Landing spot selection for UAV emergency landing

P.T. Eendebak A.W.M. van Eekeren, R.J.M. den Hollander

TNO, PO Box 96864, 2509 JG The Hague, The Netherlands

ABSTRACT

We present a robust method for landing zone selection using obstacle detection to be used for UAV emergency landings. The method is simple enough to allow real-time implementation on a UAV system. The method is able to detect objects in the presence of camera movement and motion parallax. Using the detected obstacles we select a safe landing zone for the UAV.

The motion and structure detection uses background estimation of stabilized video. The background variation is measured and used to enhance the moving objects if necessary. In the motion and structure map a distance transform is calculated to find a suitable location for landing.

Keywords: UAV landing, human activity detection, safe landing

1. INTRODUCTION

In an ideal situation a UAV will land in a cleared area, supervised or operated by a human operator. However, sometimes a UAV has to land in unplanned zones without an operator. Reasons for this could by malfunctioning hardware (wireless data link between the UAV and operator, problems with the engines), weather conditions etc.

Many UAVs have a fallback mode that allows them to land in a controlled way if no operator or pre-programmed flight plan is available. A typical scenario for a small helicopter would be to hover for several minutes at a fixed position and then slowly descend until the ground is reached. The hovering and slow descend will give people nearby the time to make clear the area below the UAV. When operated in populated areas, such a controlled emergency landing can still impose danger for humans.

In this paper we present a method to detect 3D structures and human activity below a UAV using computer vision. We assume the UAV has a camera (either RGB, grayscale or infrared) that can be directed towards the ground. If structure or human activity is detected, then it can be decided to perform the emergency landing at a different location.

The method is based on background estimation for moving cameras. Using this background estimation both structure and moving objects are detected, which are used to predict whether the landing site is suitable or not. The algorithms selected are suitable for real-time implementation on an embedded platform.

When considering UAV emergency landings it is important to have a clear vision of the boundary conditions. For example, what are the platform characteristics (rotary, fixed-wing), which sensors are available (gps, INS, etc.), remaining flight time, availability of maps with alternative landing sites. Below we summarize the scenario we have in mind for our method. For variations on this scenario our method can still be suitable (sometimes with modifications), but we do not discuss all the different possibilities in detail.

- We have a UAV that is capable of hovering at a fixed position (most rotary wing UAV's satisfy this condition)
- The UAV is able to fly. In particular a controlled descend is still possible.
- No communication between a human operator and the UAV is possible.
- The system has a downward facing camera (either visual or infrared).

Corresponding author: P.T. Eendebak: E-mail: pieter.eendebak@tno.nl, Phone: +31 88 866 8129

Unmanned Systems Technology XV, edited by Robert E. Karlsen, Douglas W. Gage,
Charles M. Shoemaker, Grant R. Gerhart, Proc. of SPIE Vol. 8741, 874105
© 2013 SPIE · CCC code: 0277-786X/13/$18 · doi: 10.1117/12.2017910

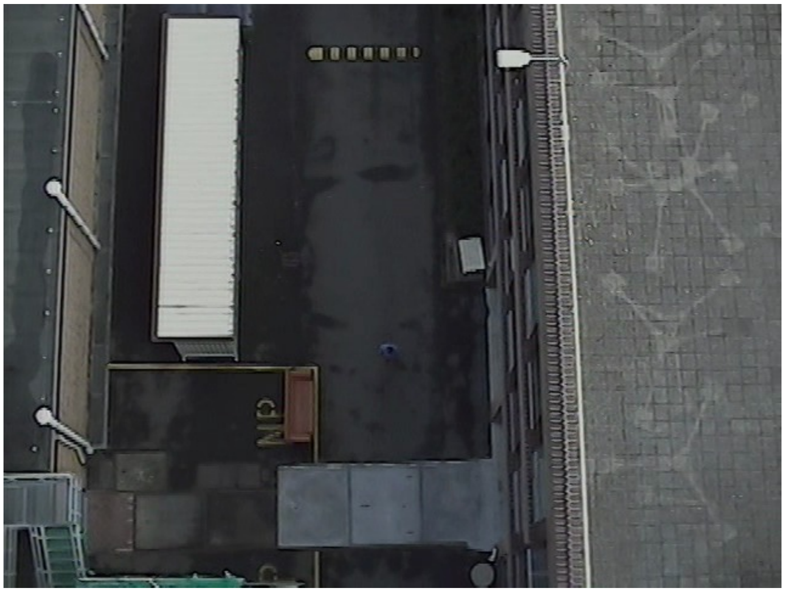

Figure 1. Typical image from a UAV. The height is 30 meters. Just below the centre of the image is a person walking.

2. LITERATURE REVIEW

2.1 Image registration

The registration of images has been an active research topic for many years and a variety of methods have been developed.[1] The methods differ mainly in the type of visual features used (keypoints, lines, patches) and the motion model (e.g. Euclidean, projective or non-rigid).

The visual features can be coarsely classified as belonging to either image patches or keypoints. The first type, image patches, can be used for finding similar image regions based on raw pixel values, where the search for the best matching region is often based on gradient-search,[2] a variant of block matching or a measure of mutual information. The local transformation that describes corresponding image patches in images is usually assumed to be a translation, although more complex transformation can also be used.

The second type of feature is a keypoint which is a position in the image that shows a specific recognizable structure. Examples of such keypoints are (intensity) corner points[3] or local scale-space intensity extrema.[4] At the position of the interest points, a local descriptor of the image neighborhood is computed that is largely unaffected by illumination or viewpoint changes. A frequently used descriptor is for instance the SIFT (Scale Invariant Features Transform) descriptor.[5] The comparison of the local descriptors results in a set of corresponding keypoints that describe the motion between the frames.

The result after the comparison of visual features, either calculated from image patches or keypoints, is a collection of motion vectors between subsequent frames. Depending on the motion of the camera w.r.t. some static scene we can identify a global transformation of all the vectors. In particular, when the camera movement does not include any translation but only rotation about its center, a good approximation for the global transformation is a projective transformation. Another possibility for the approximation by a projective transformation to be valid, is that all scene points lie on a plane. In practice this will hold approximately for distant scenes. In a static world with a non-static UAV the projective transformation, or its approximation by an affine transformation, will be adequate for describing the frame-to-frame motion.

The merit of calculating a global transformation for the motion is that it can be used to correct noisy and also erroneous motion vectors, the so-called outliers. These are caused by areas showing little image structure so that the underlying movement can not be determined. Also when the assumed motion model for the segment is not correct, such outlying motion vectors may appear. Robust estimation methods, like e.g.,[6] are therefore applied for the estimation of the global transformation.

2.2 Background estimation for moving cameras

For detection of moving objects a key component is background subtraction or background estimation. The key idea here is that in a video the main component is a static background with sparse moving objects. Once a background image is estimated, the moving objects can be determined by subtraction of the background image from the current frame. In the difference image a pixel is classified as foreground if the pixel value is above a certain threshold. For a static camera background estimation can be performed by taking the mean of the last k frames. Many variations exist such as taking a running average, mixture of Gaussians, kernel density estimators, mean-shift estimators, etc.[7]

For moving cameras a background estimate can be created by calculating the movement of the image, and transforming the image to a stabilized frame for the background image. Since the camera is moving, an important issue is to make a distinction between the changes in the scene induced by the movement of the camera and the changes induced by the moving objects. The type of movement of the camera (e.g. free moving, shaking, vibrating) plays an important role.

With optic flow[8, 9] the full motion induced by camera motion can be captured. The goodness of fit of the optic flow combined the estimated motion with a change in the pixel color values. If the moving objects have different color vales than the background these objects will be visible as pixels with lower score in the optic flow map.

2.3 Moving object detection

If the camera has a fixed position (shaking camera, pan-tilt camera, slowly moving camera) then a reasonable approach is to try to stabilize the camera image and then perform background estimation on the stabilized video. This can also be done for freely moving cameras with additional assumptions such as a flat scene with no 3D structure. The transformation stabilizing the frames is usually a projective transformation or a restricted projective transformation (e.g. affine or similarity). For freely moving cameras with a 3D scene the changes induced by movement parallax cannot be stabilized using a projective transformation.

For any video sequence we can use keypoint tracking (for example Kanade-Lucas-Tomasi tracking[9, 10]) to build keypoint trajectories. For a static camera the keypoint trajectories of background points consist of a single point. The keypoints corresponding to moving object consist of trajectories of arbitrary shape. For a moving camera (with some additional assumptions) the trajectories of background points are restricted to a 3-dimensional subspace. Using this knowledge we can try to estimate this 3-dimensional subspace and classify keypoints into foreground and background points according to their projection onto the 3-dimensional background trajectory subspace. This approach is used in Sheikh.[11] In Cui et al.[12] the low rank approach is combined with group sparsity of the motion. The keypoint method works well if there are enough keypoints in both the background and foreground. In our application the foreground objects are too small to detect a reasonable number of foreground trajectories. For example in the video from figure 1 there are typically zero or one keypoints detected on the person walking.

2.4 UAV emergency landing

Finding a proper landing spot for a UAV in an emergency situation is very important to prevent collision with humans or non-flat structures such as buildings. In the literature a lot of research can be found on this topic. In[13] an area is scanned with a Cessna and safe landing regions are detected using surface classification. Also in Bosch et al.[14] detection of 3D structures is used for finding safe landing spots. This is similar to our work, but we also consider moving objects in our analysis. In indoor scenarios special landing targets are used.[15] For outdoor regions this approach is less suitable and the UAVs can cover larger areas.

3. METHOD

The method for detection of safe landing spots consists of 4 steps: camera motion estimation, background estimation, moving object detection and safe landing spot selection. In figure 2 there is an overview of the method. In the next subsections we will describe the individual components.

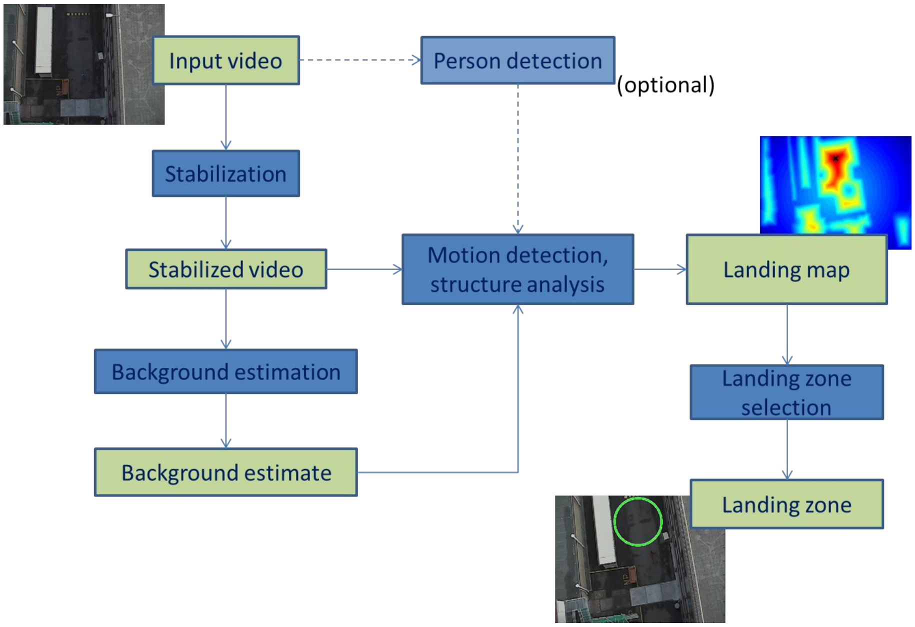

Figure 2. Overview of the method

3.1 Camera motion estimation

Stabilization of the color video sequence was performed using the following steps:

- **Initialization** of tracking keypoints using Harris corner detection. We select the 150 keypoints with highest corner score for tracking. To obtain an even distribution of keypoints over the image we set the minimal distance between two keypoints to 40 pixels.
- **Tracking** of keypoints using the Kanade-Lucas-Tomasi tracker from OpenCV.[16] If the number of good tracks drops below a specified threshold (we use 100 tracks in our experiments) or if a fixed number of frames has been processed (we use 20 frames), then the tracks are re-initialized using the method above. A typical frame with tracked keypoints is shown in figure 3.
- **Motion estimation** of a *projective transformation* using RANSAC and a DLT estimator.[17]

For infrared video sequences the keypoint calculation does not work well due to a lack of texture in the scene. We therefore apply a gradient-based motion estimation method. However, before a gradient-based search can be succesful we have to correct for the non-uniformities in the image frames. The non-uniformity or fixed pattern noise is estimated from the sequences and subsequently used to correct the individual frames using the techniques from Schutte et al.[18] The gradient-based search method that is applied is the multi-level Lucas Kanade optical flow in OpenCV.[16]

3.2 Background estimation

In order to allow for real-time processing on a UAV platform we choose a relatively simple model for the background estimates, we take a running Gaussian average.[19] We choose not to use the mixture of Gaussian model[20] that has been used recently. The reason is that the model is more complex and hence requires more processing power. Also the complex model requires a longer time to stabilize.

The background is modelled by a single background image B_t and a background variance Σ_t. At each timestep the background is updated as

$$\begin{aligned} B_t &= (1-\alpha)w_t(B_{t-1}) + \alpha I_t \\ \Sigma_t^2 &= (1-\alpha)w_t(\Sigma_{t-1}^2) + \alpha(B_t - I_t)^2 \end{aligned} \tag{1}$$

Here $w_t(X)$ is the image X warped from timestep $t-1$ to timestep t using the stabilization. The parameter α determines the update time of the model. In our experiments we have used $\alpha = 0.1$, which means the update time of the background is of the order of 10 frames. The background image B_0 is initialized to the first image frame, the first background variance Σ_0^2 is initialized to a uniform high variance.

Because the images are warped, there are pixels at the border of the images that are undefined. For the background image we set the undefined pixels to the current frame (this is the best estimate we have for the background). For the background variance we set the undefined pixels to the maximal variance. In principle we could track the video stabilization over a area larger then the current frame and create more sophisticated methods for updating the undefined pixels. In our application the camera movement is small so there are relatively few undefined pixels. The update method described above is simple and works well for the application.

3.3 Moving object and structure detection

The difference between the current frame and the background estimate $D_t = I_t - w_t(B_{t-1})$ is the basis for the moving object and structure detection. Here the background frame B_{t-1} is warped using the transformation described in section 3.1. The points with high intensity correspond to changes. These changes are introduced for mainly 2 reasons:

- When moving objects are present, the current frame will not correspond to the background image and these differences will appear as blobs in the difference image.
- Since we are using projective transformations to stabilize the images, there can be differences due to parallax effects. These parallax effects occur when there are 3D structures in the view of the camera.In the difference image these structures will appear as long thin structures.

Some points have a high variance all the time, we reduce the contribution of these points using the estimated background variance. For reduction we use the following formula:

$$X_t = D_t s(\Sigma_t). \tag{2}$$

Here the multiplication is pixelwise and s can be any function that takes value 1 if the variance is 0 and is monotonous decreasing. To the input background variance Σ_t and the output X_t smoothing is applied using a 3×3 box kernel. The precise format of the function s makes a trade-off between the importance of the moving objects and the 3D structures. The 3D structures will be reduced because they tend to have a high background variance.

3.4 Safe landing spot selection

From the processed data a good landing spot should be selected. When the UAV is capable of precise steering this can be a specific region below the UAV. If steering is not possible, the system should either accept or reject the current landing position. The method described in this subsection estimates the best possible landing spot in the field of view of the UAV. This means that moving objects and 3D structures (fences, trees, walls of buildings) should be avoided. The safe landing spot selection consists of the following steps for each frame:

- With the background estimation and the (optional) person detection a difference image is constructed which indicates the moving objects and 3D structures (regions with parallax).
- From this difference image a segmentation is calculated using a minimal graph cut method.[21] This segmentation results in a binary image indicating obstacles.
- A distance map to all detected obstacles is calculated from this segmented image and averaged over time taking the last 10 frames into account. This distance map is combined with the distance to the center of the image, giving priority to locations in the center.
- In the resulting distance map the location with the maximum distance is calculated. If this maximum distance is above a certain threshold, then this location is selected as the landing spot. The location with highest distance value corresponds to the largest circular area which is free of obstacles.

4. RESULTS

4.1 Dataset

We have recorded a dataset that is representative for a small UAV landing in a controlled fashion. The data was recorded from the tower at the TNO building at the Waalsdorpervlakte. The data recorded was done using the same cameras as is used on the Delft Dynamics RH4 Spyder.[22]

From a height of 30 meters several videos have been recorded. Both RGB and infrared video have been recorded. The cameras were hand-held during recording* to simulate the typical movements of cameras during UAV flight.

4.2 Camera motion estimation

The tracking of keypoints works well for image stabilization. There are enough stable keypoints in the background to calculate a homography. Using the keypoints for estimation of trajectories as described as described in section 2 does not work very well. There are hardly any keypoints on the foreground objects (in the scene in figure 1 there is only one foreground keypoint). Also, the keypoints are not evenly distributed over the input image. The stabilization of the video works well for frame-to-frame stabilization. For longer stabilization there appears to be a small amount of drift in the stabilization.

The infrared stabilization was more challenging due to the limited amount of texture in the images. The gradient based method was successful in stabilizing the video. For large jumps in the scene (either due to strong camera movements or problems with the video recording) the stabilization does not work properly, but after a limited amount of time the system recovers because the background image is updated.

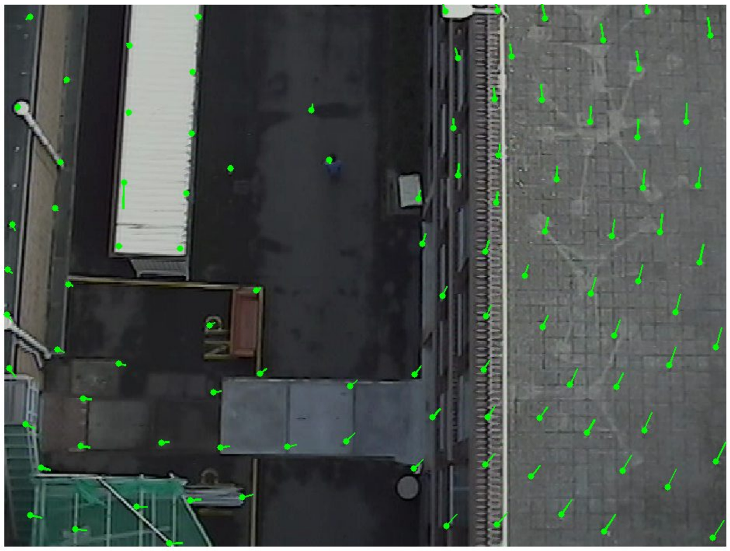

Figure 3. Tracking of keypoints using KLT

4.3 Moving object and structure detection

Moving object detection was successfully performed on video sequences. The main results are shown in figure 4.3. Shown in the figure are (for a single moment in time) the original video, the estimated background, the difference between background and current frame and the filtered motion image. In the difference image (bottom left) both the person moving and the building edges are present. By filtering this difference image using the background variation estimate, the moving person appears clearly.

*RGB video was captured with a VTQ-SPC500 Mini Colour Camera, 1/3" Super HAD CCD, 550 tv lines. The infrared video was captured with a Flir Quark 640, FOV 32 deg, 640 x 512 pixels. Both videos were recorded from a height of 30 meters.

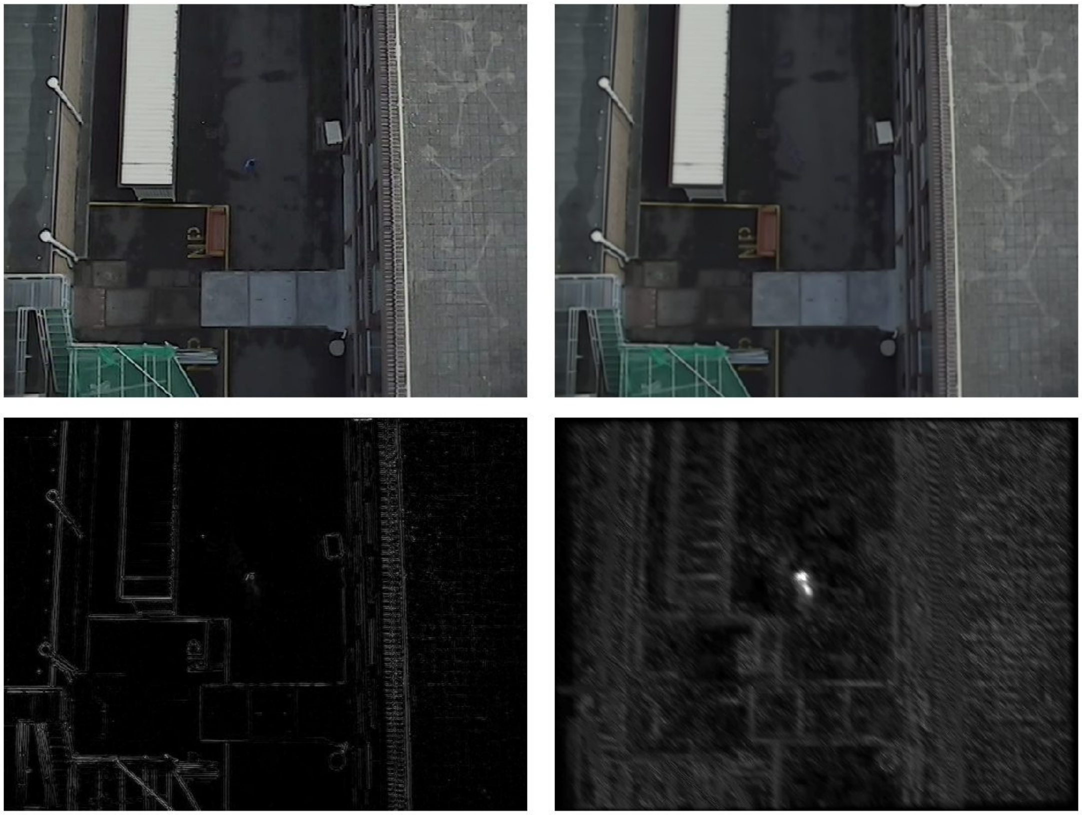

Figure 4. Top: from left to right: input image, background estimate. Bottom: from left to right: difference between current frame and background estimate, filtered result. For the filtered result the moving objects are enhanced by suppressing the pixels with high background variation. Video 1: http://dx.doi.org/10.1117/12.2017910.1

4.4 Landing spot selection

In figure 5 and figure 6 some results are depicted of the safe landing spot selection on different kind of imagery (visual and infrared). A green circle overlayed on a frame indicates the best landing spot for that frame. Next to the camera frames the corresponding distance map is shown with the crossmark indicating the maximum distance to the detected obstacles. Note that the results of the safe landing spot selection can best be appreciated by watching a video.

5. CONCLUSIONS

We have implemented a method for selection of safe landing zones from a UAV. The requirements for the method are quite modest: only a downward facing camera and some processing power are required. The next step would be to test the algorithm with a lightweight UAV that is safe for humans to walk under at small heights.

There are some additional methods that can be of interest when determining the safety of landing sites. One of them is static person detection. This allows for detection of persons standing still. Integration of this information into the current model is straightforward. Developing robust real-time static person detection in RGB video from a UAV (top view) is a challenge that is currently under research.

ACKNOWLEDGMENTS

The authors would like to thank Delft Dynamics for capturing the imagery used for this research.

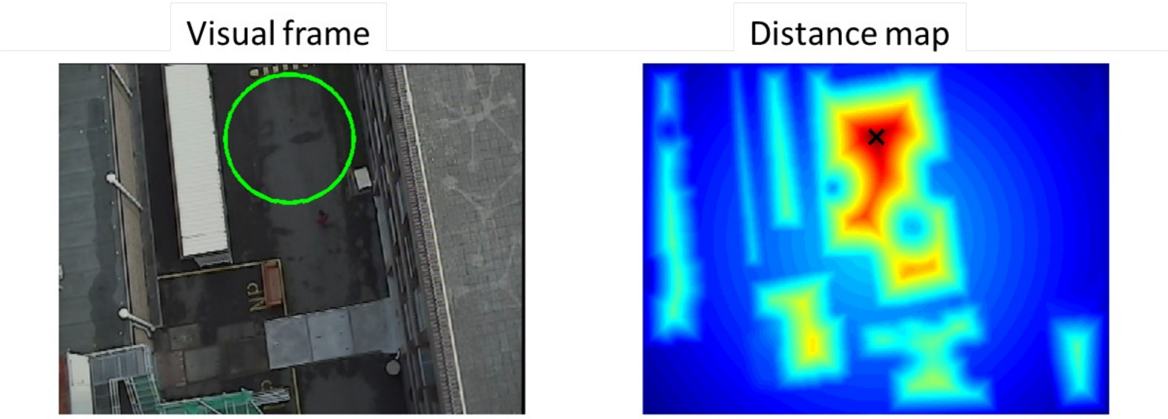

Figure 5. Left: Visual frame overlayed with the selected safe landing spot (green circle). Right: Corresponding distance map. Note that the person detection is visible in the distance map.
Video 2: http://dx.doi.org/10.1117/12.2017910.2

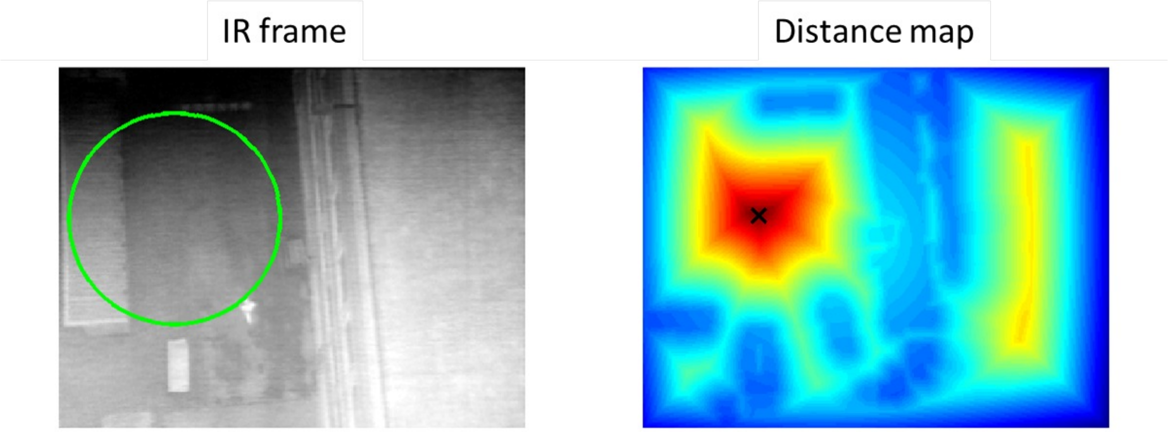

Figure 6. Left: IR frame overlayed with the selected landing spot (green circle). Right: Corresponding distance map.

REFERENCES

[1] Zitov, B. and Flusser, J., "Image registration methods: a survey," *Image and Vision Computing* **21**, 977–1000 (2003).

[2] Lucas, B. D. and Kanade, T., "An iterative image registration technique with an application to stereo vision," 674–679 (1981).

[3] Harris, C. and Stephens, M., "A combined corner and edge detector," in [*In Proc. of Fourth Alvey Vision Conference*], 147–151 (1988).

[4] Lowe, D. G., "Object recognition from local scale-invariant features," in [*Proc. of the International Conference on Computer Vision*], (1999).

[5] Lowe, D. G., "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision* **60**, 91110 (Nov. 2004).

[6] Fischler, M. A. and Bolles, R. C., "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM* **24**, 381–395 (June 1981).

[7] Piccardi, M., "Background subtraction techniques: a review," in [*IEEE International Conference on Systems, Man and Cybernetics*], *IEEE International Conference on Systems, Man and Cybernetics* **4**, 3099–3104, IEEE (Oct. 2004).

[8] Horn, B. K. P. and Schunck, B. G., "Determining optical flow," *ARTIFICAL INLIGENCE* **17**, 185203 (1981).

[9] Lucas, B. D. and Kanade, T., "An iterative image registration technique with an application to stereo vision (ijcai)," in [*Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI '81)*], 674–679 (April 1981).

[10] Tomasi, C. and Kanade, T., "Detection and tracking of point features," tech. rep., International Journal of Computer Vision (1991).

[11] Sheikh, Y., Javed, O., and Kanade, T., "Background subtraction for freely moving cameras.," in [*ICCV*], 1219–1225, IEEE (2009).

[12] Cui, X., Huang, J., Zhang, S., and Metaxas, D. N., "Background subtraction using low rank and group sparsity constraints," in [*ECCV (1)*], 612–625 (2012).

[13] FitzGerald, D., Walker, R., and Campbell, D., "A vision based forced landing site selection system for an autonomous UAV," in [*Proceedings of the 2005 International Conference on Intelligent Sensors*], 397 – 402 (dec. 2005).

[14] Bosch, S., Lacroix, S., and Caballero, F., "Autonomous detection of safe landing areas for an UAV from monocular images," in [*IEEE/RSJ International Conference on Intelligent Robots and Systems*], 5522 –5527 (oct. 2006).

[15] Lange, S., Sunderhauf, N., and Protzel, P., "A vision based onboard approach for landing and position control of an autonomous multirotor UAV in GPS-denied environments," in [*International Conference on Advanced Robotics*], 1 –6 (june 2009).

[16] Bradski, G., "The OpenCV Library," *Dr. Dobb's Journal of Software Tools* **25** (2000).

[17] Hartley, R. I. and Zisserman, A., [*Multiple View Geometry in Computer Vision*], Cambridge University Press, ISBN: 0521540518, second ed. (2004).

[18] Schutte, K., de Lange, D.-J. J., and van den Broek, S. P., "Signal conditioning algorithms for enhanced tactical sensor imagery," *Proc. SPIE 5076, Infrared Imaging Systems: Design, Analysis, Modeling, and Testing XIV* , 92–100 (2003).

[19] Wren, C., Azarbayejani, A., Darrell, T., and Pentland, A., "Pfinder: Real-time tracking of the human body," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **19**, 780–785 (1997).

[20] Stauffer, C. and Grimson, W. E. L., "Adaptive background mixture models for real-time tracking," in [*Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*], *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on* **2**, 246–252 Vol. 2, IEEE, Los Alamitos, CA, USA (Aug. 1999).

[21] Boykov, Y. and Kolmogorov, V., "An experimental comparison of min-cut/max- flow algorithms for energy minimization in vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **26**, 1124 –1137 (sept. 2004).

[22] "Delft dynamics," (2012). `http://www.delftdynamics.nl`.