Multi-Resolution Unmanned Aerial Vehicle Video Stabilization

Steven Hong
Dept. of Electrical Engineering
Stanford University, Stanford, CA
Email: hsiying@stanford.edu

Tracey Hong
Dept. of Biomedical Engineering
Vanderbilt University, Nashville, TN
Email: tracey.s.hong@vanderbilt.edu

Wu Yang
Dept. of Electrical Engineering
Wright State University, Dayton, OH
Email: wuyang1977@gmail.com

***Abstract*—This paper presents an efficient multiresolution video stabilization algorithm based on the Scale Invariant Feature Transform (SIFT) algorithm. The algorithm utilizes the Haar wavelet transformation of registered images, reducing the memory consumption while maintaining accuracy of the feature points, which we will quantitatively contrast in the paper.**

I. Introduction

Video stabilization techniques are becoming an integral component of target tracking as increased applications in the medical and military sectors utilize moving sensors, requiring advanced real time video stabilization in order to process the data. Target tracking when the camera is stationary is a relatively mature field but the moving sensor poses uniquely challenging problems because relative to the camera, everything in the scene appears to be moving. The moving sensor paradigm has proven to be uniquely different as compared to the stationary sensor because in addition to not knowing what the coordinates, rotation, and translation of the camera are with respect to the target, the motion of the targets must be distinguished from the global motion of the scene. This motion could result from uneven terrain, high frequency vibrations from the engine, uneven air currents, or instability of the pan-tilt camera system and presents the most challenging aspect of analyzing data from moving sensors. The movement of the camera makes the stationary background appear to move and thus even if all of the objects are stationary, the external movement makes the objects appear in different locations with different camera coordinates [1][2].

In order to distinguish the movement of the camera from the movement of actual targets in the background, it is easiest to think of ego-motion as an unintended motion transformation (the platform movement causes shifting, rotation, and scaling of the target frame). By establishing a metric from which a frame to frame parametric model is fitted, the algorithm is able to transform different sets of data, obtained from sampling the same scene at different times, into one coordinate system. This enables the comparison or integration of data obtained from multiple time periods. The metric used in the presented algorithm are image features detected with SIFT (Scale Invariant Feature Tracker), which are used as control points from which subsequent frames are transformed with respect to. This homographic transformation undoes the sensor platform motion and thus allows background modeling to be performed.

In order to identify features, several groups have used the Kanade-Lucas-Tomasi (KLT) algorithm to identify control points for global registration, which is not very robust [3][4]. For instance, problems arise when objects that are fixed but elevated from the ground are difficult to register and indistinguishable from moving targets. 3D structures present serious problems due to the fact that they generally display the type of features that are integral to the registration fit (lines and corners). Because 3D structures are ”high in strong corners”, buildings may cause alignment to roofs rather than the ground, especially if the relative displacement between corresponding corners is uniform, i.e. the roof is flat [5]. This ruins the goal of registering corresponding ground features from frame to frame [6]. Additionally, because KLT is a sparse feature–based tracker, in relatively featureless areas like flat grassy plains, the tracker breaks down much more rapidly than a SIFT based registration algorithm would.

Unfortunately, a stabilization system must run at a near real-time rate to be of any operational value in the field. In our group’s previous work [1], the registration algorithm was run on a 2.4 GHz dual core desktop and only operated at a speed of approximately $\leq$ 1 frames/second. Running this program, even when using suitable parallel specialized hardware, would not be feasible in real time. In order to solve this dilemma, a multiresolution video decomposition scheme is utilized. Using a Haar wavelet based decomposition scheme, the target frame is broken down into a wavelet hierarchy representation. Processing only the low-low-pass, the SIFT registration algorithm runs at a greatly accelerated pace.

This paper presents a multiresolution SIFT (Scale Invariant Feature Transform) based registration algorithm that undoes the motion transformation by translation and rotation of the frame based on feature points. The feasibility of this approach is demonstrated with an offline MATLAB implementation of a video sequence obtained from a UAV (Unmanned Aerial Vehicle). We will first introduce the Scale Invariant Feature Transformation Algorithm, followed by its implementation in a homographic transformation registration scheme. Next, a Haar wavelet transform is applied in order to reduce memory consumption while maintaining accuracy to the greatest extent possible. An analysis of statistically evaluated comparisons

978-1-4244-6578-1/10/$26.00 ©2010 IEEE

on the original video stabilization methods and the multiresolution approach will be performed. The results of the presented algorithm will then be compared to another well known stabilization algorithm based on KLT. Conclusions will then be made about the feasibility of the approach and the direction of future work.

II. Control Point Detection

The first step in our video stabilization is to identify control points from which subsequent frames are transformed with respect to. The Scale Invariant Feature Transform (SIFT) Algorithm is one of several computer vision techniques for extracting distinct features from images. Specifically, the features obtained by SIFT are invariant to image scale, rotation, and robust to changing viewpoints and illumination. By transforming image data into scale-invariant coordinates relative to local features, SIFT enables the presented algorithm to track specific control points over successive frames.

The first stage of feature point detection is to identify locations and scales that can be assigned with repeatability under different views of the same object by searching for stable features across all possible scales using a continuous scale space. The scale space of an image is defined as a function, $L(x, y, \sigma)$, that is produced from the convolution of a variable-scale Gaussian function, $G(x, y, \sigma)$ with an input image $I(x, y)$[7].

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \tag{1}$$

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \tag{2}$$

A DoG (Difference-of-Gaussian) function is then used to obtain the scale-normalized $\sigma^2\nabla^2 G(x,y,\sigma)$. The normalization of the Laplacian with the factor $\sigma^2$is the critical step and is required for true scale invariance.

$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) && (3) \\ &= L(x, y, k\sigma) - L(x, y, \sigma) && (4) \end{aligned}$$

Once DoG images have been obtained, a feature point is identified as local minima/maxima by comparing it to its 26 neighbors, 9 above, 9 below, and 8 surrounding. If it is the highest or lowest pixel value, it is selected as a local extrema or “keypoint candidate” [7].

But scale-space extrema detection, produces too many “keypoint candidates”, some of which are unstable. Performing a detailed fit to the nearby data for location, scale, and ratio of principal curvatures allows the algorithm to filter out points that have low contrast or are distorted by noise [7]. The derivative of the Taylor expansion of the sample space D(x,y,σ) is used to find the location of the extremum $\hat{x}$.

$$D(x) = D + \frac{\partial D^T}{\partial x} x + \frac{1}{2} x^T \frac{\partial^2 D}{\partial x^2} x \tag{5}$$

$$\hat{x} = -\frac{\partial^2 D^{-1}}{\partial x^2} \frac{\partial D}{\partial x} \tag{6}$$

The function value at the extremum,$D(x)$, is used to reject unstable extrema with low contrast. This can be obtained by substituting equation for $\hat{x}$, into the equation for $D(x)$ where we obtained

$$D(\hat{x}) = D + \frac{1}{2} \frac{\partial D^T}{\partial x} \hat{x} \tag{7}$$

$|D(\hat{x})|$ indicates how much of a local extremum a keypoint is. Keypoints are then discarded if they are underneath a certain threshold so that we are left with only keypoints that have high contrast [7].

The next step is to assign orientations to each feature point location based on local image properties, which allows a descriptor to be represented relative to this orientation and therefore achieve invariance to image rotation. This operation is performed for each scale σ. For each sample $L(x, y)$at a certain scale σ, the vector/gradient magnitude, $m(x, y)$ and orientation $\theta(x,y)$ are used as keypoint descriptors to uniquely describe each keypoint. The keypoint descriptor is enables the algorithm to search for the same keypoint in future image frames based on the region surrounding the keypoint itself [7].

III. Frame-by-Frame Registration

Registration is the process of transforming the different sets of data obtained from sampling the same scene at different times into one coordinate system in order to compare or integrate the data obtained from multiple time periods. The central idea behind the registration algorithm is to track features across successive frames of the video sequence; the tracked features are used as control points to which a frame-to-frame parametric registration model is fitted, with either an affine or projective transformation. The features are identified in a robust way that increases their likelihood of being tracked across the frames, described in section 2. Registration is based on the assumption that the pixels which make up the moving object will be statistical outliers from a model of the scene which has been constructed over an extended period of time and thus by performing registration, the effects of the camera motion are removed [8].

Longer time intervals create the potential of platform and camera maneuvers which may make referencing frame 1 directly with frame x>>1 very difficult because the feature points available in the first frame are no longer available in the xth frame. Because the video sequence changes dramatically over the course of a few seconds, often times the entire background is completely different. In order to solve this paradigm, a reference is created indirectly through overlapping feature points as the feature points in frame 1 disappear, new feature points will appear and be classified and registered. Visually, the registration process outputs a video stream that is displayed in sets of these stacks/groups with the first frame of each stack being the corresponding reference image for each

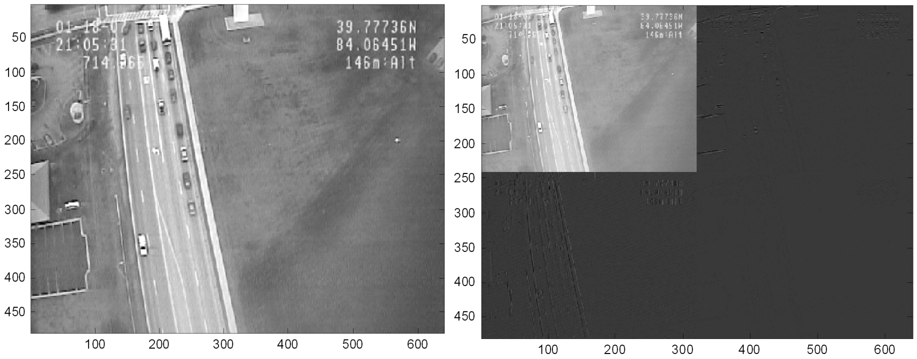

Fig. 1. *Results of 1 level Discrete Wavelet Transform (DWT).*

set of frames. The scene "resets" every N frames so that the field of view never completely disappears.

A. Affine Transformation

Affine transformation is defined as the geometrical relation between two frames such that any given point in one figure corresponds to one specific point in the other figure and is represented as a 2x2 matrix, $\boldsymbol{A}$, capturing the parameters of an linear transformation between a feature point $i$with coordinates $x_i(t)$ and $y_i(t)$ in reference frame $\boldsymbol{F}_t$ and the same feature point i with coordinates $x_i(t+1)$ and $y_i(t+1)$ in target frame $\boldsymbol{F}_{t+1}$ while the translation components of the transformation are captured by $\boldsymbol{a}$ as shown in eq. (8).

$$\begin{bmatrix} x_i(t+1) \\ y_i(t+1) \end{bmatrix} = \mathbf{A} \begin{bmatrix} x_i(t) \\ y_i(t) \end{bmatrix} + \mathbf{a} \qquad (8)$$

$$\text{where} \qquad \mathbf{A} = \begin{bmatrix} a_{xx} & a_{xy} \\ a_{yx} & a_{yy} \end{bmatrix}, \quad \mathbf{a} = \begin{bmatrix} a_x \\ a_y \end{bmatrix} \qquad (9)$$

Using the detected feature locations in both the reference and target frames for feature points i = 1,2,... , we can solve for the transformation needed to undo camera motion. As there are more equations provided by the large number of feature points than there are variables to solve for, we solve the system of linear equations in eq. (11) via least squares for $\boldsymbol{A}$ and $\boldsymbol{a}$ [9]

B. Multiresolution Registration

The aforementioned SIFT based registration algorithm is extremely robust but not computationally efficient; a stabilization system must run at a near real-time rate to be of any operational value in the field. The algorithm operated at a speed of approximately frames per second when run on a 2.4 GHz dual core laptop. Running this algorithm, even when using suitable parallel specialized hardware, would not be feasible in real time. In order to remedy this problem, a multiresolution registration approach is introduced. The principle of the Haar wavelet based transform can best be described by figure 1 which shows the transform using a two channel filter bank.

The upper left sub-image is obtained by lowpass filters in both the horizontal and vertical directions, also referred to as low-low-pass (LL). The other three subimages display much lower intensity and have details involving high frequencies as they are passed through varying high-pass filters. The LL image is then passed into our stabilization algorithm where feature points are detected and registration is performed. By systematically reducing the number of feature points across the image, the algorithm's computational efficiency is greatly increased. Additionally, the influence of noise of the local frames is reduced by the filtering process. The original image will be referred to as the bottom level while the LL image will be referred to as the upper level.

IV. Implementation Of Multiresolution Video Stabilization Algorithm

We test our multiresolution video stabilization algorithm with a video sequence captured at 30 frames/second with each image composed of 480 pixels by 640 pixels. A free-source MATLAB based implementation of SIFT was obtained from [10] and modified to include a top level registration algorithm, a multiresolution registration algorithm, and an accuracy/refinement algorithm. The entire process was performed offline with an Intel Core 2 Duo 2.4 GHz laptop with 4 GB of RAM.

The SIFT feature point detection portion of the algorithm detects 952 feature points in the first frame and 967 feature points in the second frame. Based on the keypoint descriptors illustrated in Section 2.4, 322 of the first 952 feature points in the first frame are determined to be the same feature points in different locations in the second frame. In order to increase computation speed while minimizing accuracy degradation, the targeted frames are passed through a DWT function which reduces the influence of noise, whose results are displayed in Fig. 2. 302 feature points are detected in the first frame while 324 are detected in the second frame and of these feature points, 105 of them are declared matches. The relation between feature points is described as a 3x3 matrix, $\boldsymbol{H}$, capturing the parameters of an affine transformation between reference frame 1 with feature points $\boldsymbol{P}$ and reference frame 2 with

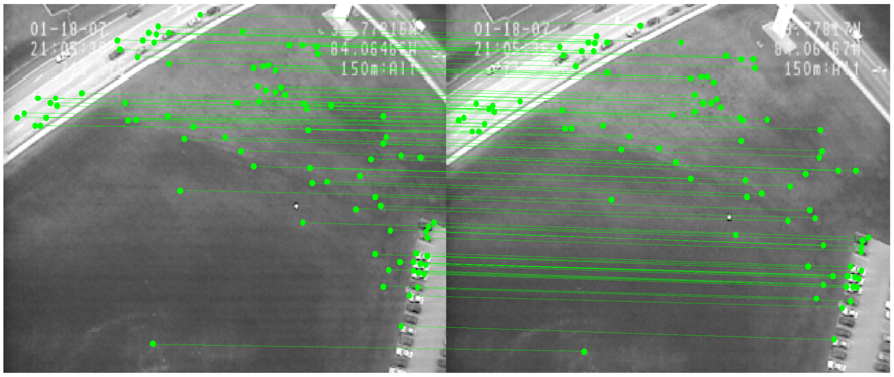

Fig. 2. *Left Image displays 480x640 unprocessed frames and Right Image displays 240x320 frames passed through DWT. There are 322 matches in bottom level while there are 105 matches in upper level*

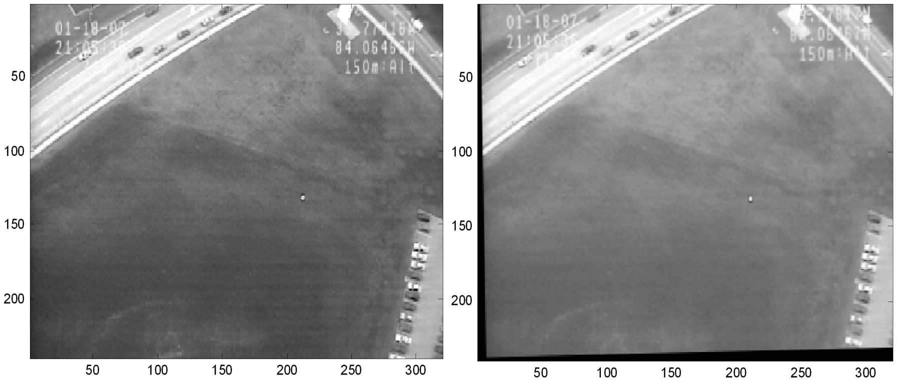

Fig. 3. *Both Images display 240x320 frames. Left frame is unprocessed reference frames and Right Image displays target frame homographic transformation applied. Black area on the right hand side image represents points that exist in target frame but not in reference frame*

feature points ***P'***. This matrix ***H*** is stored for each feature point so that the information can be recalled later when the frames need to be shifted in a certain manner to eliminate ego-motion.

In order to actually stabilize the image, these keypoints must be referenced from frame to frame and their trajectory stored so that the algorithm can undo the motion of the camera. By establishing the displacement of stationary feature points, we can see how much the targets actually moved. Before stabilization, this is skewed because we do not know how much the camera contributes to the movement of the targets. Using homographic transformation at the upper level, the second frame is translated into the first as shown on the right side of Fig. 3. You can see that the motion of the camera has been effectively removed. Frame 2 is now shifted with an affine transformation to reverse the movement of the camera. There is a black edge around frame 2 because that region was not part of frame 1 and so no reference can be established. Feature points and background that are not part of the reference frame but are introduced in later frames are removed. However, the black region is not discarded, it is retained so that it can be mapped with the next group of frames. If these feature points come back into the picture, they are also reintroduced into the background model. However visually, the black border will eliminate any pixels not found in the reference frame. This is why we have stacks of images because after a certain time when there are no more corresponding pixels between the current frame and frame 1, the reference frame must be renamed or else the entire frame will be blacked out.

The transformed target frame at the upper level must then be registered back to the bottom level target frame. In order to do so, the original high pass filtered information is used and through simple multiplication reconstruction, the original size image is obtained. Fig. 4 shows the final product of our stabilization scheme. Comparing the right hand side of Fig. 3 and the right hand side of Fig. 4, we can see that they are not exactly the same. Because of the DWT decomposition, each registered pixel at the upper level corresponds to four pixels at the bottom level, which leads to registration errors at the bottom level.

V. Discussion

In order to exactly quantify the computational improvement and measure the loss in accuracy, a metric to quantify speed

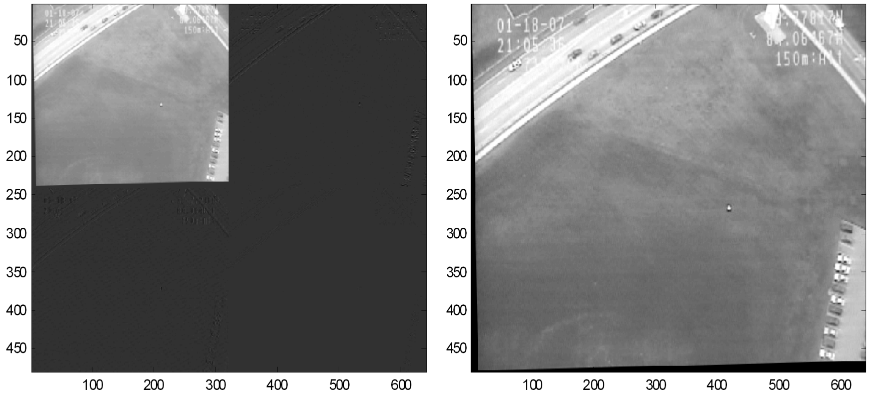

Fig. 4. *Adding back in the high pass filtered components to the registered upper level frame results in a registered version of the 480x640 frame*

TABLE I
Speed Gains of Multiresolution approach. 1 Level DWT results in a 292% increase in speed with the most noticeable time reduction in the Frame Matching component.

	Scale Space Computation	Keypoint Detection	Descriptor Computation	Frame Matching	**Total Time**
SIFT (Original)	3.808 s	0.293 s	3.614 s	0.755 s	**8.470 s**
SIFT(1 Level MR)	0.951 s	0.095 s	1.052 s	0.061 s	**2.159 s**

gains and accuracy was established. In order to measure computation efficacy, a measurement of the processing time for each algorithm component was established. Accuracy measurements were made by subtracting the homographic transformed frame from the reference frame in a 50x50 region in the middle of the image and taking the root mean squared of the pixel intensity differences. I exhibits the speed comparisons while table 2 exhibits the accuracy comparisons between the multiresolution registration approach and the original algorithm.

TABLE II
Accuracy Degradation of Multiresolution approach. 1 Level DWT results in a 21.6% increase in root mean square error of pixel intensity differences.

	Root Mean Squared Error
SIFT (Original)	5.097
SIFT(1 Level MR)	6.1990

Because the computational time is a function of the size of the inputted image, we sought to reduce the amount of processed data by applying the Haar wavelet transform, which filters out pixels that are unlikely to be considered features. The Haar wavelet transform reduces the amount of processed data to one quarter of the original data size, theoretically enabling us to quadruple our processing speed using the multiresolution approach. Due to some overhead computational processing, the actual increase in computation speed was 392% of the original speed as shown in table 1.

The trade-off for the processing speed is a marginal increase in root mean square error of 21.6% as shown in table 2. Therefore, by inflicting a 21.6% increase in RMS registration error, we are able to quadruple (392%) our processing speed. The marginal increase in error is acceptable because without the trade-off, our algorithm is impossible to implement in real-time. Currently, SIFT registration is a post-processing algorithm that runs at $\leq$ 1 Hz, far too slow for any real-time processing. Our multi-resolution program runs at 4 Hz which in-line with our Our multi-resolution program runs at 4 Hz which in-line with our GPS-sampling rate needed for real-time UAV trajectory estimation. By utilizing this speed-error tradeoff, we are able to implement our stabilization algorithm in real-time (in our case, at the GPS update rate).

VI. Conclusion

This paper presents the development of a very robust and consistent SIFT based registration algorithm designed to stabilize video data and eliminate ego motion for moving sensors, in particular unmanned aerial vehicles. The presented algorithm registers video frames over a short temporal window and draws on a number of algorithms from the computer vision community library and combines them in a novel, robust stabilization algorithm. In particular, existing algorithms leveraged included SIFT feature tracker, DWT, RANSAC and global registration. The final product of this algorithm provides a suitable video sequence to which target tracking and other image processing techniques can be applied similar to the way they would be implemented in a static camera system. The algorithm was tested in a variety of scenarios offline and performed adequately in every situation.

The main focus of a video stabilization algorithm is to deliver the best system possible while maintaining real-time implementation capability. The indicated system in this paper has demonstrated significant speed improvements at the expense of marginal error increases. Given suitable parallel non-specialized hardware, the algorithm allows a near real-time solution to the moving sensor paradigm.

Readers may notice that the high frequency components resulting from the DWT of the target image were not used in this implementation and may question why a wavelet transform was utilized instead of another decomposition scheme. Future development of our algorithm will include a KLT-SIFT based method to more accurately access the correct regions of the quadtree, which will require analysis of high frequency components. This will enable the algorithm to perform multi-level DWT on the image further increasing speed, while maintaining accuracy. Thus, the wavelet decomposition scheme is an appropriate method for image decomposition.

The limitations of the commonly used KLT stabilization algorithm were presented in multiple scenarios. The algorithm presented in this paper was applied to the same scenarios and successfully stabilized the same video sequence with significantly fewer misregistration errors than the KLT based algorithm had. The KLT algorithm would not be deployable in a featureless region or in the presence of sporadically placed 3D structures. In terms of computation speed, the presented algorithm compares very favorably with the KLT algorithm, obtaining a near real time frame rate.

Moving video target tracking is a critical and challenging problem in the electrical engineering domain that has several applications in aeronautics. Because almost all guided navigation systems require this technology to have a high degree of autonomy, this technology is a critical component in several applicable fields. Target tracking and recognition is an ongoing challenge and will be critical to developing stable robust tracking systems that can be used in automated spacecraft. Without this technology, automated flight and control are not possible.

References

[1] Hong, S., Atkins, E., "Moving Sensor Video Image Processing Enhanced with Elimination of Ego Motion by Global Registration and SIFT", *IEEE International Conference on Tools with Artificial Intelligence* , November 2008

[2] Weber, M., Welling, M., Perona, P., "Unsupervised learning of models for recognition," *European Conference on Computer Vision* , Dublin, Ireland, pp. 18-32, 2000

[3] Overman, K., Leahy, K., Lawrence, T., Fritsch, R., "The Future of Surface Surveillance, Revolutionizing the Battlefield," *IEEE National Radar Conference*, pp. 1-6, August 2000

[4] Seeger, S., Häusler, G., "A Robust Multiresolution Registration Approach", *Proc. of Vision, Modeling and Visualization,* pp. 75-82, 1999

[5] Moon, H., Chellappa, R., Rosenfeld, A., "Optimal Edge-Based Shape Detection," *IEEE Transactions on Image Processing,* Vol. 11, No. 11, 1209- 1227, November 2002

[6] Jones, R., Booth, D., Redding, N., "Video Moving Target Indication in the Analysts' Detection Support System," *Defense Science and Technology Organization*, Australia, pp. 1-39, 2006

[7] Lowe, D., "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, Vol. 60, No. 2, pp. 91-110, Nov 2004

[8] Gelfand, N., Mitra, N., Guibas, L., Pottman, H., "Robust Global Registration," *ACM International Conference Proceeding Series* , Vol. 255,Article no. 197, July 2005

[9] Brown, A., Sullivan, K., Miller, D., "Feature-Aided Multiple Target Tracking in the Image Plane" *Proceedings of the SPIE, The International Society for Optical Engineering,* v 6229, no. 1, pp. 62290Q.1-12, 2006

[10] http://vision.ucla.edu/∼vedaldi/code/sift/sift.html