Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Temporally Stable Boundary Labeling for Interactive and Non-Interactive Dynamic Scenes

Petr Bobák[a,*], Ladislav Čmolík[b], Martin Čadík[a,b]

[a] *Brno University of Technology, Faculty of Information Technology, Bozetechova 2/1, 612 00 Brno, Czech Republic*
[b] *Czech Technical University in Prague, Faculty of Electrical Engineering, Technicka 2, 166 27 Praha 6, Czech Republic*

ARTICLE INFO

Article history:
Received 14 February 2020
Received in final form 4 July 2020
Accepted 8 August 2020
Available online 14 August 2020

Keywords: boundary labeling, label layout, temporal coherence, stable layout, interaction, dynamic scene, visualization

ABSTRACT

We propose two novel temporally stable screen-space labeling methods for dynamic scenes. The first one is suitable for offline processing of the entire interaction or the video in advance. The second method is designed for interactive applications. The main idea of our proposed methods is to minimize the vertical and horizontal movement of the labels during the interaction with the scene (*e.g.*, zooming or translating the camera). According to the results of quantitative evaluation, our labeling is more stable during the interaction than labeling produced by the current state of the art. Moreover, participants of a comprehensive user study declared that the labeling produced by the proposed methods allows them to follow moving labels significantly more accurately, and it is significantly more pleasing than with previously published methods. Furthermore, the proposed methods can be extended by the prominence of the features and easily parameterized to fit different requirements to the label layout.

1. Introduction

Short textual annotations (so-called *labels*), more or less distant from the features of interest, are used to communicate the position of features within an object together with additional information (*e.g.*, the name of the feature). The right visual correspondence of the label with the annotated feature is crucial for functional and aesthetic label placement. All the labels of a single visualization form a *labeling* or *labeling layout*. In high-quality label layout, the labels should be unambiguous and well-readable, labels should not overlap with each other, one should be able to conclusively assign each annotated feature to a corresponding label and vice versa. The labeling should also be aesthetic, even though aesthetic aspects are often subjective.

Labeling has several variants according to the type of the feature (point, area, line) and positioning of corresponding labels in the label layout. Labels can be placed within the space of visualization tightly next to the features, so-called *internal labeling*. If the feature density is too high or if the background must not be cluttered, the labels can be placed outside the visualization, so-called *external labeling*. In order to visualize the mapping of labels with corresponding features, each label is connected with its feature using a line, also known as *leader*.

The term *external* labeling is accepted in current literature as an umbrella that covers several labeling techniques, such as *excentric* or *focus-region* labeling [1], *contour* labeling [2], *boundary* labeling [3] and its variant for panorama images [4, 5]. A comprehensive state-of-the-art survey by the founder of boundary labeling Bekos *et al.* [6] provides a first unified taxonomy for categorizing external labeling techniques. A survey by Oeltze-Jafra and Preim [7] provides an additional overview of labeling techniques in the medical domain.

Interactive applications of labeling algorithms introduce a new aspect of temporal coherence. Applying only static algorithms on a frame-by-frame basis leads to temporally unstable behavior. In such a case, labels often jump abruptly from

*Corresponding author.
e-mail: ibobak@fit.vutbr.cz (Petr Bobák), cmolikl@fel.cvut.cz (Ladislav Čmolík), cadik@fit.vutbr.cz (Martin Čadík)

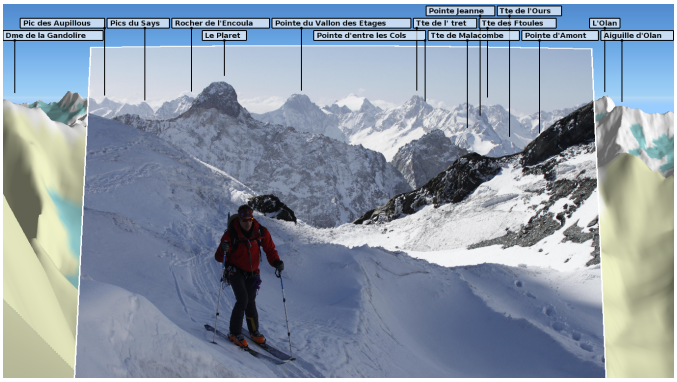


Figure 1. Labeling created by our method on a photo aligned with a terrain model using camera pose estimation techniques of Baboud *et al.* [8].

one position to another, breaking several assumptions of high-quality labeling.

In this paper, we specifically focus on the one-sided boundary labeling of dynamic scenes, where labels are placed on the top of the scene (the static case was introduced by Gemsa *et al.* [4] as *panorama* labeling). The features in the scene are approximated by points denoted as *anchors*. The visual relationship between labels and corresponding features is established by vertical leaders that connect the label with the anchors, see Fig. 1.

We propose two labeling methods suitable for a diverse range of applications. The first is designated for the offline processing of the entire interaction in advance. Such a method can be valuable for creating, *e.g.*, educational visualizations, television news infographics, or generally in the movie industry, where the complete interaction with the scene (*i.e.*, all the frames of the video-sequence) is known in advance. Imagine video-footage from a drone flying through mountain terrain or a city, where one would like to label peaks or tourist attractions, respectively. The second method is designed for the online processing of continuously delivered frames created on-demand as a result of interacting with a dynamic scene. Such a method can be applied in, *e.g.*, games, 3D map viewers, and augmented or virtual reality applications. Imagine an interactive application presenting a 3D map (digital elevation model), where one would like to know nearby points of interest and could move along the scene by interacting with the camera (*e.g.*, pan or rotate). For an overview of the proposed methods, see Fig. 3.

This work significantly expands on our previous paper [5] initially presented at *Computer Graphics International 2019* and presents the following contributions: (1) A temporally stable labeling method designed for the offline processing of the entire interaction with the scene in advance. (2) A novel temporally stable labeling method designed for interactive visualizations. (3) An extended labeling terminology of Bekos *et al.* [6] suitable for interactive and non-interactive labeling of dynamic scenes. (4) A formulation of visibility optimization based on feature prominence, and an extension for smooth label transitions. (5) A comparison of the proposed methods with three others, and the results of an extensive user study on several aspects of labeling.

2. Related Work

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

Fixed model. The input is a set of anchors and a set of labels placed on top of the scene. The task is to connect each label with the corresponding anchor with a leader. Bekos *et al.* [3] introduced a method for boundary labeling where a set of anchor points is connected with a set of predefined labels positioned in one or up to three rows with rectilinear leaders. The method finds the leaders whose combined length is minimal. Benkert *et al.* [9] later showed that better label layouts could be produced if we consider criteria such as the number of bends of the leaders and distance between the leaders, in addition to the length of the leaders.

Flexible model. The input is a set of anchors placed on top of the scene and a set of labels without positions. The task is to determine the positions of the labels and create a connection with the corresponding anchor. Maass and Döllner [10] presented two methods that produce boundary label layouts. Labels are processed according to the camera-feature distance and centered on the vertical leader. Gemsa *et al.* [4] presented a dynamic programming approach that, for a set of anchor points, places the labels on the lowest possible number of rows using dynamic programming. Each label is connected with a corresponding anchor point with a vertical leader, and no leader intersects with any label.

Temporal coherence. The methods addressing the temporally coherent movement of labels strive to determine a label layout, where the labels do not change the position abruptly and in a predictable manner.

Ali *et al.* [2] proposed an anchor point stabilization based on an additional attractive force that aims to keep anchor points close to their previous positions. Čmolík and Bittner [11], and later Balata *et al.* [12] proposed a similar technique based on additional coherence terms for features and label positions.

Preim et al. [13] proposed a method for temporally coherent boundary labeling, but they allow the leaders to intersect, which makes determining the correspondence between anchors and labels hard, especially for layouts with many anchors and labels.

Mühler and Preim [14] proposed a method for the labeling of 2D slices and 3D reconstructions of segmented medical structures for surgical planning. They propose to lock the once-calculated position of a label over multiple slices until the labeling is infeasible (*e.g.*, overlap of several labels). A similar approach was published by Mogalle *et al.* [15], who presented a more constrained label placement technique for the labeling of 2D slice data.

Götzelmann *et al.* [16] presented an approach focusing on the labeling of animated 3D objects such as combustion engines with moving pistons. The entire animation is analyzed to determine the calm and fluctuating regions. Afterward, the

labels are placed in calm regions such that they move as little as possible during the animation. The label placement of fluctuating parts is resolved by the visualization of the trajectory whose midpoint is connected to the corresponding label. Unfortunately, the approach considers the exact position and shape of illustrated objects and does not rely on any shape simplification with bounding objects.

Stein and Décoret [17] presented a greedy approach for the dynamic labeling of interactive scenes. To address the problem of temporal discontinuity, they incorporated the interpolation of the label position. Therefore, the velocity of the moving label is bounded, and it may take several frames to reach the computed position. However, this approach does not solve the problem; instead, the movement of the labels is smoothed to avoid visual discontinuity.

Tanzgern *et al*. [18] proposed an 3D object space labeling approach (so-called *hedgehog* labeling). In contrast to most previously published approaches, they define labels as elements of the 3D scene, which overcome the lack of temporal coherence. Therefore, the leader is part of the line defined by the center of the sphere enclosing the illustrated object and the feature of the corresponding label in 3D space. They suggest two methods to prevent label occlusions: center-based where labels move along a 3D pole sticking out from the annotated object, and plane-base where the labels are placed within a corresponding plane using a force-based approach proposed by Ali *et al*. [2].

Maass and Döllner [10] proposed a similar hysteresis approach to make the movement of labels temporally stable. During the interaction, the labels keep their current positions, and once the user pauses or finishes the interaction, the label layout is recalculated, and the labels perform a continuous movement to the computed position.

Unfortunately, none of the approaches are suitable for the temporally stable one-sided boundary labeling, where a small movement of a label can decrease the available space for another label, which in turn can lead to an abrupt change in the position of the label. Consequently, the change in position can again limit available space for another label.

3. The Definition of Dynamic Labeling

In this section, we describe the formal definition of boundary labeling for dynamic scenes. We extended the terminology and definitions proposed by Bekos *et al*. [6], that are suitable for *static* labeling, to capture specific aspects of *dynamic* labeling. As input we are given a sequence S of frames $F_0, F_1, \ldots, F_{n=|S|}$. A frame is a tuple $F = (D, \mathcal{A}, \mathcal{L})$ described by a *drawing region* D partitioned into the *image region* I and *labeling region* $L = D \setminus I$; see Fig. 2. The drawing region D has the same dimensions (D_w, D_h) for the entire sequence S. A set of *anchors* $\mathcal{A}$ denotes the points of interest to be labeled. Each anchor a is a point of the image region I with coordinates (a_x, a_y). Furthermore, each anchor has additional information attached (*e.g*., the name of the anchor). The axis-aligned bounding box of additional information is denoted as *label* ℓ of dimensions (ℓ_w, ℓ_h). Each label is a rectangular sub-region of L placed at coordinates (ℓ_x, ℓ_y); see Fig. 2. We denote a set of all instances (*e.g*., labels, anchors) in frame F with a superscript such as $\mathcal{L}^F$ (labels) and a set of all instance that occur at least once in any frame of sequence S such as $\mathcal{L}^S$. Each anchor is connected to its label ℓ by a *leader* λ at an attachment point called *port* π on the boundary of ℓ. The distance between ℓ_x and π is called *offset* ℓ_o (*i.e*., horizontal coordinate $\ell_x = a_x - \ell_o$). A *callout* is the collection $\gamma = (\lambda, \pi, \ell)$ of a leader λ connected to a label ℓ at the point π. A set C of callouts is called *labeling* or *label layout*. A labeling of frame F and sequence S is denoted as C^F and C^S, respectively. We call a labeling C of a sequence S *valid* if it satisfies the following requirements.

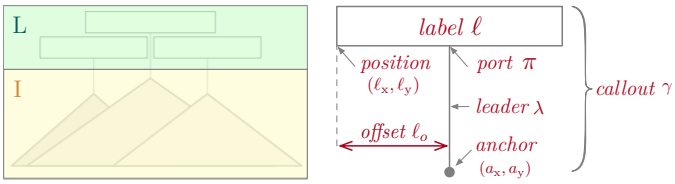


Figure 2. Illustration of terminology based on the report of Bekos *et al*. [6].

(R1) The labels do not overlap with each other [2, 10, 14, 19, 20].

(R2) The labels are connected with the corresponding anchors with vertical leaders [4, 10, 14, 19].

The *labeling quality* among a set $\mathcal{S}$ of all valid labelings can be described by a cost function $c : \mathcal{S} \to \mathbb{R}^+$. The *optimal labeling* $C^0 \in \mathcal{S}$ satisfies the condition of optimality $c(C^0) < c(C) \forall C \in \mathcal{S}$ and respects all the following criteria C.

(C1) The number of stacked layers of labels in the label layout is minimized [2]. To put it differently, the labels are placed as close as possible to the corresponding anchor [10, 14, 19].

(C2) The leader is connected to the label as close to the center of the label as possible to provide clear mapping [10], and to achieve aesthetic and symmetric layout [4, 19].

(C3) The movement of the labels through the interaction is temporally coherent [2, 14]. In other words, the vertical and horizontal movement of the labels should be continuous without abrupt changes [20], and minimized through the interaction with the scene [19].

(C4) [Optional] The vertical positions of labels should correspond to the distances of labeled anchors in a scene from the camera center. The labels of the closest anchor should be the lowest in the label layout [10].

Please note, that high-quality human-friendly labeling, in general, is hard to formalize as it is relative to various subjective and domain aspects. Therefore, high-quality boundary labeling is often a compromise among the described criteria. Because of the previous statement, we provide a quantitative evaluation in Sec. 9 and an extensive user study in Sec. 10–11.

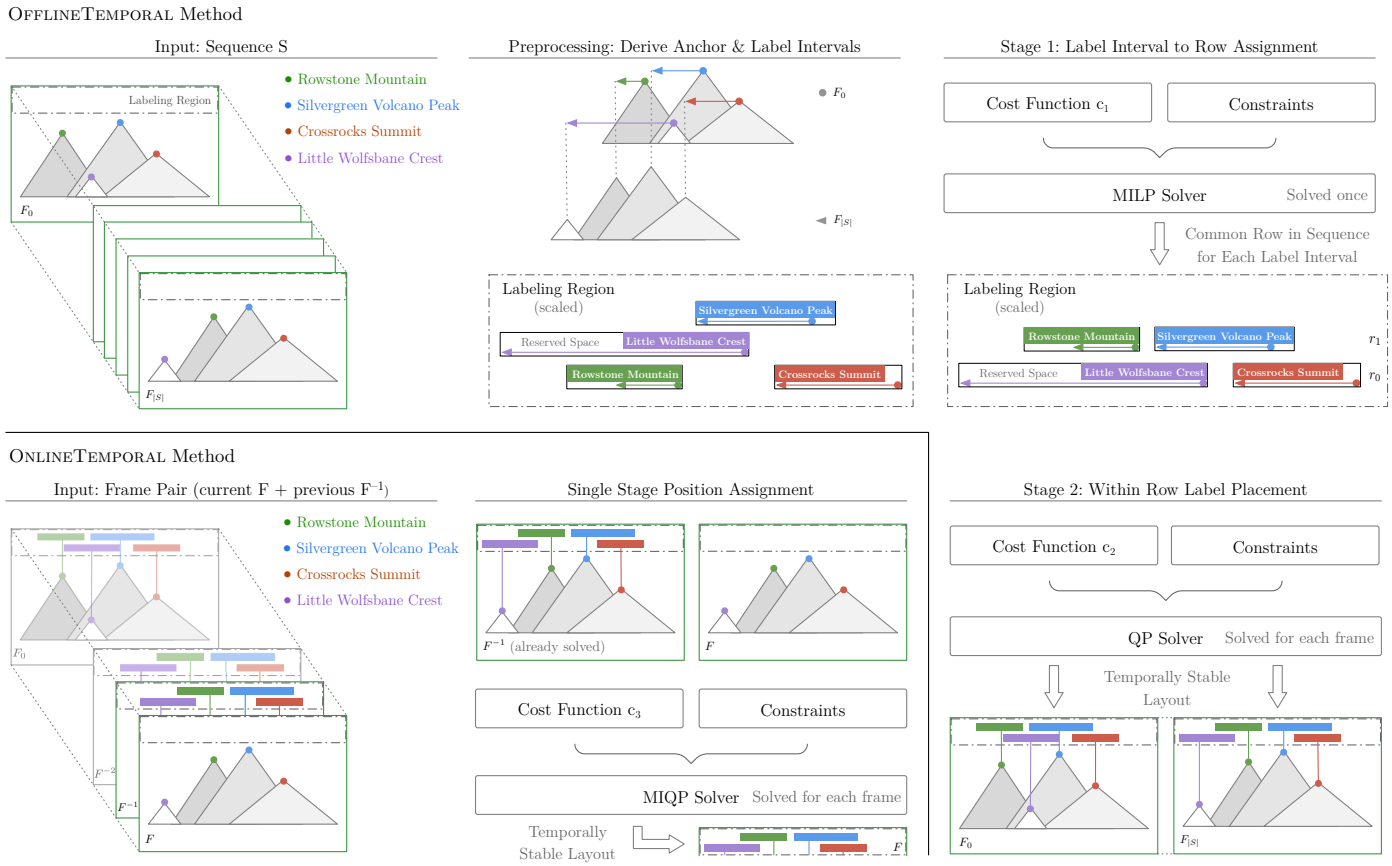


Figure 3. High-level overview of processing stages applied in the proposed methods. The OfflineTemporal method consists of the *preprocessing* stage followed by the *Label Interval to Row Assignment* and *Within Row Label Placement* stage. The OnlineTemporal is designed as a single-stage method.

4. Offline Labeling Method

When the entire interaction with the scene is known in advance, we process the sequence S as a whole. We place labels into discrete rows such that the vertical position ℓ_y is approximated by *row* $r \in R$, where $R = \{\, r \mid r \in \mathbb{N} \;\wedge\; r \leq |\mathcal{A}^S| \,\}$ is a set of available rows. Furthermore, we suppose the width ℓ_w and the height ℓ_h is fixed for all $F \in S$. The proposed temporally coherent method denoted as OfflineTemporal consists of the following three stages (for a high-level overview, see Fig. 3).

Preprocessing. Given a set of anchors $\mathcal{A}^S$, we create an *anchor interval* $\alpha = [\alpha_{\min}, \alpha_{\max}]$ for each anchor $a \in \mathcal{A}^S$. The anchor interval α captures the horizontal movement of anchor a through all the frames $F \in S$. The minimum x-coordinate $\alpha_{\min} = \min(a_x)$ and the maximum $\alpha_{\max} = \max(a_x)$ of an anchor define the bounds of the corresponding anchor interval α (see Fig. 4). We denote the set of anchor intervals derived from the sequence S by A^S.

Given a set $\mathcal{L}^S$ of labels, we then define a *label interval* $\lambda = [\lambda_{\min}, \lambda_{\max}]$ for each label $\ell \in \mathcal{L}^S$. We construct the label interval λ to reserve space for the horizontal movement of the label. This way, we can fixate the vertical movement of a label and allow movement only in the horizontal direction. Therefore, one label cannot influence the movement of any other label, which fulfills criterion C3. We derive the width of a label interval λ from the associated anchor interval a and the label ℓ as $\lambda_w = \max(\alpha_{\max} - \alpha_{\min}, \ell_w)$. We denote the set of label intervals derived from the sequence S by Λ^S.

Furthermore, we calculate an average camera-to-anchor distance α_d for each anchor $a \in \mathcal{A}^S$ to be able to satisfy criterion C4. Because the label interval λ is associated with the anchor interval α, the camera-to-anchor distance $\lambda_d = \alpha_d$.

Label Interval to Row Assignment. In this stage, we determine the vertical position $\ell_y = r$ and the left bound $\lambda_{\min}$ of the label interval λ. The right bound of the label interval λ can then be derived as $\lambda_{\max} = \lambda_{\min} + \lambda_w$. Since the label moves inside the reserved space given by its label interval λ, the label intervals must not overlap with each other (requirement R1). Similarly, for the same reason, the anchor interval α must be a sub-interval of label interval λ; otherwise, the requirement R2 could be violated (label ℓ could not be connected with its anchor a). Furthermore, as we determine the row r in which the label will be fixed throughout the sequence S, we also take the criteria C1 and C4 into account in this stage.

Please note that this stage is solved only once for the given sequence S. For an example of a label-to-row assignment, see Fig. 4.

Within Row Label Placement. In this stage, we derive the horizontal position ℓ_x of label ℓ fixed within row r by optimizing the offset ℓ_o from the port π for each frame $F \in S$. Therefore,

we take into account the criterion C2. However, in order to better fulfill criterion C2 at the bounds of λ, we enable the label to pop out of its associated label interval λ. Consequently, we consider also requirements R1 and R2.

4.1. Label Interval to Row Assignment

We formulate the problem as mixed-integer linear programming (MILP), which combines combinatorial optimization over binary variables with linear optimization over continuous variables [21].

The instance of MILP is formulated as the minimization of the cost function c_1 with respect to decision variable $I^r_\lambda \in \{0, 1\}$ that indicates whether the label interval λ is placed in row r, and with respect to continuous variable λ_{min} considered further in the definition of constraints. The cost function is defined as

$$c_1\left(\Lambda^S\right) = \sum_{\lambda \in \Lambda^S} \sum_{r \in R} I^r_\lambda \hat{r} + I^r_\lambda \delta(\lambda_{\hat{\text{d}}}, \hat{r}). \tag{1}$$

The hat modifier in the above-given variable (*e.g.*, $\hat{r}$) denotes the unity-based normalized value of that variable. The product in the first term of c_1 supports the criterion C1. The function $\delta(\hat{d}, \hat{r})$ in the second term of c_1 is defined as

$$\delta\left(\hat{d}, \hat{r}\right) = \frac{\left|\hat{r} - w_1\hat{d}\right| + \left|w_2\hat{r} - \hat{d}\right| + w_3\left|\hat{r} - \hat{d}\right|}{\left(\hat{d} + w_2\right)^2} \tag{2}$$

and supports the criterion C4. The purpose of the δ function is to establish a relation between normalized distance $\hat{d}$ and the row r of the label interval λ. By observing the influence of various values of the weights w_1, w_2 and w_3 on the resulting layouts, we recommend using the weights $w_1 = 0.1$, $w_2 = 0.8$, and $w_3 = 0.5$, see Fig. 5. Please note that we have selected these values with the criterion C4 in mind.

To fulfill the requirements R1 and R2, we define the following constraints. First, we define the constraint to satisfy requirement R1 as

$$\lambda^{(i)}_{\text{min}} + \lambda^{(i)}_{\text{w}} \leq \lambda^{(j)}_{\text{min}} + M \cdot \left(1 - I^r_{\lambda^{(i)}}\right) + M \cdot \left(1 - I^r_{\lambda^{(j)}}\right), \tag{3}$$

where we define the order so that the associated anchor interval $\alpha^{(i)}_{\text{min}} \leq \alpha^{(j)}_{\text{min}} \wedge \alpha^{(i)} \neq \alpha^{(j)}$ and $\alpha^{(i)}, \alpha^{(j)} \in A^S$. This constraint only needs to be applied in the case that both label 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 known trick in MILP [22, 4]. The constant M needs to be sufficiently large to deactivate the constraint (*i.e.*, the constraint is always true for any combination of $\lambda^{(i)}$ and $\lambda^{(j)}$ that are not in the same row). We set M equal to the width of drawing region D_w.

From the definition of the label interval λ and from the requirement R2 it follows that the interval must completely overlap its associated anchor interval α. Therefore, we introduce constraints to enforce that α is the subinterval of λ as

$$\lambda_{\text{min}} \leq \alpha_{\text{min}}, \tag{4a}$$

$$\lambda_{\text{min}} + \lambda_{\text{w}} \geq \alpha_{\text{max}}. \tag{4b}$$

Finally, the label interval λ is allowed to occupy only a single row r. Therefore, we define this restriction as

$$\sum_{r \in R} I^r_\lambda = 1. \tag{5}$$

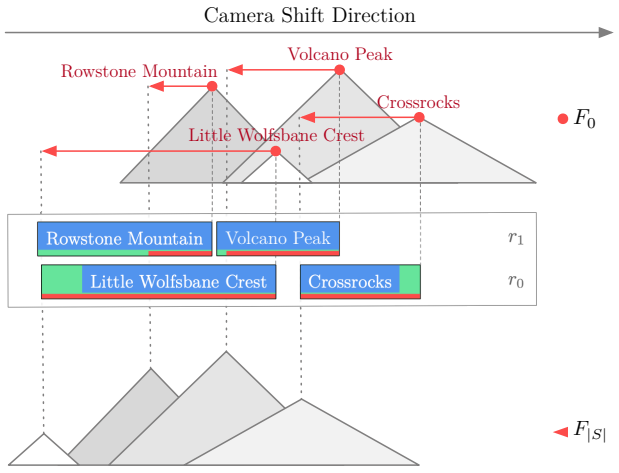


Figure 4. Example of sequence preprocessing in the OfflineTemporal method. The movement of the anchor in the x-axis starts at the position denoted by a red circle in frame F_0 and ends at the position denoted by a red triangle in frame $F_{|S|}$. The label interval λ (a green rectangle with a black stroke) reserves the space for the horizontal movement of its label. The length of the label interval λ is derived from the label width ℓ_w (blue rectangle) and the length of anchor interval α (red line). The left bound of label interval λ_{min} and its vertical position (row r) is optimized in *Label Interval to Row Assignment* stage.

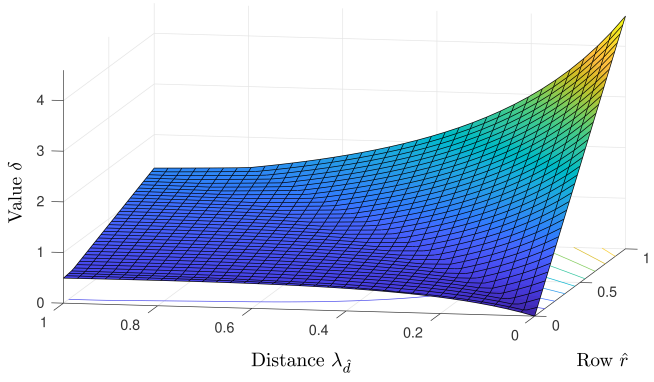


Figure 5. Function δ with parameters $w_1 = 0.1$, $w_2 = 0.8$ and $w_3 = 0.5$.

4.2. Within Row Label Placement

We formulate the problem as convex quadratic programming (QP). When each label interval is assigned to a row and its left bound λ_{min} is set, it remains to determine the vertical position of each label for a given frame F so that the criterion C2 is reflected. The instance of QP is formulated as the minimization of the cost function c_2 with respect to the continuous offset variable ℓ_o. The cost function for the given frame F is defined as

$$c_2\left(F\right) = \sum_{\ell \in \mathcal{L}^F} \left(\ell_o - \frac{\ell_w}{2}\right)^2. \tag{6}$$

The function c_2 enforces criterion C2 only. To enforce requirement R1, we define a constraint for each pair of labels $\ell^{(i)}$ and $\ell^{(j)}$ associated with anchors $a_x(\ell^{(i)})$ and $a_x(\ell^{(j)})$ in the given

frame F as

$$a_{\mathrm{x}}\left(\ell^{(i)}\right) - \ell_{\mathrm{o}}^{(i)} + \ell_{\mathrm{w}}^{(i)} \leq a_{\mathrm{x}}\left(\ell^{(j)}\right) - \ell_{\mathrm{o}}^{(j)}, \tag{7}$$

where we suppose an order so that $a_{\mathrm{x}}(\ell^{(i)}) < a_{\mathrm{x}}(\ell^{(j)}) \wedge \ell^{(i)} \neq \ell^{(j)}$ and $\ell^{(i)}, \ell^{(j)} \in \mathcal{L}^F$. Furthermore, to satisfy requirement R2 we define the constraints

$$\ell_{\mathrm{o}} \geq 0, \tag{8a}$$
$$\ell_{\mathrm{o}} \leq \ell_{\mathrm{w}}. \tag{8b}$$

Finally, we want to restrict a label overflow with vertical bounds of drawing region D_w. This is accomplished by a pair of

$$a_{\mathrm{x}}(\ell) - \ell_{\mathrm{o}} \geq 0, \tag{9a}$$
$$a_{\mathrm{x}}(\ell) - \ell_{\mathrm{o}} + \ell_{\mathrm{w}} \leq D_w. \tag{9b}$$

5. Interactive Labeling Method

The previously described method (in Sec. 4) is not suitable when the entire interaction with the scene is not known in advance. Continuous delivery of frames makes it impossible to retrieve the anchor and label intervals. Furthermore, the performance of the method is also an aspect of concern in interactive applications.

Therefore, we propose an interactive method OnlineTemporal that removes described pitfalls and at the same time reflects the requirements R1, R2 and satisfies the criteria C1–C4. The OnlineTemporal method, in contrast to OfflineTemporal, processes the entire interaction frame by frame and consists of only single stage wherein the position of the label is determined at once individually for each F (see a high-level overview in Fig. 3). We again place labels into discrete rows such that the vertical position ℓ_y is approximated by row $r \in R$, where $R = \{\, r \mid r \in \mathbb{N} \ \wedge \ r \leq |\mathcal{A}^F| \,\}$. Furthermore, we suppose that the width ℓ_{w} and the height ℓ_{h} are consistent for each F.

Single Stage Position Assignment. Given a frame F and its immediate predecessor F^{-1}, we determine the temporally stable position $(\ell_{\mathrm{x}}, \ell_{\mathrm{y}})$ for each label ℓ at once and without any knowledge of the following frames. Unlike the OfflineTemporal method, we can not fixate the vertical movement of a label due to its uncertain unfolding in the future. Therefore, we allow a label to change its vertical position; nevertheless, in the definition of the cost function, we minimize this behaviour in favor of criteria C3. Furthermore, we restrict the vertical change of a label within two consecutive frames to be at most a single row (*i.e.*, row ℓ_{r}^F of label ℓ in frame F is either $\ell_{\mathrm{r}}^F = \ell_{\mathrm{r}}^{F^{-1}}$ or $\ell_{\mathrm{r}}^F = \ell_{\mathrm{r}}^{F^{-1}} \pm 1$). This technique also narrows down the optimization search space, which in turn speeds up the computation of subsequent frames.

5.1. Single Stage Position Assignment

We formulate the problem as mixed-integer quadratic programming (MIQP). The instance of MIQP is formulated as the minimization of the cost function c_3 with respect to decision variable $I_\ell^r \in \{0, 1\}$, that indicates whether a label ℓ is placed in row r, and with respect to continuous offset variable ℓ_{o}. The cost function for any given frame F is defined as (for simplification the superscript of a current frame is omitted)

$$c_3\left(F^{-1}, F\right) = \sum_{\ell \in \mathcal{L}^F} \sum_{r \in R} I_\ell^r \left(w_{row} \hat{r} + w_{dist} \delta(\ell_{\hat{\mathrm{d}}}, \hat{r}) \right) \tag{10a}$$
$$+ \sum_{\ell \in \mathcal{L}^F} \sum_{r \in R} I_\ell^r \left(w_{row_\Delta} \left(r - \ell_{\mathrm{r}}^{F^{-1}} \right)^2 \right) \tag{10b}$$
$$+ w_{offset} \sum_{\ell \in \mathcal{L}^F} \left(\frac{1}{\ell_{\mathrm{w}}} \left(\ell_{\mathrm{o}} - \ell_{\mathrm{o}}^{F^{-1}} \right) \right)^2 \tag{10c}$$
$$+ w_{center} \sum_{\ell \in \mathcal{L}^F} \left(\frac{1}{\ell_{\mathrm{w}}} \left(\ell_{\mathrm{o}} - \frac{\ell_{\mathrm{w}}}{2} \right) \right)^2. \tag{10d}$$

The variable ℓ_{d} denotes the distance of anchor $a_{\mathrm{x}}(\ell)$ associated with label ℓ from the camera center. Further definitions of the delta function $\delta(\hat{d}, \hat{r})$ and hat modifier from Sec. 4.1 hold. The first term (10a) reflects the definition of a cost function c_1 from Sec. 4.1; therefore supports the criteria C1 and C4. The second term (10b) minimizes the vertical positional change of a label (criterion C3) in two consecutive frames F and F^{-1} . Similarly, the third term (10c) minimizes the horizontal positional change. The last term (10d) reflects the criterion C2. By observing the influence of various weights on the resulting layouts with the defined criteria (in Sec. 3) in mind, we recommend using the weights of the terms as $w_{row} = 0.5, w_{dist} = 0.8, w_{row_\Delta} = 1.0, w_{offset} = 0.1$ and $w_{center} = 0.3$.

To fulfill the requirement R1, we define the following constraint

$$a_{\mathrm{x}}\left(\ell^{(i)}\right) - \ell_{\mathrm{o}}^{(i)} + \ell_{\mathrm{w}}^{(i)} \leq a_{\mathrm{x}}\left(\ell^{(j)}\right) - \ell_{\mathrm{o}}^{(j)} + M \cdot \left(1 - I_{\ell^{(i)}}^r\right) + M \cdot \left(1 - I_{\ell^{(j)}}^r\right), \tag{11}$$

where we define the order so that $a_{\mathrm{x}}(\ell^{(i)}) \leq a_{\mathrm{x}}(\ell^{(j)}) \wedge \ell^{(i)} \neq \ell^{(j)}$ and $\ell^{(i)}, \ell^{(j)} \in \mathcal{L}^F$. To restrict the vertical position of any given label ℓ in frame F, which is placed in row $p = \ell_r^{F^{-1}}$ in the preceding frame F^{-1}, we introduce the constraint defined as

$$I_\ell^{p-1} + I_\ell^p + I_\ell^{p+1} = 1. \tag{12}$$

Furthermore, requirement R2 is defined in a similar way as in Sec. 4.1. Finally, a label ℓ is allowed to occupy only a single row r. Therefore, we define this restriction as

$$\sum_{r \in R} I_\ell^r = 1. \tag{13}$$

6. Extensions

Both proposed methods can be easily extended by additional terms and constraints to customize the resulting layout. In this section, we describe an extension for visibility optimization based on prominence and alpha-blending extension for smooth transitions of labels.

6.1. Feature Prominence and Visibility Optimization

Typically, some features of the visualized scene are more important than the other ones. Therefore, we define a *prominence* φ to express the importance of a label ℓ corresponding with an anchor $a(\ell)$. The prominence can be defined by a compound of several (potentially weighted) attributes. For example, the prominence of a mountain peak may be defined as a compound of its elevation, isolation, topographical prominence, and distance from the current viewpoint. To illustrate the *compound prominence* in our application, we define it as the weighted-sum of peaks' elevation ε, distance from the current viewpoint d, and Google score ω derived from a number of search results

$$\ell_\varphi = w_\varepsilon \ell_\varepsilon + w_d(1 - \ell_\mathrm{d}) + w_\omega \ell_\omega. \tag{14}$$

All attributes are normalized into the range $[0, 1]$. The weights were experimentally chosen as $w_\varepsilon = 1.0$, $w_d = 0.2$, and $w_\omega = 0.8$.

In crowded visualizations (*e.g.*, mountain terrain or city skyline) with many features to be labeled, it is sometimes useful not to show all the possible labels to prevent cluttered and chaotic label layout. More prominent features are more likely to be labeled and visualized; on the other hand, less prominent features do not need to be labeled at all.

Therefore, the definition of constraint Const. (13) from Sec. 5, and ditto Const. (5) from Sec. 4, can be redefined as

$$\sum_{r \in R} I^r_\ell \leq 1, \tag{15}$$

such that the label ℓ or the interval label λ, respectively, does not have to be placed in any row r. Furthermore, the cost function c_3 from Sec. 5.1, and ditto c_1 from Sec. 4.1, can be extended by term e_1 defined as

$$e_1(F) = -\rho_{\textcircled{v}} \sum_{\ell \in \mathcal{L}^F} \sum_{r \in R} I^r_\ell \left(\ell_\varphi + V(\ell, n)\right), \tag{16}$$

where the $\rho_{\textcircled{v}}$ is a reward for keeping the label visible, ℓ_φ is a compound prominence, and V is a *visibility function* that prevents labels from rapid disappearing. The visibility function V is piecewise-defined as

$$V(\ell, n) = \begin{cases} 1 & f - f_\ell^{\textcircled{n} \to \textcircled{v}} < n \\ 0 & else, \end{cases} \tag{17}$$

where f is the index of the current frame, $f_\ell^{\textcircled{n} \to \textcircled{v}}$ is the index of frame in which the last change from an *invisible state* $\textcircled{n}$ ($\forall r \in R : I^r_\ell = 0$) to a *visible state* $\textcircled{v}$ ($\exists r \in R : I^r_\ell = 1$) has occurred, and n is the number of frames for that the label should stay visible. To illustrate e_1 in our application, we set $\rho_{\textcircled{v}} = 0.9$, $n = 20$.

Similarly, to prevent labels from rapid changing from $\textcircled{n}$ to $\textcircled{v}$ we define the term e_2 as

$$e_2(F) = -\rho_{\textcircled{n}} \sum_{\ell \in \mathcal{L}^F} \left(1 - \sum_{r \in R} I^r_\ell\right)\left((1 - \ell_\varphi) + N(\ell, n)\right), \tag{18}$$

where the $\rho_{\textcircled{n}}$ is a reward for keeping the label invisible, ℓ_φ is a compound prominence, and N is an *invisibility function* piecewise-defined as

$$N(\ell, n) = \begin{cases} 1 & f - f_\ell^{\textcircled{v} \to \textcircled{n}} < n \\ 0 & else, \end{cases} \tag{19}$$

where $f_\ell^{\textcircled{v} \to \textcircled{n}}$ is the index of frame in that the last change from $\textcircled{v}$ to a $\textcircled{n}$ has occurred. The rest mimics the definition of e_1. To illustrate e_2 in our application, we set $\rho_{\textcircled{n}} = 0.1$, $n = 10$.

6.2. Smooth Label Transition

To prevent labels from popping in and out abruptly, we implement smooth alpha-blending. An extended labeling method is then denoted by the suffix Alpha (*e.g.*, OnlineTemporalAlpha). Let $\kappa^F_\ell \in [0, 1]$ be the *alpha value* of label ℓ in frame F. When a label ℓ is added to the scene, the alpha value is set to $\kappa^F_\ell = \kappa^{\textcircled{v}}_\Delta$, where $\kappa^{\textcircled{v}}_\Delta$ is a constant that defines the fade in speed. In the following frames $f, \ldots, f^{\textcircled{v} \to \textcircled{n}} - 1$, the κ^F_ℓ is increased by a *fade in* function such as

$$\kappa^F_\ell = \min\left(\kappa^{F^{-1}}_\ell + \kappa^{\textcircled{v}}_\Delta, 1\right), \tag{20}$$

where $\kappa^{F^{-1}}_\ell$ alpha value of label ℓ in the previous frame F^{-1}. In our application we use $\kappa^{\textcircled{v}}_\Delta = 0.1$ (*i.e.*, the label is fully visible in 10 frames). Similarly, when a label ℓ is removed from the scene, the κ^F_ℓ is decremented by a *fade out* function such as

$$\kappa^F_\ell = \max\left(0, \kappa^{F^{-1}}_\ell - \kappa^{\textcircled{n}}_\Delta\right), \tag{21}$$

where the $\kappa^{\textcircled{n}}_\Delta$ defines the fade out speed. In our application we use $\kappa^{\textcircled{v}}_\Delta = 0.2$ (*i.e.*, the label fully disappears in five frames). For simplicity, we use the linear fade in/out function.

Furthermore, to create a smooth transition of a label ℓ during the fade-out blending in the proposed OnlineTmeporal method, the position of corresponding anchor $a_\mathrm{x}(\ell)$ needs to be predicted because the disappearance of the anchor $a_\mathrm{x}(\ell)$ cannot be precomputed in advance. Therefore, we apply *linear extrapolation* to calculate the $a^F_\mathrm{x}(\ell)$ from the two consecutive frames F^{-2} and F^{-1} such as

$$a^F_\mathrm{x}(\ell) = a^{F^{-2}}_\mathrm{x}(\ell) + 2\left(a^{F^{-1}}_\mathrm{x}(\ell) - a^{F^{-2}}_\mathrm{x}(\ell)\right). \tag{22}$$

7. Results

We used GUROBI 9.0 with a C++ interface as an optimization solver for the OfflineTemporal as well as OnlineTemporal method. 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 and MIQP problem [23]. In case of the minimization of MIP problems, the branch-and-cut algorithm (for more details see, *e.g.*, Bixby *et al.* [24]) keeps track of the *upper bound* and *lower bound*. The upper bound UB (also called *incumbent*) is an objective value of the best feasible solution found so far. On the other hand, the

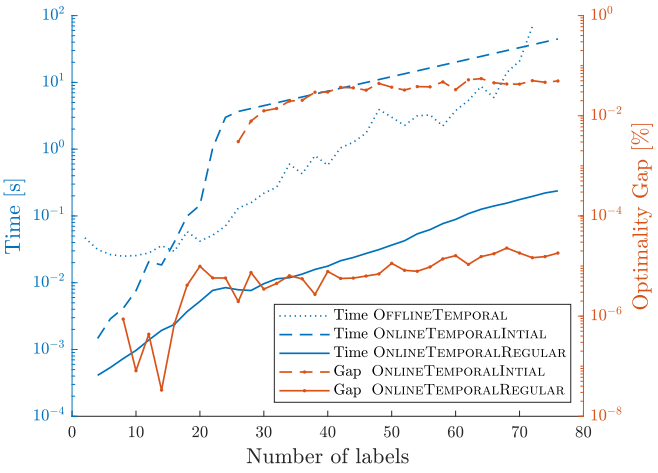


Figure 6. Time (left y-axis) and relative optimality gap (right y-axis) measurements for the proposed methods considering the number of labels in the scene.

lower bound LB is a minimum objective value of the LP-relaxed solutions (*i.e.*, integral constraints on variables are relaxed) in the leaf nodes of the branching tree. The absolute difference between upper and lower bound serves as a quality measure of the solution to optimality. The optimization is terminated when the relative *optimality gap* G defined as

$$G = \frac{UB - LB}{UB} \cdot 100 \tag{23}$$

is less than the G_T % (the value of $G_T = 0.01$ % is, in fact, the recommended termination criterion by the authors of GUROBI), meaning that the solver found a near-optimal solution that can not be further than the G_T % from the true-optima.

Note that, since the MILP, as well as MIQP, is NP-hard, only exponential-time algorithms are known, and the computational time can grow significantly with an increasing number of binary variables (*i.e.*, with the number of labels) [25, 26]. The following measurements were performed on Intel® Core i7-9700K @ 3.60GHz with 64GB of RAM.

7.1. Offline Labeling Method

The solution of the *label interval to row assignment* for the sequence with 40 labels takes 575ms and for smaller instances (20 labels and less) it is found in less than 42ms (see Fig. 6) with a relative optimality gap less than $G_T = 0.01$ %. The reported time measurements and optimality gaps are averaged over 100 runs.

The optimization in the *within row label placement* is defined as convex QP; hence it can be solved in polynomial time [27]. Moreover, the label placement can be solved independently for each row; therefore, the optimization is prompt and can run in parallel.

7.2. Online Labeling Method

The computation of the proposed OnlineTemporal method can be split into two phases. The solution for the *initial frame* F_0 is largely dependent on the search space, which is given by the number of labels. Each label ℓ can be placed at any row $r \in R$. The time needed to solve the initial frame is a period of time when the interaction is not possible; thus, one has to wait until the solution is found.

On the other hand, the search space for the following *regular frames* $F_1, F_2, \ldots, F_n$ is narrowed by the constraint Const. (12) (*i.e.*, the vertical change of any given label must be at most a single row). Therefore, label ℓ can be placed at any row ${\ell_r}^{F^{-1}} + c$ where the $c \in \{-1, 0, 1\}$.

To limit the duration of the optimization for the initial frame, we apply the time restriction

$$t_{limit} = \exp\left(0.05 \cdot |\mathcal{L}|\right), \tag{24}$$

which in turn can increase the relative optimality gap G. The solution for the initial frame F_0 with 40 labels takes 7.4s with a relative optimality gap of $G = 0.03$ %. The solution of smaller instances (20 labels or less) is found in less than 145ms with a relative optimality gap of $G \approx 0$ %.

To limit the duration of the optimization for the regular frames, we apply the time restriction

$$t_{limit} = \log\left(1 + 0.005 \cdot |\mathcal{L}|\right). \tag{25}$$

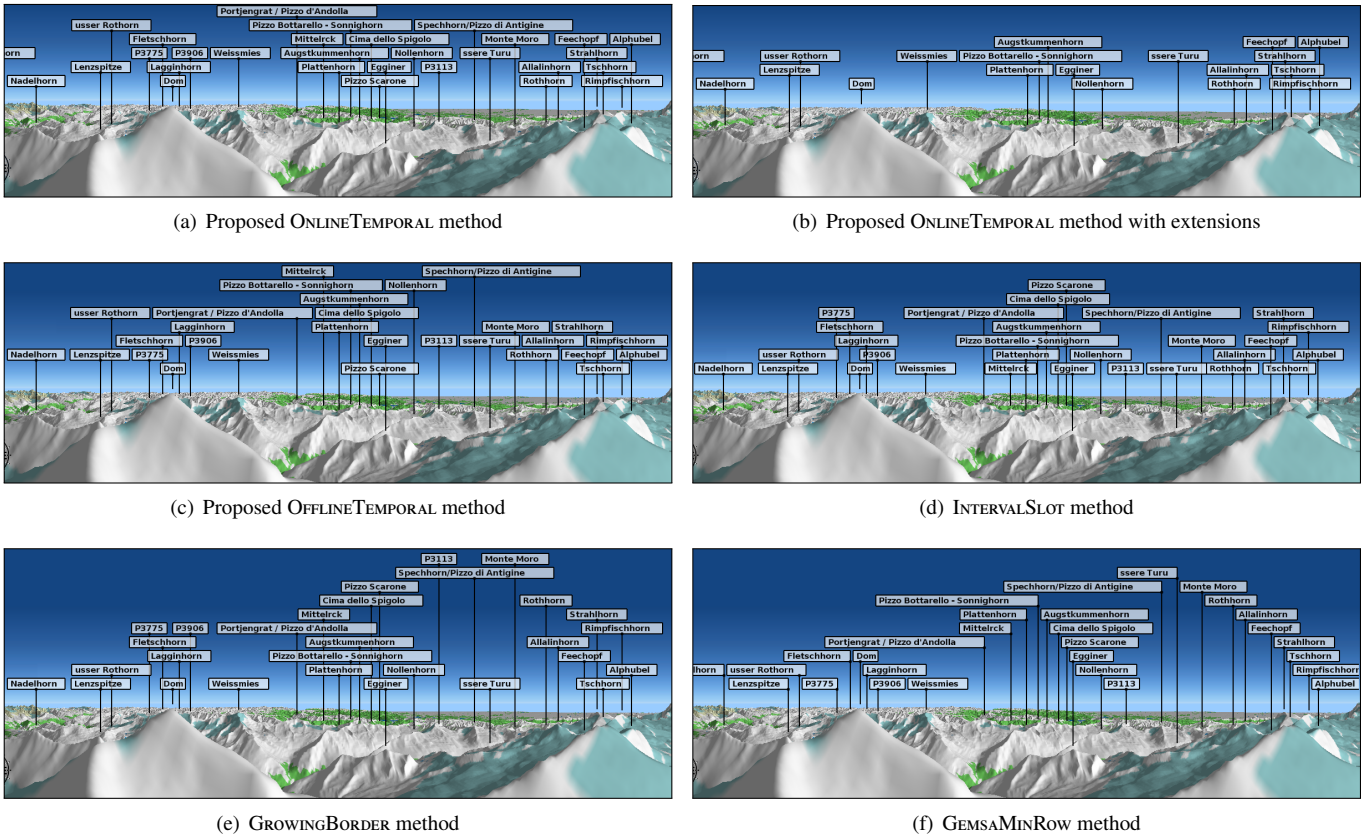
The solution for regular frames with 40 labels takes 18ms with a relative optimality gap of $G = 7.7 \times 10^{-6}$ %. The solution of smaller instances (20 labels or less) is found in less than 5ms with a relative optimality gap ranging from 3.33×10^{-8} to 9.87×10^{-6} %. The reported time measurements and optimality gaps are averaged over 100 runs. For more details see Fig. 6.

8. Comparison with State of the Art

We have implemented three previously published methods GrowingBorder [10], IntervalSlot [10], and GemsaMinRow [4] to compare them with the proposed OfflineTemporal and OnlineTemporal methods. The label layouts produced with these methods are shown in Fig. 7(a)–7(f).

The GrowingBorder and IntervalSlot methods [10] were designed for the annotation of dynamic virtual landscapes. They connect each label to its vertical leader at a port in the center of the bottom boundary of the label. Due to this consistency, each label changes only its relative vertical position to its anchor. The relative horizontal positions of each label to its anchor is always the same (*i.e.*, only the length of the leader changes). This makes the movement of labels temporally coherent. The consistency can also facilitate finding the corresponding label to the given anchor (and vice versa). However, due to this fact, the methods may produce label layouts with longer leaders. Furthermore, they allow the leaders to intersect with labels of other anchors. Note that the latter two features can make finding the corresponding label to the given anchor (and vice versa) harder.

The GemsaMinRow method [4] was not intended for the annotation of dynamic scenes. Therefore, we apply the method to each frame independently. We do not expect the method to achieve a temporally coherent movement of labels. The method



(a) Proposed OnlineTemporal method

(b) Proposed OnlineTemporal method with extensions

(c) Proposed OfflineTemporal method

(d) IntervalSlot method

(e) GrowingBorder method

(f) GemsaMinRow method

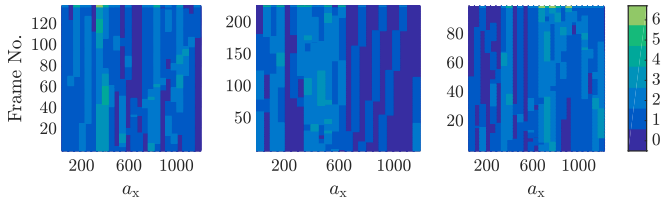
Figure 7. Example of label layouts calculated for the mountain peaks in the test sequence S_3 approximated by point anchors.

produces label layouts where the leaders do not intersect with any label, but create clusters of long leaders due to this constraint, see Fig. 7(f). We have included the method into the comparison to examine whether the intersections of leaders with labels influence the users' ability to find the corresponding label to the given anchor (or vice versa).

All compared methods were evaluated on three different sequences S_1, S_2, S_3 with a minimum length of 101 frames. The sequences are created by a series of horizontal and vertical movements that simulate, *e.g.*, the flight of a drone. Sequence S_1 consists of a long left *truck* (a leftwards horizontal movement of a camera) with a close anchor in front followed by a short *pedestal* down (a downwards vertical movement of a camera) where the anchors start to disappear rapidly. Sequence S_2 is composed as a sequence of a left truck, followed by a *dolly in* (a forward movement of a camera) completed by a pedestal up where the anchors start to appear rapidly. Sequence S_3 is created by a long right truck. Pan and tilt movements are not included as they do not introduce a parallax effect; therefore, the labeling does not change a lot, and it remains almost the same except for the newly appeared anchors on the edges of the drawing region D. Nevertheless, pan and tilt movements are included in the interactive experiment in Sec. 11.2.

To further describe the sequences, we measured several anchor related parameters and characteristics for each sequence, please see Tab. 1 and Fig. 8. The mean number of anchors within sequence S_1 and S_2 is 27 anchors, whereas, in the sequence S_3, it is only 21 anchors. The most noticeable changes concerning the anchors' x-position happen in the sequence S_1 (with the maximum shift of 31.97 pixels) and S_3 (with the maximum shift of 8.98 pixels). Regarding the anchors' y-position, only the anchors in the sequence S_1 and S_2 move dramatically. The length of an anchor interval reflects the distance that the anchor travels from the point it appears to the point it disappears. Allow us to point out that this also defines the space that is allocated for the smooth and uninterrupted movement of its label. Therefore, as the *Anchor Interval Length* section within Tab. 1 shows, in the sequence S_1 and S_3 exists an anchor that (a) moves very quickly in comparison to the other anchors, and (b) whose label interval allocates approximately a half of a row in a label layout. On average, an anchor is present in 98 out of 139 frames (71 %) in the sequence S_1, and more anchors disappear than appear throughout the sequence. The other two sequences S_2 and S_3 follows almost the same presence of 69 % and 67 %, respectively. The density maps of anchors' x-coordinates (a_x) depicted in Fig. 8 reveals that (with respect to the bin size of 50 px, $D_w \approx 1200$ px) in sequence S_1, there is one cluster of six anchors, 32 clusters of four anchors, 171 clusters of three anchors and 754 clusters of two anchors. For more details about sequences, please see Tab. 1 and Fig. 8. In addition, the tested sequences are available online as a part of the supplementary material.

For all compared methods, we evaluate if the label layouts produced for the sequences S_1, S_2, and S_3 are temporally co-



(a) Sequence S_1 (b) Sequence S_2 (c) Sequence S_3

Figure 8. Density of anchors' x-coordinates (a_x) over frames of test sequences S_1 (a), S_2 (b) and S_3 (c). The anchors are accumulated into bins with a width of 50 pixels.

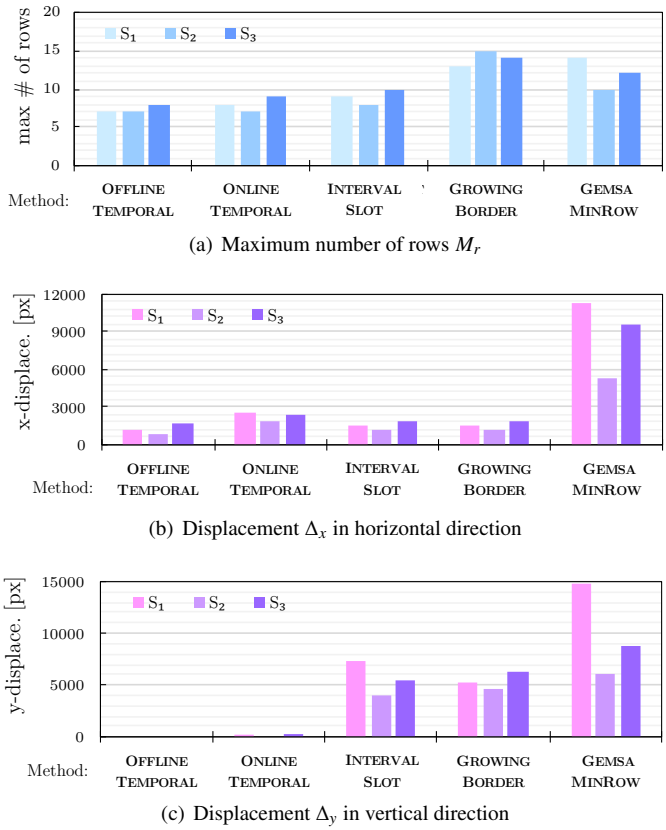
Parameters	Sequence		
	S_1	S_2	S_3
Number of Frames	139	228	101
Frame Dimensions	1200×400	1200×400	1281×346
Number of Anchors	*anchors*		
Minimum	21	19	26
Mean	27	21	27
Maximum	30	24	30
Anchor Shift in Frame	*pixels (x, y)*		
Minimum	(0.00, 0.00)	(0.00, 0.00)	(0.00, 0.00)
Mean	(0.36, 0.32)	(0.20, 0.19)	(0.54, 0.19)
Maximum	(31.97, 5.51)	(1.66, 2.40)	(8.98, 0.86)
Anchor Interval Length	*pixels*		
Minimum	1.20	0.10	0.90
Mean	38.40	30.68	44.48
Maximum	663.73	159.15	534.69
Anchor Presence	*no. of frames*		
Shortest	7	6	9
Mean	98	158	68
Longest	139	228	101
Anchor Presence Change	*no. of anchors*		
Appeared	19	18	23
Disappeared	24	19	22
Anchor Clusters	*no. of clusters (bin size 50 px)*		
Size 6	1	0	3
Size 4	32	6	3
Size 3	171	144	132
Size 2	754	1445	613

Table 1. Parameters of test sequences. Mean *Anchor Shift in Frame* is calculated from individual mean shifts of anchors over a sequence. *Anchor Interval Length* is equivalent to an absolute change of anchor's x-position.

herent and if the label layouts allow users to find the corresponding label to the given anchor (or vice versa). Furthermore, we evaluate users preferences among all compared methods.

9. Quantitative Evaluation

In the quantitative evaluation, we have measured properties (vertical compactness, vertical displacement, and horizontal displacement) of the produced label layouts for the sequences S_1, S_2, and S_3 for each of the compared methods.



(a) Maximum number of rows M_r

(b) Displacement Δ_x in horizontal direction

(c) Displacement Δ_y in vertical direction

Figure 9. Quantitative metrics: (a) The maximum number of rows per sequence of the label layout. Total label displacement per sequence in horizontal (b) and vertical (c) direction. The Δ_y for the OFFLINETEMPORAL method is zero. The tested sequences S_1, S_2, S_3 are represented by individual colors.

The vertical compactness of the label layout can be described by the *maximum number of rows* M_r in a sequence. The methods producing label layouts with lower M_r are able to position the same number of labels on a lower number of rows. The compactness is important as the labels must fit into a labeling region L of finite height. Furthermore, label layouts with high M_r will lead to long leaders that can make finding the corresponding label to the given anchor (and vice versa) harder.

The results (see Fig. 9(a)) show that the OFFLINETEMPORAL method, followed by the ONLINETEMPORAL method, achieves the best results. Note that the GROWINGBORDER and GEMSAMINROW methods achieve the worst results leading to longer leaders.

For the temporally coherent movement of labels, the labels must not jump abruptly. Therefore, we have calculated the *displacement metric* for the compared methods in horizontal Δ_x and vertical Δ_y direction separately as the sum of differences in the positions of all labels between all pairs of subsequent frames.

The results (see Fig. 9(b) and 9(c)) suggest that labels in our proposed methods are more temporally coherent than in the other compared methods. The most significant discrepancy is visible in horizontal displacement Δ_y. The GEMSAMINROW method achieves the worst results for both horizontal displacement Δ_x and vertical displacement Δ_y. This is expected as the

method was not designed for the labeling of dynamic scenes.

10. Accuracy Experiments

We have conducted a user study to assess whether the proposed OfflineTemporal and OnlineTemporal methods (1) improve the ability of the user to follow the labels in time and (2) influence the ability of the label layout to mediate the interconnection between the labels and the features.

For the evaluation, we have created a web application that 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. The first experiment was one factor with four levels. The independent variable was the labeling method. The four levels were our OfflineTemporal method and IntervalSlot, GrowingBorder, and GemsaMinRow methods. The follow-up experiment was one factor with one level. Again, the independent variable was the labeling method. The only level was the OnlineTemporal method. In both experiments, we evaluated the methods for three sequences S_1, S_2, S_3.

Both experiments were 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 sequences was counterbalanced with a 3x3 balanced Latin square [28, Section 5.11] to eliminate the carry-over effect. In a between-subject design, combining the results of the two experiments is trivial as each level is evaluated independently of the other levels.

Both experiments consisted of a series of three tasks defined as follows:

Task 1. Locate the label associated to a highlighted anchor.

Task 2. Locate the anchor associated to a highlighted label.

Task 3. Follow a moving label for two seconds and then select the label in the blind view (*i.e.*, the text of the label is not shown).

For a detailed description of the tasks, please see the supplementary document and video[1].

Each participant repeated each task 10 times for each sequence. We measured the *error rate* (the number of wrongly selected labels/anchors relative to all selected labels/anchors). Afterward, we conducted a subjective evaluation of the visual search easiness (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; σ = 6.49) in the first experiment and by 35 participants (three females) with the age ranging from 20 to 38 years ($\bar{x}$ = 24.03; σ = 4.62) in the second experiment. 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; σ = 7.04) in the first experiment and by 25 participants (two females) with the age ranging from 20 to 38 years ($\bar{x}$ = 23.92; σ = 4.76) in the second experiment. 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; σ = 7.29) in the first experiment and by 24 participants (two females) with the age ranging from 20 to 38 years ($\bar{x}$ = 23.92; σ = 4.86) in the second experiment.

For each task and each measured variable, we define a family of two null hypotheses – H_0^1: "The difference in the mean value of the OfflineTemporal method and the mean value of each other compared method is zero," and H_0^2: "The difference in the mean value of the OnlineTemporal method and the mean value of each other compared method is zero." The family of hypotheses consists of 7 pairwise comparisons.

We evaluated the collected data for all sequences together. We performed a statistical evaluation of the measured data using confidence intervals. We calculated the confidence intervals of the error rates as adjusted Wald intervals, a method recommended for completion rates [29, 30]. We calculated the confidence intervals for Likert scales as confidence intervals for rating scales [31, Chapter 3]. We used 95% confidence intervals for error rates and Likert scales.

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. 10(a) and 10(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. 10(c). Please note that the 95% confidence intervals cannot be directly used to visually evaluate the difference between the means if multiple pairwise caparisons are evaluated.

To detect whether the means of the measured data are significantly different, we have calculated [32] the p-value from the 95% confidence interval of the difference between the means for all seven pairwise comparisons. To keep the type 1 error at significance level $\alpha = 0.05$ for the whole family of hypotheses, we have adjusted the p-values with the Holm's [33] sequentially-rejective method using the Šidák equation [34] as described by Ludbrook [35]. Please note that in certain cases, the method produces the same adjusted p-values for several pairwise comparisons. We report the adjusted p-values in Fig. 10. As we are adjusting the p-values, not the significance level α, we compare all adjusted p-values with $\alpha = 0.05$.

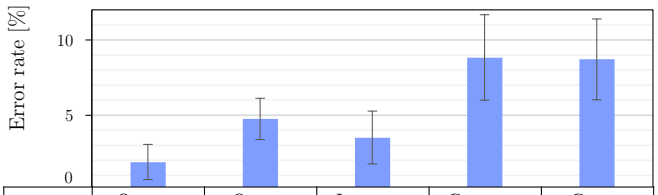
10.1. Task 1: Assign Label to Highlighted Anchor

The results (Fig. 10(a)) show that the OfflineTemporal method achieves a significantly lower error rate than the OnlineTemporal, GrowingBorder, and GemsaMinRow methods. The OnlineTemporal method achieves a significantly lower error rate than the GrowingBorder and GemsaMinRow methods.

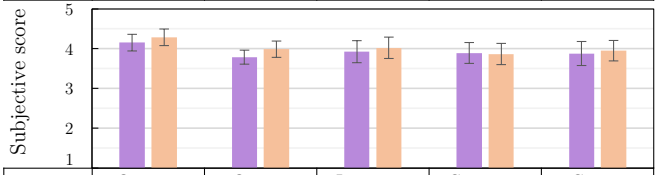
In the subjective evaluation, we have not detected any significant difference for the easiness and for the confidence.

The results indicate that the label layouts with longer leaders (GrowingBorder and GemsaMinRow) negatively influence the ability of users to assign the correct label to the selected anchor.

[1] Supplementary material is available at the project page `http://cphoto.fit.vutbr.cz/interactive-labeling/`

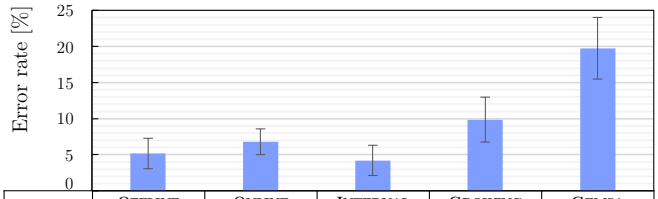


Method	OFFLINE TEMPORAL	ONLINE TEMPORAL	INTERVAL SLOT	GROWING BORDER	GEMSA MINROW
Error rate	1.89(0.73, 3.06)	4.75(3.38, 6.13)	3.52(1.77, 5.27)	8.83(6.0, 11.64)	8.71(6.03, 11.4)
OFFLINE TEMPORAL	—	$p = 0.0095$	$p = 0.2439$	$p < 0.0001$	$p < 0.0001$
ONLINE TEMPORAL	—	—	$p = 0.2810$	$p = 0.0397$	$p = 0.0397$

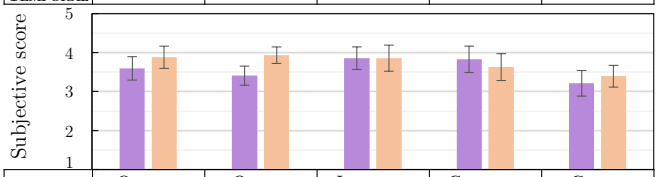


Method	OFFLINE TEMPORAL	ONLINE TEMPORAL	INTERVAL SLOT	GROWING BORDER	GEMSA MINROW
Easiness	4.15(3.95, 4.36)	3.79(3.61, 3.97)	3.93(3.65, 4.20)	3.90(3.63, 4.16)	3.88(3.58, 4.18)
OFFLINE TEMPORAL	—	$p = 0.0602$	$p = 0.5929$	$p = 0.5690$	$p = 0.5690$
ONLINE TEMPORAL	—	—	$p = 0.8023$	$p = 0.8023$	$p = 0.8023$
Confid.	4.29(4.08, 4.50)	3.99(3.78, 4.19)	4.02(3.76, 4.29)	3.87(3.60, 4.14)	3.95(3.70, 4.21)
OFFLINE TEMPORAL	—	$p = 0.2386$	$p = 0.4199$	$p = 0.2386$	$p = 0.2809$
ONLINE TEMPORAL	—	—	$p = 0.9733$	$p = 0.8672$	$p = 0.9733$

(a) Task 1: Assign label to highlighted anchor

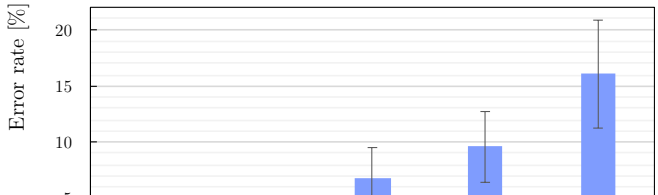


Method	OFFLINE TEMPORAL	ONLINE TEMPORAL	INTERVAL SLOT	GROWING BORDER	GEMSA MINROW
Error rate	5.17(3.06, 7.28)	6.8(5.01, 8.58)	4.22(2.12, 6.31)	9.87(6.76,12.98)	19.75(15.48, 24)
OFFLINE TEMPORAL	—	$p = 0.4399$	$p = 0.5397$	$p = 0.0688$	$p < 0.0001$
ONLINE TEMPORAL	—	—	$p = 0.2380$	$p = 0.2531$	$p < 0.0001$

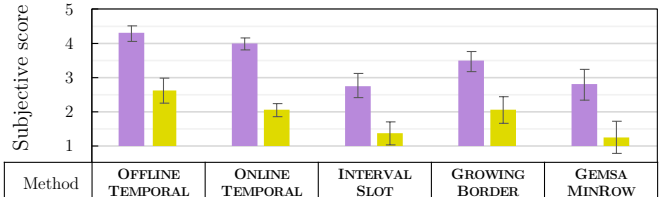


Method	OFFLINE TEMPORAL	ONLINE TEMPORAL	INTERVAL SLOT	GROWING BORDER	GEMSA MINROW
Easiness	3.60(3.30, 3.90)	3.41(3.16, 3.65)	3.86(3.57, 4.15)	3.83(3.49, 4.17)	3.21(2.88, 3.54)
OFFLINE TEMPORAL	—	$p = 0.6793$	$p = 0.6305$	$p = 0.6793$	$p = 0.3811$
ONLINE TEMPORAL	—	—	$p = 0.1337$	$p = 0.2562$	$p = 0.6793$
Confid.	3.88(3.60, 4.17)	3.93(3.72, 4.15)	3.86(3.52, 4.19)	3.63(3.28, 3.97)	3.39(3.11, 3.67)
OFFLINE TEMPORAL	—	$p = 0.9772$	$p = 0.9772$	$p = 0.7218$	$p = 0.0976$
ONLINE TEMPORAL	—	—	$p = 0.9772$	$p = 0.5299$	$p = 0.0183$

(b) Task 2: Assign anchor to highlighted label



Method	OFFLINE TEMPORAL	ONLINE TEMPORAL	INTERVAL SLOT	GROWING BORDER	GEMSA MINROW
Error rate	0.71(-0.10, 1.51)	2.41(1.31, 3.51)	6.77(4.03, 9.51)	9.56(6.41, 12.72)	16.05(11.24,20.86)
OFFLINE TEMPORAL	—	$p = 0.0145$	$p = 0.0001$	$p < 0.0001$	$p < 0.0001$
ONLINE TEMPORAL	—	—	$p = 0.0076$	$p = 0.0001$	$p < 0.0001$



Method	OFFLINE TEMPORAL	ONLINE TEMPORAL	INTERVAL SLOT	GROWING BORDER	GEMSA MINROW
Easiness	4.28 (4.05, 4.50)	3.97 (3.8, 4.15)	2.75 (2.40, 3.11)	3.46 (3.16, 3.75)	2.77 (2.32, 3.23)
OFFLINE TEMPORAL	—	$p = 0.0401$	$p < 0.0001$	$p < 0.0001$	$p < 0.0001$
ONLINE TEMPORAL	—	—	$p < 0.0001$	$p = 0.0065$	$p < 0.0001$
Focus	2.60 (2.23, 2.97)	2.03 (1.84, 2.22)	1.34 (1.01, 1.68)	2.03 (1.64, 2.42)	1.23 (0.76, 1.70)
OFFLINE TEMPORAL	—	$p = 0.0204$	$p < 0.0001$	$p = 0.0729$	$p < 0.0001$
ONLINE TEMPORAL	—	—	$p = 0.0030$	$p = 0.9903$	$p = 0.0085$

(c) Task 3: Follow the moving label

Figure 10. Results of the Accuracy Experiment: Error rate and subjective score for the task 1 (a), task 2 (b), and task 3 (c).

The leaders that do not intersect the labels (GemsaMinRow) do not compensate for the longer leaders. leaders created by the GemsaMinRow method are the reason for the poor performance of the method.

10.2. Task 2: Assign Anchor to Highlighted Label

The results (Fig. 10(b)) show that the OfflineTemporal and OnlineTemporal methods achieve a significantly lower error rate than the GemsaMinRow method.

In the subjective evaluation of task 2, we have not detected any significant difference for easiness. However, the participants were significantly more confident that they are selecting the correct anchor with the OnlineTemporal method than with the GemsaMinRow method. In our opinion, the clusters of long

10.3. Task 3: Follow the Moving Label

The results (Fig. 10(c)) show that the OfflineTemporal method achieves a significantly lower error rate than all other methods. The OnlineTemporal method achieves a significantly lower error rate than the IntervalSlot, GrowingBorder and GemsaMinRow methods.

In the subjective evaluation of task 3, the participants reported that the task was significantly easier to complete with the OfflineTemporal method than with all other methods. The task was significantly easier to complete with the OnlineTemporal

method than with the IntervalSlot, GrowingBorder, and GemsaMinRow methods. Furthermore, the participants reported that they had to focus significantly less with the OfflineTemporal method than with the the OnlineTemporal, IntervalSlot, and GemsaMinRow methods. With the OnlineTemporal method they had to focus significantly less than with the IntervalSlot and GemsaMinRow methods.

The results strongly indicate that lower displacement of labels between the frames improves the accuracy of the users in following a moving label.

10.4. Summary

In general, the results show that for the three sequences S_1, S_2, S_3 our proposed OfflineTemporal method followed by the OnlineTemporal method allow to follow labels moving in time significantly more accurately than the compared methods. At the same time, our OfflineTemporal and OnlineTemporal methods mediate the interconnection between labels and anchors the same as (IntervalSlot for tasks 1 and 2 and GrowingBorder for task 2) or better than (GemsaMinRow for tasks 1 and 2 and GrowingBorder for task 1) the compared methods. We were especially surprised by the poor performance of the GemsaMinRow method in tasks 1 and 2. It seems that forcing the leaders not to intersect with labels is counterproductive as it leads to clusters of long leaders.

In conclusion, we recommend using the proposed methods over the compared methods for sequences with similar characteristics as the sequences S_1, S_2, S_3.

11. Preference Experiments

To assess the users' preferences among different labeling methods, we have conducted two subjective experiments. The first was designed as non-interactive (*i.e.*, participants could not influence the pose of the camera in the scene), and the second as interactive (*i.e.*, participants were asked to interact with the camera in the scene).

11.1. Non-Interactive Environment

To capture users' preferences in a non-interactive environment, we conducted an experiment based on a psychophysical technique of paired comparisons [36, 37]. Specifically, we exploited the two-interval forced choice (2IFC) paradigm to verify the perceived quality of labeling methods (GemsaMinRow, GrowingBorder, IntervalSlot, OfflineTemporal, OnlineTemporal, and its extended version by alpha-blending denoted as OnlineTemporalAlpha).

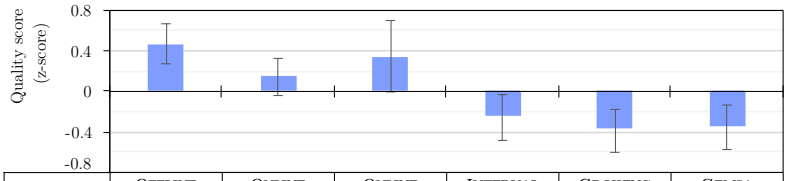
At the beginning of the experiment, participants were familiarized with the experimental procedure by the written instructions. Participants were asked to focus on the visual presentation of the labels and then select the method they liked the most. During the experiment, participants were able to play the assigned sequence as many times as they wanted. The names of the methods were transcoded with numbers. The stimuli were represented by three different video sequences presented in a web browser, and we evenly distributed them among the participants. Each participant was sequentially stimulated by a pair of labeling methods applied to the assigned sequence.

We have collected two groups of participants A and B, each consisting of 40 persons. The participants in group A — 40 males and 10 females with the age ranging from 19 to 54 years ($\bar{x} = 26.61$; $\sigma = 7.46$) — were asked to compare all pairs of GemsaMinRow, GrowingBorder, IntervalSlot, OfflineTemporal. Therefore, each participant in group A contributed with $\binom{m}{2} = 6$ pairwise comparisons where $m = 4$. Moreover, the order of the pairs of methods to compare was counterbalanced with a 6x6 balanced Latin square [28, Section 5.11] to eliminate learning and carry-over effects. Based on the outlier analysis tool provided by Pérez-Ortiz and Mantiuk [38], four participants (two males, two female) that behave very differently from the others were removed. The participants in group B — 36 males and four females with the age ranging from 17 to 44 years ($\bar{x} = 26.18$; $\sigma = 5.76$) — were asked to compare randomized pairs of {GemsaMinRow, GrowingBorder, IntervalSlot, OfflineTemporal} × OnlineTemporal extended by a pair of OnlineTemporal × OnlineTemporalAlpha which was presented as the last pair of the experimental procedure. Therefore, each participant in group B contributed with five pairwise comparisons. Together, group A and B created an incomplete experimental design (*i.e.*, only several pairs are compared) to reduce the number of needed pairwise comparisons.

We stored the data in the count matrix $\mathbf{C}$ for each participant separately. The element c_{ij} represents the number of times that method i was selected over method j. We converted the per-participant-count matrices $\mathbf{C}$ into a quality score (z-score) scale and computed a statistical significance using a customized MATLAB framework [38].

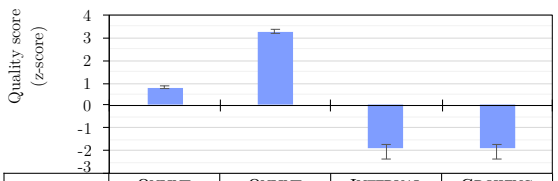
To transform the count matrix $\mathbf{C}$ to the quality score scale, we used Thurstone's Law of Comparative Judgment model concerning Case V [38, 37]. In order to reject the null hypothesis H_0^3: "the difference in perceived quality scores is zero," we applied the Two-tailed test at a significance level of $\alpha = 0.05$.

The quality scores for compared methods are depicted in Fig. 11(a). The results show that the proposed OfflineTemporal method exhibit the best quality score followed by OnlineTemporalAlpha and OnlineTemporal. The results also suggest that the best of the previously published methods is considered to be IntervalSlot followed by GemsaMinRow and GrowingBorder. The statistical significance for surveyed methods is presented in Fig. 12(a). The quality difference between the proposed OfflineTemporal and OnlineTemporal is statistically significant. Therefore, we can reject the null hypothesis H_0^3 for the pairs of OfflineTemporal × {OnlineTemporal, IntervalSlot, GrowingBorder, GemsaMinRow}. The OnlineTemporalAlpha has a higher quality score ($q = 0.34$) than OnlineTemporal ($q = 0.15$). However, H_0^3 can not be rejected for this pair and more generally for any other pair of {OfflineTemporal, OnlineTemporal} × OnlineTemporalAlpha. Therefore, the suggested additional quality of alpha-blending is not statistically proved. In addition, we have not detected significant difference in perceived quality among the IntervalSlot, GrowingBorder and GemsaMinRow methods.



Method	OFFLINE TEMPORAL	ONLINE TEMPORAL	ONLINE TMPALPHA	INTERVAL SLOT	GROWING BORDER	GEMSA MINROW
Quality score	0.46(0.27, 0.67)	0.15(-0.04, 0.32)	0.34(0, 0.7)	-0.24(-0.48, -0.03)	-0.37(-0.6, -0.18)	-0.34(-0.57, -0.13)

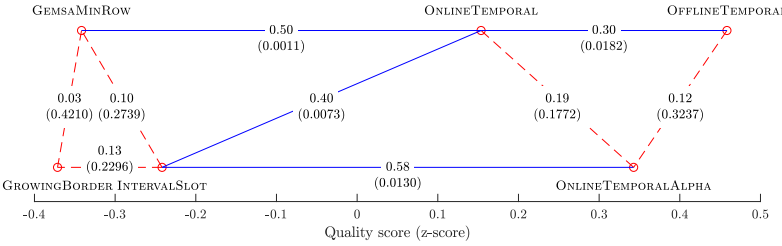
(a)



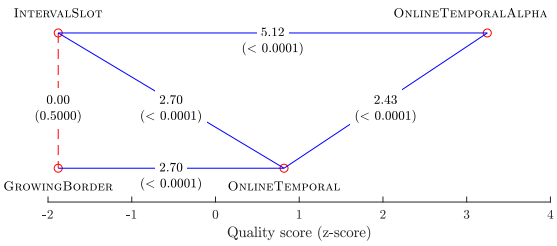
Method	ONLINE TEMPORAL	ONLINE TEMPALPHA	INTERVAL SLOT	GROWING BORDER
Quality score	0.82(0.75, 0.87)	3.25(3.19, 3.37)	-1.88(-2.37, -1.72)	-1.88(-2.37, -1.72)

(b)

Figure 11. Quality scores and 95% confidence intervals for (a) non-interactive and (b) interactive experiment.



(a)



(b)

Figure 12. Statistical significance (p-values reported in brackets) and quality scores for (a) non-interactive and (b) interactive experiment.

11.2. Interactive Environment

To capture users' preferences in the nature of an interactive environment, we conducted a follow-up experiment. Participants had to rank the labeling methods suitable for interactive applications (GROWINGBORDER, INTERVALSLOT, ONLINETEMPORAL and ONLINETEMPORALALPHA) from the best perceived method to the worst. We transcoded the names of the methods with numbers, and the first shown method was randomized. At the beginning of the experiment, participants were familiarized with the experimental procedure by the written instructions. We instructed the participants to focus predominantly on the assessment of label placement and the movement of the labels in time. After they read the instructions, the supervisor again repeated all the important details. The stimuli were represented by an interactive visualization of mountain terrain presented at a resolution of 1200x900, where the independent variable was the labeling method. During the experiment, participants were guided along the same predefined path above mountain peaks. They could interact with the scene by the following operations whenever they wanted: (1) fly forward/backward and stop, (2) rotate the camera, (3) zoom in and out, (4) return to the beginning of the path, and (5) change the labeling method. For convenience, we provided participants with printed cards to ease the ranking **R** during the experiment. At the end of the experiment, participants were asked to describe their decision process and to justify their ranking.

A total of 15 participants (two females) with the age ranging from 22 to 35 completed the experiment in $\bar{t} = 14$ minutes ($\sigma = 3.5$). Based on the described decision process and justification description, we removed three participants (two males, one female) that were wrongly focused on the other aspects of the presented stimulus (*e.g.*, the length of leaders, the rendering method, or correspondence of the label's row with the distance from the camera). The removed participants were also suggested by the outlier analysis tool provided by Pérez-Ortiz and Mantiuk [38].

To be consistent with the previous experiment, we chose once again to apply the pairwise comparison paradigm. We used the transitive closure to transform the ranking **R** to the count matrix **C**. For example, the participant's ranking A, B, C was transformed to a pairwise comparison $[A, B], [B, C], [A, C]$. Afterward, we derived the quality score by Thurstone's Law of Comparative Judgment model concerning Case V [38, 37]. In order to reject the null hypothesis (the same as H_0^3 from Sec. 11): "the difference in perceived quality scores is zero," we applied the Two-tailed test at a significance level of $\alpha = 0.05$.

The quality scores for the compared methods is depicted in Fig. 11(b). The results show that the proposed methods — ONLINETEMPORALALPHA, ONLINETEMPORAL — exhibit the best quality score followed by INTERVALSLOT and GROWINGBORDER. The statistical significance for the surveyed methods is presented in Fig. 12(b). The ONLINETEMPORALALPHA has a higher quality score ($q = 3.25$) than ONLINETEMPORAL ($q = 0.82$). Unlike the previous non-interactive experiment, the results show that the quality difference between the proposed ONLINETEMPORAL and ONLINETEMPORALALPHA is statistically significant. Therefore, we can reject the null hypothesis H_0^3, and the suggested additional quality of alpha-blending is, in this case, statistically proved. Furthermore, we can reject the null hypothesis H_0^3 for the pairs of {ONLINETEMPORAL ONLINETEMPORALALPHA} $\times$ {INTERVALSLOT, GROWINGBORDER}. Consistently with the previous experiment, we have not detected a significant difference in perceived quality between the INTERVALSLOT and GROWINGBORDER.

12. Conclusions

We proposed two novel temporally stable screen-space methods for boundary labeling of dynamic scenes using an optimization approach. The OfflineTemporal method is designed for the offline processing of the dynamic scene in advance (*e.g.*, drone-shot videos annotation, visualizations in television news). On the other hand, the OnlineTemporal method is designed for interactive applications (*e.g.*, terrain viewers, augmented, and virtual reality applications). Both proposed methods can be easily extended by additional terms and constraints to customize the resulting label layout, such as visibility optimization based on prominence and alpha-blending extension for smooth label transition. We show that according to the results of quantitative evaluation, the label layout is as compact as previous methods. At the same time, labels are more stable during an interaction with the scene. Furthermore, we compared the methods with three previously published methods in an extensive user study. The results of the accuracy experiment show that with our methods, the users can follow moving labels significantly more accurately than with the concurrent methods. At the same time, our methods mediate the interconnection ability between labels and features the same as or better than the other methods. Moreover, the results of the preference experiment show that the proposed methods were ranked the best for both interactive and non-interactive boundary labeling of dynamic scenes.

The proposed methods are seemingly more involved and harder to implement than the compared state-of-the-art methods. However, the labeling approached as an optimization problem yields a higher level of flexibility, which allows extending the proposed formulations for diverse needs. Furthermore, the difference between the lower and upper bounds used to solve MIP problems provide a quality measure of the solution to optimality. Another drawback of our approaches is the time-to-solve span for the initial frame which grows with an increasing number of labels in a scene. We tackled this issue by limiting the available time in favor of optimality; however, in future work, this could be approached differently (*e.g.*, by further narrowing the search space similar to the approach for regular frames).

Acknowledgments

This work was supported by project no. LTAIZ19004 Deep-Learning Approach to Topographical Image Analysis; by the Ministry of Education, Youth and Sports of the Czech Republic within the activity INTER-EXCELENCE (LT), subactivity INTER-ACTION (LTA), ID: SMSM2019LTAIZ; and by project no. CZ.02.1.01/0.0/0.0/16_019/0000765 Research Center for Informatics; by the Ministry of Education, Youth and Sports of the Czech Republic; and by project no. 7AMB17AT021; by the Ministry of Education, Youth and Sports of the Czech Republic within the activity MOBILITY (MSMT-539/2017-1); and partly by project no. TH03010330; by Technology Agency of the Czech Republic.

Computational resources were further supplied by the project e-Infrastruktura CZ (e-INFRA LM2018140) provided within the program Projects of Large Research, Development and Innovations Infrastructures.

References

[1] Fekete, JD, Plaisant, C. Excentric labeling: Dynamic neighborhood labeling for data visualizaition. In: CHI '99. ISBN 0201485591; 1999, p. 512–519.

[2] Ali, K, Hartmann, K, Strothotte, T. Label layout for interactive 3d illustrations. Journal of the WSCG 2005;13(1):1–8.

[3] Bekos, MA, Kaufmann, M, Potika, K, Symvonis, A. Multi-stack boundary labeling problems. WSEAS Transactions on Computers 2006;5(11):2602–2607.

[4] Gemsa, A, Haunertand, JH, Nöllenburg, M. Multi-row boundary-labeling algorithms for panorama images. ACM TSAS 2014;1(1):289–298.

[5] Bobák, P, Čmolík, L, Čadík, M. Video sequence boundary labeling with temporal coherence. In: Proceedings of Computer Graphics International 2019. Springer. ISBN 978-3-030-22514-8; 2019, p. 40–52.

[6] Bekos, MA, Niedermann, B, Nöllenburg, M. External labeling techniques: A taxonomy and survey. Computer Graphics Forum 2019;38(3):833–860.

[7] Oeltze-Jafra, S, Preim, B. Survey of labeling techniques in medical visualizations. In: Proc. of VCBM 14. Eurographics Association. ISBN 9783905674620; 2014, p. 199–208.

[8] Baboud, L, Čadík, M, Eisemann, E, Seidel, HP. Automatic photo-to-terrain alignment for the annotation of mountain pictures. In: CVPR '11. Washington, DC, USA: IEEE Computer Society. ISBN 978-1-4577-0394-2; 2011, p. 41–48.

[9] Benkert, M, Haverkort, H, Kroll, M, Nöllenburg, M. Algorithms for multi-criteria one-sided boundary labeling. In: Graph Drawing. Springer. ISBN 978-3-540-77537-9; 2008, p. 243–254.

[10] Maass, S, Döllner, J. Efficient view management for dynamic annotation placement in virtual landscapes. In: Smart Graphics. Springer. ISBN 978-3-540-36295-1; 2006, p. 1–12.

[11] Čmolík, L, Bittner, J. Layout-aware optimization for interactive labeling of 3d models. Computers & Graphics 2010;34(4):378–387.

[12] Balata, J, Čmolík, L, Mikovec, Z. On the selection of 2D objects using external labeling. In: CHI '14. ISBN 9781450324731; 2014,.

[13] Preim, B, Ritter, A, Strothotte, T, Pohle, T, Bartram, L, Forsey, DR. Consistency of rendered images and their textual label. In: Proc. of CompuGraphics 95. 1995, p. 201–210.

[14] Mühler, K, Preim, B. Automatic textual annotation for surgical planning. In: VMV '09. ISBN 9783980487481; 2009, p. 277–284.

[15] Mogalle, K, Tietjen, C, Soza, G, Preim, B. Constrained Labeling of 2D Slice Data for Reading Images in Radiology. EG VCBM 2012;:131–138.

[16] Götzelmann, T, Hartmann, K, Strothotte, T. Annotation of animated 3d objects. In: SimVis, SCS. Publishing House; 2007, p. 209–222.

[17] Stein, T, Décoret, X. Dynamic Label Placement for Improved Interactive Exploration. Proc of NPAR '08 2008;:15–21.

[18] Tatzgern, M, Kalkofen, D, Grasset, R, Schmalstieg, D. Hedgehog labeling: View management techniques for external labels in 3d space. In: 2014 IEEE Virtual Reality. 2014, p. 27–32.

[19] Hartmann, K, Götzelmann, T, Ali, K, Strothotte, T. Metrics for Functional and Aesthetic Label Layouts. Smart Graphics 2005;:115–126.

[20] Been, K, Daiches, E, Yap, C. Dynamic map labeling. IEEE Transactions on Visualization and Computer Graphics 2006;12(5):773–780.

[21] Benichou, M, Gauthier, JM, Girodet, P, Hentges, G, Ribiere, G, Vincent, O. Experiments in mixed-integer linear programming. Mathematical Programming 1971;1(1):76–94. doi:10.1007/BF01584074.

[22] Chen, DS, Batson, RG, Dang, Y. Applied Integer Programming: Modeling and Solution. John Wiley & Sons; 2011. ISBN 9780470373064.

[23] Gurobi Optimization, LLC, . Advanced Gurobi Algorithms. 2016. URL: http://www.gurobi.com/pdfs/user-events/2016-frankfurt/Die-Algorithmen.pdf.

[24] Bixby, ER, Fenelon, M, Gu, Z, Rothberg, E, Wunderling, R. Mip: Theory and practice — closing the gap. In: Powell, MJD, Scholtes, S, editors. System Modelling and Optimization. Boston, MA: Springer US. ISBN 978-0-387-35514-6; 2000, p. 19–49.

[25] Garey, MR, Johnson, DS. Computers and Intractability: A Guide to the Theory of NP-Completeness. New York, NY, USA: W. H. Freeman & Co.; 1979. ISBN 0716710455.

[26] Bliek, C, Bonami, P, Lodi, A. Solving Mixed-Integer Quadratic Programming problems with IBM-CPLEX : a progress report. Proceedings of the Twenty-Sixth RAMP Symposium 2014;(M):171–180.
[27] Ye, Y, Tse, E. An extension of karmarkar's projective algorithm for convex quadratic programming. Mathematical Programming 1989;44(1):157–179.
[28] MacKenzie, IS. Human-computer interaction: An empirical research perspective. Newnes; 2012.
[29] Agresti, A, Coull, BA. Approximate is better than 'exact' for interval estimation of binomial proportions. The American Statistician 1998;52(2):119–126.
[30] Sauro, J, Lewis, JR. 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. SAGE Publications; 2005, p. 2100–2103.
[31] Sauro, J, Lewis, JR. Quantifying the user experience: Practical statistics for user research. Elsevier; 2012.
[32] Altman, DG, Bland, JM. How to obtain the P value from a confidence interval. BMJ 2011;343(d2304). doi:10.1136/bmj.d2304.
[33] Holm, S. A simple sequentially rejective multiple test procedure. Scandinavian Journal of Statistics 1979;6(2):65–70.
[34] Šidák, Z. Rectangular confidence regions for the means of multivariate normal distributions. Journal of the American Statistical Association 1967;62(318):626–633.
[35] Ludbrook, J. Multiple comparison procedures updated. Clinical and Experimental Pharmacology and Physiology 1998;25(12):1032–1037.
[36] David, H. The method of paired comparisons. Griffin's statistical monographs & courses; C. Griffin; 1988. ISBN 9780852642900.
[37] Tsukida, K, Gupta, MR. How to Analyze Paired Comparison Data. UWEE Technical Report 206 2011;.
[38] Pérez-Ortiz, M, Mantiuk, RK. A practical guide and software for analysing pairwise comparison experiments 2017;URL: http://arxiv.org/abs/1712.03686. arXiv:1712.03686.