Label propagation in complex video sequences using semi-supervised learning

Ignas Budvytis
ib255@cam.ac.uk
Vijay Badrinarayanan
vb292@cam.ac.uk
Roberto Cipolla
cipolla@eng.cam.ac.uk

Department of Engineering
University of Cambridge
Cambridge, UK

Abstract

We propose a novel directed graphical model for label propagation in lengthy and complex video sequences. Given hand-labelled start and end frames of a video sequence, a variational EM based inference strategy propagates either one of several class labels or assigns an unknown class (void) label to each pixel in the video. These labels are used to train a multi-class classifier. The pixel labels estimated by this classifier are injected back into the Bayesian network for another iteration of label inference. The novel aspect of this iterative scheme, as compared to a recent approach [1], is its ability to handle occlusions. This is attributed to a hybrid of generative propagation and discriminative classification in a pseudo time-symmetric video model. The end result is a conservative labelling of the video; large parts of the static scene are labelled into known classes, and a void label is assigned to moving objects and remaining parts of the static scene. These labels can be used as ground truth data to learn the static parts of a scene from videos of it or more generally for semantic video segmentation.

We demonstrate the efficacy of the proposed approach using extensive qualitative and quantitative tests over six challenging sequences. We bring out the advantages and drawbacks of our approach, both to encourage its repeatability and motivate future research directions.

1 Introduction

Fast and efficient discriminative classifiers like Random Forests have shown promising results for video segmentation [2, 11]. However, training these classifiers require copious quantities of labelled video data, which unfortunately is extremely strenuous to obtain by hand labelling. To reduce the burden of hand labelling, label propagation methods for semi-supervised learning, like [1],[13], exploit the structure in the distribution of data points to infer unknown labels from the few labelled points.

Recently, [1] proposed the problem of label propagation in video sequences for training multi-class classifiers designed for video segmentation. Given hand-labelled start and end frames of a video sequence, the goal is to propagate labels throughout the rest of the video sequence. To this end, [1] proposed a coupled Bayes net for joint modelling of the image sequence and their pixel-wise labels. A simple variational EM strategy is employed to infer the

BMVC 2010 doi:10.5244/C.24.27

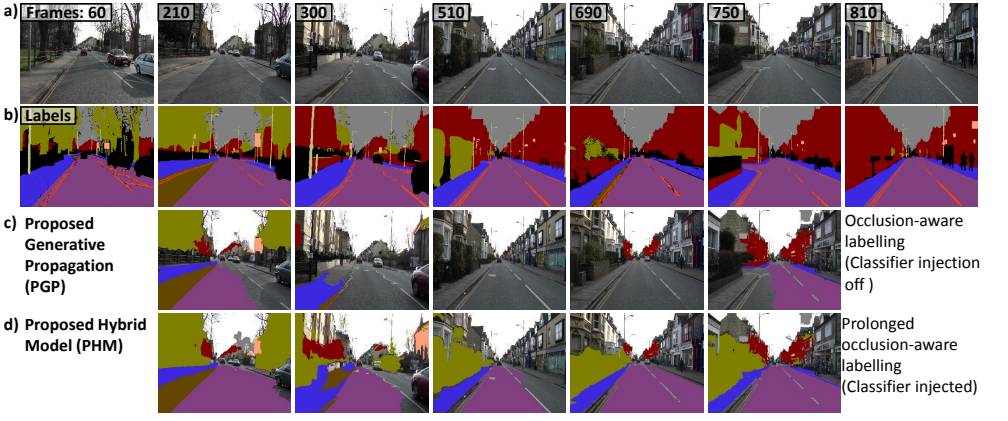

Figure 1: A motivational illustration. The reader is encouraged to zoom-in to see the details.

most probable class label for the pixels in the video. This scheme provides high quality labels for 2-3 second videos. However, performance degrades significantly for greater lengths as their model lacks a mechanism to tackle occlusions. Their algorithm is also afflicted by a *time-assymetry*, that is, the inferred labels change if the video is time-reversed. In a bid to address these drawbacks, our contribution towards label propagation in video sequences are as follows:

1. We propose a novel directed graphical model (Bayes net), which combines generative propagation and discriminative learning to tackle occlusions/disocclusions and propagate labels in long ($\approx 25s$) and complex sequences.
2. We perform time-symmetric label propagation by modelling a "doubled" sequence: the original sequence and its time-reversed version appended to it (see Fig. 2).
3. We empirically demonstrate that a classifier is more confident and accurate when trained with the propagated labels, as compared to training with only two hand-labelled video frames, to support our proposed scheme.

Fig. 1 illustrates the goals of the proposed scheme. Rows (a),(b) show samples from the publicly available CamVid driving sequence dataset [3] and their corresponding ground truth respectively. Of these, only ground truth labels of frames 60 and 810 are used to initialise label propagation. Row (c) shows the propagated labels when the classifier injection is switched "off" during inference (PGP). In contrast, when the injection is "on" distinctly more labels are obtained (PHM in row (d)).

The remainder of the paper is organized as follows. We begin with a literature review in Section 2. In Section 3 we discuss the proposed model and inference strategy. We describe the dataset, empirical parameter settings and the details of quantitative evaluation in Section 4. We devote Section 5 to a comparative analysis of the results. We conclude in Section 6.

2 Literature review

A publicly available video labelling tool is LabelMe Video [12]. Here the user draws a polygon around an object at the start, in some key frames and the end frame. This polygon is interpolated using object specific 2D or 3D velocity model on an ego-motion compensated video. In contrast, we avoid any ego-motion estimation, place no assumptions on object(s)

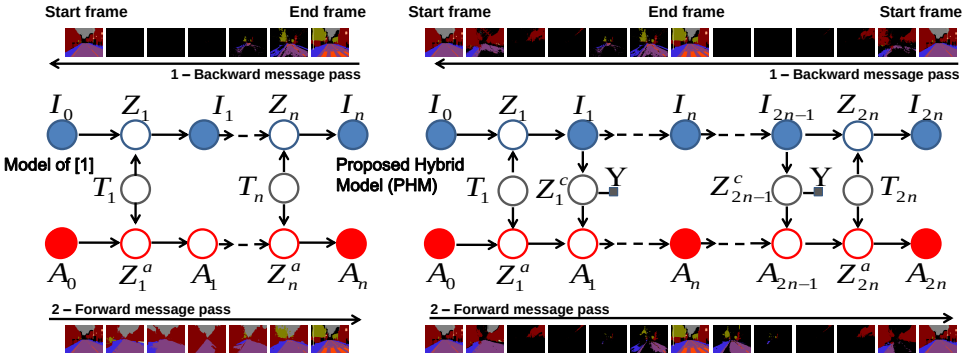

Figure 2: The model proposed in [1] and the proposed hybrid model (PHM) for label propagation. Shaded nodes represent observed/clamped variables. The Markov chains have time-reversed frames $I_{n+1:2n}$ placed to get a pseudo time-symmetric video model. The order of message passing and thumbnail images of assymetric ([1] with no void labels) and pseudo-symmetric label propagation (PHM with increasing void labels towards the middle) are shown. The gray rectangle is parameter Υ in $p(Z_k^c|I_k;\Upsilon)$.

motion and provide pixel-wise labelling. SIFT-flow [9] is a recent method to transfer labels to an image from a labelled set of similar images in a database. The labels are transferred by image matching in a SIFT descriptor space. The argument is that their approach is equivalent to optic flow for non-sequential data. However, as [1] demonstrated, *deterministic optic flow* based label propagation is inferior to probabilistic patch based methods. In addition, optic flow based methods will be further challenged over lengthy sequences with occlusions.

In image *self-similarity* based models like the Jigsaw model [6], a Pott's model based prior is imposed on mappings between the Jigsaw pixels and image pixels. Under the optimal mapping, the Jigsaw learns the repeated images structures (of arbitrary shape) in a set of images. Interestingly, a trained decision tree based classifier (based on patch mappings) is used to prune down the search space of possible mappings. In principle, labelling the image structures within the Jigsaw can transfer labels to the set of images. However, this involves learning the correct Jigsaw size which capture image structures that are intuitive to label. Alternatively, one can learn the Jigsaw, and its labelling from the two hand-labelled end frames. However, there is no mechanism to avoid erroneous labelling of objects not present in the hand-labelled frames. In comparison, our frame to frame generative model uses simpler patch matches between frames, and reserves the classifier for handling occlusions.

Badrinarayanan *et al.* [1] extend the static epitome model of [4] to a time-series model for video labelling. Their variational EM inference only captures "local" uncertainty in the labels due to messages from adjacent past and future frames. This results in an undesirable *time-assymetry*, where reversing the order of frames would result in different labelling. Understandably, this inference strategy trades-off propagation of label uncertainties for reduced complexity of inference. In this paper, we show that the same inference strategy applied to our proposed model negates the causal effects afflicting their method.

As mentioned in the introduction, there is no mechanism in the scheme of [1] for occlusion handling. In our model, we invoke a discriminative classifier to "fill-in" labels for disoccluded *static parts of the scene* (for instance, labelling road segments previously occluded by a car). Additionally, a "black-box" style treatment of classifiers in the Bayes net implies that our model can support any canonical classifier. For the sake of empirical studies, we choose a Random Forest classifier used in [11] and train it using probabilistic labels.

3 Model

Our proposed Bayesian network model is shown in Fig. 2 alongside the model proposed in [1]. The elements of the proposed model are explained below.

Nodes:
1. $I_{0:n}$ are the observed sequence of images. Images $\{I_k = I_{2n-k}\}_{k=n+1}^{2n}$ are the sequence of images in time reversed order (termed as *pseudo* observations). The order of message passing iterations are the same in both the models, but sequence "doubling" in our model leads to *occlusion aware label propagation* (see thumbnail images in Fig. 2 and Sec. 3.1).
2. Z_k is a *latent colour image* consisting of "overlapping latent colour image patches", $Z_k = \{Z_{k,j}\}_{j=1}^{\Omega}$, where j is the patch index into the set of patches Ω. As in [4], [1] we first assume these patches to be mutually independent even though they share coordinates, but then enforce agreement in the overlapping parts during inference by resorting to a Viterbi type variational approximation. This technique allows us to lay down tractable conditional distributions (Eqn. 1), and the inference (line 12 in algorithm 1) allows us to implicitly recapture correlations between latent image patches.
3. Z_k^a is a *latent labelled image* consisting of "overlapping latent labelled patches", $Z_k^a = \{Z_{k,j}^a\}_{j=1}^{\Omega}$. Each pixel i in patch j, $Z_{k,j(i)}^a$, where $j(i)$ denotes coordinate i relative to the top-left hand corner of patch j, is a multinomial random variable taking one of $L+1$ mutually exclusive values: a void (unknown class) label and L known class labels. Label 1 is reserved for void. Correlations between overlapping patches are captured as in Z_k.
4. A_k is an image sized two dimensional "grid". At each coordinate of this grid is a set of $L+1$ continuous non-negative real valued random variables which sum to unity. For instance, at coordinate v we have $\sum_{l=1}^{L+1} A_{k,v,l} = 1.0$.
5. Z_k^c is a latent labelled image obtained as a result of feeding the observed image I_k through a "black box" classifier. Each pixel $Z_{k,v}^c$ on the grid V is an independent multinomial with $L+1$ mutually exclusive states. Υ represents the internal parameters specific to the chosen classifier, for instance, the tree structure and split node functions in a random forest classifier (see 4).
6. $T_k = \{T_{k,j}\}_{j=1}^{\Omega}$ is the set of "patch mapping" variables which couple the top and bottom Markov chains. An *instance* of $T_{k,j}$ maps latent image patch $Z_{k,j}$ to an observed patch $I_{k-1,T_{k,j}}$ of the same size in I_{k-1}. The same instance of $T_{k,j}$ also maps latent labelled patch $Z_{k,j}^a$ to a patch $A_{k-1,T_{k,j}}$ of the same size on the grid A_{k-1}. $T_{k,j}(i)$ denotes pixel i in the patch mapped to by $T_{k,j}$.

Edges:
1. The latent image Z_k is predicted from observed image I_k as shown below.

$$p(Z_k|I_{k-1},T_k) = \prod_{j=1}^{\Omega}\prod_{i\in j} \mathcal{N}\left(Z_{k,j(i)}; I_{k-1,T_{k,j}(i)}, \phi_{k-1,T_{k,j}(i)}\right), \tag{1}$$

where, index j runs over all the (overlapping) latent patches $Z_k = \{Z_{k,j}\}_{j=1}^{\Omega}$. $Z_{k,j(i)}$ is pixel i inside patch j at time k. $T_{k,j}(i)$ indexes the pixel $I_{k-1,T_{k,j}(i)}$ in I_{k-1}. $\mathcal{N}(.)$ is a normalized Gaussian distribution over $Z_{k,j(i)}$, with mean $I_{k-1,T_{k,j}(i)}$ and variance $\phi_{k-1,T_{k,j}(i)}$ (held constant in our experiments, see 4).
2. The observed image I_k is "explained" by latent image Z_k as shown below.

Algorithm 1: Proposed inference for label propagation.

Input: Image sequence $I_{0:n}$ with user provided labels for I_0 and I_n
Output: Labels for static parts of the scene in $I_{1:n-1}$

Initialization// See [1] for initialization of $Z_{1:2n}$.
$\{I_k = I_{2n-k}\}_{k=n+1}^{2n}$; // time-reversed sequence
$Z^a_{1:2n-1} = 0$; // this improper initialization does not affect the iterations.
Z^c_n, Z^a_n are clamped to the end frame label; Z^c_{2n}, Z^a_{2n} are clamped to start frame label;
$Z^c_{1:2n-1} = 0$ // classifier injection is initially "off".
$\Pi_k = \{\pi_{k,v,l} = \frac{1}{L+1}\}_{l=1}^{L+1}, k = 1:2n-1 and \forall v \in V$
// A_0, A_n, A_{2n} and $T_{k,j}$ initialization
for $k = 0:2n$ **do**
  $p(Tk, j) \propto rect(j, 30, 40)$, where $rect(j, w, h)$ represents uniform values over a rectangular window of dimension $w \times h$ centered on patch j.
  if $k = 0, n, 2n$ **then**

$$A_{k,v,l} = \begin{cases} 1.0 & \text{if pixel label} = l \text{ and } l > 1, \\ 0.0 & \text{if pixel label} \neq l \text{ and } l > 1, \\ 1.0/(L+1) & \text{if pixel label} = 1 \text{ (void)}, \end{cases}$$

  else
    $A_{k,v,l} = 1.0/(L+1), \forall l = 1:L+1$ // "flat" distribution
// Variational approx. - 'q' function. $A^*_k = A_k$ for $k = n, 2n$.
$q(Z_{1:2n}, Z^a_{1:2n}, A_{1:2n}, Z^c_{1:2n}, T_{1:2n}) = \prod_{k=1}^{2n} q(T_k)\delta(Z_k - Z^*_k)\delta(Z^a_k - Z^{a*}_k)\delta(A_k - A^*_k)\delta(Z^c_k - Z^{c*}_k)$.
LabelPropagation // Note: our interest is in $Z^a_{1:n-1}, Z^c_{1:n-1}$.
for *iter* $= 1:M$ **do**
  $Z^a_{1:n-1} \leftarrow$ InferLabels$(I_{0:2n}, Z_{1:2n}, Z^a_{1:2n}, A_{0:2n}, Z^c_{1:2n}, T_{1:2n}, \Pi_{1:2n-1})$ // See alg. 2.
  $Z^c_{1:n-1}, \Pi_{1:n-1} \leftarrow$ LearnClassifier$(I_{0:n}, Z^a_{0:n})$ // $Z^a_0 = A_0$. See alg. 3 and the text in Sec. 3.1.

$$p(I_k|Z_k) = \prod_{v \in V} \mathcal{N}(I_{k,v}; \frac{1}{N_v} \sum_{\substack{j=1 \\ s.t. j \supset v}}^{\Omega} Z_{k,j(v)}, \psi_{k,v}), \tag{2}$$

where $I_{k,v}$ denotes the intensity of pixel v in the image sized grid V. j indexes patches in Z_k which overlap pixel v. $\psi_{k,v}$ is the variance of the normalized Gaussian which is held constant in our experiments (Sec. 4). Note that $j(v) = j(i')$ where i' is a coordinate in patch j which overlaps global coordinate v.

3. The latent labelled image Z^a_k is predicted from A_{k-1} as follows.

$$p(Z^a_k|A_{k-1}, T_k) = \prod_{j=1}^{\Omega} \prod_{i \in j} \prod_{l=1}^{L+1} A_{k-1, T_{k,j}(i), l}^{Z^a_{k,j(i),l}}, \tag{3}$$

where the indices on the first two products are the same as in Eqn.1. The last term is the discrete class probability distribution of the random variable $Z^a_{k,j(i)}$ corresponding to pixel i in patch j.

4. A_k is predicted from Z^a_k and Z^c_k as shown below.

$$p(A_k|Z^a_k, Z^c_k) = \prod_{v \in V} \frac{\Gamma(\alpha_{v,0})}{\Gamma(\alpha_{v,1}) \cdots \Gamma(\alpha_{v,L+1})} \prod_{l=1}^{L+1} A_{k,v,l}^{\alpha_{v,l}-1}, \tag{4}$$

which sets a Dirichlet prior on the (independent) parameters $\{a_{k,v}\}_{v \in V}$. Γ denotes the gamma function with parameters $\alpha_{v,l} = \frac{1}{N_v} \sum_{\substack{j=1 \\ s.t. j \supset v}}^{\Omega} z^a_{k,j(v),l} + z^c_{k,v,l} + \lambda$ for $l = 1 \ldots L+1$ and $\alpha_{v,0} = \sum_{l=1}^{L+1} \alpha_{v,l}$. Note that j indexes patches in Z^a_k which overlap pixel index v in the image sized grid V. N_v is the number of elements in the sum. λ is a real positive constant ($>= 1.0$) to avoid infinities.

5. The "black-box" classifier output for image I_k is defined as follows.

$$p(Z_k^c|I_k;\Upsilon) = \prod_{v\in V}\prod_{l=1}^{L+1} \pi_l\left(I_k,\Upsilon\right)^{Z^c_{k,v,l}}, \tag{5}$$

where, class probabilities obey $\sum_{l=1}^{L}\pi_l = 1.0$, and v is a coordinate on the Z_k^c grid V.

3.1 Inference and Learning

Given $\{I_{0:2n}, A_0, A_n, A_{2n}\}$, we lower bound its log probability as shown below.

$$\log p(I_{0:2n}, A_0, A_n, A_{2n}) \geq \int_\Theta q(Z_{1:2n}, Z^a_{1:2n}, Z^c_{1:2n}, A_{0:2n}, T_{1:2n}) \times \log \frac{p(Z_{1:2n}, Z^a_{1:2n}, Z^c_{1:2n-1}, A_{0:2n}, T_{1:2n}, I_{0:2n})}{q(Z_{1:2n}, Z^a_{1:2n}, Z^c_{1:2n}, A_{0:2n}, T_{1:2n})}, \tag{6}$$

where $q(.)$ is an auxiliary distribution. The form of this approximating distribution is:

$$q(Z_{1:2n}, Z^a_{1:2n}, A_{1:2n}, Z^c_{1:2n}, T_{1:2n}) = \prod_{k=1}^{2n} q(T_k)\delta(Z_k - Z_k^*)\delta(Z_k^a - Z_k^{a*})\delta(A_k - A_k^*)\delta(Z_k^c - Z_k^{c*}). \tag{7}$$

The above computationally tractable form allows us to approximate a posterior distribution over the mapping variables and a MAP estimate over the remaining ones. We estimate this distribution via the variational EM algorithm (V-EM) which aims to maximize the above lower bound. The analytic expressions for this alternating scheme are shown in Alg. 2. In the E-step we fix the latent variables and derive an approximate posterior over the mappings and in the M-step the optimal values of the latent variables are computed using the fixed-point equations (Alg. 2) which depend on this approximate posterior.

Learning under our proposed model implies estimating the the classifier internal parameter Υ. In principle, the inferred values of the latent variable $Z^c_{1:2n}$ are to be used as the "desired output" while training a multi-class classifier (random forest) for the image sequence $I_{0:2n}$. However, due to the "explaining away" effect (see [5]) the inferred estimates of $Z^c_{1:2n}$ remain highly anti-correlated to the corresponding $Z^a_{1:2n}$ estimates, if the fixed point equations are not lead to convergence in a bid to reduce computation. Instead of waiting until convergence, we can speed up the learning process by using the estimates of $Z^a_{1:2n}$ in place of the $Z^c_{1:2n}$ estimates. This can be justified by the fact that in our model, at convergence, the MAP estimates of $Z^a_{1:2n}$ are seen to be a good approximation to the MAP estimates of $Z^c_{1:2n}$. Note that this speed up hack is primarily necessitated due to the use of a factorial auxiliary distribution.

For further details of inference and learning, the reader is referred to Algs. 1, 2 and 3, which are presented as pseudo-codes.

3.2 Discussions

The analysis of the proposed model under three different themes is provided below.

Void propagation for occlusion handling The M-step in algorithm 2 involves summing over all the possible states of each mapping variable $T_{k,j}$. This expense can be reduced by approximating $q(T_{k,j})$ by a delta distribution centered on the maximum probable mapping. When the EM iterations are started from the end frame (backward message pass), the M-step update equations for $A_{k,v,l}$, under this approximation, assigns a “flat” distribution to

Algorithm 2: InferLabels( )

Input: $I_{0:2n}, Z_{1:2n}, Z^a_{1:2n}, A_{0:2n}, Z^c_{1:2n}, T_{1:2n}$
Output: $Z^a_{1:n-1}$; the labels for static parts of the scene in $I_{1:n-1}$

1 **E-step** `// Do for` $k=1:2n$. $q(T_{k,j}) \propto \prod_{i \in j} \mathscr{N}\left(Z^*_{k,j(i)}; I_{k-1,T_{k,j}(i)}, \phi_{k-1,T_{k,j}(i)}\right) \prod_{l=1}^{L+1} A_{k-1,T_{k,j}(i),l}^{Z^{a*}_{k,j(i),l}} p(T_{k,j})$

M-step `// Do for` $k=2n-1:1$ `(backward pass), then for` $k=1:2n-1$ `(forward pass), with` $k \neq n$. Ψ `is the Digamma function.`

2 `Update for` $Z^*_{k,j(i)}$ `is same as in [1].`

3 $\nabla Z^{a*}_{k,v,l} = \log A^*_{k,v,l} + \sum_{j}^{\Omega} \sum_{s.t. v \in j} \sum_{T_{k,j}} q(T_{k,j}) \log A^*_{k-1,T_{k,j}(v),l} - \Psi(Z^{a*}_{k,v,l} + Z^{c*}_{k,v,l} + \lambda)$

4 $Z^{a*}_{k,v,l} = \begin{cases} 1 & \text{if} \quad \nabla Z^{a*}_{k,v,l} > \nabla Z^{a*}_{k,v,l'}, l' = 1,...,L+1, l' \neq l, \\ 0 & \text{otherwise.} \end{cases}$

5 $A^*_{k,v,l} \propto \underbrace{\left(Z^{a*}_{k,v,l} + Z^{c*}_{k,v,l} + \lambda - 1\right) + \sum_{j=1}^{\Omega} \sum_{T_{k+1,j}} \sum_{i \ s.t. T_{k+1,j(i)}=v} q(T_{k+1,j}) Z^{a*}_{k+1,j(i),l}}_{b_{k,v,l}}$ leads to $A^*_{k,v,l} = \frac{b_{k,v,l}}{\sum_{l=1}^{L+1} b_{k,v,l}}, l \in 1:L+1$

6 $\nabla Z^{c*}_{k,v,l} = \log A^*_{k,v,l} - \Psi(Z^{a*}_{k,v,l} + Z^{c*}_{k,v,l} + \lambda) + \log \pi_l(I_k, \Upsilon)$

7 $Z^{c*}_{k,v,l} = \begin{cases} 1 & \text{if} \quad \nabla Z^{c*}_{k,v,l} > \nabla Z^{c*}_{k,v,l'}, l' = 1,...,L+1, l' \neq l, \\ 0 & \text{otherwise.} \end{cases}$

8 **return** ()

coordinates v which are not involved in any mapping between time instants k and $k+1$. Consequentially, at these coordinates, pixels $Z^{a*}_{k,v,l}$ are labelled void. The overall outcome is propagation of void (uncertain) labels; in row (d) of Fig. 4 (PGP), as we move backwards in time, *newly appearing parts of the scene are labelled void*. This reduces erroneous labelling by remaining conservative.

Comparison to the model in [1] In the *time-assymetric model* [1] the mapping variables $T_{k,j}$ are in the *causal direction*, i.e. latent patches in Z_k are mapped to patches in I_{k-1}. The effect is that, the propagated "void" labels from the backward message pass are erroneously assigned 1 of L known class labels in a subsequent forward message pass (see Fig. 2). To negate these causal effects, we double the time-series by placing *pseudo* observations to get "more symmetric mappings". By this construction, the model includes both *causal* mappings $(0:n)$ and *anti-causal* mappings $(n+1:2n)$ to enable more *pseudo time-symmetric* inference (see PHM in Fig. 2). Note that it is not equivalent to a fully symmetric model, hence we term it "pseudo-symmetric" model. However, the cost is doubled to 4 message passing iterations per inference call.

Blackbox classifier training The M-step update for model parameter Υ is shown in line 4 of Alg. 3. The RHS term can be cast as a sum of negative KL divergences as a function of Υ and an independent entropy term. Therefore, the M-step simply updates a classifier's internal parameters to reduce the total error over the training set (inferred labels). This general argument leads us to treat the classifier as a "blackbox".

4 Experimental design

We use the publicly available CamVid driving video dataset [3] for our experiments. We chose seq05VD (30Hz) in this dataset and divided it into 6 sequences (See Table 1). Our focus in this paper is on classes like roads, pavements, roadmarkings and others (10 classes

Algorithm 3: LearnClassifier( )

Input: $I_{0:n}, Z^a_{0:n}, Z^c_0 = Z^a_0$
Output: $Z^c_{1:n-1}$, $\Pi_{1:n-1}$ where $\Pi_k = \{\pi_{k,v,l}\}_{l=1}^{L+1} and \forall v \in V$

Initialization
Pixels with label 1 (void) are considered "equi-probable" over the known classes
Remaining pixels have a label between $2 : L+1$ with probability 1
Blackbox classifier training
$\hat{\Upsilon} = \text{argmax}_\Upsilon \sum_{k=0}^{n} \sum_{v \in V} \sum_{l=1}^{L} Z^{c*}_{k,v,l} \log \pi_l(I_k, \Upsilon)$ `// sum over L classes only`
$\hat{\Pi}_{1:n-1} \leftarrow$ `EstimateLabelsfromClassifier(`$I_{0:n}, \hat{\Upsilon}$`)`
`FilterLabels(`$\hat{\Pi}_{1:n-1}, \mu$`)` {
for *k = 1:n-1* **do**
  for $v \in V$ **do**
    if *max class probability* $\pi_{k,v,l} \geq \mu$ **then**
      Set $pi_{k,v} = \delta_l$
      Set $Z^c_{k,v,l'} = 1, if l' = 1, else = 0$
    else
      Set estimated distribution $pi_{k,v}$ to "flat"
      Set $Z^c_{k,v,1} = 1$
$\Pi_{1:n-1} \leftarrow \hat{\Pi}_{1:n-1}$
return $Z^c_{1:n-1}, \Pi_{1:n-1}, \Upsilon = \hat{\Upsilon}$ } `// Obtain` $Z^c_{n+1:2n-1} = Z^c_{n-1:1}, \Pi_{n+1:2n-1} = \Pi_{n-1:1}$ `(time-reversal).`

including void) relevant to learning the static parts in the driving scene. We merged grass and tree classes, and chose a single sign class for different varieties of traffic signs. The experimental details of various aspects of the proposed approach are listed below.

Labelling protocol We work with 320×240 sized images. We set a difference of 750 frames between the start and end frames. We then sample every 5^{th} frame (6Hz) to reduce the sequence length. For the selected (hand-labelled) start and end frames, we manually assign label 1 (void) to the following parts to aid in occlusion handling.
1. Difficult, and therefore mislabelled parts of the frame, e.g leaves and far away objects.
2. Parts around the vanishing point and entry points of moving objects into the scene, such as cross roads. (See Figs. 1, 4 and supplementary videos.)

Quantitative evaluation We perform quantitative evaluation to report global and class average accuracies (See [11] for definitions) with and without small static classes (SSC = {signs, poles, road markings}) over 6 sequences. Due to the labelling protocol and void labels, we compute these accuracies only over pixels labelled into known classes by our method and in the ground truth. For the same reasons, we tabulate the number of points labelled by our scheme versus PGP in Table 1.

Label inference The RGB channels are treated independently and scaled between $0.0 - 1.0$. In algorithm 2, we use a patch size of 7×7 with their centers shifted by a pixel in both axes, and set the prior $p(T_{k,j})$ to "flat" over a 30×40 pixel-grid centered on the patch j. The search area exceeding the image border is cut-off. The variances of all the Gaussians are fixed to 1.0. We perform two iterations including the first where the classifier is cut-off.

Learning the classifier We choose the 1^{st} stage Random Forest (RF) classifier, as in [11], with 16 trees, each of depth 10. Input LAB 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 rectangular patches. We use the same kind and number of features as in [11]. The key difference in training is

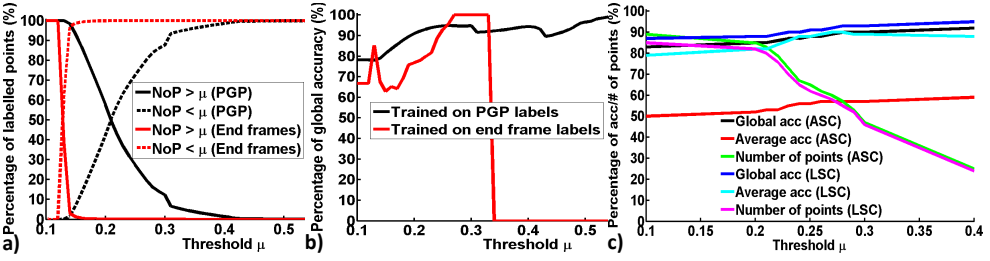

Figure 3: The effect of threshold μ for Seq. S1 on: (a). training with proposed generative propagtion (PGP) labels (graceful drop) versus training with end frame labels (steep drop) , (b). global accuracy of classifier estimates (steep drop in the end frames case) and (c) Performance of large classes (LSC) greater than all static classes (ASC). Zoom-in to see the details. See supplementary material for videos.

that we train with *soft labels delivered by PGP*. One way to handle soft labels is to *replicate* the data point with *deterministic labels* according to class probabilities. To avoid this computational overhead, 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. Consequentially, we are able to train using uncertain (void) labels which have "flat" distributions alongwith certain labels (delta) in a seamless manner. In contrast, semi-supervised learning methods such as [8] device loss functions in order to include unlabelled data alongside labelled data. Conceptually, we automatically balance out the function of the RF, between the extremes as a classifier (fully labelled data) and a clustering method (void labels only), while [8] use the RF only in the classifier mode.

In algorithm 3, we manually select the *maximum probability threshold* μ to maintain a balance between the number of labelled points and their global average accuracy. This parameter could be included a random variable in the model but at a cost of increased model complexity. Therefore, we set μ by visual inspection to gate through the estimated labels.

Training with hand-labelled frames v/s learning with proposed generative propagation (PGP) The graph (a) in Fig.3 plots the number of labelled points which are above and below the uncertainty threshold μ, and (b) the corresponding global average accuracies, for sequence S1 in Fig. 4. These plots are also shown for the case of learning with the two hand-labelled end frames and the remaining frames assigned complete void labels. In this case, as μ is increased, there is a steep drop in the number of labelled points (and accuracy) above the threshold. This indicates that the classifier is uncertain about its estimates. In contrast, these curves drop (increase) gracefully for learning with our label propagation scheme. This encourages the use of our scheme for classifier training.

Performance versus μ The graph (c) in Fig.3 plots the global and class avgerage accuracies of PHM with and without small classes, the pecentage of labelled points (excluding voids). We see that including the small classes causes a significant drop in the class average. This is due to low image resolution affecting both PGP and classifier estimates (CE).

Computational requirements The E-step in algorithm 2 was implemented in C# with a 8 core processor at a cost of 90s/frame. The M-step in matlab costed 15s/frame on the same

Settings			Class accuracies for static classes									All static classes (ASC)			Large static classes(LSC)			Small static classes (SSC)			MC
Sequence	Frames	Model	Sky	Building	Sign	Pole	Road marking	Road	Pavement	Tree	Concrete	Global class acc	Average class acc	Number of points	Global class acc	Average class acc	Number of points	True positives	Uncertain (void)	False positives	Uncertain (void)
	60	PGP	62	69	31	0	14	93	94	95	84	86	60	23	89	83	22	5	73	22	97
S0	to	CE	35	20	54	0	0	92	84	100	75	83	51	16	84	68	16	1	92	7	98
	810	PHM	75	26	27	0	7	93	91	97	78	79	55	53	82	77	51	5	52	44	87
	810	PGP	87	99	2	1	30	98	94	65	-	93	60	22	96	89	21	3	79	18	93
S1	to	CE	100	91	0	0	44	93	99	85	-	95	64	22	95	94	21	1	94	5	99
	1560	PHM	99	88	1	0	7	90	87	77	-	88	56	65	91	88	62	3	36	61	76
	1560	PGP	89	89	82	46	74	94	96	66	-	88	79	69	91	88	61	54	24	22	77
S2	to	CE	99	75	60	23	77	87	92	52	-	82	71	58	87	84	51	37	39	23	65
	2310	PHM	95	80	60	20	75	80	85	66	-	79	70	91	85	83	80	53	4	43	5
	2310	PGP	99	96	82	44	62	94	97	36	-	90	76	79	94	86	69	58	12	30	75
S3	to	CE	100	83	52	29	67	93	87	41	-	83	69	66	90	83	58	39	29	32	56
	3060	PHM	99	87	77	30	65	88	78	35	-	83	70	94	88	80	82	61	1	39	2
	3060	PGP	98	98	17	47	63	97	97	19	-	95	67	13	98	82	12	9	82	9	94
S4	to	CE	100	99	0	0	43	93	96	0	-	94	54	13	95	78	13	1	96	3	84
	3810	PHM	98	92	16	12	37	93	85	9	-	90	55	45	92	76	44	19	43	38	38
	3810	PGP	96	99	5	2	41	97	81	54	-	93	59	26	96	86	25	6	72	22	94
S5	to	CE	100	99	0	3	75	99	97	5	-	98	60	36	99	80	36	1	90	9	84
	4560	PHM	100	97	1	1	17	94	76	4	-	90	49	79	93	74	76	7	24	69	35

Table 1: Quantitative test results with $\mu = 0.25$. Note the high accuracies for PHM for Seq. S1 in Fig. 4 (blue). Compare the accuracies with and without small classes (pink). Note that true positives and uncertain labels together share the majority for SSC (orange). Note (yellow) the void labels assigned to moving classes (an unoptimised μ reduces performance in S2,S3,S4 and S5).

processor. This code was not optimised. With a C# implementation classifier training took 3 hours for 150 frames. We are working towards making these codes publicly available.

5 Comparative analysis

We list below our key comparative observations for various methods.

Generative propagation of [1] the inferred labels using the model of [1] is shown in row (c) of Fig. 4. The labels erroneously converge to a few large classes as there is no provision for occlusion handling (See frame 1260 in row(c)).

Generative propagation in our model (PGP) the inferred labels in row (d) of Fig. 4 (classifier is cut-off) shows our *conservative labelling* approach, where the frames farther away from the ends have increasing amounts of void labels (See frame 1260). We replace void labels by image pixels for clarity. However, from Table 1 it is clear that although both accuracies are high, only a small percentage of points are labelled into known classes, except in sequences S2,S3 where the camera is nearly static. It is also apparent that, except in S2,S3 (small classes have reasonable accuracy), most small classes are assigned voids. This brings down the percentage of false positives. Also, importantly, moving objects are assigned a void label to a high degree to help *tackle occlusion*. Finally, note that the untextured sky parts are assigned a void label due to unreliable patch mappings.

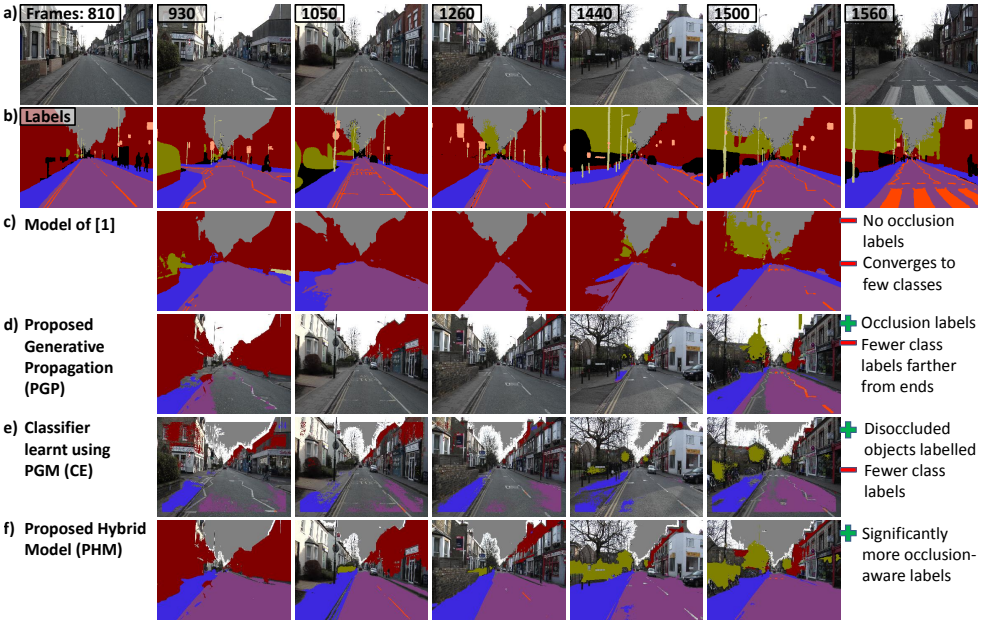

Figure 4: Qualitative comparisons on sequence S1 (see Table 1). The reader is encouraged to zoom-in to see the details. Also refer the supplementary material for this result video and others.

Classifier estimates (CE) the classifier is learnt from the inferred labels (PGM) of row (d) and the user labelled frames. The classifier estimated labels (with $\mu = 0.25$) are shown in row (e) of Fig. 4. From Table 1 (CE) it can be seen that the accuracies nearly follow those of generative propagation, confirming the fact that the classifier "overfits" the scene well. Note the high degree of void labels assigned to small classes and moving objects. Untextured sky parts, disoccluded and newly appearing parts of the static scene, *e.g*. pavements, are correctly labelled (see frames $1050, 1260$). This is key to *prolong label propagation*. However, the percentage of labelled points is similar to PGP.

Proposed hybrid model (PHM) From row (f) of Fig. 4 we can observe a marked increase in the number of labelled points. We point the reader to frame 1260 to observe this effective increase. Note how both PGP (void labels over entering cars) and CE (sky, repeating disoccluded structures) estimates are *beneficially fused*. Classes similar in appearance, such as pavements and roads are clearly distinguished. Small classes like roadmarkings are labelled reasonably well too. Table 1 shows a significant percentage increase in the labelled points over both PGP and CE methods. However, there is a decrease in the global and class averages. This is clearly due to low averages over small classes, as discounting these classes leads to a significant increase in accuracy. The reason, firstly, is that the classifier has very little data to learn small classes in low resolutions. Secondly, for fairness, the value of μ is unoptimised, except for S1. This reduces void labels over moving classes too and brings down the class averages. These drawbacks can be removed in higher resolutions and including μ into the model as a random variable (see [10]).

6 Conclusion

We presented a Bayesian network model, which is a hybrid of generative label propagation and discriminative classification, to propagate labels in complex video sequences with occlusions. We chain together the sequence and its time-reversed version to negate the effects of causality. In contrast, a corresponding undirected model involves defining joint distributions with normalization issues. As compared to more principled hybrid models [7], we sacrifice rigor and propose the treatment of a classifier as a "blackbox" within the Bayes net to include off-the-shelf classifiers. The similarity with their approach lies in the fact that the mapping variables in our hybrid model balance between the generative and discriminative extremes in modelling the visible data. We empirically demonstrated the advantage of learning with inferred labels over learning with two hand-labelled frames. This encourages the use of our method to train classifiers for large scale applications, like driving scene recognition. Extensive quantitative tests indicate the efficacy of our approach over generative propagation alone.

References

[1] V. Badrinarayanan, F. Galasso, and R. Cipolla. Label propagation in video sequences. In *CVPR, SanFrancisco*, 2010.

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

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

[4] V. Cheung, B. J. Frey, and N. Jojic. Video epitomes. In *CVPR, SanDiego*, 2005.

[5] G. E. Hinton, S Osindero, and Y. Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18:1527–1554, 2006.

[6] J. Lasserre, A. Kannan, and J. Winn. Hybrid learning of large jigsaws. In *CVPR, Minneapolis*, 2007.

[7] J. A. Lasserre, C. M. Bishop, and T. P. Minka. Principled hybrids of generative and discriminative models. In *CVPR, NewYork*, 2006.

[8] C. Leistner, A. Saffari, J. Santner, and H. Bischof. Semi-supervised random forests. In *ICCV, Kyoto*, 2009.

[9] C. Liu, J. Yuen, and A. Torralba. Nonparametric scene parsing: Label transfer via dense scene alignment. In *CVPR, Miami*, 2009.

[10] S Maskell. A bayesian approach to fusing uncertain, imprecise and conflicting information. *Information Fusion*, 9(2):259–277, 2008.

[11] J. Shotton, M. Johnson, and R. Cipolla. Semantic texton forests for image categorization and segmentation. In *CVPR, Anchorage*, 2008.

[12] J. Yuen, B. C. Russell, C. Liu, and A. Torralba. Labelme video: building a video database with human annotations. In *ICCV, Kyoto*, 2009.

[13] X. Zhu and Z. Ghahramani. Learning from labeled and unlabeled data with label propagation. Technical report, CMU-CALD-02-107, CMU., 2002.