Computational Visual Media
DOI 10.1007/s41095-xxx-xxxx-x
Vol. x, No. x, month year, xx–xx

Article Type (Research/Review)

Smoothness Preserving Layout for Dynamic Labels by Hybrid Optimization

Yu He[1], Guo-Dong Zhao[2] and Song-Hai Zhang[1]

Abstract Stable label movement and smooth label trajectory are critical for effective information understanding. Sudden label changes cannot be avoided by whatever forced directed methods due to the unreliability of resultant force or global optimization methods due to the complex trade-off on the different aspects. To solve this problem, we proposed a hybrid optimization method by taking advantages of the merits of both approaches. We first detect the spatial-temporal intersection regions from whole trajectories of the features, and initialize the layout by optimization in decreasing order by the number of the involved features. The label movements between the spatial-temporal intersection regions are determined by force directed methods. To cope with some features with high speed relative to neighbors, we introduced a force from future, called temporal force, so that the labels of related features can elude ahead of time and retain smooth movements. We also proposed a strategy by optimizing the label layout prediction the trajectories of features so that such global optimization method can be applied to streaming data.

Keywords Label Layout, Smoothness Preserving, Dynamic label, Forced based, Static OptimizationHybrid Optimization.

1 Introduction

The automated placement of labels on maps and other visualizations is a long-standing goal of information visualization. Human map makers and designers can solve the labeling problem quite well, but an automated approach is often required for large dynamic data sets of visualizations.

Most optimization problems related to label placement are NP-hard, but several heuristics with different objectives and underlying principles have been proposed over the years. In this paper, we focus on the problem of boundary labeling, that is, the labels are placed around the perimeter of the visualization and connected to their corresponding features(e.g., points or image features) with leaders (i.e., lines). A drawback of boundary labeling, is that the associations between features and their labels are not immediately apparent nor easy to follow: one has to trace the leader across the visualization to find the corresponding label. In the literature, automated dynamic label placement is often split into two main approaches, each of which has its features and drawbacks:

- Global optimization with previous frames. The global optimization originally used into static label layout problem. Several aspects are taken into consideration, including relative position to its features, non-overlapping to other features, and other aesthetic rules, assuring a balanced layout without overlapping after optimization. For dynamic labeling, a common extension is optimization incorporating the label positions in previous frames. This strategy is global optimization only in spatial and NOT in temporal domain, so as to be in a dilemma in optimization occasionally. A more sophisticated approach is optimization in both spatial and temporal domain so that labels can adjust their trajectories ahead of time to remain moving smoothly. However, the problem is quite easy to formulate and quite difficult to optimize for all labels at all frames together. Moreover due to

[1] Affiliation of authors Yu He and Song-Hai Zhang, with the Department of Computer Science and Technology, Tsinghua University, Beijing, China.

[2] Affiliation of Guo-Dong Zhao, with the Department of Computer Science and Technology, Tianjin University of Technology, Tianjin, China.

Manuscript received: 2014-12-31; accepted: 2015-01-30.

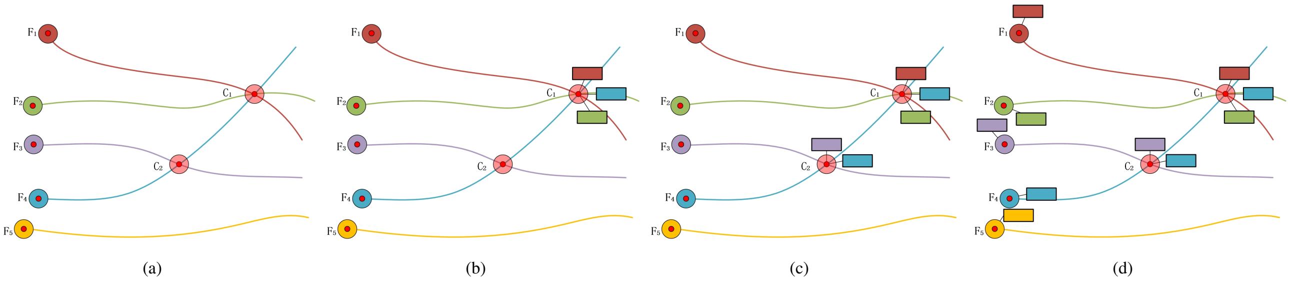


Fig. 1 A workflow of global spatiotemporal joints optimization on non-real-time data.

the trade off between temporal coherence and all other aspects in optimization also cannot guarantee temporal coherence especially in consecutive frames.

- Forced directed methods. All labels are moving according to the resultant force caused by all neighboring features and labels. Due to the forces calculated inverse to their relative distances, forced directed methods are inherently retaining smoothness of label movements. However, the resultant force only considers neighboring situations, and may cause interleaving and sudden changes if the labels haven't been carefully places initially.

In this paper, we propose a novel approach for solving the labeling problem in hybrid optimization. We define joint set as the subset of the features who influence the label layout of others in the set. We extract the joint sets from spatial and temporal domain, and layout relative labels with global optimization in descending order of the complexity of the joint sets. In between two joint sets, the force directed method are applied with initial positions of two joint sets. And thus we achieve the good initial layout on key regions and key frames, and move the labels according to resultant forces to achieve smoothness of the trajectories in between. In force directed framework, we also introduce a temporal force that copes with features with high speed relative to neighbors. The temporal force is calculated by the position and speed and added to the resultant forces so that the labels can elude ahead of time and retain smooth movements.

We also proposed a strategy to extend our hybrid framework to streaming data. By prediction of all feature trajectories, we estimate the joint sets in the future, so that the current layout can be optimized globally and our hybrid framework can be applied for the current layout.

Our main contributions are:

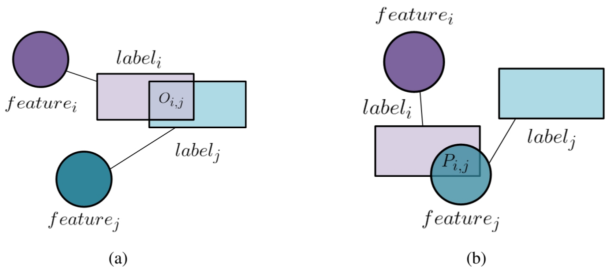
- A hybrid optimization method for dynamic label layout is proposed by cluster layout propagation to achieve smooth changing results.
- A temporal force is introduced to cope with features with high speed relative to neighbors so that the labels can elude ahead of time and retain smooth movements.
- The proposed layout method is extended to streaming data by prediction to avoid sudden changes.

2 Related work

Automatic label layout has been an active research area for decades, so a comprehensive review of all related work is beyond the scope of this paper. We focus on highlighting major developments and the works closely related to our approach. We refer the reader to the bibliography by Wolff and Strijk [20] for an extensive list of published heuristics and to the survey by Van Garderen [18] for a detailed literature review.

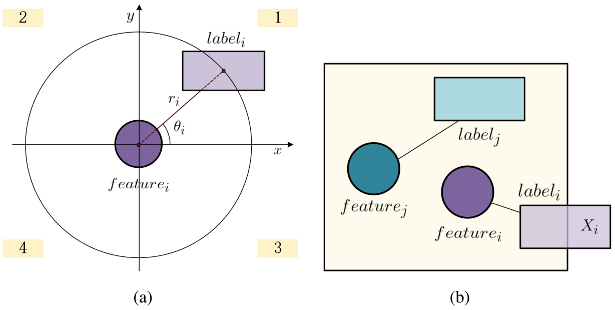
The labeling problem can be applied to geographical maps, technical maps or any other kind of diagrams, such as 3D scene, graph, etc. Labels are not necessarily textual labels; they can also be graphical representations of any form. The label can be placed internally or adjacent to its feature sometimes including leaders. Such problem has been studied extensively. It is formulated as an optimization with Several guidelines for "good" label placement[6] and non-overlapping constraints, which is known as a NP-Hard problem. Thus many approximations and heuristic approaches are proposed including integer linear programs[21], expert systems [4], simulated annealing[2], and combined optimization[19]. In recent years, Meng et al.[10] introduced a clutter model to prevent visual clutter in the layout. Lhuillier et al.[8] proposed a versatile density-based approach to label placement by applying kernel density estimation to the input features. These approaches presented a well organized layout result in the global scale, and however those optimization approaches did not take dynamic layout into consideration. In this paper, we proposed a hybrid solution taking the both advantages of global optimization[2] on global arrangement and forced based approaches on local smoothing together.

An common extension of the static label layout into dynamic labels is global optimization not only in spatial domain but also in temporal domain. However, the problem is quite easy to formulate and quite difficult



(a) (b)

Fig. 2 Two types of overlap. (a) overlap between labels and labels, (b) overlap between labels and features



(a) (b)

Fig. 3 (a) Feature-centered polar coordinate system, (b) label out of the axes.

to optimize for all labels at all frames together. A reasonable solution is optimization frame by frame with the initial position of labels in previous frame[1, 12, 13]. As the temporal coherence is only one aspect of all constraints, the trade off between temporal coherence and all other aspects in optimization cannot guarantee temporal coherence especially in consecutive frames.

Force-based approaches is another branch for label layout. A force-based approach was first presented by Hirsch[5]. He uses a gradient-driven heuristic to label point features. Labels are placed on a circle around their corresponding features. To resolve conflicts, vectors between overlapping labels are computed based on the intersection area. As a single label can collide with multiple other annotations, the sum of all its vectors guides its movement, thus improving the global labeling layout. Attractive forces pull labels to their point feature, while repelling forces push labels away from other features. Thus, step by step, the map layout converges towards a final labeling state through a gradient descent method. As far as we know, the forces are mainly consider the influence within spatial domain. In this paper, we proposed a temporal force which can influence over time, so as to cope with features with high speed relative to neighbors, so that the labels can elude ahead of time and retain smooth movements.

Force-based approaches[11, 15, 16] are inherently preserve local smoothness for dynamic label movements when the forces are defined as smooth function. So that force-based approaches can be directly extended to dynamic label layout with temporal coherence enforcement. However, the resultant force only considers neighboring situations, and may cause interleaving and sudden changes if the labels haven't been carefully places initially. To fully use of the feature of local smoothness, and coping with the sudden change problem of force-based methods, we incorporated global optimization to achieve good initial place of labels. A hybrid approach is presented by Stadler et al.[14]. They obtain an initial placement with the help of image processing. Then, iterative forces improve the chosen labeling positions.

3 Spatiotemporal Hybrid Optimization

Our method focuses on high-density and high-dynamic data visualization. Taking the point-feature label placement (PFLP) as an example [2], we solve both the problem of static layout optimization and dynamic movement smoothing of labels by spatiotemporal decoupling. Two key frameworks are performed in our method: global spatiotemporal joints optimization and label motion planning based on force traction.

3.1 Global Spatiotemporal Joints Optimization

In the process of this framework, we first locate all joint sets from spatial and temporal as is shown in Fig. 1a, and sort them in descending order of the complexity. There are two joint sets $\{C_1, C_2\}$ in this example. Then we perform static label placement on the frame where the joint sets are located, and the position constraints of labels transmit according to the order of the joint sets. Finally, the initial label layout of the first frame and the label position constraint of each joint set are obtained. The entire workflow is illustrated in Fig. 1.

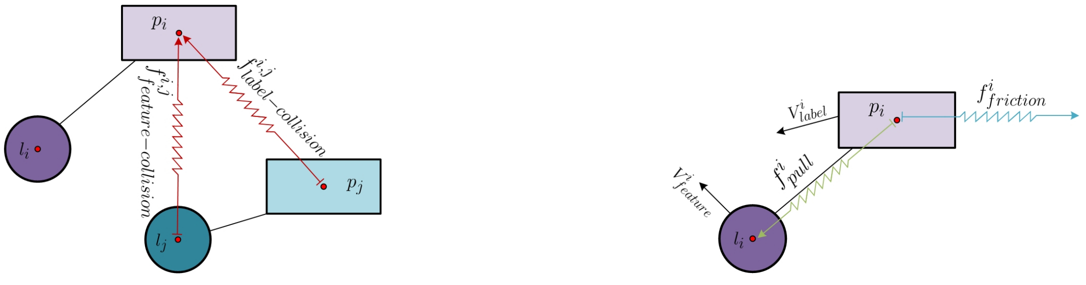
3.1.1 Static Label Placement Optimization

According to the definition of the PFLP problem in [2], one or zero of the candidate positions for placing a label have to be chosen for each feature, such that the value of the objective function of the resulting labeling is minimal. Following the labeling quality rules proposed in [7], our optimization objective function is as follows,

$$E_{static} = E_{overlap} + E_{position} + E_{aesthetics}$$

which consists three energy terms: overlap between labels and labels and overlap between labels and features $E_{overlap}$, the relative position of a label to a feature $E_{position}$, and aesthetics factors $E_{aesthetics}$.

$$E_{overlap} = \sum_{i}^{N} \sum_{j \neq i}^{N} W_{label-label} O_{i,j} + \sum_{i}^{N} \sum_{j}^{N} W_{label-feature} P_{i,j}$$



(a)

(b)

Fig. 4 Basic forces. (a) Repulsive forces between labels and repulsive forces between labels and features, (b) pulling forces between labels and the corresponding features and the friction force

where N is the number of feature. As is shown in Fig. 2, $O_{i,j}$ is the area of overlap between label i and label j, $P_{i,j}$ is the area of overlap between label i and feature j. $W_{label-label}$ and $W_{label-feature}$ are the corresponding weights.

$$E_{position} = \sum_{i}^{N} W_{orient}\Delta(\theta_i) + \sum_{i}^{N} W_{distance} r_i$$

where θ_i is the angle of the label in the corresponding feature-centered polar coordinate system as shown in Fig. 3a. $\Delta(\theta_i)$ is to determine which quadrant the label is in. W_{orient} has four values representing the priorities of the different quadrants. r_i is the distance between the label and feature, $W_{distance}$ is the corresponding weight.

$$E_{aesthetics} = \sum_{i}^{N} W_{out-of-axes} X_i + \sum_{i}^{N} W_{intersect} I_i$$

where X_i is the area of the label out of the axes as shown in Fig. 3b, I_i is the number of leader line intersections. $W_{out-of-axes}$ and $W_{intersect}$ are the corresponding weights.

Since PFLP is NP-complete, we adopt the heuristics search to find good solutions in acceptable time. Based on the algorithm evaluation in [7], simulated annealing algorithm is applied to solve the objective function. The annealing schedule is chosen according to [2].

3.1.2 Position Constraint Transmit

Fig. 1 shows an example to illustrate how the label position constraint transmits. Five features (F_1, F_2, F_3, F_4, F_5) and their corresponding motion paths are marked in different colors (red, green, purple, blue, yellow). There are two joint sets ($C_1\{F_1, F_2, F_4\}$, $C_2\{F_3, F_4\}$) in the entire spatial and temporal domain.

For the maximum joint set, we use the objective function E_{static} which is proposed in Sec. 3.1.1 to layout the labels, then we transmit this layout as position constraint to the next joint set based on the complexity descending order. The objective function is formulated as follows,

$$E = E_{static} + E_{constraint}$$

where E_{static} is the static label placement energy term, $E_{constraint}$ is the label position constraint transmitted from last joint set, which is formulated as

$$E_{constraint} = W_{radius} \|r_p - r_l\| + W_{angle} \|\theta_p - \theta_l\|$$

where (r, θ) is the coordinate value of the label in the corresponding feature-centered polar coordinate system as is shown in Fig 3a. (r_p, θ_p) represents the current joint set, and (r_l, θ_l) represents the last joint set. We only calculate the features which occur in both joint sets. In the above-mentioned example in Fig 1, only F_2 and F_4 need to calculate this term in the second joint set C_2. After traversing all joint sets, every feature in joint sets will obtain a position constrain at the first frame. As for features where no path intersection occurs such as F_5, a specified initial label position will be assigned.

3.2 Dynamic Label Movement Planning

In this framework, we focus on the movement of the labels, and both smoothness and readability are considered. After we obtain the initial layout constraints on key regions and key frames (Sec. 3.1), the force directed method is adopted to move the labels according to resultant forces to achieve smoothness of the trajectories in between [3]. In addition, to further improve readability, we also introduce a temporal force that copes with features with high speed relative to neighbors.

3.2.1 Force-based Label Movement

In this paper, we adopt a free annotation type to display label placement [17]. A free label i is positioned by an optional offset from its corresponding feature i as illustrated in Fig. 3a. Following the force descriptions in [3], we define four basic forces to maintain the smoothness and readability of label movement as is shown in Fig. 4.

Label-label Collision Force: To avoid overlap between labels, the first force introduced is repulsive force between labels. Since all labels are rectangles oriented in the same axis, we simplify the calculation of the distance between label i and label j,

$$d^{i,j}_{label-label} = max(|p_i - p_j| - \frac{1}{2}(s_i + s_j), 0)$$

where p is the center position of a label and s is the extent of a label. A label-label collision happens when $d_{label-label}$ is less than 0. The formula of label- label collision force is

$$f^{i,j}_{label-collision} = min(\frac{d^{i,j}_{label-label}}{D_{label-collision}} - 1, 0)\frac{p_i - p_j}{\|p_i - p_j\|}$$

where $D_{label-collision}$ is maximum distance the force will act, an illustration of this force can be seen in Fig. 4a.

Label-feature Collision Force: In this paper, each point feature has a specific size. Hence, repulsive forces between labels and features are also needed to prevent them from collision. Similar to label-label collision force, the distance between label i and feature j is

$$d^{i,j}_{label-feature} = max(|p_i - l_j| - \frac{1}{2}(s_i + r_j))$$

where l is the center position of a feature and r is the extent of a feature. The formula of label-feature collision force is as follows,

$$f^{i,j}_{feature-collision} = min(\frac{d^{i,j}_{label-feature}}{D_{feature-collision}} - 1, 0)\frac{p_i - l_j}{\|p_i - l_j\|}$$

where $D_{feature-collision}$ is maximum distance the force will act.

Feature-label Pulling Force: To ensure the label stays close to the corresponding feature, a pulling force is needed. As is shown in Fig. 4b, this force proportional to the distance, and which is formulated as follows,

$$f^i_{pull} = -ln\left(\left\||p_i - l_i| - \frac{1}{2}(s_i + r_i)\right\| - D_{pull} + 1\right)\frac{p_i - l_i}{\|p_i - l_i\|}$$

where D_{pull} is the minimum distance at which the force will be acted. The force act only if

$$\left\||p_i - l_i| - \frac{1}{2}(s_i + r_i)\right\| - D_{pull} > 0$$

Friction Force: To stabilize the movement of the labels, a friction force is acted to each label which is formulated as

$$f^i_{friction} = -c_{friction} \cdot (V^i_{label} - V^i_{feature})$$

where $c_{friction}$ is the friction coefficient, V_{label} and $V_{feature}$ are the velocities of labels and features. This force has a dampening effect on the labels except when they are moving in the same direction as their corresponding features.

3.2.2 Spatiotemporal Force Constraint

In allusion to high density and hybrid high dynamics data, two more forces are introduced to enhance the visualization performance.

Space Constraint Force: To reduce the label movement chaos caused by the intersections of feature motion paths, a space constraint force is acted to plan the movement path of a label as shown in Fig. 5. The formulation of this force is as follows,

$$f^i_{space} = ln\left(\|p_i - p'_i\| + 1\right)\frac{p_i - p'_i}{\|p_i - p'_i\|}$$

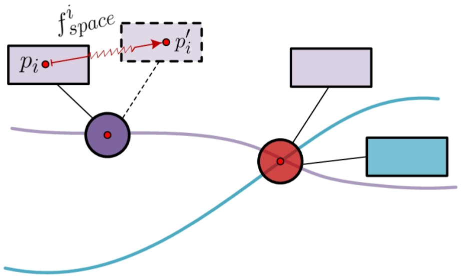


Fig. 5 Space force

where p' is the constraint position of label at upcoming feature joint set which is described in Sec. 3.1. If there is no new upcoming intersection, the constraint will remain the same as the previous one or the original one.

Time Constraint Force: The features of hybrid dynamics data have very different velocities. The basic forces mentioned in Sec.3.2.1 are unable to deal with the overlap under such a large velocity difference. Therefore, we introduce a time constraint force which is acted from where the features will be in the future. As is shown in Fig. 6, $V^j_{feature}$ is much bigger than $V^i_{feature}$. l'_j is the relative future position of the feature j, which is calculated as follows,

$$l'_i = l_i + (V^i_{feature} - V^j_{feature}) \cdot \delta t$$

where δt is the time of the feature movement to the future. The time constraint force is formulated as

$$f^{i,j}_{time} = ln\left(\frac{max(|v_i|, |v_j|)}{min(|v_i|, |v_j|)}\right) min(\frac{\|p_i - l_j\|}{\|p_i - l'_j\|} - 1, 0)\frac{p_i - l'_j}{\|p_i - l'_j\|}$$

3.2.3 Force Combination

After all forces are calculated for all labels, we combine them into one resultant force acting on each label. To weigh the forces against each other, we introduce a constant scaling factor for each force. The summarized force for label i is

$$\begin{aligned} f^i_{resultant} = &\sum_{j \neq i}\left(w_{label-collision} f^{i,j}_{label-collision} + w_{time} f^{i,j}_{time}\right) \\ &+ \sum_j\left(w_{feature-collision} f^{i,j}_{feature-collision}\right) \\ &+ w_{pull} f^i_{pull} + w_{friction} f^i_{friction} + w_{space} f^i_{space} \end{aligned}$$

We introduce a virtual mass m_i for label i. The current acceleration a_i is computed by Newton's second law

$$a_i = f^i_{resultant}/m_i$$

The time interval between successive frames is Δt, the new velocity v'_i can be calculated by Euler's integration method

$$v'_i = v_i + a_i \cdot \Delta t$$

In this paper, we take virtual mass of labels equal to 1.

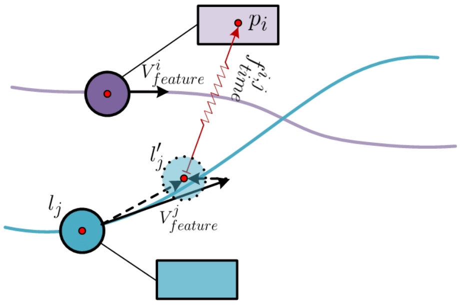


Fig. 6 Time force.

4 Implementation

We make two different implementations to evaluate the performance of our dynamic label layout method on both non-real-time and real-time data. They are both written in Python, are only based on CPU calculations. Both of them use epidemiological data on covid-19 in china. However, the first implementation use the data as recorded video, and the second one use the data as real-time video stream.In order to better visualize the epidemic data, we make several improvements to the visual elements. As is shown in Fig. 7, we extend the point feature to a ring feature to convey more information about the epidemic. We encode the cumulative number of confirmed cases of a province as the outer radius of the feature (red line in Fig. 7). The red-green ratio of the ring represents the ratio of the cumulative number of deaths to the cumulative number of cures. In addition, with so many provinces to display, the total area of all labels and features will be larger than the screen. We have to selectively hide some features and labels. Thus, we introduce a feature priority evaluation system for choosing which labels to be hidden, and the priority of a feature can be calculated as follows,

$$P^i_{feature} = \omega_1 \cdot R^i_{death} + \omega_2 \cdot R^i_{cure} + \omega_3 \cdot ln\left(N^i_{confirmed}\right)$$

where R_{death} is the ratio of death, R_{cure} is the ratio of cure, $N_{confirmed}$ is the cumulative number of confirmed cases, ω is the corresponding weight which allows the user to control tendency of priority determination. The lower priority labels or features will be hidden when overlapping with higher priority labels, but if features with different priorities are overlapping, the lower priority features do not need to be hidden. Video result demonstrations of the implementations can be found in supplementary material.

4.1 Non-real-time Data

The first implementation follows the steps described in Sec. 3. In the first step, we calculate the initial label positions and the position constraints at joint sets. In the second step, the priority of features are calculated to determine whether labels and features need to be hidden or not. In the third step, we calculate the resultant force for each label to guide the movement. As described in Sec.3.2.3, there are six forces acting on each label. However, after feature priority is introduced, we hope that the movement of the labels with higher priority are more pronounced, and the lower priority labels will give way to the higher priority labels during movement. When two labels are displayed and $label_i$ has a higher priority than $label_j$, the label collision force $f^{i,j}_{label-collision}$ will not be applied to $label_i$. But, the opposite forces $f^{j,i}_{label-collision}$ are still applied to $label_j$. In another case, when $label_i$ is displayed and $label_j$ is hidden, $f^{i,j}_{label-collision}$ is not applied to $label_i$ and the opposite forces $f^{j,i}_{label-collision}$ are still applied to $label_j$.

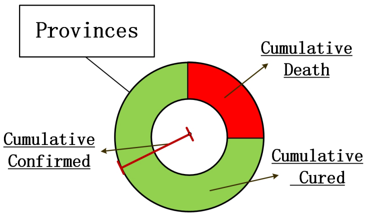


Fig. 7 Feature design for epidemic data.

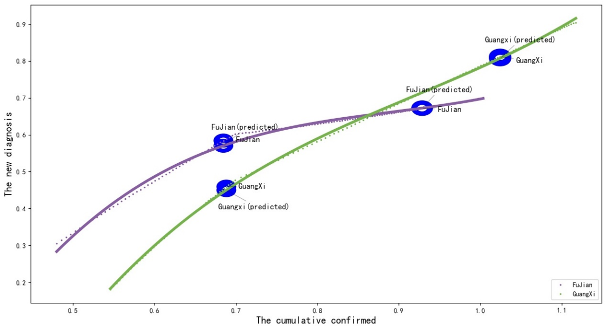
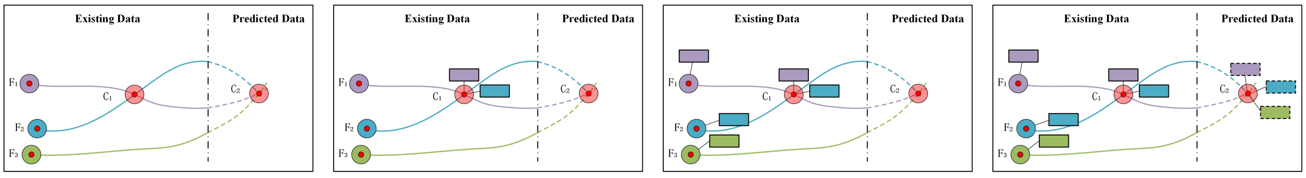


Fig. 8 Label layout for real-time data stream, two example cities which are FuJian (purple) and GuangXi (Green). The dotted line represents the movement trajectory of the city based on the real data, while the solid line represents the prediction data.

4.2 Real-time Data

The second implementation is for real-time video stream. Since we are unable to obtain the full path of the features, our hybrid optimization method needs a small modification. In the first step, we can only use the existing joint sets to calculate the label position constraints. In order to enable the labels to have a preperception to the upcoming joint set, we predict the feature motion based on the existing data by using an exponential curve prediction model [9]. As is shown in Fig. 9, the entire workflow is slightly modified from which is described in Sec. 3.1. For the existing part, we apply static optimization following Sec. 3.1, the initial label positions and the position constraints of joint



(a) (b) (c) (d)

Fig. 9 A workflow of global spatiotemporal joints optimization on real-time data.

Tab. 1 Evaluation of the effectiveness of the hybrid optimization method.

Method	Overlap	Position	Aesthetics	Smoothness	
				θ	r
Static optimization [7]	86.6	147.1	8.2	61.6	102.5
Force based [3]	116.7	157.7	7.7	67.9	36.9
Hybrid optimization	92.2	156.8	7.9	**47.8**	**28.2**

set ($C_1\{F_1, F_2\}$) can be calculated. For the predicted part, if a new intersection ($C_2\{F_1, F_2, F_3\}$) occurs, the static optimization of this joint set will be constrained by the maximum intersection in the existing data. The next two steps follow the flow of the first implementation, which is converting the label position constraint into force traction to guide the label movement.

5 Experiments

We implement the hybrid optimization method on the epidemiological data on covid-19 in China. Figure 10 shows the visualization results of four different frames of the epidemic data. The ring-features represent provinces of China, which are placed based on their new diagnosis (y-axis) and the cumulative confirmed (x-axis). Epidemic data of 34 provinces (some provinces in Fig. 10 are hidden due to low priority) from January 21 to April 10 are included in this experiment. This figure shows effectiveness of our hybrid optimization on static label placement, in which both readability and aesthetics are maintained. The dynamic video visualization result can be found in the supplementary material, which shows effectiveness of our method on label movement smoothness. To verify and evaluate the effectiveness of our hybrid optimization method on smoothness preserving, we compare the results produced by the force-based label placement method [3] with and without spatiotemporal force constraint. The force-based label placement method can place labels both with smoothness and readability for dynamic data, thus choosing this label placement method as a base can reflect the effectiveness of our spatiotemporal force constraint in these two aspects. Here we use the force-based label placement method with the parameters mentioned in [3]. To compare the performance of methods on high density and hybrid high dynamics data, the comparisons are performed on the epidemic data of China.

As shown in Fig. 11, we compare the partial layout results on the epidemic data at four different frames. In the video sequence, JiangSu first approaches and intersects HeNan, and then HeNan approaches FuJian. Basic forces without space constraint force are shown in Fig. 11a. Due to the absence of initial position constraints and early awareness of upcoming intersection, the motion path of the label appears chaotic and the leads are too long (in order to escape overlapping), which greatly reduces the readability of labels. Oppositely, the path of the label can be planned rationally with space constraint force, and initializing the label position can also effectively improve the label movement and reduce the visual confusion caused by unreasonable paths. Complete video of the comparison experiments can be found in the supplementary material.

Fig. 12 shows the comparison results of basic forces with or without the time constraint force. Partial features of epidemic data are displayed in this experiment. As is shown in Fig. 12, the movement velocity of HeBei is much bigger than other features. In a situation that there is a huge difference in velocity between the features, the basic forces can not timely adjust the label's motion trajectory to avoid label overlap. It is obviously in Fig. 12a, with HeBei approaching quickly, the label of HeiLongJiang cannot elude the feature in time, so that the lead line has to be stretched, which seriously affects the readability. However, as shown in Fig. 12b, the time constraint force can make the label perceive the approaching feature in advance and adjust the label position in advance, so as to make the label movement more reasonable and smooth. The comparison video also can be found in supplementary material.

Fig. 8 shows the performance comparison of our proposed method on real-time data stream. Two example cities which are FuJian (purple) and GuangXi (Green). The dotted line represents the movement trajectory of the city based on the real data, while the solid line represents the movement trajectory of the city based on the prediction as described in Sec.4.2. The deviation of tag trajectory and the result of tag layout show that our method can carry out automatic layout of moving labels on the real-time data stream of real data.

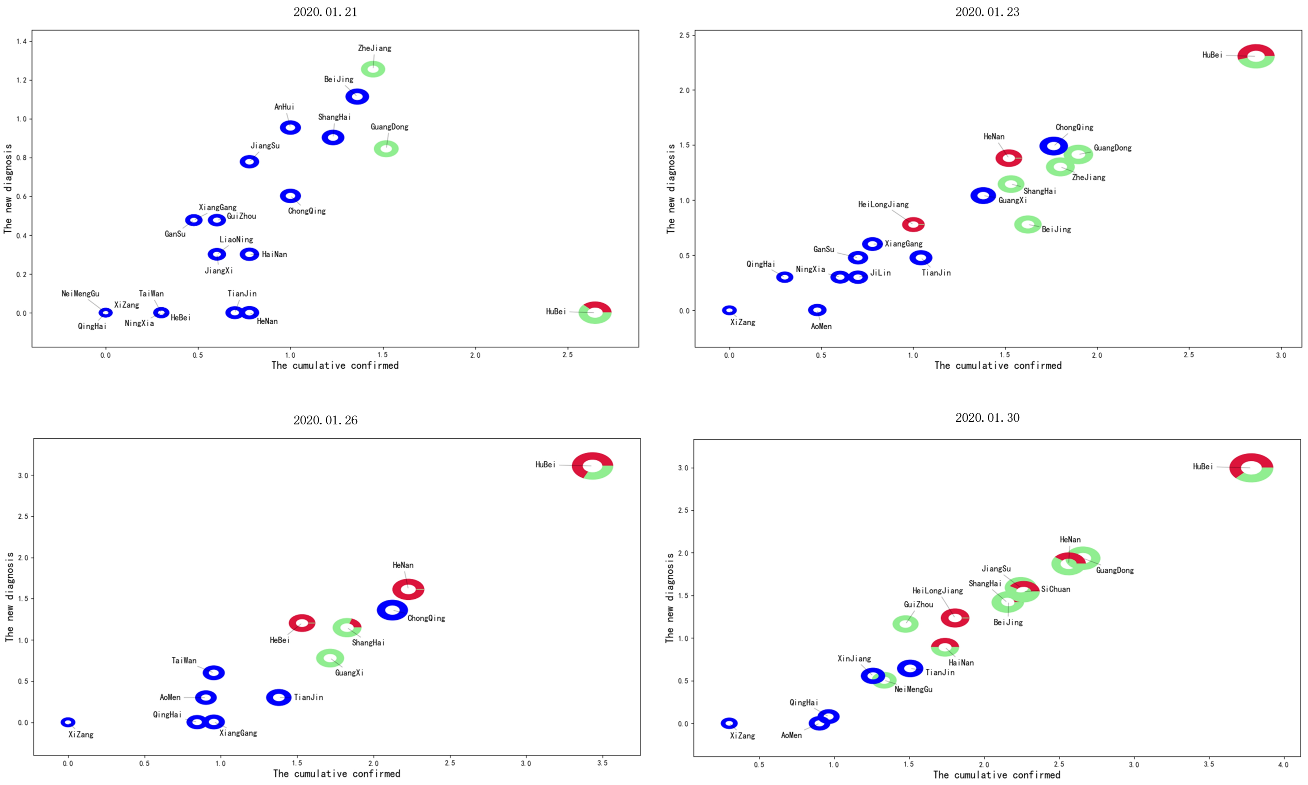


Fig. 10 Hybrid optimization dynamic label layout on epidemiological data on covid-19 in china .

In order to evaluate the performance of our method quantitatively, we compare the results produced by static optimization method [2], force-based method [3] and our hybrid optimization method. The static optimization method can place labels with both rationality and readability, and the force-based label placement method can maintain the smoothness of label movement. Thus we compare these labeling methods to reflect the effectiveness of our hybrid optimization in aforementioned three aspects. The comparisons are performed on epidemiological data on covid-19 in China.

As shown in Table 3, we compared the results in the following aspects: the overlap between labels and labels and overlap between labels and features $S_{Overlap}$, the distance from the label to the corresponding feature $S_{Position}$, the number of leader line intersections $S_{Aesthetics}$, and the relative position changes of labels between adjacent frames $S_{Smoothness}$.

The first three are based on the static label layout quality rules [7], which mainly evaluates the quality of the label layout in each frame, including rationality and readability. The calculation methods are similar to the calculation of energy terms described in Sec.3.1.1 without adjustment weights. The calculation formulas are as follows,

$$S_{Overlap} = \frac{1}{M}\sum_{k}^{M}(\sum_{i}^{N}\sum_{j\neq i}^{N} O_{i,j} + \sum_{i}^{N}\sum_{j}^{N} P_{i,j})$$

where N is the number of features, M is the number of frames. As is shown in Fig. 2, $O_{i,j}$ is the area of overlap between label i and label j, $P_{i,j}$ is the area of overlap between label i and feature j. The unit of overlap is $pixel^2$.

$$S_{Position} = \frac{1}{M}\sum_{k}^{M}\sum_{i}^{N} r_i$$

where r_i is the distance between the label and feature as shown in Fig. 3a. The unit of distance is *pixel*.

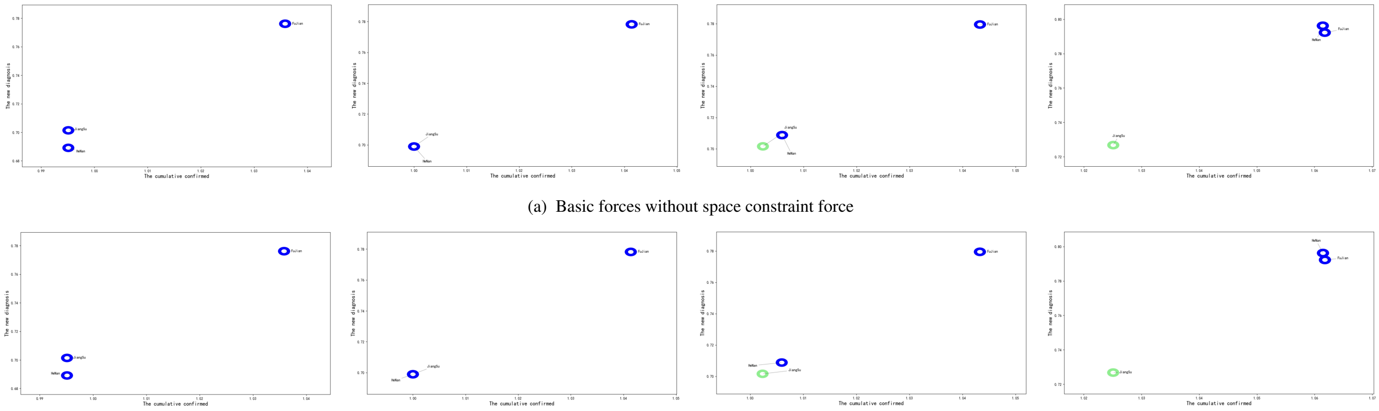
$$S_{Aesthetics} = \frac{1}{M}\sum_{k}^{M}\sum_{i}^{N} I_i$$

where I_i is the number of leader line intersections.

The last item evaluates the smoothness of label movement. We calculate the relative position changes of labels between adjacent frames. In the feature-centered polar coordinate system, the relative position changes include two parts ($S^{\theta}_{Smoothness}$ and $S^{r}_{Smoothness}$). The calculation formula is as follows,

$$S^{\theta}_{Smoothness} = \frac{1}{M-1}\sum_{k}^{M-1}\sum_{i}^{N} |\theta_{i,k} - \theta_{i,k+1}|$$

where θ_i is the angle of the label in the corresponding feature-centered polar coordinate system as shown in



(a) Basic forces without space constraint force

(b) Basic forces with space constraint force

Fig. 11 Space force constraint.

Fig. 3a. The unit is *degree*.

$$S^{r}_{Smoothness} = \frac{1}{M-1}\sum_{k}^{M-1}\sum_{i}^{N}\left|r_{i,k} - r_{i,k+1}\right|$$

where r_i is the distance between the label and feature, and the unit is *pixel*.

According to the data in Table 3, the average overlap area and label-feature distance of static optimization method are less than other two methods' results. Although as a trade off, the number of lead line intersections increases. The overall quality of label layouts produced by static optimization method is slightly better than those produced by force-based method. However, the static optimization method has a poor performance in the smoothness of label movement, although we add the change of label position between frames as an energy term to the objective function. Compared to static optimization method, the results produced by force-based method are much better in terms of smoothness. However, when dealing with high density and hybrid high dynamics data, the quality of label layouts produced by force-based method with basic forces become poor because of lacking good path planning and decoupling of high differential dynamic features. The hybrid optimization method proposed in this paper can effectively solve the confusion of label path caused by dense data intersection, obtain better initial label position through global optimization, and reasonably plan the label path through joint sets decoupling. The quality of label layouts are comparable to that produced by static optimization method. Further more, the introduced time force can effectively solve the problem of label overlap caused by the huge difference of feature speed. In terms of smoothness of label movement, hybrid optimization method is also better than the other two methods.

Three more experiments of real-world data are shown in the supplementary video. Additional screen shots are found in appendix B.

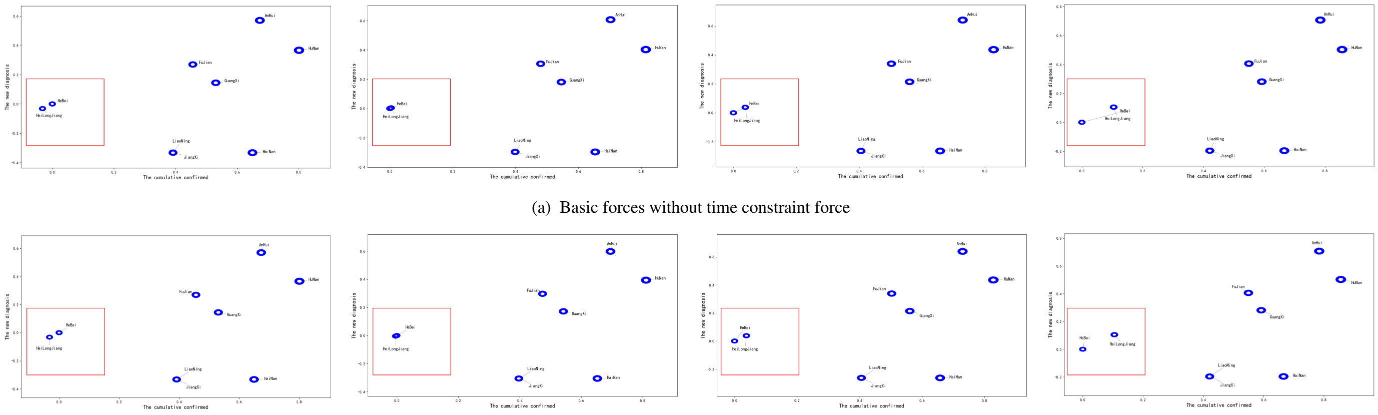
6 Conclusion

We have proposed an effective hybrid optimization method for dynamic label layouts. The hybrid framework consists of global spatiotemporal joints optimization and dynamic label movement planning. In global spatiotemporal joints optimization, the initial label positions and position constraints of joint sets are calculated. In dynamic label movement planning, the position constraints are converted into forces to guide the label motion, four basic forces and a temporal force are introduced to maintain the smoothness of label movement. Experiments on real-world data sets show that our hybrid optimization dynamic labeling method is able to produce clear and legible label layouts on each frame and smooth label movements in the dynamic process.

Our current hybrid optimization dynamic labeling framework needs to be improved in some respects in the future. First, the parameter Settings should be optimized. We aim at designing a adaptive parameter selection method, which can greatly accelerate the process of adjusting parameters. Second, to achieve real-time labeling, we need an efficient computation of forces between labels. The GPU can be used for parallel processing of these tasks [17].

Acknowledgements

The authors would like to thank all reviewers for their thoughtful comments. This work was supported by the National Key Technology R&D Program (Project Number 2017YFB1002604), the National Natural Science Foundation of China (Project Numbers 61772298, 61832016), Research Grant of Beijing Higher Institution



(a) Basic forces without time constraint force

(b) Basic forces with time constraint force

Fig. 12 Time force constraint.

Engineering Research Center, and Tsinghua–Tencent Joint Laboratory for Internet Innovation Technology.

References

[1] B. Bell, S. Feiner, and T. Höllerer. View management for virtual and augmented reality. In *ACM Sympos. on User Interface Software and Technology (UIST'01)*, pages 101–110, 11–14 Nov. 2001.

[2] J. Christensen, J. Marks, and S. M. Shieber. An empirical study of algorithms for point-feature label placement. *ACM Transactions on Graphics*, 14(3):203–232, 1995.

[3] S. Ekman. Force-based label placement for dynamic data. 2018.

[4] H. Freeman. An expert system for the automatic placement of names on a geographic map. *Information Sciences*, 45:367–378, 1988.

[5] S. A. Hirsch. An algorithm for automatic name placement around point data. *The American Cartographer*, 9(1):5–17, 1982.

[6] E. Imhof. Positioning names on maps. *The American Cartographer*, 2(2):128–144, 1975.

[7] S. Krogmann. Automatic 2d label placement, 03 2017.

[8] A. Lhuillier, M. Van Garderen, and D. Weiskopf. Density-based label placement. *The Visual Computer*, 35:1041–1052, 2019.

[9] Y. Lu, Q. Li, and Y. Guo. Improved prediction model of modified exponential curve. pages 513–516, 07 2012.

[10] Y. Meng, H. Zhang, M. Liu, and S. Liu. Clutter-aware label layout. pages 207–214, 2015.

[11] S. Pick, B. Hentschel, I. Tedjo-Palczynski, M. Wolter, and T. Kuhlen. Automated positioning of annotations in immersive virtual environments. In *Proceedings of the Joint Virtual Reality Conference of EGVE - EuroVR - VEC, Stuttgart, Germany, 2010*, 2010.

[12] E. Rosten, G. Reitmayr, and T. Drummond. Real-time video annotations for augmented reality. pages 294–302, 2005.

[13] C. Sirk, D. Kalkofen, D. Schmalstieg, and A. Bornik. Dynamic Label Placement for Forensic Case Visualization. In B. Kozlikova, T. Schreck, and T. Wischgoll, editors, *EuroVis 2017 - Short Papers*. The Eurographics Association, 2017.

[14] G. Stadler, T. Steiner, and J. Beiglb?ck. A practical map labeling algorithm utilizing morphological image processing and force-directed methods. *Cartography Geographic Information Science*, 33(3):207–215, 2006.

[15] M. Tatzgern, D. Kalkofen, R. Grasset, and D. Schmalstieg. Hedgehog labeling: View management techniques for external labels in 3d space. In *Virtual Reality*, 2014.

[16] M. Vaaraniemi, M. Treib, and R. Westermann. Temporally coherent real-time labeling of dynamic scenes. In *Proceedings of the 3rd International Conference on Computing for Geospatial Research and Applications*, COM.Geo '12, New York, NY, USA, 2012. Association for Computing Machinery.

[17] M. Vaaraniemi, M. Treib, and R. Westermann. Temporally coherent real-time labeling of dynamic scenes. page 17, 2012.

[18] M. van Garderen. *Pictures of the Past : Visualization and visual analysis in archaeological context*. PhD thesis, Universität Konstanz, Konstanz, 2018.

[19] F. Wagner and A. Wolff. A combinatorial framework for map labeling. In S. H. Whitesides, editor, *Proc. Symp. 6th Internat. Symp. on Graph Drawing (GD'98)*, volume 1547 of *Lecture Notes Comput. Sci.*, pages 316–331. Springer-Verlag, 13–15 Aug. 1998.

[20] A. Wolff and T. Strijk. The Map-Labeling Bibliography. `http://i11www.ira.uka.de/map-labeling/bibliography`, 1996.

[21] S. Zoraster. Integer programming applied to the map label placement problem. *Cartographica*, 23(3):16–27, 1986.

Yu He received the master's and doctoral degrees from the Zhejiang University of Technology, in 2015 and 2019, respectively. He is currently a postdoctoral researcher at Tsinghua University. His research interests include 3D vision, virtual reality and visualization.

Guodong Zhao received his B.S. degree from Dalian Minzu University. He is currently a master student in the Department of Computer Science and Engineering, Tianjin University of Technology. His research interests include computer graphics and computer vision.

Songhai Zhang received the PhD degree of Computer Science and Technology from Tsinghua University, Beijing, in 2007. He is currently an associate professor in the Department of Computer Science and Technology at Tsinghua University. His research interests include image/video analysis and processing as well asgeometric computing.

Tab. 2 Parameters for the static optimization label layout method

Description	Symbol	Value	Unit
Area of label-label overlap	$W_{label-label}$	80	-
Area of label-feature overlap	$W_{label-feature}$	50	-
Orient score	W_{orient}	1-4	-
Distance of label-feature	$W_{distance}$	20	-
Out of axes	$W_{out-of-axes}$	320	-
Number of intersecting lead lines	$W_{intersect}$	1	-

Tab. 3 Parameters for the hybrid optimization dynamic label layout method

Description	Symbol	Value	Unit
Label-label collision force scale	$w_{label-collision}$	40-60	-
Label-label collision margin	$D_{label-collision}$	30	pixels
Label-feature collision force scale	$w_{feature-collision}$	50-100	-
Label-feature collision margin	$D_{feature-collision}$	17	pixels
Pulling force scale	w_{pull}	25	-
Pulling force margin	D_{pull}	18	pixels
Friction force scale	$w_{friction}$	6	-
Static friction coefficient	$c_{friction}$	0.7	-
Space force scale	w_{space}	20	-
Time force scale	w_{time}	15	-
Time of the feature movement to the future	δt	5	seconds

A Parameters

These two tables presents all parameters for static optimization label layout method and the hybrid optimization dynamic label layout method.

B Dataset Illustrations

Below are illustrations of the dynamic datasets labeled with hybrid optimization method. All maps are attributed to Finland, © OpenStreetMap contributors, and the GIS user community. More information on openstreetmap.org/copyright.

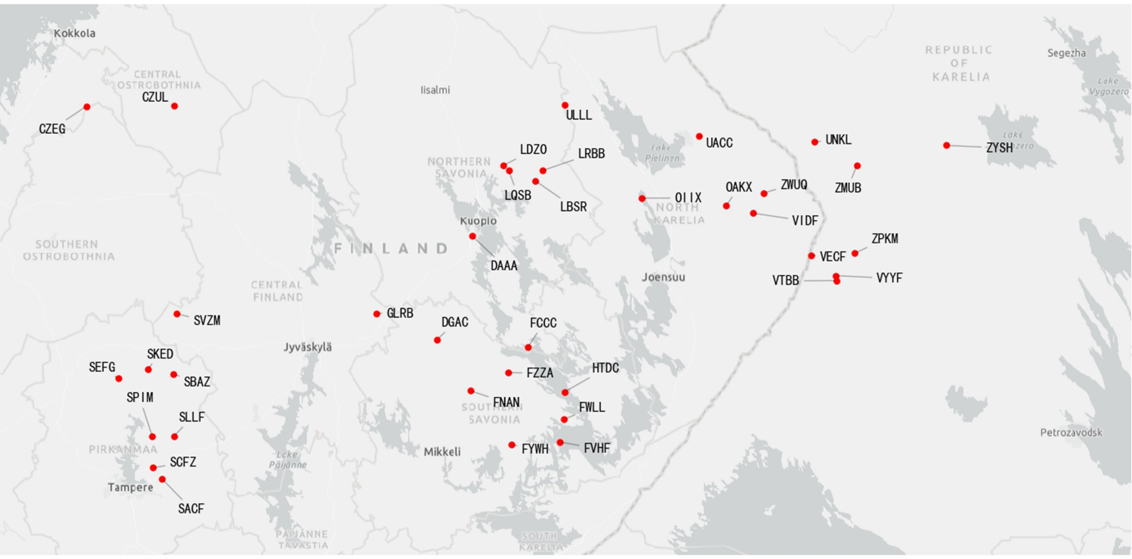


Fig. 13 Live air traffic above Finland from ADS-B Exchange, adsbexchange. com.

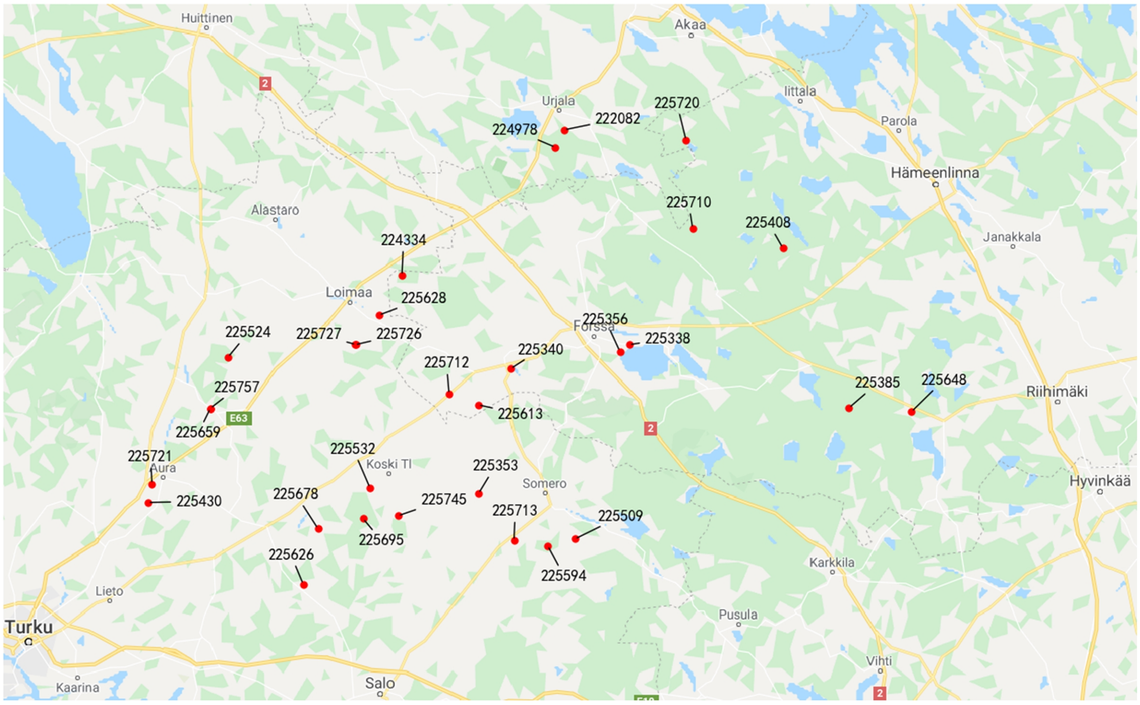


Fig. 14 Live road traffic from the Digitraffic dataset

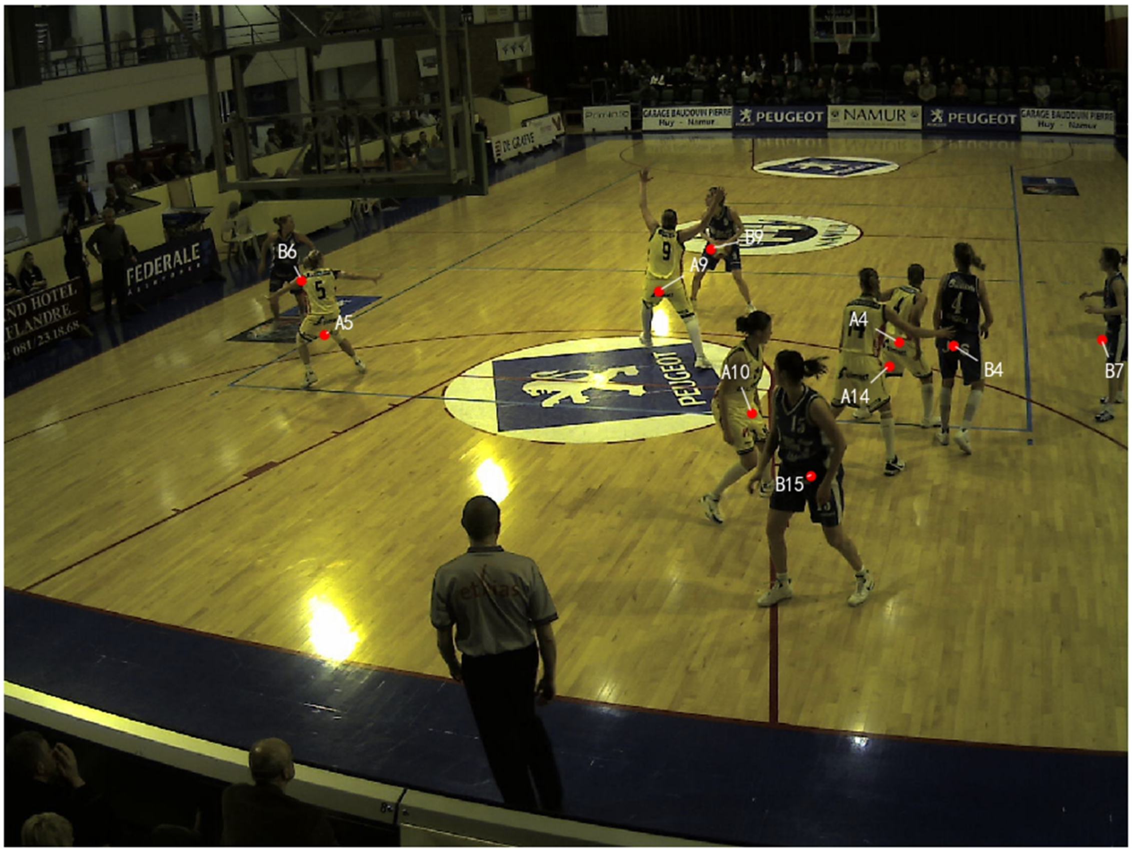


Fig. 15 Labeling of a basketball game found in the APIDIS dataset.