HKUST SPD - INSTITUTIONAL REPOSITORY

Title	MaraVis: Representation and Coordinated Intervention of Medical Encounters in Urban Marathon
Authors	Li, Quan; Lin, Huanbin; Wei, Xiguang; Huang, Yangkun; Fan, Lixin; Du, Jian; Ma, Xiaojuan; Chen, Tianjian
Source	Proceedings of the 2020 Chi Conference on Human Factors in Computing Systems (CHI'20), 2020. Article number 3376281
Version	Accepted Version
DOI	10.1145/3313831.3376281
Publisher	ACM
Copyright	© 2020 Association of Computing Machinery.

MaraVis: Representation and Coordinated Intervention of Medical Encounters in Urban Marathon

Quan Li[*]**, Huanbin Lin**[*]**, Xiguang Wei**[*]**, Yangkun Huang**[*]**, Lixin Fan**[*]**,**
Jian Du[†]**, Xiaojuan Ma**[‡]**, and Tianjian Chen**[*]

[*] AI Group, WeBank, Shenzhen, Guangdong, China
[†] CSIG Jarvis Lab, Tencent, Shenzhen, Guangdong, China
[‡] The Hong Kong University of Science and Technology, Hong Kong
{forrestli, bindylin, xiguangwei, hopyhuang, lixinfan, tobychen}@webank.com, nickdu@tencent.com, mxj@cse.ust.hk

ABSTRACT

There is an increased use of Internet-of-Things and wearable sensing devices in the urban marathon to ensure an effective response to unforeseen medical needs. However, the massive amount of real-time, heterogeneous movement and psychological data of runners impose great challenges on prompt medical incident analysis and intervention. Conventional approaches compile such data into one dashboard visualization to facilitate rapid data absorption but fail to support joint decision-making and operations in medical encounters. In this paper, we present *MaraVis*, a real-time urban marathon visualization and coordinated intervention system. It first visually summarizes real-time marathon data to facilitate the detection and exploration of possible anomalous events. Then, it calculates an optimal camera route with an arrangement of shots to guide offline effort to catch these events in time with a smooth view transition. We conduct a within-subjects study with two baseline systems to assess the efficacy of *MaraVis*.

Author Keywords

Anomaly detection; Marathon visualization; Shot chaining;

CCS Concepts

•**Human-centered computing** → **Visualization;** *Human computer interaction (HCI);*

INTRODUCTION

There is an increasing trend of using real-time Internet-of-Things and wearable devices such as smartwatches and bracelets for medical incident response in urban marathon [7, 19, 36]. Event managers and authorized healthcare personnel have to interpret a large amount of respective information and make swift and informed decisions in a short period of time in case of emergency [15]. There is thus a pressing need for a good visualization system that can provide effective representation of multivariate, time-varying data streams from different sources, as well as facilitating coordination of efforts among online and offline participating agencies, security teams, and medical services.

Conventionally, dashboard systems are used to compile heterogeneous data into one visualization to assist in rapid data absorption [4]. Although such systems have demonstrated promising performance in emergency relief in previous marathon events [4, 5, 11, 15, 31], they still face several challenges in supporting joint decision-making and operations in medical encounters. **(1) Distillation of massive, real-time information.** Traditional dashboard system tries to integrate all sorts of incoming data into one screen for providing a centralized source of information for marathon organizers to maintain real-time situational awareness of all aspects of the event, from weather to runner count [4, 15]. However, due to the constraints in screen space, only a small part of the data gets visualized to users in reality. Failing to make sense of the limited information made available in face of critical medical incidents may cause delay in response, event jeopardizing runner's health in worst-case scenarios. Hence, an effective visualization system should have the ability to distill vast data down to their essence, compressing meanings into more concise information units that can be updated seamlessly [4, 15]. **(2) Representation of runner information and detection of anomalies.** Existing dashboard systems often aggregate all types of data collected from runners such as *speed*, *location*, and *heart rate* into a simple summary view. It is difficult, even for an experienced medical staff, to identify outliers presented in such a form. It is thus necessary to preserve temporal and spatial dynamics as well as correlations across different attributes in runner data to detect anomalous events on the spot or in advance. **(3) Coordination of intervention.** Dashboard systems usually show each type of data in a designated area and require manual interactions to further inspect information on certain part of the display. According to our collaboration experts, if they notice something suspicious in the data view, they need to manipulate the control panel on a phone or tablet to zoom in to the related area on the big screen for other personnel to take a closer look. If further investigation at the scene is necessary, they need to contact and verbally direct operators of street cameras or pilots of drones to locate the incident and stream live video on the spot [4]. All of these are very labor-intensive. It would save considerable coordination efforts beforehand by streamlining the actions with different data views online and planning the camera path offline.

In this paper, we introduce *MaraVis*, a real-time urban marathon visualization system that supports better representation and coordinated intervention of medical encounters, optimizing the usage of massive marathon data streams. To

address challenge (1) and (2), the system provides an informative visual synopsis of real-time, respective marathon data and facilitates detailed exploration of possible anomalous events that are automatically identified. More specifically, we apply anomaly detection based on both the dynamics and correlation of event participants' attributes to generate a series of potential anomalous events. To address challenge (3), the system computes an optimal camera route to inspect these anomalies at the scene to support decision-making. We adapt techniques from dynamic programming and filmmaking to locate places along the route that shots with the maximum amount of unique and essential information about the event of interest can be taken. We organize these shots into an animation and compile the associated camera views to further guide the offline image capturing devices. This can minimize the communication costs among all the parties involved. The primary contributions of our work are summarized as follows:

- We extract the attribute dynamics and correlation of the heterogeneous marathon data in real-time to identify potential anomalous events, allowing event organizers to quickly follow up with critical medical incidents.
- We adapt techniques from dynamic programming and filmmaking to arrange the best camera views for delivering the essence of marathon events, which can enable online inspectors to guide offline devices to capture real-world images promptly for decision-making.
- We conduct a within-subjects study to evaluate the performance of *MaraVis*. Feedback from the users indicates the superiority of *MaraVis* over the baseline systems.

RELATED WORK

Anomaly Detection and Visualization
Anomaly detection has been extensively studied over the years, including classification-based algorithms in terms of supervised [18, 30, 46], semi-supervised [10, 28], statistics-based algorithms [3, 49], distance-based algorithms [6, 9, 17], and spectral-based algorithms [41]. For example, Donoghue et al. [34] provided a multi-step anomaly detection process which utilizes different combinations of algorithms to identify outliers and events for unsupervised athlete performance data. Although they are helpful to deal with most applications, they have a limited capability in examining the real-time data streaming collected by wearable sensor devices. In this paper, we first leverage the real-time streaming data and then adapt a distance-based algorithm to determine the potential outliers. Visualization techniques have been applied to support anomaly detection and facilitate decision-making [20, 22, 25, 48]. Dimensionality reduction techniques are also applied to understand how data distribute in a multi-dimensional space, such as MDS [21], PCA [41] and t-SNE [29]. In this paper, we augment well-established visualizations to identify anomalies by inspecting different metrics and their correlation.

Marathon Visualization and Intervention
Reviewing a marathon event mainly involves watching television [1], filming and uploading amateur footage of marathon events [14], and visualizing running trajectories. Recently, organizers track athlete in real-time through radio-frequency

chips. However, the existing race-management systems that utilize these data have been far from fully successful in visualizing a marathon event. Basdere et al. [4, 5] proposed a data visualization system *SAFE*, i.e., Situational Awareness For Events, for massive participation endurance events. It incorporates critical data into a dashboard and provides pre-event and on-site analytics to help race organizers effectively manage and oversee all event participants, monitor the dynamic locations of race participants, as well as manage health and safety resources. Our work is similar to *SAFE*, however, we focus on distilling potential anomalies by analyzing attribute dynamics and correlation. Meanwhile, we design an optimal camera view sequence and shot types to improve online observation experience and facilitate offline intervention of anomalies.

Scene Navigation and Camera Control
Scene navigation generates a guided tour in a 3D space generally constrained by a set of given landmarks [47]. Vázquez et al. [43] leveraged viewpoint entropy to quantify the amount of information that a viewpoint conveys about a specific 3D scenario. Sokolov et al. [42] generated a path that interpolates viewpoints by solving a Travel Salesmen Problem (TSP), in which the cities to traverse are the viewpoints and the cost is a combination of the Euclidean distance between the viewpoints and the visual quality along the path. Serin et al. [40] considered a semantic distance metric between good views, of which the goal is to avoid transitions between unrelated landmarks. Xie et al. [47] generated a large collection of suitable camera moves around landmarks and designed a global path which selects the best camera move for each landmark and connects them together. While we also use TSP to construct an optimal camera path, we take both the spatial and temporal factors into account to find the optimal schedule of camera views that catches events promptly and makes smooth view transitions.

Camera control in a 3D space is specified by tasks and has been addressed by many techniques [26, 27]. Blinn [8] computed viewpoints to calculate camera position and orientation by specifying on-screen properties. Visual properties in the image space have also been translated into constraints and applied on the camera freedom and solved through optimization techniques [2, 38]. However, most of them focus on the viewpoint positions, light source and moving paths. Wang et al. [44] borrowed *time remapping* and *foreshadowing* from cinematography to generate animations. Similarly, we introduce shot designs such as *panorama shot* and *dolly shot* that are commonly used in film directing to control camera views and deliver different events along the camera route.

BACKGROUND AND OBSERVATIONAL STUDY

Marathons and Marathon Planning
A marathon is a long-distance running event covering an official distance of 26.22 miles. Given the scale of participants, preparing a marathon event requires significant efforts, including the design and implementation of (1) racing course, (2) communication systems, (3) resource management [4]. Specifically, situational awareness has proven to be critical at a range of events from course rerouting (e.g., a gas leak in London Marathon 2008) to tragic medical incident (e.g., a

car accident in Klang City Marathon) [5]. During a marathon, participants may encounter medical incidents such as *falling down*, *arrhythmia*, and *heatstroke*, as well as other unforeseen incidents like *lagging too far behind*, or *running off the racing course*. In case of such race disruption events, runners could receive help from the aid stations distributed along the racing course, which are equipped with emergency supplies and connected to the command center through radio operators via intercoms and phones. It is thus critical for organizers to track runners' progression at all times to ensure timely responses.

About the Marathon Organizing Team and Penrose

To better understand existing practice of anomaly detection and intervention in marathon, we worked with a team of experts, including a chief director (E.1) from a local sports bureau, a chief designer (E.2) and a data director (E.3) from an Internet company, a smartwatch supplier (E.4), and a manager of the medical group (E.5). They jointly launched a dashboard visualization system *Penrose* (Figure 1) on April 21st, 2018 for the city annual marathon event. *Penrose* shows the geographical distribution of runners, security personnel, medical and rescue team on a large screen to assist organizers in achieving intelligent management, i.e., it visualizes spatial-temporal distributions of participants as dots and crowd density as a heat map based on data captured by smartwatches on all runners to track their locations and heart rate. Other information such as ambulance vehicles' locations are also displayed on the screen. When organizers observe race disruption events in *Penrose*, they take immediate actions to deal with the emergencies.
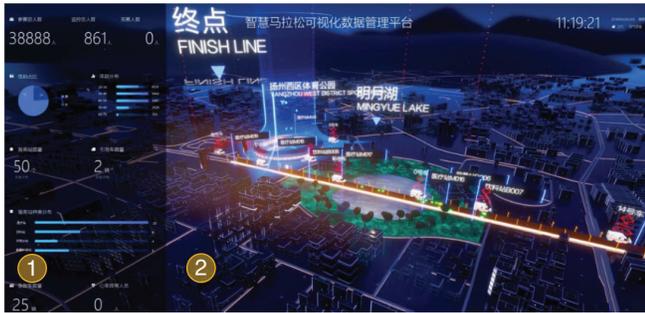

Figure 1: *Penrose* consists of (1) a dashboard and (2) a map.

Bottlenecks and Expectations

The deployment of *Penrose* at the local marathon event offers us an opportunity to observe the usage of the system, collect experts' feedback on user experiences, and identify key principles of a successful visualization system for such a context. Although *Penrose* can display essential data streams in real-time, it has several limitations in terms of *limited distillation of marathon information*, *detection and representation of anomalies*, as well as *manually coordinated intervention of medical encounters*. We interviewed E.1-5 in separate sessions to identify their primary concerns about marathon visualization and the potential obstacles in decision-making and coordination. The need for a real-time visualization system that can support intuitive representation and efficient coordinated intervention of medical encounters in marathon emerged as the key theme of the interview results. Despite individual differences in their expectations of such a system, certain requirements were expressed across the board.

R.1 Correlating Various Sources of Marathon Information. According to E.1-3, one pressing issue of *Penrose* is its way of representing event participants as separate dots. Given that the marathon data streams are heterogeneous in nature, displaying different types of data separately without considering the spatial-temporal correlation among them would hinder the observation of the whole story of the marathon event. Therefore, our experts were interested in having a well-organized representation of the key marathon information extracted from different sources of real-time data streams.

R.2 Incorporating More Data for Anomaly Detection. According to E.4-5, *Penrose* only exploits *heart rate*, *speed*, and *body motion scale* collected from smartwatches to determine whether a medical incident occurs, which could miss many potential anomalies. Medical staff in the command center found it rather difficult to judge the occurrence of possible medical incidents and their severity from such limited information. Therefore, they wanted to know whether more information could be leveraged to identify factors that may disrupt a runner's progress. For example, E.5 expressed a desire to employ raw data such as photoplethysmography (PPG) signals to jointly determine whether an anomaly occurs.

R.3 Revealing Data Dynamics and Contexts. All experts showed an interest in learning more about the contexts and dynamics of the anomalies for better sense-making. For example, some incidents may occur suddenly with no warning signs, e.g., *falling down*, while others happen some time after the initial symptoms show up, e.g., *heatstroke*. Also, *falling down* may be caused by congestion, flawed route design, or psychological status of the participant, which may result from a previous incident as well. *Penrose* currently displays anomalies in a separate list without considering the relation to the racing environment. As reported by E.5, it was thus difficult to infer the contextual information of each detected event and keep track of all runners involved in these incidents. Understanding spatiotemporal dynamics and correlations across different data attributes would help identify key marathon events on the spot or in advance and remove potential data bias in this process.

R.4 Demanding Effortless Interaction. During the interview, E.2 mentioned that when using *Penrose* to review the marathon progress or observe different areas of the racecourse, he had to manually navigate to the corresponding area on the screen through a tablet. Thus, he hoped to have more convenient interactions with the system.

R.5 Linking Online and Offline Intervention. Upon the detection of a potential medical anomalous event in *Penrose*, E.1 and E.5 had to manually dispatch this message to the nearest ambulances and security personnel for offline emergency relief, while invoking the closest cameras installed along the course or drones to capture scene images for further inspection and diagnosis. The entire process was not effective and efficient due to two reasons. First, E.1 mentioned that offline operators need to "*manually translate high-level information from him to low-level device control*", with a lot of information to clarity. Second, controlling and scheduling offline camera moves to catch anomalous events in time is quite challenging since multiple detected anomalous events may occur at the

same time so that one must simultaneously take the attributes of the detected events into consideration, i.e., event locations and occurrence time. Our experts also reported that due to the labor-intensive and time-consuming communication between online command center and offline resources (one director and two medical staffs for each reported anomalous event), they only continuously monitored athletes with particularly serious injuries and did no keep tracking other athletes with minor injuries after treatment. Therefore, they were envisioning a more automatic way of coordinating online and offline intervention.
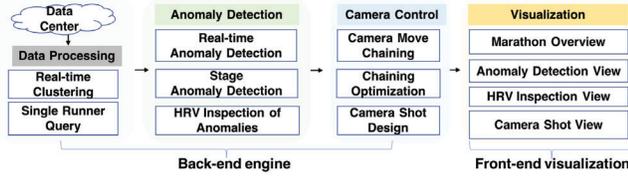

Figure 2: *MaraVis* system architecture and pipeline.

OVERVIEW OF MARAVIS

Based on the above requirements, we propose *MaraVis*, a real-time visualization system that supports better representation and coordinated intervention of medical encounters in the massive urban marathon data streams. Figure 2 illustrates the system architecture and interaction pipeline, which consists of the *back-end engine* and the *front-end visualization*. Particularly, the back-end engine comprises the *data processing*, the *anomaly detection*, and the *camera control* modules.

BACK-END ENGINE

Data Description and Processing

The first deployment of *Penrose* provides us with a dataset that records a local 2-hour 20km half-marathon event, which involves 800 staff and participants with 200 runners finishing the race and 20 service stations. We process the data in three aspects. **(1) Urban Environment.** The urban environment is simulated in a 2.5D map, which displays major roads and buildings. Along the racing course, several types of marathon facilities are set up for emergency relief, including service and aid stations. Each station has its own id and contact information and is displayed on the map. **(2) Event Participants.** The marathon runners, volunteers and event organizers are all defined as event participants. In this paper, the information of a total of 800 event participants is collected via wearable smartwatches. The dynamic movements of these participants, especially the runners are used to construct the running trajectories. Each trajectory includes a *timestamp*, *heart rate*, *peak-to-peak interval* (PPI) derived from PPG data, *speed*, *pace*, *cadence*, *distance*, *longitude*, *latitude*, and *id*. Since we are targeting a real-time marathon scenario, we replay the data streams based on their timestamps to simulate that the data is fetched in real-time when designing and implementing *MaraVis*. **(3) Real-time Clustering.** The purpose of real-time clustering is to identify the clustering center of the crowded runners. To be specific, for each timestamp t, we take the spatial positions of all runners as input and apply a density-based clustering algorithm called Mean Shift [12] to cluster these runners. In this way, we can obtain the clustering result for every timestamp for later use in the camera control module.

Anomaly Detection

The anomaly detection module of *MaraVis* consists of three parts: real-time anomaly detection, stage anomaly detection, and Heart Rate Variability (HRV) inspection of anomalies.

Real-time Anomaly Detection tracks runners' attributes in every 5 seconds and evaluates whether the value of any attributes is outside the normal range, which is determined after discussing with the medical group.

Stage Anomaly Detection is a complement of the real-time anomaly detection, since E.5 reports that "*some anomalies may occur after the initial symptoms show up*". That is, the attributes of runners should also be considered collectively within a certain time period. After discussing with E.5, we set the length of the time window as 2 minutes, which can be easily adjusted according to the practical situation. Particularly, stage anomaly detection determines potential anomalies based on the accumulated data within a predefined time window in the following four steps:

Step 1: Attribute Normalization and Averaging. We normalize all runners' attributes (i.e., *heart rate*, *PPI*, *speed*, *pace*, *cadence*, *distance*) in every 5 seconds and average the corresponding attributes. Therefore, we can attain the corresponding feature signature vector for each runner by using the mean of each normalized attribute within 2-minute period.

Step 2: Distance Matrix Construction. We compute pairwise similarity between runners by using Canberra Distance [24]: $dCan(P,Q) = \sum_n^n \frac{|P_i - Q_i|}{(P_i + Q_i)}$, where P and Q represent the feature signature vectors of two runners. We choose Canberra Distance because it is sensitive to small changes and normalizes the absolute difference of individual comparisons, benefiting to the detection of clusters and outliers [23]. Therefore, an entire distance matrix can be obtained for the following dimensionality reduction analysis.

Step 3: Dimensionality Reduction. We generate a 2D embedding of runners' attributes using t-SNE based on the attained distance matrix. We select t-SNE as the dimensionality reduction technique because it shows superiority in generating 2D projection that "*can reveal meaningful insights about data, e.g., clusters and outliers*". It is more visually interpretable than naïve eigen-analysis, and depending on the distribution, more intuitive than MDS results, which preserve global structure more at the expense of local structure retained by t-SNE [23].

Step 4: Outlier Detection. For all runners' records within each time period, we apply an outlier detection algorithm, i.e., Local Outlier Factor (LOF) [9] to find isolated data points, which is one of the most widely-used outlier detection algorithms. LOF compares the k-neighborhood density of an instance a to the k-neighborhood density of a's k-neighbors and determines whether a is an outlier, which is formally defined as: $LOF_k(a) = \frac{\sum_{b \in NN_k(a)} \frac{lrd_k(b)}{lrd_k(a)}}{k}$ and $lrd_k(a)$ is a's local reachability density, which is defined as: $lrd_k(t) = (\frac{\sum_{s \in NN_t(k)} dist_k(t,s)}{k})^{-1}$, where $dist_k(a,b) = max(d_k(b), d(a,b))$ indicates the reachability distance between a and b, i.e., the Euclidean distance between a and b but no smaller than b's k-distance $(d_k(b))$.

The parameter k can be arbitrarily determined based on users' experience regarding a given dataset [48]. Outliers with the LOF score larger than 1 indicate an isolated instance.

HRV Inspection of Anomalies facilitates medical groups to leverage more information to analyze the previously detected outlier runners. We derive multiple HRV metrics from the original PPG signals collected from the wearable smartwatches, including time-domain, frequency-domain and geometric metrics [37, 39]. These metrics are summarized as diagnostic charts for further medical inspection. Furthermore, *MaraVis* implements a deep learning model to detect Atrial Fibrillation (AF) beats in HR signals [13]. The model utilizes a Long-Short Term Memory (LSTM) model to capture the time-series features of PPI signal and is trained offline using a widely-used public dataset, MIT-BIT Atrial Fibrillation Database [32, 35] (accuracy of 99.77% with blindfold validation).

Camera Control

Having determined a set of events, our next task is to compute an optimal sequence for the entire flyby by chaining each event in alternating order, to maximize the total quality (i.e., minimize the time) of the resulting trajectory.

Camera Move Chaining. We propose a method to sequence camera views to present the detected events during a marathon match, which has two aspects. First, we need to schedule the camera moves to catch those anomalous events in time and make view transition as smooth as possible. Second, the visualization should provide a guideline for invoking offline camera devices. We assume that multiple video sources have been set up and our camera views can be attained directly so we do not need to consider practical constraints such as aerial vehicle navigation or camera location. With this assumption, we formulate this problem in a solvable manner. Xie et al. [47] proposed a technique for creating and chaining camera moves and solved it as a Travel Salesman Problem (TSP). Similarly, we also need to plan the route and view transitions. However, different from their task, we do not necessarily need a complex evaluation function, whereas we should consider the temporal factors since we focus on real-time anomaly detection.

We formulate this problem as a combinatorial optimization problem to find a camera view sequence with minimal cost, which corresponds to a camera move with the highest quality. Particularly, given a set of data points with 2D coordinates and 1D timestamp, we construct a coordinate system in a 3D space by selecting the data point with the earliest timestamp as the origin, and taking the relative distances of all the other data points to the origin as their coordinates. Therefore, each data point can be represented as (x, y, t). Furthermore, we integrate the spatiotemporal dimension by introducing a speed assumption that comes from a real camera move, i.e., the speed of a flying drone or sliding rail. With this assumption, we unify the measuring approach and generalize mathematical formulations for our task of deciding which point to be visited next, which can be formally defined in the following two ways (Figure 3): **(1) Algorithm 1.** We formulate this problem as a 3D routing problem by leveraging a 3D Euclidean distance $\varphi = \Delta x^2 + \Delta y^2 + \Delta t^2$ as the measure, i.e., a 3D version of TSP with the constraint of $\sum \varphi_i = (x_0 - x_n)^2 + (y_0 - y_n)^2 + (t_0 -$

$t_n)^2 + \sum_{i=1}^{n}((x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 + (t_i - t_{i-1})^2)$. **(2) Algorithm 2.** Since our goal is to capture anomalous events in a timely and smooth manner, the problem can be also formulated as "*always move to the event that we can arrive on time*". This is equivalent to minimize the cumulative gap between spatial and temporal distance, which is essentially a combinatorial optimization problem with the measure of $\varphi = \Delta x^2 + \Delta y^2 - \Delta t^2$ and the constraint of $\sum \varphi_i = (x_0 - x_n)^2 + (y_0 - y_n)^2 - (t_0 - t_n)^2 + \sum_{i=1}^{n}((x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 - (t_i - t_{i-1})^2)$.
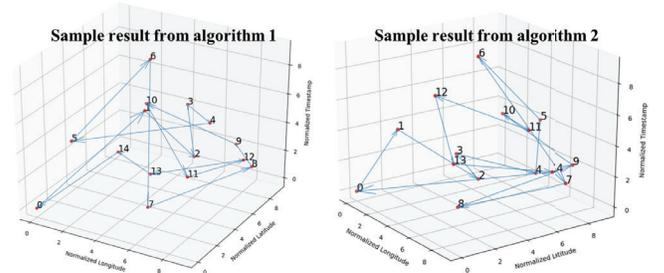

Figure 3: Camera view chaining generated by Algorithm 1 and 2, respectively. Three axes represent the normalized longitude, normalized latitude and normalized timestamp, respectively.

Chaining Optimization. In the practical application, we find the detected events usually occurring within a certain area and chaining camera views for all the events takes too much time and is also unfriendly. To resolve this issue, we cluster event points locally. Therefore, we can generate the subproblem sets by breaking the problem above into two parts: (1) Inside each local part for a single cluster, we use either Algorithm 1 or 2 to perform the camera scheduling. (2) For inter-cluster camera scheduling, we simply consider each cluster as a single point with the center of the cluster as the coordinate and then apply K-means clustering to the event points based on their 3D coordinates. Therefore, all the event points can be divided into K groups. The parameter K can be derived by observation of the historical records. Here, we set 4 as the value of K.

Camera Shot Design. We borrow the idea in filmmaking in which the expression of emotions and ideas can be enhanced by employing appropriate camera shot designs to further design the camera shot along the camera route. To be specific, we employ different camera shots to depict different events by using the following parameters in animations, including focal point c_{center}, shot duration c_d, camera distance c_h, pitch c_p, and camera orientation c_o to control the camera motion.

We report how each camera shot could be mapped into a marathon [44]: **(1) Normal shot.** It tracks the objects from the side. We use it to track the participants. **(2) Panorama shot.** It slowly moves the camera over a landscape with a wide-open space. We use it to switch the view from the surroundings to the starting point of the marathon race. **(3) Following shot.** The camera follows the subject being filmed around. We apply it to track runners from the rear. **(4) Dolly shot.** It moves the camera toward or away from a subject while filming. We use it to observe anomalies by sliding and panning the camera, and **(5) Split-screen.** It divides the screen and shows several images simultaneously to present a seamless view of reality. We use it to watch two or more anomalies simultaneously.
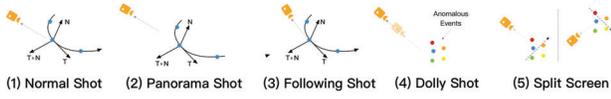
Figure 4: Camera motion for different shot types.

We apply several types of camera motions to simulate each shot type. As the most basic filmmaking technique, we use **normal shot** to track the runners. As shown in Figure 4, we leverage a third-person view as the camera direction in the three-point perspective and the center of the runner clusters as the focal point c_{center} of the camera. The camera orientation c_o of the normal shot is defined by the FrenetSerret frame [16] as $c_o = \frac{1}{\sqrt{3}}d(T + N - T * N)$, where the constant d denotes the distance between the camera and the focal point, while T and N denote the tangent unit vector and normal unit vector of the trajectory, respectively. To gain a full perspective of the surroundings, we apply **panorama shot** at the beginning to achieve a transition from the global view to the beginning of the event. We set the camera distance c_d ensuring that it is high enough to cover the areas for a panoramic view. In **following shot**, the camera tracks the subject from the rear (Figure 4). When anomalous events occur, we shift to **dolly shot** by pushing or pulling away the camera (i.e., changing the parameters of camera distance c_d and pitch c_p) to better observe and understand the emergencies. We adopt **split-screen** for displaying multiple camera views. When facing simultaneous events, the system divides the screen and allocates different sessions for monitoring individual anomaly.

FRONT-END VISUALIZATION

We develop four visualizations (Figure 5) that allow the marathon data to be easily inspected: a marathon overview presenting event progress, an anomaly detection view illustrating the real-time and stage anomaly detection, a camera shot view facilitating the anomaly exploration and coordinated intervention, and an HRV inspection view showing medical metrics of the anomalous runners.

Figure 5: The full *MaraVis* system contains four coordinated views: (1) a marathon overview, (2) an anomaly detection view, (3) a camera shot view, and (4) an HRV inspection view.

Marathon Overview

An important and innovative enhancement to *Penrose* is the marathon overview which provides a representation of runner distribution over the racecourse (**R.1**) (Figure 6). Each ring represents a geographical distance (e.g., 4km, 8km, and 12km). The curve along the ring indicates the number of runners that fall in the corresponding geographical scale (Figure 6(1)). The counterclockwise direction indicates the time segments (20 minutes per axis). The box-plot along each time axis shows the distance distribution of runners (Figure 6(2)). Red points along each ring represent the time for achieving a new distance milestone. For example, the first runner takes 88.08 minutes to finish 20km (Figure 6(3)). The rolling information billboard updates currently the fastest runner (Figure 6(4)).
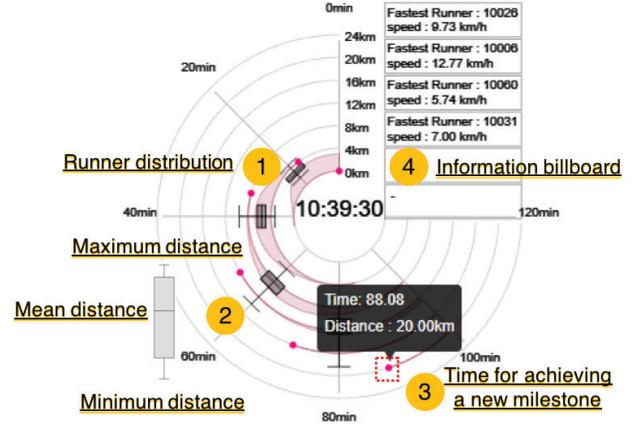

Figure 6: Visual encodings in the marathon overview design.

Anomaly Detection View

To support both real-time and stage anomaly detection, we integrate three components into the anomaly detection view, namely, an area with box plots, a scatter contour plot and an abnormal ID list (Figure 7(1-3)) (**R.2**). The real-time area with box plots shows the results of the real-time anomaly detection. To be specific, the box plot shows the distribution of sum of normalized attributes (i.e., *speed*, *PPI*, *pace*, *distance*) and the width of the area at the corresponding timestamp indicates the range of sum of normalized attributes (Figure 7(A)). The white points represent the detected anomalous runners in real-time. The scatter contour plot shows the spatial distribution of all the runners in the corresponding time window using t-SNE based on their attributes, which aggregates the multivariate runner attributes into subsets exhibiting a certain similarity. Each point in this view represents a runner whose anomaly degree is encoded by its size, which is calculated by previously mentioned LOF algorithm. The detected anomalies are encoded as red nodes (Figure 7(D)). The values of their attributes are also rendered as red lines in the area with box plots (Figure 7(B)), which enables to track the attribute changes in the corresponding time window. To enhance the anomaly analysis, we render a contour map to reveal areas with different densities. Intuitively, a normal point tends to lie in high-density areas where many other points have similar behaviors. We use the kernel density estimation (KDE) to define the density at runner's position. From this view, we can also provide another perspective for anomaly diagnosis according to the inconsistent measurements of similar runners. For example, some anomalous nodes (high LOF score) may be grouped in the low-density contour, which could represent a rare medical

Figure 7: Anomaly detection view and cases of detecting inconsistent attribute values and abnormal AF beats in HR signals.

encounter that deserves attention. Along each vertical axis (Figure 7(C)), we plot the distribution of runner's attributes from two consecutive time windows, which supports a detailed exploration of the changes before and after the predefined time window. When users hover on a specific node, links would connect to the corresponding value of attributes in the vertical axis. The detected anomalous runners' ids would be listed, with a tooltip encoding its occurrence number (Figure 7(E)).

Camera Shot View

We design a camera shot view that facilitates swift response through view shift by using an optimal chaining route and shot designs (**R.3-5**) (Figure 5(3)). This view visualizes event information, including the event object (normal runners encoded as green dots and anomalies as red dots), occurrence time and location. Particularly, it first applies a panorama shot at the beginning of the tracking to present the event overview, and then switches to a normal shot to capture the runners. When the event begins, the camera uses a following shot. As shown in Figure 8, the runners' clustering result is obtained in real-time, which is visualized by a yellow circle, with the center always at the focal point of the camera. When detecting an anomaly, a dolly shot is applied to shot the situation around the event. Meanwhile, the nearest aid station to the event would be automatically recommended. When there is a set of events, *MaraVis* computes an optimal route that connects each of them by using the proposed chaining algorithm.
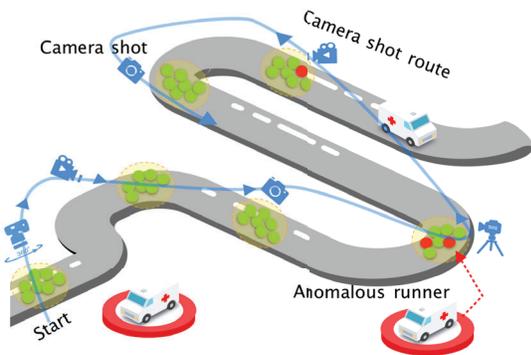

Figure 8: Camera move chaining and shot designs.

Handling Simultaneous Events. During the race, we observed around 2-4 events occurring simultaneously at different points of time. Currently, only one camera is employed in *Penrose* and in our system to capture anomalies and the previous camera shot algorithms are under this premise. For example, Algorithm 2 captures anomalous events by minimizing the cumulative gap between the spatiotemporal distance of events (Figure 3). If multiple anomalies, say, A and B occur exactly at the same time in the same place, the algorithm can shoot A and B simultaneously; however, if A and B occur simultaneously but in different places that are far apart, the algorithm determines its order based on their spatial and temporal distances, e.g., first shooting A and then B, and this may lead to a slight delay when shooting B. To handle simultaneous events, we adopt *split-screen* by applying multiple cameras for displaying multiple views. That is, when facing simultaneous events, the system divides the screen and allocates different sessions for monitoring individual anomaly, as shown in Figure 5(3).

HRV Inspection View

To help professional medical groups make accurate diagnostic decisions and preparedness, we include several additional views that link to runner selection made in the anomaly detection view (**R.2**). After a selection is made, HRV metrics extracted from the runner's PPI data are displayed. We adopt conventional visualization commonly used in medical reports to display these metrics like heart rate profile, PPI timeline, PPI Poincare chart, PPI distribution chart, HR (diff) distribution and PPI frequency spectrum (Figure 5(4)).

EVALUATION

In the dataset, we identify 216 incidents of four main types of events: inconsistent attribute values (e.g., relatively high or low heart rates) (19 incidents), abnormal or missing trajectories (15 incidents), abnormal AF beats in HR signals (30 incidents), and heart rate value is 0 (e.g., due to loosened bracelet) (152 incidents). However, in marathon scenarios, no standardized dataset nor ground truth labels exist to evaluate the accuracy of algorithms for detecting anomalies [4, 15], and we do not have the actual anomaly labels. Moreover, a real-time system emphasizes more on its efficiency in supporting fast online and offline decision-making. Hence, in this work, we introduce how *MaraVis* facilitates detecting inconsistent attribute values and abnormal AF beats in HR signals, and then focus on the evaluation of the usefulness and efficacy of *MaraVis* instead of the accuracy of anomaly detection methods.

As shown in Figure 7, in the first 2-minute period, *MaraVis* generates a list of anomalies by combining the results from real-time anomaly detection (2 incidents) and stage anomaly detection (5 incidents). When we click on one of the abnormal runner id, namely, 10056, and identify that in the 2-minute period, 10056's sum of the normalized attributes (i.e., sign index) is initially high and then gradually decreased, unlike others' performance. In addition, through the HRV inspection view, it can be witnessed that 10056's heart rate drops to 0 at a later stage, which is a potential anomaly. Similarly, the system detects an inconsistence occurs to 10028's sign index and heart rate data and determines that 10028 is a potential anomaly. Another case shows the sign index of 10044 fluctuates significantly in the middle stage and AF appears in the heart rate signal, which requires an offline inspection.

Experiment Design

We adopt a four-layer taxonomy [45] to evaluate *MaraVis*. Specifically, we conduct a within-subjects study to systematically assess the informativeness, the effectiveness in facilitating decision-making, usability, visual design, as well as camera view chaining algorithms.

Participants. We recruit 18 volunteers (9 females, 9 males, age: 28 ± 3.03) from the local marathon organization. In particular, we choose the participants with running and wearing sensible smartwatches experiences, for which they could provide us more comprehensible insights.

Figure 9: The primitive version of *MaraVis* consists of (1) a marathon overview, (2) an anomaly detection view, (3) a camera shot view and (4) a list of anomalous runners.

Experiment Procedure. We compare *MaraVis* with two alternative systems. One is *Penrose* developed by our collaboration experts (the baseline). The other one is a primitive version of *MaraVis*. The primitive system divides the detected anomalies into *tachycardia* and *bradycardia* by using a scatter plot visualization (Figure 9(2)). The differences between the primitive and full version lie in: (1) the full version provides more data attributes to support both real-time and stage anomaly detection and the primitive version only detects anomalies based on PPI in real-time; (2) the full version makes use of several visual cues and hints to illustrate the results from both real-time and stage anomaly detection; (3) the simplified version uses statistical charts to present the marathon overview (Figure 9(1)), while the full version leverages a circular box-plot design (Figure 5(1)) to encode relevant marathon data. To minimize the ordering and learning effect, we counterbalance the three systems in comparison with the three scenarios.

Informativeness. The information is *(Q1)* easy to access, *(Q2)* rich and *(Q3)* sufficient to determine an anomaly.
Decision Making. *(Q4)* The information provided for anomaly detection is trustworthy. *(Q5)* The system was helpful to observe an anomaly and *(Q6)* helps make informed decisions according to the visualization.
Visual Design. *(Q7)* The anomaly detection design is intuitive and *(Q8)* helps understand the reasons behind anomalies. *(Q9)* The marathon overview design is intuitive and *(Q10)* helps me access the overview of the marathon event.
Usability. The system was *(Q11)* easy to learn, *(Q12)* convenient to use, and *(Q13)* recommended to other scenarios.
View Chaining. *(Q14)* The event capture is relatively conflicting. *(Q15)* The view transitions are smooth and coherent. *(Q16)* The animation follows a reasonable route.

Table 1: Our questionnaire consists of five aspects: informativeness (Q1-3), decision making (Q4-6), visual design (Q7-10), usability (Q11-13), and view chaining (Q14-16).

We conduct the experiment in four sessions. In the first session, participants are briefed about the background, purpose and procedure of the experiment. Each following session lasts around 10 minutes and one of the three systems is presented and tested. Each participant is required to conduct two tasks with the provided system. The first task is to observe and track the potential anomalies. The second task is to evaluate the camera move chaining algorithms. Participants are also asked to think aloud their ideas when performing all the tasks. After finishing all the tasks with a particular system, participants are required to complete a questionnaire with 7-point Likert scale questions derived from the existing literature [33] (Table 1).

We propose the following hypotheses: **H1.** The primitive and full versions of *MaraVis* perform better than the baseline in terms of informativeness. Specifically, *MaraVis* systems enjoy their advantages on information accessibility (*H1a*), richness (*H1b*), and sufficiency (*H1c*) compared with the baseline. **H2.** The primitive and full versions of *MaraVis* are better than the baseline in facilitating decision-making. In particular, *MaraVis* systems provide more confidence (*H2a*), assistance (*H2b*), and intervention (*H2c*) compared with the baseline. **H3.** The full version is more informative than the primitive version. Specifically, the information accessibility (*H3a*), richness (*H3b*), and sufficiency (*H3c*) of the full version are better than that of the primitive version. **H4.** The full version performs better than the primitive version in facilitating decision-making in terms of confidence (*H4a*), assistance (*H4b*), and intervention (*H4c*). **H5.** The primitive version is considered more intuitive (*H5a*), easier to comprehend (*H5b*), learn (*H5c*), and use (*H5d*), and thus is better recommended (*H5e*) compared with the full version. We also propose two hypotheses to compare different camera chaining methods. Specifically, we compare Algorithm 1 and 2 with a baseline method that shots events one after another: **H6.** Both Algorithm 1 and 2 are better than the baseline in terms of less conflicting (*H6a*), smoother and more coherent (*H6b*), and follow a more reasonable route (*H6c*)). **H7.** Algorithm 2 is preferred over 1, i.e., Algorithm 1 is considered more conflicting (*H7a*), less smooth and coherent (*H7b*), as well as follows a less reasonable route (*H7c*).

| | | Mean | | | SD | | | B VS. P | B VS. F | P VS. F | | | | |
		B	P	F	B	P	F	*p*	*p*	*p*	*df*	*F*	*Sig.*	η^2
Informativeness	*accessibility*	3.22	6.06	6.11	1.77	.87	.96	0.00	0.00	1.00	2.00	28.93	0.00	.63
	richness	2.83	5.50	6.50	1.38	.86	.71	0.00	0.00	**.001**	2.00	65.96	0.00	.80
	sufficiency	2.94	5.67	6.28	1.66	1.08	.89	0.00	0.00	.26	2.00	37.57	0.00	.69
Decision Making	*confidence*	3.28	6.11	6.22	1.67	1.02	.94	0.00	0.00	1.00	2.00	33.81	0.00	.67
	assistance	3.39	6.00	6.17	2.33	.84	1.04	.002	.001	1.00	2.00	17.69	0.00	.51
	intervention	4.50	6.11	6.28	1.79	.83	.67	.01	.004	.81	2.00	11.97	0.00	.41
Visual Design	*intuitiveness*[1]	/	**5.61**	5.44	/	1.04	1.15	/	/	.64	1.00	.23	.64	.01
	comprehension[1]	/	5.78	5.78	/	1.31	1.52	/	/	1.00	1.00	0.00	1.00	0.00
	intuitiveness[2]	/	4.94	**5.72**	/	.99	1.07	/	/	**.009**	1.00	8.77	.009	.34
	comprehension[2]	/	5.50	5.89	/	1.09	1.13	/	/	.26	1.00	1.35	.26	.07
System Usability	*easy to learn*	5.50	6.00	**5.33**	1.34	1.28	1.24	.68	1.00	.19	2.00	1.59	.22	.09
	easy to use	5.39	5.83	5.89	1.58	1.04	1.08	.97	.90	1.00	2.00	.92	.41	.05
	recommendable	4.28	5.89	6.17	1.60	.96	1.04	.007	.002	.41	2.00	14.08	0.00	.45

Table 2: Repeated measures ANOVA of baseline (B), primitive (P), and full (F) version on informativeness, decision-making, visual designs, and usability ([1] and [2] indicate the result of the anomaly detection design and marathon overview design, respectively).

Results and Analysis

We report participants' quantitative ratings and feedback on *informativeness*, *decision-making*, *visual design*, and *usability*, as well as *view chaining*. We run repeated measures ANOVA on each questionnaire item, followed by the Bonferroni post-hoc test on measures with statistically significant differences.

Informativeness. The primitive and full versions of *MaraVis* receive significantly higher scores in all metrics of informativeness than the baseline (Table 2). Assessing information is significantly easier in the full and primitive versions than the baseline (**H1a supported**). No significance has been found between the full and primitive version in the Bonferroni post-hoc test ($p = 1.0$, **H3a rejected**). The information provided by the full and primitive versions is significantly richer than the baseline (**H1b supported**). We also observe a significant difference between the full and primitive version with $p < .01$, **H3b supported**. "*The full version leverages more data which can screen out potential anomalies, and this is in line with the principle of ensuring the participants' safety to the greatest extent*" (P12, male, age: 31). The information offered by the full and primitive versions is shown to be sufficient in determining an anomaly, compared with the baseline (**H1c supported**). No significant difference has been found between the full and primitive version ($p = .26$, **H3c rejected**).

Decision-Making. Participants report significantly higher confidence in anomaly detection using the two versions compared with the baseline (Table 2, **H2a supported**). No significant difference is identified between the full and primitive version in the Bonferroni post-hoc test ($p = 1.0$, **H4a rejected**). Participants also report that the full and primitive versions provide significantly more assistance than the baseline (**H2b supported**). No significant difference has been found between the full and primitive version in the Bonferroni post-hoc test ($p = 1.0$, **H4b rejected**). When asking the participants whether *MaraVis* helps make informed decisions, results show that both versions are significantly better than the baseline (**H2c supported**). However, no significant difference exists between the two versions ($p = .81$, **H4c rejected**).

In summary, results on informativeness and decision-making demonstrate that *MaraVis* provides more accessible, rich, and sufficient information to users. "*The baseline only shows anomalies without any details*" (P12, male, age: 31). Particularly, the full version enhances information richness while still maintaining good accessibility. However, the two systems do not differ significantly in facilitating decision-making. Participants comment that they lack ground-truth data to confirm the systems' performance, although they acknowledge that the full version plays a better role in anomaly detection as it is important "*not to miss any suspicious case*". "*It is better to know whether the detected anomalies are really abnormal so I can know anomaly error rate*" (P10, female, age: 29).

Intuitiveness and Comprehension. Different from our hypothesis, the anomaly detection design in the primitive version is not more intuitive or comprehensible than that in the full version (Table 2). However, the marathon overview design in the full version is more intuitive than that in the primitive version, although there is no significant difference regarding the comprehension of the design (**H5a and H5b rejected**). Participants report that they can understand the idea behind the anomaly detection design in the full version. "*Monitoring real-time metrics and then comparing with other runners accord with the idea of anomaly detection*" (P6, male, age: 26). "*The marathon overview centralizes the event essence into one design so that I don't need to locate different information on the screen*" (P2, male, age: 28).

Learn, Use and Recommendable. No significant difference exists in terms of easy to learn and use among the three systems (Table 2). The Bonferroni post-hoc test also shows that there is no significant difference between the full and primitive version in terms of easy to learn and use ($p = .19$ and 1.00) (**H5c and H5d rejected**). "*With some introduction to each view, it is easy for me to develop a path through the system for observation*" (P14, female, age: 25). However, participants are more willing to recommend the two versions than the baseline to other scenarios. "*MaraVis is very useful because it provides a novel and highly automatic way to uncover potential anomalies*" (P6, male, age: 26). "*MaraVis significantly expands from a single-page dashboard to a multiple-feature system*" (P15, female, age: 26). No significant difference exists between the primitive and full versions ($p = .41$) (**H5e rejected**).

View Chaining. Algorithm 2 is regarded as the best in terms of view chaining (Figure 10). Participants report that there are fewer conflicts of Algorithm 1 and 2 compared with the baseline (**H6a supported**). We do not observe a significant difference between Algorithm 1 and 2 in a Bonferroni post-hoc test ($p = 1.00$, **H7a rejected**). However, a significant

difference regarding smooth and coherent exists between Algorithm 1 and 2, while the baseline is in the middle (**H6b partially supported**). We also observe a significant difference between Algorithm 1 and 2 with $p < .01$, **H7b supported**. "*The chaining of Algorithm* 1 *moves fast and brings dizziness*" (P13, female, age: 26). A significant difference exists regarding following a reasonable route between Algorithm 1 and 2, while the baseline is in the middle (**H6c partially supported**). There is also a significant difference between Algorithm 1 and 2 with $p < .01$, **H7c supported**. "*I feel the chaining of Algorithm* 2 *is on schedule*" (P8, female, age: 30).
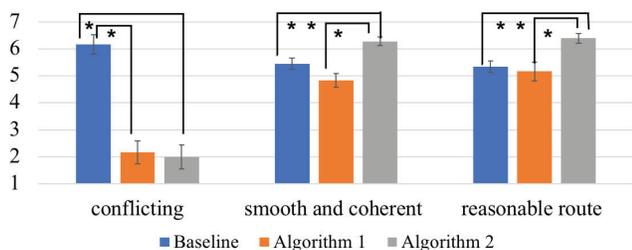

Figure 10: Means and standard errors of *Baseline*, *Algorithm 1*, and *Algorithm 2* on conflicting, smooth and coherent, as well as following a reasonable route of camera view chaining on a 7-point Likert scale (*: $p < .01$, **: $.01 < p < .05$).

Quantitative Evaluation of View Chaining. To quantitatively measure the differences between the three chaining methods, we study the time efficiency of baseline as well as algorithm 1 and 2 with an experiment shown in Figure 3. For the same series of 14 potential anomalies, we calculate both the time and distance interval between two consecutive events in the shooting order determined by each method. We obtain the distance deviation per unit time as 33.3, 39.6, and 31.5 meters of baseline, algorithms 1 and 2, respectively, indicating that algorithm 2 presents the smoothest transition, which is consistent with what we find in the user study.

DISCUSSION AND LIMITATION
We conduct a semi-structured interview with E.1-5 to collect feedback on generalizability, scalability, and learnability.

Generalizability. When discussing with the experts about which component(s) of *MaraVis* can be directly deployed for other marathons and which one(s) need customization to further explore the potential of our system, three insights are identified: (1) Visualization. E.3 mentions that design like 2.5D map, scatter plot, box plot employed in *MaraVis* are quite generic. Other marathon teams should be able to understand them without much training. (2) Data. E.3 suggests that *MaraVis* has covered most of the common data coming in and out of a marathon control center. Although we showcase the system by using one local marathon data, we can easily incorporate data from other events by simply reformatting and reconfiguration. E.4 adds that if we integrate new sensing data according to the actual sensor devices deployed at each event in a plug-and-play fashion, the system would be very adaptable. (3) Algorithms. E.1 finds it helpful that *MaraVis* has well-defined APIs for each module. With this setup, we can replace backend algorithms such as the one for anomaly detection with other methods whenever necessary. Furthermore, as indicated by the experts in the interview, our system can be extended to other healthcare scenarios such as city-scale ambulance dispatching and tracking upon emergency calls.

Scalability. Since we only test our system with real-time data from 800 participants, the experts point out we need to handle scalability issues with visualization when dealing with a larger data stream. In response to them, we propose one possible measure: data sampling. That is, the data reporting time of smartwatches is not uniform, and it is feasible that we only visualize the data available in each timestamp.

Learnability. Our experts recommend a potential means to lower the learning cost of *MaraVis* by suggesting that we could first familiarize new users with the preliminary version ($Mean = 6.00$, $SD = 1.28$) of the system, which incorporates similar data views of the full system ($Mean = 5.33$, $SD = 1.24$) while imitating the conventional practices of our target users. In this way, users would not be overwhelmed and intimidated by working with something unfamiliar. Instead, they have time and space to gain new knowledge incrementally, build trust, and ultimately embrace innovation. Meanwhile, we can take the chance to test the initial designs for later use.

Limitation. First, we do not consider the individual difference in anomaly detection. In the user study, some participants report that the maximum heart rate a person can achieve varies from individual to individual. We may include physical examination data of runners to personalize their information for more accurate analysis. Second, there is no ground truth to verify the error rate of anomaly report generated by our system, since we are currently confined to real-time data streamed by smartwatches and lack feedback regarding the actual intervention of anomaly offline. Third, we focus on camera chaining and shot design that can be well captured in simulation in the scope of this work, so we have not applied and tested in real videos. Besides, the proposed camera move chaining and shot designs are based on an ideal situation, e.g., camera views could be easily attained and no other obstacles exist when shooting. Given the complexity of actual environments like racecourse conditions that may lower system usability, one possible solution is to create and chain camera moves for Quadrotor videography [47] for offline anomaly intervention.

CONCLUSION AND FUTURE WORK
In this paper, we introduce a real-time marathon visualization system *MaraVis* that supports better representation and intervention of medical encounters. It identifies potential anomalies for organizers to quickly follow up with critical medical incidents and calculates an optimal camera route to catch the events in time, as well as make a smooth view transition. In the future, we will deploy our system in a real-world marathon for field studies. Also, we will extend the camera shot design to a formal and comprehensive design study.

ACKNOWLEDGMENTS

REFERENCES

[1] Edward Anstead, Steve Benford, and Robert Houghton. 2016. MarathOn multiscreen: group television watching and interaction in a viewing ecology. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*. ACM, 405–417.

[2] William H Bares, Somying Thainimit, Scott McDermott, and C Boudreaux. 2000. A model for constraint-based camera planning. In *Proceedings of AAAI spring symposium on smart graphics*. 84–91.

[3] Vic Barnett and Toby Lewis. 1974. *Outliers in statistical data*. Wiley.

[4] Mehmet Basdere, Gabriel Caniglia, Charles Collar, Christian Rozolis, George Chiampas, Michael Nishi, and Karen Smilowitz. 2019. SAFE: A Comprehensive Data Visualization System. *Interfaces* (2019).

[5] Mehmet Başdere, Colleen Ross, Jennifer L Chan, Sanjay Mehrotra, Karen Smilowitz, and George Chiampas. 2014. Acute incident rapid response at a mass-gathering event through comprehensive planning systems: a case report from the 2013 Shamrock Shuffle. *Prehospital and disaster medicine* 29, 3 (2014), 320–325.

[6] Stephen D Bay and Mark Schwabacher. 2003. Mining distance-based outliers in near linear time with randomization and a simple pruning rule. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 29–38.

[7] Lauren C Benson, Christian A Clermont, Eva Bošnjak, and Reed Ferber. 2018. The use of wearable devices for walking and running gait analysis outside of the lab: A systematic review. *Gait & posture* 63 (2018), 124–138.

[8] Jim Blinn. 1988. Where am I? What am I looking at?(cinematography). *IEEE Computer Graphics and Applications* 8, 4 (1988), 76–81.

[9] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. 2000. LOF: identifying density-based local outliers. In *ACM sigmod record*, Vol. 29. ACM, 93–104.

[10] Yunqiang Chen, Xiang Sean Zhou, and Thomas S Huang. 2001. One-class SVM for learning in image retrieval.. In *ICIP (1)*. Citeseer, 34–37.

[11] George Chiampas and Carrie A Jaworski. 2009. Preparing for the surge: perspectives on marathon medical preparedness. *Current sports medicine reports* 8, 3 (2009), 131–135.

[12] Dorin Comaniciu and Peter Meer. 2002. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis & Machine Intelligence* 5 (2002), 603–619.

[13] Oliver Faust, Alex Shenfield, Murtadha Kareem, Tan Ru San, Hamido Fujita, and U Rajendra Acharya. 2018. Automated detection of atrial fibrillation using long short-term memory network with RR interval signals. *Computers in biology and medicine* 102 (2018), 327–335.

[14] Martin D Flintham, Raphael Velt, Max L Wilson, Edward J Anstead, Steve Benford, Anthony Brown, Timothy Pearce, Dominic Price, and James Sprinks. 2015. Run spot run: capturing and tagging footage of a race by crowds of spectators. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 747–756.

[15] Taylor Hanken, Sam Young, Karen Smilowitz, George Chiampas, and David Waskowski. 2016. Developing a data visualization system for the Bank of America Chicago Marathon (Chicago, Illinois USA). *Prehospital and disaster medicine* 31, 5 (2016), 572–577.

[16] Gabriel A Hare, Keith Mertens, Matias Perez, ROY Neeloy, and David Holz. 2018. Non-linear motion capture using Frenet-Serret frames. (Jan. 2 2018). US Patent 9,857,876.

[17] Ville Hautamaki, Ismo Karkkainen, and Pasi Franti. 2004. Outlier detection using k-nearest neighbour graph. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, Vol. 3. IEEE, 430–433.

[18] Simon Hawkins, Hongxing He, Graham Williams, and Rohan Baxter. 2002. Outlier detection using replicator neural networks. In *International Conference on Data Warehousing and Knowledge Discovery*. Springer, 170–180.

[19] Christina R Hollis, Rachel M Koldenhoven, Jacob E Resch, and Jay Hertel. 2019. Running biomechanics as measured by wearable sensors: effects of speed and surface. *Sports biomechanics* (2019), 1–11.

[20] Andreas Kind, Marc Ph Stoecklin, and Xenofontas Dimitropoulos. 2009. Histogram-based traffic anomaly detection. *IEEE Transactions on Network and Service Management* 6, 2 (2009), 110–121.

[21] Joseph B Kruskal. 1964. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika* 29, 1 (1964), 1–27.

[22] Pavel Laskov, Konrad Rieck, Christin Schäfer, and Klaus-Robert Müller. 2005. Visualization of anomaly detection using prediction sensitivity.. In *Sicherheit*, Vol. 2. 197–208.

[23] Quan Li, Kristanto Sean Njotoprawiro, Hammad Haleem, Qiaoan Chen, Chris Yi, and Xiaojuan Ma. 2018. EmbeddingVis: A Visual Analytics Approach to Comparative Network Embedding Inspection. *arXiv preprint arXiv:1808.09074* (2018).

[24] Quan Li, Qiaomu Shen, Yao Ming, Peng Xu, Yun Wang, Xiaojuan Ma, and Huamin Qu. 2017. A visual analytics approach for understanding egocentric intimacy network evolution and impact propagation in MMORPGs. In *2017 IEEE Pacific Visualization Symposium (PacificVis)*. IEEE, 31–40.

[25] Jessica Lin, Eamonn Keogh, and Stefano Lonardi. 2005. Visualizing and discovering non-trivial patterns in large time series databases. *Information visualization* 4, 2 (2005), 61–82.

[26] Christophe Lino and Marc Christie. 2015. Intuitive and efficient camera control with the toric space. *ACM Transactions on Graphics (TOG)* 34, 4 (2015), 82.

[27] Christophe Lino, Marc Christie, Fabrice Lamarche, Guy Schofield, and Patrick Olivier. 2010. A real-time cinematography system for interactive 3d environments. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. Eurographics Association, 139–148.

[28] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. 2008. Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining*. IEEE, 413–422.

[29] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, Nov (2008), 2579–2605.

[30] Matthew V Mahoney and Philip K Chan. 2003. *Learning rules for anomaly detection of hostile network traffic*. Technical Report.

[31] Danielle M McCarthy, George T Chiampas, Sanjeev Malik, Kendra Cole, Patricia Lindeman, and James G Adams. 2011. Enhancing community disaster resilience through mass sporting events. *Disaster medicine and public health preparedness* 5, 4 (2011), 310–315.

[32] George Moody. 1983. A new method for detecting atrial fibrillation using RR intervals. *Computers in Cardiology* (1983), 227–230.

[33] Heather L O?Brien, Paul Cairns, and Mark Hall. 2018. A practical approach to measuring user engagement with the refined user engagement scale (UES) and new UES short form. *International Journal of Human-Computer Studies* 112 (2018), 28–39.

[34] Jim O'Donoghue, Mark Roantree, Bryan Cullen, Niall Moyna, Conor O'Sullivan, and Andrew McCarren. 2015. Anomaly and event detection for unsupervised athlete performance data. (2015).

[35] PhysioToolkit PhysioBank. 2000. PhysioNet: components of a new research resource for complex physiologic signals. *Circulation. v101 i23. e215-e220* (2000).

[36] Monika Pobiruchin, Julian Suleder, Richard Zowalla, and Martin Wiesner. 2017. Accuracy and adoption of wearable technology used by active citizens: a marathon event field study. *JMIR mHealth and uHealth* 5, 2 (2017), e24.

[37] Fayez Qureshi and Sridhar Krishnan. 2018. Wearable Hardware Design for the Internet of Medical Things (IoMT). *Sensors* 18, 11 (2018), 3812.

[38] Roberto Ranon and Tommaso Urli. 2014. Improving the efficiency of viewpoint composition. *IEEE Transactions on Visualization and Computer Graphics* 20, 5 (2014), 795–807.

[39] Nandakumar Selvaraj, Ashok Jaryal, Jayashree Santhosh, Kishore K Deepak, and Sneh Anand. 2008. Assessment of heart rate variability derived from finger-tip photoplethysmography as compared to electrocardiography. *Journal of medical engineering & technology* 32, 6 (2008), 479–484.

[40] Ekrem Serin, Serdar Hasan Adali, and Selim Balcisoy. 2012. Automatic path generation for terrain navigation. *Computers & Graphics* 36, 8 (2012), 1013–1024.

[41] Mei-Ling Shyu, Shu-Ching Chen, Kanoksri Sarinnapakorn, and LiWu Chang. 2003. *A novel anomaly detection scheme based on principal component classifier*. Technical Report. MIAMI UNIV CORAL GABLES FL DEPT OF ELECTRICAL AND COMPUTER ENGINEERING.

[42] Dmitry Sokolov and Dimitri Plemenos. 2008. Virtual world explorations by using topological and semantic knowledge. *The Visual Computer* 24, 3 (2008), 173–185.

[43] Pere-Pau Vázquez, Miquel Feixas, Mateu Sbert, and Wolfgang Heidrich. 2001. Viewpoint selection using viewpoint entropy.. In *VMV*, Vol. 1. 273–280.

[44] Yun Wang, Zhutian Chen, Quan Li, Xiaojuan Ma, Qiong Luo, and Huamin Qu. 2016. Animated narrative visualization for video clickstream data. In *SIGGRAPH Asia 2016 Symposium on Visualization*. ACM, 11.

[45] Stephan Weibelzahl. 2001. Evaluation of adaptive systems. In *International Conference on User Modeling*. Springer, 292–294.

[46] Weng-Keen Wong, Andrew W Moore, Gregory F Cooper, and Michael M Wagner. 2003. Bayesian network anomaly pattern detection for disease outbreaks. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*. 808–815.

[47] Ke Xie, Hao Yang, Shengqiu Huang, Dani Lischinski, Marc Christie, Kai Xu, Minglun Gong, Daniel Cohen-Or, and Hui Huang. 2018. Creating and chaining camera moves for qadrotor videography. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 88.

[48] Ke Xu, Yun Wang, Leni Yang, Yifang Wang, Bo Qiao, Si Qin, Yong Xu, Haidong Zhang, and Huamin Qu. 2019. CloudDet: Interactive Visual Analysis of Anomalous Performances in Cloud Computing Systems. *arXiv preprint arXiv:1907.13187* (2019).

[49] Kenji Yamanishi, Jun-Ichi Takeuchi, Graham Williams, and Peter Milne. 2004. On-line unsupervised outlier detection using finite mixtures with discounting learning algorithms. *Data Mining and Knowledge Discovery* 8, 3 (2004), 275–300.