Where is that Pixel in the Oblique-View Video?

Yin Li
Computer Science Department
National University of Singapore
li_yin@u.nus.edu

Teck Khim Ng
Computer Science Department
National University of Singapore
ngtk@comp.nus.edu.sg

Abstract

We investigated the problem of deducing the geographical coordinates of pixels in an oblique view video. Our goal is to register the oblique-view video of an urban scene with its cadastral map. The oblique-view videos were taken from a very low flying camera whereas the cadastral map contained only the top-down outline of buildings without any photographic content and without other objects such as trees, cars, people, or any street feature. Our registration comprises a two-step process that uses structure from motion and a matched-filter class of technique. The structure from motion step takes the video as input and outputs a 3D point cloud of the scene. As this point cloud contains objects that are not represented in the cadastral map, our algorithm was designed to emphasize automatically the scene points that are likely to be from building façade so that effects of mismatch in content between the cadastral map and the oblique video can be minimized. For the registration step, we used a coarse-to-fine iterative implementation of matched filter to get a globally optimum solution, with built in tolerance to some scale and rotation invariance. We had implemented the entire system and had tested with real world data. Good results were obtained.

1. Introduction

In many applications such as a search and rescue operation, it is important to attach geographical information (such as 3D absolute coordinates) to points of interest in an image. Modern sensors like Global Positioning Satellite (GPS) cannot solve this problem adequately because GPS can only measure the location of the camera but not the location of the remote scene that the camera is viewing. Some kind of reference is needed. One way is to use a reference map with geographical coordinates so that one can align the image with it to derive the geographical coordinates of each pixel in the image. Our choice of reference map is a cadastral map.

We chose cadastral maps because in most countries, cadastral maps of residential and commercial areas are available as they are needed for real estate taxation purposes [16]. Therefore, during a crisis situation, these cadastral maps are often readily available. In a search and rescue operation for example, drones flying near the disaster area will stream back video that captures scenes of the area with civilians that need help. By registering this video with the cadastral map, the civilian's absolute geological location can be inferred so that the rescue team can be dispatched to the correct location.

Such kind of registration is a very challenging problem due to the following reasons: 1) Drones are usually operated at a low altitude with its camera viewing obliquely at a remote scene. Registering an oblique view with a top-down view cadastral map is challenging due to the large difference in viewing angles. 2) Cadastral map is essentially a schematic diagram containing only the 2D outline of buildings without any photographic content of the scene. Trees, cars, people and street features that are present in the oblique video are completely missing in the cadastral map. Registering video with a cadastral map is therefore challenging due to the mismatch in information.

To overcome the above difficulties, we developed a solution by first transforming the oblique-view video scene into its 3D point cloud through structure from motion (SfM). We devised a technique to automatically find the vertical direction of the 3D point cloud and collapse all the 3D points along this vertical direction. In this way, we could successfully convert an oblique video scene into a top-down 2D point cloud so that we could register it with the reference cadastral map. Our method also has the benefit of enhancing the points that correspond to building outlines so that the image matching process is less affected by the mismatch in content between the point cloud the cadastral map. As structure from motion has inherent scale ambiguity problems, we used the navigation sensors on board the drone to get an initial estimate of the scale and let the registration algorithm search for the accurate scale. We used an iterative implementation of matched filter to obtain a globally optimum registration solution. Section 2 describes the related work in this problem. Section 3

explains our technique and Section 4 shows our experiment results.

2. Related Work

Registering airborne nadir view (i.e. top down view) imagery with orthographic reference images had been successfully demonstrated in the past [3,27,28]. These methods however cannot be applied to registering oblique-view images with top-down reference image due to perspective foreshortening, occlusion and the appearance of building facades that are mostly missing in top-down reference views. Researchers aiming to solve such problems often relied on texture, lines and prior knowledge of camera parameters to perform affine transformation or other 2D image transformations in order to warp the oblique-view into a top-down view. There were successful cases of such approach that essentially involve an ortho-rectification process [7,8,20,30]. However, such successes are difficult to be achieved for extremely oblique views for two reasons. Firstly, 2D warping cannot sufficiently account for the vast viewpoint differences between an extremely oblique view and a top-down view. Secondly, these warpings require the detection of image features which may not be a trivial task. For example, a common approach is to do line detection to the oblique view images to recover camera parameters needed for ortho-rectification. Such kind of approach however required the oblique image to contain a rich set of vertical and horizontal lines from building structures. This would not work if the buildings were obscured by natural objects such as trees or if the view was too narrow to contain a rich set of building structures. In our problem scenario, we have to deal with extremely oblique views as our video input is derived from air-drones flying at very low altitudes viewing a remote scene that does not necessarily contain a lot of unobstructed building linear features. Moreover, we are matching images of two different modalities: video and a cadastral map. Our cadastral map contains only the outlines of buildings and does not include any photographic information which could otherwise be used for feature matching with the input video image. Essentially, we need to do cross modality matching to register the oblique view image with a top down non-photo map. Cross-modality image registration had been a very important research topic over the years. The problem of aligning a picture of a scene with a different modality of the same scene would require techniques that exploit the higher order features or statistics that could represent a common layer for the two modalities. Viola and Jones [24] used mutual information to match images of different modalities. Huttenlocher and Ullman [12] as well as Lowe [17] matched projections of 3D points onto 2D edge images. The 3D points are from relatively complete 3D modeling that may not be available in the scenario that this project is trying to solve. Ni [19] used Hough Transform [6] to detect linear features for matching with 2D maps. Wang and You [26] used an interest region extraction and matching approach to successfully match nadir view optical images with LiDAR data. Ding et al. [5], Wang and Neumann [25] matched oblique view images with Lidar based on matching line segments between the Lidar data and image. They combined the line segments into more complex descriptors to achieve matching robustness. Zhao et al. [29] and also Labe and Forstner [15] turned an oblique video into 3D point cloud using structure from motion [11] and register the resultant point cloud with a nadir view 3D point set using Iterative Closest Point (ICP) [2]. To our knowledge, the work that is closest to our problem is by Habbecke and Kobbelt [10] and also Kaminsky et al. [14]. Habbecke and Kobbelt used a combination of vanishing points [1], corresponding corners and lines to compute the camera transformation matrix that would align the oblique view with the top down cadastral map. Their method is not applicable to our scenario as vanishing points and feature correspondences are not easily obtainable in our data set due to presence of trees and other forms of occlusion. Kaminsky used a set of cost functions to align 3D point cloud with building floor plans. Their method requires a relatively large coverage of 3D point cloud to avoid getting trapped in local minima. Our data sets were obtained from relatively narrow field of view with relatively small coverage of the scene, therefore it would be prone to local minima if a simple cost function minimization approach were used.

Other than the traditional methods of geo-registration, researchers had also used sophisticated GPS/INS equipment onboard airborne vehicles to directly associate video recordings with readings from those location-tracking devices. The advantage of this approach is that it gives a good initial estimate of the region of search for purposes of registration. Cramer and Stallmann [4], and also Grenzdorfer [9] developed a specially calibrated system that could achieve registration with good accuracy. Despite achieving small registration errors, such methods have several disadvantages. The biggest disadvantage w.r.t. our problem domain is that the location tracking devices can only return readings for the current location of the aircraft but cannot read the geo-location of the remote area that the camera onboard the aircraft is viewing. Unless the drone subsequently flies to be right on top of the area of interest, one will not be able to directly infer the geo-coordinates of those remotely seen areas.

In this project, we converted the oblique view images into 3D point cloud, estimated the vertical direction, and projected the 3D point cloud onto the horizontal plan so that it could be registered with the cadastral map. We used

an iterative implementation of matched filter to get the globally optimum solution.

3. Methodology

3.1 3D Reconstruction from Oblique Video

We reconstructed the 3D scene from the oblique video using *Structure From Motion (SfM)* [11,21]. We alleviated the scale ambiguity problem by using the onboard inertial navigation sensor. A rough estimate of the scale was thus obtained and used as the initial estimate to be refined in the registration process.

3.2 Obtaining Top-Down View of Reconstructed Point Cloud

Structure from motion produces a 3D point cloud that enabled us to view the scene from any viewpoint. The optimal choice was obviously the top-down view direction that was also used by the reference cadastral map.

To project the point cloud onto a plan view plane, we needed to estimate the geographical vertical direction of the point cloud. This was not a trivial task in general. In our scenario however, the urban scenes that we worked on were mostly situated on relatively flat grounds that were much wider than the building heights. We capitalized on this scenario and made an assumption that the SfM reconstructed point cloud would have very dominant horizontal components and a relatively weak vertical component. In such a scenario, the vertical direction could be robustly identified using Principal Component Analysis (PCA) [13]. The vertical direction of the point cloud was given by the principal component that corresponds to the smallest eigenvalue since the vertical component was the direction of least variance. After finding the vertical direction, we projected the 3D point cloud onto the PCA vector space spanned by the other 2 principal components. The z (up) coordinate became 0 as a result of the projection. The point cloud was then in the same viewing direction as the reference cadastral map.

We also used a filter to remove irrelevant objects while preserving building contours (Fig. 3.1). The filter was designed under the assumption that building walls were mainly vertical. Therefore many points on the same wall surface would be projected to a dense line in the 2D top-down view, making the projected building boundary consisting of dense points. In contrast, the top-down view projection from noise objects such as trees and roofs were more spread-out and therefore less dense. We made use of this heuristic to prune away trees and roof features which were not present in the cadastral map.

3.3 Registering Query Cloud with Reference Cloud

We had discussed how to convert the input video first to 3D point cloud using SfM, then project the point cloud onto a 2D top-down view to facilitate matching with the reference cadastral plan view. We highlighted that we also used a sparse-point removal filter to reduce noise in the projected point cloud. We will refer to this resultant point cloud as the "query cloud" in the sequel. This "query cloud" was to be registered with the reference cadastral map. The reference cadastral map was essentially a binary map that captures only the top-down outline of buildings. We discretized these building top-down outlines to a collection of points and will refer to them as the "reference cloud" from now on.

Our oblique view to top-down cadastral map registration problem has now been transformed into matching an SfM-derived 2D query cloud to its reference cloud. The registration problem remained challenging as the query cloud was still relatively noisy. Its content was also vastly different from the reference cloud as reference cloud did not contain trees and other foreign objects. There were many approaches we could take to tackle this registration problem.

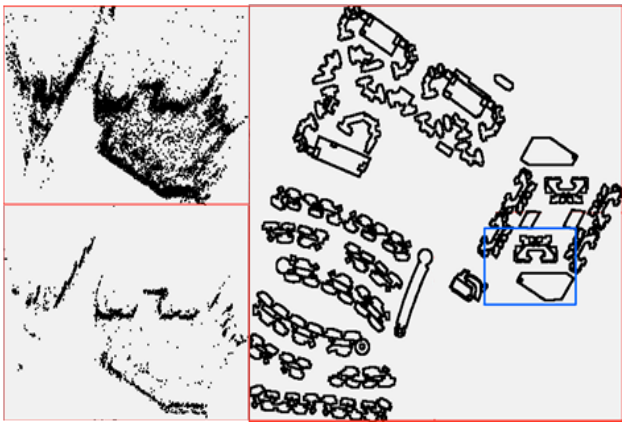


Figure 3.1: Top-left: Top-down view of reconstructed point cloud. Bottom-left: Top-down view of reconstructed cloud with noise points removed. This will be used as the query cloud. Right: The reference cadastral map with lines discretized into points (reference cloud). The blue box indicates the area occupied by the query cloud. Notice that the query cloud covers only a small section of the reference cloud, causing many optimization techniques to be trapped in local minima and failed.

We had tried using SIFT feature matching [18], Hough Transform [6], Iterative Closest Point (ICP) [2] class of techniques as well as matched-filtering. We found that SIFT feature matching did not work due to the presence of noise in the query cloud. Hough Transformation patterns were not distinctive enough to serve as good registration features. ICP's registration accuracy was low as it was prone to be trapped in local minima for the large initial position errors in our problem. We found that an iterative implementation of matched filter produced the best results.

3.4 Matched Filter for Registration

Matched filter is known to be the optimum detection filter in the presence of additive noise. We implemented our matched filter using *Fast Fourier Transform (FFT)*. Doing matched filtering in the frequency domain is significantly faster than doing it in the spatial domain due to the highly efficient FFT algorithm [22,23]. To implement matched filtering in the frequency domain, one needs to multiply the FFT of the reference image with the complex conjugate of the FFT of the query image. An inverse FFT of the product gives the matched filter output in the spatial domain. The matched filter output image will exhibit the brightest and sharpest spot at the location on the reference image that most resembles the query image. In other words, the translation of this spot from the center of the image is exactly the pixel-wise translation of the query image in order to be matched with the reference image. Unlike techniques based on optimizing functions such as ICP, the matched filter will find the global best-match and is comparable with ICP in terms of run time. However, a matched filter has no tolerance to rotation errors unless the synthetic discriminant function class of filters is used [23]. Furthermore, as in ICP, it cannot tolerate scaling errors as well. Therefore, we used a coarse-to-fine iterative method to search for the rotation angle and scale that minimized errors. We limited the search range for rotational error to be ± 10 degrees and the scaling error to be ± 10%. This search range was sufficient as it was within the errors introduced in most inertial navigation devices.

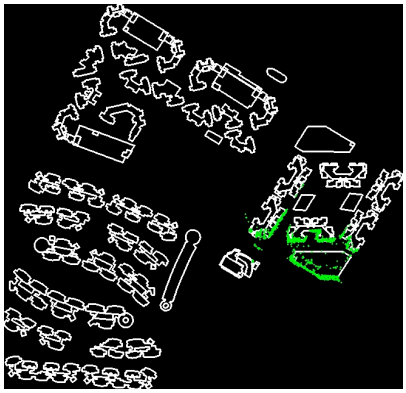
4. Experiments

All video streams used in the experiments were taken by a drone-mounted Sony HDR-CX430V color camera, which produced videos in MTS format with a frame rate of 29 frames/s. The raw frame size was 1920 x 1080 pixel. The drone was also equipped with a GPS with measurement accuracy of ± 10 meters and a compass with measurement accuracy of ±10 degrees.

We conducted tests on a total of 6 different video sequences shot from oblique view angles. 3 of these video streams captured low-rise building clusters sized at 375.25 m (EW) x 342.11 m (NS) whereas the remaining 3 captured high-rise building clusters sized at 657.47m (EW) x 646.30 m (NS).

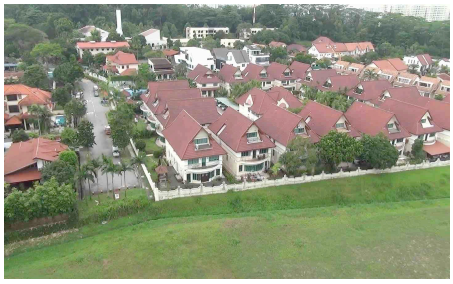
For each of these 6 sequences, we first conducted simulation tests to understand how well our algorithm will work in the presence of large initial position errors coupled with moderate scale and orientation initial errors. We have to be robust to large initial position errors as they could occur due to inaccuracies in compass readings and synchronization of these readings with the video frames captured at long distances. We programmatically injected errors in the initial values of position, scale and orientation. The range of errors was -10 degrees to +10 degrees, from 90% to 110%, and from (-500 m, -500 m) to (+500 m, +500 m) for orientation, scale and position respectively. This resulted in 45 sets of initial values for each of the 6 video sequences.

(a)

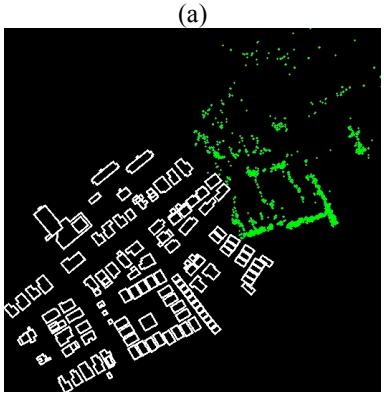


(b)

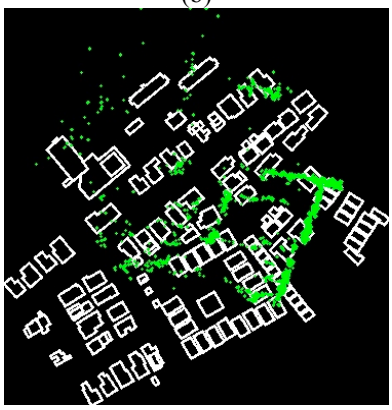
Fig.4.1 Green points represent the query cloud. White points represent the reference cloud. This figure shows an example of registration results with initial translational, rotational and scaling errors. (a) Rotation = 10 degrees from correct value, scale = 1.1 of correct scale, translation = (238.19, -374.187) from correct position. (b) Our matched filter based registration result.



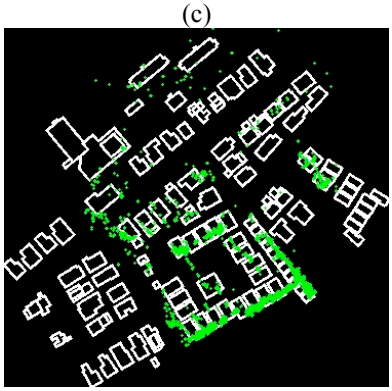
(a)



(b)



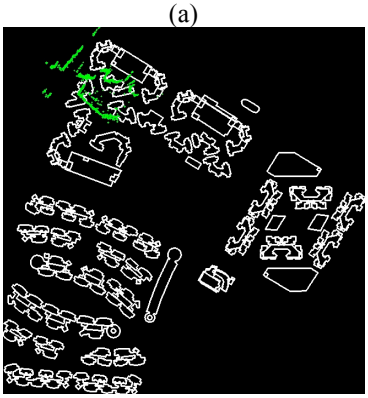
(c)



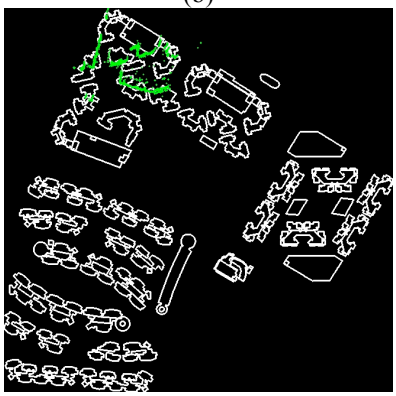
(d)

Fig.4.2 (a) one snapshot of video. (b) initial position of the query cloud (green) on the reference cloud (white). (c) result from ICP. (d) result from our matched filtering

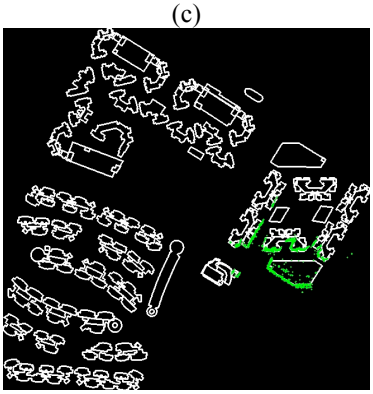
(a)



(b)



(c)



(d)

Fig.4.3 (a) one snapshot of video. (b) initial position of the query cloud (green) on reference cloud (white). (c) result from ICP. (d) result from our matched filtering

Fig.4.1 shows an example of such a test case. Fig.4.1a shows the initial position, scale and orientation of the point cloud. It was translated by 238 and 374 meters in horizontal and vertical directions respectively, and scaled bigger than correct size by 10%, and rotated by 10 degrees from the correct orientation. These errors were within the operation precision range of commercial grade inertia navigation devices. Despite the large error in initial position in addition to moderate errors in scale and orientation, our coarse-to-fine iterative matched filter method was able to produce good registration results (Fig.4.1b). Notice also the relatively small signature of our point cloud w.r.t. the cadastral map that would cause energy minimization approaches like ICP to be trapped in local minima.

Fig.4.2 and Fig.4.3 show the registration results on two more real scenes respectively. Notice the presence of trees and other ground or roof features that are missing in the cadastral map. The mismatch in information, the vast difference in viewpoints between the oblique video and the top-down view cadastral map, and the relatively small signature of the point cloud in the midst of a relatively larger area of the cadastral map present challenges not yet solved by existing techniques. In each of these cases, our matched filter based algorithm was able to find the correct registration whereas ICP method failed due to local minima. ICP was trapped in local minima essentially because of the relatively small signature of the point cloud w.r.t. the cadastral map.

The computer we used for the experiment was equipped with a four-core Intel i7-4700MQ CPU clocked at 2.40GHz. However, at run time, only single thread was used. The memory size was 8G. The average running time of our iterative matched filter and ICP to recover original rotation, scale and position were 466 seconds and 66 seconds respectively (Fig 4.4). We believe that matched filtering timing can be further improved using an image pyramid approach.

We used 6 real-life data sets in total. From these 6 sets of data, we found that the error of our iterative matched filter registration approach produced an average registration error of 0.3 degrees for rotation, 1.6% for scale and 2.2 meters for translation. In a crisis situation, this will be of sufficient precision for rescue team workers to narrow down the search for victims that require urgent help. In comparison, ICP produced an average registration error of 24.98 degrees for rotation, 9.1% for scale and 160.6 meters for translation in these scenarios when initial positioning error was large.

It should be noted that for even greater robustness, one can make use of prior knowledge of rough viewing angle to prune away irrelevant building outlines in the cadastral map prior to the matched filtering process. For example, if one knows the angle of viewing is approximately towards north, building outlines belonging to northern facing facades are likely to be occluded due to the oblique viewing. Therefore, one can safely remove those occluded building outlines from the cadastral reference map based on this prior information. The matching results will potentially be better as there is now a greater match between the 3D point cloud and the pruned cadastral map. Indeed, we had done experiments on this and verified that the methods would result in greater robustness assuming the inertial navigation system gave worse errors as shown in Fig 4.5.

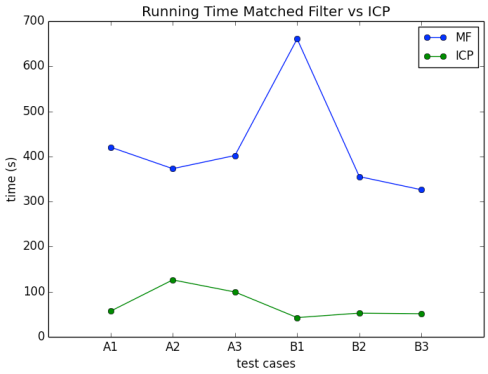
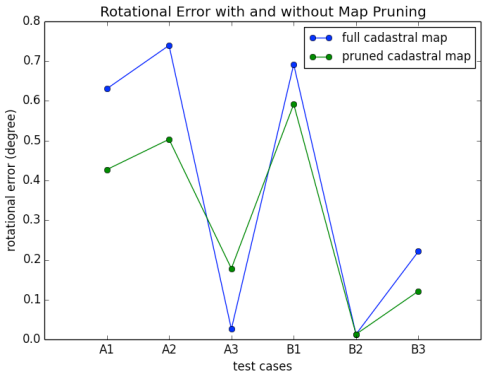


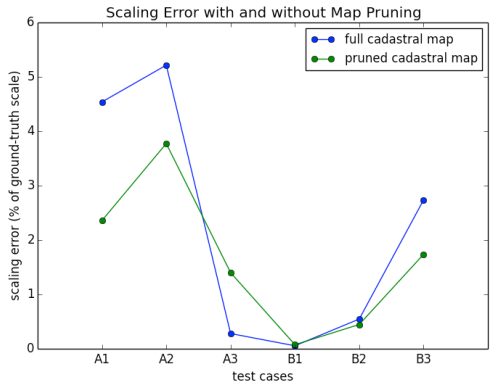
Figure 4.4: Running time of matched filtering (blue) and ICP (green).

5. Failure Cases

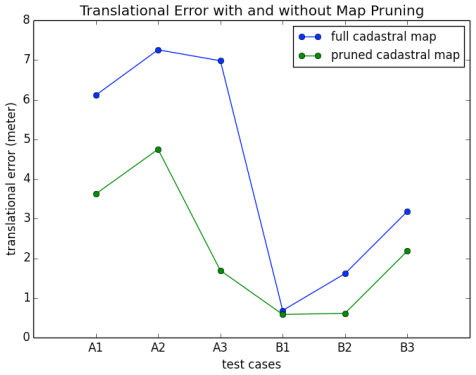
By design our approach worked best for oblique views that captured building facades (exemplified by Fig. 4.2 (a) and Fig. 4.3 (a)). Our method will not work well if the view angle is near nadir or very near horizontal. At near nadir, the view is vertically down. So the 2D top-down projection of the reconstructed point cloud will contain dense points from building roofs and grounds but not as many points from building walls. Therefore the noise-removal technique which we discussed in Section 3.2 and exemplified in Fig 3.1 will fail. When the view angle is near horizontal i.e. camera is at ground level, typically only a few buildings are visible as many will be occluded. As such, unless in contrived situations, there will not be enough geometrical information for accurate match filtering. Fig. 5.1 illustrates both failure cases with examples. We should also point out that our noise-removal method made the assumption that most buildings will have vertical walls. If the scene contains buildings with non-vertical walls such as a pyramid, our noise removal method will also fail. In such a scenario, we will have to rely on other buildings with vertical walls in the vicinity to aid the registration process, otherwise our method will fail.



(a)



(b)

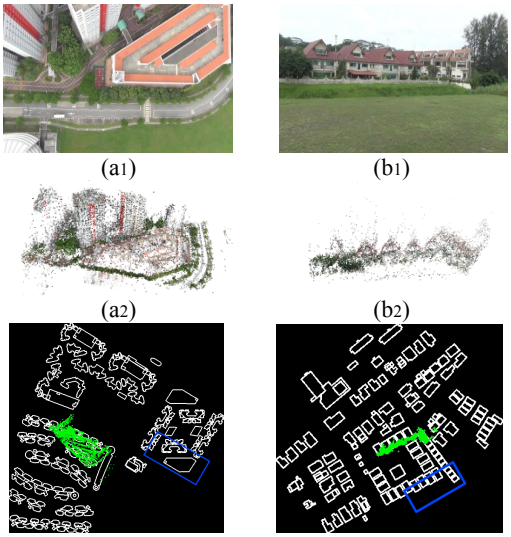


(c)

Figure 4.5: Registration errors using matched filter with and without cadastral map pruning on 6 video sequences. (a) Rotational error measured in degrees. (b) Scaling error measured as percentage of ground truth scale. (c) Translational error measured in meters.

6. Conclusion

We presented a multi-step approach that combined structure from motion and image matching techniques to accomplish the task of registering a video sequence with a cadastral map that was drastically different in view point and modality. It is not our intention to claim that each



(a1) (b1)

(a2) (b2)

(a3) (b3)

Fig.5.1: Matching failures due to extreme view angles. Top row shows 90 degrees view angle video screenshot, middle row shows 3D reconstructed point cloud, bottom row shows the matching result. In (a3) and (b3), white represents the reference cloud, green the final position of query cloud and blue the region where the query cloud should be matched to.

step involved in the workflow is optimum if used as a generic tool. There are many different registration techniques available and some work better than others in different scenarios. We have however tested our system in real life scenarios and found that it worked sufficiently robustly with our data sets. The first step in our workflow used structure from motion technique to transform the input video to 3D point cloud. This was followed by applying the principal component analysis technique to establish the top-down vertical direction. The vertical direction thus computed would be used for the projection of the 3D point cloud onto the horizontal plane. In matching with the reference cloud, an iterative implementation of matched filter was used. We chose a matched filter method instead of an objective cost function minimization method like ICP because of the relatively small coverage of the 3D point cloud and large initial positional errors that made it susceptible to be trapped in local minima. The ability to deal with large initial positional errors is of practical significance as such errors are not uncommon due to long distance viewing and/or errors in synchronization between compass readings and video frames. The test results showed that our coarse-to-fine iterative matched filtering was indeed able to produce good globally optimum registration results with very good tolerance to large initial translational error, and reasonable tolerance to rotational and scaling errors.

References

[1] A. Almansa, A. Desolneux, S. Vamech. Vanishing point detection without any a priori information. IEEE PAMI vol(25), pp.502-507, 2003.

[2] P. J. Besl and N. D. McKay. A method for registration of 3-D shapes, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.14, no.2, pp.239,256, 1992

[3] E.D. Castro and C. Morandi. Registration of translated and rotated images using finite fourier transforms. IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 3, pp700 - 703, 1987

[4] M. Cramer and D. Stallmann. System calibration for direct georeferencing. IAPRS, Volume XXXIV, Com. III, Part A. pp79-84, 2002

[5] M. Ding, K. Lyngbaek, K. Zakhor. Automatic registration of aerial imagery with untextured 3d lidar models. Proceedings of CVPR, 2008.

[6] R. O. Duda and P. E. Hart. Use of the Hough Transformation to Detect Lines and Curves in Pictures, Comm. ACM, Vol. 15, pp. 11–15, 1972

[7] C. Frueh, R. Sammon and A. Zakhor. "Automated texture mapping of 3d city models with oblique aerial imagery. Proceedings of 3DPVT, pp.396-403, 2004.

[8] M. Gerke, and A. Nyaruhuma. Incorporating scene constraints into the triangulation of airborne oblique images. ISPRS XXXVIII 1-4-7/WS, 2009.

[9] G.J. Grenzdorffer, M. Guretzki and I. Friedlander. Photogrammetric image acquisition and image analysis of oblique imagery. The Photogrammetric Record vol(23), pp.372-386, 2008.

[10] M. Habbecke and L. Kobbelt. Automatic registration of oblique aerial images with cadastral maps, Proceedings. ECCV. Conf. on Trends and Topics in Computer Vision, vol. part II, pp253-266, 2010

[11] R. Hartley and A. Zisserman. Multiple View Geometry in Computer Vision. Second Edn. Cambridge University Press. 2003.

[12] D.P. Huttenlocher and S. Ullman. Recognizing solid objects by alignment with an image. International Journal Computer Vision, 5(2), pp.195-212, 1990.

[13] I.T. Jolliffe. Principal Component Analysis, Springer Series in Statistics, 2nd ed., Springer, NY, 2002, XXIX, 487 pp28 illus. ISBN 978-0-387-95442-4

[14] R.S. Kaminsky, N. Snavely, S.M. Seitz and R. Szeliski. Alignment of 3D Point Clouds to Overhead Images. Proceedings of CVPR workshop WIV, pp.63-70, 2009.

[15] T. Labe and W. Forstner. Automatic relative orientation of images. In Proceedings of the 5th Turkish-German Joint Geodetic Days. 2006.

[16] M. Lemmens, C. Lemmens and M. Wubbe. Pictometry: Potentials for land administration. In Proceedings of the 6th FIG reg. conf., International Fed. Of Surveyers, 2007.

[17] D.G. Lowe. The Viewpoint Consistency Constraint. International Journal Computer Vision, 1(1), pp.57-72, 1987.

[18] D. G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints, International Journal of Computer Vision, 60, 2, pp91-110, 2004

[19] K. Ni. Geo-registering 3D point clouds to 2D maps with scan matching and the Hough Transform. Acoustics, Proceedings of ICASSP, pp.1864-1868, 2013.

[20] Y. Sheikh, S. Khan and M. Shah. Feature-based geo-registration of aerial images, GeoSensor Networks, 2004

[21] N. Snavely, S.M. Seitz and R. Szeliski. Modeling the World from Internet Photo Collections. International Journal of Computer Vision (IJCV), 2007.

[22] H. S. Stone. A Fast Direct Fourier-Based Algorithm for Subpixel Registration of Images, IEEE Transactions on Geoscience and Remote Sensing, V. 39, No. 10, pp.2235-2242, 2001

[1] B.V.K. Vijaya Kumar and T.K. Ng. Multiple Circular-Harmonic-Function Correlation Filter Providing Specified Response to In-Plane Rotation. Applied Optics, vol.35, pp.1871-1878, 1996.

[23] P. Viola and W.M. Wells. Alignment by Maximization of Mutual Information. International Journal of Computer Vision, 24(2):137-154, 1997

[24] L. Wang and U. Neumann. A robust approach for automatic registration of aerial images with untextured aerial lidar data. In Proceedings of CVPR, 2009.

[25] Q. Wang and S. You. Automatic Registration of Large-Scale Multi-sensor Datasets, ECCV, vol(6554), pp.225-238, 2012

[26] G. Wolberg and S. Zokai. Robust Image Registration using Log-Polar Transform. Proc. IEEE Intl. Conf. on Image Processing, Vancouver, Canada, 2000

[27] X. Wu, R. Carceroni, H. Fang, S. Zelinka and A. Kirmse. Automatic alignment of large-scale aerial rasters to road-maps. In Proceedings of ACM GIS, 2007.

[28] W. Zhao, D. Nister and S. Hsu. Alignment of continuous video onto 3D point clouds. IEEE Trans. PAMI vol(27) pp.1305-1318, 2005.

[29] Q. Zheng and R. Chellappa. Automatic registration of oblique aerial images, Image Processing. Proceedings. IEEE International Conference on Image Processing, vol. 1, 13-16, pp218-222, 1994