

Using iFMI Spectral Registration for Video Stabilization and Motion Detection by an Unmanned Aerial Vehicle (UAV)

Sören Schwertfeger, Andreas Birk and Heiko Bülow

Abstract—Unmanned Aerial Vehicles (UAV), especially in the form of Micro Aerial Vehicles (MAV) are useful tools for reconnaissance, surveillance, and general situation assessment in safety, security, and rescue missions. Many UAV have meanwhile good autonomous flight capabilities, especially by tracking pre-planned routes via GPS or for station-keeping. Here it is shown how the video stream from an UAV can be analyzed to automatically detect motion in the scene while the vehicle is moving itself. Concretely, it is shown how a spectral image registration method, the improved Fourier Mellin Invariant method (iFMI), can be used for video stabilization and motion segmentation. The method is first analyzed with scenes containing artificial markers for ground truth evaluation. Furthermore, results from aerial video data from a quadcopter platform are presented.

I. INTRODUCTION

Unmanned Aerial Vehicles (UAV) especially in the form of Micro Aerial Vehicles (MAV) are used in a wide range of SSRR applications including search and rescue, reconnaissance, and surveillance [9], [2], [24], [17], [30], [12], [1], [21]. UAV are meanwhile quite capable of autonomous flight behaviors, especially in form of tracking pre-planned routes via GPS or to do station-keeping. What is less developed is the capability to react to events. In [3], we presented among others results where a UAV (Fig. 1) has significant amounts of autonomous flight time during a surveillance mission (Fig. 2). Though the UAV is flying autonomously, an operator nevertheless has to constantly check the video stream from the vehicle for motion in the scene as the mission task is to detect intruders. Also in search and rescue missions, it is of interest to detect motion in the scene, e.g., to detect hand waving of a trapped victim when a UAV is autonomously surveying a rubble pile. The challenge is that the UAV itself is always moving - even during station-keeping it is never perfectly stable - and hence this ego-motion has to be compensated.

Here, we show how our contributions to mosaicking, also known as photo-mapping, with a UAV [3] can be used for video stabilization and motion segmentation, i.e., for the detection of moving objects from a moving platform. Given a perfect localization of a vehicle and the parameters of a camera that takes pictures of the scene, the generation of video stabilization is trivial. But UAV localization is very coarse and error prone, especially if simple sensors like GPS are used, respectively very costly in terms of necessary payload for computation power, e.g., if a high end inertial

Dept. of Computer Science, Jacobs University Bremen,
28759 Bremen, Germany [s.schwertfeger, a.birk,
h.buelow]@jacobs-university.de



Fig. 1. An Airrobot AR100 quadcopter.



Fig. 2. An example of a photo map of a camp generated during a surveillance mission from 784 video frames in real time.

navigation system would be used. An alternative is to try to use the video stream which is provided by the vehicle anyway.

The challenge is hence to determine the vehicle's motion from the image sequence to compensate for the motion. Feature extraction methods like the Scale Invariant Feature Transform (SIFT) [19], [18] are a popular basis for this. SIFT delivers point-wise correspondences between distinctive, non-repetitive local features in the two images, which can then be used to estimate the motion. The number of detected features is significantly smaller than the number of pixels in the image. Other methods for identifying features include local image descriptors like intensity patterns [20], [13] and the Kanade-Lucas-Tomasi Feature Tracker (KLT) [28]. Though there is quite some successful work with

visual features methods applied on video streams from UAV, especially in form of visual Simultaneous Localization and Mapping (vSLAM) [29], [30], they are inherently limited in rather featureless scenes, e.g., when there are larger monotone surface patches like lawn, concrete, or asphalt. Instead of using features, our mosaicking method [3] is based on a spectral registration method. It is based on the hypothesis that the whole information in the images and not only features should be used to minimize uncertainties and ambiguities in registration, especially in the unstructured environments that are typically found in SSRR missions. The mosaicking is hence based on a variant of the Fourier Mellin Invariant (FMI) transform for image representation and processing [8], [7], [25]. Our version introduced two modifications to achieve a very fast and robust method as presented in more detail in [4]: First, a logarithmic representation of the spectral magnitude of the FMI descriptor is used. Second, a filter on the frequency where the shift is supposed to appear is applied. In addition to its robustness, especially in featureless scenarios, the algorithm has the significant advantage over feature based approaches that it has a fixed computation time per registration, which is in addition suited for real-time registrations on rather low power computation platforms.

In this paper, it is shown that the iFMI registration can be used to generate a kind of video-stabilization also known as image- or camera-stabilization, or (ego-)motion-compensation [26], [22], [31], [14], [10]. This means that the photo-mapping provides the video frames in a fixed coordinate frame of the static environment despite the vehicle's ego-motion. It is well known that video-stabilization significantly eases vision processes on moving vehicles, see for example [16], [11]. Especially, it can be used as basis for object detection by motion extraction through differential images [15].

The rest of this paper is structured as follows. In section II, the improved Fourier Mellin Invariant (iFMI) descriptor for image registration is introduced. Section III describes the application of the image registration for video stabilization which itself is used for motion segmentation. Two experiments are shown in section IV while section V concludes this paper.

II. IMPROVED FOURIER MELLIN MAPPING

The classical Matched Filter (MF) of two 2D signals $r(x, y)$ and $s(x, y)$ is defined by:

$$q(x, y) = \int \int_{-\infty}^{\infty} s(a, b) r(x - a, y - b) da db \quad (1)$$

This function has a maximum at (x_0, y_0) that determines the parameters of a translation. One limitation of the MF is that the output of the filter primarily depends on the energy of the image rather than on its spatial structures. Furthermore, depending on the image structures, the resulting correlation peak can be relatively broad. This problem can be solved by using a Phase-Only Matched Filter (POMF). This correlation approach makes use of the fact that two shifted signals having the same spectrum magnitude are carrying the shift information within its phase (eq. 2).

A. Principles of the Fourier Mellin transform

The principle of phase matching is now extended to additionally determine affine parameters like rotation, scaling and afterward translation.

$$f(t - a) \circ \bullet F(\omega) e^{i\omega a} \quad (2)$$

When both signals are periodically shifted the resulting inverse Fourier transformation of the phase difference of both spectra is actually an ideal Dirac pulse. This Dirac pulse indicates the underlying shift of both signals which have to be registered.

$$\delta(t - a) \circ \bullet 1 e^{i\omega a} \quad (3)$$

The resulting shifted Dirac pulse deteriorates with changing signal content of both signals. As long as the inverse transformation yields a clear detectable maximum this method can be used for matching two signals. This relation of the two signals phases is used for calculating the Fourier Mellin Invariant Descriptor (FMI). The next step for calculating the desired rotation parameter exploits the fact that the 2D spectrum (eq. 5) rotates exactly the same way as the signal in the time domain itself (eq. 4):

$$s(x, y) = r[(x \cos(\alpha) + y \sin(\alpha)), (-x \sin(\alpha) + y \cos(\alpha))] \quad (4)$$

$$|S(u, v)| = |R[(u \cos(\alpha) + v \sin(\alpha)), (-u \sin(\alpha) + v \cos(\alpha))]| \quad (5)$$

where α is the corresponding rotation angle.

For turning this rotation into a signal shift the magnitude of the signals spectrum is simply re-sampled into polar coordinates. This can be done in a different resolution (the *polar-log-resolution*) than the original one (*image-resolution*). For turning a signal scaling into a signal shift several steps are necessary. The following Fourier theorem

$$f\left(\frac{t}{a}\right) \circ \bullet a F(a\omega) \quad (6)$$

shows the relations between a signal scaling and its spectrum. This relation can be utilized in combination with another transform called Mellin transform which is generally used for calculations of moments:

$$V^M(f) = \int_0^\infty v(z) z^{i2\pi f - 1} dz \quad (7)$$

Having two functions $v_1(z)$ and $v_2(z) = v_1(az)$ differing only by a dilation the resulting Mellin transform with substitution $az = \tau$ is:

$$\begin{aligned} V_2^M(f) &= \int_0^\infty v_1(az) z^{i2\pi f - 1} dz \\ &= \int_0^\infty v_1(\tau) \left(\frac{\tau}{a}\right)^{i2\pi f - 1} \frac{1}{a} d\tau \\ &= a^{-i2\pi f} V_1^M(f) \end{aligned} \quad (8)$$

The factor $a^{-i2\pi f} = e^{-i2\pi f \ln(a)}$ is complex which means that with the following substitutions

$$\begin{aligned} z &= e^{-t}, \ln(z) = -t, dz = -e^{-t} dt, \\ z \rightarrow 0 &\Rightarrow t \rightarrow \infty, z \rightarrow \infty \Rightarrow t \rightarrow -\infty \end{aligned} \quad (9)$$

the Mellin transform can be calculated by the Fourier transform with logarithmically deformed time axis:

$$\begin{aligned} V^M(f) &= \int_{-\infty}^{\infty} v(e^{-t})e^{-t(i2\pi f-1)}(-e^{-t})dt \\ &= \int_{-\infty}^{\infty} v(e^{-t})e^{-i2\pi ft}dt \end{aligned} \quad (10)$$

Now the scaling of a function/signal using a logarithmically deformed axis can be transferred into a shift of its spectrum. Finally, the spectrum's magnitude is logarithmically re-sampled on its radial axis and concurrently the spectrum is arranged in polar coordinates exploiting the rotational properties of a 2D Fourier transform as described before. Scaling and rotation of an image frame are then transformed into a 2D signal shift where the 2D signal is actually the corresponding spectrum magnitude of the image frame. This intermediate step is called the FMI descriptor.

The overall algorithm is sketched here. First the calculation of the POMF is shown:

- 1) calculate the spectra of two corresponding image frames
- 2) calculate the phase difference of both spectra
- 3) apply an inverse Fourier transform of this phase difference

The following steps are taken for a full determination of the rotation, scaling and translation parameters:

- 1) calculate the spectra of two corresponding image frames (in *image-resolution*)
- 2) calculate the magnitude of the complex spectral data
- 3) re-sample the spectra to polar coordinates (in *polar-log-resolution*)
- 4) re-sample the radial axes of the polar spectra logarithmically
- 5) calculate a POMF on the re-sampled magnitude spectra
- 6) determine the corresponding rotation/scaling parameters from the Dirac pulse
- 7) re-size and re-rotate the corresponding image frame to its reference counterpart
- 8) calculate a POMF between the reference and re-rotated and re-scaled image
- 9) determine the corresponding x,y translation parameters from the Dirac pulse

The steps are used in the Fourier Mellin based image registration in a straightforward way. A sequence of images I_k is acquired. Images I_n and I_{n+1} are processed with the above calculations. The result is the transformation T_n^M in the frame of reference F_n of I_n between I_n and I_{n+1} and hence the motion of the camera. Some properties of this image registration method have been described in [27] while applications for underwater [5], [23] and aerial [6] robots used the image registration for photo mapping.

III. VIDEO STABILIZATION AND MOTION SEGMENTATION

The image registration over two consecutive frames as described in the above section is the first step for the video stabilization, which itself is used to perform the motion segmentation.

A. Video Stabilization Application

The inverse of the transformation T_n^M between images I_n and I_{n+1} is applied to I_{n+1} . This way the contents (features) of the images do not move for the observer. If the application is to provide a stabilized video stream for an observer or operator additional steps have to be taken. The problem is, that the transformations accumulate and the frames thus would leave the area of the output frame.

So, PID controllers are applied to the components of the transform to keep the center of the image in the view. The errors fed into the controllers come from the transformation of the current frame with respect to the output frame. After tuning the parameters of the PID controller the image is thus always gently pulled back into the center of the output frame while still compensating for abrupt motions and vibrations.

B. Motion Segmentation

The motion segmentation can work with two (I_n and I_{n+1}) or three (I_{n-1} , I_n and I_{n+1}) consecutive frames. All work is done in the frame of reference of image I_n , such that the other images have to be transformed accordingly. Motion in parts of the frames that do not overlap cannot be detected, such that those pixels are marked as no motion (black) per definition.

In the next step the images are converted into the hue, saturation, and lightness (HSL) color space. Only the lightness value is used to calculate the difference d for each pixel between the image frames in the next step. A threshold value t determines if the lightness difference between those pixels is big enough for them to be marked as motion. In the three frame version of the algorithm both differences (between I_{n-1} and I_n as well as between I_n and I_{n+1}) have to exceed the threshold.

In the implementation used for this paper the image transformation is done on the fly while calculating the pixel differences. Pixels found to represent motion are marked by giving them the color of said pixels in frame I_n (it has to be different than black).

C. Filter

Two different kind of filters are applied to the result of the registered motion segmentation. First the pixels are applied to a neighborhood-criterion and then a minimum blob-size for segmented parts is enforced.

Often line-shaped features (e.g. edges of buildings or stairs) are present in the environment. Then small errors in the registration can lead to false positives that have a similar, line-shaped appearance. The neighborhood filter counts how many of the four direct neighbors also indicate motion. A threshold value for the minimum count (between 1 and 4) then determines, whether said pixels remains valid.

The blob filter tries to count all pixels belonging to the same segment of detected motion in the frame. A minimum size is then applied to all blobs, disregarding all of those which are too small.

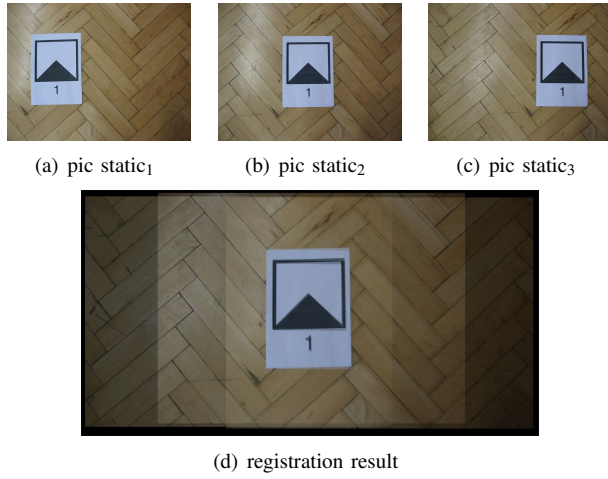


Fig. 3. Registration of 3 pictures from a scene with a static marker.

IV. EXPERIMENTS AND RESULTS

In this section the image registration and stabilization is first evaluated with a lab-experiment. Data from three aerial missions is then used in the second experiment to demonstrate the proposed algorithms.

A. Evaluation with Artificial Markers

The method is first evaluated with artificial markers as ground truth reference points. Figure 3 shows a simple static scene with a marker. Three snapshots are taken with the camera moving from right to left (Figures 3(a), 3(b), and 3(c)). Please note that the camera is simply hand-held and that no camera calibration is used. Nevertheless, the image stabilization is very accurate as indicated by the registration result shown in Figure 3(d). This is also supported by an evaluation of the displacements between the marker positions in the input images versus the registered ones (Table I). The camera has a resolution of 1600×1200 pixels. The displacement of 364 pixels for $\text{static}_1 \rightarrow \text{static}_2$ hence corresponds to about 23% of the image width, while the 413 pixels for $\text{static}_2 \rightarrow \text{static}_3$ are 26% of the image width. Nevertheless, the registration keeps the marker almost perfectly in a static place.

TABLE I
THE DISPLACEMENTS IN PIXELS BETWEEN THE MARKER POSITIONS OF TWO
SUBSEQUENT IMAGES IN THE STATIC SCENE (FIGURE 3)

	unregistered	registered
$\text{static}_1 \rightarrow \text{static}_2$	364	3
$\text{static}_2 \rightarrow \text{static}_3$	413	2

Figure 4 shows a second scene with two markers where one of them, namely marker #2, moves. Four snapshots are taken with the marker #2 moving randomly (Figures 4(a), 4(b), 4(c), and 4(d)). Please note that the camera is also here just hand-held. Again, the image stabilization is working well as indicated by the registration result shown in Figure 4(e).

This is also supported by an evaluation of the localization of the moving marker #2 between the marker positions in

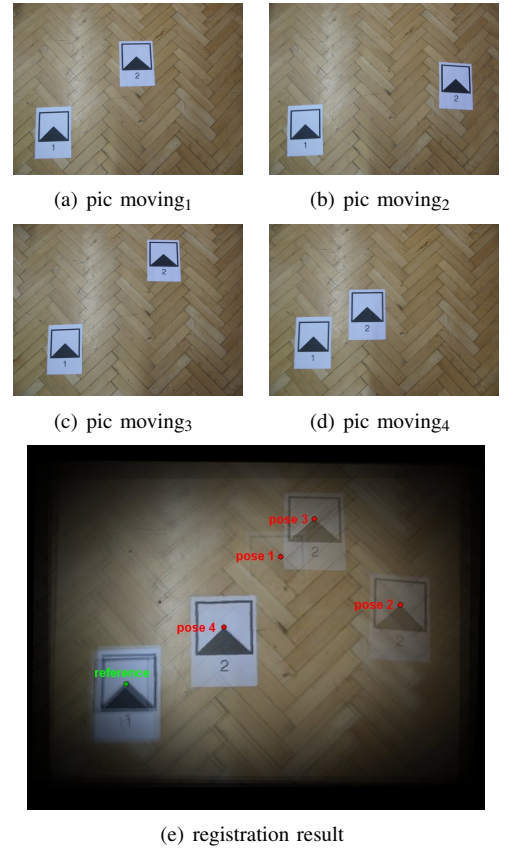


Fig. 4. Registration of 4 pictures from a scene with a moving marker (#2) and a static one (#1) as reference.

TABLE II
THE GROUND TRUTH DISTANCE D BETWEEN MARKER #1 AND THE MOVING MARKER
#2 FROM THE SCENE SHOWN IN FIGURE 4, AND THE RELATED ERROR IN THE
REGISTERED IMAGE.

	D (pixel)	error (pixel)
pose 1	630	9
pose 2	889	15
pose 3	777	33
pose 4	350	17

the input images versus the registered ones (Table II). The static marker #1 is again quite stable in a fixed position; its position varies by about 18 pixels on average between the registered images. The average of the coordinates of marker #1 is used as a reference point. The Euclidean distance in pixels between the position of marker #2 in the registered image 4(e) and this reference is then compared to the ground truth distance in each of the images 4(a), 4(b), 4(c), and 4(d).

B. Motion Segmentation on Aerial Video Data

For the second experiment data collected with a Unmanned Aerial Vehicle (UAV) is applied to the motion segmentation algorithm. On three different occasions a vehicle passed through the field of view of the robot. A total of 196 frames from those three scenes are used in this experiment. The UAV is flying over the campus of the Jacobs University

in Bremen. Due to wind and steering commands of the operator the vehicle is showing considerable ego-motion.

Please refer to the video attached to this paper for the original movie as well as the different results.

In the following, results are presented for four different approaches to the motion segmentation problem. First, the naive approach of motion segmentation without image registration is tested ("no iFMI"). In the second set, the result of the naive method is filtered. The ego-motion of the camera is compensated by applying the iFMI image registration algorithm ("iFMI") in the third approach. The fourth test additionally applies the filter to the result of the second algorithm ("iFMI plus filter").

The runtime for the image registration does not depend on the content of the frame but solely on the resolution used, which is 476x476 pixels in these experiments. The video was actually captured with a resolution of 640x480 pixels, but due to some disturbances on the top and bottom of the frames two pixels were omitted on both ends, such that the frames were cropped to the square resolution mentioned above. The registration can run with 17.1 frames per second (fps) on a 2.8 GHz laptop. The computation time for the filter depends on the amount of marked pixels, but it is anyways much less than the time needed for the image registration ("iFMI"). For the whole application which loads the images, displays the original as well as a video of the detected motions in a GUI etc. the speed is 8.1 fps.

The experiments were performed with the two frames approach described in section III-B and the neighborhood filter used a minimum of four neighbors while the blob filter demanded at least 20 marked pixels in one group.

Figure 5 shows the amount of pixels that suggest movement for each frame. Artificial entries to mark the 99 frames that actually contain moving cars have been added as "Car present". It can be seen that the amount of marked pixels (pixels that indicate motion in that part of the frame) is significantly higher for the naive approach. The number of pixels marked by the naive approach directly corresponds to the amount of movement of the UAV. As expected there are fewer marked pixels after filtering the naive approach. Without moving cars the iFMI algorithm gives more than 4.5 times fewer amounts of marked pixels, all of which are removed by the filter of the iFMI+ approach.

The significantly less amount of marked pixels when iFMI and iFMI+ are used can also be seen in the histogram in Figure 6. That histogram has 20 bins for 97 frames that do not have moving cars. The non-registration approach dominates the high-percentage range.

A more compact representation of these results is presented in Table III. The percentages of marked pixels are averaged for the four methods. The standard deviation and the maximum is also calculated. The first part of the table contains those values for all the frames. The 99 frames that actually contained movement are omitted in the bottom of this table, such that those numbers then provide an indication for the amount of false positives in the frames. Please note that these numbers correspond to the number of pixels and

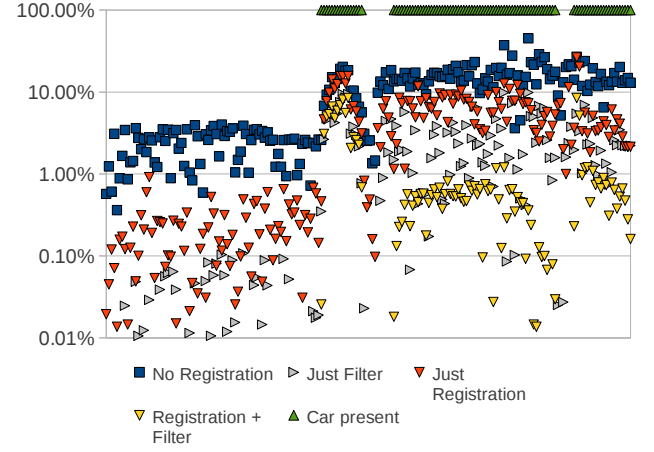


Fig. 5. The amount of pixels with detected motion is here shown for each frame. The "Car Present" entries mark the frames where there is actually motion due to the cars. Please note the logarithmic scale of the y-axis.

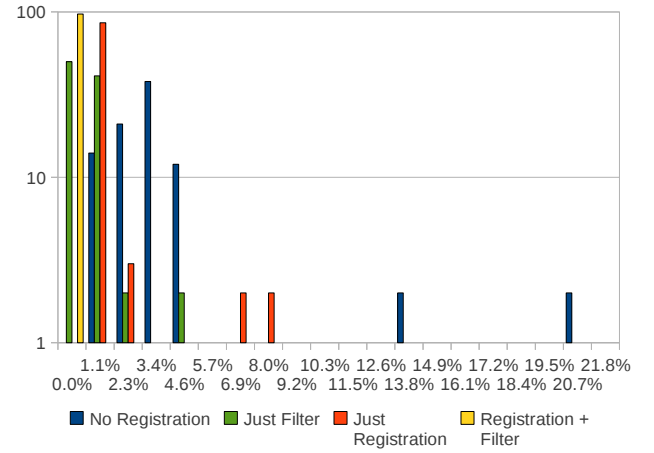


Fig. 6. Histogram for the 97 frames without cars. The 20 bins are for the number of pixel that detected motion. For the filtered case all 97 frames have zero non-black pixel while the frames using registration feature far less pixel then the ones without. Please note again the logarithmic scale of the y-axis.

not to the number of objects themselves. This means, that though not all moving pixels of a car may be fully segmented, there are typically more marked pixels than cars.

The frames can be classified as containing motion or not, simply by checking for the existence of at least one marked pixel in that frame. In Table IV the results of that classification is checked for correlation with the 99 frames found to contain car motion. It should be noted, that in 9 frames (4.6% of all images) that contained moving cars the iFMI+ algorithm did, additionally to those pixel of the car, mark other regions that contained no motion as well.

The following figures also illustrate the results in qualitative terms. Figure 7 shows the arrangement of the results of the different methods in these figures. On the left the input video frame I_n can be seen while the right side shows

TABLE III
PERCENTAGES OF PIXELS WITH DETECTED MOTION PER FRAME.

	in %	Registration:			
		None	Filter	iFMI	iFMI+
All frames	Average	9.81	1.58	3.82	0.63
	Std. deviation	8.08	2.23	4.39	1.52
	Maximum	45.27	9.86	26.41	8.47
Without car frames	Average	3.78	0.22	0.80	0
	Std. deviation	4.47	0.81	1.87	0
	Maximum	21.71	4.67	11.49	0

TABLE IV
THE CORRECTNESS OF SEGMENTED FRAMES - AT LEAST ONE DETECTED PIXEL COUNTS AS POSITIVE (MOTION). THE TRUE POSITIVE AND TRUE NEGATIVE VALUES ARE RELATIVE TO THE NUMBER OF FRAMES WITH/ WITHOUT MOTION.

in %	Registration:			
	None	Filter	iFMI	iFMI+
True Positive	100	100	100	99
True Negative	0	52	0	100
False Positive	100	48	100	0
False Negative	0	0	0	1

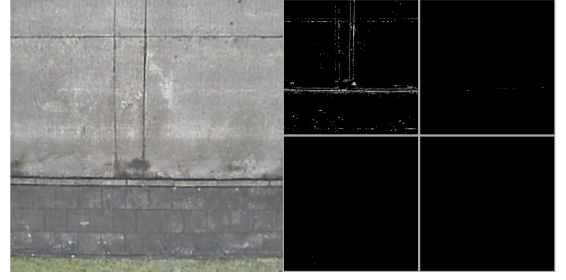
the four approaches. The top left image shows Δ_n as the thresholded differential image between I_n and I_{n+1} . The filtered differential image is shown on the top right. The image Δ_n^{iFMI} as the thresholded differential images after iFMI registration of I_n and I_{n+1} is shown on the bottom left in each figure. The same image with additional filtering is shown on the bottom right.

Figure 8(a) shows that the naive approach can create as output Δ_n (top left) significant amounts of marked pixels even if there are no moving objects. This is due to the ego-motion of the camera. Since the iFMI registration compensates for that, the images Δ_n^{iFMI} (bottom left) and Δ_n^{iFMI+} (bottom right) do not contain marked any pixel. But when there is a moving object as shown in Figure 8(b), the iFMI compensated pictures clearly shown the motion. But there can be errors for those pixels, e.g., due to:

- errors in the image registration
- image distortions if a tilt is present in the frames
- perspective errors due to the 3D nature of the environment
- due to the sometimes abrupt movement of the aerial vehicle the image can be blurred

input video frame	no iFMI	filter
	iFMI	iFMI plus filter

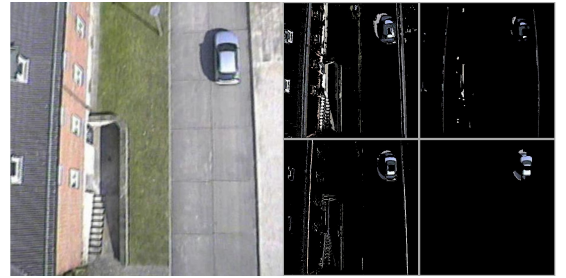
Fig. 7. The arrangement of the images in the following example figures. The left shows the input video frame I_n . On the top left Δ_n , i.e., the thresholded differential image between I_n and I_{n+1} is shown. The top right image shows, as comparison, the filtered frame of Δ_n . The image Δ_n^{iFMI} as the thresholded differential images after iFMI registration of I_n and I_{n+1} is shown on the bottom left. The same image with additional filtering, i.e., Δ_n^{iFMI+} , is shown on the bottom right.



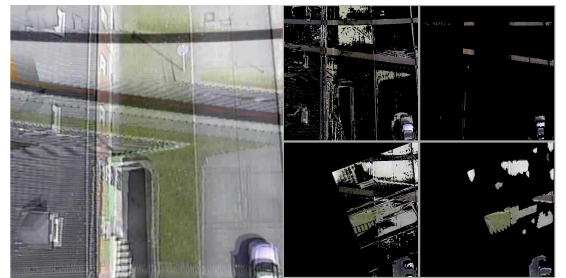
(a) An example without cars, i.e., moving objects, which corresponds to the 35-th and the 36th frame in the input video. The top left differential image Δ_{35} contains quite some noise due to the ego-motion of the robot. Even the filtering of the differential image (top right) leaves false positives in this frame. Both iFMI-processed images Δ_{35}^{iFMI} and Δ_{35}^{iFMI+} shown on the bottom contain no noise and are fully black, i.e., the absence of movement is correctly detected in them.



(b) A car is present in this example. Both iFMI-processed images Δ_{143}^{iFMI} and Δ_{143}^{iFMI+} (bottom) correctly show a correct motion segmentation of this car, despite the motion of the aerial vehicle. The unfiltered iFMI-processed image Δ_{143}^{iFMI} still contains some residual noise which is then removed by the filter. The comparison image Δ_{143} (top left) contains a lot of noise due to the displacements between the input image frames that can not be removed by the filter (top right).



(c) This example illustrates in the image Δ_{194}^{iFMI+} (bottom right) the benefit of the additional filter, which is able to remove the significant amounts of noise in Δ_{194}^{iFMI} (bottom left) when the registration occasionally is some pixel off.



(d) In frame 180 the video link to the ground station was disturbed. The registration fails in this case, as can be seen in the image Δ_{179}^{iFMI} . The disturbance results in two frames with bad object detection results: Δ_{179}^{iFMI} and Δ_{180}^{iFMI} are both affected since both use the bogus frame 180.

Fig. 8. Example frames from the experiment.

Therefore the additional filter is beneficial. Figure 8(c) shows that the filter can remove errors from the registered result while still detecting the moving objects. Please note that the filter tends to remove pixels from the outline of objects, i.e., not only their presence but also their visual center and hence their local position can still be very well determined.

Figure 8(d) shows that disturbances in the video connection cannot be compensated for and lead to two unusable detection results.

V. CONCLUSIONS

A video stabilization and motion segmentation approach suited for aerial video data was presented. It can be used to automatically detect motion in the scene from an autonomously flying UAV; for example to detect an intruder in a surveillance mission or to detect a waving victim trapped in a rubble pile during a search and rescue operation.

The method is based on improved Fourier Mellin Invariant (iFMI) registration, which uses a logarithmic representation of the spectral magnitude of the FMI descriptor. In addition, a filter on the frequency where the shift is supposed to appear is applied. This leads to robust registrations and hence motion compensation under the extensive ego-motion of aerial vehicles and under the presence of disturbance due to moving objects. The motion compensation is used for video-stabilization and subsequently for object segmentation. Experiments with artificial markers and with aerial video are presented to illustrate that the approach works as intended.

REFERENCES

- [1] G. Belloni, M. Feroli, A. Ficola, S. Pagnottelli, and P. Valigi. An autonomous aerial vehicle for unmanned security and surveillance operations: design and test. In *Safety, Security and Rescue Robotics, 2007. SSRR 2007. IEEE International Workshop on*, pages 1–4, 2007.
- [2] A. Birk, B. Wiggerich, H. Buelow, M. Pfingsthorn, and S. Schwertfeger. Reconnaissance and camp security missions with an unmanned aerial vehicle (uav) at the 2009 european land robots trials (elrob). In *IEEE International Workshop on Safety, Security, and Rescue Robotics (SSRR)*. IEEE Press, 2009.
- [3] A. Birk, B. Wiggerich, H. Buelow, M. Pfingsthorn, and S. Schwertfeger. Safety, security, and rescue missions with an unmanned aerial vehicle (uav): Aerial mosaicking and autonomous flight at the 2009 european land robots trials (elrob) and the 2010 response robot evaluation exercises (tree). *Journal of Intelligent and Robotic Systems*, 2011.
- [4] H. Buelow and A. Birk. Fast and robust photomapping with an unmanned aerial vehicle (uav). In *International Conference on Intelligent Robots and Systems (IROS)*. IEEE Press, 2009.
- [5] H. Buelow, A. Birk, and V. Unnithan. Online generation of an underwater photo map with improved Fourier Mellin based registration. In *International OCEANS Conference*, IEEE Press, 2009.
- [6] H. Bulow and A. Birk. Fast and robust photomapping with an unmanned aerial vehicle (uav). In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 3368–3373, Oct. 2009.
- [7] Q.-S. Chen, M. DeFrise, and F. Deconinck. Symmetric phase-only matched filtering of fourier-mellin transforms for image registration and recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 16(12):1156–1168, 1994.
- [8] E. De Castro and C. Morandi. Registration of translated and rotated images using finite Fourier transforms. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-9(5):700–703, 1987.
- [9] S. Dhaliwal and A. Ramirez-Serrano. Control of an unconventional vtol uav for search and rescue operations within confined spaces based on the marc control architecture. In *Safety, Security and Rescue Robotics (SSRR), 2009 IEEE International Workshop on*, pages 1–6, 2009.
- [10] A. Engelsberg and G. Schmidt. A comparative review of digital image stabilising algorithms for mobile video communications. *Consumer Electronics, IEEE Transactions on*, 45(3):591–597, 1999.
- [11] R. Goecke, A. Asthana, N. Pettersson, and L. Pettersson. Visual vehicle egomotion estimation using the Fourier-Mellin transform. In *Proceedings of the 2007 IEEE Intelligent Vehicles Symposium*, pages 451–455, 2007.
- [12] M. Goodrich, J. Cooper, J. Adams, C. Humphrey, R. Zeeman, and B. Buss. Using a mini-uav to support wilderness search and rescue: Practices for human-robot teaming. In *Safety, Security and Rescue Robotics, 2007. SSRR 2007. IEEE International Workshop on*, pages 1–6, 2007.
- [13] L. V. Gool, T. Moons, and D. Ungureanu. *Affine/Photometric Invariants for Planar Intensity Patterns*. 1996.
- [14] J. Jin, Z. Zhu, and G. Xu. A stable vision system for moving vehicles. *Intelligent Transportation Systems, IEEE Transactions on*, 1(1):32–39, 2000.
- [15] B. Jung and G. S. Sukhatme. Detecting moving objects using a single camera on a mobile robot in an outdoor environment. In *8th Conference on Intelligent Autonomous Systems*, pages 980–987, 2004.
- [16] Y.-M. Liang, H.-R. Tyan, S.-L. Chang, H.-Y. Liao, and S.-W. Chen. Video stabilization for a camcorder mounted on a moving vehicle. *Vehicular Technology, IEEE Transactions on*, 53(6):1636–1648, 2004.
- [17] D. Longo, D. Melita, G. Muscato, and S. Sessa. A mixed terrestrial aerial robotic platform for volcanic and industrial surveillance. In *Safety, Security and Rescue Robotics, 2007. SSRR 2007. IEEE International Workshop on*, pages 1–6, 2007.
- [18] D. G. Lowe. *Object Recognition from Local Scale-Invariant Features*, pages 1150–1157. 1999.
- [19] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.
- [20] K. Mikolajczyk and C. Schmid. *A Performance Evaluation of Local Descriptors*. June, 2003.
- [21] A. Ollero, M. Bernard, M. La Civita, L. van Hoesel, P. Marron, J. Lepley, and E. de Andres. Aware: Platform for autonomous self-deploying and operation of wireless sensor-actuator networks cooperating with unmanned aerial vehicles. In *Safety, Security and Rescue Robotics, 2007. SSRR 2007. IEEE International Workshop on*, pages 1–6, 2007.
- [22] A. Ollero, J. Ferruz, F. Caballero, S. Hurtado, and L. Merino. Motion compensation and object detection for autonomous helicopter visual navigation in the comets system. In *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, volume 1, pages 19–24 Vol.1, 2004.
- [23] M. Pfingsthorn, A. Birk, S. Schwertfeger, H. Bülöw, and K. Pathak. Maximum likelihood mapping with spectral image registration. In *Robotics and Automation, 2010. ICRA 2010. Proceedings of the 2010 IEEE International Conference on*, 2010.
- [24] K. Pratt, R. Murphy, J. Burke, J. Craighead, C. Griffin, and S. Stover. Use of tethered small unmanned aerial system at berkman plaza ii collapse. In *Safety, Security and Rescue Robotics, 2008. SSRR 2008. IEEE International Workshop on*, pages 134–139, 2008.
- [25] B. Reddy and B. Chatterji. An fft-based technique for translation, rotation, and scale-invariant image registration. *Image Processing, IEEE Transactions on*, 5(8):1266–1271, 1996.
- [26] T. Schamm, M. Strand, T. Gump, R. Kohlhaas, J. Zollner, and R. Dillmann. Vision and tof-based driving assistance for a personal transporter. In *International Conference on Advanced Robotics (ICAR)*, pages 1–6. IEEE Press, 2009.
- [27] S. Schwertfeger, H. Bülöw, and A. Birk. On the effects of Sampling Resolution in Improved Fourier Mellin based Registration for Underwater Mapping. In *6th International Symposium on Intelligent Autonomous Vehicles (IAV 2010)*. IFAC, 2010.
- [28] J. Shi and C. Tomasi. Good features to track. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR94)*, 1994.
- [29] B. Steder, G. Grisetti, C. Stachniss, and W. Burgard. Visual slam for flying vehicles. *Robotics, IEEE Transactions on*, 24(5):1088–1093, 2008.
- [30] N. Sunderhauf, S. Lange, and P. Protzel. Using the unscented kalman filter in mono-slam with inverse depth parametrization for autonomous airship control. In *Safety, Security and Rescue Robotics, 2007. SSRR 2007. IEEE International Workshop on*, pages 1–6, 2007.
- [31] F. Vella, A. Castorina, M. Mancuso, and G. Messina. Digital image stabilization by adaptive block motion vectors filtering. *Consumer Electronics, IEEE Transactions on*, 48(3):796–801, 2002.