ALLFLIGHT – IMAGING SENSORS FUSED WITH LADAR DATA FOR MOVING OBSTACLE DETECTION

Hans-Ullrich Döhler, Niklas Peinecke, Sven Schmerwitz, Thomas Lüken

Institute of Flight Guidance, German Aerospace Center, DLR
Lilienthalplatz 7, D-38108 Braunschweig

ulli.doehler@dlr.de
phone: +49 (0)5 31 / 2 95 – 21 79
fax: +49 (0)5 31 / 2 95 – 25 50

niklas.peinecke@dlr.de
phone: +49 (0)5 31 / 2 95 – 30 29
fax: +49 (0)5 31 / 2 95 – 25 50

sven.schmerwitz@dlr.de
Phone : +49 (0)5 31 / 2 95 – 30 09
fax: +49 (0)5 31 / 2 95 – 25 50

thomas.lueken@dlr.de
phone: +49 (0)5 31 / 2 95 – 30 28
fax: +49 (0)5 31 / 2 95 – 25 50

OVERVIEW

Supporting a helicopter pilot during landing and takeoff in degraded visual environment (DVE) is one of the challenges within DLR's project ALLFlight (Assisted Low Level Flight and Landing on Unprepared Landing Sites). Different types of sensors (TV, Infrared, mmW radar and laser radar) are mounted onto DLR’s research helicopter FHS (flying helicopter simulator) for gathering different sensor data of the surrounding world. A high performance computer cluster architecture acquires and fuses all the information to get one single comprehensive description of the outside situation.

While both TV and IR cameras deliver images with frame rates of 25 Hz or 30 Hz, Ladar and mmW radar provide geo-referenced sensor data with only 2 Hz or even less. Therefore, it takes several seconds to detect or even track potential moving obstacle candidates in mmW or Ladar sequences. Especially if the helicopter is flying with higher speed, it is very important to minimize the detection time of obstacles in order to initiate a re-planning of the helicopter’s mission timely. Applying feature extraction algorithms on IR images in combination with data fusion algorithms of extracted features and Ladar data can decrease the detection time appreciably. Based on real data from flight tests, the paper describes applied feature extraction methods for moving object detection, as well as data fusion techniques for combining features from TV/IR and Ladar data.

1. INTRODUCTION

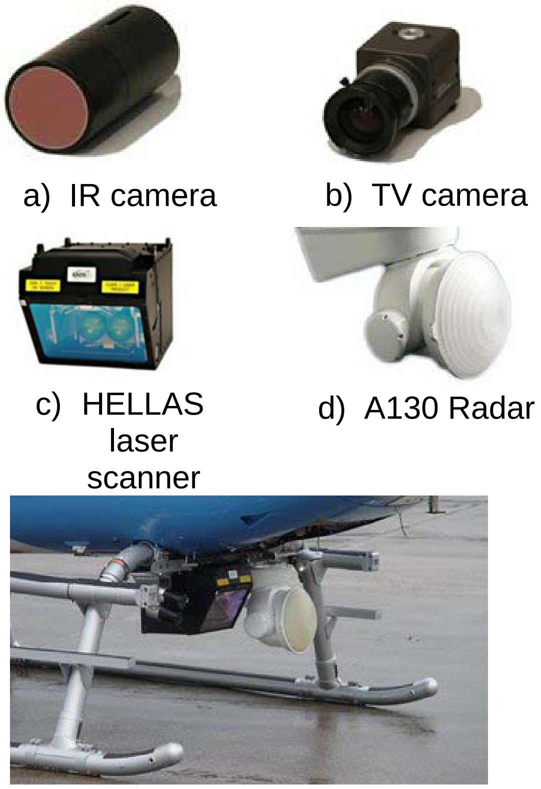
Since 2008 DLR has investigated a quite large effort within the project ALLFlight (Assisted Low Level Flight and Landing on Unprepared Landing Sites). This project was mainly driven by the objective to help pilots remaining visual control under adverse weather conditions by means of weather penetrating sensors. We equipped our research helicopter FHS (Flying Helicopter Simulator), a modified Eurocopter EC135 model, with a number of imaging sensors. Regarding the different available sensor hardware the project relied mainly onto commercial-of-the-shelf (COTS) equipment.

From MaxViz, Inc., Portland, USA, we loaned a thermal infrared bolometer camera (Figure 1a). The camera offers an internal resolution of 320 x 240 pixels with a thermal resolution of approx. 0.1 K (noise equivalent temperature). The camera is coupled to the acquiring computer by RS-170 analog video input with a virtual resolution of 640 x 480 pixels and a frame rate of 30 Hz.

The TV camera with a native resolution of 752 x 512 pixels is also interfaced by RS-170 analog video input (Figure 1b). The computer internal image resolution delivers images with 640 x 480 pixels with an image rate of 25 Hz.

The laser scanner system (HELLAS) was purchased from former EADS company, Friedrichshafen, Germany, now known as Cassidian (Figure 1c). HELLAS is said to be a mature laser scanning system, already applied to several helicopter fleets (German Army and German “Bundespolizei” helicopter fleets are partly equipped with this system and/or its successors). Our system offers a resolution of 95 x 200 pixels with a frame rate of 2 Hz. The mechanically working scanner applies a high-power laser and high-gain receiver within the 1.4 micron spectral region and has a maximum detection range of 1000 meters. The range resolution is said to be something in the range of 1 meter. The scanner’s field of view (FOV) is approximately 32 x 32 degrees. We purchased this system with a modified firmware, which delivers the sensed data via TCP/IP interface, consisting of maximal 19000 data points for each image frame. Each data point is denoted as 4 element vector, the 3D-position and a scalar time-stamp for accurate multi-sensor data fusion purpose. The 3D-position is system internally geo-referenced to WGS-84 coordinates (latitude, longitude, altitude). The altitude is referenced to the WGS-84 ellipsoid, like GPS does. Of course the system has to know the helicopter’s position data (from the helicopter’s experimental DGPS receiver using SBAS, like WAAS or EGNOS) and its Euler angles from the inertial reference system (IRS), as well as the system’s time reference for later 4D data fusion. These data are supplied from the FHS data management computer (DMC) to the HELLAS system by an ARINC-429 interface with high frequency.

The fourth installed sensor, a radar system (Type AI-130), was purchased from ICX, Canada. It is a millimeter wave (mmW) impulse radar (RF 35 GHz) with a mechanically scanning antenna. It has a beam width of about 2.4 x 1.8 degree, and is able to scan a field of view with a size of 30 x 20 degree within 1.8 seconds Figure 1d). Due to the free programmable scan pattern the system can be moved in azimuth direction from -90 up to 90 degree, as well as from 20 degree above horizon down to -90 degree below horizon. Although the system can deliver targets in a range of 8 nautical miles, we apply it with an instrumented range of 1920 meters, which is enough for our application. In that case the radar has a range resolution of 1.8 meters. Data interfacing to the acquiring computer system is realized via raw data Ethernet. System control applies a RS-432 serial communication line.



a) IR camera

b) TV camera

c) HELLAS laser scanner

d) A130 Radar

Figure 1: ALLFlight sensors mounted between the landing skid

The following chapters describe our new developed method of combining the laser scanner data with sensed image data from IR and TV cameras to detect moving targets in front of the helicopter. Beside the naked detection of moving targets and there 3D position in world coordinates, our method determines 4D trajectories (3D world coordinates and time) of these targets in real-time.

Although combinations of camera and laser scanner data play an important role within recent R&D programs (see [1] and [2]), we only found a minor number of publications regarding the field of helicopter guidance. For example, Hebel et. al. [3] deals with target detection for helicopter guidance, but speed measurement (in terms of 4D trajectories) plays only a minor role within that publication.

Our presentation is structured as follows:

Based on the state-of-the-art regarding camera calibration, we describe a modified calibration plate, which we applied for calibration of our IR camera. Then we describe how to extract image features from the image input streams (TV and IR). We call that step feature extraction. Then we describe the manner how to combine extracted features from succeeding images which delivers so-called feature tracks. These tracks are denoted as 3D-trajectories (2D image coordinates and time). Due to the fact that our cameras are directly mounted onto our helicopter which is moving around its axes, we get a quickly increasing number of such feature tracks over time. The next step is the data fusion by feeding-in the precisely time-stamped laser scanner data onto these feature tracks. This is considered as a filtering process of the 3D trajectories. Most of them “pass through the sieve” because they point to some static location in the 3D world. The statistical analysis of all remaining 3D trajectories builds up the wanted result, which we call 4D trajectories of moving targets (denoted by 3D position and time).

Thereafter we describe some results of our method based on simulated and real flight test data followed by our summary and conclusion.

2. CAMERA CALIBRATION

For the purpose of data fusion from different image sensing systems onto a common coordinate system, a precise knowledge on the specific distortion of each contributing sensor is essential. Our applied laser scanner (HELLAS) is an already calibrated system. It is manufactured to deliver the measured “3D pixel clouds” with high precision, noted in geo-referenced 3D coordinates. To combine these data with images from our IR and TV cameras one of the most important problem to be solved is an accurate camera calibration. This topic is presented in detail within this chapter.

2.1 Calibration pattern

For image processing purpose we apply the OpenCV framework [5], which offers a simply usable tool-chain for camera calibration. Thermal IR cameras (8 to 12 micron) do not deliver images with sufficient contrast when looking on-paper-printed patterns. Our first experiments with a black-and-white printed pattern on a sheet of paper which was illuminated with two flood lamps (2 x 250 Watts) confirmed this expectation. There have been published recently some quite simple methods for building up IR-camera calibration patterns [4], but these proposals offer too small dimensions which are not appropriate for our IR-camera. Since the MaxViz camera has a fixed focus lens (adjusted to infinity) sharp images below a distance of 2 meters cannot be obtained. So it was necessary to apply a pattern of sufficient size. We produced a special heated pattern on a stiff wooden plate with a size of 1.20 x 0.80 meters. On this plate we mounted an array of small resistors powered by an adjustable laboratory power supply (0 to 30 volts with 1 to 3 amps). We adjusted the power supply, so that each resistor dissipates a power of something around 0.5 Watts. The array consists of 5 x 7 resistors with a very accurate spacing of 12 cm. The outer rectangle of the grid resulted in a size of 0.60 x 0.84 meters. To minimize mirroring effects of the plate's background, we applied some black paint. Figure 2 shows the IR image from the MaxViz camera.

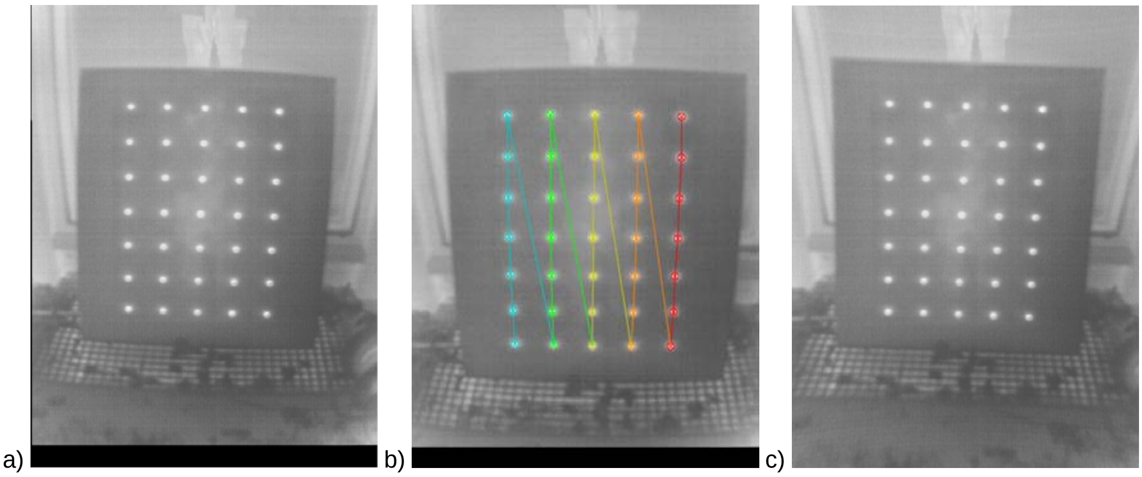


Figure 2: IR camera calibration: a) raw image, b) extracted calibration pattern with 5 x 7 blobs, c) rectified camera image

2.2 Calibration results

We applied the above described heated pattern for calibrating the IR camera. To enhance the image contrast further, we located the plate outside the lab on a rather cold winter day (outside temperature something around zero degree Celisus). Due to the large and rather heavy calibration plate we took some image sequences not by changing the plate's orientation in front of the camera, but by turning manually the IR-camera relatively to the statically fixed plate (Figure 2 a) and c)).

The calibration of the standard TV-camera was done in a similar manner (moving the camera relatively to a static pattern). But instead of using the resistor plate we applied a state-of-the-art printed pattern of the same size on a sheet of paper which was placed onto a fixed surface.

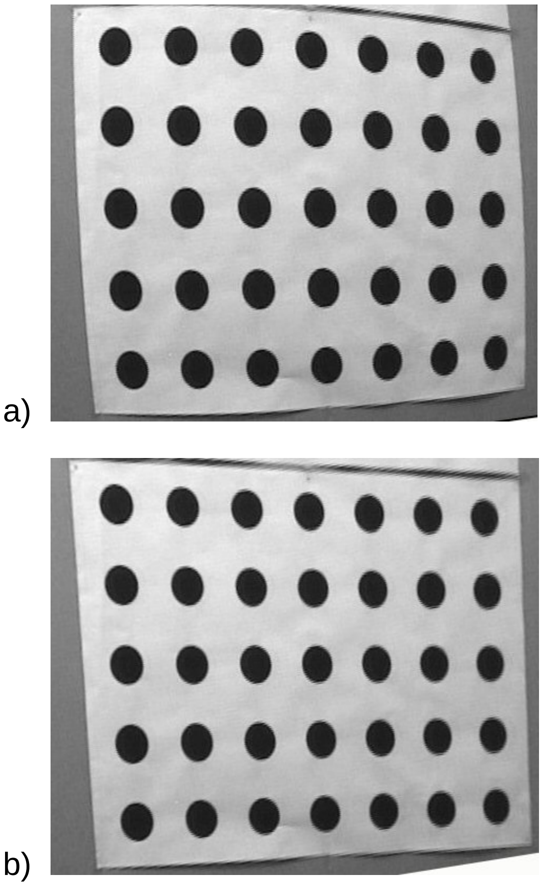


Figure 3: TV-camera calibration: a) raw image, b) rectified image

3. PROCESSING CONCEPT AND IMPLEMENTATION

The processing concept of our approach consists of several single processing steps which shall be described in some more detail now (Figure 4 and Figure 5). The implementation of our concept is coded in C/C++ and applies several elements of the OpenCV framework. We will directly refer to the names of OpenCV functions if applicable (see [5], [6].)

We start the processing chain with image rectification, which means eliminating the camera's lens distortion and all other imperfections of the real camera in order to obtain a perfect pin-hole camera model. Figure 3a) shows the visible barrel-shaped distortion of our TV camera. The parameters resulting from the camera calibration process are described by distortion and intrinsic and have to be fed into the cvInitUndistortMap() function. The manner how cvRemap() is able to produce a perfectly rectified image is shown in Figure 3b). Thereafter we apply some additional low pass and/or noise reduction filtering by cvSmooth(). The median or the gauss low pass filtering is applied for that purpose. The cvSmooth() function is guided by the CV_GAUSSIAN or the CV_MEDIAN parameter which controls the filtering behavior. The edge size of the applied filter window is controlled by the parameter filter_size.

3.1 Feature extraction in IR and TV image

The next step of data processing consists in extracting the features from each frame of the incoming image stream. For that purpose we extract two different types of features:

Corners: The cvGoodFeaturesToTrack() function delivers a list of relatively noise stable 2D locations of corners within each image. This function combines the so-called Harris detector [9] and the method published by Shi and Tomasi [8]. This type of feature is defined by its 2D location, only. The function applies several tuning parameters so that stable results are achieved. The disadvantage of the implementation of this function is that the number of generated features varies according to the gray value statistics of the image. Nevertheless this function delivers a list of corners which is added to the list of features.

Circles: A type of feature which is generated by combining several process steps. At the beginning we apply edge detection using cvCanny() [7]. Then contour following is applied via the cvFindContours() function. This results in a data structure of extracted contour features from which circular sub-sets are generated. For this purpose we apply our own function fit_circles(), which searches for approximate circular shaped sub-sets of the contour. The function is controlled by 3 additionally defined tuning parameters: circle_error, circle_radius (min, max) and circle_contrast. The result is a list of circles which is added to the list of features.

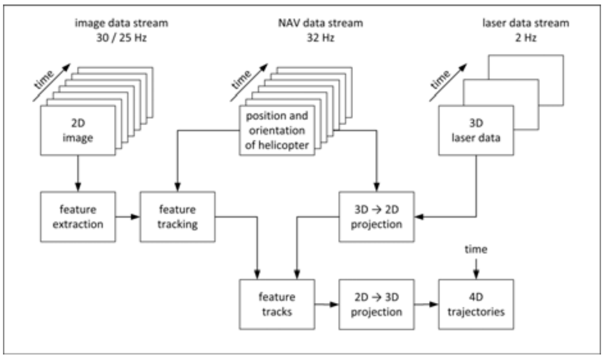


Figure 4: Block schematic of processing concept.

The resulting list of features is then appended to a list of feature tracks (image locations of features over time). Whenever a feature is close enough to the recently entered feature of an existing track (with respect to its 2D location and other parameters), then this feature is appended to that track. Otherwise, if a recently found feature does not match with any existing track, a new feature track is initiated. Each feature entry to the list of feature tracks is time-stamped with the overall image acquisition time. The list of feature tracks is controlled by a parameter called time_horizon. Feature tracks which become older than the defined time horizon are automatically deleted from the list of feature tracks. The remaining feature tracks are called “living tracks”.

3.2 Matching of laser scanned point clouds

The most important step within our processing chain is the assignment of sensed 3D locations of laser scanned points to our feature tracks. Due to the fact, that we acquire precisely time-stamped locations from the outside world, we add only such points from the 3D world, which are “close enough” to points of the detected living feature tracks (with respect to time and location). This type of precise temporal assignment of the incoming data streams is essential for achieving the required accuracy of our method.

3.3 Trajectory extraction

The final step of our method consists in reconstructing the 4D trajectories of moving targets by generating several virtual laser scanner points (VLSP). The VLSP is computed by the inverse projection to the 3D world using the depth data of the laser scanner point together with the 2D image coordinates found within the feature track. In other words, the precise 2D position of the camera image data allows to interpolate a VLSP which is precisely positioned onto the detected 2D feature track for the interpretation in 3D world coordinates.
All feature tracks with more than 2 assigned VLSPs deliver a 4D trajectory of a moving target. These 4D trajectories can finally be visualized as perspective overlay onto the present incoming camera image stream together with further information like present speed, direction and other data. However, they serve mainly as basic data for computing automatic conflict prediction for the own flight path with other moving traffic, a fact which is essential for automatic flight guidance and for unmanned air-vehicles as well.

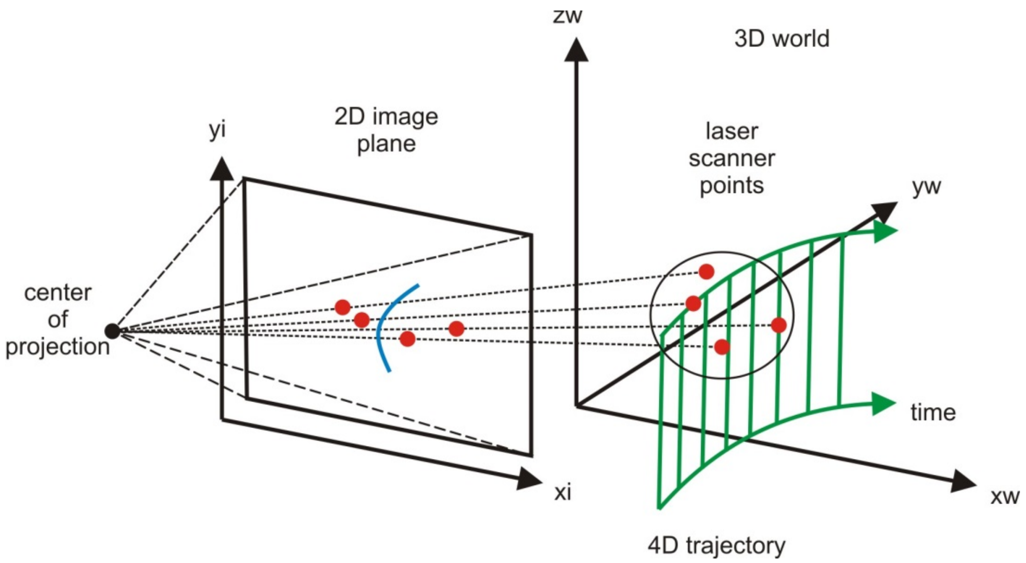


Figure 5: 3D world and 2D image coordinates. Laser scanner data are projected onto the image plane (red). Inverse projection of a feature track (blue) from 2D image plane to 3D world generates a 4D trajectory (green).

4. EXPERIMENTAL RESULTS

4.1 Slow moving objects in a high dynamic scene

First, we wanted to show, that our approach is able to extract quite slowly moving objects within a scene with high positional deviations due to a non-stabilized camera which is fixed to a helicopter. Our helicopter was hovering in front of the DLR's aircraft hangar, while two persons passed the scene. As the results show, the trajectories of the moving persons are detected very well (Figure 6).

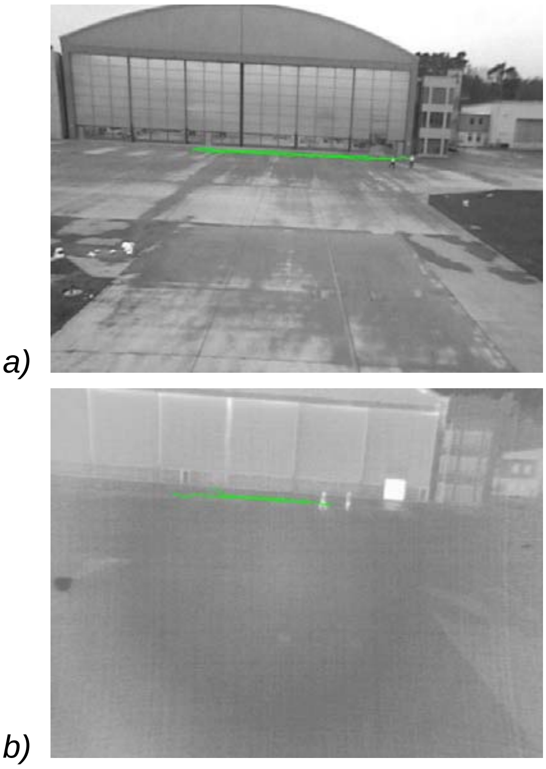


Figure 6: Two persons passing while hovering in front of an aircraft hangar. The extracted 3D trajectory is projected onto the images: a) image from TV stream, b) image from IR stream.

4.2 Fast moving objects

The second type of scenery applies some data of real test flights with DLR's FHS. We flew at low altitude (between 200 and 250 meters AGL) along a road or highway (German Autobahn). With these types of scenes we can demonstrate that our algorithm can separate 3D trajectories of the moving objects in real-time. Moreover, it can also measure the speeds of detected cars and trucks.

Figure 7 shows one image of the laser scanner data stream. The border of the Autobahn can be seen rather well, but it is obvious that the resolution of these data would not be sufficient to detect some moving targets directly.

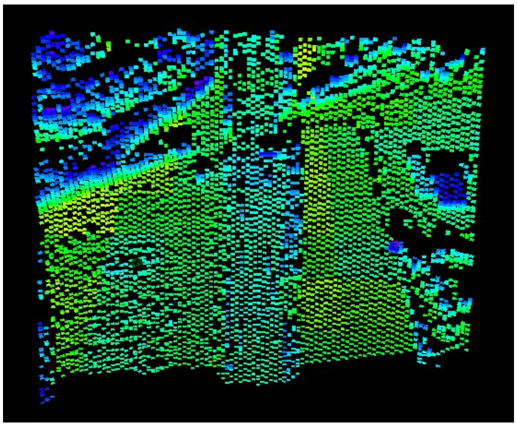


Figure 7: Laser scanner raw data of the "Autobahn" scene Figure 8). It is obvious, that the spatial resolution is not sufficient to reconstruct small moving objects like cars and trucks.

In Figure 8 we show the single steps of the processing chain. To make things better viewable when printed on black and white printers, the background TV/IR images are shown darker in b), c) and d).

Figure 9 shows the extracted 4D trajectories as perspective plot. The three cars on the right side of the autobahn, which are driving in flight direction of our helicopter, are visualized at the bottom of the plot. We use the measured speed values for referencing between Figure 8d) and the plot data. The oncoming truck on the opposite lane (93.7 km/h) on the left side of the road is visualized at the top of the plot. As can be seen the vertical direction (altitude above mean sea level) shows the greatest measuring noise. For this example we got a standard deviation of something in the range between 1 and 3 meters.

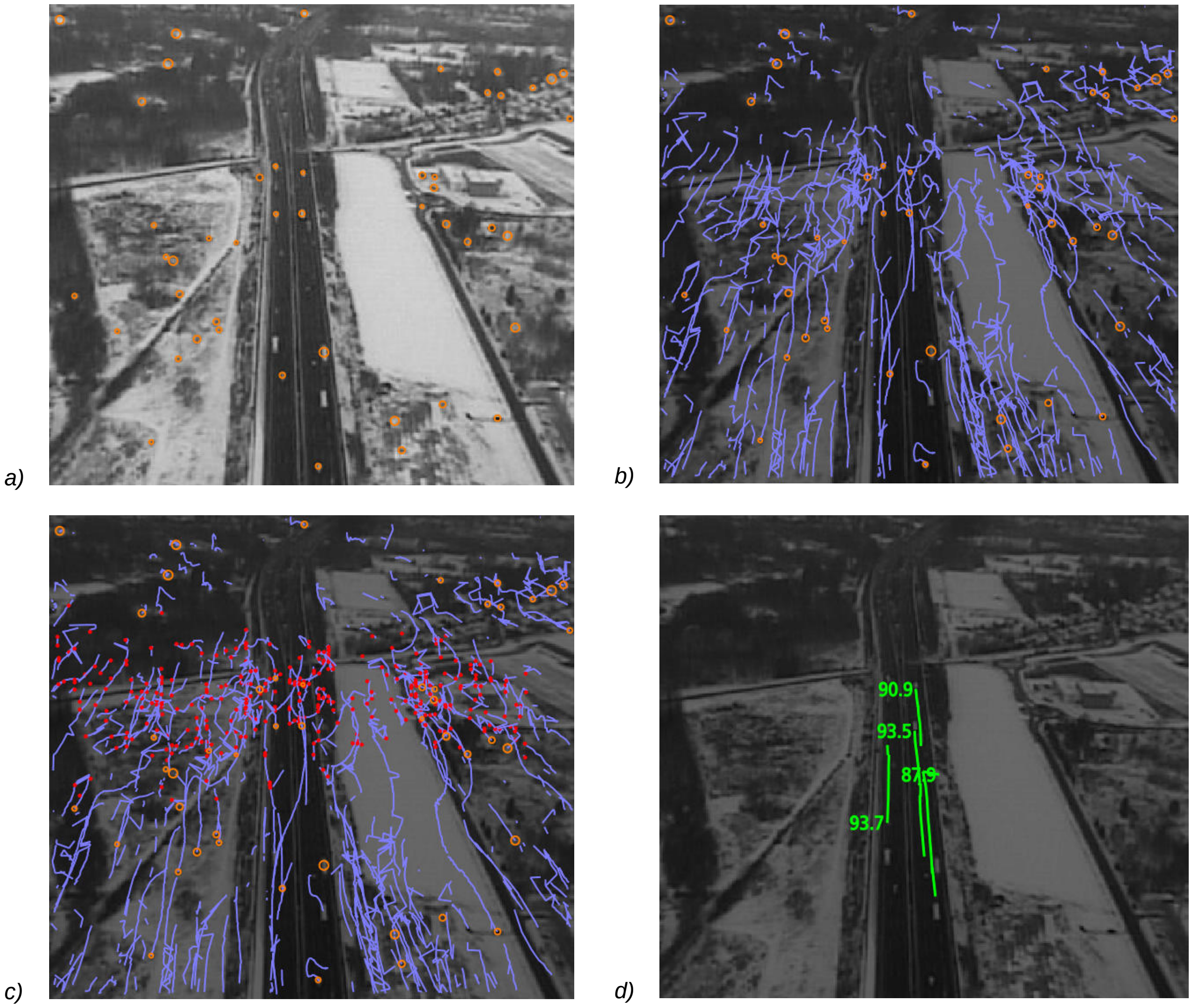


Figure 8: Traffic scene from German Autobahn: a) extracted features (orange circles) in TV-image, b) extracted feature tracks (blue “threads”), c) matched 3D- ladar pixels along image feature tracks (red dots), d) extracted 3D trajectories of moving cars including a speed indication in km/h at the head of each track (green).

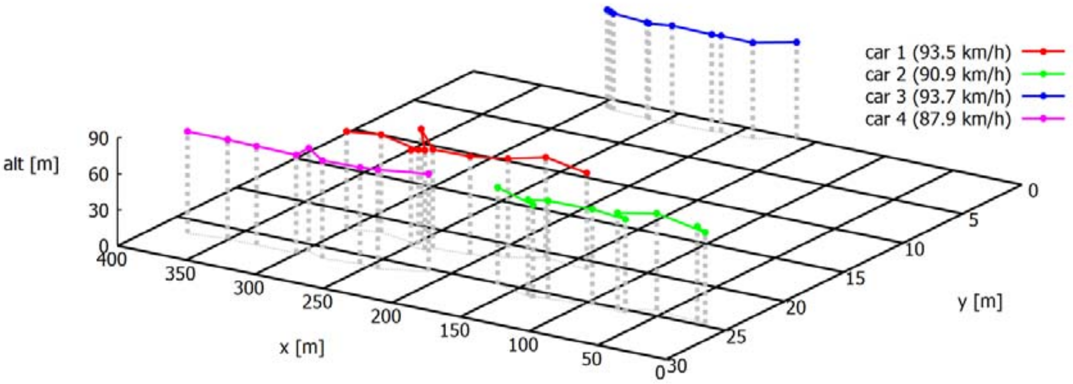


Figure 9: Perspective plot of extracted trajectories. The lower group of trajectories result form three cars on the right side. The upper one trajectory is a car driving in opposite direction.

The next example was generated with our F3S (flexible sensor simulation suite) [10], [11]. Based on 3D models describing the outside world (terrain elevation, terrain texture, buildings, bridges, aircraft) this software toolkit is able to simulate a plurality of fictive sensors like TV, IR, laser scanners and radar. F3S pumps these data into our data acquisition and data processing tool chain. In this example we generated a scene consisting of two static objects in the air (cube and sphere) which appear to come closer and closer due to our own speed (45 m/s). Two other aircraft are flying on the left side of our own track. A Boeing B747 model flies with the same speed and another aircraft (simulated F15 aircraft) is flying with high speed on a curved trajectory. As shown in Figure 10 the trajectory of the F15 aircraft is extracted very well (Figure 10). To make graphic details better viewable when printed on black and white printers, the background input image is shown darker in b), c) and d).

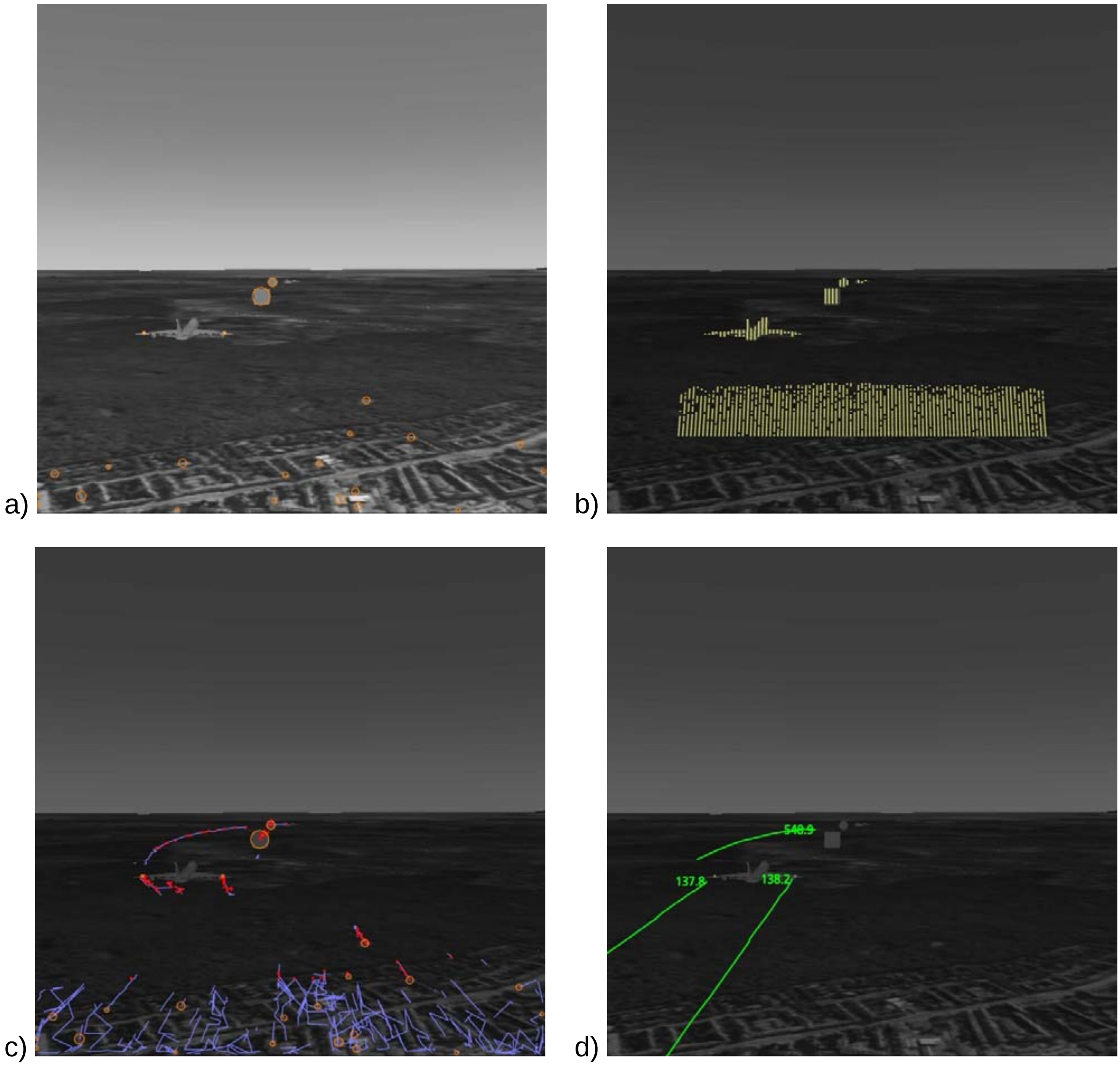


Figure 10: Simulated high speed scene with an overtaking small aircraft (F15): a) extracted features, b) overlaid laser scanner data, c) feature tracks (blue) with assigned laser scanner points (red), c) extracted 4D trajectories and speeds.

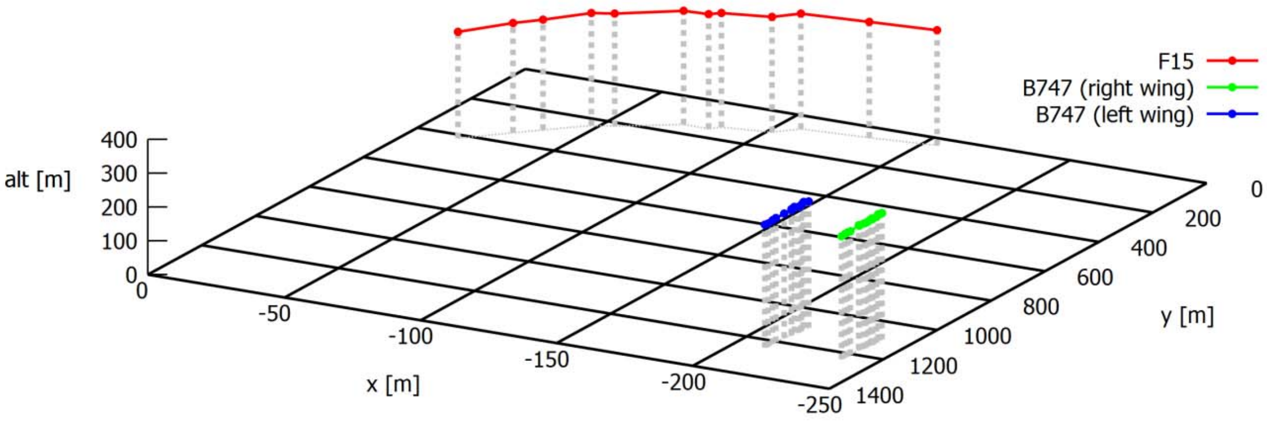


Figure 11: Perspective plot of extracted trajectories. At the right bottom the B747 aircraft, at left top the F15 aircraft. The end of the F15 trajectory is determined by the maximum detection range of the simulated laser scanner, which is 1000 meters (same as HELLAS).

Figure 11 shows a perspective plot of the extracted 4D trajectories. As can be seen the quality of these data is rather good.

5. CONCLUSION

We have presented an accurate method for separating moving objects within image streams of an agile moving camera, such as a helicopter-mounted non-stabilized camera. Our method makes intensive use of data from a laser scanning device. The main advantage of our approach is the combination of rather high resolution (angular and timely) of imaging cameras with the low resolution (spatial and timely) of the laser scanner, which compensate the missing depth dimension in the images. One might propose to detect the discussed moving objects directly within the pixel-clouds of the laser scanner. Regarding the applied HELLAS system, the spatial resolution of the data would probably be too low to detect tiny moving clusters of pixel-clouds. In our opinion the presented approach is the only way to solve this problem without too much computational effort, which would finally not fulfill our real-time requirements.

ACKKNOWLEDGEMENTS

This work was partly sponsored by the German Department of Defense (DOD) within the following projects:

- Increase of all-weather capability by developing a flight management system in connection with the use of a helmet mounted display (HMD)
- ALLFlight - Assisted Low Level Flight and Landing on Unprepared Landing Sites.

We thank Cassidian for modifying the firmware of the HELLAS system, so that we are able to make usage of a very accurate time reference of each detected point. Many thanks go also to MaxViz for providing the IR-camera for the ALLFlight project.

Copyright Statement

REFERENCES

[1] Mählisch, M., et. al., "Sensorfusion Using Spatio-Temporal Aligned Video and Lidar for Improved Vehicle Detection", Intelligent Vehicles Symposium 2006, June 13-15, 2006, Tokyo, Japan (2006)

[2] Joshi, K. A., Thakore, D.G., "A Survey on Moving Object Detection and Tracking in Video Surveillance System", International Journal of Soft Computing and Engineering, Vol. 2 (2012)

[3] Hebel, M., Bers, K., Jäger, K, "Imaging sensor fusion and enhanced vision for helicopter landing operations", In: Proceedings of SPIE. Enhanced and Synthetic Vision 2006, Vol. 6226, (2006)

[4] Vidas, S., et. al., "A mask-based approach for the geometric calibration of thermal-infrared cameras" IEEE Transactions on Instrumentation and Measurement, Vol. 61, pp. 1625-1635 (2012).

[5] Bradski, G., Kaehler, A., "Learning OpenCV – Computer Vision with the OpenCV Library", O'Reilly Media Inc., ISBN: 978-0-596-516113-0, (2008).

[6] -, "OpenCV Reference Manual v2.2", December 2010, (2010).

[7] Canny, J., "A computational approach to edge detection", IEEE Transactions on Pattern Analysis and Machine Intelligence 8, pp. 679-714 (1986).

[8] Shi, J., Tomasi, C, "Good Features to track", 9th IEEE Conference on Computer Vision and Pattern Recognition (1994).

[9] Harris, C, Stephens, M., "A combined corner and edge detector", Proceedings of the 4th Alvey Vision Conference, pp. 147-151, (1988).

[10] Peinecke, N., Döhler, H.-U., Groll, E., "Simulating the Sensor Environment in Real-Time." In: Spring 2011 Flight Simulation Conference. Royal Aeronautical Society. Spring 2011 Flight Simulation Conference, 8-9 Jun 2011, London, UK. (2011)

[11] Döhler, H.-U., Peinecke, N., "An evaluation test bed for enhanced vision". In: Proceedings of SPIE. SPIE Defense, Security and Sensing, 5-9 Apr 2010, Orlando, Florida, USA. (2010)