Victim Detection from a Fixed-wing UAV: Experimental Results

Anurag Sai Vempati, Gabriel Agamennoni, Thomas Stastny, and Roland Siegwart

Autonomous Systems Lab, ETH Zurich
http://www.asl.ethz.ch/

Abstract. This paper outlines a method to identify humans from a low-altitude fixed-wing UAV relying on various visual and inertial sensors including an infrared camera. The work draws inspiration from the need to detect victims in disaster scenarios in real-time, providing needed aid to rescue efforts. Such work can also be easily employed for surveillance related applications. We start by pointing out various challenges that arise due to camera imperfections, viewpoint, altitude, synchronization etc. We provide a pipeline to efficiently fuse thermal and visual aerial imagery and get robust real-time detections. Confident detections are tracked across various frames and the real-time GPS locations of the victims is conveyed. Performance of our detection algorithm is evaluated in a real-world victim detection scenario from an autonomous fixed-wing aircaft.

1 Introduction

Search and Rescue is a widely researched field owing to it's numerous applications in disaster scenarios. In cases like avalanches, a few minutes could make considerable difference in having better probability of suppressing the casualties. With increasing autonomy of the Unmanned Aerial Vehicles (UAV) and camera imaging technologies, it is now possible to scan large areas in a very short time and perform perception algorithms on-board at high rates with close to zero human intervention. Such technology also enables surveying in-accessible regions and hostile terrains making the task of dispatching rescue efforts considerably easy.

Visual spectrum cameras have been extensively used on UAVs, however, analyzing these images at high rates requires very robust algorithms to deal with various difficulties posed due to the size of objects of interest, motion blur, and viewpoint, to name a few. Detecting humans from a UAV cruising at an altitude of 50-100 meters requires very high resolution cameras and the ability to quickly detect objects occupying few tens of pixels in area. On the other hand thermal cameras offer an advantage in such cases which makes it easier to narrow down the search space to hotter objects. But thermal cameras have their own limitations like low Signal-to-Noise Ratio (SNR), white-black/hot-cold polarity changes, and halos that appear around very hot or cold objects [1]. We propose a technique that best utilises the pros of either cameras using sensor fusion technique.

Various works leverage on hotspot techniques to quickly narrow down the potential areas to further process but they usually perform poorly in case of fast moving cameras like on UAVs. Many techniques rely on some kind of classifier to detect presence of a human in each of these potential areas. [2], [3], [4], [1] use some kind of variant of cascade of boosted classifiers introduced by Viola and Jones [5] - which basically involves a series of weak classifiers each better than the previous one. [6] show performance of various feature based classifiers trained on thermal data collected across wide variation in temperature, altitude and camera movement. According to their work, Histogram of Gradients (HOG) feature based classifier in conjunction with a particle filter tracker was found to perform the best.

Sensor fusion and multi-modal image registration is a well explored field. But most of the works involving registration of infrared and visual camera images like [7], [8], [9], [10] involves image processing techniques that rely on feature extraction, edge detection, segmentation etc. Techniques like such, though very robust, can be quite time-consuming for applications like ours. On the other hand, [4] uses camera intrinsics and ground planarity assumption to estimate relevant part of visual camera image for stationary victims. This method has it's own limitations due to additional criteria enforced on targets' positions and the environment being scanned. In our work we make use of camera extrinsics and use techniques from multi-view geometry to get one-to-one correspondence between infrared and visual images.

In this paper we describe an algorithm that can efficiently detect stationary victims while autonomously scanning large areas using a UAV equipped with various sensors including visual and thermal cameras and an on-board computer to perform real-time computations. We will briefly outline individual components involved and provide results on a field experimental test.

2 Victim Detection Pipeline

Detecting humans from an altitude of 50-100 meters with a camera of limited resolution poses a very challenging problem. Basic blocks of our pipeline are mentioned in the following sub-sections and in the next section we illustrate a real-case scenario.

2.1 Background Subtractor

Fig. 1 shows a human as seen in false-color rendering of the thermal camera image at an altitude of about 70 meters. At this scale, the humans occupy less than 50 pixels (<0.02%) in an image of 640 x 512 resolution. An exhaustive search for such a tiny object of interest is very time consuming. We propose a Background Subtractor that returns regions of interest (ROI) and narrows down the search space considerably, thus enabling real-time detection. The foreground here is defined as a part of the image whose temperature differs quite significantly from it's surroundings. Since, humans are usually hotter (in winter) or colder (in summer) than the surroundings, we propose a technique that adaptively adjusts 2 threshold values (t_{low}, t_{high}) based on the surrounding pixel intensities. All the pixels with intensities less than t_{low} or greater than t_{high} are considered as foreground pixels.

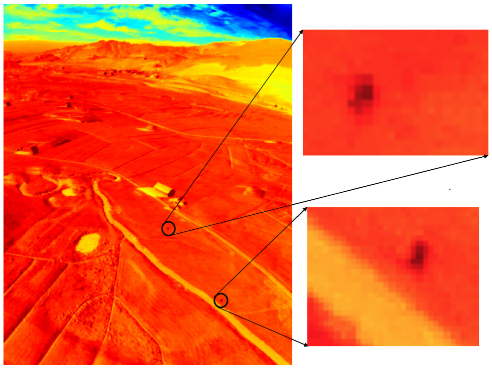
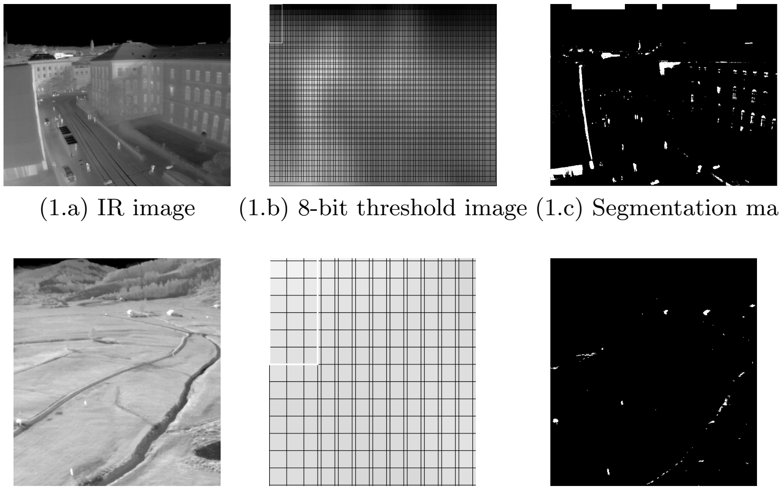


Fig. 1: A sample image recorded from the FLIR thermal camera at an approximate altitude of 70 meters. Two humans are pointed out.

We employ a very simple sliding window based approach to estimate adaptive thresholds. The image is divided into various overlapping blocks and the adaptive thresholds t^b_{low}, t^b_{high} for each block b is evaluated as the higher and lower quantiles of the Gaussian models fitted to the pixel intensities of the block. Now, the threshold values at each pixel location (i, j) are chosen as weighted average of all the block thresholds t^b_{low}, t^b_{high} if the pixel belongs to block b. The weights are chosen inversely proportional to the pixel's distance from the block's center. A segmentation map (segmap) is generated by thresholding the infrared image at each pixel location.

Fig. 2 shows two sample images collected at different altitudes, temperature and times of the day and the corresponding segmaps. A blob detection algorithm [11] is used to find blobs of desired size, thus providing ROIs to search for humans. For the UAV scenario, we further narrow down the search space by considering only the blobs whose area lies in certain range that is calculated at every time-step based on the camera specifications, mounting, UAV's IMU pose and GPS position. This approach is computationally much cheaper and is not affected by the fast moving camera. It provides much faster and better results at real-time than many conventional approaches.



(1.a) IR image (1.b) 8-bit threshold image (1.c) Segmentation map

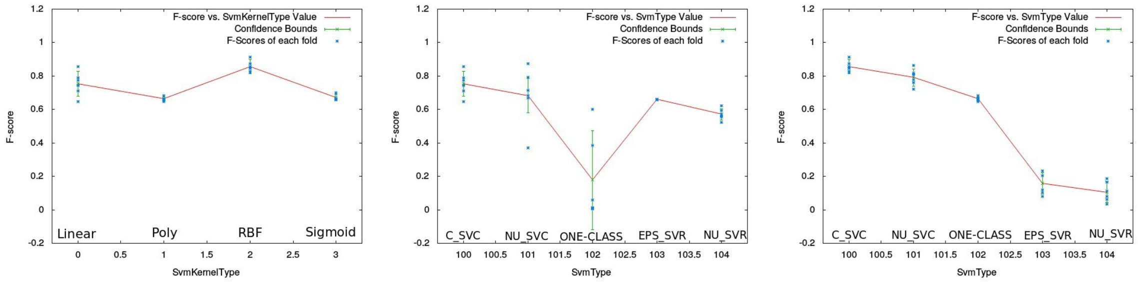
(2.a) IR image (2.b) 8-bit threshold image (2.c) Segmentation map

Fig. 2: Sequence 1 is captured in the night time at an altitude of 20 meters above the ground with some crowd. Sequence 2 was captured from a UAV on a cold winter morning at an altitude of 60 meters above the ground. (a) infrared image, (b) 8-bit image of pixel-wise threshold $t_{high}(i, j)$ (grid of blocks is overlaid and a sample block is marked in white on the top-left) and (c) segmentation map depicting regions of interest.

2.2 Human Classifiers

The ROIs obtained from Background Subtraction are exhaustively searched for presence of a human. For this, we use a HOG feature based learning classifier (which was found to be optimal [6]) to classify the extracted patch into human or non-human category. The training data is obtained by manually annotating sequences from different datasets using vBBToolbox [12]. HOG features are extracted on the image patches at multiple scales to generate a descriptor. A Support Vector Machine (SVM) classifier is trained using these image patch descriptors. Type of SVM optimization problem, kernel type and other parameters are optimized using K-fold cross validation. Fig. 3 shows performance of few chosen configurations.

Fig. 3 shows Precision-Recall curves for the performance of classifier on a validation dataset. The above method is then repeated to train a classifier for grayscale image patches. C_SVC [13] optimization problem with Linear kernel type and Radial Basis Function (RBF) kernel type are found to be two top performing configurations for infrared classifier. C_SVC optimization problem with Linear kernel is found to be optimal configuration for grayscale classifier.



(a) Choosing kernel type for C_SVC problem (b) Choosing optimization for Linear kernel (c) Choosing optimization for RBF kernel

Fig. 3: Performance of infrared SVM classifier with different configurations of optimization problem and kernel type. It's found to be optimal to choose (a) RBF kernel type for C_SVC problem, (b) C_SVC problem for Linear kernel type and (c) C_SVC problem for RBF kernel type.

2.3 Infrared-Grayscale Image Fusion

Objects in infrared image rarely have discerning features since the temperature tends to be uniform across the objects. So, a classifier would be not so reliable in classifying such patches since it only has the shape information of the object in a very small patch. Since in all our flights using the UAV we collected both infrared and grayscale images, a fusion technique is proposed to get a one to one relationship between pixel locations of infrared and grayscale images. This will help us in extracting both infrared and grayscale image patches corresponding to the ROIs obtained from Background Subtraction which can help the classifier in producing more reliable results.

We use Kalibr [14] to calibrate the infrared and grayscale camera intrinsics and extrinsics for the sensorpod mounting. We use radial-tangential model to estimate camera distortion parameters. An april tag pattern is used to estimate grayscale camera intrinsics. A thick matte paper with checkerboard pattern under illumination of a bright light is used to calibrate infrared camera intrinsics. Now the infrared and grayscale camera setup can be treated as a Stereo pair and standard image rectification techniques [15] can be used to evaluate the necessary projection matrices. At this point, all the epipolar lines are parallel to image edge (horizontal if horizontal rectification was employed, vertical otherwise). For our particular setup, vertical stereo was chosen since the camera centers were aligned vertically on the UAV. Now that the disparities of the image are in one axis, it's fast and easy to find pixel correspondences using robust feature matching between the images. A pseudo source code of the image fusion algorithm is listed in List. 1.1.

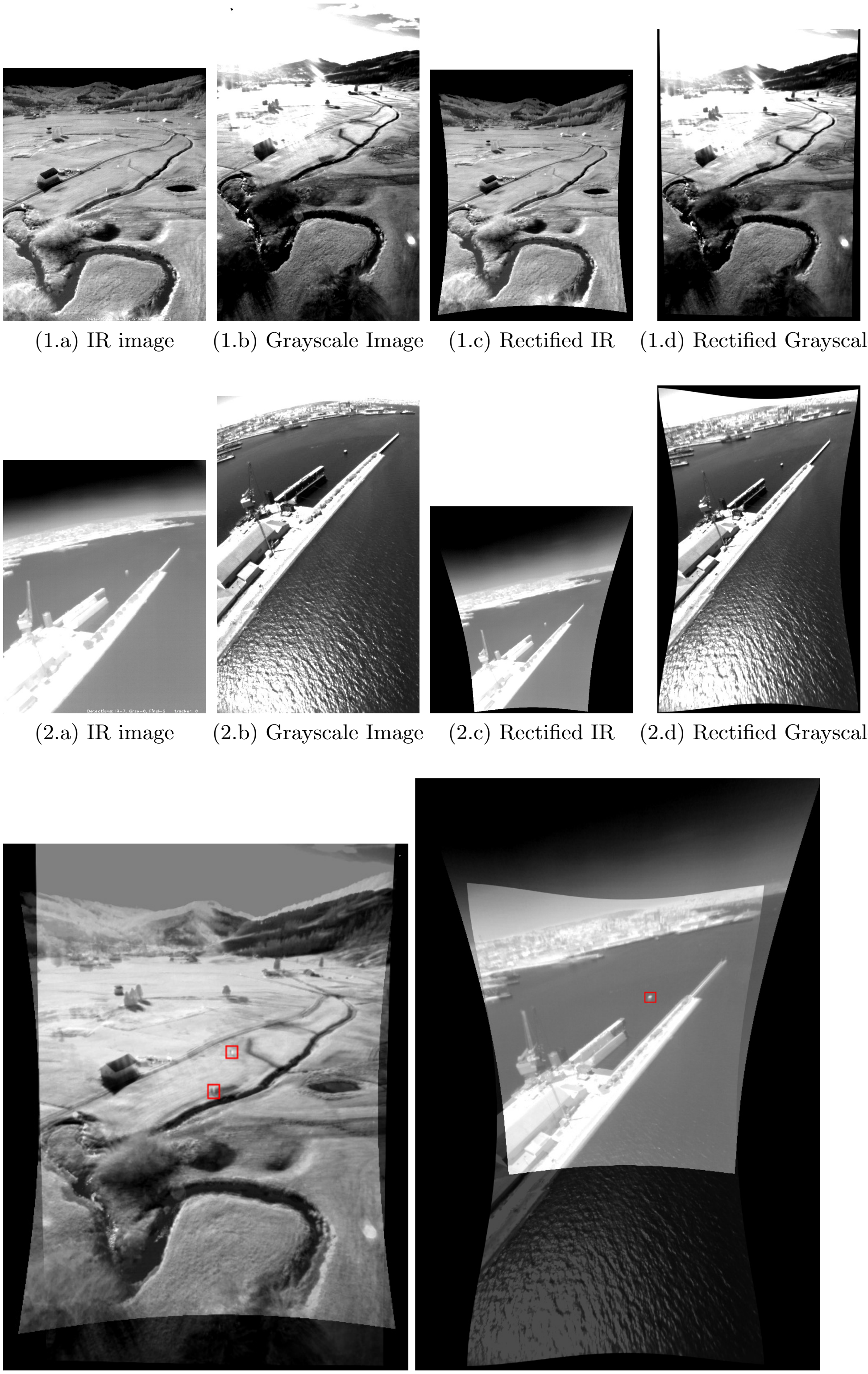
Once a one-to-one correspondence is obtained between grayscale and infrared images, the patches corresponding to ROIs are passed on to individual human classifiers and pair of patches resulting in high cumulative classifier score are retained for probable presence of a human.

Fig. 4 shows results from the fusion technique. In Sequence-1 infrared and grayscale cameras were both facing in the direction of flight and are in 25° nadir configuration with a translation of 3.5cm between the camera centers. In Sequence-2, infrared camera was in 25° nadir configuration while grayscale camera was in 50° nadir configuration with a translation of 3.5cm between the camera centers. The results show reasonable overlap accuracy.

Fig. 5 shows improvement to the Precision-Recall (PR) curves after image Fusion on a dataset collected from a UAV cruising at 60 meters above the ground. Close to 140% improvement (10% vs. 24%) in the area under curve can be seen. Since, we have various methods to eliminate false positives, as mentioned in next section, we operate our classifier on the higher recall point. Also the fact that very few of the false positives are of a same object recurring across several frames plays a vital role in suppressing them.

```
void main (int num_arguments, char** arguments) {
  struct Intrinsics {  // Camera Intrinsics.
    cv::Mat_<double> infrared;
    cv::Mat_<double> infrared_distortion;
    cv::Mat_<double> grayscale;
    cv::Mat_<double> grayscale_distortion;
  } intrinsics;
  struct Extrinsics {  // Camera Extrinsics.
    cv::Mat_<double> grayscale_to_infrared;
  } extrinsics;
  struct Rectified {  // Rectified image data.
    // Transformation matrices for rectification.
    cv::Mat_<double> grayscale_transform;
    cv::Mat_<double> infrared_transform;
    // New optimal camera matrices.
    cv::Mat_<double> new_grayscale_intrinsics;
    cv::Mat_<double> new_infrared_intrinsics;
    cv::Mat_<double> intrinsics;
  } rectified;
  struct Mappings {  // Transformation mappings.
    // Transformation maps for rectified images. transformed_image(x, y) =
     image(map(x, y)).
    cv::Mat infrared;
    cv::Mat grayscale;
    bool evaluated = false;
    cv::Mat offsets;  // Offsets to align rectified images.
  } mappings;

  loadCalibration(intrinsics,extrinsics);  // Load camera intrinsics.
  // Generate new optimal camera intrinsics that will ensure maximum possible
      number of pixels of both images are retained after un-distortion, in
     destination image size same as the original image.
  rectified.new_infrared_intrinsics = cv::getOptimalNewCameraMatrix(
    intrinsics.infrared,intrinsics.infrared_distortion,infrared_image_size);
  rectified.new_grayscale_intrinsics = cv::getOptimalNewCameraMatrix(
    intrinsics.grayscale,intrinsics.grayscale_distortion,grayscale_image_size
     );
  // Obtain single optimal camera matrix that ensures all the pixels from
     both the images are retained post un-distortion.
  rectified.intrinsics = getOptimalNewCameraMatrix(
    rectified.new_infrared_intrinsics,rectified.new_grayscale_intrinsics);
  // Compute rectification transforms for each camera in the calibrated
     stereo pair using OpenCV Stereo Rectification.
  cv::stereoRectify(
    intrinsics.infrared,intrinsics.infrared_distortion,
    intrinsics.grayscale,intrinsics.grayscale_distortion,
    extrinsics.grayscale_to_infrared,rectified.infrared_transform,
```



(1.a) IR image (1.b) Grayscale Image (1.c) Rectified IR (1.d) Rectified Grayscale

(2.a) IR image (2.b) Grayscale Image (2.c) Rectified IR (2.d) Rectified Grayscale

(1.e) Fused IR and Grayscale Images (2.e) Fused IR and Grayscale Images

Fig. 4: The top 2 rows show 2 sequences of original images obtained from (a) infrared, (b) grayscale cameras and (c, d) the resulting images post rectification. Last row shows fusion of both images (e) to visualize the quality of overlap. Red boxe highlights 2 humans in Sequence-1 and a boat in Sequence-2. As can be seen, the correspondence between infrared and grayscale images is reasonably good.

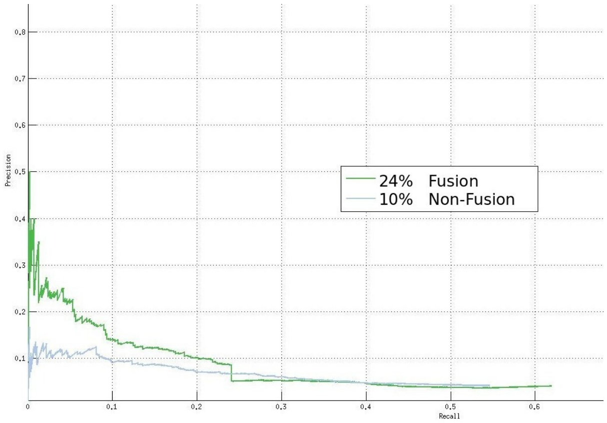


Fig. 5: Precision-Recall curves of the pipeline on a high-altitude (60 meters) data collected from a UAV, before and after image fusion.

```
    rectified.grayscale_transform);

  while (incoming ROS messages) {
    // Read ROS infrared image message.
    if (message_view.getTopic() == infrared_topic) {
      cv::Mat infrared = getImage(message_view); // Read the image.
    }
    // Read ROS grayscale image message.
    if (message_view.getTopic() == grayscale_topic) {
      cv::Mat grayscale = getImage(message_view); // Read the image.
      // Evaluate transformation maps using computed destination camera
     matrix and the stereo rectification transforms.
      if ( !mappings.evaluated ) {
        cv::initUndistortRectifyMap(
          intrinsics.infrared,intrinsics.infrared_distortion,
          rectified.infrared_transform,rectified.infrared_intrinsics,
          rectified.destination_size,mappings.infrared);
        cv::initUndistortRectifyMap(
          intrinsics.grayscale,intrinsics.grayscale_distortion,
          rectified.grayscale_transform,rectified.infrared_intrinsics,
          rectified.destination_size,mappings.grayscale);
        mappings.evaluated = true;
      }
      // Apply offsets to transformation maps using robust feature matching.
      cv::goodFeaturesToTrack(infrared,infrared_features);
      cv::goodFeaturesToTrack(grayscale,grayscale_features);
      findOffsets(infrared_features,grayscale_features,mappings.offsets);
      applyOffsets(mappings.offsets,mappings.infrared,mappings.grayscale);
      // Rectify the images.
      cv::remap(infrared,rectified_infrared,mappings.infrared);
      cv::remap(grayscale,rectified_grayscale,mappings.grayscale);
    }
    // Remap the ROIs into rectified image and extract overlapping patches in
      both the images corresponding to the new ROIs.
    new_ROI = remapRect(ROI,mappings.infrared);
    infrared_image_patch = rectified_infrared(new_ROI);
    grayscale_image_patch = rectified_grayscale(new_ROI);
  }
}
```

Listing 1.1: Pseudo source code for Image fusion

3 Experimental Results

Inspired by real-world search and rescue initiatives, we designed a representative field trial utilising our full aerial deployment and victim detection pipeline. The experiment involves the placement of multiple "victims" throughout a field over which our rapidly deployable, hand launched, fixed-wing test platform, equipped with multi-camera carrying sensorpod, must survey in a timely manner and record GPS coordinates of victims seen during the flight. In the following sections, we present the setup, enabling tools, and results.

3.1 Platform Description

All the experiments are performed on a light-weight low-altitude hand-launchable fixed-wing UAV, Techpod, with an integrated forward-facing sensor pod, see Fig. 6. Techpod weighs 2.65 kg, has a wingspan of 2.6 m, and flies at a nominal airspeed of 14 m s^{-1}. The UAV also consists of various telemetry, inertial modules and employs an open-source Pixhawk PX4 Autopilot [16] system. The sensorpod is equipped with rigidly connected infrared (FLIR Tau 2) and grayscale (Aptina MT9V034) cameras with oblique field-of-view, IMU (Analaog Devices ADIS16448), Skybotix VISensor [17] and a computer with an Intel Atom processor.

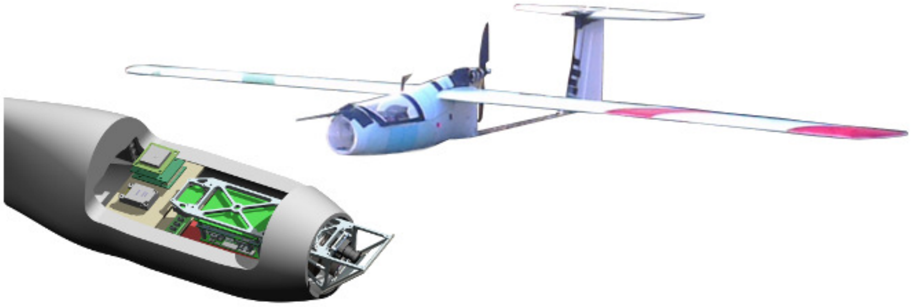


Fig. 6: Techpod UAV with integrated sensing pod.

3.2 Full coverage aerial inspection

To ensure victims are seen by the cameras' fields of view at some point during the flight, we employ a fast inspection planning algorithm which guarantees full coverage of a mesh-model world (obtained from GIS data) to ensure all ground is scanned for victims. The algorithm, details in [18], employs an iterative viewpoint re-sampling technique, using randomized sampling, to provide viewpoint configurations that maintain full coverage while simultaneously searching for the best route that visits them all, using the Lin-Kernighan heuristic [19]. Nonholonomic vehicle constraints and obstacle avoidance are further considered in the path search using a boundary value solver in combination with the Rapid Random Tree (RRT*) motion planner [20]. The path generated (see Fig. 7(a)) for this experiment ensured coverage of 52,800 sq. meters with a flight time of just under two minutes.

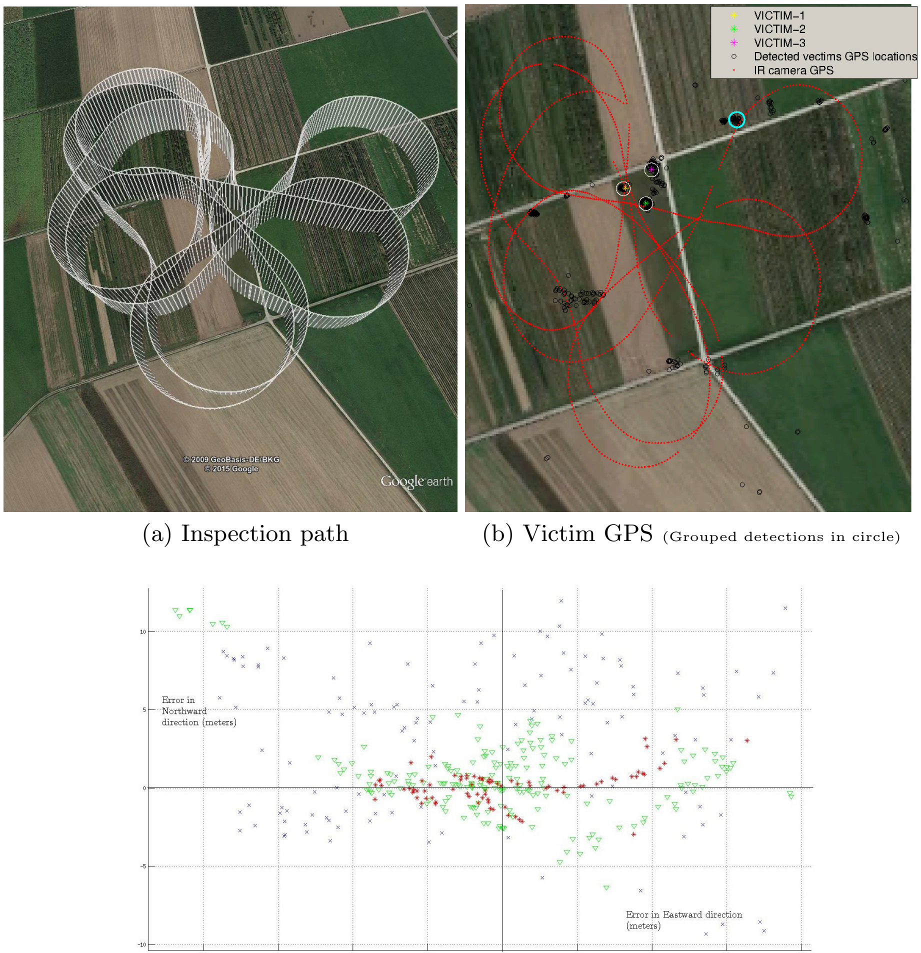
3.3 Estimating victim GPS

For estimating the GPS positions of detections, we assume the ground as a plane. Using camera intrinsics and detection's pixel location, we find the directional vector in which the object might be lying. The point where this vector hits the ground plane can be found using GPS, IMU and camera-IMU extrinsics data which finally gives victim's GPS location.

Fig. 7(b) shows the detected victim GPS positions, actual groundtruth victim location and the path followed by the UAV. The rest of the detections include unregistered humans in the explored area and some false positives. By enforcing a criterion - similar to [4] - that requires a particular GPS position to be detected consistently within the duration of the time that particular spot is observed, most of the false positives are eliminated and the set of true detections are grouped together and registered as victim locations. False positives are further eliminated by boosting the confidence of detections that are closer to the GPS location of the point where principal axis of camera hits the ground plane. The registered victims are marked in circles. As can be seen in the figure, we have all three successful detections and another detection whose status is not known. Fig. 7(c) is a plot of GPS position estimate errors for all the three registered victims. The GPS estimates are converted to Universal Transverse Mercator (UTM) coordinates using World Geodetic System (WGS84) standard [21] to represent the error in meters. The errors obtained are a cumulative of errors in groundtruth GPS measuring device, calibration imperfections, GPS to/from UTM conversions, images' resolution, planarity assumptions, and sensor imperfections (GPS, IMU), making it quite uncertain how much of the error can be attributed to the imperfections in our algorithm. Nevertheless, the errors we obtained more than suffice to serve our goal.

4 Conclusions and Future Work

At present, we have many parameters that need to be set manually. In the future, we plan to have some adaptive estimation of these values. Computing HOG descriptors for each foreground patch is quite time consuming as well. Instead, using a weak classifier with high recall could reduce computation times, as our pipeline has the ability to deal with classifier false positives. We also plan to track the detections in UTM coordinates to have consistent detections of moving targets. Use of geographic elevation data would also be a way to remove



(a) Inspection path (b) Victim GPS (Grouped detections in circle)

(c) Victim UTM position errors

Fig. 7: Full coverage field test detection results.

the planar ground assumption currently implemented and improve detection location estimations. In all, we have shown, as a proof of concept, that human victim detection from upwards of 50-60 meters via a fast moving fixed-wing aircraft is possible. Further, we have demonstrated the algorithm's real-time capacity for victim detection and localisation in a real-world search scenario.

Acknowledgements

This work was supported by the European Commission projects ICARUS (#285417) and SHERPA (#600958) under the 7th Framework Programme.

References

1. Wang, W., Zhang, J., Shen, C.: Improved human detection and classification in thermal images. In: Image Processing (ICIP), 2010 17th IEEE International Conference on, IEEE (2010) 2313–2316
2. Davis, J.W., Keck, M.A.: A two-stage template approach to person detection in thermal imagery. In: null, IEEE (2005) 364–369
3. Treptow, A., Cielniak, G., Duckett, T.: Active people recognition using thermal and grey images on a mobile security robot. In: Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on. (2005) 2103–2108
4. Rudol, P., Doherty, P.: Human body detection and geolocalization for uav search and rescue missions using color and thermal imagery. In: Aerospace Conference, 2008 IEEE, IEEE (2008) 1–8
5. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on. Volume 1., IEEE (2001) I–511
6. Portmann, J., Lynen, S., Chli, M., Siegwart, R.: People detection and tracking from aerial thermal views. In: Robotics and Automation (ICRA), 2014 IEEE International Conference on, IEEE (2014) 1794–1800
7. Toet, A., van Ruyven, L.J., Valeton, J.M.: Merging thermal and visual images by a contrast pyramid. Optical Engineering **28** (1989) 287789–287789–
8. Heo, J., Kong, S., Abidi, B., Abidi, M.: Fusion of visual and thermal signatures with eyeglass removal for robust face recognition. In: Computer Vision and Pattern Recognition Workshop, 2004. CVPRW '04. Conference on. (2004) 122–122
9. Istenic, R., Heric, D., Ribaric, S., Zazula, D.: Thermal and visual image registration in hough parameter space. In: Systems, Signals and Image Processing, 2007 and 6th EURASIP Conference focused on Speech and Image Processing, Multimedia Communications and Services. 14th International Workshop on. (2007) 106–109
10. Nandhakumar, N., Aggarwal, J.: Integrated analysis of thermal and visual images for scene interpretation. Pattern Analysis and Machine Intelligence, IEEE Transactions on **10** (1988) 469–481
11. nán, C.C.L.: cvBlob. (http://cvblob.googlecode.com)
12. Dollár, P.: Piotr's Computer Vision Matlab Toolbox (PMT). (http://vision.ucsd.edu/ pdollar/toolbox/doc/index.html)
13. Cortes, C., Vapnik, V.: Support-vector networks. Machine Learning **20** (1995) 273–297
14. Furgale, P., Rehder, J., Siegwart, R.: Unified temporal and spatial calibration for multi-sensor systems. In: Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on, IEEE (2013) 1280–1286
15. Fusiello, A., Trucco, E., Verri, A.: A compact algorithm for rectification of stereo pairs. Machine Vision and Applications **12** (2000) 16–22
16. Meier, L., Honegger, D., Pollefeys, M.: PX4: A node-based multithreaded open source robotics framework for deeply embedded platforms. In: Robotics and Automation (ICRA), 2015 IEEE International Conference on. (2015)
17. Skybotix AG: (2015) http://www.skybotix.com/.
18. Bircher, A., Alexis, K., Burri, M., Oettershagen, P., Omari, S., Mantel, T., Siegwart, R.: Structural inspection path planning via iterative viewpoint resampling with application to aerial robotics. In: Robotics and Automation (ICRA), 2015 IEEE International Conference on. (2015) 6423–6430

19. Lin, S., Kernighan, B.W.: An effective heuristic algorithm for the traveling-salesman problem. Operations research **21** (1973) 498–516
20. Karaman, S., Frazzoli, E.: Incremental sampling-based algorithms for optimal motion planning. CoRR **abs/1005.0416** (2010)
21. Agency, N.G.I.: World geodetic system 1984. (http://web.archive.org/web/20120401083859/http://earth-info.nga.mil/GandG/wgs84/index.html)