories and infers their confidence

- *V. Badrinarayanan, I.Budvytis and R.Cipolla are with the Department of Engineering , University of Cambridge, Cambridge CB2 1PZ, UK E-mail: {vb292,ib255,cipolla}@eng.cam.ac.uk*

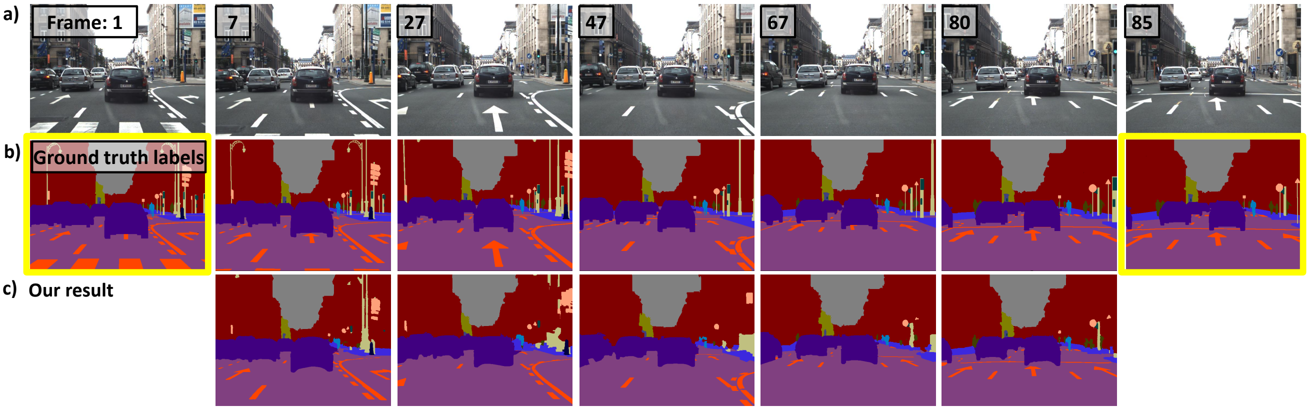


Fig. 1. An illustration of results of our proposed multi-class video segmentation approach. The image sequence and labels of frames 1,85 were used as input to our algorithm. The inferred MAP estimate of the pixel labels are shown in the bottom row. Notice the fairly clean label propagation even with frequent occlusions of the road/road markings by the moving cars. The ground truth labelling in the top row took about 30-45 minutes per frame, while the inferred labels takes only about 11 seconds per frame. Zoom-in and view in colour for best results.

estimates (posteriors). These posteriors (*soft labels*) can be used for learning pixel unaries using a Randomized Decision Forest [14] in a semi-supervised manner to further improve the segmentation.

As we perform probabilistic inference of pixel labels, a family of labellings at various confidence levels are available to the user as output. The user can then select one of these labellings at one or more frames, fix or clamp them, and re-infer the labelling over the whole video. This is similar to the self-training approach used in semi-supervised learning [10]. Probabilistic inference is also an important component for active learning methods [15], [16], [17], [10].

To summarise, we make the following contributions in this paper:

1) A patch-based probabilistic graphical model for semi-supervised video segmentation, which employs a novel temporal tree structure to link patches between frames.
2) An efficient structured variational inference scheme which infers pixel-wise labels and their confidences.

The remainder of this paper is organised as follows.

We present a detailed literature review in Sec. 2. Our proposed video model is explained in detail in Sec. 3. The inference strategy we propose for segmentation is elaborated in Sec. 3.2. A toy simulation of the algorithm and a step wise illustration of the algorithm on real data is presented. The process of semi-supervised learning of unaries using a Random Decision Forest classifier is explained. We then discuss our experiments and their results in Sec. 4. We bring out the advantages and the advantages and drawbacks of our approach in Sec. 5. We conclude in Sec. 6 with pointers to future work.

2 Literature Review

We review some of the relevant state of the art in unsupervised, classification based, semi-supervised and work flow based video segmentation.

2.1 Unsupervised Video Segmentation

The rectangular patch-based Epitome model [18], [19] and the pixel based Jigsaw model [20] learn a compact latent representation of an image or sequence of images. For a video sequence, this translates to learning correlations between pixels in both successive and non-successive frames. However, there is a model selection step (number of clusters, size of Epitome or Jigsaw) which is usually hand-crafted. In our proposed algorithm, we employ an epitomic model to learn correlations between successive frames which helps in tackling the aperture problem to an extent (see Fig. 4). However, we avoid costly learning of compact latent representations to establish correlations between non-successive frames and instead, choose a simpler alternative in the form of a Random Forest [21] to achieve the same goal.

Video super-pixellization methods such as [7], [8], [9], [22], [23] rely on grouping pixels in space and time using appearance and motion cues. The result is frequently over fragmented and their aim to understand the results by measures such as object covering, over segmentation is somewhat counter-intuitive because the performance statistics can vary depending on the task. Importantly, the results of unsupervised clustering are not easily interpreted as semantic clusters. However, consistent video super-pixellization can reduce the input dimension for structured discriminative models.

When using unsupervised segmentation as a pre-processing step for class-specific segmentation, it is necessary to deal with issues like selecting super-pixels/segments at the right scale for a class. To circumvent this issue, each super-pixel is connected to several others in adjacent frames [10]. This leads to loopy models which require approximate inference schemes which are inefficient over 3D volumes. Our approach uses simple patch cross-correlation to develop tree structured models for a video and which by construction permits efficient, exact inference. The results of exact inference are used to train a Random Forest which then helps select regions of the right scale for class specific segmentation.

2.2 Classification based Segmentation

We broadly divide methods in this category into unstructured and structured classification methods.

Unstructured Classification

Unstructured classifiers predict class labels independently for each pixel without incorporating any neighbourhood constraints. Randomized Decision Forests [21], an example of unstructured classifiers, have recently gained popularity in image and video segmentation [24], [14], [25]. In this work, we train a Random Decision Forest in a semi-supervised manner to learn pixel unaries and demonstrate that this learning can often help improve the quality of video segmentation.

Structured Classification

Structured classifiers incorporate neighbourhood constraints, such as spatial or temporal smoothness, to perform pixel class prediction. Conditional random field (CRF) models [11] are an example of widely applied structured classifiers which have lead the way in image segmentation problems. In practice, their main attraction arises from the ability to perform global optimisation or in finding a strong local minima of a particular class (submodular class) of CRF's at interactive speeds [11], [26]. There are one or two notable instances which have tried to extend their image segmentation algorithms directly for videos by propagating MAP estimates sequentially [12] or for N-D sequences [11]. As pointed out by [3], performing MAP inference on large 3D volumes can result in undesired changes in response to user input at another far away location in time. Finally, multi label MAP inference on the full video volume is extremely expensive [13].

2.3 Semi-Supervised Video Segmentation

The label propagation method of Badrinarayanan et. al. [27] jointly models appearance and semantic labels using a coupled-HMM model. The key idea is to influence the learning of frame to frame patch correlations as a function of both appearance and class labels. This method was extended to include correlations between non-successive frames using a Decision Forest classifier by Budvytis et. al. [24]. In this work, we follow these in jointly modelling appearance and semantic labels. The significant difference being that we use an undirected model which lends itself more naturally to fusion of classifiers and temporal modelling. In contrast, their directed models introduce competition (explaining away effect [28]) between classifiers and temporal models, which is not always desirable.

Tsai et. al [13] jointly optimize for temporal motion and semantic labels in an energy minimization framework. In this interesting framework, a sliding window approach is used to process overlapping n-frame grids for the sake of reducing computational burden. The result of one n-frame grid is employed as a hard constraint in the next grid and so on. Such an approach is also used in [29]. In contrast, we treat the whole video volume at once, inferring both temporal correlations and label uncertainties. Fathi et. al [10] use semi-supervised and active learning for video segmentation. Each unlabelled pixel is provided a confidence measure based on its distance to a labelled point, computed on a neighbourhood graph. These confidences are used to recommend frames in which more interaction is desired. In our approach, inference directly leads to confidences and active learning can also be pursued.

2.4 Work flow based Video Segmentation

The VideoSnapCut algorithm of [3] is an example of a work flow based system which relies on a heuristic combination of low level cues for video segmentation. Their main motivation is that methods based on global optimisation [4] can have unpredictable temporally non-local changes to user input. To avoid this, they employ spatially local classifiers and propagate their predictions over time using optical flow. However, the drawbacks are the heuristic nature of cue integration, use of unreliable flow and short time-window processing.

3 Proposed Video Model for Semi-Supervised Segmentation

We introduce a patch-based undirected graphical model for semi-supervised video segmentation which jointly models both the observed sequence of images (appearance layer) and their corresponding labels (label layer) . See Fig. 2(d) for an illustration. The model construction and label inference scheme are described below.

3.1 Model Construction

Fig. 2 illustrates a step by step construction of our model. We begin with the image epitome $\mathbf{e}$ [18], a compact version of the image and which has no spatial structure, as shown in Fig. 2(a). In this image generative model, the original frame/image I_k is assumed to be given as a set of patches $Z_k = \left\{Z_{k,j}\right\}_{j=1}^{P}$, each containing pixels from a subset of image coordinates $S_{k,j}$. The patches are taken to be square in shape and it is assumed that their coordinate sets can overlap. For each patch, a latent variable $T_{k,j}$ maps coordinates $S_{k,j}$ to coordinates in the epitome $\mathbf{e}$. A square patch is mapped to a square patch in the epitome through $T_{k,j}$. At pixel coordinate n in the epitome, a mean and variance μ_n, ϕ_n is stored. Given $\mathbf{e} = (\mu, \phi)$, the patch $Z_{k,j}$ is obtained by copying the epitome mean and adding Gaussian noise to a level prescribed by the variance map:

$$p\left(Z_{k,j}|\mathbf{e}, T_{k,j}\right) = \prod_{i \in S_{k,j}} \mathcal{N}\left(z_{k,j,i}; \mu_{T_{k,j}(i)}, \phi_{T_{k,j}(i)}\right). \quad (1)$$

Note that coordinate i is defined on the input image I_k and $z_{k,j,i}$ is the intensity or color of pixel i in patch j. Therefore, if pixel i is in two patches $Z_{k,j}$ and $Z_{k,m}$ then $z_{k,j,i} = z_{k,m,i} = I_{k,i}$. In practice, the number of possible mappings $T_{k,j}$ is taken to be equal to the number of

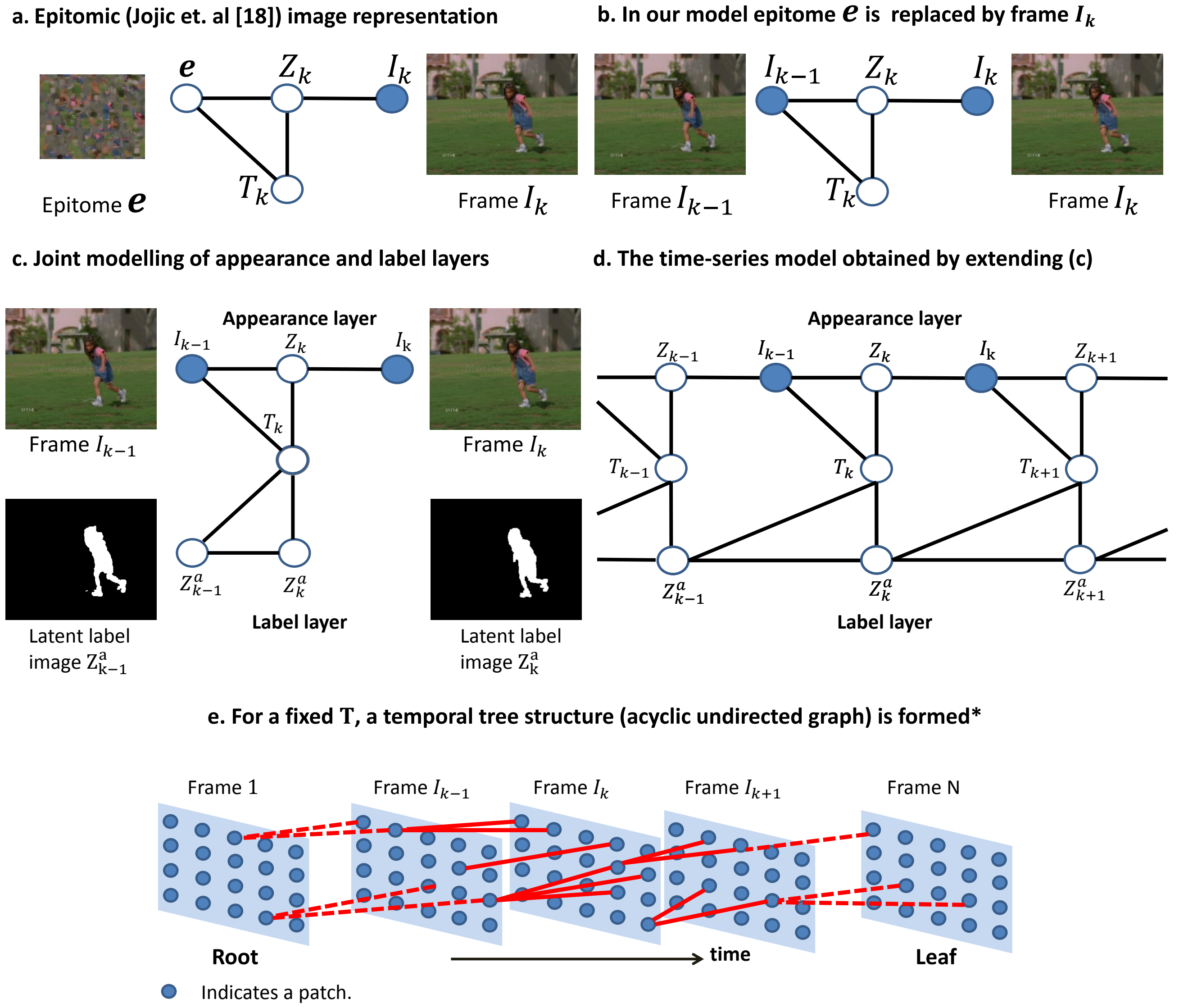


Fig. 2. Step wise build up of our proposed video model for semi-supervised video segmentation. (a) shows the underlying graphical model for the epitomic generative model of a frame I_k. In (b) we replace the epitome of frame I_k by the previous frame in the image sequence. This avoids computationally expensive learning of the epitome. (c) extends (b) to jointly model both frame appearance and corresponding labels. (d) shows the full generative model for the video sequence obtained by repeating the basic model in (c) over time. For a single sample of the mapping variables $T_{1:n}$, we obtain a temporal tree structure as shown in (e). * The term tree structure is used to denote the undirected acyclic graphical model. The tree is rarely a spanning tree and often is a *forest of sub-trees* as shown in (e). For clarity links for all the nodes in (e) are not shown.

coordinates in the epitome.

Treating the patches to be independent, the *generative model for the set of patches* is as follows.

$$p\left(\left\{Z_{k,j}\right\}_{j=1}^{P}, \mathbf{e}, \left\{T_{k,j}\right\}_{j=1}^{P}\right) = p\left(\mathbf{e}\right) \prod_{j=1:P} p\left(T_{k,j}\right) \times \prod_{i \in S_{k,j}} \mathcal{N}\left(z_{k,j,i}; \mu_{T_{k,j}(i)}, \phi_{T_{k,j}(i)}\right). \tag{2}$$

$p\left(T_{k,j}\right)$ is assumed uniform over the possibilities. From this patch generative model, the *image generative model* is defined using patch averaging as follows.

$$p\left(I_{k,i} \middle| \left\{Z_{k,j}\right\}_{j=1}^{P}\right) = \mathcal{N}\left(I_{k,i}; \frac{1}{N_{k,i}} \sum_{j,i \in S_{k,j}} z_{k,j,i}; \psi_{k,i}\right), \tag{3}$$

where $N_{k,i}$ is the number of patches which overlap pixel i. Therefore, the entire epitome model is;

$$p\left(I_k, \left\{Z_{k,j}, T_{k,j}\right\}_{j=1}^{P}, \mathbf{e}\right) = p\left(\left\{Z_{k,j}, T_{k,j}\right\}_{j=1}^{P}, \mathbf{e}\right) \times \prod_{i} p\left(I_{k,i} \middle| \left\{Z_{k,j}\right\}_{1}^{P}\right). \tag{4}$$

So far in this model, the patches $\left\{Z_{k,j}\right\}_{1}^{P}$ have been treated independently, even though their coordinates overlap. Therefore, during inference of the latent patches, the solution space is constrained to those in which overlapping coordinates share the same intensity. This is ensured by estimating a single point posterior distribution at each coordinate (see Sec. 3.2).

Learning an epitome is computationally expensive and the quality of the generated image depends strongly on the size of the epitome. These problems are more severe with video epitomes [19]. Therefore, we avoid learning epitomes and substitute frame I_{k-1} as an epitome for I_k (see Fig. 2(b)). The similarity between frames I_k and I_{k-1} makes I_{k-1} a natural source of patches used to generate I_k. With these changes, (2) is transformed to,

$$p\left(\left\{Z_{k,j}\right\}_{j=1}^{P}, I_{k-1}, \left\{T_{k,j}\right\}_{j=1}^{P}\right) = p\left(I_{k-1}\right) \prod_{j=1:P} p\left(T_{k,j}\right) \prod_{i \in S_{k,j}} \mathcal{N}\left(z_{k,j,i}; I_{k-1,T_{k,j}(i)}, \phi_{T_{k,j}(i)}\right). \tag{5}$$

Latent variable Z_k^a in the *label layer* is the counterpart of latent variable Z_k in the appearance layer (see Fig. 2(c)). $Z_k^a = \left\{Z_{k,j}^a\right\}_{j=1}^{P}$ is seen as a set of labelled patches, each containing labelled pixels from a subset of image coordinates $S_{k,j}$. The common mapping variable $T_{k,j}$ maps coordinates $S_{k,j}$ to coordinates of patches in Z_{k-1}^a. The clique potential, used to encourage label smoothness over subsequent frames, is then defined as:

$$\Psi\left(Z_{k,j}^a, Z_{k-1,T_k(j)}^a; \lambda\right) = \prod_{i \in S_{k,j}} \Psi\left(Z_{k,j,i}^a, Z_{k-1,T_k(j,i)}^a; \lambda\right), \tag{6}$$

where,

$$\Psi\left(Z_{k,j,i}^a = l, Z_{k-1,T_k(j,i)}^a = m; \lambda\right) = \begin{cases} \lambda, \text{if, } l = m, \\ 1 - \lambda, \text{otherwise,} \end{cases} \tag{7}$$

where $l, m \in \mathcal{L}$, with L denoting the label set. λ is a tunable parameter which controls the desirable amount of label smoothness. Notice that in this layer again we have avoided the issue of overlapping coordinates as in the appearance layer for sake of tractable inference. However, unlike the appearance layer, we do not explicitly enforce overlapping coordinates to share the same label by computing a single point posterior. This is because we wish to evaluate the full posterior distribution to estimate label confidences at each image coordinate. Therefore, we average the marginal posteriors of the latent variables which share the same coordinate (see Sec.3.2) and consider this average distribution as the label posterior at that coordinate.

The entire time-series model for the video sequence is now obtained by extending the basic model in Fig. 2(c). This is shown in Fig. 2(d). In this model, for any single state of the mapping variables $T_k = \left\{T_{k,j}\right\}_{j=1}^{P}$, the label layer patches are connected in a *tree structure* as shown in Fig. 2(e). Therefore, the time-series model is a *mixture of trees* graphical model. In this paper, we approximate this mixture by its most probable component to arrive at a *tree structured graphical model* for video sequences (see 3.2).

The full probabilistic joint appearance and label model for video sequences is as given below.

$$\begin{aligned} &p\left(I_{0:n}, Z_{1:n}, Z_{0:n}^a, T_{1:n} | \psi, \phi, \lambda\right) = \\ &\quad \frac{1}{\mathcal{Z}} \prod_{k=1:n} p\left(I_k | Z_k; \psi\right) p\left(Z_k | T_k, I_{k-1}; \phi\right) \times \\ &\quad \Psi\left(Z_k^a, Z_{k-1,T_k}^a; \lambda\right) \Psi_u\left(Z_k^a\right) \Psi_u\left(Z_0^a\right) p\left(T_k\right), \end{aligned} \tag{8}$$

with (3), (5), (6) defining the first three terms of the right hand side and $\mathcal{Z}$ the proportionality constant. The unary terms are defined as follows.

$$\Psi_u\left(Z_k^a\right) = \prod_{j=1:P} \prod_{i \in S_{k,j}} \Psi_u\left(Z_{k,j,i}^a\right). \tag{9}$$

The prior on the mapping variable $p(T_{k,j})$ is set to a uniform distribution within a rectangular window in frame $k-1$ around the center coordinate of patch $Z_{k,j}$.

The model in [24] had a redundant clique potential (the fusion clique) which was used to fuse an external classifier prediction with the bottom time-series chain. Our model in (8) discards this potential without affecting performance. Another latent variable representing the classifier prediction in [24] is also discarded and instead classifier predictions are set as unaries(see Sec. 3.3) without change in performance.

3.2 Inference

It is clear from (8) that the proportionality constant $\mathcal{Z}$ cannot be computed due to the combinatorial sum involved on the right hand side. Therefore, we resort to approximate inference. The log probability of the observed data V (images, labelled start and end frames) can be lower bounded as follows [30]:

$$\log p(V | \Xi = \{\psi, \phi, \lambda\}) \geq \int_H q(H) \log \frac{p(V, H | \Xi)}{q(H)}, \tag{10}$$

where $q(H)$ is a variational posterior over the latent variables in the model. We choose,

$$q(H) = q_1(\mathcal{T}) q_2(\Theta), \tag{11}$$

where $\Theta = \left\{Z_{1:n}, Z_{1:n-1}^a\right\}$, $\mathcal{T} = T_{1:n}$, and,

$$\begin{aligned} q_1(\mathcal{T}) &\triangleq \prod_{k=1}^{n} \prod_{j=1}^{\Omega} q_1(T_{k,j}) \\ q_2(\Theta) &\triangleq \prod_{k=1}^{n} \prod_{j=1}^{\Omega} \prod_{i \in j} \delta_{Z_{k,j(i)}^*}(Z_{k,j(i)}) \tilde{q}_2(\Theta_{\setminus Z_{1:n}}). \end{aligned} \tag{12}$$

The form of $q(H)$ is chosen as a compromise between performing tractable inference and retaining as much structure as possible in the posterior. Notice from the above equation that the variational posterior does not factorise into independent terms (over the latent variables Θ) as in a mean-field approximation [30]. Therefore, our approximation is a *structured variational posterior*, which leads to better performance [31]. Secondly, notice the single point posterior approximation over the latent variables $Z_{1:n}$. This ensures that overlapped coordinates have the same value.

We now apply the calculus of variations [30] to maximise the lower bound w.r.t q_1, q_2 and arrive at,

$$q_1(T_{k,j}) \propto \exp\left\{\int_{Z_{k,j},Z^a_{k,j},Z^a_{k-1,T_{k,j}}} \tilde{q}_2(Z^a_{k,j}, Z^a_{k-1,T_{k,j}}) \times \right.$$
$$\left. \log\left[\Psi(Z^*_{k,j}, I_{k-1,T_{k,j}};\phi)\Psi(Z^a_{k,j}, Z^a_{k-1,T_{k,j}};\lambda)\right]\right\} p\left(T_{k,j}\right), \tag{13}$$

$$\tilde{q}_2(\Theta_{\setminus Z_{1:n}}) = \exp\int_{\mathcal{T}} q_1(\mathcal{T}) \log p(\Theta_{\setminus Z_{1:n}}|V,\mathcal{T};\Xi). \tag{14}$$

The second of the above fixed point equations is computationally intensive as it involves marginalising over all the mapping variables. For this reason, we approximate it by,

$$\tilde{q}_2(\Theta_{\setminus Z_{1:n}}) \approx \exp\int_{\mathcal{T}} \delta_{\mathcal{T}^*}(\mathcal{T}) \log p(\Theta_{\setminus Z_{1:n}}|V,\mathcal{T};\Xi),$$
$$= p(\Theta_{\setminus Z_{1:n}}|V,\mathcal{T}^*;\Xi) \tag{15}$$

where $\mathcal{T}^* = \mathrm{argmax}_{\mathcal{T}}\, q_1(\mathcal{T})$. A second motivation for this approximation is that $p(\Theta_{\setminus Z_{1:n}}|V,\mathcal{T}^*;\Xi)$ is *temporally tree structured*. From a variational inference viewpoint, $\mathcal{T}^*$ represents the best (MAP) tree structured component of the mixture model. We exploit this temporal tree structure to perform efficient and *exact inference* of the latent variables in the set $\Theta_{\setminus Z_{1:n}}$. Notice that $\tilde{q}_2(\Theta_{\setminus Z_{1:n}})$ is a joint distribution over the MAP tree and thus the exact marginal posteriors are easily computed using standard sum-product belief propagation [30].

We also emphasize that the tree structure need not be a spanning tree. Indeed, we employ the term tree structured to mean an undirected acyclic graph on which exact inference can be performed. In practice, there can be several disjoint trees in the model or a forest of non-spanning trees (see Fig. 2).

3.2.1 From Patches to Pixel Posteriors

So far in the inference, we have exploited the best tree structure to compute the marginal posteriors of variables $z^a_{k,j,i'}$, where i is the image coordinate. As mentioned in Sec. 3.1, since patches share coordinates (overlap), we average the marginal posteriors of all latent variables which share the same coordinate. For example,

$$\hat{q}_2(z^a_{k,i}) \approx \frac{1}{N_{k,i}} \sum_{j,i\in S_{k,j}} \tilde{q}_2(z^a_{k,j,i}), \tag{16}$$

where $\hat{q}_2(z^a_{k,i})$ is the averaged posterior. Notice that the patch index is now removed on the left hand side.

3.2.2 Forward and Backward Trees

From Fig. 2(e), we see that the tree has its root in the start frame and leaves at the end frame. We denote this as the forward tree. This directionality in the temporal structure can sometimes lead to a labelling bias. For example, the user provided root frame labels can have a stronger influence than the leaf (end) frame labels on the remaining latent variables. To correct for this bias, we compute the best tree in the reverse direction (a backward tree with root at the end frame) and perform inference on it. Finally, we average the label posteriors from the two trees at each coordinate to obtain the approximate posterior at each pixel.

Algorithm 1: Semi-supervised Video Segmentation using Tree Structured Graphical Models.

```
Input: $I_{0:n}$ (video), hand labelled key frame(s) or user
    mouse strokes.
Output: Pixel label probabilities.
Intialisation
Set $Z^a_0, Z^a_n$ to user provided labels and
$Z_k = I_k, k \in 1:n$.
Set the initial values of $\lambda, \psi, \phi$ to ones given in Sec. 4.
Set unaries to uniform distributions.
Set $p\left(T_k\right), k \in 1:n$ to uniform distributions.
Inference
1. Compute the forward and backward trees using (13).
2. For each tree separately
  a. Use sum-product algorithm [30] to obtain pixel
  marginals.
  b. Obtain the coordinate wise approximate
  marginals by averaging (Sec. 3.2.3).
3. Optionally smooth the pixel labels in each frame
using loopy BP (Sec. 3.2.3).
4. Average the posteriors at each coordinate from both
the trees.
Learning Unaries
5. Learn unaries using the soft label Random Forest
trained with the label posteriors (Sec. 3.3).
Bootstrapped Inference
6. Repeat steps 1-4 using the learnt unaries. An
example of step-wise results is shown in Fig.3.
```

It can be argued that since our model is undirected (no temporal directionality is intended), the forward and backward tree could be combined into a single undirected model. However, this model would have a loopy temporal structure and undesirably would not permit efficient and exact inference of pixel labels.

3.2.3 Intra-frame Smoothing of Pixel Labels

We can *optionally* obtain a smooth, yet edge sensitive, labelling in each frame by using the pixel posteriors computed thus far as pixel unaries and applying loopy BP [30] on a standard 8-neighbourhood grid. We use contrast sensitive edge potentials as in [5] and 50 iterations of message passing. The resulting marginals provide us with label confidences (see Fig. 3(c)). The drawback of performing smoothing is that the marginals tend to be over confident (see Fig. 3(d)). This is undesirable, for example, in long sequences new objects tend to appear and inference should ideally assign low confidence to them in order to reduce false positive labelling. Therefore, we avoid smoothing in long sequences.

3.3 Learning Pixel Unaries

In the first iteration of inference, we set the unaries to uniform distributions and use our proposed inference technique to estimate the pixel label posteriors. We then train a Random Decision Forest [21] using these posteriors as soft pixel labels, i.e each pixel has a vector label instead of a scalar class label. We term this *semi-supervised* Random Forest the soft label Random Forest

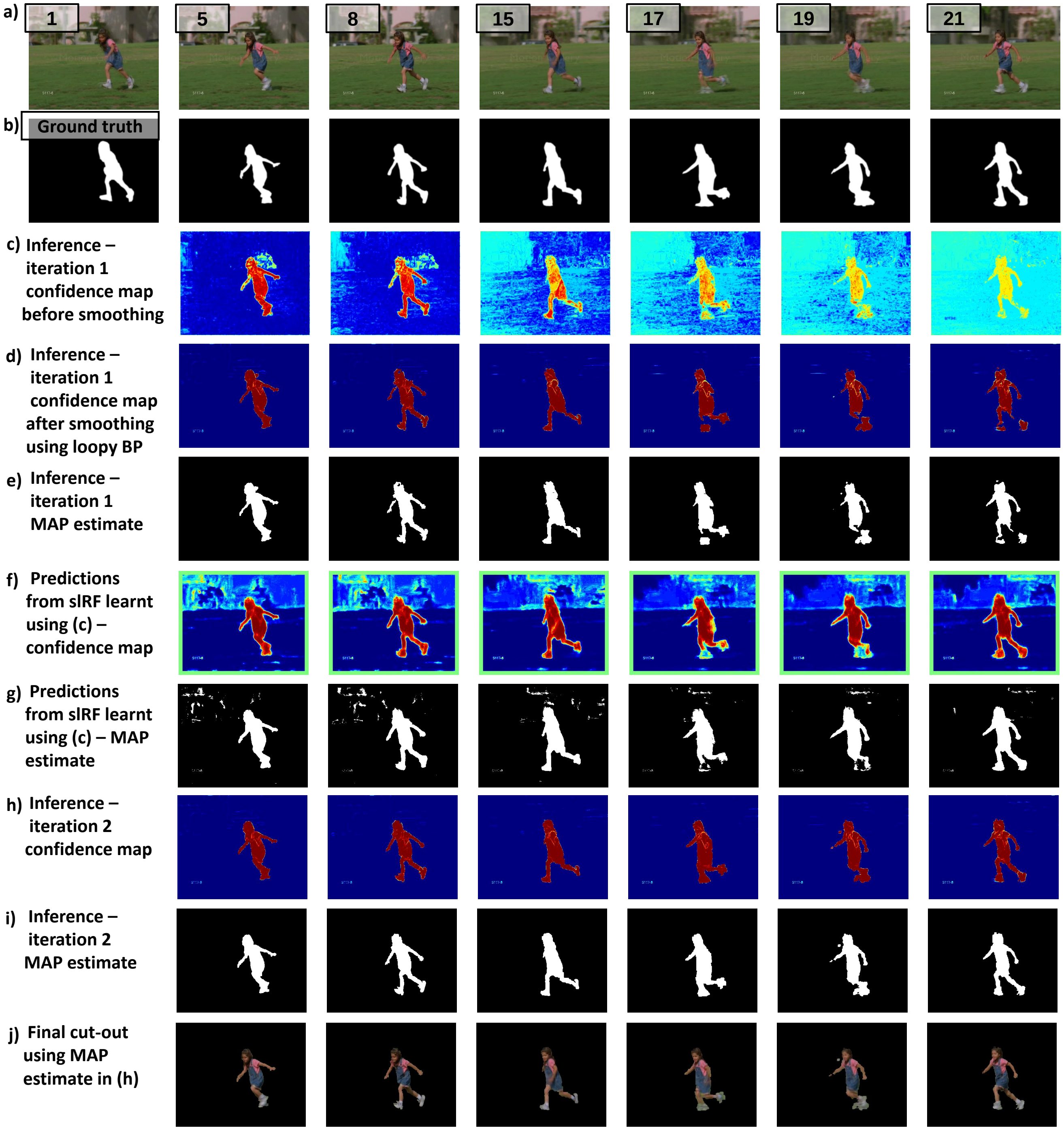

a)
1
5
8
15
17
19
21
b)
Ground truth
c) Inference – iteration 1 confidence map before smoothing
d) Inference – iteration 1 confidence map after smoothing using loopy BP
e) Inference – iteration 1 MAP estimate
f) Predictions from slRF learnt using (c) – confidence map
g) Predictions from slRF learnt using (c) – MAP estimate
h) Inference – iteration 2 confidence map
i) Inference – iteration 2 MAP estimate
j) Final cut-out using MAP estimate in (h)

Fig. 3. The first two rows show the image sequence (moving camera) and ground truth from the SegTrack dataset [13]. The segmentation algorithm in this sequence has to cope with fast shape changes, motion blur and overlap between foreground and background appearance models. Inferred marginals of $Z^a_{1:n-1}$ before smoothing are shown in row (c). Note how the confidence decreases from the labelled first frame. The marginals after smoothing are shown in row (d). Observe the increased confidence due to smoothing. The MAP estimates of these marginals are shown in row (e). Note that some part of the girl's hands and leg are missing. The unaries learnt using the marginals in row (c) are shown in row (f); and its MAP estimate is shown in row (g). We see from (h) that the legs and hands are labelled correctly along with some false positive background labels. Bootstrapping this prediction and performing inference once again results in the confidences shown in row (g). The corresponding MAP estimate in row (i) shows almost no background false positives, and the missing legs and hand in row(d) are recovered. The cut-out in row (j) has sharp edges and is clean. Zoom-in and view in color for best results.

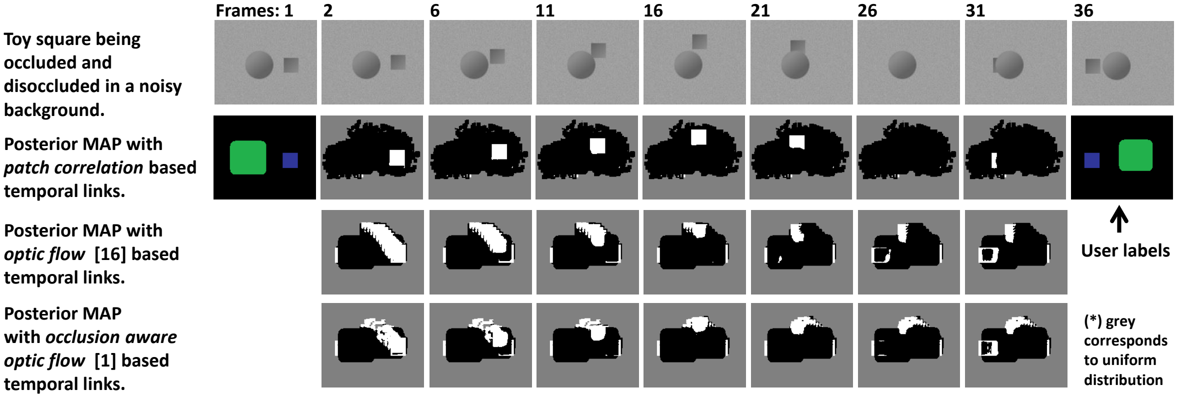

Frames: 1
2
6
11
16
21
26
31
36
Toy square being occluded and disoccluded in a noisy background.
Posterior MAP with *patch correlation* based temporal links.
Posterior MAP with *optic flow* [16] based temporal links.
Posterior MAP with *occlusion aware optic flow* [1] based temporal links.
User labels
(*) grey corresponds to uniform distribution

Fig. 4. This figure demonstrates the adverse effect of using *occlusion unaware* optic flow to establish inter-frame correlations. The dis-occluded background parts get linked to foreground parts causing a *drag effect*. *Occlusion aware* optic flow is better, but there still remains mislabelling. Our patch cross correlation based method handles dis-occlusions robustly by linking similar textured patches in the background. Zoom-in and view in color for best results.

(slRF). We use simple but computationally efficient pixel intensity difference features at each split node as in [14].

Our semi-supervised training of the Random Forest is conceptually different from the transductive forest described in [32]. In the transductive forests, labelled and unlabelled data are treated separately and a new information gain criterion is introduced to combine label and appearance based entropies. In contrast, we first assign each unlabelled data point a soft label obtained from the earlier label inference step. At training time, we compute a histogram of soft labels at each node by element wise addition of the vector labels and use the criterion of [14] to evaluate the split function.

We use the predictions from the slRF as pixel unaries in the second interation of inference. These unaries, learnt in a semi-supervised manner can help improve segmentation accuracy, as shown in Fig. 3(g,h). Unlike traditional tracking algorithms were unaries are learnt using the first frame labels, we use the entire video data and their inferred labels to learn unaries.

In Fathi et al. [10], labels are propagated to the adjacent frame and their MAP estimate is used to update the unary. This can be sub-optimal as all the frames are not used to update the unary (especially if the background is changing). In contrast, our efficient inference method permits us to use all the frames to learn the unaries.

4 Experiments and Results

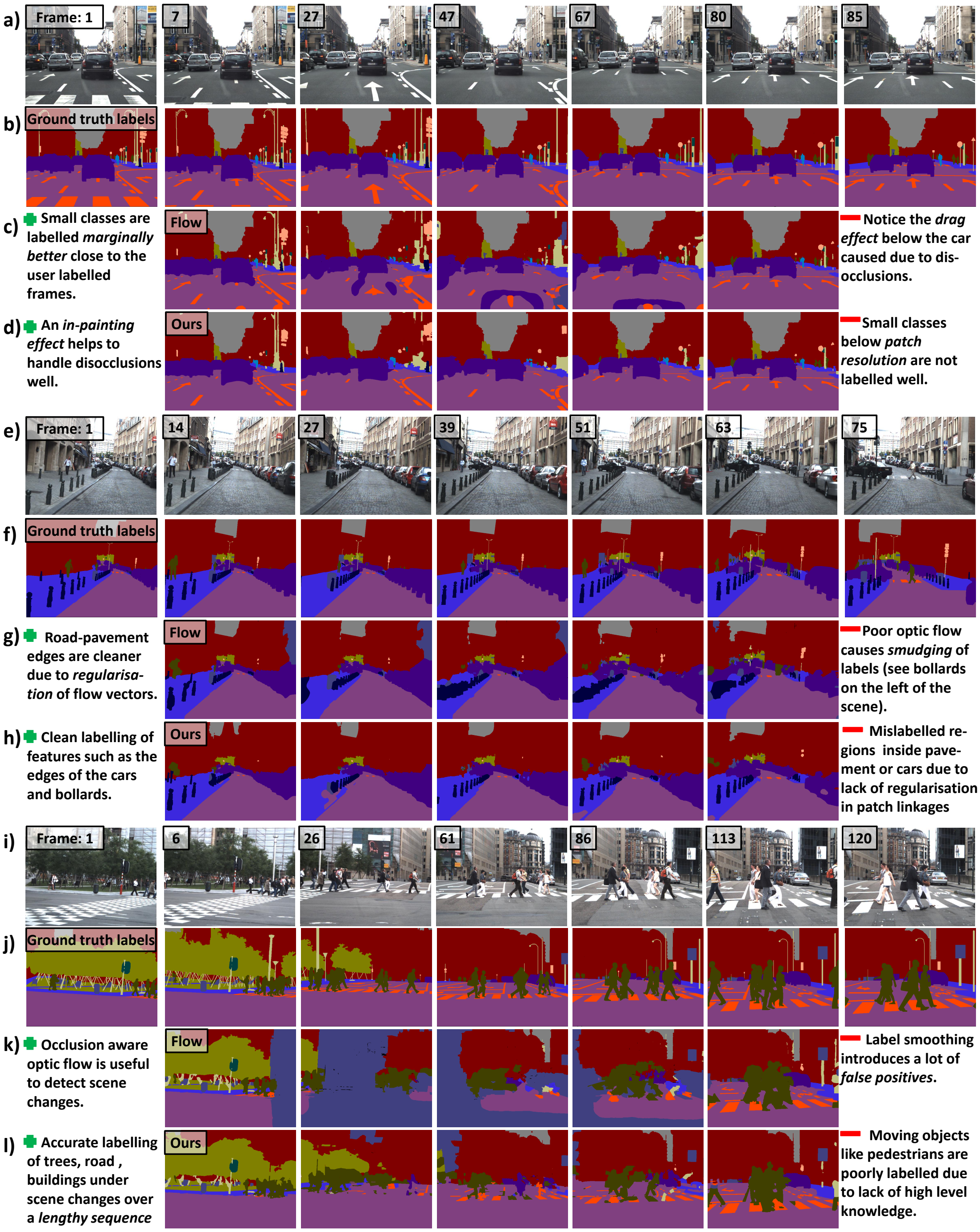
We perform three experiments to bring out the pros and cons of our approach. In all the experiments, each colour channel in all the images is scaled to lie between $[0.0, 1.0]$. We use patches of size 7×7 centred on each pixel. In our tree model, we set λ to 0.9.

4.1 Patch Mappings versus Optical Flow for Label Propagation

In this experiment, we compare the performance of our patch-based model against a pixel based model where inter-frame mappings are estimated using a state of the art occlusion aware optical flow module [33]. This algorithm produces an occlusion probability at each pixel and all vectors below a probability of 0.2 are ignored. To incorporate this flow module into our framework, we replace the inferred MAP estimates of the patch mapping variables ((15)) by the *rounded-off* flow vectors. Note that we still obtain a tree structured model by using the flow vectors to link frames. The inference is then performed as described in Alg. 1 without updating the unaries.

Dataset: We use a challenging dataset consisting of three outdoor driving video sequences (VGA resolution) captured using a camera placed on a car. The ground truths are available for 70 frames for Seq 3 & Seq1, and 98 frames of Seq 2. 15 different classes are hand labelled which cost about 45-60 minutes per frame.

Results and Discussion: Badrinarayanan et al. [27], Chuang et al. [34] have earlier brought to notice the problems associated with using optical flow for label propagation. Particularly, when using *forward flow*, their remains unlabelled pixels (holes) after a disocclusion (as there is no correspondence found). When using *reverse flow*, the *label dragging* effect happens after a disocclusion, because newly appearing pixels are forced into a wrong match in the previous frame (see Fig. 4). In contrast, our un-regularised patch mappings aid label transfer from frame to frame using an *in-painting* like step. As the reappearing patches (disocclusion) are *constructed* using a combination of patches from the previous frame they help fill in the correct labels. Secondly, unlike flow based methods we do not attempt to compute pixel accurate motion, but instead we use overlapping patches at each pixel to correct for mismatches in greedy patch matching across subsequent frames. For example, if a patch size of 7x7 is used then each pixel is overlapped by 49 patches provided the patch centers are a pixel apart. If atleast 50% of the overlapping patches match to the correct class then our method delivers sharp class boundaries. These subtle but key differences are the merits of our proposed segmentation approach. We note that some effort has been made to correct the problems associated with optic



c) + Small classes are labelled *marginally better* close to the user labelled frames.

− Notice the *drag effect* below the car caused due to dis-occlusions.

d) + An *in-painting effect* helps to handle disocclusions well.

− Small classes below *patch resolution* are not labelled well.

g) + Road-pavement edges are cleaner due to *regularisation* of flow vectors.

− Poor optic flow causes *smudging* of labels (see bollards on the left of the scene).

h) + Clean labelling of features such as the edges of the cars and bollards.

− Mislabelled regions inside pavement or cars due to lack of regularisation in patch linkages

k) + Occlusion aware optic flow is useful to detect scene changes.

− Label smoothing introduces a lot of *false positives*.

l) + Accurate labelling of trees, road , buildings under scene changes over a *lengthy sequence*

− Moving objects like pedestrians are poorly labelled due to lack of high level knowledge.

Fig. 5. Comparison of multi-class segmentation using our patch-based temporal linkage and an off the shelf occlusion aware optic flow based linkage [33], within our framework. This Toyota driving dataset was manually labelled each frame and will be made available upon request. Unlike optic flow, the patch-based linkage is unregularised (patch mapping variables are assumed independent) and handles occlusions/disocclusions better. However, optic flow performs marginally better on smaller classes which are below patch resolution. See Fig. 6 for corresponding quantitative results. Zoom-in and view in color for best results.

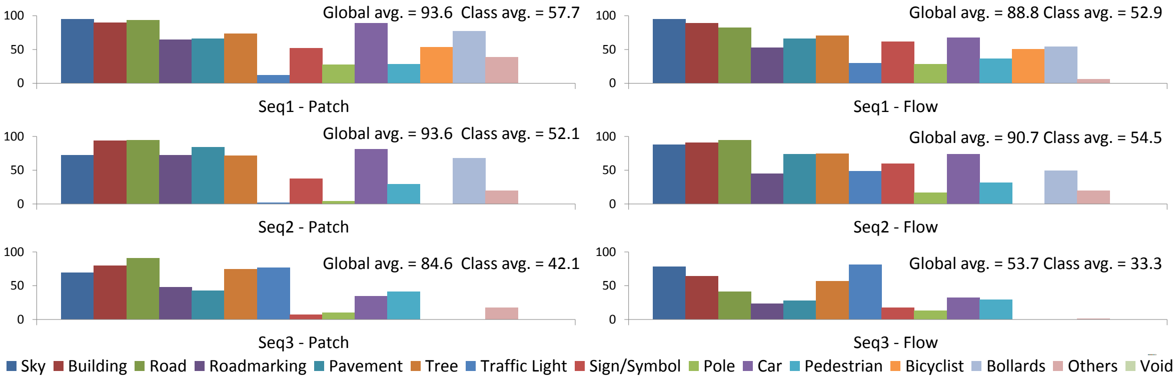


Fig. 6. Quantitative comparison of our patch-based temporal linkage versus optic flow based linkage over 15 classes. In all three sequences, the global average accuracy (total percentage of correctly labelled pixels) exceeds the optic flow based linkage. Except in Seq2 the class average (average of per class accuracy) is higher. Note that void class corresponds to unlabelled parts of the first and last frame. We also improve on the results reported in [27].

Patch size	**3x3**		**7x7**		**11x11**	
Window size	Global average	Class average	Global average	Class average	Global average	Class average
27x20	94.06	60.57	93.72	59.06	93.52	58.55
40x30	93.85	58.80	93.64	57.72	93.70	57.09
53x40	93.41	57.60	93.48	56.86	93.68	57.47
67x50	93.11	55.28	93.32	54.55	93.60	57.22
80x60	92.87	53.84	93.16	53.43	93.53	56.12

Fig. 7. Change in accuracy as a function of patch size (columns) and window size (rows) for Seq 1. Observe the smooth decrease in both accuracies as window size increases relative to any patch size. This is caused due to small sized classes being mislabelled and hence the global accuracy decreases less than the class average accuracy.

flow by computing heuristic reliability measures for flow and incorporating an external object appearance model to correct for the dragging effect (see [35]). However, this was not the goal of this experiment.

The results of our quantitative study are reported in Fig. 6 and some image samples are shown in Fig. 5. Our proposed patch-based method outperforms the flow based approach in terms of the overall correctly classified pixels (global average) and, except for one sequence, it is also better in terms of average of class accuracies. However, our method is not consistently better than the flow based method over all classes. For instance, classes smaller than or about the same as the patch resolution (sign/symbol, traffic lights) are poorly labelled. A hybrid of the two methods can potentially tackle this issue.

We used Seq1 to study the effect of patch and search window size on accuracy. From Fig. 7, it can be seen that for any patch size, there is a gradual drop in accuracy as the window size increases, although for smaller patch sizes the drop is steeper. The global accuracy is more stable than class average accuracies as small sized classes are mislabelled with larger window sizes.

4.2 Label Uncertainty Propagation versus Instantaneous Decision Propagation

In our approach, we propagate label uncertainties over time by performing inference on the tree structured graphical model. Here we compare this with serial propagation of instantaneous (per frame) label decisions as in the approach of [37] ([35], [12] are similar in principle). This experiment brings out the inadequacies of instantaneous decision making and demonstrates the need to propagate label uncertainties. These label uncertainties are useful not only to avoid false positive labelling but also for bootstrapped learning of the slRF.

We choose the 1^{st} stage Random Forest (RF) classifier [14] with 16 trees, each of depth 10. Input LAB space patches of 21×21 are extracted around every 5^{th} pixel on both axis. We leave out border pixels in a 12 pixel band to fit all the rectangular patches. We use the same kind and number of features as in [14]. The key difference is that we use the inferred *soft labels* to train the slRF (see Sec. 3.3). We compute the split function information gain and the leaf node distributions by treating the data point label as a *vector* whose elements sum to unity.

Dataset: Our test sequences are taken from the CamVid road scene dataset [36]. Each sequence in CamVid is 750 frames in length, but we down sample to every 5^{th} frame to have a length of 150 frames. Ground-truth is available every 30 frames. We study 9 static classes like sky, road, etc. and treat moving objects such as cars, pedestrians as outliers, as they are not permanent in a road scene. We assign a "uniform" distribution to these outlier classes in the labelled frames and examine their false positive rate to gain insight into outlier rejection performance. As the sequences are lengthy, we avoid label smoothing for reasons discussed in Sec. 3.2.3.

Results and Discussion: Qualitatively, the main

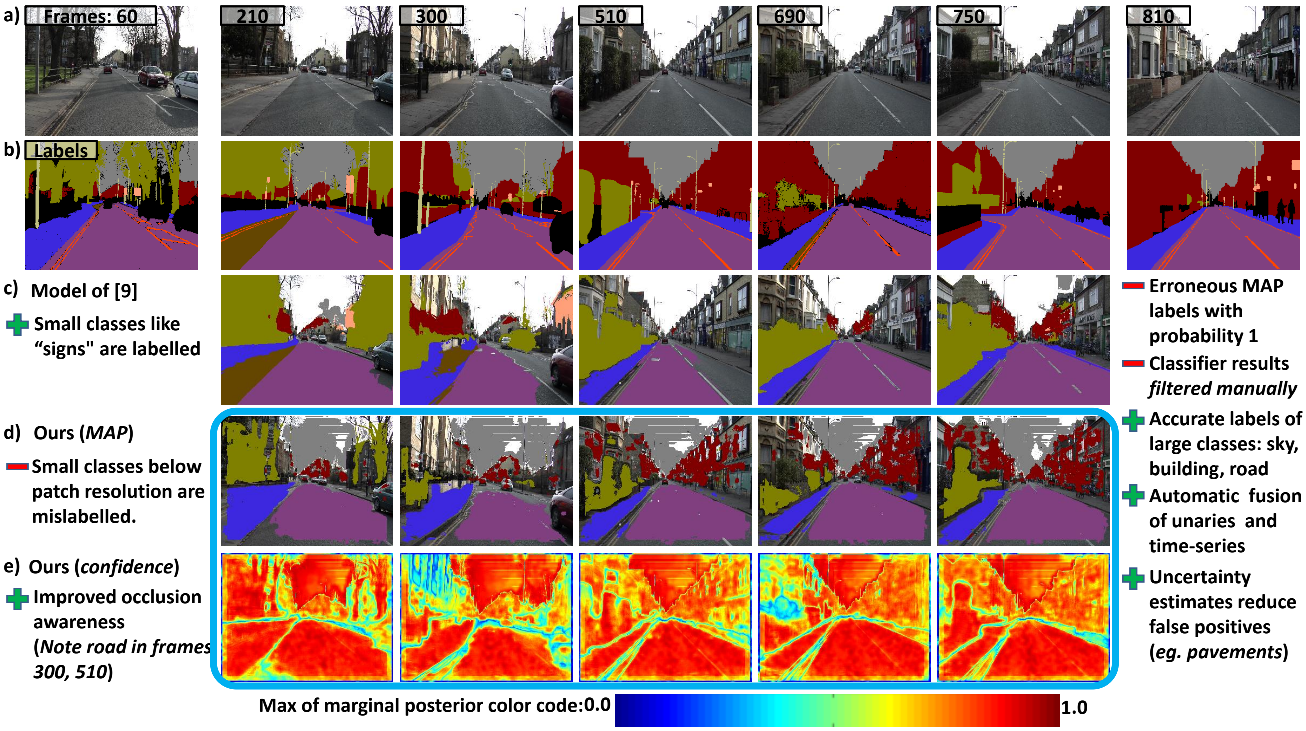
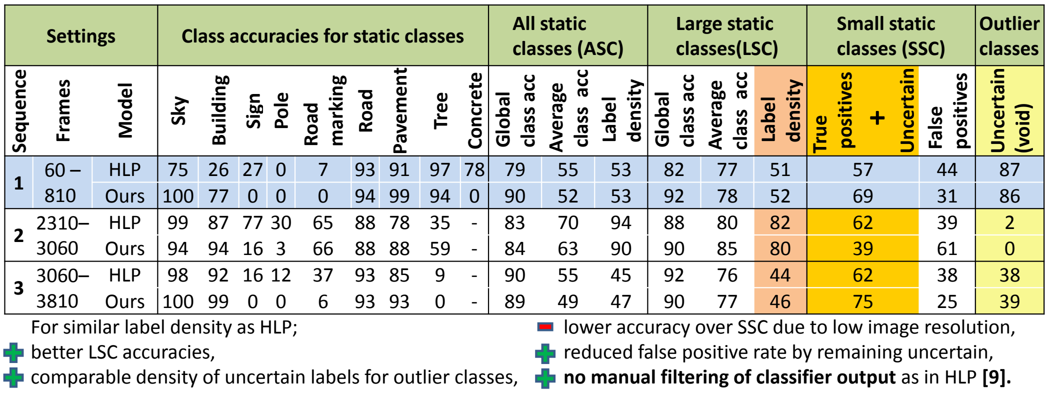

a) Frames: 60
210
300
510
690
750
810
b) Labels
c) Model of [9]
Small classes like "signs" are labelled
d) Ours (*MAP*)
Small classes below patch resolution are mislabelled.
e) Ours (*confidence*)
Improved occlusion awareness (*Note road in frames 300, 510*)
Erroneous MAP labels with probability 1
Classifier results *filtered manually*
Accurate labels of large classes: sky, building, road
Automatic fusion of unaries and time-series
Uncertainty estimates reduce false positives (*eg. pavements*)
Max of marginal posterior color code:0.0
1.0

Fig. 8. Seq05VD from the CamVid dataset with ground-truth [36]. Black "outlier" labels at the ends have uniform distributions. We encourage the reader to view the labels along with the confidence map in row (f) to see that our approach reduces false positive labelling. Zoom-in and view in color for best results.



Settings			Class accuracies for static classes									All static classes (ASC)			Large static classes(LSC)			Small static classes (SSC)		Outlier classes
Sequence	Frames	Model	Sky	Building	Sign	Pole	Road marking	Road	Pavement	Tree	Concrete	Global class acc	Average class acc	Label density	Global class acc	Average class acc	Label density	True positives + Uncertain	False positives	Uncertain (void)
1	60 –	HLP	75	26	27	0	7	93	91	97	78	79	55	53	82	77	51	57	44	87
	810	Ours	100	77	0	0	0	94	99	94	0	90	52	53	92	78	52	69	31	86
2	2310–	HLP	99	87	77	30	65	88	78	35	-	83	70	94	88	80	82	62	39	2
	3060	Ours	94	94	16	3	66	88	88	59	-	84	63	90	90	85	80	39	61	0
3	3060–	HLP	98	92	16	12	37	93	85	9	-	90	55	45	92	76	44	62	38	38
	3810	Ours	100	99	0	0	6	93	93	0	-	89	49	47	90	77	46	75	25	39

For similar label density as HLP;
✚ better LSC accuracies,
✚ comparable density of uncertain labels for outlier classes,

▬ lower accuracy over SSC due to low image resolution,
✚ reduced false positive rate by remaining uncertain,
✚ **no manual filtering of classifier output** as in HLP **[9].**

Fig. 9. Quantitative comparison on complex and lengthy (750 frames) video sequences from CamVid [36] dataset. Unlike our method, the Hybrid Label Propagation method of [37] uses manual unary monitoring to avoid false positive labelling.

advantage of propagating label uncertainties is the reduction in false positives and better occlusion awareness (see Fig. 8). In the model of [37], if the MAP estimate of temporal predictions and the unary predictions disagree then the label is set to "unknown". This is because their directed model induces a *competition* between the two kinds of predictions [28]. This necessitates manual filtering of the classifier predictions to reduce false positives. The results in Fig. 9 demonstrate better accuracy over large static classes (road, sky, building, pavements) for a similar label density (varying thresholds over confidence produces different label density). The false positive rate is also lower than that of [37]. However, accuracy decreases over smaller classes comparable in size to the patch resolution.

4.3 Joint Tracking and Segmentation

We evaluated the performance of our approach in a joint tracking and segmentation challenge using the SegTrack [38],[13] dataset. This ground truthed dataset consists of 6 sequences, captured with a moving camera, with clutter, self-occlusion, small sized objects and deformable shape (see Fig. 10). The first frame of each sequence is user labelled into a foreground and background category. For this experiment, we used a patch size of 3×3 in the temporal tree structured model and slRF was of depth 8.

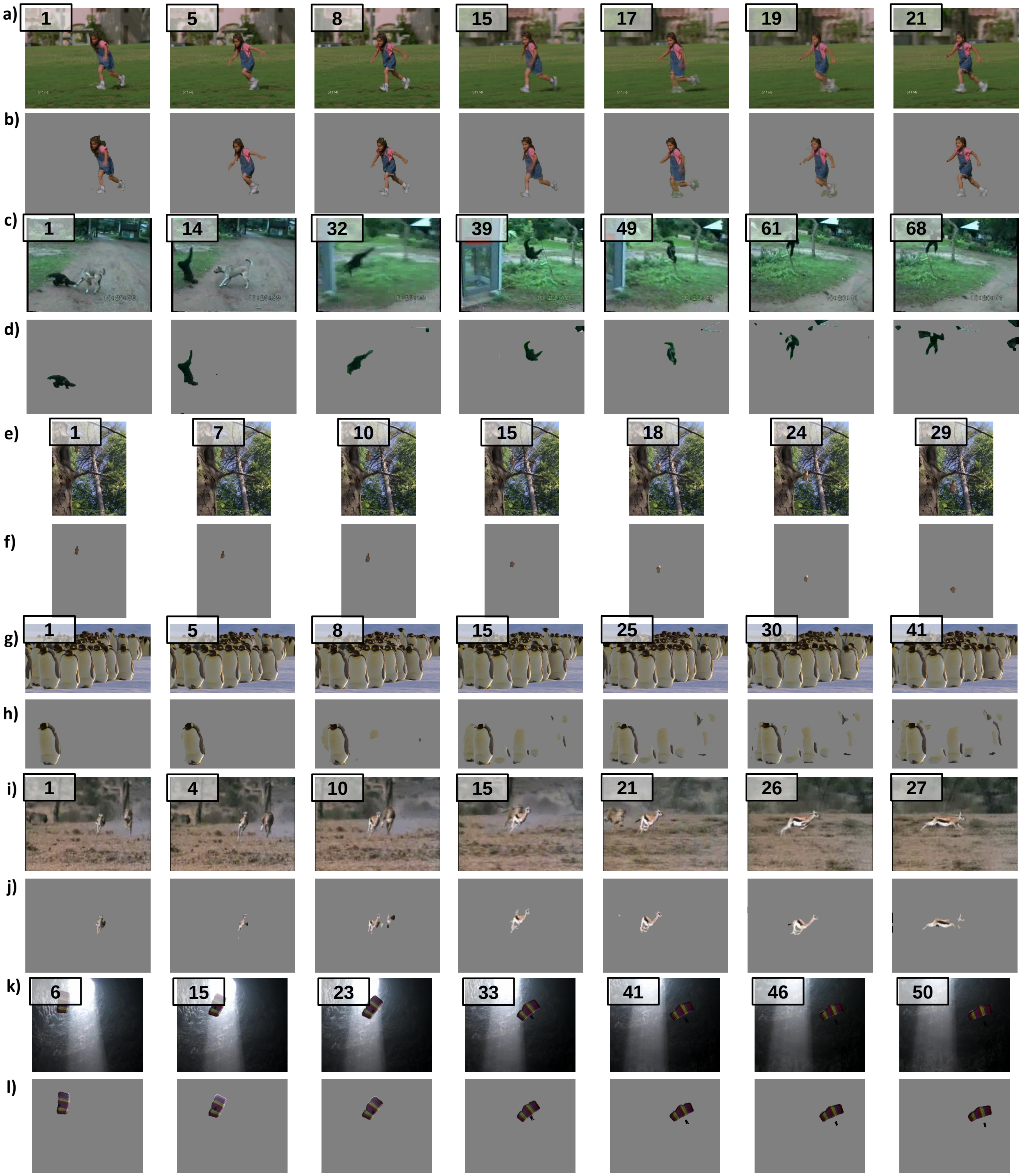
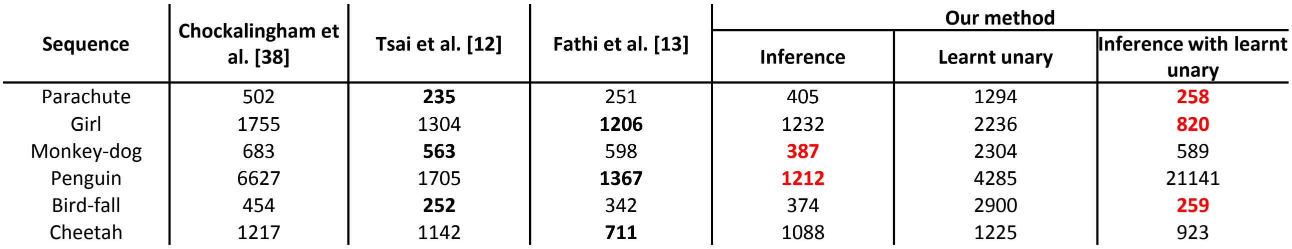


Fig. 10. Qualitative results on the SegTrack dataset [13]. In all these experiments only the start frame of the video sequence is user labelled. Notice how our algorithm is able to cope with motion blur (a), large displacement (d,j), small sized objects (f). The main failure case is (h) due to severe overlap in appearance between the foreground and background. Note that we applied label smoothing for this dataset. Zoom-in and view in color for best results.



Sequence	Chockalingham et al. [38]	Tsai et al. [12]	Fathi et al. [13]	Our method		
				Inference	Learnt unary	Inference with learnt unary
Parachute	502	**235**	251	405	1294	**258**
Girl	1755	1304	**1206**	1232	2236	**820**
Monkey-dog	683	**563**	598	**387**	2304	589
Penguin	6627	1705	**1367**	**1212**	4285	21141
Bird-fall	454	**252**	342	374	2900	**259**
Cheetah	1217	1142	**711**	1088	1225	923

Fig. 11. Quantitative evaluation on the SegTrack tracking and segmentation dataset [13]. In all these experiments only the start frame of the video sequence is user labelled. We used a single set of model parameters to obtain these results. The score is the average label mismatch per frame. Our score is better or marginally worse off in five out of the six sequences as compared to the state of the art methods. Note how the score improves after each stage of our method. In the Penguin sequence, inference without unaries outperfoms the other methods. However, poor unary accuracy results in performance degradation after bootstrapping the learnt unary for a second round of inference. See Fig.10 for qualitative results.

In Fig. 11 we report our score along with some of the recent state of the art methods. In five out of the six sequences we perform better or are marginally worse off than the competitors. In the remaining case (Penguin), inference without the learnt unaries outperforms the state of the art. The unaries in this case are very poor due to severe overlap between foreground and background. Fathi et al. [10] design an adaptive weighting scheme to adapt the contribution of the unaries. Such an approach can also be incorporated into our framework in the future.

5 Advantages and Drawbacks

The key **advantages** of our proposed approach are:

1) Our temporal tree structured model permits exact and efficient inference of pixel labels.
2) We avoid sequential propagation of erroneous instantaneous decisions and therefore reduce false positives (see [24] for quantitative arguments).
3) We avoid the use of short time-window based processing which are currently used in several video segmentation approaches [13], quite often due to computational inefficiency.
4) We can learn unaries in a semi-supervised manner using the results of inference to improve segmentation accuracy.

Our approach suffers from the following **drawbacks**:

1) We are currently restricted to segment classes which have sizes above the patch resolution of 7×7. Using higher resolution images should alleviate this problem to a large extent.
2) Our method cannot handle motion larger than half the search window size.
3) The uncertainty in the pixel marginal posteriors is based on the number of pairwise cliques a patch is part of (its neighbourhood connectivity), and does not include the uncertainty with which the clique was formed in the tree model. In future, this should be included to improve performance
4) The method is currently not real-time (See Fig. 12). It take upto 1.8GB per frame for an image resolution of 640×480 and a 15 class problem using our unoptimised C++ implementation. This entails several disk read-write operations.

6 Conclusions

We presented a novel tree structured graphical model for multi-class semi-supervised video segmentation. In this model, the video time-series is modelled as a temporal tree which links patches from the first frame to the last frame. The tree structure permits efficient and exact inference of pixel labels and their confidences. We demonstrated that in several cases robust pixel unaries can be learnt directly from pixel marginal posteriors and help improve segmentation. One of the other key benefits is the ability to propagate label uncertainties over time using exact inference. This is in contrast to most existing approaches which use short time-window based processing and sub-optimal instantaneous decision making. As part of our future work, we would like to address the issues surrounding small sized classes and ease the computational burden of the algorithm.

References

[1] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *CVPR*, 2008.

[2] E. B. Sudderth and M. I. Jordan, "Shared segmentation of natural scenes using dependent pitman-yor processes," in *NIPS*, 2008, pp. 1585–1592.

[3] X. Bai, J. Wang, D. Simons, and G. Sapiro, "Video snapcut: robust video object cutout using localized classifiers," in *ACM SIGGRAPH*, 2009, pp. 70:1–70:11.

[4] Y. Li, J. Sun, and H.-Y. Shum, "Video object cut and paste," *ACM Trans. Graph.*, vol. Vol. 24, pp. pp. 595–600, 2005.

[5] C. Rother, V. Kolmogorov, and A. Blake, "Grabcut: Interactive foreground extraction using iterated graph cuts," *ACM Transactions on Graphics*, vol. 23, pp. 309–314, 2004.

[6] A. Vezhnevets, V. Ferrari, and J. M. Buhmann, "Weakly supervised structured output learning for semantic segmentation," in *CVPR*, 2012.

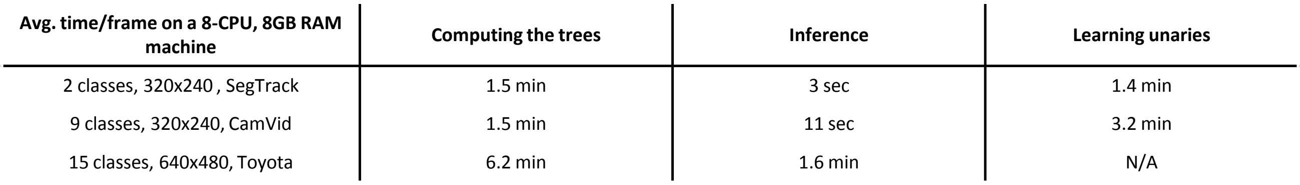
[7] A. Vazquez-Reina, S. Avidan, H. Pfister, and E. Miller, "Multiple hypothesis video segmentation from superpixel flows," in *ECCV*, 2010.

[8] J. Lezama, K. Alahari, J. Sivic, and I. Laptev, "Track to the future: Spatio-temporal video segmentation with long-range motion cues," in *CVPR*, 2011.

[9] Y. J. Lee, J. Kim, and K. Grauman, "Key-segments for video object segmentation," in *ICCV*, 2011.

[10] A. Fathi, M. Balcan, X. Ren, and J. M. Rehg, "Combining self training and active learning for video segmentation," in *BMVC*, 2011.

[11] Y. Boykov and M. P. Jolly, "Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images," in *ICCV*, 2001.



Avg. time/frame on a 8-CPU, 8GB RAM machine	Computing the trees	Inference	Learning unaries
2 classes, 320x240 , SegTrack	1.5 min	3 sec	1.4 min
9 classes, 320x240, CamVid	1.5 min	11 sec	3.2 min
15 classes, 640x480, Toyota	6.2 min	1.6 min	N/A

Fig. 12. Typical computational load of our method for different image resolutions and number of classes. Here we assume the arrays holding the marginals, BP messages, are loaded into the RAM.

[12] P. Kohli and P. Torr, "Efficiently solving dynamic markov random fields using graph cuts," in *ICCV*, 2005, pp. II: 922–929.

[13] D. Tsai, M. Flagg, and J. M. Rehg., "Motion coherent tracking with multi-label mrf optimization," in *BMVC*, 2010.

[14] J. Shotton, M. Johnson, and R. Cipolla, "Semantic texton forests for image categorization and segmentation," in *CVPR*, 2008.

[15] R. Yan, J. Yang, and A. Hauptmann, "Automatically labeling video data using multi-class active learning," in *ICCV*. Press, 2003, pp. 516–523.

[16] B. Settles, "Active learning literature survey," Computer Sciences Technical Report 1648. University of Wisconsin Madison, Tech. Rep., 2010.

[17] X. Zhu and Z. Ghahramani, "Learning from labeled and unlabeled data with label propagation." CMU-CALD-02-107, CMU., Tech. Rep., 2002.

[18] N. Jojic, B. J. Frey, and A. Kannan, "Epitomic analysis of appearance and shape," in *ICCV*, 2003.

[19] V. Cheung, B. J. Frey, and N. Jojic., "Video epitomes," in *CVPR*, 2005.

[20] A. Kannan, J. Winn, and C. Rother, "Clustering appearance and shape by learning jigsaws," in *NIPS, Volume 19.*, 2006.

[21] L. Breiman, "Random forests," in *Machine Learning*, 2001, pp. 5–32.

[22] M. Grundmann, V. Kwatra, M. Han, and I. Essa, "Efficient hierarchical graph-based video segmentation," in *CVPR*, 2010.

[23] T. Brox and J. Malik, "Object segmentation by long term analysis of point trajectories," in *ECCV*, 2010.

[24] I. Budvytis, V. Badrinarayanan, and R. Cipolla, "Semi-supervised video segmentation using tree structured graphical models," in *CVPR*, 2011.

[25] G. Brostow, J. Shotton, J., and R. Cipolla, "Segmentation and recognition using structure from motion point clouds," in *ECCV, Marseille*, 2008.

[26] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," in *ICCV*, 1999.

[27] V. Badrinarayanan, F. Galasso, and R. Cipolla, "Label propagation in video sequences," in *CVPR*, 2010.

[28] G. E. Hinton, "Learning to represent visual input," *Philosphical Transactions of the Royal Society, B.*, vol. 365, pp. 177–184, 2010.

[29] C. Wang, M. Gorce, and N. Paragios, "Segmentation, ordering and multi-object tracking using graphical models," in *ICCV*, 2009.

[30] C. M. Bishop, *Pattern Recognition and Machine Learning*, M. J. . J. K. . B. Schlkopf, Ed. Springer, 2006.

[31] L. K. Saul and M. I. Jordan, "Exploiting tractable substructures in intractable networks," in *NIPS*, 1996.

[32] A. Criminisi, J. Shotton, and E. Konukoglu, "Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning," *Foundations and Trends in Computer Graphics and Vision*, vol. 7, no. 2-3, 2012.

[33] A. Ayvaci, M. Raptis, and S. Soatto, "Sparse occlusion detection with optical flow," *IJCV*, 2011.

[34] Y. Chuang, A. Agarwala, B. Curless, D. H. Salesin, and R. Szeliski, "Video matting of complex scenes," *ACM SIGGRAPH*, vol. 21, No. 3, pp. 243–248, 2002.

[35] A. Y. C. Chen and J. J. Corso, "Propagating multi-class pixel labels throughout video frames," in *Proceedings of Western New York Image Processing Workshop*, 2010.

[36] G. Brostow, J. Fauqueur, and R. Cipolla, "Semantic object classes in video: A high-definition ground truth database," *PRL*, vol. 30(2), pp. 88–97, 2009.

[37] I.Budvytis, V. Badrinarayanan, and R. Cipolla, "Label propagation in complex video sequences using semi-supervised learning," in *BMVC*, 2010.

[38] P. Chockalingam, N. Pradeep, and S. Birchfield, "Adaptive fragments-based tracking of non-rigid objects using level sets," in *ICCV*, 2009.

Vijay Badrinarayanan obtained his bachelors degree in Electronics and Communication Engineering from Bangalore University in 2001. After completing his M.S from Georgia Tech, Atlanta in 2006, he went on to obtain his Ph.D from INRIA Rennes, France in 2009. He is currently a post-doctoral research associate in the Machine Intelligence Laboratory, Department of Engineering, University of Cambridge, U.K. His research interests are in probabilistic graphical models and learning applied to segmentation, object detection and recognition, interactive vision applications.

Ignas Budvytis received his BA degree in Computer Science from the University of Cambridge in 2008. He is currently finishing his doctoral studies in the Machine Intelligence Laboratory, Department of Engineering, University of Cambridge. His research interests include semi-supervised video segmentation and object class recognition.

Roberto Cipolla obtained a B.A. (Engineering) from the University of Cambridge in 1984 and an M.S.E. (Electrical Engineering) from the University of Pennsylvania in 1985. From 1985 to 1988 he studied and worked in Japan at the Osaka University of Foreign Studies (Japanese Language) and Electrotechnical Laboratory. In 1991 he was awarded a D.Phil. (Computer Vision) from the University of Oxford and from 1991-92 was a Toshiba Fellow and engineer at the Toshiba Corporation Research and Development Centre in Kawasaki, Japan. He joined the Department of Engineering, University of Cambridge in 1992 as a Lecturer and a Fellow of Jesus College. He became a Reader in Information Engineering in 1997 and a Professor in 2000. His research interests are in computer vision and robotics and include the recovery of motion and 3D shape of visible surfaces from image sequences; object detection and recognition; novel man-machine interfaces using hand, face and body gestures; real-time visual tracking for localisation and robot guidance; applications of computer vision in mobile phones, visual inspection and image-retrieval and video search. He has authored 3 books, edited 8 volumes and co-authored more than 300 papers.