Coarse-to-fine Semantic Video Segmentation using Supervoxel Trees

Aastha Jain *
Linkedin
asjain@linkedin.com

Shaunak Chatterjee *
University of California, Berkeley
shaunakc@cs.berkeley.edu

René Vidal
Johns Hopkins University
rvidal@cis.jhu.edu

Abstract

We propose an exact, general and efficient coarse-to-fine energy minimization strategy for semantic video segmentation. Our strategy is based on a hierarchical abstraction of the supervoxel graph that allows us to minimize an energy defined at the finest level of the hierarchy by minimizing a series of simpler energies defined over coarser graphs. The strategy is exact, i.e.*, it produces the same solution as minimizing over the finest graph. It is general,* i.e.*, it can be used to minimize any energy function (*e.g.*, unary, pairwise, and higher-order terms) with any existing energy minimization algorithm (*e.g.*, graph cuts and belief propagation). It also gives significant speedups in inference for several datasets with varying degrees of spatio-temporal continuity. We also discuss the strengths and weaknesses of our strategy relative to existing hierarchical approaches, and the kinds of image and video data that provide the best speedups.*

1. Introduction

Segmenting moving objects in a video sequence is a key step in video interpretation. Most of the prior work on motion segmentation (see, *e.g.*, [12, 28, 11, 30]) uses local motion and appearance cues to segment the video in a bottom-up, unsupervised manner. However, the use of category-specific information about the object being segmented can be really helpful in the segmentation task.

This has motivated the development of semantic motion segmentation algorithms, which use supervision to label the pixels in a video according to the object class they belong to. Most existing approaches to semantic video segmentation are graph based [18, 17]. Usually, an over-segmentation of the video is obtained using standard methods [15] and a random field (RF) is defined on a graph whose nodes are the resulting supervoxels. The segmentation of the video is then obtained by minimizing a cost defined on this RF.

However, existing image and video segmentation algorithms allow every adjoining pixel (superpixel) or voxel (supervoxel) to have a different label. As a consequence, the optimization procedure is typically very slow because of the exponentially large number of possible labelings in a video. For instance, for a video with 100 frames, where each frame has 100×100 superpixels and 10 possible labels, the number of possible segmentations is $10^{1000000}$.

Many efficient inference approaches have been proposed in the past, including iterated conditional modes, mean field approximations, graph cuts, and belief propagation. In general, these approaches trade-off accuracy for efficiency by finding an approximate solution. While successful for many tasks in image segmentation, these approximate inference methods continue to be very slow for video applications.

Paper contributions. In this paper, we propose an exact, general and efficient coarse-to-fine energy minimization strategy for image and video segmentation, which can be used to speedup any approximate energy minimization approach (*e.g.*, graph cuts and belief propagation). The proposed strategy exploits the fact that real images and videos are both spatially and temporally coherent. Therefore, contiguous supervoxels (both in space and in time) are very likely to have the same label. As a consequence, *the space of coherent labelings is significantly smaller than the space of all possible labelings*. For instance, if we consider supervoxels of size $k \times k$ superpixels spanning k frames, then the number of possible segmentations for the example above reduces to $10^{\frac{1000000}{k^3}}$, which is significant even for $k = 2$.

To capture the spatial and temporal continuity of a video, we define a hierarchical abstraction of the supervoxel graph such that most supervoxels at a coarse level correspond to a single label. This will allow us to solve a much smaller optimization problem over a coarser graph and to refine this solution only when needed. We use a hierarchical graph-based supervoxel segmentation method (see [31] for an overview) to identify the supervoxels (at various scales) that are likely to have the same label. Such methods create a supervoxel tree with the biggest (coarsest) supervoxels at the highest level. The top row of Figure 1 shows the hierarchy for one of the frames from a video of the SUNY Buffalo-Xiph.org dataset [9]. The second row shows the set of superpixels used by our coarse-to-fine inference scheme. At each abstraction level, the blacked out portions denote superpixels

[*] Equal contribution

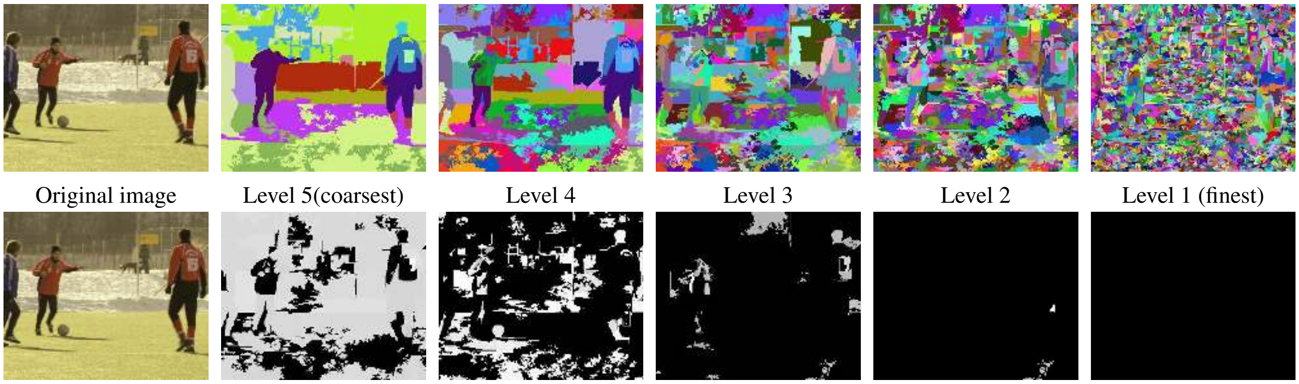

Figure 1. Supervoxel hierarchy for an image. The top row shows the various abstraction levels in the supervoxel tree. The second row shows the portion of the supervoxel tree explored by our coarse-to-fine scheme to find the optimal labeling of segments.

whose refinements were not required to find the optimal labeling. It is clear that large portions of the search space can be pruned by assigning several labels at the coarser levels.

Given this hierarchy, we construct a series of energy functions for different levels of abstraction and propose a coarse-to-fine inference scheme that minimizes these energies to find an optimal segmentation at the finest level of the hierarchy. To define the different energy functions, we first augment the set of labels with an auxiliary label called *mixed*, which accounts for the fact that coarse supervoxels may contain finer supervoxels with more than one *pure* label. We then define the unary, pairwise and higher-order costs of the energy at any level of the hierarchy as lower bounds for the costs at the finest level. By virtue of this choice, we can guarantee that the optimal segmentation upon termination is identical to the segmentation we would have obtained had we solved the original, non-hierarchical problem, which is exponentially larger in size. Our coarse-to-fine inference scheme starts by performing inference at the coarsest level of the supervoxel hierarchy using any inference method (*e.g.*, graph cuts or belief propagation). If the solution at the current level of refinement is such that no supervoxel is assigned the *mixed* label, then an optimal solution at the finest level has been found by performing inference over a very coarse graph. Otherwise, the mixed supervoxels are refined into its constituent (finer) supervoxels, and a new inference problem is solved over both coarse and fine supervoxels. This process is repeated until an optimal labeling does not assign the *mixed* label to any supervoxel.

In general, it is very hard to know if the proposed scheme is more efficient that direct inference over the finest layer. Clearly if the hierarchy of supervoxels is poorly constructed so that many refinement cycles are needed, our method could be less efficient because it solves too many small inference problems. In practice, we observe that the speedup of our approach increases with the spatio-temporal continuity of the data. Our experiments show a speedup of between 2x–10x on videos from the SUNY Buffalo-Xiph.org [9] and CamVid [7] datasets using the proposed coarse-to-fine inference scheme as opposed to the corresponding flat algorithm (graph-cuts or belief propagation).

Related work. There are several existing approaches to hierarchical image and video segmentation. One line of work in hierarchical video segmentation is a bottom-up approach based on merging supervoxels using similarity metrics based on variation of intensity inside a supervoxel [13, 18]. However, these approaches do not aim to minimize a specific energy function, which makes it difficult to compare with our method. Nonetheless, the supervoxel tree obtained by these approaches can be used as the abstraction hierarchy in our framework.

Another line of work defines a hierarchical cost function over supervoxels *at all levels*. This includes the Pylon model [25] and associative hierarchical CRFs [22]. This approach differs from our work in two key aspects. First, [22] uses *mixed* labels to enforce label continuity via a higher-order cost. In sharp contrast, we use *mixed* labels to distinguish between the very large set of unlikely segmentations and a much smaller set of more likely segmentations, and to prune the former set. Second, the tree inference methods used in [22, 25] are very different from the one we propose. Specifically, the works of [22, 25] solve a multilayer optimization problem, while we optimize a cost function defined at the finest layer only. To do this more efficiently, we use the supervoxel tree to iteratively refine the parts of the video that could have more than one label. In addition, we use lower bounds on the energy to ensure the exactness of our solution, similar to what is done in the coarse-to-fine dynamic programming [26] and temporally abstract Viterbi [8] algorithms.

There is a third line of work on hierarchical inference algorithms which do not guarantee convergence to the same solution as the corresponding flat version. [21] introduces an inference algorithm that aims to produce better solutions than α-expansion, but is much slower than the actual α-expansion. [14] proposes a version of hierarchical belief propagation for images. However, unlike our method, the abstraction used is image-agnostic and the messages at a coarse level are only used to initialize messages at the finer level and not to prevent expanding all nodes. In addition there are no theoretical guarantees that the inference method will converge to the same solution as flat belief propagation. Similarly [10] introduces the inference algorithm which uses an image based hierarchy to perform inference. However, there are no guarantees on convergence of the algorithm to the actual minimum.

Also, we can use any of these algorithms [21, 14, 10, 4] to solve the energy minimization problem in each iteration (at the current level of refinement of the graph) and hence they can complement our hierarchical inference algorithm. In that case, our algorithm converges to the solution that would be obtained by running the corresponding algorithm on the flat graph. We would like to emphasize that *further advances in supervoxel tree creation and energy minimization (both being integral components of our approach) will further increase the speedup of our hierarchical algorithm*.

2. Problem formulation

Most existing graph-based segmentation algorithms define a random field (RF) whose nodes correspond to pixels (superpixels) or voxels (supervoxels) in the image or video. For the sake of concreteness, we will describe our formulation using a RF whose nodes are the supervoxels in a video. However, the formulation is valid in the other cases as well.

Let $\mathcal{V}$ be the set of supervoxels in a video V. Each node of the RF is associated with a state $x_i \in \mathcal{L} = \{1, \dots, L\}$, which represents the category label at supervoxel $v_i \in \mathcal{V}$. Let $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ denote the set of edges of the RF. The edges are defined by using the neighborhood structure of the supervoxels, *i.e.*, $e_{ij} \in \mathcal{E}$ if supervoxels i and j share a common boundary. Let $\mathcal{C} \subset 2^{\mathcal{V}}$ be set of cliques involving three or more supervoxels. These cliques are defined to capture higher-order interactions among regions in the video, such as label consistency [20] or top-down information [29, 19]. The labeling of all the nodes in clique c is denoted by a vector $x_c \in \mathcal{L}^{|c|}$, while the labeling of all the supervoxels in a video is denoted by $x \in \mathcal{L}^{|\mathcal{V}|}$.

Given the structure of the RF, $(\mathcal{V}, \mathcal{E}, \mathcal{C})$, we define an energy function (or segmentation cost), $E(x, V)$, as

$$\lambda_U \sum_{v_i \in \mathcal{V}} \psi_i^U(x_i, V) + \lambda_P \sum_{e_{ij} \in \mathcal{E}} \psi_{i,j}^P(x_i, x_j, V) + \lambda_H \sum_{c \in \mathcal{C}} \psi_c^H(x_c, V). \tag{1}$$

The unary potential, $\psi_i^U(x_i, V)$, captures the cost of assigning the label $x_i \in \mathcal{L}$ to the supervoxel v_i in video V. Unary potentials are usually obtained by training a classifier for every class on appropriate supervoxel descriptors extracted from the videos in the training data. The pairwise potential $\psi_{ij}^P(x_i, x_j, V)$ for an edge $e_{ij} \in \mathcal{E}$ in video V captures the cost of interaction for v_i and v_j for label assignments x_i and x_j. The pairwise potentials are usually designed to enforce spatial smoothness and temporal continuity of the labels. The higher-order potential $\psi_c^H(x_c, V)$ for video V captures the cost of assigning a label x_c to all the supervoxels inside clique c, and can be used to measure the consistency of the labels of all supervoxels inside c. Finally, λ_U, λ_P and λ_H are weights representing the relative importance of the unary, pairwise and higher-order potentials. These weights are learnt using structural SVMs [16]. We will provide more details of the specific form of these potentials in Section 4.

Given an energy function, the segmentation x^* of a video V is obtained by minimizing $E(x, V)$. In general, finding the global minimum is an intractable problem. Therefore, energy minimization is usually done using approximate inference methods such as graph cuts, belief propagation, and their extensions to higher-order potentials. While these methods have been generally successful in image segmentation, they continue to be fairly slow for video segmentation due to the huge number of possible labelings a video can take. For instance, in the case of a video with around 100 frames, each one having a resolution of 960 × 720 pixels, the number of supervoxels could easily be on the order $100,000$. Thus, the number of labelings could be $|\mathcal{L}|^{100,000}$.

3. Coarse-to-fine strategy

In this section we propose a coarse-to-fine approach for solving the energy minimization problem more efficiently. Our approach exploits the fact that labels are coherent both in space and in time, hence we expect many large, contiguous patches of supervoxels to have the same category label.

3.1. Supervoxel hierarchy

The first step in our approach is the construction of a hierarchical supervoxel tree [13, 18]. The coarsest level of the tree (*i.e.*, level m) contains the biggest supervoxels and the finest level (*i.e.*, level 1) contains the smallest supervoxels. A supervoxel at site i and level j is denoted by ${v_i}^j$ and its label by ${x_i}^j$. The set of all supervoxels at level j is denoted by $\mathcal{V}^j$ and its labeling by x^j. The refinement of a supervoxel ${v_i}^j$ $(j \geq 2)$ is the set of supervoxels at the next finer level $(j-1)$ that occupy the same set of voxels in the video as ${v_i}^j$. We denote the refinement of v_i^j as $\mathcal{R}(i, j, j-1) \subset \mathcal{V}^{j-1}$. For $k < j$, we also let $\mathcal{R}(i, j, k) \subset \mathcal{V}^k$ denote the set of supervoxels obtained by refining v_i^j for $j - k$ times. The reverse function $Parent : \mathcal{V}^j \rightarrow \mathcal{V}^{j+1}$ maps a supervoxel to its parent supervoxel at the next coarser level. In this paper, we will only consider hierarchical supervoxel *trees*, and hence each supervoxel has a unique parent.

The supervoxel hierarchy can be obtained by running any of the existing hierarchical video segmentation algorithms, such as those in [18, 31]. These algorithms provide an option of either creating very large and few supervoxels or very fine and numerous supervoxels. By varying this parameter, we can get the desired supervoxel hierarchy.

3.2. Coarse-to-fine inference scheme

Given a hierarchical supervoxel tree, we propose a coarse-to-fine algorithm for efficient inference. The algorithm is designed to distinguish between two scenarios. The likely scenario is when all the (contiguous) supervoxels at level $j-1$ that constitute a supervoxel at level j get the same label from the set $\mathcal{L}$. The unlikely scenario is when a supervoxel at level j has constituents with different labels.

To represent the latter scenario, we introduce a new label $L+1$, to denote the case where a supervoxel ${v_i}^j$ ($j \geq 2$) has constituents with more than one label. We refer to label $L+1$ as the *mixed* label, and to the original L labels as *pure*. Of course, only supervoxels that can be further refined can have the *mixed* label, *i.e.*, ${x_i}^1 \neq L+1$. The augmented label set is denoted by $\mathcal{L}^A = \mathcal{L} \cup \{L+1\}$. A similar label augmentation scheme was used in [8].

The proposed coarse-to-fine inference scheme proceeds as follows. Let $E_{\mathcal{V}^m}(x, V)$ be an energy function defined at level m. This energy will be constructed from $E(x, V)$, as described in Section 3.4. We start by finding a labeling for the coarsest supervoxels in $\mathcal{V}^m$ from the augmented label set $\mathcal{L}^A$. This labeling is found by minimizing $E_{\mathcal{V}^m}(x, V)$ using some inference algorithm $\mathcal{A}$, which can be graph cuts, belief propagation or some linear program, depending on the form of the energy function being optimized. All current supervoxels (in $\mathcal{V}^m$) that receive a label $L+1$ are replaced in the current optimization problem (at level m) by their constituent supervoxels from the next finer level ($m-1$). This refinement is always possible, since a supervoxel can only receive the *mixed* label if it can be further refined.

After a refinement is done, a new RF $R_{\mathcal{V}^{curr}}$ is defined by the set of current nodes $\mathcal{V}^{curr}$ and the set of edges $\mathcal{E}^{curr}$ and cliques $\mathcal{C}^{curr}$ connecting these nodes. Notice that $\mathcal{V}^{curr}$ need not coincide with the set of nodes at any level j, $\mathcal{V}^j$, because the nodes in $\mathcal{V}^{curr}$ could correspond to supervoxels at different levels of refinement. For example, we can have a pair of neighboring supervoxels ${v_{i_1}}^{j_1}$ and ${v_{i_2}}^{j_2}$ with $j_1 \neq j_2$. Let $E_{\mathcal{V}^{curr}}(x, V)$ be an energy function defined on the current RF, which will be constructed from the energy $E(x, V)$ as described in Section 3.4. As before, we can obtain a labeling for the supervoxels in $\mathcal{V}^{curr}$ by minimizing $E_{\mathcal{V}^{curr}}(x, V)$ using algorithm $\mathcal{A}$. We can then refine a supervoxel v_i^j that receives the *mixed* label $L+1$ by replacing it by its constituent supervoxels in $\mathcal{R}(i, j, j-1)$. We repeat this process iteratively, until all supervoxels receive *pure* labels. Since every supervoxel eventually refines to its finest constituents, which in turn can only take *pure* labels, this process is guaranteed to terminate. Also, at any point in the algorithm, there exists exactly one ancestor of every finest level supervoxel ${v_i}^1$ in the current set of supervoxels.

The pseudocode of the proposed coarse-to-fine inference algorithm is provided in Algorithm 1.

Algorithm 1 Coarse-to-fine Inference Algorithm ($\mathcal{V}^{1:m}, \psi$)

```
1: $\mathcal{V}^{curr} \leftarrow \mathcal{V}^m$
2: **repeat**
3:     Find $x_{\mathcal{V}^{curr}}$ which minimizes $E_{\mathcal{V}^{curr}}$
4:     **for all** ${v_i}^j \in \mathcal{V}^{curr}$ such that ${x_i}^j = L+1$ **do**
5:         Refine ${v_i}^j$
6:         $\mathcal{V}^{curr} \leftarrow \mathcal{V}^{curr} \cup \mathcal{R}(i, j, j-1) \setminus {v_i}^j$
7:     **end for**
8: **until** $L+1 \notin x_{\mathcal{V}^{curr}}$
9: **return** $x_{\mathcal{V}^{curr}}$
```

3.3. Exactness of the coarse-to-fine solution

To make our coarse-to-fine inference scheme converge to the same labeling as that obtained by running $\mathcal{A}$ on the finest level of the supervoxel hierarchy (*e.g.*, a flat graph cuts algorithm), the potentials of the energy at a coarse level, $E_{\mathcal{V}^{curr}}$, need to be chosen in a specific manner. We will use the notion of admissible heuristics in the A* algorithm [27] to define the potentials of $E_{\mathcal{V}^{curr}}$. Since our goal is to minimize the energy function E, the admissible heuristics for the unary, pairwise and higher-order potentials of $E_{\mathcal{V}^{curr}}$ need to be chosen as lower bounds for the values of the corresponding potentials of E. Specifically, let x denote any label assignment for the finest level supervoxels and let x^* denote the optimal labeling. Let x_{cf} and x^*_{cf} denote the same for the coarse-to-fine setting at any stage of the algorithm. If we use admissible heuristic costs, then

$$E(x^*, V) \geq E(x^*_{cf}, V) \text{ and } E(x, V) \geq E(x_{cf}, V).$$

These inequalities ensure that when we terminate upon finding a *pure* labeling for the current set of supervoxels (at various levels), all other possible assignments have a higher or equal cost (since their lower bound cost is worse than the current optimal cost of the *pure* labeling).

It is important to note, however, that Algorithm 1 is exact only with respect to the underlying optimization algorithm $\mathcal{A}$ used in Step 3. One of the most popular choices for $\mathcal{A}$ is graph cuts [6] via α-expansion and α-β swap moves. Another algorithm frequently used is belief propagation [13]. Other alternatives include various linear and non-linear optimization algorithms and the choice is often guided by the particular form of the energy function we are trying to minimize. Since these optimization algorithms are not guaranteed to find the global minimum and our method will find the exact same solution that these optimization algorithms would find when run on the original finest level of the hierarchy, our method enjoys the same (approximate) optimality properties as those of the chosen optimization method $\mathcal{A}$.

3.4. Admissible coarse potentials

As we discussed in Section 3.3, in order for Algorithm 1 to converge to the same solution as that obtained by running $\mathcal{A}$ on the finest level, the potentials associated with the nodes at the coarse levels should be lower-bounds on the cost associated with the patches of fine nodes constituting these coarse nodes. In this section, we show how those lower bounds can be computed. For the sake of simplicity, we discuss the construction of the lower bounds for an energy function consisting of unary and pairwise terms only. However, our methodology is applicable to higher-order potentials as well, as long as lower bounds can be computed.

Coarse Unary Potentials. We define the unary cost $\psi^U_{(i,j)}(x_i{}^j)$ of assigning a *pure* label $l \in \mathcal{L}$ to a coarse supervoxel $v_i{}^j$ at level j as the sum of the unary costs of assigning label l to all the nodes at level 1 that constitute $v_i{}^j$, *i.e.*,

$$\psi^U_{(i,j)}(x_i{}^j) = \sum_{k \in \mathcal{R}(i,j,1)} \psi^U_{(k,1)}(x_i{}^j), x_i{}^j \in \mathcal{L}. \tag{2}$$

Notice that this is an exact cost, *i.e.*, it is a tight lower bound.

We define the unary cost of assigning a *mixed* label to a coarse supervoxel as the minimum cost associated with the RF defined by the constituent supervoxels at level 1 *subject to the constraint that all the constituent supervoxels cannot get the same label*. This minimum can be obtained by using α-expansion on $R_{\mathcal{R}(i,j,1)}$ if we do not have the constraint that the the nodes in this subgraph can not take the same label. This results in a weaker lower bound. To find the minimum cost with this constraint, we can formulate it as an integer programming problem [5] with an extra constraint which prevents all the nodes from taking the same label. While integer programing could be costly in general, notice that here we are solving an integer program on a small portion of the video given by the set $\mathcal{R}(i,j,1)$.

Pairwise Potentials. We define the pairwise potential of coarse supervoxels $v^{j_1}_{i_1}$ and $v^{j_2}_{i_2}$, $\psi^P_{(i_1,j_1)(i_2,j_2)}(x^{j_1}_{i_1}, x^{j_2}_{i_2})$, as

$$\begin{cases} 0 & \text{if } x_{i_1}{}^{j_1} = x_{i_2}{}^{j_2} \\ \sum_{\hat{\mathcal{E}}} \psi^P_{(i,1)(j,1)}(x_{i_1}{}^{j_1}, x_{i_2}{}^{j_2}), & \text{otherwise,} \end{cases} \tag{3}$$

where the set $\hat{\mathcal{E}} \subseteq \mathcal{E}$ is defined as $\hat{\mathcal{E}} = \{e_{(i,1)(j,1)} \in \mathcal{E} : i \in C(i_1, j_1, 1), j \in C(i_2, j_2, 1)\}$. Therefore, the cost of the edge between two coarse supervoxels $v^{j_1}_{i_1}$ and $v^{j_2}_{i_2}$ is the sum of the costs of the edges connecting the constituent supervoxels of $v^{j_1}_{i_1}$ and $v^{j_2}_{i_2}$ at level 1. In the case where one of the supervoxels gets the *mixed* label, the potential associated to the edge is set to zero. Although this is a loose lower bound to the actual cost of these edges (while minimizing the cost on $R_{\mathcal{V}^1}$) this saves us a lot of computation time.

3.5. Practical considerations

As discussed earlier, in general it is very hard to know if the proposed scheme is more efficient that direct inference over the finest layer. In this section, we discuss some practical considerations to ensure the efficiency of our approach.

Trade-off between accuracy and computation time. Consider two scenarios: all the lower bounds on potentials in scenario 1 are tighter than the corresponding bounds in scenario 2, for the same supervoxel hierarchy. In that case, the number of refinements required in scenario 1 will be strictly non-greater than the number of iterations required in scenario 2. Hence, *it is always beneficial to consider tighter lower bounds*. This is true for any algorithm using admissible heuristics. However, the downside of using tighter bounds is that they generally are more expensive to compute. Thus, a trade-off exists between accuracy of heuristic costs and time required to compute them.

On-demand supervoxel refinement. In most cases, only a small number of nodes in the supervoxel tree are used in the entire inference procedure. Thus, we can save computation time by not computing the entire supervoxel tree upfront, and only refining the supervoxels with the *mixed* label when needed. This on-demand refinement scheme, however, can be more expensive if we end up expanding most of the nodes in the supervoxel tree. We see moderate benefits using this scheme as reported in Section 4.

Extension to label hierarchy. In this work, we have only considered a flat label hierarchy. However, it is possible to consider a hierarchy among labels as well. For instance, since the sky and the sea labels are (often) similar, we might get additional computational benefits by considering them together (and therefore eliminating them via a single consideration for non-sky, non-sea nodes). Such a hierarchical scheme would have a *mixed* label at every label level. For more details on how to manage a label hierarchy simultaneously with a supervoxel hierarchy, we refer the reader to [8], where such a scenario is considered.

4. Experiments

This section provides an experimental evaluation of the proposed coarse-to-fine approach to video segmentation. Since the goal of this paper is to reach the same segmentation quality of existing algorithms in a more efficient manner, *the experiments are not designed to demonstrate improvements in accuracy of the segmentation with respect to state-of-the-art algorithms.* Moreover, since our goal is to find an optimal labeling for the finest layer of the hierarchy only, *the experiments are not designed to find the optimal labeling at every layer of the hierarchy*. We simply use the more abstract layers and the associated lower bound costs to find the optimal labeling at the finest layer. While the proposed coarse-to-fine scheme can be exponentially faster than flat optimization in the best case, it can also be much slower when all the supervoxels need to be refined down to their finest level. Hence, it is important to validate the usefulness of our coarse-to-fine approach to see whether it actually provides a speedup and how large this speedup is.

It is also expected that the speedup will be much larger for videos with greater spatio-temporal continuity. *The experiments are hence designed to answer these latter questions.*

4.1. Dataset

Experiments are done on two datasets: the SUNY Buffalo-Xiph.org 24-class Dataset [9] and the CamVid dataset [7].

SUNY Buffalo-Xiph.org 24-class Dataset. This dataset is a collection of general-purpose videos collected at Xiph.org. The frame-by-frame labels in the video have a fair bit of temporal consistency, which our algorithm is well-equipped to exploit. For each video, we use half of the frames for training and the remaining half for testing. We retain all 24 labels for this dataset, although most videos only have $5 - 10$ labels.

CamVid Dataset. This dataset has 700 hand segmented frames of street scenes with varying backgrounds captured from a video camera in a moving car. The video sequences have been annotated into 32 classes. However, in this paper we have combined some of the classes and are working with a total of 5 classes: people, vehicles, sky, road and background. Around 250 frames were used for training and the rest of them were used for testing. This represents a more challenging dataset in terms of an expected speedup.

4.2. Energy potentials

The unary potentials. The unary potential $\psi_i^U(x, V)$ is defined as the cost of assigning a class label x to supervoxel v_i. This cost is obtained as the score of an SVM classifier applied to the descriptor d_i of supervoxel v_i. This classifier is trained on the supervoxel descriptors for each class.

Supervoxel descriptors. The supervoxel descriptor needs to be chosen such that it captures the discriminative characteristics of the supervoxels (both appearance and motion attributes) across various classes. Also the descriptors need to satisfy some invariance properties like rotation invariance. In our experiments, we found the Spatio-Temporal Interest Points (STIP) [24, 23] descriptors to be a good fit. More specifically, each volume is subdivided into a grid of cells. For each cell, histograms of oriented gradients (HOG) and histograms of optical flow (HOF) are computed and concatenated. This is in the spirit of the well known SIFT descriptor. For our experiments, we used 5 levels (including the coarsest and finest levels). The number of supervoxels at the two extreme levels and time taken in minutes (using the on-demand generation scheme) is detailed in Table 2.

Pairwise potentials. We define the pairwise potential between the sites i and j as:

$$\psi_{ij}^P(x_i, x_j) = \frac{l_{ij}}{1 + |I_i - I_j|}\delta(x_i \neq x_j), \qquad (4)$$

where l_{ij} is the area of the common boundary between the supervoxels v_i and v_j and I_i denotes the average intensity of supervoxel v_i.

4.3. Experimental setup

In our experiments, we use α-expansion (graph cuts [6]) and belief propagation [13] as the optimization method in Step 3 of Algorithm 1. We implemented the algorithms in Python and used various functionalities from the graph libraries **Python-graph** [3] and **igraph** [1]. We used the LIBSVX [2] library's implementation of a graph-based hierarchical method [18] to generate the supervoxel trees.

4.4. Results

Computational speedup. The computation time taken by the flat algorithm (both α-expansion and belief propagation) and its coarse-to-fine counterpart are reported in Table 1. For the CamVid videos, the speedup is between 3x–5x, while for the SUNY videos the speedup is between 7x–10x. The increased speedup for the SUNY videos is expected due to the increased spatio-temporal consistency in those videos. Notice that these speedups do not account for the supervoxel tree creation time, which is only required by our algorithm. If we include the time of the on-demand refinement scheme discussed in Section 3.5, the overall speedup reduces to 2x–4x for CamVid and 5x–6x for the SUNY videos. Thus, our approach provides significant speed up, even including the time to construct the supervoxel tree.

Bbout $40\% - 45\%$ of the total inference time is spent on the integer programming to determine the unary cost of "mixed" labels while using α-expansion as the optimization algorithm. The proportion decreases to $20\% - 25\%$ of total inference time when belief propagation is used.

Reduced problem size. Besides computation time, it is also informative to look at the explored portions of the supervoxel tree to get a better understanding of where the computational savings come from. In Figure 2, we show the explored portions of the state space for one frame in each of the three SUNY videos. The leftmost image is the original image, followed by its ground truth labeling. The next four images from left to right are superpixels (since we are only looking at one frame) at different levels of abstraction (coarsest on the left). The blacked out superpixels at each level are the ones which *never* received the *mixed* label and hence were never refined. Eliminating entire superpixel (supervoxel) trees rooted at these blackened superpixels (supervoxels) is the key to our computational speedup.

Accuracy. We do not show any segmentation results in the paper since the quality of the segmentation is the same as what would be obtained by using α-expansion or belief propagation with the chosen energy function.

Accuracy vs time. Another interesting metric to track is the percentage of correctly classified voxels. When the hierarchical algorithm terminates, this percentage reaches the

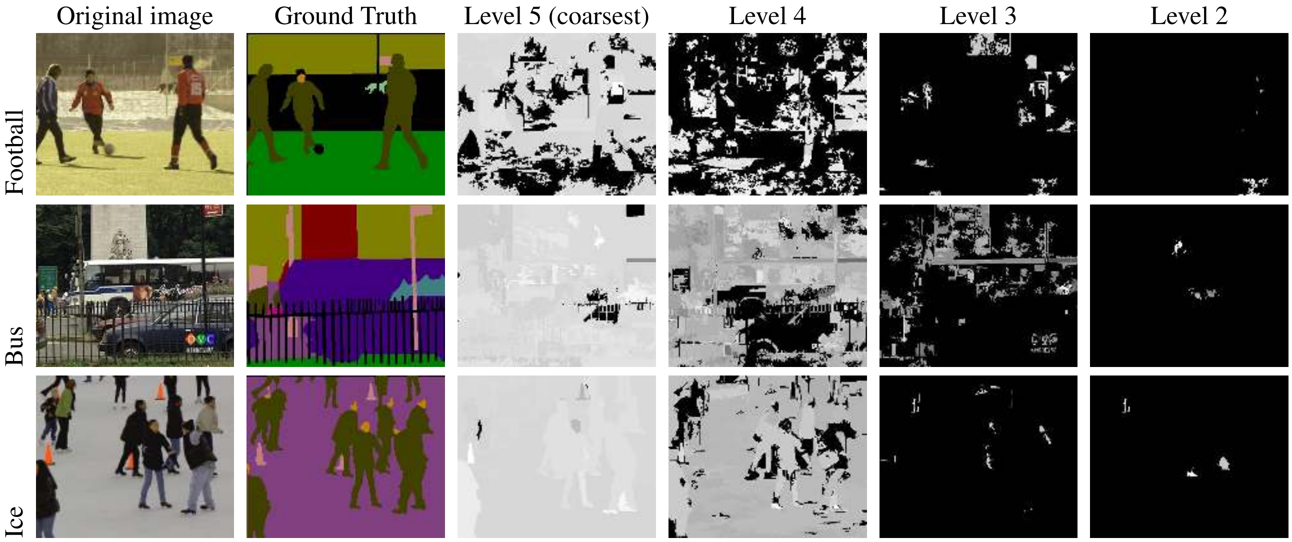

Figure 2. Explored portions of the supervoxel tree. The blacked out portions in each superpixel level denotes the patch of superpixels which were never refined during inference. The top row shows results from the "football" video, the middle row from the "bus" video and the bottom row from the "ice" video (all from the SUNY dataset).

Algorithm		**CamVid**					**SUNY**		
		CamVid1	CamVid2	CamVid3	CamVid4	CamVid5	Bus	Football	Ice
α-expansion	Flat	130.1	137.3	117.6	145.1	140.1	35.3	25.0	32.7
	Coarse-to-fine	32.7	40.9	27.3	43.8	29.4	6.5	2.3	5.3
Belief Propagation	Flat	256.0	270.1	258.3	307.0	319.2	50.3	34.7	50.9
	Coarse-to-fine	50.5	79.1	61.5	107.7	90.5	9.3	4.1	8.3

Table 1. Time taken by the different inference algorithms on different data sets (in minutes). The times reported for the coarse-to-fine case *do not* include supervoxel tree computation time. For the CamVid videos, the speedup is between 3x–5x, while for the SUNY videos the speedup is between 7x–10x. If we include the time of the on-demand refinement scheme discussed in Section 3.5, the overall speedup reduces to 2x–4x for CamVid and 5x–6x for the SUNY videos.

same value that would be obtained by running inference on the flat problem formulation. As shown in Figure 3, this final accuracy lies between 55% and 75%. However, there seems to be no clear trend in how this accuracy is achieved as a function of iterations. For the "bus" video, the accuracy quickly spikes up and then reaches a plateau, while for "ice", it spikes up after a few iterations. A surrogate for this accuracy (the percentage accuracy is often unavailable since there is no ground truth) is the cost function. We can use the cost function to design an anytime version of the algorithm, where termination could be guided by sharp spikes (or the lack thereof) in the cost function.

5. Conclusion

We have presented a general coarse-to-fine scheme for video segmentation. The key intuition behind the proposed solution is the fact that the set of likely label assignments is exponentially smaller than the set of all possible label assignments. A flat problem formulation works with the latter large set, while we use an abstraction scheme (namely supervoxel trees) to identify the former smaller set and work on the smaller problem. The framework is general since it can use any optimization algorithm to find the optimal label for the intermediate problems. It is also exact since it uses admissible heuristic costs for the coarser supervoxel potentials. Results using α-expansion and belief propagation on two different video datasets showed speedups ranging from 2x–10x. As expected, the speedup obtained is larger for videos with more spatio-temporal continuity.

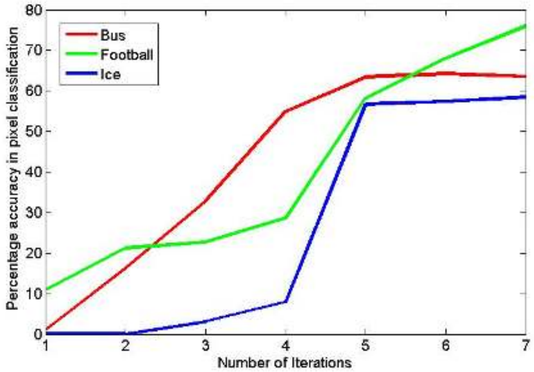

Figure 3. Percentage of correctly classified supervoxels after every iteration of the coarse-to-fine belief propagation algorithm.

As with any general framework, there remains a fair bit of exploration to do. Other abstraction schemes and op-

	CamVid					**SUNY**		
	CamVid1	CamVid2	CamVid3	CamVid4	CamVid5	Bus	Football	Ice
#supervoxels (coarsest)	215	32	54	45	47	29	18	18
#supervoxels (finest)	20528	17688	14773	14839	14393	9422	8353	10163
Time taken	9.1	7.6	7.4	8.3	7.2	2.7	2.3	3.0

Table 2. Number of supervoxels at the coarsest and finest level, alongwith the supervoxel generation time using the modified on-demand supervoxel generation scheme.

timization algorithms could yield better results (for other specific data sets). There is also the accuracy to computation time trade-off in the heuristics computation. Another direction would be to compromise on the exact nature of the solution and design really fast (and perhaps, slightly more inaccurate) segmentation algorithms.

References

[1] igraph. http://igraph.sourceforge.net/. 6

[2] Libsvx. http://www.cse.buffalo.edu/ jcorso/r/supervoxels/. 6

[3] Python graph. http://code.google.com/p/python-graph/. 6

[4] D. Batra and P. Kohli. Making the right moves: Guiding alpha-expansion using local primal-dual gaps. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1865–1872. IEEE, 2011. 3

[5] E. Boros and P. L. Hammer. Pseudo-boolean optimization. *Discrete Appl. Math.*, 123(1-3):155–225, Nov. 2002. 5

[6] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2001. 4, 6

[7] G. J. Brostow, J. Fauqueur, and R. Cipolla. Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters*, 2008. 2, 6

[8] S. Chatterjee and S. Russell. A temporally abstracted Viterbi algorithm. In *UAI*, pages 96–104, 2011. 2, 4, 5

[9] A. Chen and J. Corso. Propagating multi-class pixel labels throughout video frames. In *Western New York Image Processing Workshop (WNYIPW)*, pages 14 –17, 2010. 1, 2, 6

[10] **J. J.. Corso**, A. Yuille, and Z. Tu. Graph-Shifts: Natural Image Labeling by Dynamic Hierarchical Computing. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2008. 3

[11] D. Cremers and S. Soatto. Motion competition: A variational framework for piecewise parametric motion segmentation. *Int. Journal of Computer Vision*, 62(3):249–265, 2005. 1

[12] T. Darrel and A. Pentland. Robust estimation of a multilayered motion representation. In *IEEE Workshop on Visual Motion*, pages 173–178, 1991. 1

[13] P. Felzenszwalb and D. Huttenlocher. Efficient Graph-Based Image Segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004. 2, 3, 4, 6

[14] P. Felzenszwalb and D. Huttenlocher. Efficient belief propagation for early vision. *International Journal of Computer Vision*, 70:41–54, 2006. 10.1007/s11263-006-7899-4. 2, 3

[15] G. Floros and B. Leibe. Joint 2d-3d temporally consistent semantic segmentation of street scenes. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2012. 1

[16] B. Fulkerson, A. Vedaldi, and S. Soatto. Class segmentation and object localization with superpixel neighborhoods. In *IEEE Int. Conf. on Computer Vision*, 2009. 3

[17] E. Galmar, T. Athanasiadis, B. Huet, and Y. S. Avrithis. Spatiotemporal semantic video segmentation. In *MMSP*, pages 574–579. IEEE Signal Processing Society, 2008. 1

[18] M. Grundmann, V. Kwatra, M. Han, and I. Essa. Efficient hierarchical graph-based video segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2010. 1, 2, 3, 6

[19] A. Jain, L. Zappella, P. McClure, and R. Vidal. Visual dictionary learning for joint object categorization and segmentation. In *European Conference on Computer Vision*, 2012. 3

[20] P. Kohli, L. Ladicky, and P. H. S. Torr. Robust higher order potentials for enforcing label consistency. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2008. 3

[21] M. Kumar and D. Koller. MAP estimation of semi-metric MRFs via hierarchical graph cuts. In *Proceedings of the Twenty-fifth Conference on Uncertainty in AI (UAI)*, 2009. 2, 3

[22] L. Ladicky, C. Russell, P. Kohli, and P. Torr. Associative hierarchical CRFs for object class image segmentation. In *IEEE Int. Conf. on Computer Vision*, 2009. 2

[23] I. Laptev. On space-time interest points. *International Journal of Computer Vision*, 64(2-3):107–123, 2005. 6

[24] I. Laptev and T. Lindeberg. Space-time interest points. In *IEEE International Conference on Computer Vision*, pages 432–439, 2003. 6

[25] V. S. Lempitsky, A. Vedaldi, and A. Zisserman. A pylon model for semantic segmentation. In *Neural Information Processing Systems*, 2011. 2

[26] C. Raphael. Coarse-to-Fine Dynamic Programming. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:1379–1390, 2001. 2

[27] S. J. Russell and P. Norvig. *Artificial Intelligence - A Modern Approach*. Pearson Education, 3rd edition, 2010. 4

[28] J. Shi and J. Malik. Motion segmentation and tracking using normalized cuts. In *IEEE Int. Conf. on Computer Vision*, pages 1154–1160, 1998. 1

[29] D. Singaraju and R. Vidal. Using global bag of features models in random fields for joint categorization and segmentation of objects. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011. 3

[30] R. Vidal, R. Tron, and R. Hartley. Multiframe motion segmentation with missing data using PowerFactorization and GPCA. *International Journal of Computer Vision*, 79(1):85–105, August 2008. 1

[31] C. Xu and J. Corso. Evaluation of super-voxel methods for early video processing. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012. 1, 3