Video Sequence Boundary Labeling with Temporal Coherence – Supplementary Material

Petr Bobák[1[0000−0003−1914−4592]], Ladislav Čmolík[2[0000−0002−8546−6568]], and Martin Čadík[1,2[0000−0001−7058−9912]]

[1] Faculty of Information Technology, Brno University of Technology, Czechia
{ibobak,cadik}@fit.vutbr.cz
[2] Faculty of Electrical Engineering, Czech Technical University in Prague, Czechia
{cmolikl,cadikm}@fel.cvut.cz

1 Introduction

This document is a supplementary material to the Video Sequence Boundary Labeling with Temporal Coherence paper. First, we describe an implementation of the *label box interval to row assignment subproblem* based on approximation. The implementation provides real-time performance even for 100 labels, however the results of the implementation, unlike the results of the implementation based on optimization presented in the paper, are sub-optimal. Further, we present detailed information about performance of both the implementation based on optimization and the implementation based on approximation. Last, but not least, we present detailed description of the tasks that participants fullfilled during the user experiment. We expect that the reader has already read the paper and will look into the paper while reading the supplementary material.

2 Label box interval to row assignment based on approximation

The second method that solves the *label box interval to row assignment subproblem* is based on creation of the set of maximum non-overlapping intervals. The positions of label boxes should correspond to distances of the labeled objects from the camera. The closest label boxes should be in the lowest row. To fulfill this requirement, the label box intervals are processed by the average distance from the camera d_{α_i} of their associated anchor interval. Further, we assign initial position $min_{\lambda_i} = max_{\alpha_i} - w_{\lambda_i}$ to each label box interval λ_i, but the position of the interval can change unless its associated anchor interval α_i is not entirely inside of the label box interval λ_i and the label box interval λ_i does not overlap with any other label box interval.

We use a binary tree T to represent the maximum set of non-overlapping intervals N. In the binary tree T a node n represents one label box interval λ_i, the left child of the node represents a label box interval that is entirely on the left side from the interval λ_i and the right child of the node represents a label box interval that is entirely on the right side from the interval λ_i.

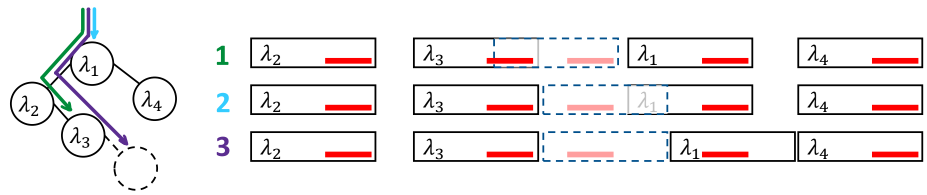

Figure 1. A label box interval (dashed) is inserted into the binary tree in the third attempt. In the first two cases, conflicts with other interval are resolved. The red lines represent the anchor intervals associated with the label box intervals.

The algorithm to solve the *label box interval to row assignment subproblem* is as follows:

1. Create empty list of rows R.
2. Create list I containing all label box intervals.
3. While list I is not empty:
   (a) Create empty binary tree T.
   (b) Try to insert each label box interval $\lambda_i \in I$ to the tree T.
   (c) Add the tree T as new row to R.
   (d) Remove the intervals in the tree T from the list I.

The crucial part of the previous algorithm is the insertion of the label box interval to the tree T. Note that for each label box interval λ_i, we know its minimum min_{λ_i} and maximum max_{λ_i} x-coordinates and minimum min_{α_i} and maximum max_{α_i} x-coordinates of its associated anchor interval. Further, for each node n representing the interval λ_i, we are able find node m in the tree T representing the closest label box interval on the right side from λ_i that is already inserted to the tree with the function $m = next(n)$. Here, we show the algorithm for testing node n representing the interval λ_j with an inserted interval λ_i:

1. If the label box interval λ_i is entirely on the left side from the label box interval λ_j then:
   (a) If the left child of the node n exists then repeat the test with the left child of the node n.
   (b) Otherwise, create new node for the label box interval λ_i and set it as the left child of the node n.
2. Otherwise, if the label box interval λ_i is entirely on the right side from the label box interval λ_j then:
   (a) If the right child of the node n exists then repeat the test with the right child of the node n.
   (b) Otherwise, create new node for the label box interval λ_i and set it as the right child of the node n.
3. Otherwise:

(a) If $max_{\lambda_j} > min_{\lambda_i} \wedge max_{\lambda_j} < min_{\alpha_i}$ (see case 1 in Fig. 1, $\lambda_j = \lambda_3$). Move the label box interval λ_i such that $min_{\lambda_i} = max_{\lambda_j}$ and try to insert interval λ_i to the tree T again (in the example in Fig. 1 this will result in case 2).

(b) Otherwise, if $max_{\lambda_i} > min_{\lambda_j} \wedge max_{\lambda_i} < min_{\alpha_j}$ (see case 2 in Fig. 1, $\lambda_j = \lambda_1$), try to move the label box interval λ_j to the right by distance $max_{\lambda_i} - min_{\lambda_j}$. If the label box interval λ_j is moved, try to insert the label box interval λ_i to the tree T again (in the example in Fig. 1 this will result in case 3). Otherwise, the label box interval λ_i cannot be inserted into the tree T.

(c) When neither 3a nor 3b are satisfied, the label box interval λ_i cannot be inserted into the tree T.

The algorithm for moving the label box interval λ_j represented by the node n to the right by distance Δ is as follows:

1. Calculate maximum distance Δ_{max} the label box interval λ_j can be moved to the right as $min_{\alpha_j} - min_{\lambda_j}$.
2. If $\Delta_{max} < \Delta$ then the label box interval λ_j cannot be moved to the right by the requested distance Δ.
3. Otherwise, find the node $m = next(n)$ representing the closest label box interval λ_c to the right of the label box interval λ_j and calculate the distance Δ_{jc} between the label box intervals λ_j and λ_c as $min_{\lambda_c} - max_{\lambda_j}$.
4. If $\Delta_{jc} \geq \Delta$ then move the label box interval λ_j to the right by distance Δ and the algorithm ends.
5. Otherwise, try to move the label box interval λ_c to the right by the distance $\Delta - \Delta_{jc}$. If the label box interval λ_c is moved then move the label box interval λ_j by the distance Δ. Otherwise, the label box interval λ_j cannot be moved and the algorithm ends.

3 Detailed information about performance of the implemented methods

We used GUROBI 8.0 with MATLAB interface as optimization solver in our implementation based on optimization. The implementation based on approximation was implemented in Java. The running time needed to solve the subproblem *label box interval to row assignment* with the implementation based on optimization is approximately 1.5s (for 40 labels), and the optimal solution for smaller instances (20 labels and less) is found in less than 200ms. The running time needed to solve the subproblem with the implementation based on approximation is below 10ms even for 100 label boxes. See Fig. 2. The measurement was performed for one sequence on Intel® Core i5-3570 @ 3.40GHz with 24GB RAM.

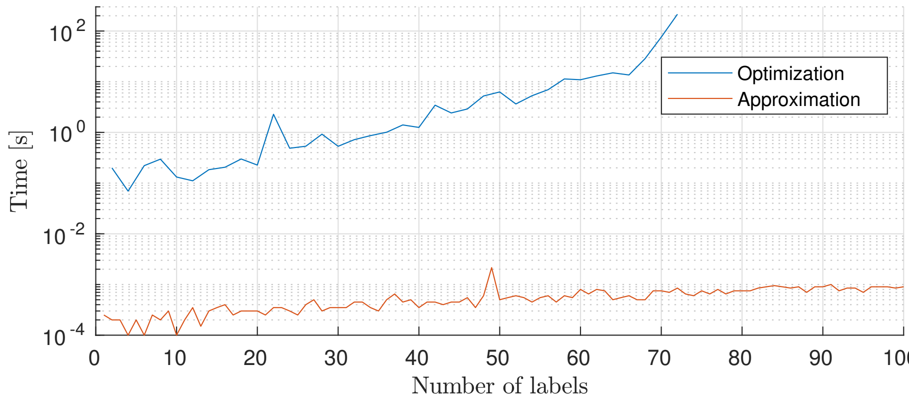

Figure 2. Computation time needed for the optimization and approximation in the first stage in relation to the number of labels.

4 Detailed description of tasks evaluated in the Experiment 1 – Accuracy

In this section, we present the detailed description of the tasks evaluated in the Experiment 1 – Accuracy. To see all the three described tasks being fulfilled please look at the supplementary video. Please note that the participants tried each task on a demo scene to get familiar with the task before they started fullfilling the task.

4.1 Task 1

The first task was to find the label associated to a highlighted anchor. An image with a labeled scene was presented to the participant. After one second, one of the anchors was highlighted. The participant has to find the label associated with the highlighted anchor and press the space-bar. Then only the label boxes (without text) remained visible on the screen and the participant has to click on the associated label box. The participant was instructed to click on free space between label boxes if s/he was not able to find the associated label box. We measured the reaction *time*, measured as a time between highlighting of the anchor and pressing the space-bar, and the *error rate*, measured as the number of wrongly selected labels relative to all selected labels. When the participant could not decide which label belongs to the highlighted anchor we counted this as an error. The whole process was repeated 30 times (10 different labels in 3 different scenes). After each scene we have presented to the participant three statements:

1. It was easy to select the correct label boxes.
2. I was able to find the correct label boxes fast.

3. I was confident that I am selecting the correct label boxes.

The participant indicated their subjective agreement or disagreement with each statement on Likert scale from 1 to 5.

4.2 Task 2

The second task was to find the anchor associated to a highlighted label. The task was similar to the Task 1. An image with a labeled scene was presented to the participant. After one second, one of the label boxes was highlighted. The participant has to find the anchor associated with the highlighted label box and press the space-bar. Then only small boxes around each anchor remained visible on the screen and the participant has to click on the associated anchor. The participant was instructed to click on free space between the boxes if s/he was not able to find the associated anchor. Similarly as in the Task 1 the reaction *time* and the *error rate*. The whole process was repeated 30 times (10 different labels in 3 different scenes). After each scene we have presented to the participant three statements:

1. It was easy to select the correct anchors.
2. I was able to find the correct anchors fast.
3. I was confident that I am selecting the correct anchors.

The participant indicated their subjective agreement or disagreement with each statement on Likert scale from 1 to 5.

4.3 Task 3

The third task was to follow certain label moving in time and then select the label. The task was similar to the tasks 1 and 2. An image with a labeled scene was presented to the participant. After one second, one of the label boxes was highlighted. The participant pressed space-bar, then the highlight of the label box disappeared and the animation started playing with the speed of 10 frames per second. The participant has to follow movement of the initially highlighted label box. After two seconds the animation stopped and only the label boxes (without text) were displayed on the screen. The participant should click to the label box that s/he was following. The participant was instructed to click on free space between the boxes if s/he was not able to find the correct label box. In this task, we measured the *error rate* only. Again, the whole process was repeated 30 times (10 different labels in 3 different scenes). After each scene we have presented to the participant two statements:

1. The label layout made it easy to follow the label boxes.
2. I didn't have to focus hard to be able to follow the labels.

The participant indicated their subjective agreement or disagreement with each statement on Likert scale from 1 to 5.