Temporally Consistent Multi-Class Video-Object Segmentation with the Video Graph-Shifts Algorithm

Albert Y. C. Chen and Jason J. Corso
Computer Science and Engineering
SUNY at Buffalo
{aychen, jcorso}@buffalo.edu

Abstract

We present the Video Graph-Shifts (VGS) approach for efficiently incorporating temporal consistency into MRF energy minimization for multi-class video object segmentation. In contrast to previous methods, our dynamic temporal links avoid the computational overhead of using a fully connected spatiotemporal MRF, while still being able to deal with the uncertainties of the exact inter-frame pixel correspondence issues. The dynamic temporal links are initialized flexibly for balancing between speed and accuracy, and are automatically revised whenever a label change ($shift$) occurs during the energy minimization process. We show in the benchmark CamVid database and our own wintry driving dataset that VGS improves the issue of temporally inconsistent segmentation effectively—enhancements of up to 5% to 10% for those semantic classes with high intra-class variance. Furthermore, VGS processes each frame at pixel resolution in about one second, which provides a practical way of modeling complex probabilistic relationships in videos and solving it in near real-time.

1. Introduction

Segmentation of multiple *semantic objects* (such as *human, building, tree, sky, ...*) in images and videos is a problem of broad interest in computer vision. One of the many reasons is because these algorithms [4,5,15] not only detect what types of objects are in the image and localize them, but also output detailed inter-object boundaries. With these information, higher-level semantic relations such as *a car is next to a pump in a gas station* can be inferred, which could benefit problems ranging from scene understanding, video surveillance, to autonomous-driving applications. Although many advancements have been reported on existing image datasets, such as the MSRC [14] and PASCAL VOC Challenge [10], single-image-based multi-class semantic object segmentation algorithms are prone to producing inconsistent segments when applied directly to videos, as shown in Fig. 1. One main reason is because these algorithms are largely based on 2D spatial features alone—complications such as lighting, occlusion, and even sensor noises could cause the spatial features to vary from frame to frame even for static objects, thus producing inconsistent output.

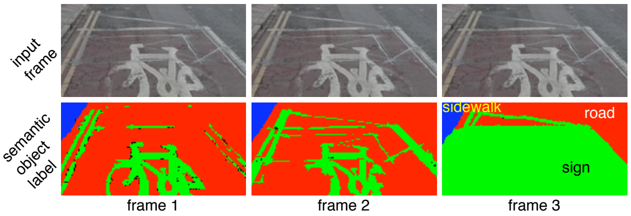

Figure 1. An example of inconsistent segmentation of near-static semantic objects in video frames.

Among the many methods for multi-class object segmentation, pixel labeling via energy minimization on Markov Random Fields (MRF) [12] (including Conditional Random Fields (CRF) [11]) have been quite popular, which is likely due to their mathematical elegance, empirical power [16] and efficient recent algorithmic developments like graph-cuts [2] and belief propagation [18]. However, previous approaches in introducing temporal consistency to MRFs are often either too restrictive (only applies for videos captured by static cameras at high frame rates and static background) [9, 13, 19] or are computationally expensive [6]. For every pixel $p^t_{\mathbf{z}}$ in frame t at coordinate $\mathbf{z} = (x, y)$: Zhou et al.'s method [19] limits $p^t_{\mathbf{z}}$'s temporal connectivity to only $p^{t-1}_{\mathbf{z}'}$ where $\mathbf{z} = \mathbf{z}'$, which only works well for videos obtained by static cameras, e.g. surveillance videos; Chen and Tang's method [6] connects $p^t_{\mathbf{z}}$ to all $p^{t-1}_{\mathbf{z}'}$ in frame $t-1$ (as shown in Fig. 2), which increases the nodal connectivity by an order of N^2 (N is the number of pixels in an image, e.g. $N = 640 \cdot 480$).

When the exact pixel motions are known, every $p^t_{\mathbf{z}}$ in frame t need only be connected to at most one $p^{t-1}_{\mathbf{z}'}$ in frame

The authors are grateful for the financial support provided in part by NSF CAREER IIS-0845282, NSF CNS-0855220, and DARPA HR0011-09-1-0022.

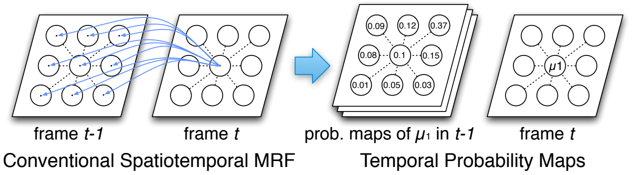

Figure 2. The temporal consistency constraint enforced by using a fully connected spatiotemporal MRF, as in [6].

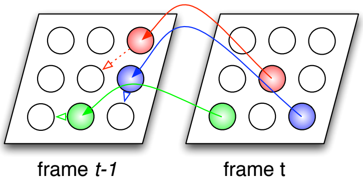

Figure 3. Ideal temporal links. Dotted arrows represent the nodal motions and solid arrows are the ideal temporal links.

$t-1$, where $\mathbf{z}=\mathbf{z}'+\mathbf{v}$, $\mathbf{v}$ is the motion vector from $\mathbf{z}'$ to $\mathbf{z}$, as shown in Fig. 3. In this case, the temporal consistency constraint can be easily formulated as a binary relation between one pixel in frame t and one in frame $t-1$, in the form of $f(p^t_{\mathbf{z}}, p^{t-1}_{\mathbf{z}'})$. One exception is when the camera zooms out, motion vectors originating from multiple $\mathbf{z}'$ could point towards the same $\mathbf{z}$; nevertheless, $p^t_{\mathbf{z}}$ would still be enforced to be consistent with one or multiple $p^{t-1}_{\mathbf{z}'}$'s. The biggest problem of this *exact pixel motion* assumption is that dense optical flow algorithms are often not as reliable as we expect it to be [1,7,17]. One remedy is to connect every $p^t_{\mathbf{z}}$ to all $p^{t-1}_{\mathbf{z}'}$ as in [6] (Fig. 2), so that the temporal relationships are multiple *soft* ones instead of one *hard* one. However, due to the local characteristics of optical flows, many of these temporal links are redundant (temporal likelihood value close to 0), which results in unnecessary wastes in computing power and memory storage.

Inspired by the merits of these previous approaches, we propose using a single dynamic temporal link to enforce temporal consistency to 2D MRFs, as illustrated in Fig. 4. Our driving motivation is the acknowledgement that even the best optical flow estimations can make mistakes. Therefore, we allow the temporal links to be modified when the other evidences (unary likelihood from classifier and spatial smoothness prior) show otherwise. The temporal links between t and $t-1$ are initialized by either coarsely estimated or precisely computed dense optical flows, then modified dynamically during the 2D MRF's energy minimization process at frame t. Dynamic modification of temporal links are enabled by any type of energy function that is dependent on the class label, e.g. the Potts energy. Note that the labels (segments) of frame $t-1$ are fixed when the temporal links between t and $t-1$ are dynamically modified, i.e., we perform energy minimization on the 2D MRF of frame t with an additional temporal term that is defined on the 2D MRF of $t-1$ instead of doing it on a strict spatiotemporal MRF. Although this simplification is not exactly a spatiotemporal MRF, our results show that it does sufficiently enforce temporal consistency without adding significant complexities.

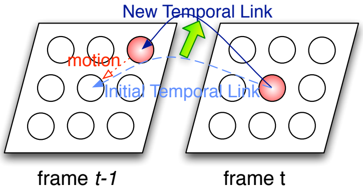

Figure 4. *Dynamic Temporal Links*: in our proposed algorithm, temporal links self-modify while the 2D MRF in frame *t* is energy-minimized.

We develop and define the energies used in our algorithm in Sec. 2, followed by the full Video Graph-Shifts (VGS) algorithm in Sec. 3. We efficiently establish the initial temporal links for our VGS algorithm by exploiting the multi-level adaptive hierarchy structure as in [8]. Due to the lack of ground-truth video datasets of the segmentation of multiple semantic classes in the community, we recorded and hand-labeled an hour-long wintry driving dataset and tested our VGS algorithm on it, as well as on the CamVid video benchmark [3]. We discuss the database and experiment setup along with the results in Sec. 4. The whole process of temporal link construction and energy minimization takes about 1 second per 320 x 240 frame to converge, compared to the minute-long fully-connected temporal links approach used in [6]. We consistently achieve an accuracy rate increase of 5% to 10% for the semantic classes where the classifier alone suffers from noise and large intra-class variance therefore produce temporally inconsistent segments. Such result demonstrates the effectiveness of our VGS algorithm.

2. Temporally Consistent Energy Model

We first discuss the energy models used in the standard 2D MRF, inspect existing and possible ways of incorporating temporal consistency constraints, then define and analyze our *temporal links* and *temporal energy*. Ultimately, the energy function is:

$$E[\{m_\mu : \mu \in D\}] = \lambda_1 \sum_{\mu \in D} E_1(\mathbf{I}(S[\mu]), m_\mu) + \lambda_2 \sum_{\langle \mu,\nu \rangle} E_2(m_\mu, m_\nu) + \lambda_3 \sum_{\mu \in D} E_t(m_\mu, m_\rho) \quad (1)$$

where D is the image lattice, $S[\mu]$ is the local sub-image surrounding μ, $m_\mu = \{\mathcal{L}_1, \mathcal{L}_2, ..., \mathcal{L}_k\}$ is the label $\mathcal{L}_l$ taken by μ, for example *tree, car, building* and so on. $\langle \mu, \nu \rangle$ denotes that ν is a neighbor of μ, ρ is the node in the previous frame that is corresponded to μ through μ's *temporal link*, and $\sum_i \lambda_i = 1$ are the weights of the energy terms. Discussion of the individual energy terms follows.

2.1. The Standard Energy Terms

The energy used in standard 2D MRFs is the first two terms of (1). E_1 (unary energy) is the potential of each node μ belonging to a certain class m_μ, and E_2 (binary energy) is the energy induced by the interaction between neighboring nodes. Low-energy or high-likelihood configurations of the labels are preferred.

The unary potentials can be calculated from suitable manually-defined or machine-learned classifiers, and can incorporate various color, texture and even shape properties. In VGS, we set E_1 as:

$$E_1(\mathbf{I}(S[\mu]), m_\mu) = -\log P\big(m_\mu|\mathbf{I}(S[\mu])\big) \ . \tag{2}$$

where the probability $P(m_\mu|\mathbf{I}(S[\mu]))$ is conditioned on a local sub-image of μ. We use a context-sensitive discriminative model—the Probability Boosting Tree (PBT) algorithm—to generate $P(m_\mu|\mathbf{I}(S[\mu]))$.

The binary term can be designed to incorporate our prior knowledge of the inter-nodal relationship, e.g., the spatial smoothness constraint. E_2 can be viewed as the penalty applied to neighboring nodes possessing different class labels:

$$E_2(m_\mu, m_\nu) = 1 - \delta(m_\mu, m_\nu) \ . \tag{3}$$

2.2. Temporal Link and the Temporal Energy

We use only one dynamic temporal link per node. A label dependent energy function defined on this temporal link is used to motivate the alteration of this dynamic temporal link. We develop the temporal energy used in our algorithm in the following paragraphs.

Let us first explain the concept of a node μ's *temporal correspondent* ρ in frame $t-1$ with a toy example: a video is simply the movements of k disparate nodes (i.e., nodes with distinct feature values), where each instance (frame) of the k nodes forms a graph D_i. The temporal correspondent of $\mu \in D_t$ is simply the search of $\mu' \in D_{t-1}$, where $\mu = \mu'$. In real-world problems, however, nodes in the MRF are not guaranteed to be disparate, i.e., there may exist two nodes with exactly the same attributes (color, shape, texture) in the same frame. Also, large homogeneous regions complicates the calculation of exact nodal movement, even with the information of object motion. Thus, the definition of μ's *temporal correspondent* ρ is relaxed to the searching of $\kappa \in D_{t-1}$ that minimizes the feature space distance to μ:

$$\rho = \operatorname*{argmin}_{\kappa} ||\boldsymbol{X}_\mu - \boldsymbol{X}_\kappa||_p, \quad \kappa \in D_{t-1} \ . \tag{4}$$

$\boldsymbol{X}_\omega$ is the feature vector of any node ω, and $||\cdot||_p$ is the L^p norm used for distance measurements.

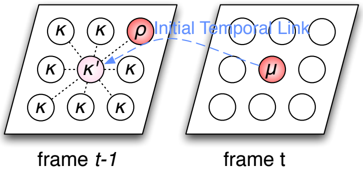

Figure 5. Finding μ's *temporal correspondent* ρ through the initial estimate κ', in μ's 2^{nd} order neighborhood.

The above definition of *temporal correspondents* still fails to account for an important prior knowledge we have about videos: object motion $\boldsymbol{o}_m$ is usually limited in-between neighboring frames—the higher the frame rate, the smaller the movement. Therefore, only nodes within a certain range need to be examined while searching for ones temporal correspondent (those with suddenly fast movement can be handled by coarser level nodes of our adaptive-hierarchy, which will be discussed in detail in 3.2.) In terms of the MRF neighborhood, a node $\mu \in D_t$ will only have to search within the n^{th} order spatial neighborhood of the initial estimate of the temporal correspondent node $\kappa' \in D_{t-1}$, where $n \propto |\boldsymbol{o}_m|$:

$$\rho = \operatorname*{argmin}_{\kappa} ||\boldsymbol{X}_\mu - \boldsymbol{X}_\kappa||_p, \quad \kappa \in \{\cup\eta : \langle\kappa', \eta\rangle\} \ . \tag{5}$$

The temporal relationship between μ and its temporal correspondent ρ can thus be formulated as a binary energy:

$$E_t(m_\mu, m_\rho) = 1 - \Psi(\mu, \rho), \tag{6}$$

where the similarity measure Ψ is defined as

$$\Psi(\mu, \rho) = \begin{cases} 0 & \text{if } m_\mu \neq m_\rho \\ \exp(-\alpha||\boldsymbol{X}_\mu - \boldsymbol{X}_\rho||_p) & \text{otherwise} \end{cases} . \tag{7}$$

α is a non-negative coefficient. When a node μ insists on taking a label $\mathcal{L}_l$ that its temporal correspondent's neighboring region shows no supports for, E_t is assigned the highest possible energy (i.e., 1) to discourage this selection.

3. The Video Graph-Shifts (VGS) Algorithm

The VGS algorithm is composed of three major steps: (1) *Coarsening*: for each frame t, an adaptive hierarchy G_t is built on top of the standard MRF by recursively grouping the nodes (pixels) with similar attributes (color, intensity, depth); (2) *Linking*: temporal links are constructed between corresponding nodes in G_t and G_{t-1}; (3) *Shifting* (as in Fig. 6): nodes *shift* iteratively by taking different labels to minimize the energy function. The search is a gradient descent minimization in as search space defined by the adaptive hierarchy (i.e., not a local search space). After a shift occurs, we refine the temporal links to enforce consistency with the label change.

3.1. Coarsening the Adaptive Dynamic Hierarchy

The adaptive-dynamic hierarchy differs from traditional pyramidal structures of the image in the following ways:

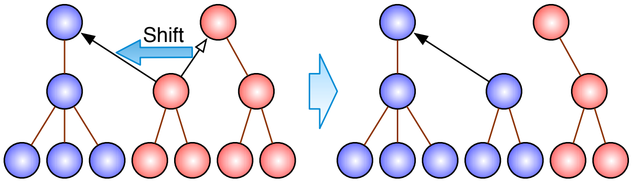

Figure 6. Process of a *shift* on a toy adaptive hierarchy.

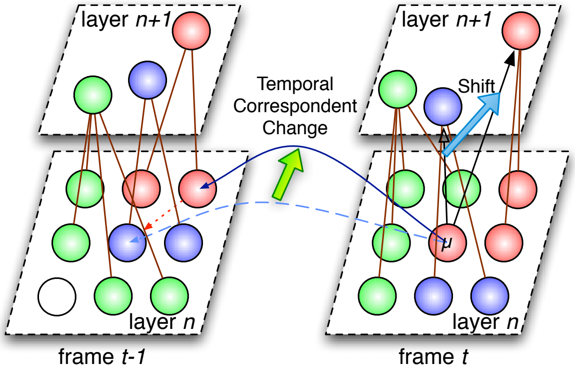

Figure 7. *Video Graph-Shifts*: The red dotted arrow denotes the movement of μ from frame $t-1$ to t. When μ *shifts* from its original parent (black hollow arrow) to its new parent (black solid arrow), *temporal shifts* will follow and find its most similar node in frame $t-1$ (light blue dotted arrow to dark blue solid arrow).

(1) The hierarchy is built according to the data instead of pixel coordinates; (2) The hierarchy is dynamic, so that the structure of nodes (parent-child relationship) changes dynamically while the energy is being minimized.

Nodes with similar attributes are grouped together to form a node in the next layer. The top layer of the hierarchy consists of K nodes, where each node represents one of the K labels. The *parent-label constraint* forces a node to have the same label as its parent; hence, an instance of the adaptive hierarchy is a full pixel labeling. The energy is accumulated recursively from bottom to top throughout the hierarchy G: Leaf nodes calculate their energy directly from (1) - (3), while non-leaf nodes sum the energy from all their child nodes. (Refer to [8] for details on the recursion.)

3.2. Initializing the Temporal Links

Because the temporal links are dynamically altered during energy minimization, only a coarse estimate of object motion is needed to initialize them. This coarse estimation can involve little or no calculation of the motion field at all, depending on how fast the objects are moving, the moving directions, and the frame rate.

For nodes with Brownian motion, since the mean motion vector is 0, we initialize the temporal link of $\mu \in D_t$ to $\rho \in D_{t-1}$, where their spatial coordinates are the same. (Of course, these will change later during the energy minimization process.) For nodes with non-Brownian motion, the initial temporal correspondent can be estimated efficiently with the adaptive hierarchy. Since most objects are rigid with a coherently moving inner region, when μ's hierarchical parent ρ_h's spatial displacement $\boldsymbol{o}_m$ is known, it is reasonable to estimate that μ's temporal correspondent is in the same direction as $\boldsymbol{o}_m$. For this reason, all but the nodes in the coarsest level of the hierarchy requires the explicit calculation of nodal-motion. The computational complexity for finding the nodal-motion is thus reduced from N^2 to $[N/(\gamma^\iota)]^2$, where N is the number of pixel-level nodes, γ is the reduction factor of the hierarchy coarsening process, and ι is the number of layers in the graph-shifts hierarchy. Using the same setting as in [8] on a 320 x 240 video frame, the number of nodes that requires this explicit attention is 77000 versus less than one hundred.

3.3. Shifts and Temporal Correspondent Changes

A *shift* is when a node μ decides to change its parent (and here, class label m_μ) to its neighboring node's parent (and takes a new label $\hat{m}_\mu$), all μ's descendants follow; it is denoted $m_\mu \rightarrow \hat{m}_\mu$. The resulting change in global energy, called the *shift-gradient* $\Delta E(m_\mu \rightarrow \hat{m}_\mu)$, is defined as:

$$\begin{aligned} \Delta E(m_\mu \rightarrow \hat{m}_\mu) &= \lambda_1 \big[E_1(\mu, \hat{m}_\mu) - E_1(\mu, m_\mu)\big] \\ &+ \lambda_2 \big[\sum_{\eta:\langle\mu,\eta\rangle} [E_2(\hat{m}_\mu, m_\eta) - E_2(m_\mu, m_\eta)] \big] \\ &+ \lambda_3 [E_t(\hat{m}_\mu, m_{\hat{\rho}}) - E_t(m_\mu, m_\rho)] \ , \end{aligned} \tag{8}$$

where μ can be a node at any level in the hierarchy.

Potential *Temporal Correspondent Changes* are evaluated after a *shift*, which seeks a new *temporal correspondent* $\hat{\rho}$ that minimizes the *temporal energy gradient* $E_t(\hat{m}_\mu, m_{\hat{\rho}}) - E_t(m_\mu, m_\rho)$.

$$\hat{\rho} = \underset{\kappa}{\operatorname{argmin}}\ E_t(\hat{m}_\mu, m_\kappa), \quad \kappa \in \{\rho, \cup\eta : \langle\rho, \eta\rangle\} \ . \tag{9}$$

For each node in the hierarchy, *shift-gradients* among all neighbors are calculated and only those that are negative (which would cause an energy decrease) are stored in the list of potential shifts S. The shift with the steepest shift-gradient is chosen at each iteration, causing the node and all its descendants' labels to be changed. The nodes and edges affected by the shift will be updated, along with the list S with recomputed energies. The process is iterated until convergence when S becomes empty, which means that any further shift will not decrease the energy. Convergence is guaranteed as this is a steepest-gradient algorithm.

3.4. Video Pixel-Labeling via Video Graph-Shifts

The first frame is labeled with all E_t set to 0, i.e., the VGS algorithm essentially only performs the following two

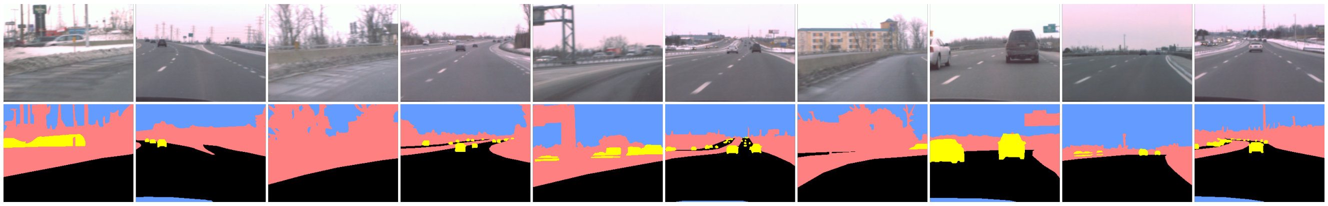

Figure 8. Sample frames (first row) and labels (second row) from the “highway” sequences of our wintry-driving dataset.

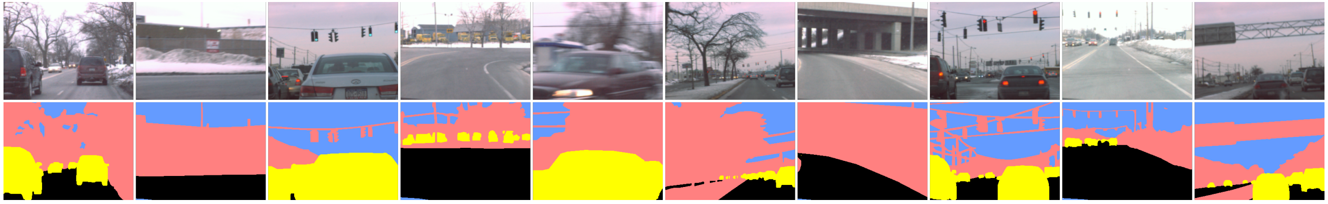

Figure 9. Sample frames (first row) and labels (second row) from the “major road” sequences of our wintry-driving dataset.

steps: (1) the adaptive hierarchy is coarsened on top of the standard MRF, and (2) all potential *shifts* are calculated, stored, then iterated until convergence. Starting from the second frame: (1) the hierarchy is coarsened, (2) the initial temporal links are constructed, and (3) each *shift* is followed by a *temporal correspondent change*, which refines the temporal link between G_t and G_{t-1} (Fig. 7). When the energy is minimized, temporal links between G_t and G_{t-1}'s lowest layer can be viewed as a motion field.

In practice, for typical videos with little frame-to-frame motion, the independently coarsened adaptive hierarchies of the two frames should be similar. With the dynamically changing nature of the hierarchy, one can superimpose the hierarchy of frame $t-1$ onto frame t to serve as its initial hierarchy. As long as the energies are updated and propagated correctly throughout the hierarchy, *shifts* will alter this superimposed hierarchy to its energy-minimized state. The final labeling result of this superimposed hierarchy is similar to those obtained from an independently coarsened hierarchy when the overall object movement and appearance/disappearance rate is small. However, when the overall object movement is significant between two frames, a new hierarchy should be coarsened instead of using the superimposed one from the previous frame. One practical solution is to interlace frames using superimposed hierarchies with frames using coarsened hierarchies; this is similar to the interlaced I, P, B frames of the MPEG compression.

4. Experiments and Results

As pointed out by Brostow et al. in their recently published video-pixel labeling benchmark dataset—CamVid [3], there is a lack of pixel-wise multiple-semantic-class labeled video recorded by non-stationary cameras. Theirs is the first such dataset. At the same time CamVid was being developed, we also realized this void and constructed our own video database with per-pixel ground truth of each semantic class. Sample frames and their corresponding pixel-wise labels of our dataset are shown in Fig. 8, 9. In this section, we apply our method to both datasets. Our dataset and code will be available for download upon publication to allow for future comparison. We discuss our new video database in detail in 4.1, followed by experiments on it in 4.2 and experiments on the CamVid database in 4.3.

4.1. Our Wintry-Driving Video Database

Our database, as of date, consists of 49 driving video sequences recorded in various weather, lighting, and scene conditions throughout the winter that spans roughly 90 minutes of time at 15 fps. Each sequence is approximately 2 minutes. The videos are recorded uncompressed at a resolution of 640x480 using a fixed focal length IEEE1394 industrial-grade camera. The sequences can be classified into three types of weather conditions: heavily snowing, not snowing but the road is slushy, and not snowing and the road is dry; three types of lighting conditions: day time, dusk, and night (vehicles' headlights and road lights on); three types of surrounding scenes: highway, major roads, and residential neighborhood driving. Currently, 375 frames have been manually labeled with four high-level semantic classes: vehicles, road-side obstacles, road, and others (mostly sky). It is now being expanded to more semantic classes as well as adding higher semantic relationship between objects, e.g. pole in front of trees, cars on the road.

Our wintry driving database differs significantly from the CamVid database in the following aspects: (1) Our dataset is recorded in a wide variety of weather conditions, and (2) the surrounding scene differs more in our sequences, e.g. highway scenes v.s. residential neighborhood scenes. Therefore, our full dataset provides a different and challenging benchmark for not only pixel labeling and multi-class video object segmentation problems, but also for bounding-box based classifiers, per-frame/image classifiers, etc.

VGS

	🟦	🟥	⬛
sky	**97.4**	2.6	
obstacles	1.9	**97.1**	1.0
road		19.6	**80.4**

2D MRF

	🟦	🟥	⬛
sky	**97.2**	2.8	
obstacles	1.9	**97.4**	0.7
road		29.7	**70.3**

Table 1. Confusion matrices of VGS (left) versus 2D MRF (right) on the major road testing set. Empty cells have values < 0.1. The per-class averaged accuracy rate is 91.63% versus 88.3% while the global accuracy rate is 94.76% versus 93.32%. Notice the 10.1% improvement of the *road* class is because of the temporally inconsistent classifier output improved by VGS.

4.2. Experiments on our Wintry-Driving Database

For the ease of comparison, we follow two conventions commonly used in the pixel-labeling society while setting up the experiments for our wintry driving dataset: the frames are down-sampled to the standard size (320x240), and the semantic classes with less than several percents of pixel-wise appearance frequency along those with high intra-variance (e.g. partially occlusions vehicles) are discarded. We randomly take 70% of the non-snowing sequences and train a single Probability Boosting Tree (PBT) classifier, which is used to generate the $P\big(m_\mu|\mathbf{I}(S[\mu])\big)$ for E_1 in both the 2D MRF and VGS for fair comparison. Object motions are assumed Brownian for temporal correspondent initialization, and a new hierarchy is built for each frame. It takes about 1 second on average for our method to process a new frame, which consists of: building the adaptive hierarchy, temporal links, and energy minimization.

The quantitative labeling results for our VGS algorithm versus 2D MRFs are shown in table 1. Our method shows a near 3.5% increase of the per-class averaged accuracy [1] on the "major road" test sequences. The major contributor to this improvement is the 10.1% improvement of the *road* class. Such result is driven by the fact that: the similarity between the roadside dirty snow pile and slush-covered roads confuses the classifier from time to time in the major road test sequences (Fig. 10), therefore allowing the temporal consistent constraint to make a significant impact on the segmentation accuracy. The roadside obstacles and the sky are rarely mislabeled, therefore the temporal consistency has little impact on the results of these two classes.

4.3. Experiments on the CamVid Video Database

The CamVid video database consists of five 960x720 video sequences that are recorded at 30Hz and spans a total of 10 minutes of city driving scenes. The videos are categorized into two types: those recorded during the day time and those at dusk. Most of their frames are labeled at 1Hz with a significant change of scene content from frame to frame. We use the same setting as in [4] to test our proposed method on the CamVid dataset (11-class subset), and the sequences shown in Fig. 13 are reserved for testing.

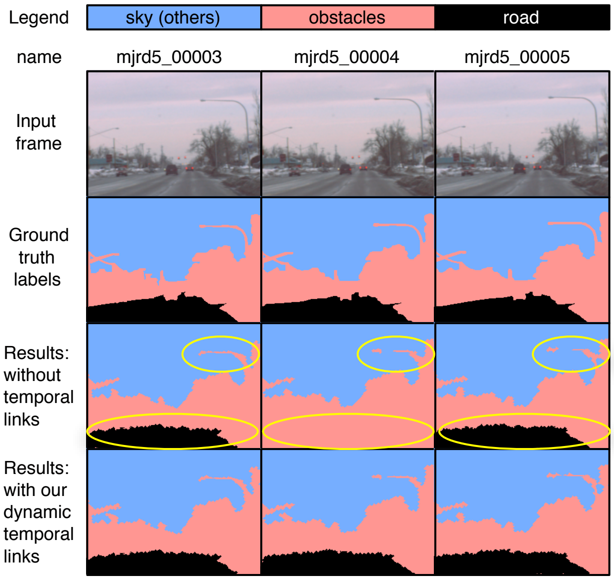

Figure 10. A frequently occurring example of VGS outperforming 2D MRF's energy minimization results on the 3-class subset of our wintry driving dataset. Single frame based energy minimization mistakenly labeled the whole *road* region as *obstacles*, while our method avoids being trapped in this type of local minima.

VGS observes a consistent improvement of labeling accuracy over 2D MRF methods again by producing more temporally consistent results. The results are better explained by on a per-class improvement basis due to the intrinsic difference in semantic classes in the test sequences. The dusk test sequences consists of many *stop and go* traffic scenes, with lots of *cars*, *pedestrians* and *bicyclists* randomly appearing and swiftly moving on the sidewalks (Fig. 11). Due to the large intra-class difference in these semantic objects, the classifier outputs are inconsistent from frame to frame. With the VGS algorithm, temporal consistency increased 6.2%, 7.3%, and 4.4% respectively in these three *flickering* classes while remaining largely the same for the more static objects, such as *building*, *tree*, amd *sky*. The day test sequences, as a contrast, are recorded in a much faster-moving vehicle with fewer appearances of *pedestrian* and *bicyclist* classes (Fig. 12). As a result, the frame-to-frame difference of the same *tree* and *fences* objects is higher, thus the temporal consistency constraint improved the per-class accuracy rates by 5.1% and 4.1% respectively.

The qualitative comparisons of the results are shown in Fig. 13 and the quantitative ones are shown in table 2 and 3. The per-class average accuracy shows a near 2% im-

[1] Recent literature [3, 4] have argued that the *per-class averaged accuracy rate* is a better measure than the conventional *global accuracy rate* (total number of correctly-labeled pixels over the total number of pixels), since the easiest way to boost global accuracy is to neglect everything that infrequently appears and give preference to the frequently appearing ones.

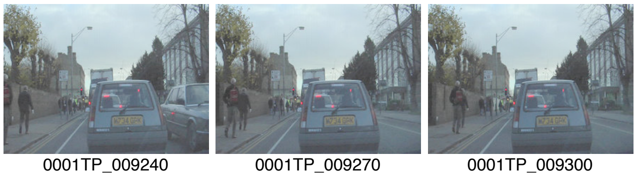

Figure 11. Sample clips from the CamVid dusk test sequences that consists of many *stop and go* traffic scenes. The images shown are brightened and contrast enhanced for the ease of visualizing.

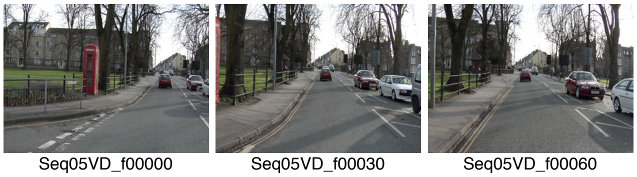

Figure 12. Sample clips from the CamVid day test sequences that are recorded in a comparatively faster moving vehicle. The tree, fence, sidewalk, and buildings classes have a larger intra-class variance in these sequences.

provement on the dusk test sequence, and The global accuracy rate of our method is 81.40% versus the 80.52%. We observe a 71.74% global accuracy rate on the day test sequences, which is also a near 1% higher than 2D MRF results. For comparison, the global accuracy reported in Brostow et al. [4] is 69.1%. Our per-class or balanced average accuracy is 46.45% whereas it is 53%—however, we note that, in both cases, their scores are computed on combined training and testing data, so comparing the two scores is unfair.

5. Conclusion

Our proposed Video Graph-Shifts algorithm provides an efficient way of modeling and energy minimizing MRFs with additional dynamic temporal links that promotes temporal consistency. The dynamic temporal links efficiently achieves a good level of temporal consistency as compared to a fully-connected spatiotemporal MRF while using only a fraction of computational time. Furthermore, the multiple available choices for initializing the temporal links provides the flexibility for different levels of accuracy and efficiency requirements. Experiments on our new wintry driving video dataset and the CamVid benchmark show that our VGS algorithm not only produces visually more consistent segmentation results, but also quantitative improvements over plain 2D MRFs. A consistent 5% to 10% improvement is observed on the classes where the classier alone suffers from noise and large intra-class variance.

References

[1] X. Bai, J. Wang, D. Simons, and G. Sapiro. Video SnapCut: robust video object cutout using localized classifiers. In *ACM SIGGRAPH*, 2009.

[2] Y. Boykov, O. Veksler, and R. Zabih. Fast Approximate Energy Minimization via Graph Cuts. *IEEE Trans. PAMI*, pages 1222–1239, 2001.

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

[4] G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla. Segmentation and recognition using structure from motion point clouds. In *Proc. of ECCV*, pages 44–57, 2008.

[5] A. Y. C. Chen, J. J. Corso, and L. Wang. HOPS: Efficient region labeling using higher order proxy neighborhoods. In *Proc. of ICPR*, 2008.

[6] J. Chen and C. Tang. Spatio-Temporal Markov Random Field for Video Denoising. In *Proc. of IEEE CVPR*, 2007.

[7] Y. Chuang, A. Agarwala, B. Curless, D. Salesin, and R. Szeliski. Video matting of complex scenes. In *ACM SIGGRAPH*, 2002.

[8] J. J. Corso, A. Yuille, and Z. Tu. Graph-Shifts: Natural Image Labeling by Dynamic Hierarchical Computing. In *Proc. of IEEE CVPR*, 2008.

[9] A. Criminisi, G. Cross, A. Blake, and V. Kolmogorov. Bilayer Segmentation of Live Video. In *Proc. IEEE CVPR*, 2006.

[10] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88(2):303–338, 06 2010.

[11] J. Lafferty, A. McCallum, and F. Pereira. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proc. of ICML*, 2001.

[12] S. Li. *Markov random field modeling in image analysis*. Springer, 2001.

[13] F. Luthon, A. Caplier, and M. Liévin. Spatiotemporal MRF approach to video segmentation: Application to motion detection and lip segmentation. *Signal Processing*, 76(1):61–80, 1999.

[14] J. Shotton, M. Johnson, R. Cipolla, T. Center, and J. Kawasaki. Semantic Texton Forests for Image Categorization and Segmentation. In *Proc. of IEEE CVPR*, 2008.

[15] J. Shotton, J. Winn, C. Rother, and A. Criminisi. TextonBoost for Image Understanding: Multi-Class Object Recognition and Segmentation by Jointly Modeling Texture, Layout, and Context. *IJCV*, pages 1–22, 2007.

[16] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother. A Comparative Study of Energy Minimization Methods for Markov Random Fields with Smoothness-Based Priors. *IEEE Trans. PAMI*, pages 1068–1080, 2008.

[17] J. Wang and M. Cohen. An iterative optimization approach for unified image segmentation and matting. In *Proc. of IEEE ICCV*, pages 936–943, 2005.

[18] J. Yedidia, W. Freeman, and Y. Weiss. Generalized Belief Propagation. In *Proc. of NIPS*, volume 13, pages 689–695, 2000.

[19] Y. Zhou, W. Xu, H. Tao, and Y. Gong. Background Segmentation Using Spatial-Temporal Multi-Resolution MRF. In *IEEE WMVC*, volume 2, pages 8–13, 2005.

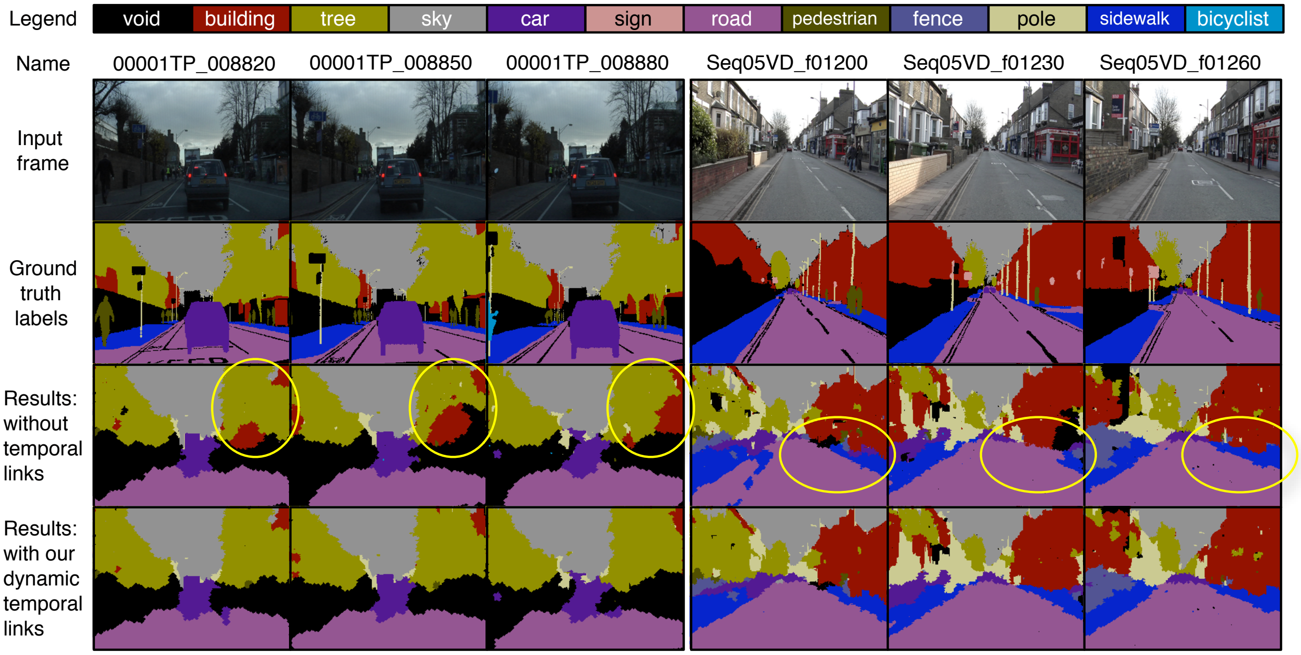

Figure 13. Examples of VGS outperforming 2D MRF energy minimization results on the Camvid 11-class dataset. Frames 00001YP_008820–80 demonstrate how dynamic temporal links produces consistent *tree* labels at the upper right corner (circled in yellow) of each frame while 2D MRFs alone fail to do so. The sidewalk region on the right side of frames Seq05VD_f01200–60 (circled in yellow) shows the same case, where our dynamic temporal links are able to correctly infer the *sidewalks* while others do not.

VGS

	■	■	■	■	■	■	■	■	■	■	■
building	**72.2**	20.4	0.5	2.0			0.5	0.3	4.0		
tree	16.4	**79.9**	1.4	1.0			0.3		0.9		
sky	1.1	5.7	**92.0**	0.4					0.8		
car	0.5	0.1		**68.8**		29.7	0.2		0.1	0.6	
sign	80.3	7.1		12.6							
road				2.4		**93.0**				4.6	
pedestrian	9.9	16.7	0.1	18.9		1.1	**25.0**	0.4	12.9	12.8	2.2
fence	43.4	2.2		23.0		15.6	3.8		5.2	6.2	0.6
column	12.4	37.5	18.4	7.3		1.2	0.6		**20.4**	1.9	0.3
sidewalk				2.6		59.7				**37.5**	0.1
bicyclist	7.6	6.7		23.5		20.3	14.2		11.8	6.2	**9.7**

2D MRF

	■	■	■	■	■	■	■	■	■	■	■
building	**71.6**	20.7	0.5	1.9			0.9	0.3	4.2		
tree	17.0	**78.3**	1.5	0.9			0.7	0.1	1.5		
sky	1.2	5.6	**91.9**	0.4					0.9		
car	0.3	0.1		**62.6**		35.9	0.4		0.1	0.6	0.1
sign	76.0	8.2		15.8							
road				1.6		**93.7**				4.7	
pedestrian	12.5	23.3	0.1	19.4		2.3	**17.7**	0.8	6.6	16.3	0.8
fence	47.0	2.1		22.6		17.8	4.3		0.1	6.1	
column	13.0	36.4	19.0	6.7	0.1	1.2	0.9		**20.8**	1.9	
sidewalk				1.6		62.0				**36.3**	
bicyclist	9.4	4.6		24.8		20.5	27.7		1.8	6.0	**5.3**

Table 2. Confusion matrices of VGS (left) versus the 2D MRF results (right) on the testing set of the camvid dusk sequences. Empty cells have values < 0.1. Global accuracy rate is 81.40% for VGS versus 80.52% for 2D MRFs while the per-class average accuracy rate is 45.31% versus 43.47%. Note that the high intra-class variance of the *car*, *pedestrian*, and *bicyclist* classes causes the classifier to output inconsistent results, and the temporal consistency constraint in VGS help improve the accuracy rate by 6.2%, 7.3%, and 4.4% respectively while remaining largely the same for the remaining more static classes.

VGS

	■	■	■	■	■	■	■	■	■	■	■
building	**58.2**	17.2	3.2	2.2			0.3	1.3	16.8	0.7	
tree	24.4	**60.8**	4.8	0.7	0.1		0.2	0.3	8.6		
sky	1.4	2.9	**93.1**	0.1					2.5		
car	0.5	1.0		**94.4**		0.7	0.1	0.5	1.3	1.4	
sign	34.9	40.9	0.5	0.2			0.1		23.5		
road			0.2	1.6		**94.9**			0.1	3.2	
pedestrian	31.5	9.7	0.1	11.1		0.2	**7.5**	2.9	33.2	3.3	0.4
fence	22.1	13.5		10.1			0.5	**25.3**	23.7	4.8	
column	27.3	18.7	5.6	4.5	0.1	0.9	0.6	2.4	**36.7**	3.1	0.1
sidewalk	0.4		0.2	3.8		54.4	0.1	0.4	0.6	**40.1**	
bicyclist	46.8	9.2	0.1	7.6		2.3		14.8	16.0	3.1	

2D MRF

	■	■	■	■	■	■	■	■	■	■	■
building	**56.2**	17.0	3.8	2.1	0.1	0.1	0.5	1.9	17.6	0.7	
tree	28.3	**55.7**	5.4	0.6	0.2		0.3	0.3	9.2		
sky	1.2	2.7	**93.5**	0.1					2.4		
car	0.4	1.1		**93.6**	0.1	0.6	0.1	0.7	1.6	1.9	
sign	32.7	47.9	0.6		**0.1**		0.1	0.1	18.4		
road			0.2	1.5		**95.4**			0.1	2.7	
pedestrian	22.6	6.6	0.2	11.3		0.3	**17.4**	5.2	33.7	2.4	0.3
fence	19.7	16.7		11.7		0.1	3.6	**21.2**	22.4	4.6	
column	26.0	17.6	6.1	4.0	0.1	1.3	1.5	2.7	**38.5**	2.2	0.1
sidewalk	0.3		0.2	3.2		57.2	0.1	0.5	0.7	**37.7**	
bicyclist	25.5	7.3	0.1	7.7		2.3	4.4	36.6	15.1	1.1	

Table 3. Confusion matrices of VGS (left) versus the 2D MRF results (right) on the testing set of the camvid day sequences. Empty cells have values < 0.1. The Global accuracy rate is 71.74% for VGS versus 70.8% for 2D MRFs while the per-class average accuracy rate is 46.45% versus 46.3%. Note that because the videos are recorded in a comparatively faster moving vehicle with fewer occurrences of the *pedestrian*, *bicyclist* classes and larger variances for the *tree* and *fence* classes. As a result, *tree* and *fence* benefited the most from the additional temporal consistency constraint, demonstrating 5.1% and 4.1% improvement respectively.