Video Sequence Boundary Labeling with Temporal Coherence

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

Abstract We propose a method for video sequence boundary labeling which maintains the temporal coherence. The method is based on two ideas. We limit the movement of the label boxes only to the horizontal direction, and reserve free space for the movement of the label boxes in the label layout. The proposed method is able to position label boxes in video sequence on a lower number of rows than existing methods, while at the same time, it minimizes the movement of label boxes. We conducted an extensive user experiment where the proposed method was ranked the best for panorama video sequences labeling compared to three existing methods.

Keywords: labeling · boundary labeling · temporal coherence

1 Introduction

Labels and short textual annotations, are used to communicate the position of objects together with additional information about them (e.g., names of the objects) in a single image. The position of the labels in relation to the labeled objects (i.e., the label layout) is crucial for functional labeling. In this work, we focus on boundary labeling, where the labeled objects are approximated with points denoted as anchors. The labels are enclosed in label boxes positioned on the top side of the scene so that no pair of label boxes overlaps. The labels are associated with the labeled objects by vertical leader lines that interconnect the label boxes with the anchors.

We specifically focus on boundary labeling of panorama video sequences. Imagine a footage from a drone flying through the mountain terrain (or a city full of skyscrapers). The labels and corresponding positions of mountain summits can be obtained from geo-referred terrain models using camera pose estimation techniques [2]. For various panorama boundary labeling examples see Fig. 2. Creating label layouts for such video sequences introduce the problem of *temporal coherence* of the resulting label layouts. In other words, the labels should not jump abruptly from one position to another, but should keep their positions or move in a predictable manner.

In this work, we present three main contributions: (1) A novel algorithm designed both for video sequences and images. The algorithm is capable of positioning the labels on a lower number of rows than existing algorithms, making the label layout more compact in vertical direction. (2) For video sequences the algorithm minimizes movement of the labels during playback of the sequences. If the labels move during the playback, they move in a predictable manner. (3) We present the results of a user study designed to evaluate label layouts of panorama video sequences. In the study we have evaluated our algorithm against three existing boundary labeling methods. The results of the study show that the proposed method is preferred over the other three methods with statistical significance.

2 Related Work

In this section, we divide the boundary labeling methods into two groups based on the flexibility of the label boxes. Finally, we discuss methods that provide temporally coherent movement of the label boxes.

The methods working with *fixed labels* take as the input a set of anchors and a set of label boxes positioned on top of the scene. The task is to assign one label box to each label and connect each label box with the corresponding anchor with a leader line. Bekos et al. [3] introduced a method for boundary labeling where a set of anchor points is connected with a set of predefined label boxes positioned in one or up to three rows with rectilinear leader lines. The method finds the leader lines whose combined length is minimal. Benkert et al. [5] later showed that better label layouts can be produced if we consider criteria such as the number of bends of the leader lines and distance between the leader lines, in addition to the length of the leader lines criterion.

The methods that work with *flexible labels* take as the input a set of anchor points only. The task is to determine positions of the label boxes and connect each label box with the corresponding anchor with a leader line. Maass and Döllner [15] presented two methods that produce panorama label layouts. In both methods the labels are processed according to the distance of the labeled objects from the camera and the label box is centered with respect to the vertical leader line. Gemsa et al. [10] presented an optimization method that for a set of anchor points positions the label boxes on the lowest possible number of rows using dynamic programming. Each label box is connected with a corresponding anchor point with vertical leader line and no leader line intersects any label box.

The methods that are addressing the temporally coherent movement of label boxes strive to determine such label boxes that do not change their positions abruptly and move in a predictable manner. Götzelmann et al. [11] focus on the labeling of animated 3D objects such as engines with moving pistons. Čmolík and Bittner [7] proposed real-time external labeling technique for 3D objects where the label boxes are moving coherently during slow interaction (e.g., rotation) with the scene. Vaaraniemi et al. [22] first determine positions of label boxes in a 3D space and during interaction with the scene use a force based approach to resolve overlaps of the labels. Maass and Döllner [15] and Tanzgern et al. [20] proposed similar hysteresis approaches to make the movement of label boxes temporally coherent. Kouřil et al. [13] proposed another hysteresis approach adapted for internal label boxes.

Unfortunately, none of the approaches is applicable to the panorama video sequence labeling problem, where a small movement of one label box can decrease the free space available for another label box which in turn can lead to an abrupt change in the position of the label box. Consequently, the change in position can again limit free space available for another label box.

3 Problem Definition

In this section we define the video sequence boundary labeling problem. The input is a video sequence S with $|S|$ frames and a set of label boxes $\mathcal{L}$. Each frame $f_i, i \in 1 \dots |S|$ has the same width w_S and height h_S. Each label box $l_k \in \mathcal{L}$ is visible at least in one frame in the video sequence and has defined width w_{l_k}, height h_{l_k}, and text t_{l_k}. The height h_{l_k} is constant for all label boxes in $\mathcal{L}$.

Each frame f_i has a set of anchors A where each anchor a_j has its position (x_{a_j}, y_{a_j}), $j \in 1 \dots |A|$ and a set of label boxes $L \subseteq \mathcal{L}$ where each label box l_j is associated with the anchor a_j. The task is to find a position of each label box $l \in L$ associated with each anchor $a \in A$ of each frame $f \in S$ so that the label boxes fulfill the following requirements for the video sequence boundary labeling problem:

1. The label boxes are aligned to rows starting from the defined line hl_S, e.g., the horizon.
2. The label boxes do not overlap with each other.
3. [Optional] 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.
4. The label boxes should be positioned on the lowest number of rows possible.
5. The label boxes are connected with the corresponding anchors with vertical leader lines.
6. The leader line is connected to the label box as close to the center of the label box as possible.
7. The movement of the label boxes through the video sequence is temporally coherent. In other words, vertical and horizontal movement of the label boxes is minimized across all the frames in the sequence S.

4 Temporally Coherent Labeling Method

Our approach is based on two fundamental ideas: (1) we restrict the movement of the label boxes only in a horizontal direction and (2) we reserve space for the horizontal movement of the label boxes in the label layout. This way the movement of one label box cannot influence the movement of any other label box, thus requirement 7 is fulfilled. To reserve the space for movement of the label boxes in the label layout, we propose to:

- Create an anchor interval $\alpha_i = [min_{\alpha_i}, max_{\alpha_i}]$ and calculate an average camera-to-anchor distance d_{α_i} for each label box $l_i \in \mathcal{L}$ where min_{α_i} and max_{α_i} are the minimal and maximal x-coordinates of anchors that are associated with the label box l_i through all the frames of the sequence S (see Fig. 1(a)). Similarly, the distance d_{α_i} is the average distance of the anchors that are associated with the label box l_i through all the frames of sequence S.
- Create a label box interval $\lambda_i = [min_{\lambda_i}, max_{\lambda_i}]$ for each label box $l_i \in \mathcal{L}$ where the width of the label box interval $w_{\lambda_i} = \max(max_{\alpha_i} - min_{\alpha_i}, w_{l_i})$. The label box interval λ_i is associated with the anchor interval α_i, thus $d_{\lambda_i} = d_{\alpha_i}$(see Fig. 1(a)).

In order to determine a temporally coherent labeling of the given sequence S, we need to solve the following two subproblems.

1. *Label box interval to row assignment:* Determine the row r and left bound min_{λ_i} of the label box interval λ_i (then $max_{\lambda_i} = min_{\lambda_i} + w_{\lambda_i}$) for each label box $l_i \in \mathcal{L}$ so that the label box intervals fulfill the requirements 1-5 from Sec. 3. Please note that this subproblem is solved only once for the given sequence S. See Fig. 1(a) for an example of a label box to row assignment.
2. *Within row label box placement:* Determine the offset o_i between the x-coordinate of anchor x_{a_i} and the x-origin min_{l_i} of the label box l_i (origin refers to lower left corner) for any given frame $f \in S$. This reflects the requirements 2 and 5-6 from Sec. 3. Please note that this subproblem is solved for each frame of the given sequence S. See Fig. 1(a) for an example of label box placement in each row.

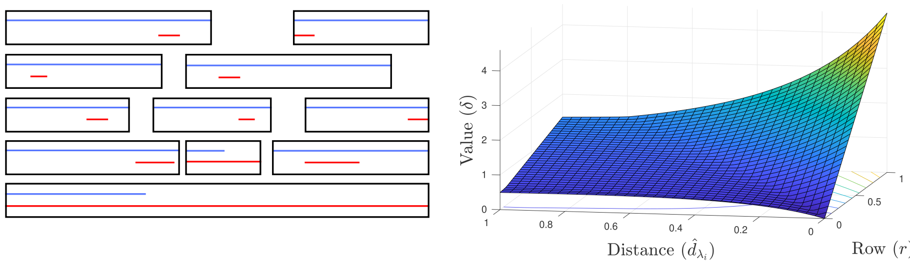
4.1 Label Box Interval to Row Assignment

We formulate the problem as a mixed-integer linear programming (MILP), which combines combinatorial optimization over binary variables with linear optimization over continuous variables [4].

The instance of MILP is formulated as the minimization of the objective function F_1 with respect to decision variables $I^r_{\lambda_i}$ and min_{λ_i} (the latter is considered in constraints). The objective function is defined as

$$F_1 = \sum_{i=1}^{|\mathcal{L}|} \sum_{r=1}^{R} I^r_{\lambda_i} \hat{r} + I^r_{\lambda_i} \delta(\hat{d}_{\lambda_i}, \hat{r}), \qquad (1)$$

where $I^r_{\lambda_i} \in \{0, 1\}$ is a binary variable that indicates if the label box interval λ_i is placed in row r and enforces the requirement 1. We consider at the most $R = |\mathcal{L}|$ rows. The hat modifier in the above given variable (e.g., $\hat{d}_{\lambda_i}$) denotes the unity-based normalized value of that variable. The product in the first term



(a) (b)

Figure 1. (a) Graphical visualization of the label box intervals λ_i (boxes) in an example of video sequence. The red solid line inside the box shows the corresponding anchors interval α_i, the blue solid line represents the label width w_{l_i}. (b) Function δ with $c_1 = 0.1$, $c_2 = 0.8$ and $c_3 = 0.5$.

of F_1 supports the requirement 4. The function $\delta(\hat{d}_{\lambda_i}\hat{r})$ in the second term of F_1 is defined as

$$\delta\big(\hat{d}_{\lambda_i}, \hat{r}\big) = \frac{|\hat{r} - c_1\hat{d}_{\lambda_i}| + |c_2\hat{r} - \hat{d}_{\lambda_i}| + c_3|\hat{r} - \hat{d}_{\lambda_i}|}{(\hat{d}_{\lambda_i} + c_2)^2} \tag{2}$$

and supports the requirement 3. The purpose of the δ function is to a establish relation between normalized distance $\hat{d}_{\lambda_i}$ and the row r where the label box interval λ_i and the corresponding label box interval α_i is placed. The constants c_1, c_2 and c_3 were selected experimentally with the requirement 3 in mind. We have achieved the best results with $c_1 = 0.1$, $c_2 = 0.8$ and $c_3 = 0.5$, see Fig. 1(b).

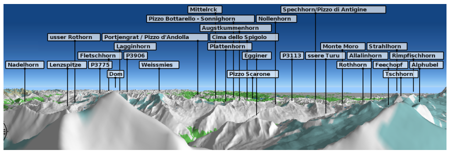
In order to fulfill the requirement 2, we define the following four objective constraints. First, we define the constraint for an overlap restriction as

$$min_{\lambda_i} + w_{\lambda_i} \le min_{\lambda_j} + M \cdot (1 - I^r_{\lambda_i}) + M \cdot (1 - I^r_{\lambda_j}), \tag{3}$$

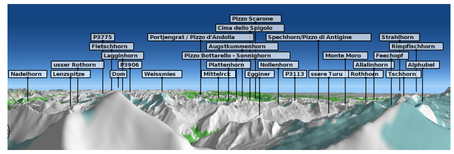
where we define the order so that $min_{\alpha_i} \le min_{\alpha_j} \land l_i \ne l_j$ and $l_i, l_j \in \mathcal{L}$. This constraint only needs to be applied in the case that both label box intervals are in the same row r which is indicated by the binary decision variables $I^r_{\lambda_i}$ and $I^r_{\lambda_j}$. The use of a binary variable to activate and deactivate the constraint is a well-known trick in MILP [6, 10]. The constant M needs to be sufficiently large in order to deactivate the constraint (i.e., the constraint is always true for any combination of λ_i and λ_j that are not in the same row). We set M equal to the frame width w_S, which works well in our experiments.

From the definition of the label box interval λ_i and from the requirement 5 it follows that the interval must completely overlap its associated anchor interval α_i. Therefore, we introduce constraints to enforce that α_i is the sub-interval of λ_i as $min_{\lambda_i} \le min_{\alpha_i}$ and $min_{\lambda_i} + w_{\lambda_i} \ge max_{\alpha_i}$.

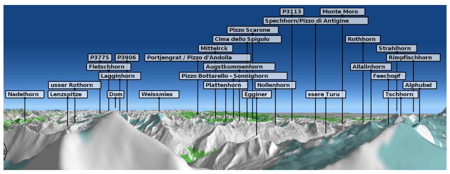
Finally, only one variable $I^r_{\lambda_i}$ for the label box interval λ_i is allowed to be 1. This reflects that λ_i is allowed to occupy only one row. We define this restriction as $\sum_{r=1}^{R} I^r_{\lambda_i} = 1$.



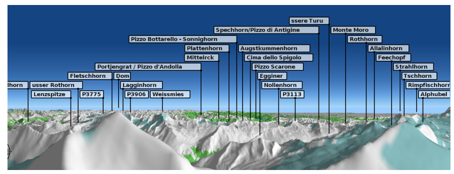
(a) Our temporal method



(b) Interval slots



(c) Growing border



(d) Gemsa layout

Figure 2. Four different panorama label layouts calculated for the mountain tops.

4.2 Within Row Label Box Placement

We formulate the subproblem as a convex quadratic programming (QP). When each label box interval is assigned to a row and its left bound min_{λ_i} is set, it remains for us to determine the positions of each label box for a given frame f so that the requirements 2 and 5-6 are fulfilled. The instance of QP is formulated as the minimization of the objective function F_2 with respect to the offset decision variable o_i. The objective function for the given frame f is defined as

$$F_2 = \sum_{a_i \in A} \left(o_i - \frac{w_{l_i}}{2} \right)^2 . \tag{4}$$

The function F_2 enforces requirement 6 only. In order to enforce requirements 2 and 5, we need to define objective constraints.

To enforce requirement 2, we define a constraint for each pair of label boxes l_i and l_j associated with anchors a_i and a_j in the given frame f as $x_{a_i} - o_i + w_{l_i} \leq x_{a_j} - o_j$, where we suppose an order so that $x_{a_i} < x_{a_j} \wedge l_i \neq l_j$. Furthermore, in order to fulfill requirement 5 we define the constraints $o_i \geq 0$ and $o_i \leq w_{l_i}$. Finally, we want to restrict a label box overflow with vertical bounds defined by width w_S of the given frame. This is accomplished by a pair of constraints $x_{a_i} - o_i \geq 0$ and $x_{a_i} - o_i + w_{l_i} \leq w_S$.

5 Results

We used GUROBI 8.0 with a MATLAB interface as optimization solver. The running time needed to solve the subproblem 1 (*label box interval to row assignment*) 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 measurement was performed for one video sequence on Intel® Core i5-3570 @ 3.40GHz with 24GB

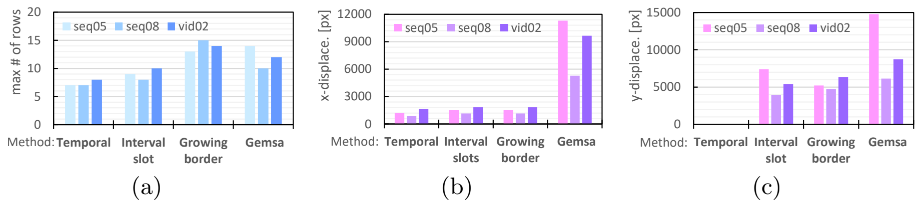


Figure 3. The maximum number of rows per video sequence determines the vertical compactness of the labeling (a). Total label displacement per video sequence in x-axis (b) and y-axis (c). The y-displacement for the *temporal* method is zero.

RAM (for more details please see the supplementary material[3]). However, since even the MILP is NP-hard, the computational time for MILP can vary from sequence to sequence and can grow significantly an increasing number of binary variables (i.e., with the number of labels) [4, 9]. The solution is accepted as optimal when optimality gap (relative distance between the best known solution and bound) is less than 0.01 %. The solver applies several primal heuristics and a branch-and-cut algorithm with different types of cutting planes (e.g., Gomory, MIR, StrongCG) to solve the MILP problem [12].

The optimization in the subproblem 2 (*within row label box placement*) is defined as convex QP, thus it can be solved in polynomial time [23]. Furthermore, the label box placement can be solved independently for each row, hence the optimization is prompt and highly parallelizable.

We have implemented three existing methods (*gemsa* [10], *growing border* [15], and *interval slots* [15] methods) to compare them with the proposed method. The label layouts produced with these methods are in Fig. 2(b)-2(d).

We have calculated the maximum number of rows in the layout for all implemented methods for three video sequences with a minimum length of 100 frames. The results show that the *temporal method* achieves the best results (see Fig. 3(a)). For the temporally coherent movement of the label boxes, it is crucial that the label boxes do not jump abruptly. Therefore, we have calculated the *displacement metric* for all implemented methods (see Fig. 3(b) and 3(c)). We calculate the displacement as the difference in the position of the given label between two subsequent frames. We calculate the displacement in x and y direction separately. The results suggest that labels in our proposed methods are more temporally coherent than in the other reviewed methods. The greatest discrepancy is visible for the y-displacement, where the labels placed using our methods are fixed in a single row.

6 User Experiments

We have conducted an evaluation with users to assess if our *temporal method* (1) improves the ability of the user to follow the label boxes in time, (2) how it

[3] Supplementary material: `http://cphoto.fit.vutbr.cz/panorama-labeling/`

influences the ability of the label layout to mediate the interconnection between the labels and the labeled objects, and (3) if the users prefer such a label layout to the other layouts.

For the evaluation, we have created a web application which the participants accessed through a web browser. First, each participant was instructed about the testing procedure; then, the participant provided their age and gender. The evaluation was divided into two experiments.

6.1 Experiment 1 – Accuracy

In the first experiment, we assessed the impact of the label layout on the users accuracy in the object-label and label-object assignment tasks. The experiment was one factor with four levels. The independent variable was the labeling method. In the evaluation, we used four methods to calculate the label layouts: *gemsa*, *growing border*, *interval slots*, and our *temporal* methods. We calculated the label layouts for three video sequences.

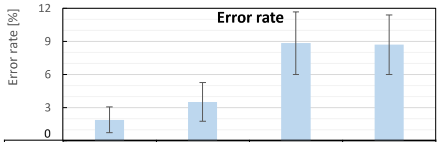
The experiment was designed as a between-subject. In other words, one participant was tested with only one labeling method to eliminate the learning effect and fatigue. For each participant, the order of video sequences was counterbalanced with a 3x3 balanced Latin square [16, Section 5.11] to eliminate the carry-over effect.

The first experiment consisted of a sequence of three tasks defined as follows. Task 1: Find the label associated to a highlighted anchor. Task 2: Find the anchor associated to a highlighted label. Task 3: Follow a certain moving label for 2 seconds and then select the label. For a detailed description of the tasks, please see the supplementary material of this paper.

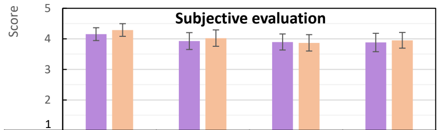
Each participant repeated each task 10 times for each video sequence. We measured the *error rate* (the number of wrongly selected labels/anchors relative to all selected labels/anchors). After each video sequence we have conducted a subjective evaluation of the easiness of the visual search (task 1-3), the confidence (task 1-2) and the need to focus (task 3).The participants provided their subjective evaluation on Likert scales from 1 to 5.

Task 1 and its subjective evaluation was completed by 60 participants (12 females) with the age ranging from 19 to 54 years ($\bar{x} = 25.31$; $\sigma = 6.49$). Task 2 and its subjective evaluation was completed by 49 participants (11 females) with the age ranging from 19 to 54 years ($\bar{x} = 25.86$; $\sigma = 7.04$). Finally, task 3 and its subjective evaluation was completed by 44 participants (10 females) with the age ranging from 19 to 54 years ($\bar{x} = 26.32$; $\sigma = 7.29$).

We evaluated the collected data for all video sequences together. We performed a statistical evaluation of the measured data using confidence intervals. We transformed the measured number of errors onto error rates with the LaPlace method [14] and calculated the confidence intervals of the error rates as adjusted Wald intervals, a method recommended for completion rates [1, 18]. We calculated the confidence intervals for Likert scales as confidence intervals for rating scales [19, Chapter 3]. We use 95% confidence intervals for error rates, comple-

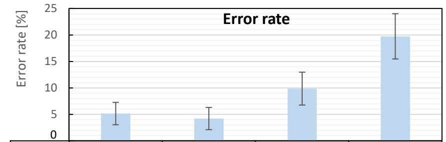


Method	Temporal	Interval slots	Growing border	Gemsa
Error rate	1.89 (0.73, 3.06)	3.52 (1.77, 5.27)	8.83 (6.0, 11.64)	8.71 (6.03 ,11.4)

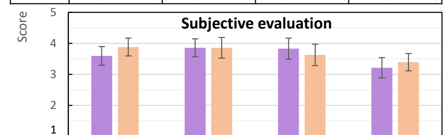


Method	Temporal	Interval slots	Growing border	Gemsa
Easiness	4.15 (3.95, 4.36)	3.93 (3.65, 4.20)	3.90 (3.63, 4.16)	3.88 (3.58, 4.18)
Confid.	4.29 (4.08, 4.50)	4.02 (3.76, 4.29)	3.87 (3.60, 4.14)	3.95 (3.70, 4.21)

(a) Task 1



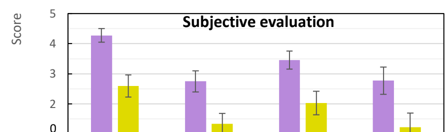
Method	Temporal	Interval slots	Growing border	Gemsa
Error rate	5.17 (3.06, 7.28)	4.22 (2.12, 6.31)	9.87 (6.76, 12.98)	19.75 (15.48, 24)



Method	Temporal	Interval slots	Growing border	Gemsa
Easiness	3.60 (3.30, 3.90)	3.86 (3.57, 4.15)	3.83 (3.49, 4.17)	3.21 (2.88, 3.54)
Confid.	3.88 (3.60, 4.17)	3.86 (3.52, 4.19)	3.63 (3.28, 3.97)	3.39 (3.11, 3.67)

(b) Task 2

Method	Temporal	Interval slots	Growing border	Gemsa
Error rate	0.71 (0.00, 1.51)	6.77 (4.03, 9.51)	9.56 (6.41, 12.72)	16.05 (11.24, 20.86)



Method	Temporal	Interval slots	Growing border	Gemsa
Easiness	4.28 (4.05, 4.50)	2.75 (2.40, 3.11)	3.46 (3.16, 3.75)	2.77 (2.32, 3.23)
Focus	2.60 (2.23, 2.97)	1.34 (1.01, 1.68)	2.03 (1.64, 2.42)	1.23 (0.76, 1.70)

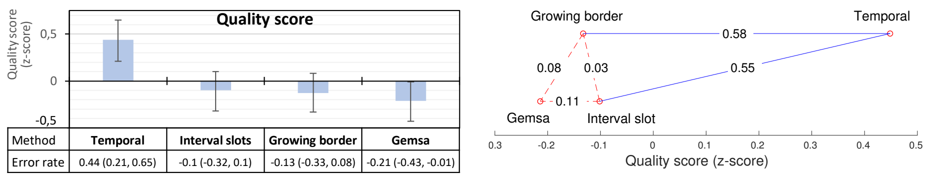
(c) Task 3

Figure 4. Results of the Experiment 1 – Accuracy: Error rate and subjective evaluation for the task 1 (a), task 2 (b), and task 3 (c).

tion times and Likert scales. When the confidence intervals are disjointed, we can report that the means of the measured data are significantly different.

For tasks 1 and 2, the average error rates and average score from subjective evaluation together with their 95% confidence intervals are shown in Fig. 4(a) and 4(b). For task 3, the average error rates, and average scores from the subjective evaluation, together with their 95% confidence intervals, are shown in Fig. 4(c).

For task 1, the results show that the *temporal* and *interval slots* methods achieve a significantly lower error rate than the *growing border* and *gemsa* methods. For task 2, the results show that the *temporal* and *interval slots* methods achieve a significantly lower error than the *gemsa* method. Furthermore, the *interval slots* method achieves a significantly lower error than the *growing border* method. The participant stated that the task 2 was significantly easier with the *interval slots* method than with the *gemsa* method. For task 3, the results show that the *temporal* method achieved significantly lower error rate than the other methods. In the subjective evaluation, the participants reported that the task 3 was significantly easier to complete with the *temporal* method than with the other methods. Furthermore, the participants reported that they had to focus significantly less with the *temporal* method than with the *interval slot* and



(a) (b)

Figure 5. (a) Overall quality scores. (b) Statistical significance and quality scores.

gemsa methods. We have not detected any other significant differences between the methods.

In general, the results show that due to temporally coherent movement of the label boxes, our *temporal* method allows us to follow label boxes moving in time significantly accurately than the other methods. At the same time, our *temporal* method mediates the interconnection between the labels and the labeled objects, the same as or better than the other methods.

6.2 Experiment 2 – Preference

To assess the users preferences among different label layouts, we have conducted the subjective experiment using a psychophysical technique of paired comparisons [8, 21]. We have used specifically the two-interval forced choice (2IFC) experiment paradigm to verify the overall quality of labeling methods (*gemsa*, *growing border*, *interval slot*, *temporal*) where the number of methods is denoted as $m = 4$. We chose set of $s = 3$ video sequences. For single given video sequence each participant had to compare $\binom{m}{2} = 6$ pairs – all possible combinations of m methods. A total of 40 participants (10 females) with the age ranging from 19 to 54 years ($\bar{x} = 26.61$; $\sigma = 7.46$) completed in total 240 parwise-comparisons. For each participant, the order of the pairs of methods to compare was counterbalanced with a 6x6 balanced Latin square [16, Section 5.11] to eliminate learning and carry-over effects.

The data were stored in count matrix $\mathbf{C}$ with a $m \times m$ shape for each participant. The element c_{ij} represents the number of times that method i was selected better than method j. We converted the participant matrices $\mathbf{C}$ into interval quality score (z-score) scale and computed a statistical significance using customized MATLAB framework [17].

In order to convert the count matrix $\mathbf{C}$ to the interval quality score scale the Thurstone's Law of Comparative Judgment model is used with respect to Case V [17, 21]. In order to reject the null hypothesis H_0, where the difference in perceived quality scores is zero, we applied the Two-tailed test at a significance level $\alpha = 0.05$.

The overall quality score is depicted in Fig. 5(a). The results for panorama video sequence labeling, using our proposed *temporal* method, exhibit the best quality score, followed by *interval slot* and *growing border*. The worst perceived method is considered the *gemsa* method.

The statistical significance for surveyed methods is presented in Fig. 5(b). The results show that difference between our *temporal* method and the rest of the surveyed methods is statistically significant. Thus, we can reject the null hypothesis H_0 in the *temporal*-other pairs. However, H_0 can not be rejected for the other-other pairs. This means that we have not detected significant difference in perceived quality among the *interval slot*, *growing border* and *gemsa* methods.

7 Conclusions

We proposed a novel method for video sequence boundary labeling using optimization. We compared the method with three other methods in an extensive user study. The results of the study show that with our method, the users are able to follow moving label boxes significantly more accurately than with the concurrent methods. At the same time, our method mediates the interconnection between the labels and the labeled objects the same as or better than the other methods. The proposed method was ranked the best for the boundary labeling of panorama video sequences by participants of the study. In other words, the proposed method should be preferred for the labeling of the panorama video sequences to the other methods.

Acknowledgements. This work was supported by Research Center for Informatics No. CZ.02.1.01/0.0/0.0/16_019/0000765; by V3C – "Visual Computing Competence Center" by Technology Agency of the Czech Republic, project no. TE01020415; by the Ministry of Education, Youth and Sports of the Czech Republic within the activity MOBILITY (MSMT-539/2017-1) ID: 7AMB17AT021, and from the "National Programme of Sustainability (NPU II) project IT4Innovations excellence in science - LQ1602"; and by the IT4Innovations infrastructure which is supported from the Large Infrastructures for Research, Experimental Development and Innovations project "IT4Innovations National Supercomputing Center - LM2015070". Access to computing and storage facilities owned by parties and projects contributing to the National Grid Infrastructure MetaCentrum provided under the programme "Projects of Large Research, Development, and Innovations Infrastructures" (CESNET LM2015042).

References

1. Agresti, A., Coull, B.A.: Approximate is better than 'exact' for interval estimation of binomial proportions. The American Statistician **52**(2), 119–126 (1998)
2. Baboud, L., Čadík, M., Eisemann, E., Seidel, H.P.: Automatic photo-to-terrain alignment for the annotation of mountain pictures. In: CVPR '11. pp. 41–48. IEEE Computer Society, Washington, DC, USA (2011)
3. Bekos, M.A., Kaufmann, M., Potika, K., Symvonis, A.: Multi-stack boundary labeling problems. WSEAS Transactions on Computers **5**(11), 2602–2607 (2006)

4. Belotti, P., Kirches, C., Leyffer, S., Linderoth, J., Luedtke, J., Mahajan, A.: Mixed Integer Nonlinear Programming. Cambridge University Press (2012)
5. Benkert, M., Haverkort, H., Kroll, M., Nöllenburg, M.: Algorithms for multi-criteria one-sided boundary labeling. In: Graph Drawing. pp. 243–254. Springer (2008)
6. Chen, D.S., Batson, R.G., Dang, Y.: Applied Integer Programming: Modeling and Solution. John Wiley & Sons (2011)
7. Čmolík, L., Bittner, J.: Layout-aware optimization for interactive labeling of 3d models. Computers & Graphics **34**(4), 378–387 (2010)
8. David, H.: The method of paired comparisons. Griffin's statistical monographs & courses, C. Griffin (1988)
9. Garey, M.R., Johnson, D.S.: Computers and Intractability; A Guide to the Theory of NP-Completeness. W. H. Freeman & Co., New York, NY, USA (1990)
10. Gemsa, A., Haunertand, J.H., Nöllenburg, M.: Multi-row boundary-labeling algorithms for panorama images. ACM TSAS **1**(1), 289–298 (2014)
11. Götzelmann, T., Hartmann, K., Strothotte, T.: Annotation of animated 3d objects. In: SimVis, SCS. pp. 209–222. Publishing House (2007)
12. Gurobi Optimization, LLC: Advanced Gurobi Algorithms (2016), `http://www.gurobi.com/pdfs/user-events/2016-frankfurt/Die-Algorithmen.pdf`
13. Kouřil, D., Čmolík, L., Kozlíková, B., Wu, H., Johnson, G., Goodsell, D.S., Olson, A., Gröller, M.E., Viola, I.: Labels on levels: Labeling of multi-scale multi-instance and crowded 3d biological environments. IEEE TVCG **25**(1), 977–986 (2019)
14. Lewis, J.R., Sauro, J.: When 100% really isn't 100%: improving the accuracy of small-sample estimates of completion rates. Journal of Usability studies **1**(3), 136–150 (2006)
15. Maass, S., Döllner, J.: Efficient view management for dynamic annotation placement in virtual landscapes. In: Smart Graphics. pp. 1–12. Springer (2006)
16. MacKenzie, I.S.: Human-computer interaction: An empirical research perspective. Newnes (2012)
17. Perez-Ortiz, M., Mantiuk, R.K.: A practical guide and software for analysing pairwise comparison experiments (2017), `http://arxiv.org/abs/1712.03686`
18. Sauro, J., Lewis, J.R.: Estimating completion rates from small samples using binomial confidence intervals: comparisons and recommendations. In: Proceedings of the human factors and ergonomics society annual meeting. vol. 49, no. 24, pp. 2100–2103. SAGE Publications (2005)
19. Sauro, J., Lewis, J.R.: Quantifying the user experience: Practical statistics for user research. Elsevier (2012)
20. Tatzgern, M., Kalkofen, D., Grasset, R., Schmalstieg, D.: Hedgehog labeling: View management techniques for external labels in 3d space. In: 2014 IEEE Virtual Reality. pp. 27–32 (2014)
21. Tsukida, K., Gupta, M.R.: How to Analyze Paired Comparison Data. UWEE Technical Report 206 (2011)
22. Vaaraniemi, M., Treib, M., Westermann, R.: Temporally coherent real-time labeling of dynamic scenes. In: Proceedings of the 3rd International Conference on Computing for Geospatial Research and Applications. pp. 17:1–17:10. COM.Geo '12, ACM, New York, NY, USA (2012)
23. Ye, Y., Tse, E.: An extension of karmarkar's projective algorithm for convex quadratic programming. Mathematical Programming **44**(1), 157–179 (1989)