

Navigation with SAR and 3D-Map Aiding

Tomas Toss*, Patrik Dammert* and Zoran Sjanic*,**

*Saab AB

Linköping/Göteborg, Sweden

Email: {tomas.toss, patrik.dammert, zoran.sjanic}@saabgroup.com

Fredrik Gustafsson**

**Division of Automatic Control

Department of Electrical Engineering

Linköping University

Linköping, Sweden

Email: fredrik.gustafsson@liu.se

Abstract—This paper presents a method for matching spotlight Synthetic Aperture Radar (SAR) images with a georeferenced 3D-map as means for navigational aid. A hypothesis of the flying platform’s absolute position, velocity and direction – which later can be used to correct the inertial navigation system – is attained by image matching and optimization.

A projective model with 6 DoF is used to create a simulated SAR image from a 3D map. The parameters of the projective model represents the most important of the platform’s navigation state, and these are adjusted by Chamfer matching the captured SAR image to simulated ones.

The performance is demonstrated on real spotlight SAR images and 3D-map, and the error is shown to be only a few pixels in average, which in our case is about 3 meters.

I. INTRODUCTION

A radar that is carried by a flying platform can be used to obtain an image of the environment by mapping radar reflections to image pixels. The image resolution is governed by the radar antenna size and radar frequency, and realistically, this resolution will be in tens of meters in size. An approach to improve the resolution is to synthesize a larger antenna, which is achieved by moving the platform and taking many radar echoes from the same area. On modern systems, this technique allows for resolution in order of decimeters, and is the basic principle of the Synthetic Aperture Radar (SAR), [1], [2]. Normally, the radar antenna is fixed in the platform’s body. Such a configuration generates striplike images, i.e. strip SAR images, while configurations where the antenna is moved and pointed to a certain point in the scene during the whole image acquisition time, generates spotlight SAR images, see e.g., [3]. In this work spotlight SAR images acquired with the experimental system, [4]–[6], have been used. See Figure 1 for an image example. Usually, these images are used for remote sensing, mapping or surveillance, but in some cases they have been used for navigation purposes, see e.g., [7], [8]. The idea behind this is that the pixels of the SAR image can be related, by a geometric relationship, to the position that the platform had when the image was taken. If the image is, in turn, related or matched to a map of the environment with known coordinates in some reference frame, an absolute position can be obtained in the same reference frame. Such navigation aid could be used to stabilize long term drift present in Inertial Navigation Systems (INS), similar to how Global Navigation Satellite Systems, e.g., GPS, are used. This method has many similarities with terrain aided navigation, [9], where

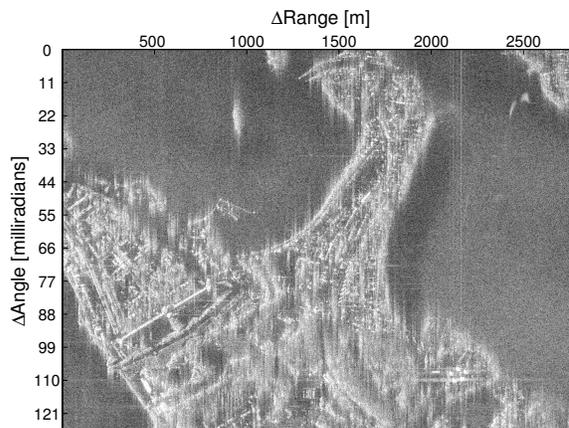


Figure 1: Example of a spotlight SAR image in slant range/cone angle coordinates.

a 3D terrain elevation map is used to support navigation, or visual aided navigation, where optical cameras and maps are used in a similar way, [10], [11]. Optical cameras, however, have the drawback of darkness or bad weather occlusions, while SAR can operate under these conditions, giving an all-weather sensor.

In this work, the image matching idea is utilized to relate a SAR image and a 3D-map of the environment in order to correct the flying platform’s position and orientation. This is done by projecting a 3D-map onto an imaged surface, and in that way creating a simulated SAR image. The simulated image is then matched, or co-registered, with the SAR image. Since the simulated image is created from the 3D-map by using the assumption about platform’s position and orientation, the best match should correspond to the best position and orientation setting. More details about the projection of the 3D-map and its parametrisation are given in Section II.

II. SAR GEOMETRY

As mentioned above, SAR creates high resolution images by measuring the time delay and dopplershift of microwaves back-scattered from the environment. The propagation speed of the transmitted microwaves is normally known (or estimated), so that the measurements can be expressed in terms of

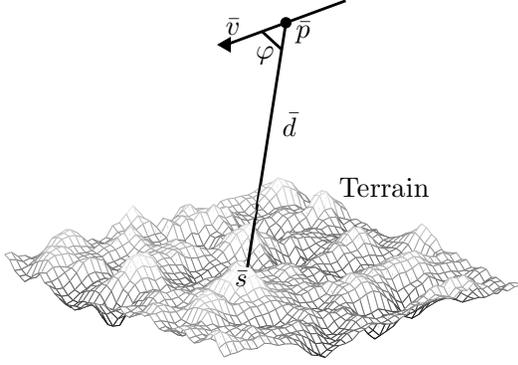


Figure 2: Basic spotlight SAR image geometry.

slant range, R , and cone angle, φ . The obtained measurements are basically a projection of the 3D environment into a 2D image plane. This implies that with a 3D-map of the environment available, a simulated SAR image can be created. In this context, a 3D-map is defined as a terrain elevation for each environment position. It is also assumed that map positions are related to a fixed coordinate system, denoted xyz , in which the position of the platform can be expressed.

To create the simulated SAR image $I_{\theta}^{\text{sim}}(R, \varphi)$, the 3D positions of the 3D-map, $\bar{s} = [s_x, s_y, s_z]^T$, is transformed to a slant range/cone angle representation, using the parametric projective model described below. θ is the parameter vector of the projective model, and consists of the aperture center position, $\bar{p} = [p_x, p_y, p_z]^T$, and the aperture direction encoded as the velocity vector, $\bar{v} = [v_x, v_y, v_z]^T$, i.e., $\theta = [\bar{p}^T \bar{v}^T]^T$. These parameters correspond to the platform's mean position, speed and traveling direction during image acquisition. Note that this parameter vector can be obtained from the INS, but it will normally be wrong due to e.g., drift errors. The parameter vector obtained from INS is called θ_0 . Define further a vector from the center position of the aperture \bar{p} to any 3D-map position \bar{s} as $\bar{d} = \bar{s} - \bar{p}$.

Now, each 3D point can be projected to the {slant range, cone angle}-plane by computing the magnitude of \bar{d} and the angle between the velocity vector \bar{v} and the vector \bar{d} as (subscript G stands for "geometric")

$$R_G(\theta) = \|\bar{d}\| = \sqrt{d_x^2 + d_y^2 + d_z^2} \quad (1a)$$

$$\varphi_G(\theta) = \arccos\left(\frac{\bar{d}^T \bar{v}}{\|\bar{d}\| \|\bar{v}\|}\right) \quad (1b)$$

The intensity values (in range 0 to 1) in these coordinates is simply assumed to be proportional to the incidence angle between local terrain plane in point \bar{s} and the vector \bar{d} as

$$I_{\theta}^{\text{sim}}(R_G, \varphi_G) = \frac{1}{\pi} \arccos\left(\frac{\bar{d}^T \bar{n}}{\|\bar{d}\| \|\bar{n}\|}\right) \quad (2)$$

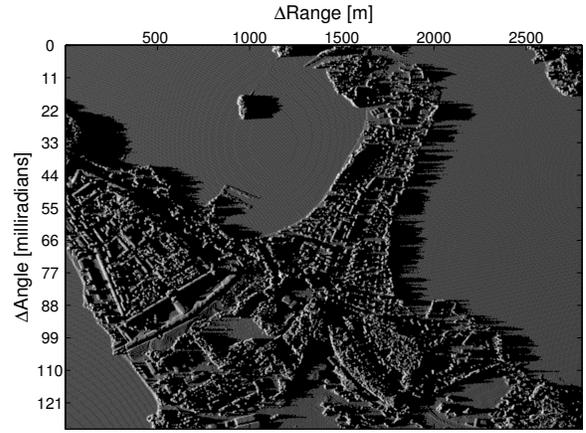


Figure 3: A simulated slant range/cone angle SAR image which has been used for evaluation of the method.

where \bar{n} represents the normal to the local plane defined as

$$\bar{n} = \left[\frac{\partial}{\partial x} \bar{s} \quad \frac{\partial}{\partial y} \bar{s} \quad \frac{\partial}{\partial z} \bar{s} \right]^T \quad (3)$$

Refer to Figure 2 for an illustration of the setup and the parameters defined above.

Important to note is that the image is constructed under the assumption of straight flight trajectory. This assumption is usually valid for these cases, since aperture lengths are quite small. An example image created from a 3D-map with the the projection model (1) is depicted in Figure 3. For comparison, the real slant range/cone angle SAR image is shown in Figure 1 in Section I.

Since SAR measures Doppler shifts and time delay of the transmitted signals, the platform speed affects the image construction. Hence, the pure geometrical projection model in (1) is not sufficient when comparing SAR images to a 3D-map. We have chosen to keep the SAR image and its metadata unspoiled, and instead compensate for the aberration that depends on the platform speed by altering the projection model. The effect on time delay, and thus range measurements, of transmitted signals caused by erroneous platform speed is well below the range resolution of the sensor. Hence, no compensation for range projection equation (1a) is necessary. The effect of erroneous platform speed is incorporated on the cone angle measurement in the model by using the Doppler frequency equation

$$f_d = \frac{2}{\lambda} \left(\frac{\bar{d}^T \bar{v}}{\|\bar{d}\|} \right) \quad (4)$$

where λ denotes the microwave's wave length. Combining (1b) and (4), the cone angle can be expressed as

$$\varphi = \arccos\left(\frac{\lambda f_d}{2 \|\bar{v}\|}\right) \quad (5)$$

This expression is actually used by SAR systems to calculate cone angle image coordinates from the Doppler measurements by using the speed from the navigation system.

By using Taylor expansion on (5), a cone angle formula that compensates for speed errors can be derived as

$$\begin{aligned}
\varphi &\approx \varphi_L + \Delta \|\bar{v}\| \left. \frac{\partial \varphi}{\partial \|\bar{v}\|} \right|_{\|\bar{v}\|=\|\bar{v}_L\|} = \\
&= \varphi_L + \Delta \|\bar{v}\| \frac{\lambda f_d}{2\|\bar{v}_L\|^2 \sqrt{1 - \left(\frac{\lambda f_d}{2\|\bar{v}_L\|}\right)^2}} = \\
&= \varphi_L + \Delta \|\bar{v}\| \frac{\lambda f_d}{2\|\bar{v}_L\|^2 \sqrt{1 - \cos^2(\varphi_L)}} = \\
&= \varphi_L + \Delta \|\bar{v}\| \frac{2\|\bar{v}_L\| \cos(\varphi_L)}{2\|\bar{v}_L\|^2 \sin(\varphi_L)} = \\
&= \varphi_L + \frac{\Delta \|\bar{v}\|}{\|\bar{v}_L\| \tan(\varphi_L)} \quad (6)
\end{aligned}$$

where φ_L and $\|\bar{v}_L\|$ are evaluated at the linearisation point, and $\Delta \|\bar{v}\| = \|\bar{v}\| - \|\bar{v}_L\|$ represents the deviation from it. Now, this relation can be used to calculate the speed error compensated cone angle by setting the linearisation point to the parameter \bar{v} . This gives that φ_L is set to the geometrical cone angle as given before in (1b), and $\Delta \|\bar{v}\|$ is set to the difference between the parameter speed and the INS measured speed, i.e.,

$$\varphi_L = \varphi_G(\theta) = \arccos\left(\frac{\bar{d}^T \bar{v}}{\|\bar{d}\| \|\bar{v}\|}\right) \quad (7a)$$

$$\Delta \|\bar{v}\| = \|\bar{v}\| - \|\bar{v}_0\| \quad (7b)$$

Now the term $\frac{\Delta \|\bar{v}\|}{\|\bar{v}\| \tan(\varphi_G)}$ is used to compensate for aberration that the deviation, $\Delta \|\bar{v}\|$, from the actual speed, which we want to find, and the speed measured by the navigation system causes. This gives the final relation for the simulated image coordinates ($R_{\text{sim}}, \varphi_{\text{sim}}$) as

$$R_{\text{sim}}(\theta) = R_G(\theta) = \|\bar{d}\| = \sqrt{d_x^2 + d_y^2 + d_z^2} \quad (8a)$$

$$\varphi_{\text{sim}}(\theta) = \varphi_G(\theta) + \frac{\Delta \|\bar{v}\|}{\|\bar{v}\| \tan(\varphi_G(\theta))} \quad (8b)$$

which can be used to generate simulated images together with relationship in (2).

III. POSITION AND VELOCITY PARAMETER ESTIMATION

Given the 3D-map introduced in Section II, a simulated SAR image I_θ^{sim} , which is assumed to cover the whole area captured by the real SAR image, can be created for any parameter vector θ using relationship in (8). The captured SAR image can be used as a whole or it can be split into smaller parts, e.g., if the whole image is too large making the calculations computationally heavy. This is expressed as I_k^{SAR} , $k = 1, \dots, N$, where N is the number of (sub)images, e.g., $N = 1$ means that the whole image is used. The real image(s) can now be compared to the simulated SAR image to find the best correspondence between images. Posed as an optimisation problem, a solution $\hat{\theta}$ to the following minimisation problem is sought

$$\hat{\theta} = \arg \min_{\theta} \sum_{k=1}^N V(I_\theta^{\text{sim}}, I_k^{\text{SAR}}) \quad (9)$$

Here $V(\cdot, \cdot)$ is some similarity measure between images that attains its minimum value for the best match. Many similarity measures could be used, e.g., simple image difference or some image feature based comparison. Since the 3D-map that is used has no radar reflectivity information included, all direct comparison methods would most likely fail due to too large differences in pixel values.

Another approach is to use the relative intensity differences, i.e., edges, that arise along the contour of structures. The edges that are formed along the structure, e.g., coast lines and buildings, can be extracted and compared according to [12]–[14].

One quite robust method, which relies on edge extraction, and that has been successfully used for template matching, is Chamfer matching, [8], [14]–[16]. Consequently we use this measure as our image similarity measure V . The Chamfer measure is explained in some more detail in Section III-A.

One thing to note about the used similarity measure is that it is not analytically differentiable. This implies that if any gradient based algorithms are to be used, numerical methods for differentiation must be used, such as finite differences. An alternative is to use gradient-free or grid based search methods. For our experiments we have chosen the gradient-free alternative, in particular Nelder-Mead method [17]. Further, the optimisation procedure is initialised with the parameter values obtained from the platform's navigation system.

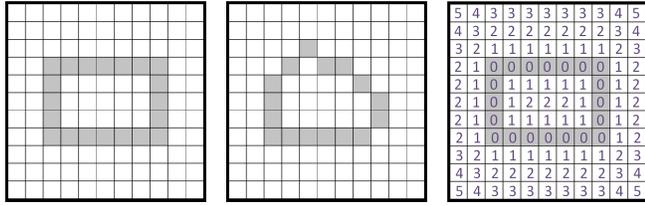
A. Chamfer Matching

Chamfer image matching is a computationally cheap image matching algorithm, which expresses image similarity in terms of distances between edge pixels in binary template, I_F , and target, I_T , images, see Figures 4a and 4b for an example. To calculate the Chamfer distance measure between these two images, $V(I_T, I_F)$, a distance image, D , is first constructed from the target image. The distance image is constructed such that the value of each pixel states the distance to the closest edge pixel. See Figure 4c for an example where city-block distance metric is used. In our experiments, we have used the method presented by [18] that efficiently calculates exact Euclidean distances.

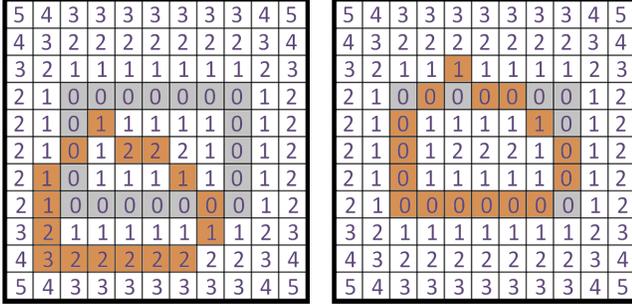
The Chamfer measure is calculated by taking the average of the accumulated distance values from D along the contour of the template image as

$$V(I_T, I_F) = \frac{1}{N} \sum_{e \in I_F} D(e) \quad (10)$$

where e represents the coordinates of the non-zero pixels in the template image and $N = |I_F|$, i.e., the number of non-zero pixels. In Figure 4d and 4e the coordinates e are colored orange and it can be seen that the sum above evaluates to 24 and 2, respectively (the sum of all values that are hit by orange pixels). Since the total number of non-zero pixels in the template is $N = 16$, the Chamfer measures for these examples are $24/16 = 1.5$ and $2/16 = 0.125$. The latter is the minimal distance that can be achieved for this set of target and template images. Since Chamfer measure requires



(a) Binary target image, I_T . (b) Binary template image, I_F . (c) City-block distance transform of the target image, D .



(d) Template image (in orange) superimposed on the distance transformed image. (e) Template image (in orange) superimposed on the distance transformed image giving the best fit.

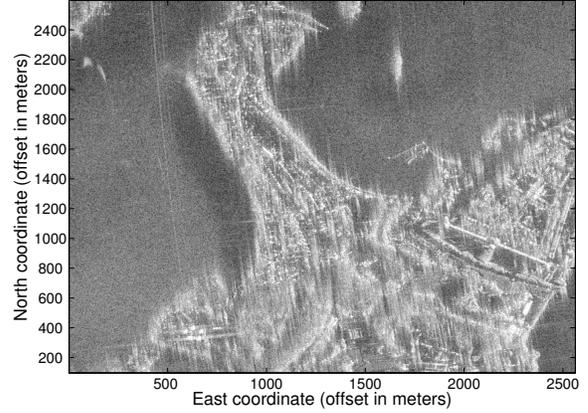
Figure 4: To calculate the city-block Chamfer measure, the binary target image has to be transformed into a distance image, D . The similarity of the target and template images is acquired by taking the average of all distance values along the contour of the template edges. The measures for the bottom left and right images are consequently 1.5 (24/16) and 0.125 (2/16), respectively.

binary images to be used, some kind of edge detector must be applied to the images. One quite robust and well-known variant is the Canny edge detector, [19]. It uses image gradients and thresholding to detect edges in the images. Its main advantage is high robustness to the noise in the images, which is a preferable feature when it comes to the SAR images. The main principle behind this detector is hysteresis with two different thresholds, one high and one low. In this way the problem of broken edges, or streaking, which is almost always present in detectors with only one threshold is avoided. The higher threshold is used to detect edges, while the lower one is used to implement hysteresis and keep an edge even if the gradient response would fall under the higher threshold.

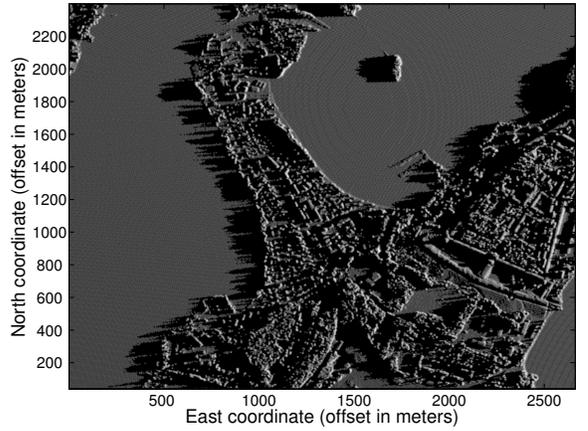
The Canny detector is not free from the general problem of threshold tuning, and individual thresholds for different images must be found on a case by case basis. In this work we are using an existing Canny edge detector implemented in the Image Processing Toolbox in Matlab.

IV. RESULTS

The optimisation approach from Section III is illustrated on a real SAR image in Figure 5a. The corresponding simulated image is shown in Figure 5b where the initial parameter values, i.e., the ones from navigation system, are used. These images are basically the same ones as the images in Figure 1 and



(a) Real SAR image.



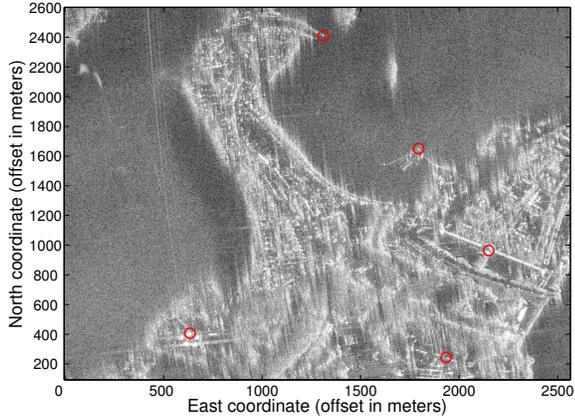
(b) Simulated image.

Figure 5: Real SAR and simulated image used for the experiment. Note that the images are projected on the East-North-Up local plane.

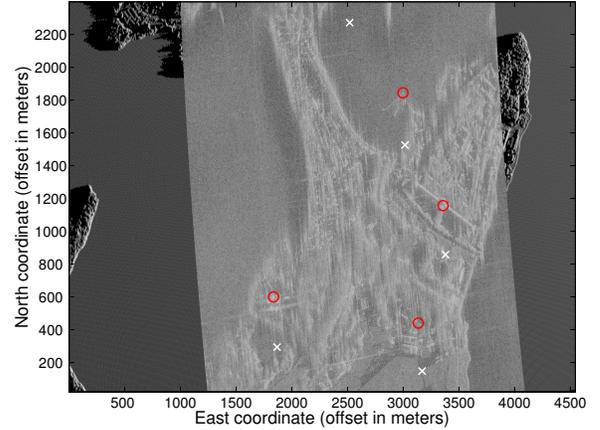
3, but projected from range/cone angle coordinates to a East-North-Up local plane. This is done to easier evaluate matching results in terms of distances on the ground.

Unfortunately, we don't have ground truth for the navigation parameters for these images, but only measured values from the actual flight. For that reason we choose to evaluate the performance of the method by comparing a set of control points manually chosen in the images and evaluate the matching efficiency. We also give the difference between initial and final parameter values as information just to illustrate the order of magnitude for the correction that method provides. The control points for both images are shown in Figure 6.

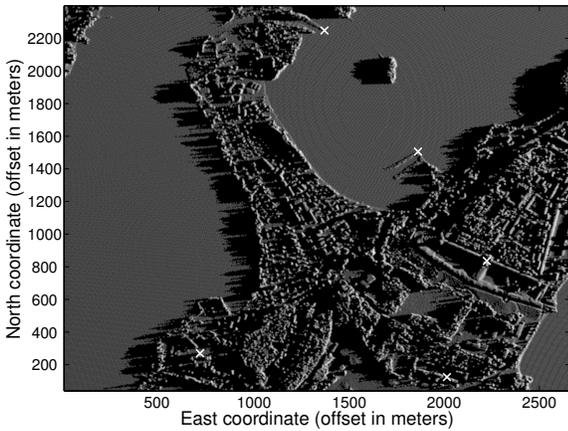
When the images are superimposed with the initial value of θ , it can be seen that the control points are quite far away from each other. This is mostly caused by the error in the velocity vector of the platform which manifests itself as a rotation of the image in the plane, see Figure 7a. As shown by Figure 7b, the distance between the corresponding



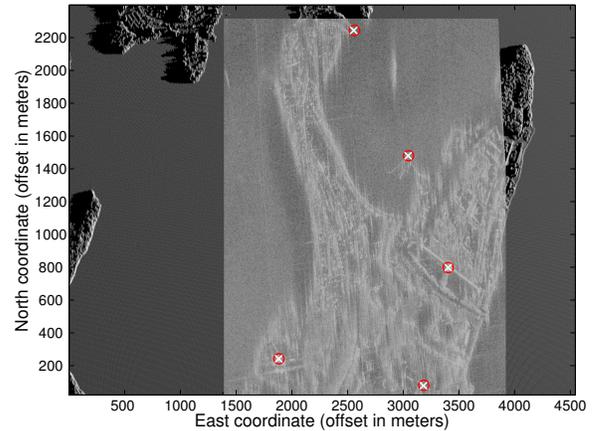
(a) Control points in the real SAR image.



(a) The real SAR image is superimposed on the simulated SAR image, where the simulated image was generated with the initial parameter values.



(b) Control points in the simulated image.



(b) The real SAR image is superimposed on the simulated SAR image, where the simulated image was generated with the optimised parameter values.

Figure 6: Control points in the real (red circles) and simulated (white crosses) image used for the evaluation of the performance.

Point	Initial error [m]	Final Error [m]
1	317	6
2	319	4
3	299	0
4	294	3
5	308	2
Average	307	3

Table I: Estimation results for the control points.

control points is much shorter after the optimization has been performed. The distances (error) between the corresponding control points for the initial and final value of θ , as well as the average error for all the points, are given in Table I. The difference between the final and initial parameter value is $\hat{\theta} - \theta_0 = [6.62, 13.7, -3.75, -0.299, 0.740, -0.0135]^T$. The first three values, which represents the correction of the platform's position in m, is much smaller than the control points' total correction (which is about 300 m). This is due to the fact that image is captured at a large distance, and even

Figure 7: The results of the optimisation approach.

the smallest deviation in the velocity vector (the last three parameter values, expressed in m/s) generates large geocoding errors.

V. CONCLUSIONS AND FUTURE WORK

In this work we present a method for estimating position, velocity and flight direction of a flying platform by matching a spotlight SAR image with a 3D-map of the environment. The basis for the image comparison is the Chamfer matching measure, which is a robust way of matching template images to target images. The performance of the algorithm is evaluated on real SAR images and 3D-map. Great improvement of the correspondence between the images are shown. In this way, an extra all-weather "sensor", SAR, can be used to support navigation, even without presence of the high-precision systems like GPS.

It is however worth noting that it is crucial to have feature rich environment for this method to work, since the matching

performance is highly dependent on this. In this case it is the presence of the well defined (and possibly unique) edges that is important. This implies that in e.g., ocean or desert areas the method will perform quite poorly.

In the future it would be interesting to add some kind of radiometric (radar reflectance) information in the 3D-map in order to get more accurate simulated SAR images and in this way make image matching even more robust. Also, an alternative edge detector, more tailored to SAR image properties, instead of generic one, like Canny, would be interesting to test.

ACKNOWLEDGMENTS

The authors would like to thank Dr. Pelle Carlbom from Saab Vricon for supplying a 3D-map. This work has been supported by Saab AB, the Industry Excellence Center LINK-SIC founded by The Swedish Governmental Agency for Innovation Systems (VINNOVA) and Saab AB and Swedish Defence Material Administration.

REFERENCES

- [1] L. J. Cutrona, W. E. Vivian, E. N. Leith, and G. O. Hall, "A high-resolution radar combat-surveillance system," *IRE Transactions on Military Electronics*, vol. MIL-5, no. 2, pp. 127–131, Apr. 1961.
- [2] C. Oliver and S. Quegan, *Understanding Synthetic Aperture Radar Images*, ser. The SciTech Radar and Defense Series. SciTech, 2004.
- [3] C. V. J. Jakowatz, D. E. Wahl, P. H. Eichel, D. C. Ghiglia, and P. A. Thompson, *Spotlight-Mode Synthetic Aperture Radar: A Signal Processing Approach*. Springer, 1996.
- [4] D. Östling and P. Dammert, "Generation and Geocoding of High Resolution SAR Images with the Ericsson SAR/GMTI Experiment System," in *Synthetic Aperture Radar (EUSAR), 2004 5th European Conference on*, May 2004.
- [5] M. Smedberg and J. Eriksson, "Detection and Positioning of Ground Moving Targets Using a Three-channel SAR/GMTI Experiment System," in *Synthetic Aperture Radar (EUSAR), 2004 5th European Conference on*, May 2004.
- [6] P. Dammert and H. Hellsten, "Focusing ground movers using fast factorized SAR backprojection," in *Synthetic Aperture Radar (EUSAR), 2010 8th European Conference on*, Jun. 2010.
- [7] M. Greco, G. Pinelli, K. Kulpa, P. Samczynski, B. Querry, and S. Querry, "The study on SAR images exploitation for air platform navigation purposes," in *Proceedings of the International Radar Symposium (IRS), 2011*, Sep. 2011, pp. 347–352.
- [8] Z. Sjanic and F. Gustafsson, "Fusion of Information from SAR and Optical Map Images for Aided Navigation," in *Proceedings of 15th International Conference on Information Fusion*, Singapore, Jul. 2012, pp. 1705–1711.
- [9] L. Hostetler and R. Andreas, "Nonlinear Kalman filtering techniques for terrain-aided navigation," *Automatic Control, IEEE Transactions on*, vol. 28, no. 3, pp. 315–323, 1983.
- [10] F. Lindsten, J. Callmer, H. Ohlsson, D. Törnqvist, T. Schön, and F. Gustafsson, "Geo-referencing for UAV navigation using environmental classification," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, May 2010, pp. 1420–1425.
- [11] B. Grelsson, M. Felsberg, and F. Isaksson, "Efficient 7D Aerial Pose Estimation," in *IEEE Workshop on Robot Vision 2013*, Jan. 2013.
- [12] J. D. Wagner, "Automatic Fusion of SAR and Optical Imagery," Sep. 2007, Master's Thesis, Leibniz Universität, Hannover, Germany.
- [13] C. Taylor and D. Kriegman, "Structure and motion from line segments in multiple images," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 17, no. 11, pp. 1021–1032, Nov. 1995.
- [14] J. Ericsson and A. Thid, "Automatic SAR Image to Map Registration," Jan. 2006, Master's Thesis, Chalmers, Göteborg, Sweden.
- [15] H. G. Barrow, J. M. Tenenbaum, R. C. Bolles, and H. C. Wolf, "Parametric correspondence and chamfer matching: two new techniques for image matching," in *Proceedings of the 5th international joint conference on Artificial intelligence - Volume 2*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1977, pp. 659–663.
- [16] G. Borgefors, "Hierarchical Chamfer Matching: A Parametric Edge Matching Algorithm," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 10, pp. 849–865, Nov. 1988.
- [17] J. A. Nelder and R. Mead, "A simplex method for function minimization," *Computer Journal*, vol. 7, pp. 308–313, 1965.
- [18] J. Maurer, C.R., R. Qi, and V. Raghavan, "A linear time algorithm for computing exact euclidean distance transforms of binary images in arbitrary dimensions," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 25, no. 2, pp. 265–270, Feb. 2003.
- [19] J. Canny, "A Computational Approach to Edge Detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. PAMI-8, no. 6, pp. 679–698, Nov. 1986.