

# Detecting and Visualizing Rip Current Using Optical Flow

S. Philip <sup>†1</sup> and A. Pang<sup>1</sup>

<sup>1</sup>University of California, Santa Cruz, United States

## Abstract

Rip current are fast moving and narrow currents that are strongest near the beach. They are dangerous not only for novice but also experienced swimmers. Once caught in the current, the flow of the water pulls the person away from the beach. Many people in panic try to swim against the current and end up drowning due to exhaustion since these currents are usually faster than the speed at which one can swim. They have led to the drowning deaths of more than 100 beachgoers each year in United States alone [USL16]. For a knowledgeable person, these rip-currents are fairly easy to spot. But they are a threat to most people who are unaware of rip-currents or who do not know how to identify them. In this paper, we discuss a novel approach that uses optical flow to detect rip currents and then visualize them in an intuitive manner:

Keywords: Time-dependent, vector fields, and visualization.

## 1. Introduction

Rip currents are very fast moving and narrow channels of water that pose a serious hazard to unwary beach goers. These currents will generally pull a person out to sea. Trying to swim against the current often lead to exhaustion and can lead to drowning. According to United States Lifesaving Association, there are four ways to visually identify rip currents:

- A channel of churning, choppy water;
- A line of sea foam, seaweed, or debris moving steadily seaward;
- Different colored water beyond the surf zone; and
- A break in the incoming wave pattern as waves roll onto shore [USL16].

The rip currents are formed along the coastline where breaking waves can be found. Certain water circulation regions are formed when the waves break strongly at some regions and weakly in others. These generally occur at beaches with sand bars and channel system nearby. The water flows back towards the sea following a narrow path thereby forming a rip current [USL16]

In this paper, we discuss how we can leverage the optical flow algorithm to identify and visually represent these dangerous rip current.

## 2. Motivation

According to a study by the United States LifeSaving Association, the annual death rate caused by rip current exceeds 100. Further-

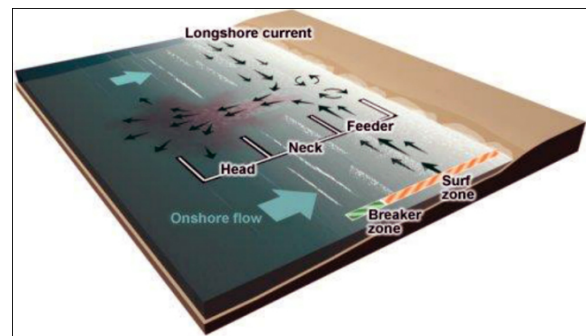


Figure 1: Formation of a rip current (Image courtesy of National Oceanic and Atmospheric Administration, National Weather Service) [rip16]

more, about 80% of the rescues were due to people almost drowning in rip currents. Rip currents usually flow at the speed of 0.5 m/s or 1.1 mph. Some of them can also speed up to 2.5 m/s or 5.6 mph. This is faster than most people can swim [USL16]. From expert to novice swimmers, rip currents have killed more Americans than hurricanes in an average year.

In a study conducted at Pensacola Beach, Florida, 97 participants were surveyed about their ability to identify a rip current. The study showed that the majority of them felt confident they could identify rips e.g. based on the visual descriptions mentioned above. However, less than 20% were actually successful in identifying these currents accurately [CHMA13].

<sup>†</sup> Chairman Eurographics Publications Board

Having an application to detect rip currents at beaches can help inexperienced people identify the location and extents of such dangerous currents. It will help towards preventing drowning of beachgoers as well as reduce the number of rescues performed by lifeguards who also put their own lives in danger in order to save people in need.

### 3. Related Work

Over the years, there are many studies done by universities to understand the behavior of these currents. Many beaches also have set up signs to explain how to safely come out of a rip current if one is caught in it. The University of Delaware Sea Grant College Program and United States Lifesaving Association have set up a website that explains the formation of rip currents and safety precautions one can take [Del16] [USL16]. Various other departments at the University of Delaware and other universities such as Stevens Institute of Technology, New Jersey, Johns Hopkins University, Maryland and many more are pursuing research to break new grounds in understanding behavior of these rip currents. They are also helping National Weather Service and local beach controls to improve the prediction of these rip currents [DSG16].

Apart from these efforts, a patented device was created by Gregory Perrier in 2001. His patent is for an Automated Rip Tide Detection System took various images of the said rip tide and mimic the human perception to identify these rip tides with the help of neural networks. A camera is used to take pictures, and image processing is used to enhance the features that differentiate the rip tides from the normal waves, such as texture, color and direction. Rip tides are essentially powerful currents caused by tide pulling the water through an inlet along a barrier beach, whereas rip currents are created by all the water rushing out from the shore along a single path [GBMM11]. They assume that the rip currents leave the shore perpendicularly, whereas the normal waves hit the shore obliquely. This is followed by training a neural network model, which learns based on the information available from the processed camera images to differentiate rip vs normal currents [Per].

Another patented device created by Earl Senchuk and Michael Rucinski needs to be anchored in the body of water. This device then monitors the speed of the current flowing in its vicinity. The measured speed is compared with a certain amount of threshold. As soon as the speed crosses this threshold, it sends a rip current warning. To be effective, the device must be placed at or near potential rips. It also requires power and maintenance to operate properly [Ruc].

Both of the patented devices mentioned above require specialized modules for processing the data and generate a warning, which makes it inaccessible to a general population. The first approach would require the device to be trained on different beaches. This is necessary since the underwater floor structure may not be the same for all beaches. While rip tides do recur in one place, they are highly unpredictable, and a change in the location would require the device to be re-trained. Also there are some rip currents that do not run perpendicular to the beach. These would be hard to detect with the said device due to the assumptions they made.

Therefore, we want to create an application that can run on a sim-

ple computer (current prototype) but ideally on smart phones. The latter are readily available, usually have a built-in video camera, and about a third of those in circulation have a GPU. Our approach is based on the optical flow calculated from a short video clip of where the waves break on the beach. The processed video is used to identify the general direction of the incoming waves and regions with strong opposing flows, if any. If such regions are found, they are highlighted in the same video clip to indicate the presence and extent of the rip current. The strength or speed of the rip current is also depicted on the highlighted region video using the line thickness of arrows. We believe that providing visualization capability such as this can help save lives.

## 4. Methodology

### 4.1. Optical-Flow

The optical flow algorithm is used to capture the apparent motion objects in a scene. There are many different algorithms to perform optical flow but they all basically consider the same two assumptions [Luc16]:

1. Pixel intensity of an object between consecutive frames remains the same
2. Neighboring pixels follow the same motion.

One such optical flow algorithm is attributed to Lucas-Kanade. We selected this algorithm since it is highly sensitive in capturing small motions between frames. This property is also very good for analyzing dynamic textures e.g. water as opposed to objects with rigid boundaries. The algorithm is explained below:

Consider a pixel with intensity  $I(x, y, t)$  in the first frame, which moves a distance of  $dx, dy$  to its position in the second frame. A time dimension is added to represent how much the pixel moves in time  $dt$ . Taking the first assumption into account, we can write the following equation:

$$I(x, y, t) = I(x + dx, y + dy, t + dt) \quad (1)$$

Assuming that  $dx, dy$  is small, the image constraint at  $I(x, y, t)$  with Taylor series leads to following equation:

$$\frac{\delta I}{\delta x} \frac{\delta x}{\delta t} + \frac{\delta I}{\delta y} \frac{\delta y}{\delta t} + \frac{\delta I}{\delta t} = 0 \quad (2)$$

$$\frac{\delta I}{\delta x} u + \frac{\delta I}{\delta y} v + \frac{\delta I}{\delta t} = 0 \quad (3)$$

$$I_x u + I_y v + I_t = 0 \quad (4)$$

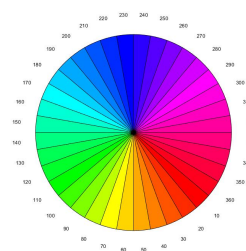
Based on the second assumption that the neighboring pixels have a similar motion, the Lucas-Kanade algorithm uses a 3x3 neighborhood around that point and obtains the values of  $(I_x, I_y, I_t)$ . There are 9 points in consideration, hence we get 9 equations and two unknowns  $(u, v)$  which are the velocity components. This setup results in an over-determined system of equation. A least squares fit

method is then used to find the velocity vector. We use the following equation to determine the vector components:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \sum_i I_{x_i}^2 & \sum_i I_{x_i} I_{y_i} \\ \sum_i I_{x_i} I_{y_i} & \sum_i I_{y_i}^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum_i I_{x_i} I_{t_i} \\ -\sum_i I_{y_i} I_{t_i} \end{bmatrix} \quad (5)$$

## 4.2. Representing Direction and Magnitude

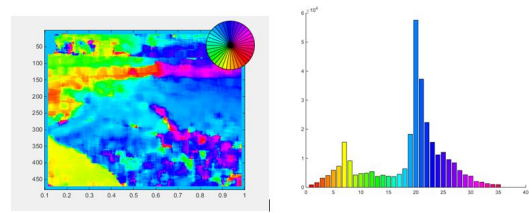
Now that we have obtained the velocity vectors between two frames, we look for a good representation to easily indicate the individual directions. One way is to represent discrete points on the frame with arrows glyphs to indicate the direction and the magnitude at those locations of the vector field. Aside from the problem of clutter, another problem with this representation is that the motion between two frames is very small which makes it difficult to get a sense of the water movement. Another commonly used technique for representing the results of the optical flow algorithm is to pseudocolor the vector field using the HSV color model [Sav16] [UCF16]. In particular, the direction is mapped to the hue component and the magnitude is mapped to the saturation component of the model. In this paper, we adopt this scheme to represent the direction but display the magnitude as a grayscale using a separate image. Note that this coloring scheme is primarily used to explain the algorithm. What the users see is a highlighting of rip current regions in the clip, where the highlighting color is distinct from what one would normally see on the water surface. In Figure 2, we have divided the directions into 36 bins. Each hue representing a range of 10 degrees. For example, red means from the center towards lower right with the range from  $0^\circ$  to  $10^\circ$ .



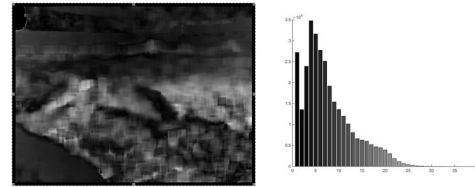
**Figure 2:** Color wheel for mapping directions to hue.

While the direction is mapped to hue, the magnitude is normalized to  $[0, 1]$  and mapped to grayscale. Black represents 0 values and no movement while white represents values of 1 and the fastest displacement of a pixel between two frames. Figure 3 illustrates the directional field from the optical flow map between two consecutive frames along with a histogram recording the frequency of each direction.

Figure 4 represents the magnitude of the velocity along with a histogram of magnitude values. Similar to the direction histogram, magnitude histogram has 36 bins which divides the range into 36 similar sub-ranges. This histogram records the frequency of magnitude values that fall in a particular range.



**Figure 3:** Direction values mapped to hue (left). Frequency of occurrence of each direction (right).



**Figure 4:** Magnitude value as a grayscale image (left) and histogram of the magnitude values (right).

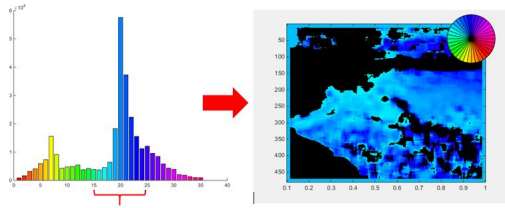
## 4.3. Isolating the Rip Current

Now that we have the direction representation of the frames, we want to threshold the figures to obtain only the direction and magnitude of the rip current. We do this by first finding which color represents the motion of water going back into the ocean. Let us first take a look at the first frame of the video show in Figure 5.



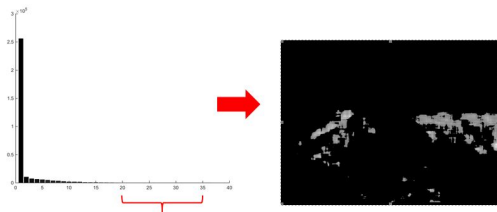
**Figure 5:** First frame of the video

By observing the frames and the corresponding histogram, we can see a two predominant and opposing value directions: incoming waves and outgoing waves. We decide to threshold using a range from  $160^\circ$  to  $230^\circ$  as seen in Figure 2 to extract the region with outgoing flow directions. Any value that falls outside this window is set to zero, otherwise we keep the values as it is. This helps us to remove directions that are not going towards the ocean, thus identifying our potential rip-current. The result of this thresholding is seen in Figure 6.



**Figure 6:** The direction image (right) after masking out the regions that fall outside the range of directions corresponding to potential rip directions (left).

Because the input video clip was obtained using a handheld cell-phone camera, it is naturally susceptible to some shaking which are seen as motion artifacts. Our next step is to remove such unwanted motion artifacts by looking at the magnitude image. Parts of the image such as the sky, land, cliffs, etc. can be considered stationary relative to the motion of the water. To account for the false positive motion detected by the optical flow algorithm in such areas of the image, and to find regions where the speeds might represent a dangerous level, we threshold the magnitude image to only retain the higher values of speed. The result of such thresholding is shown in Figure 7. Note that there is a very high frequency in the first bin since we have set all magnitude values of speed whose direction fell outside of the range  $160^\circ$  to  $230^\circ$  to zero.



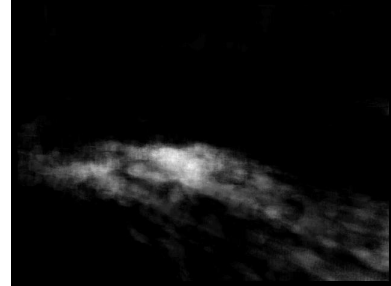
**Figure 7:** The magnitude image (right) showing only regions with values indicated in the histogram (left).

#### 4.4. Representing the Rip Current

Up to this point, we have described how to identify potential regions of rip currents from a pair of consecutive frames. As evident from Figure 7, such areas are usually not contiguous. In addition, these regions obtained from consecutive frames contain noisy artifacts as well as regions that may in fact not be significant. Given that a typical cellphone camera captures video at 30fps and given that waves capable of producing hazardous shore breaks usually have longer periods e.g. 10 seconds or more, we use a sliding time window of 3 seconds or 90 frames to smooth out noise and identify strong and more persistent rips that are more obvious between arrival times of waves hitting the beach. An accumulation buffer with the same resolution as each frame is used. Each element in the buffer is incremented by 1 if that pixel is identified as being a potential rip in an image pair sequence. This is carried out for 89 pairs of image frames. The values in the accumulation buffer are

then normalized and used to represent the likelihood that a rip current exists at a particular pixel. This averaging procedure produces a smoother and more contiguous region.

The image in Figure 8 shows the contents of the accumulation buffer after 3 seconds into the video.



**Figure 8:** Contents of accumulation buffer after 90 frames are analyzed. Cell counts are normalized and mapped to gray scale so that black is zero and white is 1. A higher value indicates that a pixel is tagged as a potential rip current more often than one with a lower value.

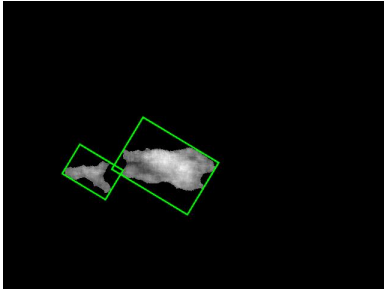
Now that the region and relative likelihood of the presence of a rip current has been identified, our next step is to provide an obvious indication of their presence to the end users. The regions where rip currents are found can be easily displayed as regions of different color overlaid on the video clip. The direction of the rip and its strength can be indicated by an arrow glyph for each contiguous region. We want the length of the arrow glyph to match (actually be a percentage of) the length of the region along the flow direction, while the strength of the rip be mapped to the thickness of lines and arrow head.

To determine the length of the region along the flow direction, we essentially want to find the bounding boxes of each rip region such that the main axis of the bounding boxes is aligned with the flow. That is, we want an oriented bounding box (OBB) around each rip region. The following procedure is used to obtain the OBB shown in Figure 9:

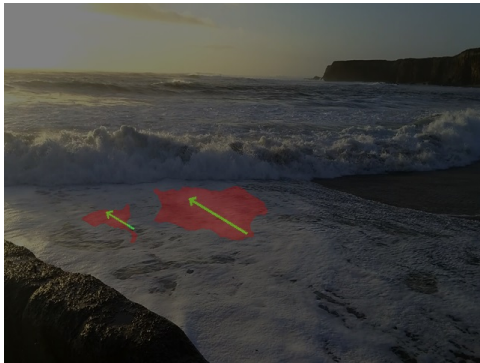
- Perform a threshold on the accumulation buffer to filter out low likelihood portions e.g. darker portions of Figure 8.
- Rotate the image by the main direction of the rip current so that rotated image has the flow direction along the x-axis.
- Find the minimum bounding box for each patch in image.
- Once the bounding boxes are found, rotate the image back to get the OBBs on the un-rotated image.

Note that small patches are ignored since they are difficult to see and do not pose a serious threat.

Once the OBB is obtained, we draw an arrow within the box aligned with the flow direction and set the length of the arrow to be 75% of the box length. Note that to get a continuous patch of rip current, we perform hole filling as well as removing very small patches. These results are then combined frame by frame to produce a video clip visualizing the places with strong rips. A frame from the resulting video is shown in Figure 10.



**Figure 9:** Oriented bounding boxes around each region with rip current.



**Figure 10:** A frame from the video of the beach with visualization of the rip currents.

The reddish patches show places with fast outgoing flows. Reddish hue is chosen because it is usually not a color on naturally occurring ocean water and hence can be easily discernible when overlaid on a video clip. It is also because red is used fairly universally to impart a sense of warning or caution. The arrows in Figure 10 now represent the direction of the rip. The width of the arrows also represent the average magnitude of the velocity in their respective OBB. To get the average magnitude, first we find out the average magnitude of all pixel locations for 90 frames. Then we perform an average of all the values falling in the OBB that we found. This magnitude is used to increase or decrease the width of the arrow. Thus every patch represents an average magnitude over time (90 frames) and space (within the OBB). We calculate these results using a sliding window of 90 frames for the entire video. We combine the resulting frames to obtain a full video that indicates the dominant direction and average magnitude of the rip current over the course of the video. The following url contains video results of this work: <https://users.soe.ucsc.edu/~pang/RipCurrentvideo.mp4.avi>

## 5. Conclusions and Future Work

We have created a prototype implementation to demonstrate the feasibility of detecting and visualizing rip currents. The Lucas-Kanade algorithm is quite robust for finding the direction and magnitude of the motion of water. We combine the output of the optical flow detection to segment the rip current successfully by threshold-

ing direction and magnitude separately based on the patterns of the respective histograms. It is also robust against noise due to small camera movements since we take an average over 90 frames.

While the results are promising, there are two general areas of improvement. First, in terms of rip current features, the current implementation only looks for the predominant rip current direction. On some beaches, there could be multiple rip currents, each going in different directions. Furthermore, there are some rip currents that are more long lived and goes out further to sea on a curved trajectory. A possible course of attack would be to analyze different modes of the histogram and see if the associated patches are connected (for curved trajectories) or not (for different rips going in different directions on different parts of the beach). Secondly, we need to improve on efficiency of the implementation. As this is a proof-of-concept effort, we utilized the MATLAB R2014b<sup>TM</sup> implementation of the Lucas-Kanade algorithm. This was the most time consuming part of the process and could benefit most from a customized implementation. On a laptop with an i7 CPU and 16GB RAM, it takes 40 seconds to compute the optical flow between a pair of 480 x 640 images. Whereas it takes 1.5 sec for the rest of the processing. Ultimately, we want an efficient implementation that can run directly on smart phones and make this application available to everyone.

## References

- [CHMA13] CALDWELL N., HOUSER C., MEYER-ARENDE K.: Ability of beach users to identify rip currents at Pensacola Beach, Florida, 2013. (Natural Hazards'13) <http://link.springer.com/article/10.1007/s11069-013-0673-3/fulltext.html>. doi:10.1007/s11069-013-0673-3. 1
- [Del16] Rip current safety, 2016. (University of Delaware Sea Grant Program - Rip Current Safety.) <http://www.ceoe.udel.edu/ripcurrents/safety/index.html>. 2
- [DSG16] 2016. (University of Delaware Sea Grant Program - Research page) <http://www.ceoe.udel.edu/ripcurrents/Research/index.html>. 2
- [GBMM11] GAULITIERI T., BUDD J., MCGUIRK P., MOTT M.: Rip tide, rip current, undertow: Knowing the difference can save your life, 2011. (Stony Brook University, New York : E.L.I.C.C.A.) <http://www.seagrants.sunysb.edu/cprocesses/pdfs/BeachHazards-RipCurrents0711.pdf>. 2
- [Luc16] Optical flow, 2016. (Optical Flow OpenCV) [http://docs.opencv.org/master/d7/d8b/tutorial\\_py\\_lucas\\_kanade.html](http://docs.opencv.org/master/d7/d8b/tutorial_py_lucas_kanade.html). 2
- [Per] PERRIER G.: Automated rip tide detection system, Aug. 16, 2005. (US Patent 6,931,144) <http://www.google.com/patents/US6931144>. 2
- [rip16] Surviving a rip, 2016. (National Oceanic and Atmospheric Administration, National Weather Service: Surviving a Rip) <http://www.ripcurrents.noaa.gov/science.shtml>. 1
- [Ruc] RUCINSKI M.: Rip current sensor and warning system with anchor, July 18, 2013. (US Patent App. 13/740,186) <http://www.google.com/patents/US20130181842>. 2
- [Sav16] SAVARD J.: Color charts, 2016. <http://members.shaw.ca/quadibloc/other/colint.htm>. 3
- [UCF16] Source code: Optical flow, 2016. (UCF: Center for Research in Computer Vision : Optical Flow Matlab) <http://csrcv.ucf.edu/source/optical>. 3
- [USL16] United States Lifesaving Association rip current survival guide, 2016. <http://www.usla.org/?page=RIPCURRENTS>. 1, 2