Pattern Recognition and Structure From Motion for a UAV

Keegan R. Kinkade
Graduate Student
University of Michigan
Robotics and Autonomous Vehicles Program
Ann Arbor, MI USA
kinkadek@umich.edu

Pratik Agarwal
Graduate Student
University of Michigan
Computer Science Engineering
Ann Arbor, MI USA
pratikag@umich.edu

***Abstract*—The following paper presents computer vision methods designed to aid an autonomous quadrotor at the International Aerial Robotics Competition. Pattern recognition methods are used in order to allow the quadrotor to recognize security compound signs inside the competition environment. Once detected and recognized, these security compound signs are then used as landmarks within a structure from motion method intended to aide a simultaneous localization and mapping algorithm. The computer vision methods described are simple and time efficient, meant to be processed on a ground station unit. Ultimately, this paper demonstrates a novel pattern recognition method combining previous recognition and vision techniques which provides the necessary data to perform structure from motion.**

I. INTRODUCTION

The Michigan Autonomous Aerial Vehicles (MAAV) program is a research group focused around an autonomous quadrotor for competition in the International Aerial Robotics Competition. The goal of the research competition is to navigate autonomously through an environment constructed of passageways and rooms, locate and retrieve a USB key, and navigate back out of the environment. Each room within the environment is labeled with one of the three Arabic security compound signs as seen in figure 1. The location of the USB key is provided prior to competition. It's location is given as residing in one of the three rooms adorned with the security compound.

Security Compound Signs

Security Compound مجمع امني

Ministry of Torture وزارة التعذيب

Chief of Security رئيس الألمن

Figure 1: Arabic labels, alongside their English equivalents, which are posted throughout the competition environment.

One of MAAV's goals is to be able to use computer vision techniques to locate the security compound signs within the environment and accurately identify which room the located sign corresponds to. The ability to locate these signs is pivotal to MAAV's success, as it would reduce the need to investigate every room within the environment in search of the USB key, thus reducing the amount of time the quadrotor operates within the environment. Less time spent inside the environment corresponds to less errors within the simultaneous localization and mapping algorithm. Time in the environment is also a metric used in the grading scheme for the competition.

Figure 2: Image of MAAV's quadrotor design. From [1].

After locating a security compound sign and correctly identifying its corresponding room, computer vision methods can then aide in the simultaneous localization and mapping (SLAM) algorithm which is employed from the onset of entering the building. The SLAM algorithm employed by MAAV uses laser range finders, inertial measurement units, and sonar sensors to accurately map the environment while predicting the quadrotor's location and orientation with respect to the map. The SLAM position and orientation predictions will be updated by comparing multiple images taken of the located security sign while the sign is within the perceptual view of the quadrotor. Structure from motion with respect to these security signs will allow for a more accurate pose prediction to be processed by the SLAM algorithm.

The remainder of this paper will present the previous methods in computer vision utilized to accomplish the desired recognition and structure from motion for the quadrotor, the technical aspects of our methods in detail, and lastly, experiments using our methods along with the overall results of our research. Figure 3 displays the overall concept figure of our desired vision techniques to aid MAAV in the International Aerial Robotics Competition.

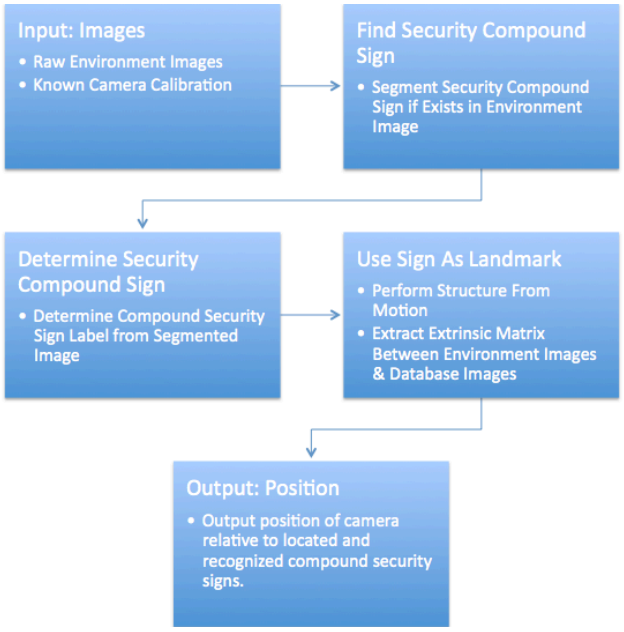

Figure 3: Concept Figure showing desired vision techniques to aid the quadrotor.

II. Previous Methods/Key Contributions

A. *Review of Previous Work Researched*

During the development process, we looked at many papers dealing with recognition methods. Our goal was to take an environment image and determine if there is a security sign within the image, and if so, which security sign is it. One of the most popular vision techniques to extract features and their corresponding descriptors from an image is David Lowe's Scale Invariant Feature Transform (SIFT). In his paper, "Distinctive Image Features from Scale Invariant Keypoints", Lowe presents a method for extracting distinctive features from images which are invariant to scale and rotation [2]. SIFT builds upon Lindeberg's scale spaced image defined as

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x,y),$$

where $\mathbf{L(x, y, \sigma)}$ is the image's scale space, $\mathbf{I(x,y)}$ is the original input image, and $\mathbf{G(x, y, \sigma)}$ is the gaussian kernal which is convolved with the input image [3]. This scale space is then used to create a difference-of-Gaussian function, defined as

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x,y)$$

$$= L(x, y, k\sigma) - L(x, y, \sigma),$$

where **k** is a constant multiplicative factor creating the separation in scaling [2]. Figure 4 shows the process of convolving an image with different scales of Gaussian kernals on different scales of the image and how these scale spaces are used to create their corresponding difference-in-Gaussian values for the differing image scales.

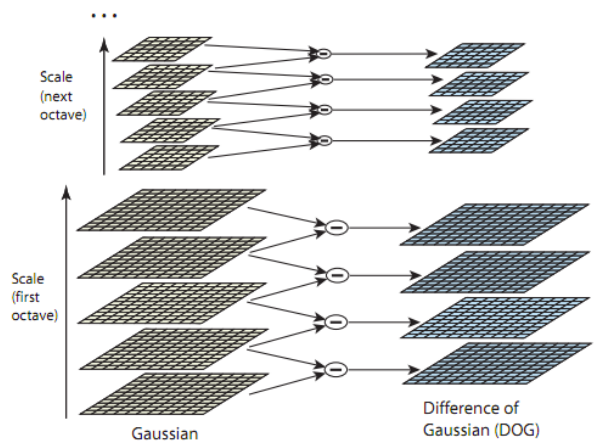

Figure 4: Difference of Gaussian Function. From [2].

The difference-of-Gaussians function is then used to compare each pixel to its neighbors in order to create thresholding with which to choose keypoints. Such thresholding includes using the minima and maxima of the difference-of-Gaussian function, applying a threshold on the minimum contrast, and adding an additional threshold based upon ratio's of principal curvatures. These keypoints are then assigned descriptors. Descriptors are created by computing the gradient magnitude and orientation of each sample point in a region around the keypoint. A Gaussian window is then used to weight all the gradients which are then assigned to orientation histograms with each orientation section representing the total of the magnitudes for that orientation. Figure 5 shows an example of a keypoint's descriptor, created from an 8x8 set of sample gradients. Note that Lowe's SIFT implementation uses a 4x4 set of descriptors computed from a 16x16 set of samples.

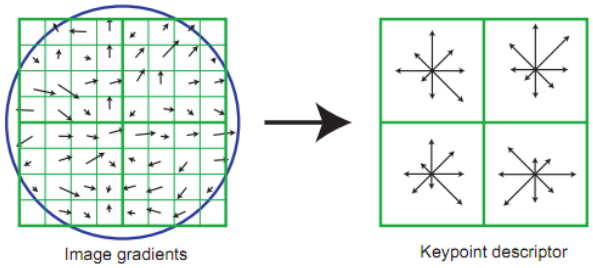

Figure 5: Example of a keypoint's descriptor. Sample gradients are shown on the left, with the full descriptor shown on the right. From [2].

Another paper which aided in our work was from the winning team of the 2009 Semantic Robot Vision Challenge, a robotic competition designed to push the state of the art in object recognition. This team from the United States Naval Academy used SIFT to compare images of objects downloaded from the internet to images taken within an environment containing the same objects [4]. In order to determine if the robot had found the desired object within the environment, they used Lowe's

matching scheme, which matches each keypoint descriptor from one image to a descriptor in another image by minimizing their Euclidean distance and satisfying a given threshold. This creates corresponding points between the downloaded images of the objects and the environment images. Figure 6 shows this matching scheme.

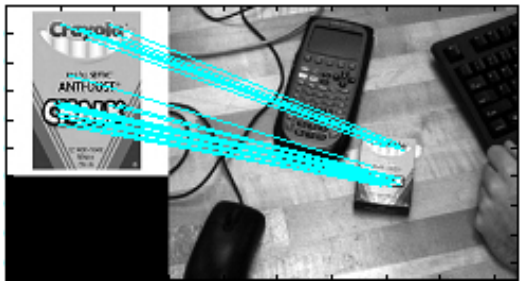

Figure 6: Initial matches between object image on left and environment image on right. From [4].

In an effort to remove outlier matches, initial matches were used to scale and rotate the downloaded object image to mimic that of the object within the environment image. Next, the sift keypoint descriptors were again matched between the two images. Correct matches between the two images form lines which have extremely similar slopes. Those matches forming lines with differing slopes from the others are classified as being outlier matches and are removed. This allows for more accurate matches to be considered when determining whether an object has been discovered. Figure 7 shows this process of removing outliers.

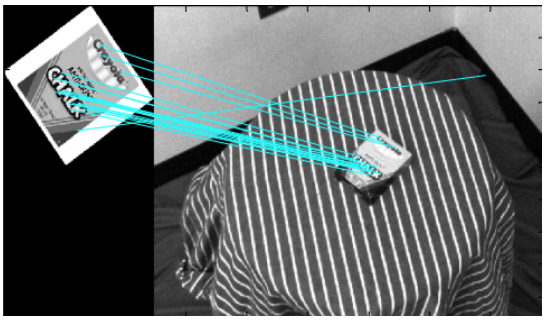

Figure 7: Example of inlier and outlier matches. Inlier matches create lines with similar slopes. From [5].

The last previous work which aided in our research and our structure from motion algorithm was that of the AprilTag by Edwin Olson [6]. The AprilTag is a fiducial system which uses 2D bar codes, as seen in figure 8, to localize a system. As will be shown in Section III, these 2D bar codes placed in a single known plane and scaled to a known size allow a camera to recreate an extrinsic matrix. This extrinsic matrix is computed from the intrinsic parameters of the camera system as well as the homography between a known image of the tag and the environment image of the tag. The extrinsic matrix ultimately allows for a 6DOF localization to be extracted between the location of the camera and the location of the tag.

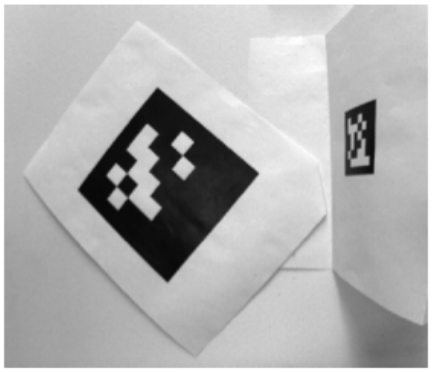

Figure 8: Image of an AprilTag. From [6].

B. *Contributions of Work*

Our work brings together elements from the previous described research in order to build robust vision methods to aid the quadrotor. Using SIFT descriptors we are able to create an object recognition method to detect compound security signs. Furthermore, we implemented additional vision techniques to improve upon SIFT matching schemes, allowing us to decipher outlier matches similar to the recognition scheme discussed from the Semantic Robot Vision Competition. Lastly, using the same techniques behind the AprilTag, we are able to use the compound security signs as a landmark from which to compose structure from motion.

III. TECHNICAL DETAILS

The following section discusses in detail the vision techniques we applied to aid in the quadrotor's mission. We begin by examining each image in order to segment out a compound security sign using thresholding techniques. Next, applying SIFT descriptors with a random sample consensus algorithm allows us to determine a more accurate amount of SIFT matches between the segmented image of the compound security sign and those Arabic words shown in figure 1. Lastly, using the homography matrix from the random sample consensus algorithm used to determine outlier matches, we are able to create an extrinsic matrix from the environment image. This matrix allows us to compute the 6DOF localization of the camera with respect to the compound security sign.

A. *Segmentation*

In order to segment out a compound security sign from an environment image, we operate under the assumption that we have some type of access to the environment prior to sending in the quadrotor. Access to the environment is provided by the competition before operation, and thus makes segmentation using thresholding extremely simple and efficient to implement. Figure 9 is an environment image from the 2010 International Aerial Robotics Competition.

Capturing an image such as this prior to the competition allows for a grayscale threshold to be determined.

Figure 9: Image from the 2010 International Aerial Robotics Competition of the environment.

Using the image in figure 9, a value of 0.8 was obtained from the white section of the sign after the image was remapped into a normalized grayscale image. Using a slightly lower threshold value, we create a binary image by setting all pixel values to binary '1' whose values are above the threshold, and all pixel values to binary '0' whose values are below the threshold. This creates a binary image comprised of true values being white, and false values being black. Figure 10 shows the original environment image side by side with the binary thresholded image. While in our implementation we use a single threshold, it is also possible to set low and high thresholds, creating a more robust threshold gate.

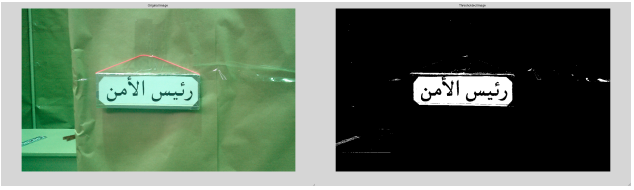

Figure 10: Segmentation of compound security sign using thresholding.

As figure 10 shows, there is a lot of noise using the thresholding method of segmentation. Thus, in order to clean up the noise and only return the security sign, our algorithm returns the largest white blob in the image and reduces the rest of the pixels carrying a white value but not associated with the largest blob to black:

Figure 11: Segmentation of compound security sign post noise reduction by returning maximum white blob.

Lastly, our segmentation algorithm finds the dimensions of the segmented security sign, and crops the original environment image according to this dimension. The result of this cropping is the final segmentation of the compound security sign, and is then used for comparison with the database of possible security signs. Figure 12 shows the final segmented image created from the original environment image shown in figure 9.

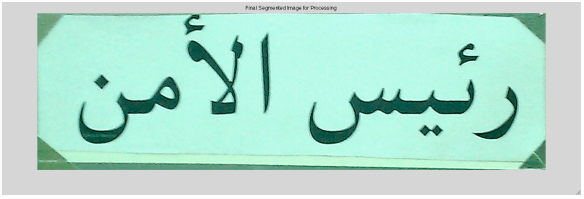

Figure 11: Final segmented image of the compound security sign from the environment image in figure 9.

B. Security Sign Recognition

For image processing, we use the VLFeat MATLAB toolbox from [7]. This toolbox includes SIFT functions modeled after David Lowe's implementation from [2]. The toolbox also includes a matching function which allows two sets of SIFT descriptors to be compared and matched. Matching is done by comparing the Euclidean distance between descriptors as explained in Section II and seen in [2]. The default Euclidean distance threshold for this matching of descriptors is set to 1.5 within the VLFeat toolbox and was not changed during our use.

This toolbox allowed us to build basic functions to determine the SIFT features and descriptors for two images, as well as match the two sets of features to one another based on their descriptors. For example, consider the segmented image from figure 11. If we compare this image to the security compound signs in our database of signs from figure 1, we can determine that this security sign represents the sign corresponding to the room of the Chief of Security. Figure 12 shows the matching SIFT features for the segmented environment image compared with each of the database images:

Figure 12: Matching SIFT descriptors of the segmented environment image compared with all other database images.

The correct pairing, the top image in figure 12, has a total of 51 matches. The other two incorrect comparison have 28 and 18 matches respectively. Although the correct database image returns the most matches of all the database images, outlier matches still occur between itself and the environment image. In order to make this process more robust, we implemented a random sample consensus (RANSAC) algorithm to clean up the outlier matches.

A homography is a projective transformation which transforms points $\mathbf{x_i}$ from on image to their corresponding points $\mathbf{x'_i}$ from a second image. To solve for the homography matrix **H** between a database image and a segmented security compound sign we use four matching features between the two images calculated from SIFT. These four matching features are applied to a Direct Linear Transformation (DLT) algorithm described in [8] having the following notation:

$$\begin{bmatrix} \mathbf{0}^\mathsf{T} & -w'_i\mathbf{x}_i^\mathsf{T} & y'_i\mathbf{x}_i^\mathsf{T} \\ w'_i\mathbf{x}_i^\mathsf{T} & \mathbf{0}^\mathsf{T} & -x'_i\mathbf{x}_i^\mathsf{T} \end{bmatrix} \begin{pmatrix} \mathbf{h}^1 \\ \mathbf{h}^2 \\ \mathbf{h}^3 \end{pmatrix} = \mathbf{0}.$$

where prime elements correspond to the matching feature positions in the second image, non-prime elements corresponds to the matching features positions in the first image, and **w** represents the homogenous scaling factor which is 1 for our purposes. The above equation as the form $\mathbf{A_i h} = \mathbf{0}$, where **h** is a 9x1 vector representing the homography matrix as seen in [8]:

$$\mathrm{H} = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix}$$

The homography H can then be determined from four point correspondences using the Singular Value Decomposition, described in [8], to produce $\mathbf{A} = \mathbf{UDV}^\mathbf{T}$. So long as the diagonal matrix **D** has positive diagonal entries arrange in descending order down its diagonal, then **h** is found to be the last column of **V**.

Using both the SIFT matches and the above DLT formulation for calculating homography transformations, we apply a RANSAC algorithm. The RANSAC algorithm, as shown in figure 13, provides not only a strong homography transformation matrix between the database images and the segmented compound security sign, but also the sets of inlier matches and outlier matches.

Objective

Robust fit of a model to a data set S which contains outliers.

Algorithm

(i) Randomly select a sample of s data points from S and instantiate the model from this subset.
(ii) Determine the set of data points S_i which are within a distance threshold t of the model. The set S_i is the consensus set of the sample and defines the inliers of S.
(iii) If the size of S_i (the number of inliers) is greater than some threshold T, re-estimate the model using all the points in S_i and terminate.
(iv) If the size of S_i is less than T, select a new subset and repeat the above.
(v) After N trials the largest consensus set S_i is selected, and the model is re-estimated using all the points in the subset S_i.

Figure 13: RANSAC Algorithm. From [8].

The RANSAC algorithm used in our algorithm follows that shown in figure 13 except we allow 50 iterations of homography computations to occur. After 50 homographies have been computed from random point correspondences from SIFT, our algorithm returns the best homography along with the inlier matches corresponding to the best homography. This is done for computational speed, and it will be shown in Section IV to be extremely effective. Figure 14 demonstrates the before and after effects of using RANSAC to remove outlier SIFT matches.

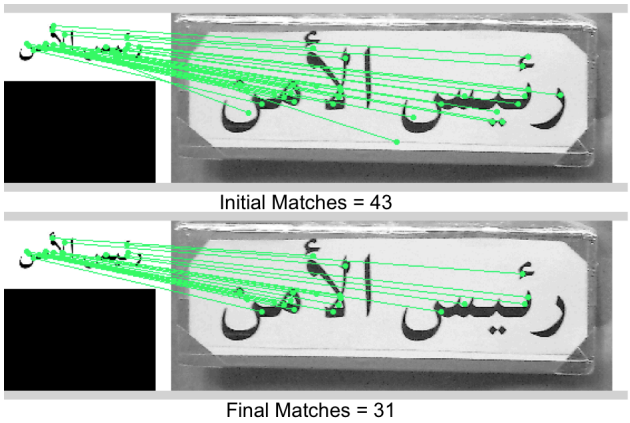

Figure 14: SIFT Matches between correct database image and segmented compound security sign before and after outliers are removed using RANSAC.

C. Localization using Structure from Motion

The homography created from our RANSAC algorithm allows for the extrinsic matrix to be calculated . From this extrinsic matrix, we can perform the 6DOF localization of the camera with respect to the compound security sign. Before processing, we normalize the images in our library database such that $[0\ 0\ 1]^\mathrm{T}$ is at the center of the sample image and ensure that the image extends one unit in length (a factor which depends upon the measuring scheme used). Next, the homography matrix which projects the homogenous points from the sample image to the image coordinate system is computed as previously described.

Because the homography matrix is defined only up to scale, computation of the Arabic Sign's position and orientation requires additional information. Prior to localization, the camera must be calibrated in order to

determine the intrinsic parameters. Because we know the compound security sign is going to be confined in a single plane, every position on the compound security sign can be defined as having z = 0 with respect to it's own coordinate system. This allows us to truncate the extrinsic matrix by removing the third column. Thus, we can write the already determine homography matrix as shown in [6] as being:

$$\begin{bmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{bmatrix} = sPE$$

$$= s \begin{bmatrix} 1/f_x & 0 & 0 & 0 \\ 0 & 1/f_y & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R_{00} & R_{01} & T_x \\ R_{10} & R_{11} & T_y \\ R_{20} & R_{21} & T_z \\ 0 & 0 & 1 \end{bmatrix}$$

where **P** is our projection matrix, **E** is the extrinsic matrix, and **s** is an unknown scale factor. Due to the projection matrix being rank deficient, we are unable to directly solve for **E**. However, we can expand the above to create a system of equations for the individual homography matrix as also shown:

$$\begin{aligned} h_{00} &= sR_{00}/f_x \\ h_{01} &= sR_{01}/f_x \\ h_{02} &= sT_x/f_x \\ &\dots \end{aligned}$$

These equations can be rewritten in terms of their **R** or **T** values. In order to solve for the scaling factor s, we constrain it's magnitude to being the geometric average of the rotation magnitudes as the columns of the rotation matrix must all be of unit magnitude. The sign of s is determined from requiring that $\mathbf{T_z}$ is less than zero due to the fact that the image must be in front of the camera. Although truncated, the third column of the rotation matrix used to determine the magnitude of s is calculated from the cross product of the first two columns of the rotation matrix. This satisfies the condition that the columns of the rotation matrix are orthonormal to each other.

Before computing s as described above, we ensure that the rotation matrix is strictly orthonormal by minimizing the Frobenius matrix norm of the error via the polar decomposition of **R**. For the polar decomposition, we pull out the 3x3 rotation matrix and find its singular value decomposition, producing $\mathbf{R} = \mathbf{UDV^T}$. We then multiple **U** by $\mathbf{V^T}$ to recover the normalized orthonormal rotation matrix, which is further decomposed to recover the Euler angles with respect to our camera axis.

IV. EXPIRIMENTS

In order to test our methods, we created three different data sets of compound security images using a digital camera, calibrated via MATLAB's camera calibration software. This allowed us to retrieve a project matrix with zero skew, making our calculations for the 6DOF localization simpler and aligning with the method described in Section III. Each of the three data sets represent a different set of images corresponding to a specific compound security sign. Figure 15 shows all of the images within database 1.

Figure 15: Database test set images. This dataset represents one of the three types of compound security signs.

Each image within all three testing datasets were tested via the image processing and recognition method described in Section III. Table 1 shows the results and overall accuracy of these tests:

Data Set ID	Number of Images	Number of Images Correctly Identified	Average Elapsed Time (s)
1	10	10	0.5126066
2	11	11	0.8205818
3	5	4	0.4676871

Table 1: Results of running all testing images in the three datasets through our image recognition method.

As table 1 shows, our method takes approximately half a second to process each environment image. While this is not fast enough in order to process realtime video feed, it would be more than efficient if the quadrotor only took images when it believed it was near a door of a room. This is a reasonable desire as doorways can easily be distinguished using the laser range finder on the

quadrotor. Furthermore, the timing associated with table 1 includes the determination of the extrinsic matrix using an additional RANSAC method for normalizing the known image of the compound security sign. These computation can be removed to increase the speed of the algorithm if localization using the compound security sign is not desired. Lastly, our method could be further sped up by reading in the SIFT features and descriptors of the library signs prior to taking images of the environment. While these are simple implementations which would improve the overall speed of our image recognition system, we chose not to implement them. We believe that the current setup processes images quick enough so long as the quadrotor only takes images when sensing it is near doorways.

Figure 16 shows one of the database images after segmentation being compared with the library images. From this figure, we can determine that the RANSAC method not only removed outliers, but creates a large gap between the number of matches for the correct library image and the number of matches for the other images.

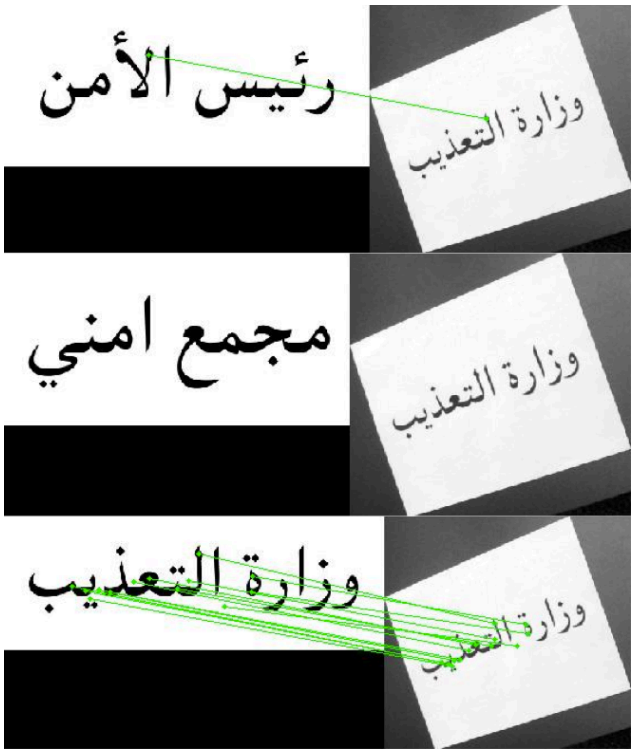

Figure 16: Segmented compound security sign compared with library images using object recognition method discussed in Section III.

Note that in figure 16 the correct library image (the bottom comparison) had 18 matches after running the RANSAC algorithm on the two images. The other two images had 1 and 0 matches after comparison. This figure, as well as the data provided in table 1, prove that this method is extremely robust. However, in our testing, there was one poor data association within dataset 3. This mistaken data association is shown in figure 17.

Figure 17: Matches from the only incorrect recognition of all images tested.

As we can see from figure 17, the library image which had the most matches was the incorrect library image. However, the maximum matches for this environment image was 1. Therefore, one measure we can take to ensure this doesn't happen is set a minimum threshold value of matches required for a recognition to occur. For the environment image in figure 17, the actual compound security sign was so small that when it was compared with the true library sign, the RANSAC method was unable to come up with a good homography capturing any of the initial matches.

We also tested each database of test images with our localization method. Figure 18 displays the 3D points and orientation association with the camera poses from figure 15.

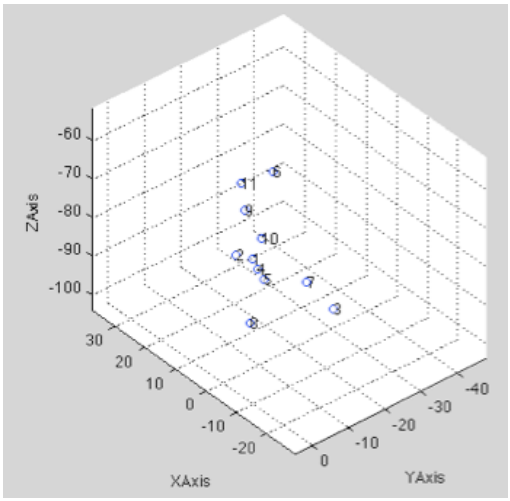

Figure 18: 6DOF localization poses from images in figure 15.

Up to this point, we have not been able to test the accuracy of this 6DOF method with the SLAM algorithm of the quadrotor as the SLAM algorithm is currently being implemented. However, from our experiments, we are able to determine that the overall accuracy of this method depends on the accuracy within the homography matrices created for each image. Furthermore, all of our test datasets returned expected 6DOF poses from the environment images. Until the system is able to be incorporated with the SLAM method, it is uncertain as to how much influence it will be able to have on the operation of the quadrotor within the environment.

V. Conclusion

The Michigan Autonomous Aerial Vehicle program requires an autonomous quadrotor UAV designed to fly in an office environment and equipped with cameras in order to recognize signs on the wall and aide in Simultaneous Localization and Mapping algorithms. This report has presented a solution in which pattern recognition and structure from motion aid the UAV.

A SIFT based pattern matching technique enhanced by a RANSAC algorithm method provided a robust implementation for object recognition. Furthermore, the complete 6DOF position of the UAV was recovered using a single calibrated camera. We have tested our code on real images, and produced expected results which show that these methods are capable of providing the desired computer vision techniques needed by MAAV. The next step with this project is to begin to run the system in an online fashion and incorporate the vision measurements into the SLAM solution for motion planning of the quadrotor. Some measures that will be taken include porting the implementation to a compiled coding language such as Java. This is an ongoing project which will continue to be implemented for the next International Aerial Robotics Competition.

Personally, we have both gained invaluable experience in utilizing current state of the art vision methods as well as the ability to apply these methods to the field of mobile robotics. These tools will aid in our future education and work within the robotics society, providing methods with which we can employ on robotic platforms.

References

[1] Ellis, D. et al., “Quadrotor Unmanned Aerial Vehicle for the International Aerial Robotics Competition,” unpublished. *Accessed from: http://iarc.angel-strike.com/2010SymposiumPapers/University_of_Michigan_2010.pdf*

[2] David G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision, 60, 2 (2004), pp. 91-110.*

[3] Tony Lindeberg, “Feature Detection with Automatic Scale Selection,” *International Journal of Computer Vision, 30, 2 (1998), pp. 77-116.*

[4] Felps, C.M.; Fick, M.H.; Kinkade, K.R.; Searock, J.; Piepmeier, J.A.; “Integration of Semantic Vision Techniques for an Autonomous Robot Platform,” *System Theory (SSST), 2010 42nd Southeastern Symposium on, 7-9 March 2010, pp. 243 – 247.*

[5] Felps, C.M.; Fick, M.H.; Kinkade, K.R.; Searock, J.; Piepmeier, J.A.; *U.S. Naval Academy’s Semantic Vision Workshop Presentation. Accessed From http://www.semantic-robot-vision-challenge.org/presentations/2009SRVC-USNA.pdf*

[6] Edwin Olson, “AprilTag: A robust and flexible multi-purpose fiducial system” *Accessed from: http://april.eecs.umich.edu/papers/details.php?name=olson2010tags*

[7] Fulkerson, B., and A. Vedaldi. *An Open and Protable Library of Computer VIsion Algorithms. 2008. Accessed from: http://www.vlfeat.org/.*

[8] Hartley, Richard, and Andrew Zisserman. *Multiple View Geometry in Computer Vision. Cambridge University Press, 2004.*

[9] Forsythe, David, and Jean Ponce. *Computer Vision: A Modern Approach. Prentice Hall, 2002.*

[10] Tomasi, C; Kanade, T; “Shape and Motion from Image Stream Under Orthography: A Factorization Approach,” *International Journal of Computer Vision, 9(2): 137 - 154, November 1992.*