USING 3-D VIRTUAL REALITY MODELS FOR IMAGE ORIENTATION IN MOBILE COMPUTING

Charalampos Georgiadis, Anthony Stefanidis, Peggy Agouris

Dept. of Spatial Information Science and Engineering
348 Boardman Hall
University of Maine, USA
{harris, tony, peggy}@spatial.maine.edu

Commission V, WG V/5

KEY WORDS: Close Range, Motion Imagery, Orientation, Virtual Models, Modelling

ABSTRACT:

Advances in sensor technology and computing capabilities and modalities are revolutionizing close-range image collection and analysis for geospatial applications. These advances create the need for new ways of handling and processing video datasets at quasi real time rates. In this paper we present an innovative two step orientation technique for ground level motion imagery using a 3 dimensional virtual model as control information. In the first step few select anchor frames are orientated precisely via an image orientation-through-queries approach. In the second step, intermediate frames are orientated relatively to these anchor frames through an innovative analysis of building façade variations in them. Combined, these two steps comprise a complete approach to motion imagery orientation using a VR as control information.

1. INTRODUCTION

During the last years we are experiencing great advances in sensor technology and wireless communications. That progress created new data collection schemes, in which users can collect data roaming a scene with a GPS enabled digital camera or camcorder. These collection schemes create a vast amount of data that has to be stored and processed. As a result new techniques have to be developed that allow for fast processing of motion imagery, either offline or in a quasi real time manner. In the context of this paper we use the term motion imagery to refer to imagery collected at video rates, or even as select frames captured a few seconds (or even minutes) apart, using either a video or a still camera. Examples of such datasets include imagery collected by hand-held cameras captured while roaming an urban environment, or imagery collected by a network of fixed sensors (e.g. surveillance cameras) monitoring a scene. The computing capabilities of mobile devices are also improving with the advances in technology, creating the opportunity of location based computing. In order to be able to acquire modeling information from such data we have to use some kind of control information, the traditional techniques of acquiring control points are still time consuming. In addition research advances in the development of 3 dimensional virtual models of large scale complex urban scenes have resulted in the creation of impressive and complex VR models. These advances provide opportunities for the integration of such models in motion imagery analysis. The accuracy and complexity of these models provide an excellent use as control information for motion image processing.

In this paper we present an approach for the recovery of sensor orientation and position information using a two-step procedure. We focus on the use of motion imagery datasets (at quasi-video rates) captured by sensors roaming an urban scene. In such a scenario the orientation variations among successive frames are small. The first step of our approach entails the comparison of object configurations depicted in a frame to corresponding configurations identified in the VR model. This provides excellent approximate values that are then refined (using precise matching) to provide the orientation parameters for that frame. This is a process that can be performed for few anchor frames (e.g. every few minutes) to provide accurate orientation information at these instances. For subsequent frames we estimate their orientation by determining their variation from the nearest anchor frames. To do so we use 2-D transformations of objects depicted in these frames, and orientation differences between the frames computed using vanishing points to translate these transformation parameters to orientation variations. Combined, these two steps comprise a novel process of progressive orientation recovery that meets the computational requirements of mobile mapping applications. Most of the known techniques for computing relative orientation need points in different planes. In [Simon and Berger, 2002] a similar approach is presented in which they estimate the orientation of an image using the previous image of known orientation and a planar homography between these two images. In another approach [Chia et al, 2002], compute the relative orientation based on one or two reference frames, exploiting epipolar geometry and using recursive methods. Our approach tries to solve this problem using points from only one plane, and using linear methods. Furthermore the information acquired during the orientation process can be used to update the existing virtual model. Such update procedure includes change detection both in geometric and radiometric content, in the existing objects and detection of new objects or deletion of objects in the model. In our paper we present the approach for the orientation estimation in the intermediate frames.

The paper is organized as follows. In section 2 we present an overview of our navigation-through-virtual models approach. In Section 3 we present the indices we use for comparing the intermediate frames with anchor frames for orientation recovery. Experimental results in section 4 demonstrate the performance of our approach and conclude with future work plans in section 5.

2. MOBILE IMAGE ORIENTATION: GENERAL OVERVIEW

We assume that we have a GPS-enabled camera roaming a scene that is partially (or completely) covered in a 3D model database. Sensor imagery is tagged by a time stamp, while the GPS sensor allows us to tag each frame with approximate position information. Our objective is to determine the camera's pose and update the sensor's location.

Our approach can be characterized as a two step procedure:

- the first step is the use of an image query-based scheme to determine the approximate location and orientation of few select anchor frames, and
- the second step entails the relative orientation of the remaining frames (relative to the anchor frames)

Thus we proceed by determining directly the precise orientation parameters of few anchor frames, and then determine minor corrections to these parameters in order to express the orientation of the intermediate frames. This is visualized in Fig. 1. Anchor frames may be selected in pre-determined temporal intervals (e.g. once every a couple of minutes), or at pre-determined spatial intervals (e.g. once every 50 meters).

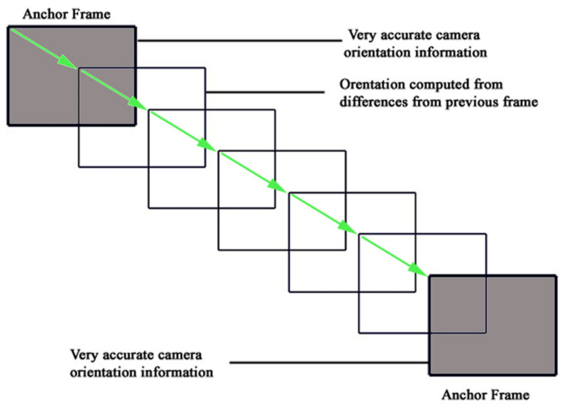

Figure 1 Proposed two step approach scheme

As we can see the proposed scheme has a similarity with the MPEG compression standards. In MPEG compression few frames in the video sequence are chosen to act as anchor frames, and they are compressed as JPEG files. For the rest of the frames the MPEG compression scheme saves only changes between consecutive frames. Drawing from this MPEG philosophy we proceed by computing directly accurate sensor position information in few select instances (the equivalent of anchor frames). The orientation of intermediate frames is recovered by analyzing changes in image content (location, and size of object facades in them).

In figure 2 we can identify the main algorithmic steps of our approach. We can identify two clusters of processes, corresponding to anchor frame processing (left) and intermediate frame pose estimation (right). Our work on anchor frame orientation estimation through image queries has been presented in some extent in [Georgiadis C. et al, 2002]. Briefly, we should mention here that our innovative approach integrates image queries with image registration and sensor orientation. Classic image queries have as a goal to retrieve images from a database based on certain image characteristics. In our approach we use image queries to recover sensor orientation information by comparing abstract metrics of a scene configuration in an image to the corresponding configuration in a geospatial database. This is complemented by an adjustment of co-linearity equations to determine sensor position. Thus we integrate image retrieval and orientation estimation in a single step.

The advantage of this orientation-by-queries approach to anchor frame orientation is that it produces very accurate results, while its drawback is that it requires good approximate values in order to initialize it. However, this is in accordance with our overall assumed modus operandi. As we assume the use of a GPS-enabled camera in an urban environment, it is realistic to consider that the accuracy of the initial approximations of sensor locations is in the order of 3-10 meters. This is visualized in Fig., 3, with the big red sphere representing the uncertainty of the approximation (the actual location can be anywhere within this sphere).

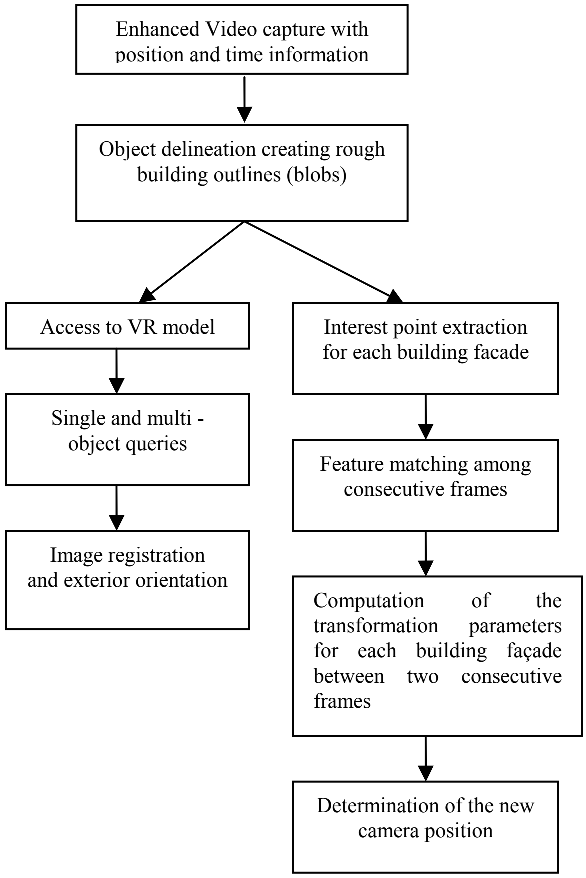

Figure 2 Approach outline

We already have approximate values for the position of the camera by using the GPS sensor, but we don't have any information about the rotation angles. The nature of the problem (close range applications) makes the whole system sensitive to the rotation angles and noise. We assume that the rotation of the camera axis will be near to zero so our problem is to find the approximate value just for one rotation angle, specifically the rotation angle around the Z axis in a world reference system, which basically the azimuth the angle between the true north and where our camera is looking.

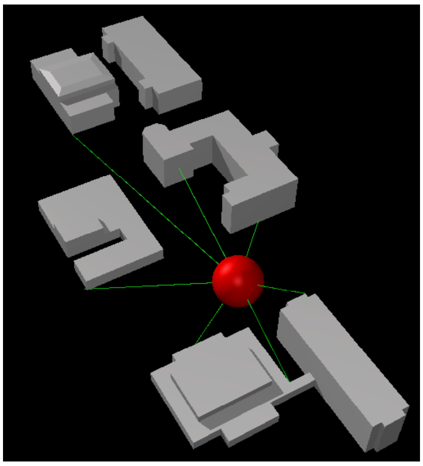

Figure 3 Visualization of initial sensor position and orientation uncertainty

In order to estimate this angle we use a queries scheme instead of classical photogrammetric techniques in the pixel level. The query scheme is a two-part process, one part using single object query scheme while the second part processes a multi object configuration. For further information on our single query approach the reader can refer to [Stefanidis A. et al, 2003], while for the multi object queries [Stefanidis A. et al, 2002]. After the estimation of the parameters we run a least squares adjustment and produce accurate coordinates for the camera position and rotation.

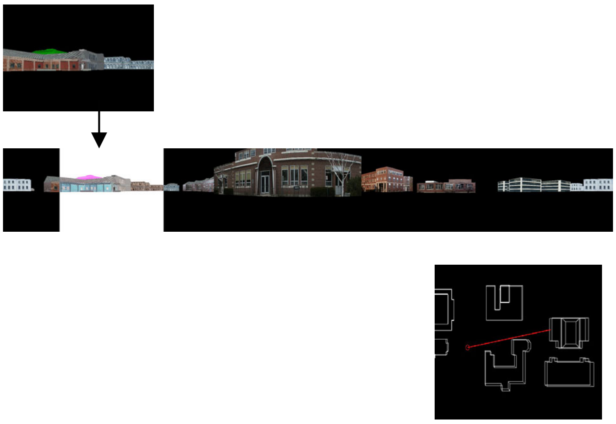

Figure 4 Representation of the anchor frame procedure

In figure 4 we can see a representation of how the anchor frame orientation scheme works. The top image is the one captured by our sensor, in the middle image we can see the panorama created with the help of the virtual model. The highlighted portion of the middle image depicts the position of the captured image as computed using the single and multi object queries. Finally the bottom image shows the sensor's location and orientation after precise matching is performed in the query results.

In intermediate frame orientation, which is the focus of this paper, we aim to recover the orientation of intermediate frames by orienting them relative to the nearest anchor frames. In order to accomplish this goal we developed a framework to translate object representation variations (i.e. changes in an object's size, location, and orientation within an image frame relative to the same object's image in an anchor frame) into orientation variations (i.e. changes in the orientation parameters of the corresponding frame relative to the anchor frame). Thus we develop a dynamic image orientation scheme that allows us to recover image orientation for every frame in our feed using few, select oriented anchor frames. The nature of our data collection modus operandi (sensors roaming urban scenes) implies that small differences will occur in sensor location and rotation between consecutive frames.

This process is visualized in figure 5 where we see a portion of a 3-dimensional virtual model of an urban scene. Using anchor frame orientation in an orientation-through-queries process we have already determined the orientation of the sensor in position A. using the second step we will determine the orientation in position B. In figure 6 we can see the two captured images, left image captured in position A, and right image captured in position B. Our objective in this case is to compute a relative orientation between the two captured images and using the orientation information about position A to compute the new position B.

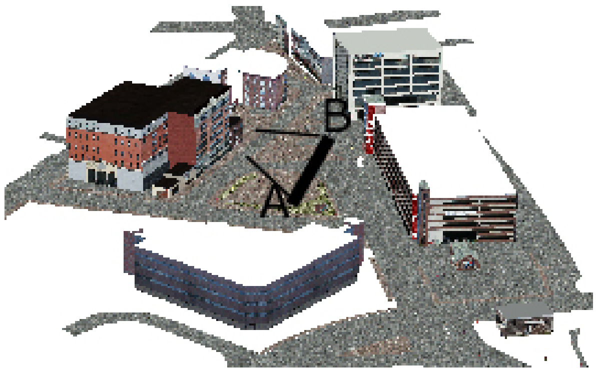

Figure 5 Portion of 3 dimensional Virtual model

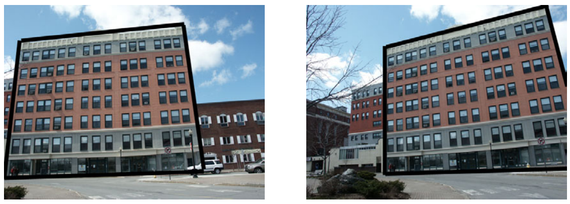

Figure 6 Consecutive frames captured from sensor, with the façade of a building delineated in them.

3. PROPOSED APPROACH

In this section we are going to analyze the procedure that allows the computing of relative orientation between two consecutive frames. For that procedure we assume that we have absolute orientation information for the first image and also that in the first image we know the real world coordinates for the objects that appear in it. We also assume that we for each building façade we know their corner points in both images. Our observations are object facades, which we consider to be planar elements. We are going to follow a two step procedure. The first step is to compute the rotation angles between the two sensor positions, while the second one will allow us to compute the translation between the two sensors.

For the computation of the rotation angles in each image we use vanishing points. The advantage of using vanishing points is that can work with only on object in the image and in the worst case scenario with only a portion of a building façade provided that we can find lines parallel to the outline of the building. We assume a local coordinate system in which the X axis of the image is the X axis in the building façade (width of the façade), the Y axis of the image is parallel to the Y axis in the building façade (height of the façade) and the Z axis is the distance from the sensor position. The two coordinate systems are shown in figure 7.

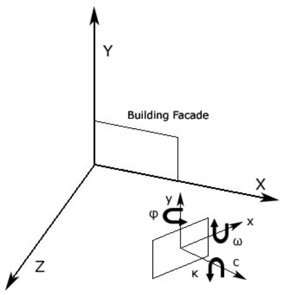

Figure 7 The two coordinate systems

As shown in [Petsa E, Patias P., 1994] we are able to calculate the three rotation angles in a local coordinate system parallel to the object using only two directions on the image plane. In figure 8 we can see the determination of the vanishing points. The rotation angles and focal length can be computed with the following equations:

$$c = \sqrt{-x_H x_V - y_H y_V} \tag{1}$$

$$\tan\kappa = \frac{x_V}{y_V} \tag{2}$$

$$\tan\phi = \frac{c}{y_H \sin\kappa - x_H \cos\kappa} \tag{3}$$

$$\tan\omega = \frac{c}{x_V \sin\kappa + y_V \cos\kappa} \tag{4}$$

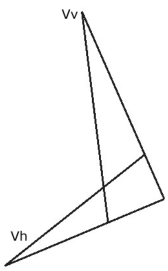

Figure 9 Vanishing Points computation

After the determination of the rotation angles we proceed in the computation of the translation. In order to achieve this task we use the previous information of the rotation angles. From the rotation angles and a known line segment in one image we can compute the new distance of the sensor from the object.

$$H_i = \frac{S}{\dfrac{R^i_{11}xi_a + R^i_{21}yi_a - R^i_{31}ci}{R^i_{13}xi_a + R^i_{23}yi_a - R^i_{33}ci} - \dfrac{R^i_{11}xi_b + R^i_{21}yi_b - R^i_{31}ci}{R^i_{13}xi_b + R^i_{23}yi_b - R^i_{33}ci}} \tag{5}$$

$$H_j = \frac{S}{\dfrac{R^j_{11}xj_a + R^j_{21}yj_a - R^j_{31}cj}{R^j_{13}xj_a + R^j_{23}yj_a - R^j_{33}cj} - \dfrac{R^j_{11}xj_b + R^j_{21}yj_b - R^j_{31}cj}{R^j_{13}xj_b + R^j_{23}yj_b - R^j_{33}cj}} \tag{6}$$

if we name the denominator of equation 5 A and the denominator of equation 6 B, and take the ratio of the two equations we have:

$$\frac{H_i}{H_j} = \frac{\dfrac{S}{A}}{\dfrac{S}{B}} = \frac{A}{B} \Leftrightarrow H_j = H_i \frac{B}{A} \tag{7}$$

Using equation 7 we can compute the range of the second image knowing the range of image 1, the coordinates for a known line segment in both images, and the rotation angles of the images. From these equations we are able to derive the DZ component in our local coordinate system.

We proceed by creating a quasi rectified version of the two images using the rotation angles. In order for the image to be fully rectified we have to use the projective transformation. In our case we use the rotation angles to rotate the image points in plane parallel to the plane of the façade. As a result the two quasi rectified images have the same orientation. In figure 10 we can see the procedure, in the left image we have the two sensor position and the rotation angles as recovered from the vanishing points, while in the right image we can the system after the quasi rectification procedure. We will use these two images to compute the translation of the sensor along the X and Y axis of our coordinate system.

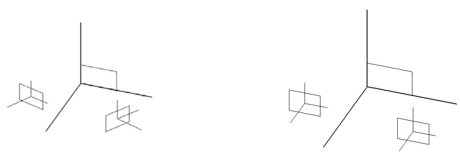

Figure 10 The quasi rectification procedure

We can compute the scale of each point in the images using the rotation information and image range, DZ that we have already determined.

$$\frac{1}{scale} = \frac{R_{13}x + R_{23}y - R_{33}c}{Z - Z_0} \tag{9}$$

Where $Z\text{-}Z_0$ is the range of the image in our local coordinate system and refers to DZ. Computing the scale of the four points in the two images we can compute a mean scale for each image and furthermore to compute a relative scale between the two images. In order to compute the DY, and DX we will use the translation of the points from the quasi rectified images, and the scale factor between the two images and the scale of the known orientation image.

$$DX_{ij} = R_{S_{ij}} * Sc_i * dx_{ij} \tag{10}$$

$$DY_{ij} = R_{S_{ij}} * Sc_i * dy_{ij} \tag{11}$$

Where R_{Sij} is the relative scale between the two images, Sc_i is the scale of image i and d_{xij}, d_{yij} is the translation of the points in the quasi rectified images.

4. EXPERIMENTS

In order to test our approach we contacted a series of experiments. In this setup we only used just a plane object in our image. We created a simulated dataset for 16 different sensor positions. For each position we have the coordinates in real space and the supposed rotation angles. In figure 11 we can see a top view of the setup and how an image is viewed in station 7, while in table 1 we can see the orientation values for each station in the local coordinate system. In this setup we only processed the bold façade of the building.

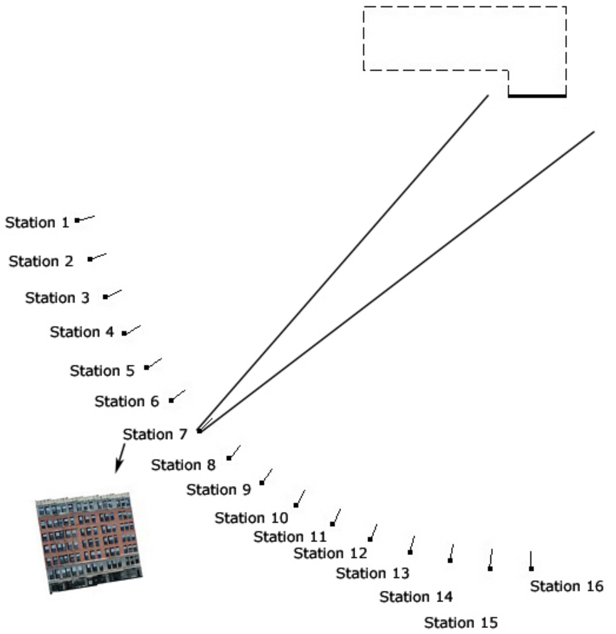

Figure 11 Simulation dataset setup.

	X_0 (m)	Z_0 (m)	Y_0 (m)	φ deg	ω deg	κ deg
Station 1	-83.8307	-29.5928	4	-75	5	30
Station 2	-81.2074	-37.9129	5	-70	10	25
Station 3	-77.8689	-45.9727	6	-65	15	20
Station 4	-73.8407	-53.7109	7	-60	20	15
Station 5	-69.1533	-61.0685	8	-55	25	10
Station 6	-63.8426	-67.9896	6	-50	20	5
Station 7	-57.9488	-74.4216	5	-45	15	0
Station 8	-51.5169	-80.3153	4	-40	10	-5
Station 9	-44.5958	-85.6261	3	-35	5	-10
Station 10	-37.2381	-90.3134	2	-30	0	-15
Station 11	-29.5	-94.3417	1	-25	-5	-20
Station 12	-21.4402	-97.6801	0	-20	-10	-25
Station 13	-13.12	-100.3035	-1	-15	-15	-30
Station 14	-4.603	-102.1916	-2	-10	-20	15
Station 15	4.013	-103.3303	-1	-5	-25	45
Station 16	12.7619	-103.7109	0	0	0	0

Table 1 Station Orientation Information

In Table 2 we can se he results for the recovery of the rotation angles using the vanishing points approach. we can see that the angles ω, κ were recovered very accurate while for the rotation φ the highest error in the recovered accuracy was around 2 degrees, which is accurate enough for our applications. We also run the full algorithm in a different dataset created using the same stations but only φ, and κ rotation angles. The results are presented in Table 3. We can see that for a total traveled distance of 130 meters the errors are in the neighborhood of centimeters.

	Errors		
	φ deg	ω deg	κ deg
Station 1	-0.054689	0.000071	-0.000001
Station 2	-0.283546	-0.000084	-0.000002
Station 3	-0.769276	-0.000021	0.000000
Station 4	-1.566688	0.000032	0.000005
Station 5	-2.689448	-0.000085	0.000001
Station 6	-1.763289	-0.000073	0.000003
Station 7	-0.992982	-0.000091	-0.000002
Station 8	-0.431297	0.000083	-0.000001
Station 9	-0.102587	-0.000026	-0.000003
Station 10	-0.000005	-0.000054	0.000000
Station 11	-0.083557	-0.000070	-0.000002
Station 12	-0.280196	-0.000008	0.000001
Station 13	-0.489229	-0.000007	-0.000002
Station 14	-0.591965	-0.000010	0.000002
Station 15	-0.466409	-0.000102	-0.000002
Station 16	0.000000	0.000000	0.000000
Std	0.731904	0.000055	0.000002

Table 2 Accuracy in rotation recovery

	Errors X m	Errors Z m	Errors Y m
Station15	0.049373	0.000000	-0.000012
Station14	0.095555	0.000000	-0.000401
Station13	0.740550	-1.163700	0.067356
Station12	0.080210	-0.000100	-0.014127
Station11	0.106520	-0.000100	-0.016784
Station10	0.123070	-0.000100	-0.021921
Station9	0.129600	-0.000200	-0.027239
Station8	0.126160	-0.000200	-0.029113
Station7	0.113720	-0.000300	-0.030154
Station6	0.093610	-0.000300	-0.030232
Station5	0.067760	-0.000400	-0.029265
Station4	0.038540	-0.000500	-0.027232
Station3	0.008150	-0.000500	-0.024185
Station2	-0.020780	-0.000600	-0.030226
Station1	-0.046070	-0.000700	-0.036643
Std	0.175498	0.290207	0.024736

Table 3 Accuracy in Position recovery

5. FUTURE WORK

In this paper we presented a method for the recovery of orientation between two consecutive frames using a method that first determines the rotation angles and proceeds to determine the translation between the two frames. Using the presented approach we achieved very good results for the recovery of the rotation angles but we used very accurate measurements in the image points. We also achieved very good results in position recovery, but we took into account only two of the three rotation angles in the creation of our dataset. Another aspect of the created dataset is that we only used one object (building façade) in our approach. We are planning to further examine the behavior of algorithm by creating datasets with different cases of pathways. We will also like to examine how the algorithm works with the addition of noise in image measurements, so we will introduce noisy images and different kind of lens distortion in our model, and the interior orientation parameters. Finally we would like to explore the behavior of the algorithm when multiple objects are present in the image and how the combinations of multiple solutions each for a different object affect the accuracy of the results.

REFERENCES

Chia K., A. Cheok, S. Prince 2002 Online 6 DOF Augmented Reality Registration from Natural Features, International Symposium on Mixed and Augmented Reality (ISMAR'02) September 30 - October 01, 2002 Darmstadt, Germany pp. 305-313.

Georgiadis Ch., A. Stefanidis, and P. Agouris, 2002 Fast Recovery of Image Orientation using Virtual Urban Models, International Archives of Photogrammetry & Remote Sensing, 34 (5), Corfu, pp. 161-166.

Petsa, E., P. Patias, 1994. Formulation and assessment of straight line based algorithms for digital photogrammetry. International Archives of Photogrammetry and Remote Sensing, 30(5):310-317.

Simon G., M.-O Berger. 2002. Pose Estimation for Planar Structures. *IEEE Computer Graphics and Applications,* 22 (2), pp. 46-53.

Stefanidis A., P. Agouris, M. Bertoloto, J. Carswell, and Ch. Georgiadis, 2002. Scale- and Orientation-Invariant Scene Similarity Metrics for Image Queries, *International Journal of Geographical Information Science*, 16(8), pp. 749-772.

Stefanidis A., Ch Georgiadis., P. Agouris 2003. Registration of Urban Imagery to VR Models Through Radiometric Queries Videometrics VII, SPIE Proceedings Vol. 5013, Santa Clara, CA, pp. 176-185.

ACKNOWLEDGEMENTS

This work is supported by the National Science Foundation through grant ITR-0121269 and by the National Geospatial-Intelligence Agency (NGA) through NMA 401-02-1-2008. We would also like to acknowledge Dr. Arie Croitoru, for providing us with the simulation software.