2016 IEEE International Symposium on
Safety, Security, and Rescue Robotics (SSRR)
EPFL, Lausanne, Switzerland, October 23-27, 2016

# Visual Pose Estimation of USV from UAV to Assist Drowning Victims Recovery

Jan Dufek
Department of Computer Science and Engineering
Texas A&M University
College Station, Texas 77843-3112
Email: dufek@tamu.edu

Dr. Robin Murphy
Department of Computer Science and Engineering
Texas A&M University
College Station, Texas 77843-3112
Email: murphy@cse.tamu.edu

*Abstract*—**This paper presents the visual localization subsystem of the ongoing EMILY project. This project aims to assist responders with establishing contact with drowning victims as quickly as possible through the use of a 1.3-meter autonomous unmanned surface vehicle (USV). Once victims are reached, the device can serve as a flotation device to support them. An unmanned aerial vehicle (UAV) provides a live video feed to responders and aids USV's navigation by visually estimating the USV's position and orientation. The movement of the UAV and the USV along with varying lighting, distance, and perspective make the task of estimating USV's position and orientation challenging. We present two implemented solutions for reliable visual localization, the first relying on color thresholding and contour detection, and the second using histograms, back-projection, and CamShift algorithm. The visual localization system was validated in four proof-of-concept field trials. Other unsuccessful methods are discussed.**

## I. Introduction

The main pathway for refugees who attempt to escape war and find asylum in Europe involves a journey across the Mediterranean Sea. For example, the Greek island of Lesbos is separated from Turkey by only 5.5 km.

The large majority of refugees often arrive on overloaded boats that are in poor condition and sometimes do not have a rudder. Many refugees drown each year after falling off the boat, or after their boat sinks. Other boats end up on dangerous shores with hidden currents, big waves, and sharp cliffs.

Rescue boats can guide refugees to safe shores, and individual rescuers can wait in shallow waters near the shore to assist refugees as they disembark. However, the rescue boats cannot get too close to the shore due to shallow water and cliffs. At the same time, individual responders cannot get too far from the shore. Therefore, there is approximately 100 m stripe where refugees are on their own. Furthermore, the number of responders is too low to help everybody.

A solution is to use an unmanned surface vehicle (USV) to operate in otherwise unreachable areas and to assist responders. An USV called Emergency Integrated Lifesaving Lanyard (EMILY) by Hydronalix was tested in teleoperation mode by Murphy et al. at Lesbos in 2016. EMILY is a robotic flotation device that can support up to 6 people.

Teleoperation reduces the number of responders that can assist in person. Therefore, the idea is to make EMILY autonomous and send it to less urgent cases while responders can help refugees who could be facing more urgent issues.

The ongoing autonomous EMILY project proposes to use EMILY in combination with an unmanned aerial vehicle (UAV). The UAV serves two purposes. First, it provides live video feed to the operator who can identify the victims EMILY should assist. Second, the UAV visually estimates EMILY's position and orientation to navigate EMILY to the victims. The operator would use a tablet to circle the victims in the live video feed from the UAV and EMILY would approach them autonomously.

This paper presents the localization subsystem used to determine EMILY's position and orientation from the UAV video feed. EMILY's position and orientation can be used together with the victim's relative position to visually navigate EMILY to reach the victims. The UAV can also use EMILY's position to keep it in its field of view all the time. EMILY stops after victims are reached so that they can hold on to it and wait for responders. EMILY is not able to move when people are holding on to it.

## II. Related Work

This section describes related work on UAV tracking from multiple perspectives. First, it introduces the idea of using a UAV to improve the situational awareness of the USV's operator. Second, it discusses the problem of keeping objects in UAV's field of view. Third, it examines three categories of tracking: single UAV tracking a single object, single UAV tracking multiple objects, and multiple UAVs tracking multiple objects. Fourth, it discusses alternative methods of localization. Finally, it introduces two major algorithms for visual localization.

The idea of using a UAV to support USVs or unmanned ground vehicles is not new. UAVs were used to provide situation awareness for the operator of USV [1]. It was pointed

This work was supported in part by NSF grant OISE-1637214.

out in [2], that autonomy is needed to make the assistance more efficient and to reduce the human-robot ratio.

The problem of keeping tracked objects in the field of view was investigated as well. In [3] the authors proposed an algorithm that tracks multiple ground objects and uses dispersion to keep the optimal viewpoint. Leader following method was proposed in [4], where the UAV keeps the leader marked with artificial tags in its field of view and the other robots stay close to the leader.

Three categories of tracking tasks with UAVs were found in the literature. The first category was one UAV tracking one target. A method to chase uncooperative moving object was proposed in [5]. The algorithm used moving object detection and was tested only with a fixed camera and not actual UAV. Feng [6] presented a vision-based tracking system that can detect a ground target using image processing. His experiment used a static object, but he claimed the algorithm could be easily extended to a moving object.

The second category was a single UAV tracking multiple targets. A method to track multiple ground objects using motion detection and providing this information to ground robots was designed in [7]. USV tracking was used for floating object recovery in [8]. In the work, a USV was tracked from a UAV. Then the relative position and orientation of the USV and target was estimated. A UAV was used to fly over large areas to detect victims and provided their position to a USV [9]. Sometimes, there might be multiple clusters of victims. An algorithm to estimate the probability of successful rescue for each cluster was proposed in [10]. The problem of how to track an object if there were possible occlusions was investigated in [11]. Motion detection method for object tracking was proposed in [12].

The third category was multiple UAVs tracking multiple targets. An algorithm to keep a formation of ground robots in the field of view of multiple UAVs was presented [13]. Ground moving target radar was successfully used for the same task instead of the visual sensor [14]. Most of the research used vertical takeoff and landing platforms. However, fixed wing platforms were investigated as well [15]. The main objective of the study was to keep the group of ground moving targets in the field of view of multiple fixed wing UAVs under the restricted motion assumption. Multiple UAVs were used to track multiple USVs and autonomous underwater vehicles as well [16]. They cooperated on underwater oil spill localization. The spills were detected from UAVs, and their positions were provided to the operators for situation awareness.

Various methods can be used to determine the position of a moving object. GPS and compass are frequently used onboard robots to establish position and orientation, and enable navigation. We are not considering this approach because we do not know victims' GPS coordinates. Victims are selected in the UAV's video feed, therefore their position is known only relative to the image frame. While it is possible to estimate GPS coordinates of an object in the video with the knowledge of UAV's altitude, GPS, camera tilt, and camera parameters, the information may not be available and the estimate may not be precise. By using visual localization of EMILY, both EMILY's and victims' coordinates are in the image frame. One other method facing similar issues is radio-based localization.

There is extensive research focused on detecting an object of interest in the image. Two algorithms most relevant for this study are MeanShift and CamShift. MeanShift algorithm [17], [18] can be used to track the area of maximum pixel density. CamShift [19] is based on MeanShift and can adapt with the changing size and orientation of the tracked area. The input to those algorithms can be, for example, color threshold or histogram back-projection.

Our work, as opposed to the above-mentioned studies, provides a solution for color-based visual localization of a fast moving object from a camera mounted on a moving platform that is not necessarily directly above the target. The studies described above either used fixed camera, tracked a static object, tracked the object by staying directly above it, or used motion-based detection. Motion-based detection is not reliable when the camera is moving. Due to motion parallax static objects will appear to be moving.

## III. Approach

This section describes our application developed for EMILY position and orientation estimation. The problem analysis discusses the previous methods and provides details on the lack of success of those methods.

The desired outcome is to determine coordinates of EMILY in the video and a line segment approximating its orientation and relative size. Therefore, the main question is how to recognize EMILY in the video feed? EMILY is being recognized while in water and moving at speeds of up to 22 mph with the possibility of frequent turns. The video feed is being recorded by a UAV. Therefore, the camera itself may be moving relative to the robot and the environment. The view might be from various angles, not necessarily from the top. The lighting conditions might change frequently due to the relative position to the sun.

All of these assumptions and constraints make the localization of EMILY a challenging task. In the problem analysis subsection, the discussion will outline why certain approaches do not work and then we will present a working solution in the solution subsection.

### *A. Problem Analysis*

Table I summarizes all the approaches and shows the constraints that were satisfied. We will continue our discussion by elaborating why certain approaches do not satisfy certain constraints starting from the first row.

The size of the blob representing EMILY in the image cannot be used. The apparent size might be very different and could change frequently depending on the distance of EMILY from the camera as well as the relative position and orientation of EMILY in relation to the camera. The possible difference in size is illustrated in Figure 1.

Inertia or circularity of the blob representing EMILY in the image cannot be used. These two parameters depend on the

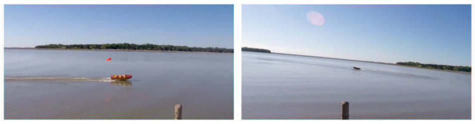


Fig. 1. EMILY size may differ significantly in the image (relative size on the left is much larger than on the right image).

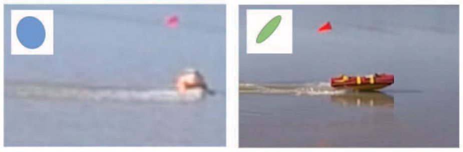


Fig. 2. EMILY inertia may vary from high inertia (left) to low inertia (right).

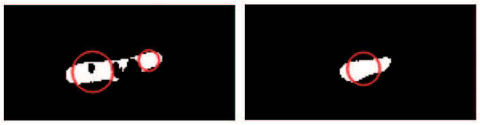


Fig. 3. EMILY's blob can be concave (left) or convex (right) depending on various conditions.

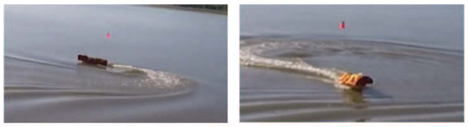


Fig. 4. EMILY brightness scale from dark blob (left) to light blob (right).

relative orientation of EMILY to the camera. If EMILY is facing towards the camera at an oblique angle, it might appear to be circular. If EMILY is turned sideways, the inertia ratio will be low. The example can be seen in Figure 2.

Convexity and concavity of the blob representing EMILY are not constant. The color is not uniform since EMILY is covered with yellow stripes. The non-uniformity might also be caused by shadows, water, or payload. Because of the non-uniformity, the whole body of EMILY might not be recognized as one blob. For this reason, although EMILY is convex in real life, the blob representing it in the image might not be convex. This is illustrated in Figure 3.

We cannot use brightness of EMILY in the image. The brightness may change significantly relative to the time of day, time of year, weather, and relative position to the sun. Brightness also depends on the relative position of the sun to the camera. If the sun is in the camera's field of view, the camera will automatically adjust the exposition making everything look darker. The perceived EMILY color may range from almost black to almost white. The difference is apparent from Figure 4.

In the previously mentioned approaches, we assumed the algorithm uses only one video frame at a time. We hypothesized that if we compare frames and extract significant changes in pixel values across multiple frames, we would be able to detect EMILY's movement. However, this method proved to be usable only under the assumption that the camera was fixed. This is unfortunately not the case for the camera mounted on a UAV. If the camera is moving by itself, it will cause the algorithm to detect stationary objects as moving, because it does not know that the camera is moving. Moreover, this background movement is not uniform and cannot be compensated for by the algorithm. Due to parallax motion, the objects which are closer to the camera will appear to move faster than objects further away from the camera.

TABLE I
COLUMNS ARE CONSTRAINTS AND ROWS ARE APPROACHES. "YES" SIGNIFIES THAT THE APPROACH SATISFIES THE CONSTRAINT.

| | Variable Size | Variable Inertia | Variable Convexity | Variable Brightness | Moving Camera |
|---|---|---|---|---|---|
| Size | No | Yes | Yes | Yes | Yes |
| Inertia | Yes | No | Yes | Yes | Yes |
| Convexity | Yes | Yes | No | Yes | Yes |
| Brightness | Yes | Yes | Yes | No | Yes |
| Movement | Yes | Yes | Yes | Yes | No |
| Color | Yes | Yes | Yes | Yes | Yes |

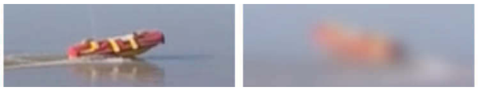


Fig. 5. Application of Gaussian blur on EMILY image to smooth the surface.

### B. Solution

After failed attempts with the previously mentioned approaches, we settled on color-based detection for our visual localization solution satisfying all the previously mentioned constraints. EMILY itself has a very bright red color. We came up with two working solutions — simple thresholding on hue values, and histogramming with back-projection and CamShift algorithm.

There are two major problems with color-based detection. First, the color of EMILY is not uniform. It has a white cross on top and numerous yellow stripes on its body, which covers a significant portion of the red. It also has black handles, colorful labels, white ropes, and black radio transceiver. Its non-uniformity could cause the surface to be broken down into multiple small red blobs. Most of the non-red parts have an important function and cannot be removed. This problem can be alleviated by using Gaussian blur to diffuse the color of the additional elements in the red. The application of Gaussian blur is illustrated in Figure 5.

The second problem is that hue is difficult to define in RGB color space used by the camera. The values might be different in all three color coordinates and still, have the same hue. It is also problematic to capture changes in saturation and intensity of particular hue. Therefore, it is challenging to set intervals for the color that need to be extracted. Furthermore, different lighting significantly affects RGB values for the same object, making it difficult to detect a single color. One solution might be the conversion of RGB color space to grayscale, which has just one dimension. This, however, leads to extreme loss

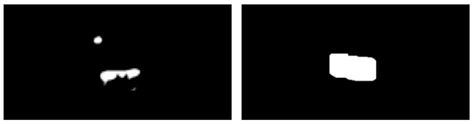


Fig. 6. Threshold (left) and application of erosion and dilatation (right).

of information. The color space is reduced from $16,777,216$ colors to $256$ colors. EMILY might not be distinguishable from the background in that case.

The solution to this problem is the conversion to HSV color space. HSV color space specifies a color in terms of hue, saturation, and value. In this color space, we can specify the hue range of the color we want to extract. It also enables to filter based on saturation and value. The hue in HSV is also less affected by lighting than in RGB.

Let us continue with two aforementioned methods to localization.

*1) Thresholding:* The first method is to apply thresholding to select pixels lying within the specified range of HSV values. The thresholding method will create a binary map where 1 (white) means that the corresponding pixel is within the specified range, and 0 (black) otherwise. The example of a threshold for EMILY can be seen on the left side of Figure 6.

The thresholding as described above inherently contains some noise. This noise is reduced with morphological erosion. Erosion will delete a certain number of pixels that border with black pixels. It will ultimately delete very small dispersed clusters and leave just larger clusters.

After erosion, we apply dilatation to amplify the clusters and also fill-in possible holes in the clusters. Holes might occur due to non-uniformity of EMILY's surface and also other factors described above. The example of threshold after applying erosion and dilatation is on the right side of Figure 6.

The next step is to detect blobs in the threshold. Two blob detection approaches were examined. The first is *OpenCV SimpleBlobDetector*. It is able to detect blobs based on color, area, circularity, convexity, and inertia. As explained above, we cannot use those filters since EMILY's image might differ significantly in those parameters. Therefore, the algorithm will usually choose all the blobs in the image.

The second approach is to find contours (boundaries of self-contained shapes) in the thresholded binary image. We can then compute various moments on the contours, such as area, shape size, and shape center. The contours method might still return multiple objects. Our algorithm selects the largest one. After selecting the biggest shape, we compute its centroid and draw an indicator on the screen. The centroid coordinates are the ultimate output of the algorithm.

*2) Histogramming:* The second method is to use histogramming. The main idea is to use a histogram instead of a simple hue range. At the beginning, our application will let the user select EMILY in the image. It will then construct the histogram of hue values. A histogram specifies for each range of hue values how many pixels in that range are contained within the selected area. Therefore, it is a much more precise description than the simple hue range. The example of a histogram for EMILY can be seen in Figure 7. We still have to specify saturation and value intervals manually.

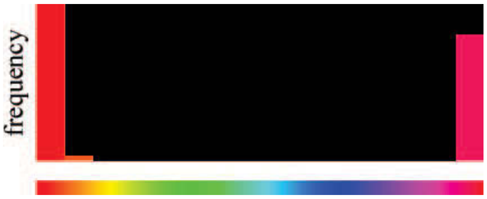


Fig. 7. EMILY histogram constructed automatically from the selected image area. The horizontal axis is the hue and the vertical axis is the frequency.

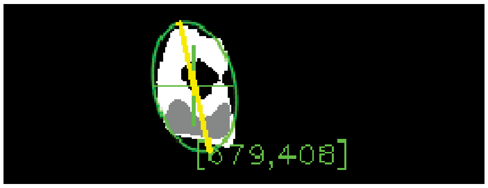


Fig. 8. EMILY histogram back-projection.

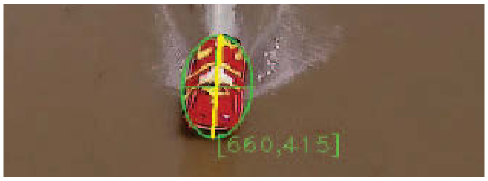


Fig. 9. EMILY localization using histogram back-projection and CamShift algorithm during the second trial. Green cross represents the centroid, the ellipse approximates the shape, and the yellow line represents the orientation.

After we have the histogram, we can calculate back-projection of each frame. This will create a grayscale map where each pixel's brightness represents how much this pixel's hue is represented in the histogram (in other words how well the pixel fits the histogram distribution). The back-projection is illustrated in Figure 8.

Now we can apply advanced CamShift algorithm [19] for finding objects. It finds the area of maximum pixel density by sliding a window towards the centroid of all the pixels in that window. This is iterated until the algorithm finds the maximum density area. The window size is adapted with each iteration dynamically to the size of the target. This algorithm not only finds the centroid of EMILY, but it also fits an ellipse to EMILY's shape which can be used later for orientation estimation. The result is illustrated in Figure 9.

### C. Orientation Estimation

So far, we discussed only how to get EMILY's position. Now we will take a look at how to estimate the orientation. If we can estimate the orientation from the image, we can navigate the robot to go to the particular direction in the coordinate system with the origin in its centroid.

Two classes of orientation estimation methods were considered. The first class uses movement to estimate the heading.

Since EMILY can only move forward, the direction of movement should represent the front part of EMILY. Unfortunately, this technique would work only for a fixed camera. However, our camera is mounted on a UAV. This might result in blob motion caused by the camera movement and not by the movement of blob itself.

The second class is based on the blob shape itself. The major axis of the blob should give a good orientation approximation.

There are many techniques that can be used to estimate the principal axis. The first method we tried was to fit a line through the blob. That method was more suitable for multiple data points and did not work well for a single blob.

The second method was to fit a rectangle with the minimum area that would contain the blob; however, that approach was not very accurate. The rectangle would oftentimes fit in a way that EMILY would be enclosed diagonally, so the orientation would be estimated incorrectly.

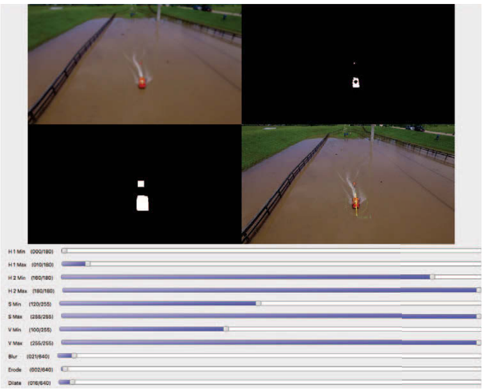
The third method was principal component analysis. This algorithm is designed to find a principle component of the data. However, it suffered from the similar problems as fitting a line. It implicitly required more data points to work correctly and did not work well with a single blob.

The fourth and best solution was to fit an ellipse with the minimum area to the blob. The fitting will cause the vertices of the ellipse to be on both ends of the robot. Then, we can compute major axis of the ellipse to approximate robot's orientation and relative size of the robot in the image. That approach worked better than fitting a rectangle because it eliminated the extra space at the corners of the rectangle. Moreover, if we use CamShift algorithm, we already have the minimum fitting ellipse, so we save some computing power. The example of the orientation estimation is the yellow line in Figure 9.

## IV. Implementation

All aforementioned approaches were implemented in *C++* using *OpenCV* library. The input video files were either prerecorded MP4 or MOV in high definition or RTSP video stream from a UAV. UAV video stream came from UAV's control tablet using screen mirroring application since the used UAVs cannot stream to a computer directly. In order to reduce computational complexity, the input frames were resized to a lower resolution.

Graphical user interface (GUI) was implemented. It enables important parameters to be set and also the frames in various phases of the processing to be visible. The parameters can include (depending on selected method) HSV ranges, blur level, erosion level, and dilatation level. This is critical because the parameters might depend on various outside conditions such as weather or time of day, and cannot be fixed. The GUI can be seen in Figure 10. The application automatically records the localization and exports it as AVI video file.



H 1 Min (000/180)
H 1 Max (010/180)
H 2 Min (180/180)
H 2 Max (180/180)
S Min (120/255)
S Max (255/255)
V Min (100/255)
V Max (255/255)
Blur (021/640)
Erode (002/640)
Dilate (016/640)

Fig. 10. Graphical user interface of the implemented application. Top left is the blurred image, top right is the threshold, bottom left is the application of erosion and dilatation, bottom right is the resulting position and orientation information, and the very bottom are sliders with parameters.

TABLE II
Average localization error in pixels. H is histogramming and T is thresholding method. Resolution is input video resolution in pixels.

| | Trial 1 | | Trial 2 | | Trial 3 | | Trial 4 | |
|---|---|---|---|---|---|---|---|---|
| | **H** | **T** | **H** | **T** | **H** | **T** | **H** | **T** |
| **Error** | 15 | 440 | 8 | 15 | 7 | 311 | 43 | 100 |
| **Resolution** | 1920x1080 | | 1920x1080 | | 2132x1200 | | 1024x640 | |

## V. Trials

Four proof-of-concept field trials were conducted to validate the visual localization system. The data were manually annotated to mark true coordinates of EMILY in the video and compared with algorithm output coordinates to measure estimation error. Table II shows the error computed as average distance in pixels between true coordinates and output coordinates from the algorithm. The table also lists the resolution of the videos used in the trials so that the trials can be compared to each other.

The first trial was held at Lake Bryan, Bryan, TX on March 28th. EMILY was controlled manually, and the video was taken from an elevated position. The video camera used was GoPro HERO4. This was a preliminary experiment, so the data was not collected from a UAV. We determined two challenges. First, GoPro has 15 mm focal length (ultra wide field of view) and therefore objects in the distance were very small. Therefore, EMILY would only be represented by few pixels if it is in the distance. This problem can be alleviated by using lens with higher focal numbers. The second challenge encountered was the sun facing directly into the camera. This will cause the camera to change the white balance and distort the brightness in the image. Therefore, EMILY's color might be desaturated to the color of the environment. This is illustrated in Figure 11, where EMILY has exactly the same color as the water in another part of the image. This problem, however, is less likely to happen for a UAV, since the UAV camera is facing downwards.

The second trial was performed during Fort Bend County, TX floods on April 23rd. EMILY was controlled in manual mode. DJI Inspire 1 was used to get low altitude video of EMILY. Both EMILY and the UAV moved with frequent

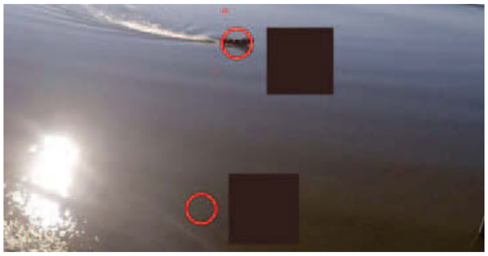


Fig. 11. In this sample image, EMILY has the same color as the water in another area of the image.

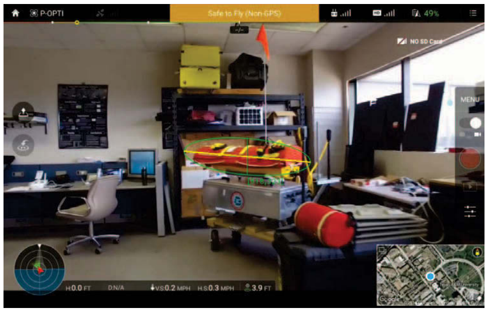


Fig. 12. Testing the reliability of position and orientation estimations under various viewpoints. View from UAV.

changes in direction. UAV was chasing EMILY and changed view angles frequently. The weather was sunny with no cloud cover, and the water was dirty flood water. Figure 10 presents the GUI while processing video from this trial using the thresholding algorithm. Figure 9 is a snapshot from the video from this trial processed using the histogram algorithm. We encountered a problem with flood water having desaturated red color. However, this was easily filtered by raising the lower limit for saturation.

The third trial was conducted at Lake Bryan, Bryan, TX on May 10th. EMILY was used in autonomous waypoint navigation mode. DJI Phantom 3 Professional was used to get high altitude video from the shore. The UAV was not chasing EMILY, but rather was looking from above while still directly above the shore, simulating the intended use. The weather was cloudy causing EMILY to go from the sun to shadow.

The fourth trial was conducted in our laboratory and was designed to try various viewpoints. EMILY was stationary, and the UAV was carried around to observe EMILY from various angles. The example of the position and orientation estimations can be seen in Figure 12.

Each trial required certain modifications of the parameters to account for local conditions (weather, time of day, water color). This was done easily using the GUI and usually took a couple of minutes. As can be seen from Table II, the histogram method generally worked better than hue thresholding because it enables the algorithm to work with more information about the object and also to dynamically create the histogram in local lighting conditions. Furthermore, histogramming method requires fewer parameters to be set in the GUI (only saturation and value limits) making the process more scalable.

## VI. Future Work

One area of future work to be explored will be the use of feature tracking for object detection, allowing to use structure as well as color for improved robustness. A potential challenge we anticipate is perspective, as EMILY may be seen by the UAV from various sides.

Next, we want to automate the parameter tuning. In the case of histogramming method, this would require estimating saturation and value that would separate the object of interest from the background.

More importantly we would like to continue to fully integrate the position and orientation estimation into the entire system. First, the control signals will be issued back to the UAV to tilt the camera and possibly to change position so that EMILY is kept in the field of view. Second, the position and orientation of EMILY together with victim's position will be used to navigate EMILY to the victims.

In its final form, first responders will view EMILY from UAV video camera and will be able to circle victims. EMILY will then autonomously navigate to the victims using data from the UAV and our application.

## VII. Conclusion

Responders need a solution to assist them with saving refugees without sacrificing manpower and to enable them to operate in unreachable areas. The ongoing autonomous EMILY project proposes using an autonomous USV cooperating with a UAV to serve that purpose. In this system, the operator would see video feed from UAV, circle refugees, and EMILY would autonomously navigate there.

This paper presented the subsystem for localizing EMILY in the video feed from a UAV to estimate its position and orientation. The information is necessary to enable robust navigation. The discussion provided complete analysis of the problem including solutions that did not work and explanations of why.

We presented our application for reliable visual localization. The video images are preprocessed first using Gaussian blur, conversion to HSV, and equalization on HSV values. Next, two possible methods were proposed. The first method is based on color thresholding, erosion, dilatation, and contour blob detection. The second method is based on histogramming, back-projection, and CamShift algorithm.

The orientation is estimated by fitting an ellipse with the minimum area and taking its principal axis. Other methods that did not work were discussed as well. Those include motion detection, line fitting, minimum rectangle fitting, and principal component analysis.

The proposed solution was implemented in *C++* and *OpenCV* including graphical user interface to view multiple frames at various stages of processing and to set important parameters. Four experimental field trials were conducted as proofs-of-concept for the localization system. The first was a hand-held outdoor test with a GoPro HERO4. The second was an outdoor field test at Fort Bend with a DJI Inspire One. The third was at Lake Bryan with a DJI Phantom 3 Professional.

The final trial was also with the DJI Phantom 3 Professional, but was conducted indoors to test different viewpoints. The results of those trials demonstrated the viability of the presented localization subsystem and will help direct our efforts for future work.

## Acknowledgment

I would like to thank my advisor, Dr. Robin Murphy, and all the fellow students working on other parts of the autonomous EMILY project. In alphabetical order Matt Hegarty, Jesus Orozco, Tim Woodbury, and Xuesu Xiao. I would also like to thank Justin Adams for providing the data from DJI Inspire 1 in the second field trial.

## References

[1] R. Murphy, S. Stover, K. Pratt, and C. Griffin, "Cooperative damage inspection with unmanned surface vehicle and micro unmanned aerial vehicle at hurricane wilma," in *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, Conference Proceedings, pp. 9–9.

[2] M. Lindemuth, R. Murphy, E. Steimle, W. Armitage, K. Dreger, T. Elliot, M. Hall, D. Kalyadin, J. Kramer, M. Palankar, K. Pratt, and C. Griffin, "Sea robot-assisted inspection," *IEEE Robotics & Automation Magazine*, vol. 18, no. 2, pp. 96–107, 2011.

[3] M. Cognetti, G. Oriolo, P. Peliti, L. Rosa, and P. Stegagno, "Cooperative control of a heterogeneous multi-robot system based on relative localization," in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, Conference Proceedings, pp. 350–356.

[4] E. H. C. Harik, F. Guerin, F. Guinand, J. F. Brethe, and H. Pelvillain, "Uav-ugv cooperation for objects transportation in an industrial area," in *Industrial Technology (ICIT), 2015 IEEE International Conference on*, Conference Proceedings, pp. 547–552.

[5] W. Ding, Z. Gong, S. Xie, and H. Zou, "Real-time vision-based object tracking from a moving platform in the air," in *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, Conference Proceedings, pp. 681–685.

[6] L. Feng, M. C. Ben, and L. K. Yew, "Integration and implementation of a low-cost and vision-based uav tracking system," in *2007 Chinese Control Conference*. IEEE, 2007, pp. 731–736.

[7] Z. Sun, "Vision-assisted adaptive target tracking of unmanned ground vehicles," in *Control Conference (CCC), 2014 33rd Chinese*, Conference Proceedings, pp. 3685–3690.

[8] N. Mikovi, S. Bogdan, . Na, F. Mandi, M. Orsag, and T. Haus, "Unmanned marsupial sea-air system for object recovery," in *Control and Automation (MED), 2014 22nd Mediterranean Conference of*, Conference Proceedings, pp. 740–745.

[9] F. F. Ramirez, D. S. Benitez, E. B. Portas, and J. A. L. Orozco, "Coordinated sea rescue system based on unmanned air vehicles and surface vessels," in *OCEANS, 2011 IEEE - Spain*, Conference Proceedings, pp. 1–10.

[10] T. Tao and R. Jia, "Uav decision-making for maritime rescue based on bayesian network," in *Computer Science and Network Technology (ICCSNT), 2012 2nd International Conference on*, Conference Proceedings, pp. 2068–2071.

[11] Q. Gao, Z. C. Zeng, and D. Hu, "Long-term tracking method on ground moving target of uav," in *Guidance, Navigation and Control Conference (CGNCC), 2014 IEEE Chinese*, Conference Proceedings, pp. 2429–2432.

[12] M. Siam and M. ElHelw, "Robust autonomous visual detection and tracking of moving targets in uav imagery," in *Signal Processing (ICSP), 2012 IEEE 11th International Conference on*, vol. 2, Conference Proceedings, pp. 1060–1066.

[13] M. Aranda, L. N. G, C. Sagues, and Y. Mezouar, "Formation control of mobile robots using multiple aerial cameras," *IEEE Transactions on Robotics*, vol. 31, no. 4, pp. 1064–1071, 2015.

[14] A. Sinha, T. Kirubarajan, and Y. Bar-Shalom, "Autonomous ground target tracking by multiple cooperative uavs," in *Aerospace Conference, 2005 IEEE*, Conference Proceedings, pp. 1–9.

[15] H. Oh, S. Kim, H. S. Shin, and A. Tsourdos, "Coordinated standoff tracking of moving target groups using multiple uavs," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 51, no. 2, pp. 1501–1514, 2015.

[16] A. Vasilijevic, P. Calado, F. Lopez-Castejon, D. Hayes, N. Stilinovic, D. Nad, F. Mandic, P. Dias, J. Gomes, J. C. Molina, A. Guerrero, J. Gilabert, N. Miskovic, Z. Vukic, J. Sousa, and G. Georgiou, "Heterogeneous robotic system for underwater oil spill survey," in *OCEANS 2015 - Genova*, Conference Proceedings, pp. 1–7.

[17] K. Fukunaga and L. Hostetler, "The estimation of the gradient of a density function, with applications in pattern recognition," *IEEE Transactions on information theory*, vol. 21, no. 1, pp. 32–40, 1975.

[18] Y. Cheng, "Mean shift, mode seeking, and clustering," *IEEE transactions on pattern analysis and machine intelligence*, vol. 17, no. 8, pp. 790–799, 1995.

[19] G. R. Bradski, "Computer vision face tracking for use in a perceptual user interface," 1998.

[20] Z. Henkel, J. Suarez, B. Duncan, and R. R. Murphy, "Sky writer: sketch-based collaboration for uav pilots and mission specialists," in *Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction*. ACM, 2014, pp. 172–173.