https://doi.org/10.2352/ISSN.2470-1173.2021.17.AVM-175

Unify The View of Camera Mesh Network to a Common Coordinate System

Haney W. Williams
Department of Systems Engineering
Colorado State University
Fort Collins, Colorado, USA
Haney.W.Williams@gmail.com

Steven J. Simske, PhD
Department of Systems Engineering
Colorado State University
Fort Collins, Colorado, USA
Steve.Simske@colostate.edu

Fr. Gregory Bishay, PhD
COCC, City of Orange
Orange County, California, USA
frgregb@gmail.com

Abstract

The demand for object tracking (OT) applications has been increasing for the past few decades in many areas of interest, including security, surveillance, intelligence gathering, and reconnaissance. Lately, newly-defined requirements for unmanned vehicles have enhanced the interest in OT. Advancements in machine learning, data analytics, and AI/deep learning have facilitated the improved recognition and tracking of objects of interest; however, continuous tracking is currently a problem of interest in many research projects. [1] In our past research, we proposed a system that implements the means to continuously track an object and predict its trajectory based on its previous pathway, even when the object is partially or fully concealed for a period of time. The second phase of this system proposed developing a common knowledge among a mesh of fixed cameras, akin to a real-time panorama. This paper discusses the method to coordinate the cameras' view to a common frame of reference so that the object location is known by all participants in the network.

Keywords: *Continuous Tracking, Object Tracking, Depth Estimation, Stereo Vision, Trajectory Prediction, Surveillance.*

1. Introduction

Object tracking is an active research area in computer vision thanks to the increasing demands in the Intelligence, Surveillance and Reconnaissance (ISR) applications and the Autonomous Vehicles Systems (AVS). Many algorithms have been developed to track the Object of Interest (OOI) across the view of the camera, and even predict its position when it is obfuscated; however, the tracking system doesn't coordinate its finding about the OOI position with nearby cameras [2]. This paper discusses ways to resolve this issue, and will introduce a method to unify the mesh of cameras to a common coordinate system and relay information about the OOI on a common grid without prior knowledge of the location and orientation of the cameras as shown on Figure 1-1 below.

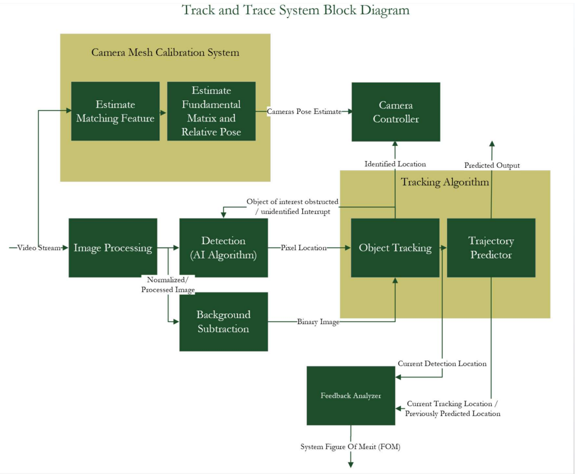

Track and Trace System Block Diagram
Camera Mesh Calibration System
Estimate Matching Feature
Estimate Fundamental Matrix and Relative Pose
Cameras Pose Estimate
Camera Controller
Identified Location
Predicted Output
Tracking Algorithm
Object of interest obstructed / unidentified Interrupt
Video Stream
Image Processing
Detection (AI Algorithm)
Pixel Location
Object Tracking
Trajectory Predictor
Normalized/ Processed Image
Background Subtraction
Binary Image
Current Detection Location
Feedback Analyzer
Current Tracking Location / Previously Predicted Location
System Figure Of Merit (FOM)

Figure 1-1: Proposed System Block Diagram

2. Surveyed Positions Solution

This solution requires to have a priori knowledge of the location and orientation of each camera in the mesh [11].

Assuming that latitude and longitude (lat/long) of each camera are expressed as $\varphi\ and\ \lambda$(radians) respectively; then the distance can be calculated using the great-circle between two points, also known as the ‘haversine’ formula. In the case shown on Figure 2.1 below, the focal center of the image was chosen to be the reference lat/long point of the camera. The distance is calculated as following:

$$a = sin^2(\frac{\Delta\varphi}{2}) + cos(\varphi_1)\, cos(\varphi_2)\, sin^2(\frac{\Delta\varphi}{2})$$

$$c = 2\, atan2(\sqrt{(a)}, \sqrt{(1-a)})$$

$$d = R\, c$$

Where R is the earth mean radius (6,371km), a is the square of half the chord length between the points and c is the angular distance in radians. To properly localize the object of interest (OOI) a stereo vision system must be in place. The vector $\bar{v}$ pointing to the OOI centroid is expressed by an azimuth and elevation. Thus, the distance from any camera on the network can be calculated by adding these aforementioned steps. The advantage of this method is its accuracy of defining the mesh parameters; whereas the main disadvantage of this method is the amount of information needed makes it harder to be autonomous and self-calibrating.

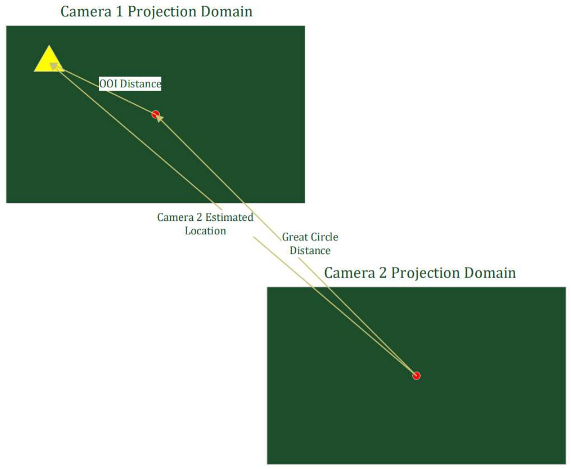

Camera 1 Projection Domain
OOI Distance
Camera 2 Estimated Location
Great Circle Distance
Camera 2 Projection Domain

Figure 2-1: Known Camera Locations

3. Distant Point Calibration Solution

Given two stereo camera systems, the general idea of a this method to determine where the OOI is with respect to a second stereo camera system is to determine, via the calibration process, the relative alignment of the cameras using the “distant point method” explained below and the relative position vector connecting the cameras [3]. The relative position vector is determined by using the cameras relative alignment and their view the same object that is close enough for a reasonably accurate position determination from the stereo cameras; in other words, no external position survey is required. Using the results of the aforementioned camera calibration which will be discussed later in this section, and the first camera system’s position vector measurement of the OOI position, the expected position of the OOI with respect to the second camera can be calculated. The OOI’s expected position with respect to the second camera can be far out of its field of view because that position can be converted to a fully spherical azimuth and elevation to which the second camera can be commanded to point. The advantages of this method are:

- The expected azimuth and elevation with respect to the second camera can be far out of its field of view (e.g. behind or far above where the camera is pointing).
- The mathematics is much simpler, and therefore easier to debug than the mathematics of determining the camera’s relative alignment and relative positions using multiple parallax observations of the same objects by two non-stereo camera systems
- Trajectory estimation/prediction are not required.

Once the camera calibration is completed, the equation below provides the OOI expected position with respect to the second camera given the information from the first camera from which an azimuth and elevation can calculated with the following equation and as Figure 3-1 shows below.

$$[R_2^{OOI}]_2 = [R_2^1]_2 + C_1^2[R_1^{OOI}]_1$$

Where,

$[R_2^{OOI}]_2$ is the position vector of OOI relative to camera 2 in camera 2 frame of reference coordinate system.

$[R_2^1]_2$ is the position vector of camera 1 relative to camera 2 in camera 2 frame of reference coordinate system.

C_1^2 is the direction cosine matrix which transforms camera 1 vector to coordinates to vector coordinates of camera 2 frame of reference coordinate system.

Lastly, $[R_1^{OOI}]_1$ is the position vector of OOI relative to camera 1 in camera 1 frame of reference coordinate system.

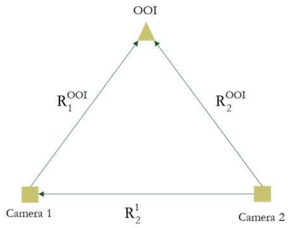


Figure 3-1: OOI Position Calculation

When equation described above is solved for the "position vector of camera 1 relative to camera 2 in camera 2 coordinates" the position measurements made by the two stereo cameras provide the calibration process's determination of the relative position between the cameras. If a position survey were to be used, the camera's orientation relative to the Earth would be needed because survey coordinates such as latitude, longitude, and altitude are relative to Earth Centered Earth Fixed axis. Obtaining the camera's orientation relative to Earth would be very inconvenient. The distant point method of determining the camera's relative alignment will finally be discussed. The fundamental principal employed is that the directions of position vectors connecting the cameras to distance points such as stellar constellations do not depend on the camera's position. Thus, if the two cameras measure the directions specified by a unit vector or equivalently azimuth and elevation of three distant points, two different views of the same coordinate system are obtained. The coordinates of each camera's view of the common coordinate system is used to determine the direction cosine matrix relating the cameras.

Each distant point of the three will be expressed in a different coordinate system for each of the camera. We can express these unit vectors to the three points with $\widehat{d_1}, \widehat{d_2}, \widehat{d_3}$. We can orthogonize the system using the Gram-Schmidt as the following equations then normalize the system.

$$\begin{cases} \widehat{D_1} = \widehat{d_1} \\ \widehat{D_2} = \widehat{d_2} - \dfrac{\widehat{d_2^T}\,\widehat{D_1}}{\widehat{D_1^T}\,\widehat{D_1}}\,\widehat{D_1} \\ \widehat{D_3} = \widehat{d_3} - \dfrac{\widehat{d_3^T}\,\widehat{D_1}}{\widehat{D_1^T}\,\widehat{D_1}}\,\widehat{D_1} - \dfrac{\widehat{d_3^T}\,\widehat{D_2}}{\widehat{D_2^T}\,\widehat{D_2}}\,\widehat{D_2} \end{cases}$$

Where $\widehat{D}$ the axis system is common to all the cameras but different coordinates for each camera; in other words, they are different coordinate because each camera is pointed differently; however unit vectors $\widehat{D_1}, \widehat{D_2}, \widehat{D_3}$ point in the same direction because the points are too far away.

The Direction Cosine Matrix $C_{axis\ 1}^{axis\ 2}$ transforms the axis from system 1 to system 2. It is expressed in the following matrix:

$$C_{axis\ 1}^{axis\ 2} = \begin{bmatrix} \langle\widehat{x_2},\widehat{x_1}\rangle & \langle\widehat{x_2},\widehat{y_1}\rangle & \langle\widehat{x_2},\widehat{z_1}\rangle \\ \langle\widehat{y_2},\widehat{x_1}\rangle & \langle\widehat{y_2},\widehat{y_1}\rangle & \langle\widehat{y_2},\widehat{z_1}\rangle \\ \langle\widehat{z_2},\widehat{x_1}\rangle & \langle\widehat{z_2},\widehat{y_1}\rangle & \langle\widehat{z_2},\widehat{z_1}\rangle \end{bmatrix}$$

Where $\langle\widehat{x_2},\widehat{x_1}\rangle$ are the inner product of vectors $\widehat{x_2}\ and\ \widehat{x_1}$; where $\widehat{x_1}, \widehat{y_1}\ and\ \widehat{z_1}$ are the basis vector coordinates of the two axis systems

Thus, changing from the old coordinate to the new coordinate can be expressed as following:

$$\begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = \boldsymbol{C_{axis\ 1}^{axis\ 2}} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix}$$

The Direction Cosine Matrix C_1^D, C_D^2 relating the two cameras, where C_1^Dtransforms a given vector from camera 1 axis to D axis described above, and C_D^2 transforms a given vector from D axis to camera 2 axis can be calculated as following:

$$C_1^D = \begin{bmatrix} D_{1x} & D_{1y} & D_{1z} \\ D_{2x} & D_{2y} & D_{2z} \\ D_{3x} & D_{3y} & D_{3z} \end{bmatrix} \ ,\ C_D^2 = \begin{bmatrix} D_{1x} & D_{2x} & D_{3x} \\ D_{1y} & D_{2y} & D_{3y} \\ D_{1z} & D_{2z} & D_{3z} \end{bmatrix}$$

Where $\widehat{D}$ is the new axis coordinate system in the C_1^D transformation where its components are calculated by the Gram-Schmidt above and expressed as the following:

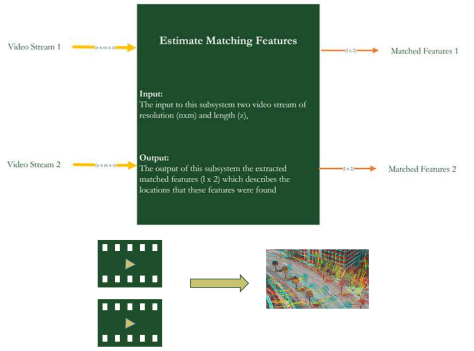
$$\widehat{D_1} = \begin{bmatrix} D_{1x} \\ D_{1y} \\ D_{1z} \end{bmatrix} \ ,\ \widehat{D_2} = \begin{bmatrix} D_{2x} \\ D_{2y} \\ D_{2z} \end{bmatrix} \ ,\ \widehat{D_3} = \begin{bmatrix} D_{3x} \\ D_{3y} \\ D_{3z} \end{bmatrix}$$

Thus, the transformation from camera 1 to camera 2 can be calculated as following $C_1^2 = C_D^2\, C_1^D$.

The main disadvantage of this method is that a stereo camera needs to be used for every camera position.

4. Point Correspondence Solution

This method of coordinating and calibrating the camera network relies on overlap between the cameras. The system is composed of two subsystems. The first subsystem extracts the matching features between two frames of a video feeds from two difference sources. This is done by detecting the edges and corner, then it extracts the neighborhood features to these corners and edges. Next it finds the matching features in the correspondent image. The subsystem is shown on Figure 4-1 below.



Input:
The input to this subsystem two video stream of resolution (nxm) and length (z),

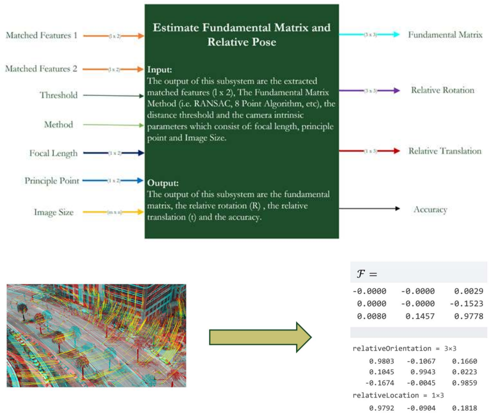
Output:
The output of this subsystem the extracted matched features (l x 2) which describes the locations that these features were found

Figure 4-1: Estimate Matching Feature Subsystem

The second subsystem estimates the Fundamental Matrix $\mathcal{F}$ and estimates the relative pose between the two cameras; in other words, it estimates the relative rotation and translation between the cameras [8]. There are several methods to estimate the Fundamental Matrix for example:

- The Random Sample Consensus algorithm (RANSAC)
- The M-Estimator Sample Consensus (MSAC) which converges faster than RANSAC.
- The Least Median Squares algorithm (LMedS).
- Least Trimmed Squares (LTS) which converges faster than LMedS.
- Or by the 8 point correspondent algorithm developed by Longuet-Higgins.

The estimate pose is calculated and it is dependent on the camera intrinsic calibration. Figure 4-2 below shows the block diagram of the second subsystem.



Input:
The output of this subsystem are the extracted matched features (l x 2), The Fundamental Matrix Method (i.e. RANSAC, 8 Point Algorithm, etc), the distance threshold and the camera intrinsic parameters which consist of: focal length, principle point and Image Size.

Output:
The output of this subsystem are the fundamental matrix, the relative rotation (R) , the relative translation (t) and the accuracy.

```
𝓕 =
-0.0000   -0.0000    0.0029
 0.0000   -0.0000   -0.1523
 0.0080    0.1457    0.9778

relativeOrientation = 3×3
    0.9803   -0.1067    0.1660
    0.1045    0.9943    0.0223
   -0.1674   -0.0045    0.9859
relativeLocation = 1×3
    0.9792   -0.0904    0.1818
```

Figure 4-2: Estimate Fundamental Matrix and Relative Pose Subsystem

In other words, this method leverages the stereo vision concept and applies it to a much higher scale.

If camera X locates the Object of Interest, All the cameras in the network can coordinate their relative position to that camera X, by performing the cumulative transformation. The next section will describe the algorithms used for each subsystem and will show the results.

5. Results

5.1. Dataset Gathering

The data was generated by a movie created by Google Earth Studio, The cameras were then placed at random positions where there was some overlap between them. Figure 5-1 shows below the trail of the camera where each white dot shows the major keyframe that was used as camera placement.

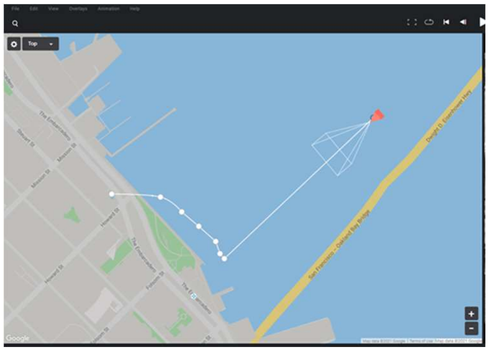


Figure 5-1: Synthesized Data Overview

5.2. Algorithm Detailed

In the first subsystem mentioned above, the features of the image are detected using the Harris Features detection algorithm. Then the features were extracted between the images by a combination of algorithms namely Speeded-Up Robust Features (SURF) and Fast Retina Keypoint algorithms. Then these features gets corresponded between the images. Figure 5-2 visualizes the point correspondence between the two algorithms.

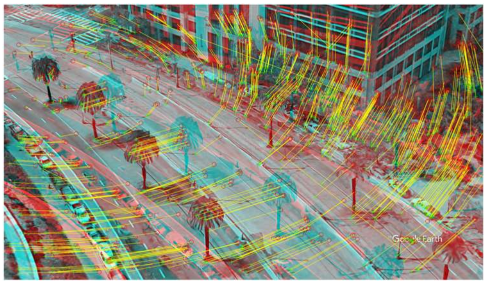


Figure 5-2: Features Correspondence

Figure 5-3 below shows the strongest corresponding points between the two images after removing the outliers.

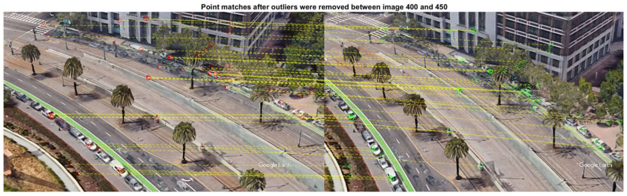
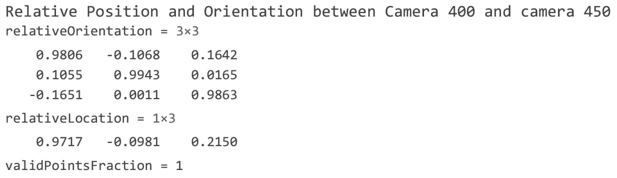

Point matches after outliers were removed between image 400 and 450
Google Earth
Google Earth

Figure 5-3: Synthesized Data Overview

In the second subsystem, the fundamental matrix is generated by the Random Sample Consensus (RANSAC) algorithm such that the following equation is satisfied.

$$x_2^T \boldsymbol{\mathcal{F}}\, x_1 = 0$$

To estimate the relative location of the cameras, the intrinsic properties of the cameras were assumed to be ideal for distortion and skew factors since the data was synthesized. The focal length is assumed to be 3000 millimeters in the x and y directions and the optical center of the camera is exactly in the middle. The focal length for simulated data is infinite, an analysis was performed to achieve a realistic estimate. Focal length was tested from 1600 to 5000, beyond that 3000 there was insignificant improvement; thus focal length of 3000 was chosen. Figure 5-4 below shows the result of the relative orientation and relative position of the two cameras. Where camera 1 (on left) is placed at the origin (0,0,0) and camera two (on the right) is relatively placed based on the position described by the Rotation and Translation matrices.



```
Relative Position and Orientation between Camera 400 and camera 450
relativeOrientation = 3×3

    0.9806   -0.1068    0.1642
    0.1055    0.9943    0.0165
   -0.1651    0.0011    0.9863

relativeLocation = 1×3

    0.9717   -0.0981    0.2150

validPointsFraction = 1
```

Figure 5-4: Relative Position and Orientation

5.3. Scene Reconstruction and Testing

To assess the proposed algorithm, the scene has been reconstructed by triangulating the matched points calculated by the correspondence algorithm that was discussed in the previous section. Figure 5-5 below shows the scene reconstruction.

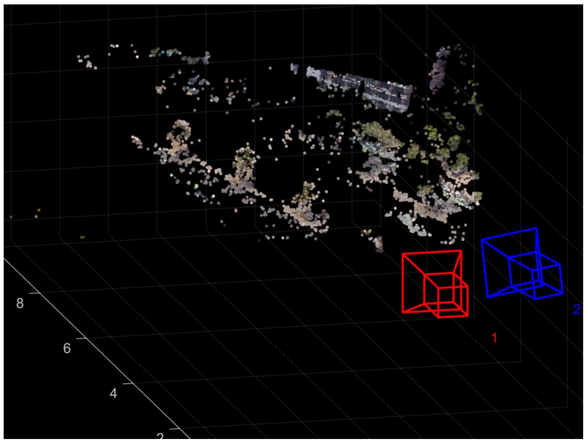


Figure 5-5: Scene Reconstruction

From the reconstructed scene, several points were chosen randomly to remap them into the projected space onto the two images, as shown by Figure 5-6(a,b) and Figure 5-7(a,b) below. These set of images show there is an error when comparing the two images; for instance, the point chosen shows that it is at the corner of the building by the middle of the window, whereas the second camera remaps it into the corner of the building by the top of the window.

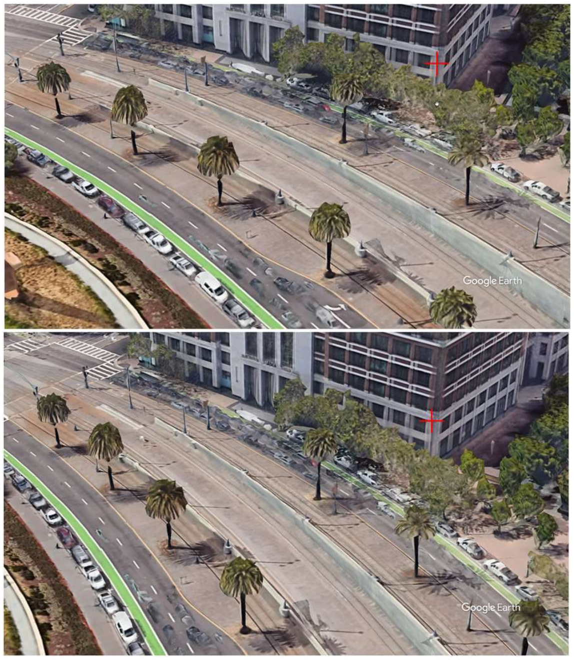


Figure 5-6: First Example (a - Top) Remap onto Camera1, (b - Bottom) Remap onto Camera2.

Similarly, for the second example, there was an error when comparing the two images. Thus, the next section will discuss the method used to quantify the associated error.

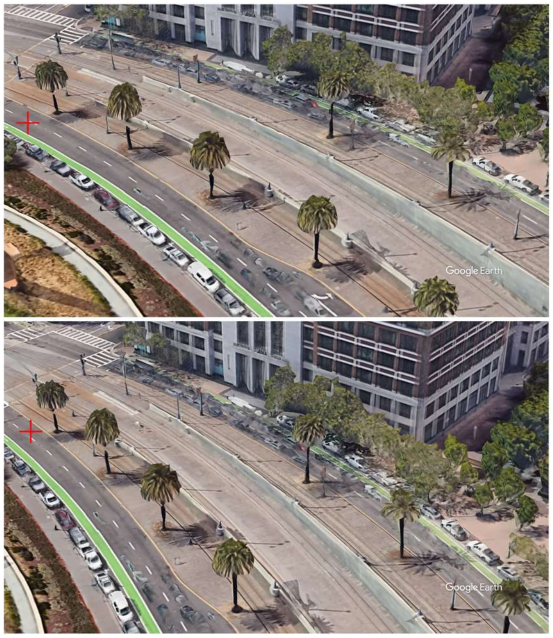


Figure 5-7: Second Example (a - Top) Remap onto Camera1, (b - Bottom) Remap onto Camera2.

5.4. Error Estimation

The error was estimated by performing a normalized 2-D cross-correlation between a template taken from image one and the a section of image 2. For example, Figure 5-8 (a) shows the template to be chosen as the 50x50 pixels from the center of the remapped point from camera 1; Figure 5-8(b,c) show a window around the of 200x200 pixels from the centers of the remapped points from camera 1 and camera 2 respectively.

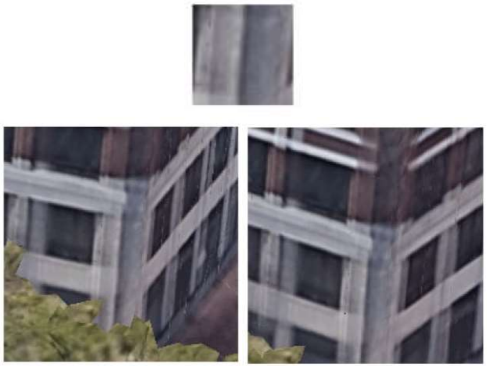


Figure 5-8: (a – top) Template (b - Left) Remapped ROI Camera1, (c – Right) Remapped ROI Camera2.

Figure 5-9 shows the results of the 2-D cross-correlation as a surface map, where x,y are the pixels of the image and the z axis is the correlation coefficient magnitude.

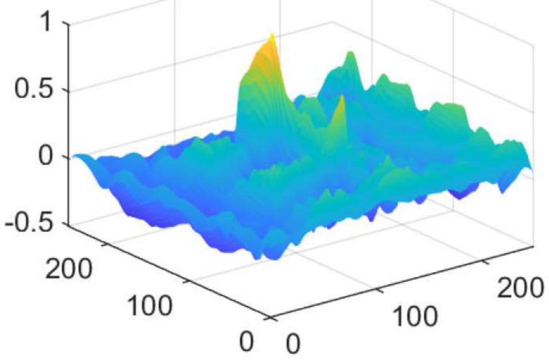
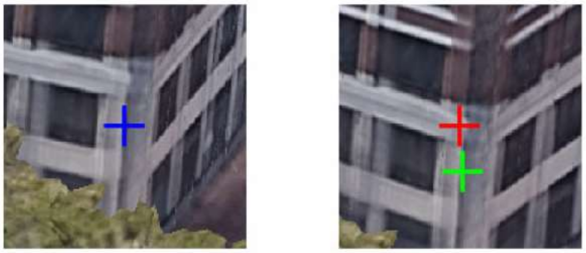


Figure 5-9: Cross-Correlation Result

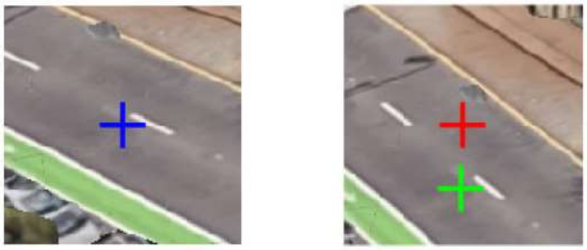
Figure 5-10 below shows the estimated error. (a) shows the original location from camera 1 indicated by the blue-cross/dark-cross; whereas, (b) shows the original mapped location indicated by the red-cross/dark-cross and the found location with the cross-correlation indicated by the green-cross/light-cross. The error is estimated to be -38 in the Y direction and about -2 in the X direction.



```
distanceY = -38, distanceX = -2
```

Figure 5-10: (a – Left) Original Mapped location Camera1, (b - Right) Error Estimated On Camera2

Similarly, the same process was done to the second random point discussed above. In this case, the error was estimated to be -40 in the Y direction and 1 in the x direction as shown on Figure 5-11.



```
distanceY = -40, distanceX = 1
```

Figure 5-11: (a – Left) Original Mapped location Camera1, (b - Right) Error Estimated On Camera2

6. Future Development

The first enhancement to the proposed algorithm will deal with optimizing the cameras distances and pose estimation [10], where the minimum overlap between the two images will be estimated. Another enhancement is to optimize the execution time; currently the execution time for 7 cameras is approximately 15.2 seconds. Another enhancement to the system is to implement a method that mimics the idea of MPEG-2 "I" frame to change the frame of reference after N number of cameras to minimize the error from accumulating the rotation and translation from one camera to another. Lastly, the aforementioned system will be integrated with the overall system described by the previously published paper to locate the Object Of Interest (OOI) [5][6][7].

7. Conclusion

This paper discussed three different methods to correlate a mesh of camera network to a common knowledge. The first method had to have a priori knowledge of the camera location and orientation. The second method correlate the cameras based on several non-orthogonal distant points such as stellar constellation. The last method proposed a system that leverages the idea of stereo vision at a larger scale, and the fact that cameras are available everywhere nowadays. In the later system, points in the scene are corresponded and the scene is reconstructed in three dimensions where a common knowledge about the object of interest can be inferred. This paper discussed the results from the synthesized data that was created to build the algorithm prototype.

8. Acknowledgement

Section 3 of this paper and the research behind it have greatly benefited from the remarkable insights my dear colleague Victor Zahn, Boeing Global Services System Engineer. His enthusiasm and keen knowledge in the subject matter had been substantial to this research.

9. References

[1] H. Williams and S. Simske, "Object Tracking Continuity through Track and Trace Method" *Electronic Imaging, Autonomous Vehicles and Machines*, pp.299-1-299-7(7), 2020.

[2] L. Maddalena and A. Petrosino, "A Self-Organizing Approach to Background Subtraction for Visual Surveillance Applications," *IEEE,* vol. 1057, no. 7149, pp. 1169-1177, 2008.

[3] C. Veness, "Calculate distance, bearing and more between Latitude/Longitude points", https://www.movable-type.co.uk/scripts/latlong.html, 2020.

[4] S.-C. S. Cheung and C. Kamath, "Robust techniques for background subtraction in urban traffic video," *in SPIE 5308, Visual Communications and Image Processing* , San Jose, CA, , 2004.

[5] L. Maddalena and A. Petrosino, "A self-organizing approach to detection of moving patterns for real-time applications," *in 2nd Int. Symp. Brain, Vision and Artificial Intellicence*, Springer, Berlin, Heidelberg, 2007.

[6] L. Maddelena and A. Petrosino, "A self-organizing approach to detection of moving patterns for real-time applications," http://www.na.icar.cnr.it/~maddalena.l/HSV-SO/HSV-SO2007.html, 2007.

[7] N. Pradyumna, R. Beveridge and B. Draper, "Gesture Recognition: Focus on the hands" *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5235-5244, 2018.

[8] QT. Luong and O.D. Faugeras, "The fundamental matrix: Theory, algorithms, and stability analysis." *Int J Computer Vision* 17, 43–75, 1996.

[9] P. Torr and D. Murray, "The Development and Comparison of Robust Methods for Estimating the Fundamental Matrix." *International Journal of Computer Vision* 24, 271–300, 1997.

[10] A. Trabelsi, M. Chaabane, N. Blanchard and R. Beveridge, "A Pose Proposal and Refinement Network for Better 6D Object Pose Estimation." *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 2382-2391, 2021.

[11] V. Nastro and U. Tancredi, "Great Circle Navigation with Vectorial Methods" *The Journal of Navigation*, Vol. 63, Iss. 3, 557-563 Cambridge, July, 2010.

[12] V. Zahn, "Distant Point Correlation", Personal Correspondence, Sep. 2020

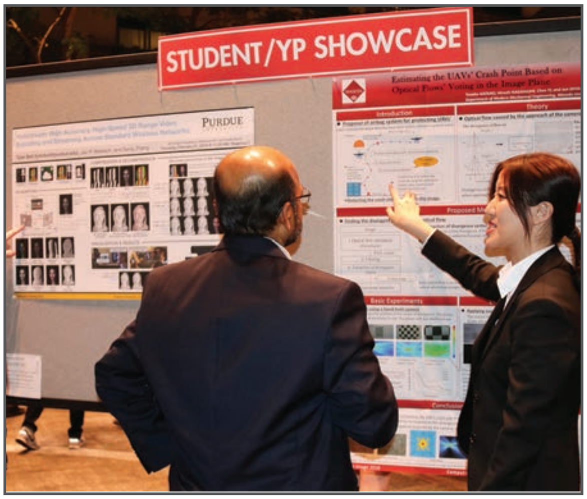
JOIN US AT THE NEXT EI!

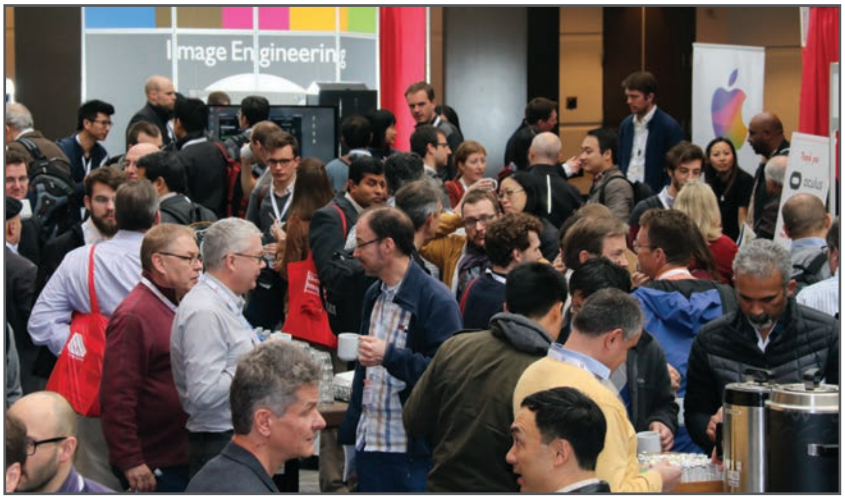
IS&T International Symposium on

Electronic Imaging

SCIENCE AND TECHNOLOGY

Imaging across applications . . . Where industry and academia meet!





AR is a fundamental and transformative interface and platform.

20+ years from my initial working demo to today's AR displays

Optimistic that it won't take another 20 to reach ubiquitous consumer AR!

• SHORT COURSES • EXHIBITS • DEMONSTRATION SESSION • PLENARY TALKS •
• INTERACTIVE PAPER SESSION • SPECIAL EVENTS • TECHNICAL SESSIONS •

www.electronicimaging.org

IS&T
imaging.org