Visualizing sailing regatta events in Sailing+

Alexandru Lungu
Responsible Professor: Elmar Eisemann
Supervisor: Mark Winter
EEMCS, Delft University of Technology, The Netherlands
22-6-2022

A Dissertation Submitted to EEMCS faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering

Abstract

This paper proposes a novel approach to visualizing events in sailing regattas, in a way that is engaging, informative and interactive to the users of the Sailing+ application. The approach used is to create two different types of visualizations for each type of event: artistic, which contains animations and effects such as depth of field, and informative, which shows key information about the event. The events are visualized with the help of animated thumbnails, in which the two views are blended together. These thumbnails have been added to the race timeline, and a static shot is shown first after which the user can play the animation. To allow users to express their preference for the static image, an interactive slider allows the selection of any frame in the thumbnail animation.

1 Introduction

In sailing, as in any other sport, there are certain moments in every race that can have a big impact on the race outcome. These moments are defined as race events, and they are a key part in making regattas engaging for spectators. For example, when sailboats collide or when competitors are slowed down by being in the shadow cast by another participant's boat. Having a way to summarize such events would then be convenient for the users.

In this paper, a novel and interactive method in which race events can be visualized is presented through the use of animated thumbnails. The animation will transition from an artistic view, which might include cinematographic effects such as motion blur or depth of field [Lin et al. (2012)], while portraying different camera angles, to an informative one that has no effects and includes details such as wind direction or trails showing previous boat directions.

The implementation was realised in a software called Sailing+, which is a new application developed in Unity by The Computer Graphics and Visualization Group at TU Delft, in collaboration with the Sailing Innovation Center. It is meant to provide an engaging way both for people new to sailing, as well as to the more experienced, to enjoy sailing competitions, also called regattas, in Augmented Reality (AR) and Virtual Reality (VR). Even though the ideas presented in this paper were implemented in Sailing+, they could be adapted to other environments.

The main research question we are looking to answer is "How can interesting race events be visualized in time and space, using thumbnails, to help with user interaction, engagement and understanding?".

We attempted to tackle this question by considering two separate views: an engaging view, referred to as artistic, and an informative view. The purpose of the first one is to provide a compelling visualization to engage the users, while the second one should provide the user with the main information needed about an event. These two views have been merged together into animated thumbnails. On the race timeline, an informative frame is displayed, and the users can either start the animation or select their preferred static shot from all the frames in the animation, through an interactive slider.

The main research question can then be further divided into three subquestions, as follows:

- How can photographic and cinematographic techniques be applied to provide an engaging visualization of race events?
- How can useful information in a race event be showcased in an informative way?
- How can a user interactively select their preferred image from a thumbnail animation?

There are three main contributions this paper provides:

1. Creating a way to visualize sailing events as thumbnails using cinematographic techniques and showing relevant event information.
2. Providing an adaptation for Incident and Wind Shadow events, apart from a general view that can be changed to fit certain needs depending on event type.
3. Facilitating user interaction with event thumbnails by allowing them to select their desired static frame.

This paper will go through the steps of answering each research question, with the sections divided as follows: Section 2 gives a detailed presentation of the methodology used to create a general view and how that was adapted for Incident and Wind Shadow event types, and Section 3 presents the results that were gathered from the use of said methodology. Section 4 provides a discussion on obtained results and on the possible improvements of the presented approach. The last section displays the conclusions of this research. In Appendix A, the ethical aspects of the conducted research are presented. In Appendix B, examples of the animated thumbnails can be accessed with the provided link, and in Appendix C, an overview of the mathematical notation used in this paper is shown along with a few examples.

2 Methodology

The aim of this research project is to create engaging visualizations for key events from regattas in the Sailing+ application, that can be represented as animated thumbnails, on a timeline.

2.1 Overview

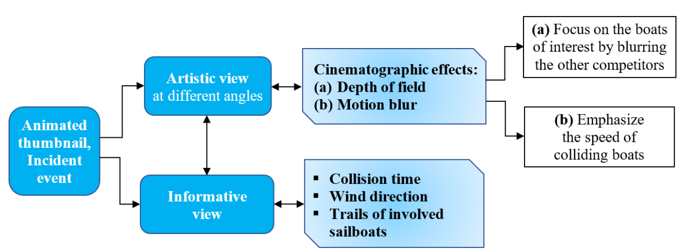


Figure 1: Overview of an animated thumbnail designed for Incident events. The visualization uses elements from the general method and is adapted for the Incident event type

To obtain adequate information about an event along with compelling visualizations, two separate views are created and the thumbnails are animated such that they blend between these views. The first one is the artistic view, which alternates between different camera angles and might contain effects such as depth of field to emphasize focus, or motion blur to highlight movement. The second one is the informative view whose scope is to allow the user to quickly grasp what happened in an event and the artistic effects are removed. The artistic view is still informative for the users, but the informative view has no artistic effects. An overview for the method applied for Incident events can be seen in Figure 1.

The process of achieving an animated thumbnail, in this approach, needs manual tweaking for each event type, because each type might have distinct representations. In this section, a general method is first presented that applies to all event types. Then, the general method is adapted and tweaked for two event types: Incident, which represents the collision of two boats, and Wind Shadow, an event where one of the boats blocks the wind from reaching the boat in its shadow, which slows it down.

The methodology will be broken down into four further subsections: General algorithm description, Incident adaptation, Wind Shadow adaptation and Placing the thumbnails on the application timeline.

2.2 General algorithm

All direction vectors defined below refer to the (x, z) plane as the y coordinate is set separately when needed. The general method of obtaining an animated event thumbnail is described in the steps below:

Create a camera for each event. As some events might happen at the same time, a different camera is necessary for each event. All of the cameras have two commonly defined parameters: a field of view of 60 and the near clip plane set to 0.001, to ensure the cameras render properly when they are close to the regatta field.

Set the starting time of each view. A thumbnail animation takes 13 seconds in total, as follows: 5 seconds for the starting, artistic view; 3 seconds for the informative view, during which the camera stands still; and 5 seconds at the end, back to an artistic visualization. A timeline that shows all of the timepoints used in this section can be seen in Figure 2.

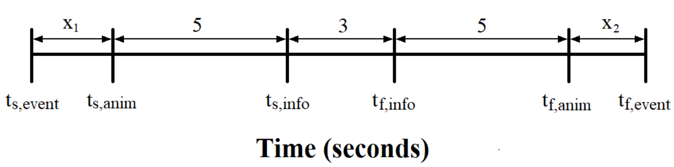


Figure 2: Timeline of an event thumbnail animation. An animation has a fixed duration of 13 seconds, from $t_{\text{s,anim}}$ to $t_{\text{f,anim}}$. x_1 and x_2 are determined by $t_{\text{s,anim}}$ and they vary based on event type.

As it is impossible to find an optimal informative time without having information about event type, the start time for an informative view in the general case has been semi-arbitrarily set, after observing a few events, to

$$t_{\text{s,info}} = t_{\text{s,event}} + 8 \tag{1}$$

where $t_{\text{s,event}}$ is the event start time, and all time units are defined in seconds. The animation start time is then defined as

$$t_{\text{s,anim}} = t_{\text{s,info}} - 5 \tag{2}$$

Approximate the direction of competitors. For each boat involved in an event, we approximate their direction from the event start time up to the animation start time by sampling their positions in intervals of 0.5 seconds and normalizing each sampled direction to unit length. The direction of a competitor, up to $t_{\text{s,anim}}$, is defined as

$$\mathbf{d}_c = \sum_{\substack{t=t_{s,event} \\ t+=0.5}}^{t_{s,anim}} \frac{\mathbf{p}_c(t) - \mathbf{p}_c(t-0.5)}{||\mathbf{p}_c(t) - \mathbf{p}_c(t-0.5)||} \tag{3}$$

where $\mathbf{p}_c(t)$ is the position of competitor c at time t.

Define the direction that the camera is placed during an informative view. The informative direction is defined as the sum of all normalized competitor directions:

$$\mathbf{d}_{\text{info}} = \sum_{i=0}^{n} \hat{\mathbf{d}}_{c_i} \tag{4}$$

where n is the number of competitors involved in an event.

Define the camera direction for the start of the animation. For a general view we are aiming to start with a 3/4 shot, to have the main subject boats visible both from the front and the side when their directions are similar. To achieve this, we are taking the direction orthogonal to $\mathbf{d}_{\text{info}}$, on the right side, and adding it with $\mathbf{d}_{\text{info}}$, which creates a 45° angle between $\mathbf{d}_{\text{info}}$ and $\mathbf{d}_{\text{side}}$. This can be seen in Figure 3.

$$\mathbf{d}_{start} = \hat{\mathbf{d}}_{side} + \hat{\mathbf{d}}_{info} \tag{5}$$

where $\mathbf{d}_{\text{side}} = \hat{\mathbf{d}}_{\text{info}} \times [0 \quad 1 \quad 0]^T$

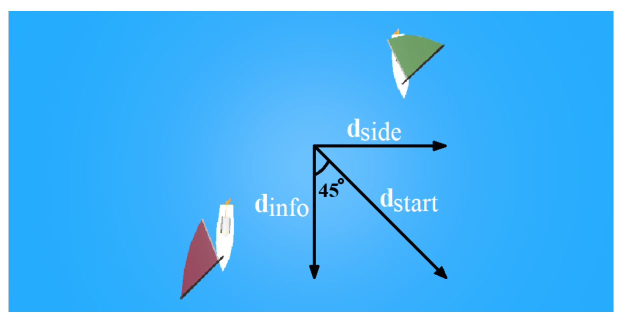


Figure 3: An overview of the directions used to define camera positions in an event visualization.

Find overhead camera position. For a general view, the camera is placed directly overhead at $t = t_{\text{s,info}}$ to allow for unobstructed visualizations. The Incident event is given as an example: Figure 4 presents several different views of a collision event with increasing elevation, starting with what would be classified as a "long shot". It can be seen this is not a very informative shot as it is too close. It provides a clear view of the collision, but little to no context about what had caused the collision to take place (Figure 4a). Figure 4c displays an

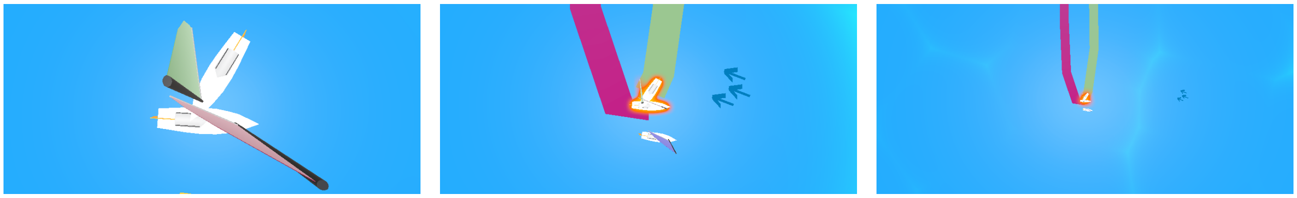

(a) (b) (c)

Figure 4: An informative overhead view of an Incident event, zoomed out gradually to portray different shot levels. (a) Medium to long shot. This shot is too close to the competitors and it is hard to figure out what had caused them to collide (b) Extreme long shot. This is the view we have chosen as it is far enough away to give enough context, we can now see the reason they collided is because of the boat on the bottom given their position, and at the same time it allows more space to show the wind direction as an extra informational element (c) Another extreme long shot. This time the camera is too far away and it is almost impossible to tell what is happening

extreme long shot, and it is hard to deduce what is happening as the camera is too far away. Figure 4b presents a shot which gives enough context while still focusing on the subjects.
The elevation of the camera is determined by approximating the percentage of screen space the participant boats would take up at various camera elevations y_{cam}, which essentially dictates how much context is shown. This can be seen in Figure 5: the rectangular base of a competitor's bounding box, marked with yellow, occupies more screen space when the camera is at a lower elevation (y_1), and less at a higher elevation (y_2). The screen space that the base of a boat occupies in an overhead view, at a given camera elevation y_{cam}, is computed as follows:

$$space_{\text{boat}}(y_{cam}) = \frac{w_{base}(y_{cam}) * h_{base}(y_{cam})}{w_{screen} * h_{screen}} \quad (6)$$

where screen is the screen space, defined in pixels and $base$ is the rectangular base of the bounding box of the boat model in screen space, at camera elevation y_{cam}, while w and h are shorthand for width and height.
The role of the overhead view is to provide information, so a significant amount of context would be needed. The view in Figure 4b can be classified as an "extreme long shot", with competitors taking up about 2% screen space [Mademlis et al. (2020)]. However, it is important to note that we are using wide shots, and a lot of space is left unused. While the percentage may seem low, it does not shift away the focus from the competitors.
The value for the overhead camera elevation is computed as

$$y_{\text{info}} = \underset{y_{cam}}{\text{argmin}} \left|2 * space_{boat}(y_{cam}) - 0.02\right| \quad (7)$$

$\forall y_{cam} \in [0, 1]$

Define camera position and rotation. The center of mass (COM) of the competitors that partake in an event is given by

$$\mathbf{COM}(t) = \frac{1}{n} * \sum_{i=1}^{n} \mathbf{p}_{c_i}(t) \quad (8)$$

where n is the number of competitors in an event.
The camera is initially placed at $\mathbf{COM}(t_{\text{s,anim}})$, and moved

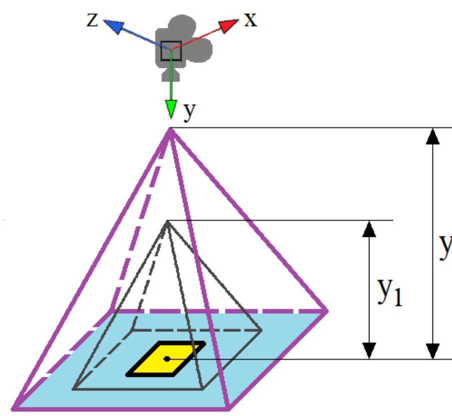


Figure 5: Event camera at 2 different elevations. The rectangular base of a competitor model's bounding box is marked with yellow. At elevation y_1, the boat occupies more screen space, delimited by the base of the gray pyramid, than at a higher elevation y_2, where the screen space is delimited by the base of the purple pyramid

along $\mathbf{d}_{\text{start}}$ by 0.05 units. The elevation is set to $0.75 * y_{\text{info}}$.

$$\mathbf{p}_{i,cam} = \mathbf{COM}(t_{\text{s,anim}}) + 0.05 * \hat{\mathbf{d}}_{\text{start}} + \begin{bmatrix} 0 \\ 0.75 * y_{\text{info}} \\ 0 \end{bmatrix} \quad (9)$$

The rotation of the camera is determined by the LookAt vector [Christie et al. (2008)], with the camera looking at the center of mass of competitors at any point in the event interval.

$$\mathbf{LookAt}_{\text{cam}}(t) = \mathbf{COM}(t) - \mathbf{p}_{cam}(t) \quad (10)$$

$\forall t \in [t_{\text{s,anim}}, t_{\text{f,anim}}]$. The initial camera direction is set to $\mathbf{LookAt}_{\text{cam}}(t_{\text{s,anim}})$. Afterwards, the zoom level is computed according to how close the main subjects are to the grid lines defined by the rule of thirds [Mai et al. (2011)][Amirshahi et al. (2014)], in screen space. This gives us the starting camera position

$$\mathbf{p}_{s,cam} = \mathbf{p}_{i,cam} + zoom * \mathbf{LookAt}_{\text{cam}}(t_{\text{s,anim}}) \quad (11)$$

Define the final camera position. The animation will transition towards this position starting at

$$\mathbf{t}_{\text{f,info}} = \mathbf{t}_{\text{s,info}} + 3 \quad (12)$$

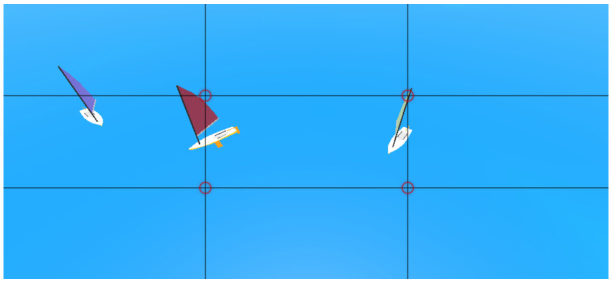


Figure 6: Beginning frame of an artistic view, chosen according to the rule of thirds. The subjects were placed along the grid lines, close to the lines' intersection points

The final direction is symmetric to $\mathbf{d}_{\text{start}}$, and the final camera position equation is similar to Eq. (9):

$$\mathbf{p}_{\text{f,cam}} = \mathbf{COM}(t_{\text{f,anim}}) - 0.05 * \hat{\mathbf{d}}_{\text{start}} + \begin{bmatrix} 0 \\ 0.75 * y_{\text{info}} \\ 0 \end{bmatrix} \tag{13}$$

Transition between views. The camera will first transition to the informative view and stand still for a duration of 3 seconds, in a position given by

$$\mathbf{p}_{\text{info,cam}} = \mathbf{COM}(t_{\text{s,info}}) + \begin{bmatrix} 0 \\ y_{\text{info}} \\ 0 \end{bmatrix} \tag{14}$$

The camera position at any frame in the animation is given by a linear interpolation defined as

$$\mathbf{p}_{cam}(F) = \mathbf{p}_{cam}(F-1) + (\mathbf{p}_{arr}(t) - \mathbf{p}_{cam}(F-1)) * \Delta t \tag{15}$$

where F is the current frame, and Δt is the time difference between F and $F-1$. Moreover, $\mathbf{p}_{arr}(t)$ is the camera position we want to arrive at, based on the current time t

$$\mathbf{p}_{\text{arr}}(t) = \begin{cases} \mathbf{p}_{\text{info, cam}} & t \leq t_{\text{s,info}} \\ \mathbf{p}_{\text{f,cam}} & t_{\text{s,info}} < t \leq t_{\text{f,anim}} \end{cases} \tag{16}$$

Add the depth of field effect. This artistic effect provides a way of keeping the focus on the involved competitors, as in our implementation it blurs everything in the background, regardless of depth, and it is especially important when the area around the participants is bloated with other boats. The effect takes place when $t \in [t_{\text{s,anim}}, t_{\text{s,info}}) \cup (t_{\text{f,info}}, t_{\text{f,anim}})$.

Add informative elements. The boats of interest in an event are highlighted with a thin orange contour, to help the users keep their focus on the main subjects as the other effects get removed. Furthermore, the trails of these sailboats are enabled to show their previous movements. These elements are shown when $t \in [t_{\text{s,info}}, t_{\text{f,info}}]$.

For events such as Incident and Wind Shadow, the parameters have been modified to fit with their respective views.

2.3 Event type: Incident

The Incident event type will closely follow the general algorithm. With the exception of finding $t_{\text{s,info}}$, all of the equations are the same. The starting time for the informative view is computed as the moment the boats collide, which is approximated by the time when they are closest to each other, in Euclidean distance.

$$t_{\text{s,info}} = \underset{t}{\operatorname{argmin}}\{d(\mathbf{p}_{c_1}(t), \mathbf{p}_{c_2}(t)) | \forall t \in [t_{\text{s,event}}, t_{\text{f,event}}]\} \tag{17}$$

where c_1 is the first competitor and c_2 is the second competitor involved in a collision.

The artistic effects for this event type were depth of field and motion blur, used separately.

The informative elements include the ones in the general method, which were boat highlights and trails, with the addition of arrows that indicate the wind direction at the time of collision. The arrows are placed at the midpoint between the center of the screen and the right screen margin, and they are slowly moving in the wind direction. An illustration of all these elements can be seen in Figure 4b.

2.4 Event type: Wind Shadow

The visualization for Wind Shadows starts with a side view, and the informative position is set diagonally in the direction of the shadow. As the shadowed boats start taking action, such as changing direction, after the middle timepoint in the event, the informative time has been set to

$$t_{\text{s,info}} = \frac{1}{2} * (t_{\text{s,event}} + t_{\text{f,event}}) \tag{18}$$

We place the camera on the side that is closest to the shadowed boat. To do that, first we compute the cross product between the projection of the shadowing boat on dir_{inf} and the shadowed boat, translated to the projection of the shadowed boat as origin.

$$cross = (\text{proj}_{\mathbf{d}_{\text{info}}}(\mathbf{p}_{c_1}) - \text{proj}_{\mathbf{d}_{\text{info}}}(\mathbf{p}_{c_2})) \times (\mathbf{p}_{c_2} - \text{proj}_{\mathbf{d}_{\text{info}}}(\mathbf{p}_{c_2})) \tag{19}$$

where c_1 is the shadowing boat and c_2 is the shadowed boat.

This cross product gives us the starting direction, defined for this event as

$$\mathbf{d}_{\text{start}} = \begin{cases} \mathbf{d}_{\text{side}} & cross < 0 \\ -\mathbf{d}_{\text{side}} & cross \geq 0 \end{cases} \tag{20}$$

where $\mathbf{d}_{\text{side}}$ refers to the orthogonal vector on the right side of $\mathbf{d}_{\text{inf}}$ and is defined in Eq. (5).

The camera is being placed, at the informative start time, to the back of the shadowed boat, and in the direction of the shadow. Taking the current $\mathbf{d}_{\text{info}} = \mathbf{d}_{\text{front}}$, the updated informative direction is obtained:

$$\mathbf{d}_{\text{info}} = \hat{\mathbf{d}}_{\text{start}} - \hat{\mathbf{d}}_{\text{front}} \tag{21}$$

The last parameter to set is the camera position at informative start time, defined as

$$\mathbf{p}_{\text{info,cam}} = \mathbf{COM}(t_{\text{s,info}}) + 0.05 * \hat{\mathbf{d}}_{\text{info}} + \begin{bmatrix} 0 \\ 0.75 * y_{\text{info}} \\ 0 \end{bmatrix} \tag{22}$$

(a) (b) (c)

Figure 7: An artistic view of an Incident event, without effects. (a) The beginning of the animation. This is where the artistic view starts, at an elevated 3/4 angle. The zoom level is set to portray the main competitors following the rule of thirds. (b) Middle point between animation start and informative view. As the camera transitions to an informative visualization, the elevation is increased and the angle converges to the informative camera direction. The purple boat can be seen too close to the red sailboat, which causes it to switch direction and collide, and the view transitions to informative (Figure 4b). (c) The end of the animation. After the informative view, the camera is now back to showing an artistic visualization in the opposite direction to where the artistic view started

(a) (b) (c)

Figure 8: An artistic view of an Incident event, with the depth of field effect enabled to emphasize focus on the involved competitors. (a) The beginning of the animation. The competitor in the background is blurred and the focus is given to the competitors in the middle (b) With this effect enabled, it can be harder to notice the purple boat causing the collision (c) The end of the animation. As there are no other competitors in the camera's field of view, the image is very similar to Figure 7c

2.5 Placing thumbnails on the timeline

The final step is to render the achieved views into an animation and attach them to the timeline in Sailing+. To do this, each frame in the time interval $[t_{\text{s,anim}}, t_{\text{f,anim}}]$ has been captured and saved as a png file. These files have then merged together into an mp4 video, which has been chosen as it is a compact format and it preserves animation quality. The resolution of the render texture is the same both for the pictures and the video, and it is defined by the size of the user's screen, which preserves the aspect ratio.

The thumbnails can be accessed by pressing the green buttons next to the event start times on the timeline, which toggle a panel that displays a static shot of the thumbnail along with a play button and a frame selection button. The default picture is an informative view of the event, and the animation starts if the users press on the play button. As users might consider another static shot to fit better as a static thumbnail, they can select their preferred frame with the help of a slider, and their preference is saved to be used in future application runs. An illustrative example of these elements can be seen in Figure 10 and Figure 11.

3 Results

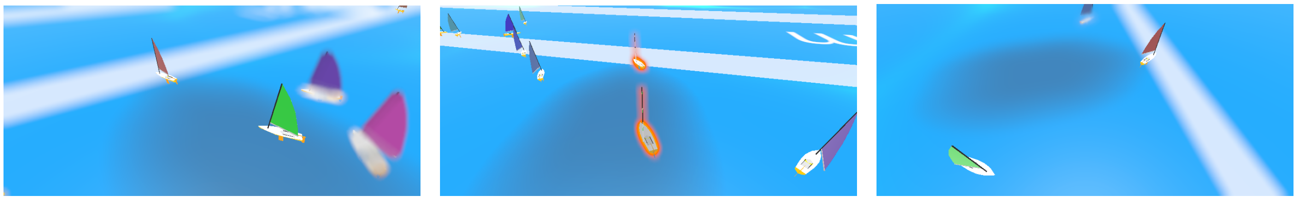
The results presented in this section are from the computer version of the Sailing+ application run in the Unity environment. The described methodology for obtaining the thumbnails is the same for any setup, including mobile devices and AR environments.

In the thumbnails, the animation alternates between showing artistic effects while the camera is moving, and removing those effects and adding event-specific elements with the camera sitting still.

In Figures 7 and 8 the artistic view of an Incident event is shown. Figure 7 has no post-processing effects enabled, while Figure 8 uses depth of field. In both cases, the start of the animation is displayed, after which the middle point before the informative view begins, and lastly where the animation finishes. (a) shows the start of the animation, with the competitors portrayed according to the rule of thirds, which means the subjects are already in focus. Adding depth of field emphasizes the focus on the event participants and adds to the artistic effect; (b) is the point between the animation start and the overhead view (Figure 4b). One major difference is that it can be harder to notice the purple sailboat, which is the main reason the collision happened in the first place as the red boat barely managed to avoid it; (c) is the finish point of the animation which ends in an artistic shot, in a direction symmetric to the one in the beginning. The differences here are not so noticeable since there are no other competitors in view.

Even though the visibility is more limited in Figure 8b, that is mitigated by the fact that you can see the boat more clearly in the other frames, and in the informative view the effects are disabled. This makes the view with depth of field preferable as it achieves a visualization that is more compelling, and keeps the focus on the main competitors.

Motion blur was another effect we used for this event type.



(a) (b) (c)

Figure 9: A visualization of a wind shadow effect, with the depth of field effect enabled. (a) The beginning of the animation. The view starts to the side of the shadowed green boat (b) Informative view. The effects are disabled and the boats are highlighted, with the camera placed in the direction of the shadow (c) The end of the animation. Back to an artistic view, where the shadowed boat can be seen changing its course

Figure 10: Thumbnail of a collision event. The thumbnail is placed on a panel that can be accessed by pressing the green button next to the event on the timeline, highlighted by the red circle in the image. The users can play the animation by pressing the play button represented with the yellow circle, or they could click on the button shown in the blue circle to access a slider that lets them select a static shot of their choosing and can be seen in Figure 11.

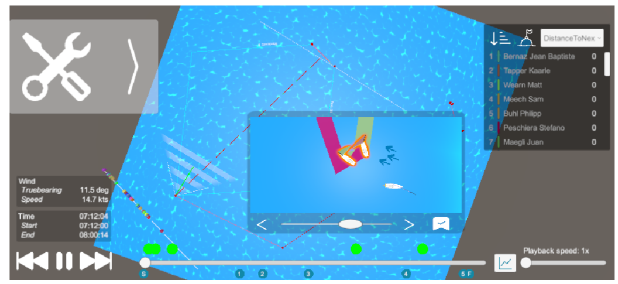


Figure 11: Event thumbnail, frame selection view. Users can utilize the slider, or the left and right arrow buttons, to select any frame in the thumbnail animation that will thereafter be used as the static image in the panel view (Figure 10).

The implementation was done using Unity's Universal Render Pipeline post-processing stack, which worked well for depth of field but inadequately for the motion blur effect. Even though we used the parameters for which the view should have been the most intense, a significant difference was not observed. Another type of motion blur implementation might be a better fit for Incidents, that will be considered for future work.

The other event that we manually designed for, Wind Shadow, has shown promising results. Figure 9 showcases all stages of a thumbnail animation for a Wind Shadow event: artistic - informative - artistic. As the transition from the artistic view to the informative view takes place, the depth of field effect is removed and the boats become highlighted with an orange contour. In the end, the green boat that was being shadowed is seen changing its direction. This is a good example of the depth of field effect having a beneficial application, other than purely for artistic purposes. In Figure 9a, there are two more boats next to the sailboat being shadowed, and the blur effect makes it clear both what the subject in this photo is and which boat is shadowing it.

An example of both of the previously mentioned events along with another one using the general method can be found in Appendix B, where the full mp4 files of the animated thumbnails can be accessed. Note that for the Incident example, the purple and red sailboats might seem to collide, but the red boat actually manages to avoid it at the last second and they collide only with the green sailboat.

4 Discussion

An issue with the approach we have taken in this paper is that each event type needs to be considered separately and manually modelled. This can make the visualizations better catered to each event type, but the design is time consuming. A general method has been provided, but that might not work well enough for all event types and manual design would be required.

Another limitation with the current approach is that the views are constrained to defined positions and some factors might not be considered. For example, some competitors might be occluded in the current views. This is mitigated by the fact that the camera angle changes and the event participants are highlighted, either themselves or by blurring the other competitors around using effects like depth of field.

Considering alternative solutions for the approach we have taken, animating a static thumbnail would have been another option. Instead of creating a view over time, elements of capturing user attention and creating understanding could have been gradually added to a static shot that could alternate between these elements. Dropping the animations altogether and choosing artistic angles and representations while still keeping the thumbnails informative might have been another option.

In this paper, we have presented two event types and how these were modelled. To extend these ideas to other event types, the general visualizations can be used as a template and then adapted by following a few basic steps:

- Define how the informative view will look like
- Starting from the informative view, think about how this can be reached from the artistic view. This can also be done the other way around, depending on the event
- Define camera parameters given the constraints of the specific event type, for each view
- Define how the transition between the views will look like

The approach can also be extended with more cinematographic techniques. For example, modelling each frame in the animation with regards to rules of shot composition, such as keeping the subjects along intersection lines during the animation by using the rule of thirds, or using visual balance [Abdullah et al. (2011)].

Another way to extend our approach in the future is to consider camera placement as an optimization problem, and set constraints that decide what is optimal for a visualization given an event type [Normand (2010)][Christie and Languénou (2003)].

The tools we used for obtaining the motion blur effect have not been effective. Another way that effect could be achieved is by taking each static frame and modifying it, such that the main competitors in an event appear blurred [Luo et al. (2018)][Navarro et al. (2011)]. This would be done before the video is created to ensure the animation is modified accordingly.

The process of gathering each frame in an event visualization and saving them for the user to choose their preferred frame using the interactive slider, can quickly fill up a user's storage, especially when considering the application will be mostly used on mobile devices or VR consoles. Moreover, for the thumbnails to even appear, the user would first have to wait until the event takes place for the event camera to render the animation, and not modify the speed of the timeline slider for proper frame capturing. On future iterations the thumbnails will be visible and the process does not need repeating, but this could still place an unnecessary burden on the user. One solution to fixing these issues is to render the thumbnails externally, and hosting the frames on a server so the user is not required to locally store them. The downside to this is that an internet connection would be required for changing static thumbnail frames.

A final idea of improving the visualizations is to use Bezier curves when transitioning between camera angles. We used a simple linear interpolation approach to decide the camera position during animations, and moving the camera in a curve could make the animation look smoother [Baydas and Karakas (2019)].

5 Conclusions

We presented a novel method for visualizing sailing regatta events in the Sailing+ application, through the use of animated thumbnails. Our solution portrays events in an artistic way, through effects and animations that showcase different camera angles, as well as in an informative way, displaying important information relating to each event type. The two views alternate between each other during thumbnail animations. Promising results were obtained by using the depth of field cinematographic effect, providing focus on the main competitors involved in an event. The results achieved in the informative view yielded a clear overview of an event along with key elements about the circumstances that could have caused the event. The thumbnails were added to the race timelines in Sailing+, with a static, informative shot being displayed first on a panel. The user is given the choice to either play the animation, or use an interactive slider to select their preferred frame to replace the static image and have the preference saved on future application runs. The solutions we proposed can be adapted to other environments that portray sailing competitions.

References

Rafid Abdullah, Marc Christie, Guy Schofield, Christophe Lino, and Patrick Olivier. 2011. Advanced composition in virtual camera control. *Smart Graphics* (2011), 13–24. https://doi.org/10.1007/978-3-642-22571-0_2

Seyed Ali Amirshahi, Gregor Uwe Hayn-Leichsenring, Joachim Denzler, and Christoph Redies. 2014. Evaluating the rule of thirds in photographs and paintings. *Art amp; Perception* 2, 1-2 (2014), 163–182. https://doi.org/10.1163/22134913-00002024

Senay Baydas and Bulent Karakas. 2019. Defining a curve as a Bezier curve. *Journal of Taibah University for Science* 13, 1 (2019), 522–528. https://doi.org/10.1080/16583655.2019.1601913

Marc Christie and Eric Languénou. 2003. A Constraint-Based Approach to Camera Path Planning, Vol. 2733. 172–181. https://doi.org/10.1007/3-540-37620-8_17

Marc Christie, Patrick Olivier, and Jean-Marie Normand. 2008. Camera control in computer graphics. *Computer Graphics Forum* 27, 8 (Jul 2008), 2197–2218. https://doi.org/10.1111/j.1467-8659.2008.01181.x

Huei-Yung Lin, Kai-Da Gu, and Chia-Hong Chang. 2012. Photo-consistent synthesis of motion blur and depth-of-field effects with a real camera model. *Image and Vision Computing* 30 (09 2012), 605–618. https://doi.org/10.1016/j.imavis.2012.07.004

Xuejiao Luo, Nestor Z. Salamon, and Elmar Eisemann. 2018. Adding Motion Blur to Still Images. In *Proceedings of Graphics Interface 2018* (Toronto, Ontario) *(GI 2018)*. Canadian Human-Computer Communications Society / Société canadienne du dialogue humain-machine, 108 – 114. https://doi.org/10.20380/GI2018.15

Ioannis Mademlis, Nikos Nikolaidis, Anastasios Tefas, Ioannis Pitas, Tilman Wagner, and Alberto Messina. 2020. Autonomous UAV Cinematography: A Tutorial and a Formalized Shot-Type Taxonomy. *Comput. Surveys* 52, 5 (2020), 1–33. https://doi.org/10.1145/3347713

Long Mai, Hoang Le, Yuzhen Niu, and Feng Liu. 2011. Rule of Thirds Detection from Photograph. In *2011 IEEE International Symposium on Multimedia*. 91–96. https://doi.org/10.1109/ISM.2011.23

Fernando Navarro, Francisco J. Serón, and Diego Gutierrez. 2011. Motion blur rendering: State of the art. *Computer Graphics Forum* 30, 1 (2011), 3–26. https://doi.org/10.1111/j.1467-8659.2010.01840.x

Jean-Marie Normand. 2010. Computer Graphics and constraint solving: An application to Virtual Camera Control. *Trends in Constraint Programming* (Jan 2010), 395–404. https://doi.org/10.1002/9780470612309.ch27

A Responsible Research

When any research is conducted, ethical considerations must be taken into account. In the case of this project, one concern that may arise is if a user utilizes the effects we described to modify pictures maliciously such that they provide misleading information.

The methods shown here are fully reproducible, and the results will look similar given that the user simulates the environment of the Sailing+ application. Which means that aspects such as trail lines, wind direction or wind shadows should be modelled accordingly, and the events should be similarly trimmed to match the timings.

B Thumbnail Animations

A link to the animations of all presented events, along with one of an Initiative event which uses the general method described in the paper, can be found here:
https://drive.google.com/drive/folders/1SSVG6ilyInFIei2j4HcPzgZfyAS6GCSF?usp=sharing

C Mathematical Notation

The mathematical notation that we used in this paper is summarized in the table below:

Type	Notation	Examples
scalar	italic	t, y, F
vector or point	bold	$\mathbf{p}, \mathbf{d}, \mathbf{COM}$
normalized vector	bold + hat	$\hat{\mathbf{d}}$