Video Segmentation as a Distributed Convex Optimization Problem using Primal Decomposition

Animesh Garg* ANIMESH.GARG@BERKELEY.EDU
Jeff Mahler* JMAHLER@BERKELEY.EDU
Shubham Tulsiani* SHUBHTULS@BERKELEY.EDU
Department of EECS, UC Berkeley, CA 94720

Abstract

Getting exact video segmentations for tracking and recognition is a challenging problem. A majority of existing methodstrack but provide a bounding box rather than a an exact foreground mask for the object. For real worl applications of perception, like robotics, the silhoutte of the object perhaps even pose need to be known for hope of success in manipulation tasks.

We propose a method in this study which formulated the problem of video segmentation as a Markov random field. However solving such a large graph to global optimality may be computationally expensive. Hence we propose a distributed method using Primal decomposition.

1. Introduction

The problem of video segmentation is of interest for many areas. (Tziritas, 2007; Komodakis et al.) and (Tsai et al., 2010) have looked at the problem of modelling the MRF in terms of energies. The solution strategy they use is Dual decomposition but without integer programming.

Our study explores the use of state-of-the-art integer program solvers. Modelling integers allows us to capture more rich features in video which are usually not directly put in current models. From a given video sequence, and user intialized object(s) of interest, the aim is to track the region(s) of interest through the subsequent image frames in the video. Majority of other methods which address the problem provide locally optimal solutions. Such an approach though successful in some applications requires a substantial amount of human intervention at several points in the solution such as in cases of occlusion, change in pose, shape and color and in extreme cases object (or a part of object) egresses the frame and re-enters later.

2. Related Work

Automatic video segmentation has been studied and solutions which provide a bounding box track. However several applications require a higher fidelity track to accurately localize the foreground in the video. This problem is of importance in several areas of research like behavior analysis in animal studies and object tracking for robot perception, among others. The problem of foreground tracking and more generally, the problem of video segmentation are well studied in the field of Computer Vision. Many of the prominent approaches treat the problem of video segmentation as that of combining segments obtained independently for various frames.

A standard approach is to leverage standard image segmentation algorithms like (Carreira & et al., 2010) and filter the segmentation candidates to ensure consistency accross frames. Another commonly used approach where the generated segmentations rely on temporal information is to incoorporate the optical flow information of the frame along with the intensity as an additional layer image and use segmentation algorithms for this layered image eg (Leordeanu et al.). We observe that these common methods either do not incoorporate temporal information to generate segmentations or only use local temporal signals (from the next frame). In this work we aim to experiment with formulations which jointly capture the temporal information for the whole video aand we therefore tackle the problem of video segmentation as a global optimization problem.

(Li et al., 2013b) approached the problem with an approach using a unsupervised approach called as Segmented Pool Tracking with Composite Statistical Inference. It generates a pool of segments for each frame via a multiple figure-ground segmentation algorithm. Thereafter, it computes appearance features of each segment in all tracks. Initialize a segment track for each segment in the first frame. Si-

This is a draft version for EE227-BT Fall 2013 Project.
* denotes equal contribution.

multaneously learn appearance models for all tracks using multi-output regression. It then greedily matches the tracks across images retaining only the highest matching pair of tracks. Finally it performs a composite statistical inference [(Li et al., 2013a)] which adds temporal consistency to the solution.

Out formulation explicitly captures all the generic requirements of the video segmentation problem with out encoding domain specific information. The major advantage of our approach is the near exact solution albeit at the computational complexity. However with a careful distributed implementation of the problem generation step, commercially available solvers have shown potential to solve large scale LPs. The preprocessor finds redundant and inactive constraints and solves the problems in reasonable times.

3. Problem Formulation

Notations and Variables

- We denote the video volume by I. A pixel in I is indexed by its location in space as well as time and is denoted by I_{ijt}
- We wish to recover a complete segmentation of the video into foreground and background. This labelling is captured by the variable X where $X_{ijt} \in \{0, 1\}$
- The time continuity between frames in a video implies that any pixel in a given frame corresponds to some pixel in the next frame. We capture this notion by a weak correspondence between a pixel and its neighbors in the next frame. The correspondence weights for a pixel are denoted by W_{ijt}^{ab} $(a, b \in \{-h, .., h\})$ i.e we define a correspondence weight variable between each pixel and the $(2h+1)X(2h+1)$ grid surrounding it in the next frame.
- By $N_s(i, j, t)$, we denote the indices of the pixels in the spatial neighborhood of the pixel (i, j, t)
- We also define variables $U, V, \overline{U}, \overline{V}$ which capture the average motion direction of a pixel between consecutive frames in X, Y directions respectively. We also denote the average direction of motion of the neighborhood of a pixel by pseudo-variables $(\overline{U}_{ijt}, \overline{V}_{ijt})$. These variables are defined in terms of the previously defined variables as follows -

$$U_{ijt} = \sum_{a,b \in \{-h,..,h\}} aW_{ijt}^{ab} \tag{1}$$

$$V_{ijt} = \sum_{a,b \in \{-h,..,h\}} bW_{ijt}^{ab} \tag{2}$$

$$(\overline{U}_{ijt}, \overline{V}_{ijt}) = \frac{1}{|N_s(i, j, t)|} \sum_{Y \in N_{s(i,j,t)}} (U_Y, V_Y) \tag{3}$$

- Note that given U and V, we can recover the location that a given pixel gets mapped to in the next frame. Given this location, we can find interpolation weights for the surrounding pixels in the next frame and obtain a feasible W. Therefore, we can obtain W given U, V (and vice-versa as shown above). In the subsequent sections, we will define objectives and constraints in terms of U, V, W but not all of them will be 'real' variables. It should be clear from the context which variables are being optimized over and which ones being used for notational convenience.

Objective

$$\min_{X,W} \; \lambda_1 A(X, I) + \lambda_2 S(X) + \lambda_3 T(X, W) \tag{4}$$

$$+\lambda_4 F(W, I) + \lambda_5 C(W) + \lambda_6 M(W)$$

subject to $W \geq 0, \forall(i, j, t) X_{ijt} \in \{0, 1\}, \sum_{a,b} W_{ijt}^{ab} = 1$ and

$$\forall t |\sum_{i,j} X_{ijt} - \sum_{i,j} X_{ij(t+1)}| \leq \sigma \sum_{i,j} X_{ijt}$$

The objective function comprises of various penatly terms which are explained below. The last constraint specifies that the number of foreground pixels in do not change rapidly between consecutive frames.

Appearance Model $A(X, I)$

Given the initial user labelled segmentation X', we can form a foreground model and a corresponding penalty function $f_{I,X'}$ for a pixel's label given its value. We then define the unary potential as follows -

$$A(X, I) = \sum_{i,j,t} f_{I,X'}(X_{ijt}, I_{ijt}) \tag{5}$$

Spatial Labelling Coherence $S(X)$

We want to drive the system towards a labelling where neighbouring pixels have similar labels. The spatial labelling coherence term defined below encapsulates this.

$$S(X) = \sum_{i,j,t} \sum_{Y \in N_s(i,j,t)} |X_{ijt} - X_Y| \tag{6}$$

Temporal Labelling Coherence $T(X, W)$

For a given pixel, the corresponding pixel in the next frame should also have the same label. We formalize this notion using the penalty function below.

$$T(X,W) = \sum_{i,j,t} \sum_{a,b\in\{-h,..,h\}} W_{ijt}^{ab}|X_{ijt} - X_{i+a,j+b,t+1}| \tag{7}$$

Flow Similarity $F(W,I)$

For each pixel, the corresponding pixel in the next frame should be similar. This is enforced by the flow similarity defined below.

$$F(X,I) = \sum_{i,j,t} \sum_{a,b\in\{-h,..,h\}} W_{ijt}^{ab}|I_{ijt} - I_{i+a,j+b,t+1}| \tag{8}$$

Flow Continuity $C(W)$

The direction of movement of pixels is continuous over a small spatial neighbourhood. We therefore penalize rapid variations in flow as follows-

$$C(W) = \sum_{i,j,t} |U_{ijt} - \overline{U}_{ijt}| + |V_{ijt} - \overline{V}_{ijt}| \tag{9}$$

Momentum Continuity $M(W)$

It also needs to be enforced that the velocity of a pixel and its corresponding pixel in the next frame do not vary rapidly. This is ensured by the momentum continuity terms defined below

$$M(W) = \sum_{i,j,t} \sum_{a,b\in\{-h,..,h\}} W_{ijt}^{ab}(|a - \overline{U}_{i+a,j+b,t+1}| + |b - \overline{V}_{i+a,j+b,t+1}|) \tag{10}$$

4. Algorithm

It is clear that the minimization problem formulated above cannot be solved directly using a standard optimization solver. A common relaxation in similar problems is to allow the discreet variable to be continuous and threshold the solution at the end. Even if we follow this approach and relax the optimization problem by allowing X to be a continuous variable, the temporal labelling coherence penalty in the objective function would not be not jointly convex w.r.t X, W. In this case, we could use a sub-gradient descent based method to reach a local minima. However, we want to refrain from the approach mentioned above as it would end up finding a relaxed solution (local minima) to an already relaxed optimization problem (as we allowed X to be continuous instead of discreet). We cannot expect such a solution to be very robust. Hence, instead of pursuing an algorithm to directly optimize the joint objective function in the discreet and continuous variables, we construct two separate minimization problems over the continuous and discreet variables and alternate between solving them.

Pseudocode

Let $f_I(X,W)$ denote the objective function to be minimized. The algorithm used to minimize the objective function is as follows -

Algorithm 1 $solve(I)$

```
function $solve(I)$
  • $X1 \leftarrow initialSegment()$
  • $W \leftarrow generatePriors(I, X1)$
  • while(!$stoppingCriteria$)
      – $X \leftarrow propogateLabels(I, W)$
      – $W \leftarrow solveWeights(I, X)$
  • return $(X, W)$

function $propogateLabels(I, W)$
  • $X \leftarrow \underset{X}{argmin} f_I(X, W)$
  • return $X$

function $solveWeights(I, X)$
  • $W \leftarrow \underset{W}{argmin} f_I(X, W)$
  • return $W$
```

Analysis and Convergence

The algorithm used above is very intuitive. In order to find the minima for the objective function, we alternate between minimizing over the discreet and continuous variables. We therefore tackle two (comparitively) simpler optimization problems of finding $\underset{X}{argmin} f_I(X,W)$ and $\underset{W}{argmin} f_I(X,W)$ instead of the original complex optimization problem. This is a standard optimization approach analogous to the block coordinate descent method where at each step we find the minima rather than using a gradient/sub-gradient based descent. Note that at each step in the iteration, the value of the objective function decreases. If we draw an analogy to a two-player game with both players alternatively minimizing their cost given the other's strategy, this approach would converge to a Nash

Equilibrium. Thus, we can claim that the algorithm mentioned above converges to a point which is a local minima with respect to both X, W.

5. Alternate Objective for Tractability

5.1. Motivation

The original problem formulation we mentioned above encapsulates all the semantic properties that we would like the obtained solution to have (flow similarity, labelling coherence etc.). However, when we consider the problem of finding $\underset{W}{argmin} f_I(X, W)$ where $f_I(X, W) = \lambda_1 A(X, I) + \lambda_2 S(X) + \lambda_3 T(X, W) + \lambda_4 F(W, I) + \lambda_5 C(W) + \lambda_6 M(W)$, we encounter the following difficulties -

Non-Convexity

The momentum continuity penalty as defined above i.e. $M(W) = \sum_{i,j,t} \sum_{a,b\in\{-h,..,h\}} W_{ijt}^{ab}(|a - \overline{U}_{i+a,j+b,t+1}| + |b - \overline{V}_{i+a,j+b,t+1}|)$ is non-convex w.r.t W (beacause U is linear w.r.t W so $M(W)$ has a product of variables of W_t, W_{t+1}).

Large Number of Variables

The size of the variable W is $(2h+1)^2 * |I|$. For a reasonable sized video segment, this value becomes more than 10^7 even if we downsample the video. This makes it very difficult to solve the minimization problem of this order with limited computational resources in a reasonable time.

We had earlier observed that the variables W and (U, V) can be approximated from each other. Since the size of (U, V) is only $|I|$, we can define penalty functions equivalent/similar to the above in terms of U, V instead of W.

5.2. Brightness constancy assumption and Horn-Schunk algorithm

Before we describe the reformulation of the original objective function, we briefly describe a well established computer vision algorithm that motivates and justifies the relaxations used by us. The Horn-Schunk algorithm (Horn & Schunck, 1981) addresses the problem of finding optical flows i.e. given an image pair (I_1, I_2), we want to find the optical flow field (U_1, V_1) for each pixel in I_1. They use a first order approximation over image intensity values and therefore use $(\frac{\partial I}{\partial x}u + \frac{\partial I}{\partial x}v + \frac{\partial I}{\partial t})$ as a proxy for the image intensity difference of a pixel in I_2 at the location (u, v) away from the current pixel. Formally, the Horn-Schunk algorithm minimizes the following objective -

$$E(U,V) = \sum_{i,j}[(\frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + \frac{\partial I}{\partial t})^2 + \alpha^2(\|\Delta u\|^2 + \|\Delta v\|^2)] \tag{11}$$

Here, $(\Delta u, \Delta v)$ are the spatial derivatives of the flow field. Drawing the analogies between our original objective and the objective in Horn-Schunk algorithm, we observe that the first term $\sum_{i,j}[(\frac{\partial I}{\partial x}u + \frac{\partial I}{\partial x}v + \frac{\partial I}{\partial t})^2$ captures the Flow Similarity penalty (that the corresponding pixel in the next frame should be similar). The terms $(\|\Delta u\|^2 + \|\Delta v\|^2)$ represent the flow continuity (that neighboring pixels have similar flow). For our problem, if we assume that $(\Delta u, \Delta v)$ the spatial as well as the temporal derivatives of the flow field, we can also capture the Momentum Continuity penalty. Note that using penalties and first-order approximations similar to the Horn-Schunk algorithm, we obtain a convex objective. also, we can formualte the optimization problem in terms of the variables U, V instead of W and achieve a huge reduction in terms of the number of variables. We also see that if we use the $L1$ norm penalty instead of the $L2$ norm, we can obtain an LP using auxillary variables. These minor modifications to the original objective fnuction help us retain the original penalties while obtaining a formulation with a much lower complexity than the original one.

5.3. Reformulated problem

We incoorporate the relaxations motivated by the Horn-Schunk algorithm and modify penalties to make an LP formulation feasible. The final minimization problem that we obtain is as follows -

Objective

$$\min_{X,W} \ \lambda_1 A'(X,I) + \lambda_2 S'(X) + \lambda_3 T'(X,W) \tag{12}$$

$$+\lambda_4 F'(W,I) + \lambda_5[C'(W) + M'(W)]$$

subject to $W \geq 0, \forall(i,j,t) X_{ijt} \in \{0,1\}, \sum_{a,b} W_{ijt}^{ab} = 1$ and

$$\forall t |\sum_{i,j} X_{ijt} - \sum_{i,j} X_{ij(t+1)}| \leq \sigma \sum_{i,j} X_{ijt}$$

The objective function comprises of various penatly terms which are explained below. The last constraint specifies that the number of foreground pixels in do not change rapidly between consecutive frames.

APPEARANCE MODEL $A'(X, I)$

We compute an object model using the segmentation in the first frame. Using this, we pre-compute a cost matrix MA which determines the cost between $[0, 1]$ of a pixel (i, j, t) being a foreground pixel. Once precomputed, this cost matrix A is fixed for the subsequent stages of the optimization. The appearance cost is then computed as follows -

$$A'(X, I) = \sum_{i,j,t} A_{i,j,t} * X_{ijt} + (1 - A_{i,j,t}) * (1 - X_{ijt}) \quad (13)$$

SPATIAL LABELLING COHERENCE $S(X)$

$$S'(X) = S(X) = \sum_{i,j,t} \sum_{Y \in N_s(i,j,t)} |X_{ijt} - X_Y| \quad (14)$$

FLOW SIMILARITY $F(W, I)$

$$F'(W, I) = F(U, V, I) = \sum_{i,j,t} |\frac{\partial I}{\partial x} U_{i,j,t} + \frac{\partial I}{\partial y} V_{i,j,t} + \frac{\partial I}{\partial t}| \quad (15)$$

TEMPORAL LABELLING COHERENCE $T(X, W)$

Similar to the firsr order approximation to the intersity field I, we can also make a first order approximation to the label value field X. This gives us the following penalty -

$$T'(X, W) = T(X, U, V) = \sum_{i,j,t} |\frac{\partial X}{\partial x} U_{i,j,t} + \frac{\partial X}{\partial y} V_{i,j,t} + \frac{\partial X}{\partial t}| \quad (16)$$

FLOW CONTINUITY $C(W)$ + MOMENTUM CONTINUITY $M(W)$

$$C'(W) + M'(W) = \sum_{i,j,t} \|\Delta u\|_1 + \|\Delta v\|_1 \quad (17)$$

6. Implementation Details

Note that the reformulation mentioned above does not neccesiate any change in the proposed optimization algorithm as we still need to follow the block-gradient descent method (since even the reformulation does not make the objective jointly convex in X, W and X is still required to be discreet). In this section, we descibe the finer implementation details of our algorithm.

6.1. Initialization

We have shown earlier that the objective function is not jointly convex over X, W. So, the solution that the algorithm converges to is highly dependent on the initialization. Therefore, we need to determine a reasonable initialization for the weight variables. We experiment with two standard optical flow algorithms (Horn & Schunck, 1981), (Bruhn et al., 2005) to initialize the flow variables for each video frame (and thus the weight variables). The implementation is as follows -

Algorithm 2 $generatePriors(I)$

```
function $generatePriors(I)$
  • for $t = 1 : T - 1$
      – $(U_t, V_t) = opticalFlow(I_t, I_{t+1})$
  • $W \leftarrow UVtoWeights(U, V)$
  • return $W$
```

6.2. Unary Appearance Cost

To determine the unary costs, we learn a foreground model based on the intitial frame segmentation. We train a random forest classifier using for patches using the patches around foreground pixels in the intial frame as positive examples and the patches around other pixels as negative examples. We then predict the probability of a pixel in a given frame being a foreground pixel by classifying the patch surrounding it using the trained classifier. The foreground label cost for the given pixel is stored as $1 - classifierProbability(I_{ijt})$.

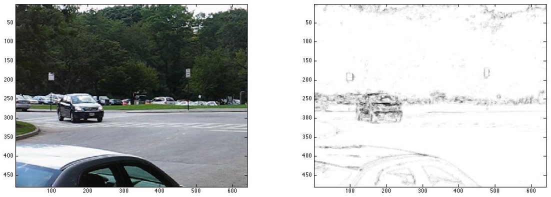

Figure 1. Appearance Costs visualization : Low intensity represents low foreground cost (object being tracked is upper car)

6.3. Finding flows given label assignments

6.4. Propogating labels

7. Experiments and Metrics

As presented in the model in Section 3 we have a complete optimization model with several integer variables for foreground-background labels.

We will test the performance of our solution on the Berkeley Motion Segmentation Dataset as provided by (Brox & Malik, 2010). The dataset has 26 video sequences with pixel-accurate segmentation annotation of moving objects. A total of 189 frames are annotated.

We will evaluate results from our approach and compare the performance with that of (Felzenszwalb et al., 2010), (Komodakis et al.) and (Brox & Malik, 2010) on this dataset.

Furthermore multiple decoupling strategies will implementation and compared, like decoupling time frames v/s decoupling in space. Finally a dual decomposition method with also be explored and compared qualitatively with (Tziritas, 2007).

We plan on completing the implementation in MATLAB with the use of CVX and CPLEX optimization libraries.

8. Conclusions and Future Work

References

Brox, Thomas and Malik, Jitendra. Object segmentation by long term analysis of point trajectories. In *Computer Vision–ECCV 2010*, pp. 282–295. Springer, 2010. URL http://lmb.informatik.uni-freiburg.de/resources/datasets/.

Bruhn, Andrés, Weickert, Joachim, and Schnörr, Christoph. Lucas/kanade meets horn/schunck: Combining local and global optic flow methods. *Int. J. Comput. Vision*, 61(3):211–231, February 2005. ISSN 0920-5691. URL http://dl.acm.org/citation.cfm?id=1028916.1035417.

Carreira, Joao and et al. Constrained parametric min-cuts for automatic object segmentation. 2010.

Felzenszwalb, PF, Pap, G, Tardos, E, and Zabih, R. Globally optimal pixel labeling algorithms for tree metrics. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pp. 3153–3160. IEEE, 2010. URL http://www.computer.org/portal/web/csdl/doi/10.1109/CVPR.2010.5540077http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5540077.

Horn, Berthold K. P. and Schunck, Brian G. Determining optical flow. *ARTIFICAL INTELLIGENCE*, 17:185–203, 1981.

Komodakis, Nikos, Paragios, Nikos, and Tziritas, Georgios. *IEEE transactions on pattern analysis and machine intelligence*. ISSN 1939-3539. doi: 10.1109/TPAMI.2010.108.

Leordeanu, Marius, Sukthankar, Rahul, and Sminchisescu, Cristian. In Fitzgibbon, Andrew W., Lazebnik, Svetlana, Perona, Pietro, Sato, Yoichi, and Schmid, Cordelia (eds.), *ECCV (4)*, pp. 516–529. Springer. ISBN 978-3-642-33764-2.

Li, Fuxin, Carreira, Joao, Lebanon, Guy, and Sminchisescu, Cristian. Composite statistical inference for semantic segmentation. *2013 IEEE Conference on Computer Vision and Pattern Recognition*, 0:3302–3309, 2013a. ISSN 1063-6919. doi: http://doi.ieeecomputersociety.org/10.1109/CVPR.2013.424.

Li, Fuxin, Kim, Taeyoung, Humayun, Ahmad, Tsai, David, and Rehg, James M. Video segmentation by tracking many figure-ground segments. In *ICCV*, 2013b.

Tsai, David, Flagg, Matthew, and Rehg, James. Motion Coherent Tracking with Multi-label MRF optimization. In *Procedings of the British Machine Vision Conference 2010*, pp. 56.1–56.11. British Machine Vision Association, 2010. ISBN 1-901725-40-5. doi: 10.5244/C.24.56. URL http://www.bmva.org/bmvc/2010/conference/paper56/index.html.

Tziritas, N.K.N.P.G. Optimization of Discrete Markov Random Fields via Dual Decomposition. Technical Report April, 2007. URL http://www.csd.uoc.gr/~komod/publications/docs/Dual_Decomposition_TR.pdf.