Real-time Simulation of a 2.5D Radar

Niklas Peinecke, Ernst Groll

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

Abstract
DLR's project ALLFlight (Assisted Low Level Flight and Landing on Unprepared Landing Sites) aims towards assisting a helicopter pilot while landing on adverse, unprepared sites. This includes landing during brown-out or white-out conditions or in unexplored surroundings with no or few data about landing obstacles present. To reach this aim a selection of sensors has been chosen, including a 2.5D radar used for dust and fog penetrating terrain scans. In order to develop and test sensory data fusion and display methods it is necessary to have a ground based simulation available for all sensors that can be used together with a flight simulator. Over the recent years, DLR has developed a suite of computer graphics based, real-time simulators for most of the sensors. We present an extension of the simulator suite to produce simulated 2.5D terrain scanning radar data. Other than more "image-like" sensors, like ladar, a 2.5D radar has a very distinct scanning pattern generated by the complex mechanical movement of the antenna sweep. Therefore, the generated data is not delivered in image frames but rather as a sequence of packages representing measurements along a deformed figure-eight path. First, we generate a cube of 3D measurements inside computer memory making use of graphics hardware acceleration. In a second process the in-memory data is rescanned along a path that resembles the movement of the radar antenna. Using state-of-the-art PC hardware the simulation approach can guarantee data rates comparable to real existing radar devices, thus allowing real-time simulation. This allows to evaluate and test algorithms developed to post-process and fuse the generated data. The implementation is based on the graphics framework OpenGL and makes use of current graphics technology like shaders. We explain implementation details and we discuss how to integrate the simulation module into the existing simulation infrastructure. Finally, we present image sequences and data streams generated by the approach.

1 Introduction

One of the major challenges in the area of helicopter pilot assistance is the landing on adverse, unprepared sites. This includes landing during brown-out or white-out conditions or in unexplored surroundings with no or few data about landing obstacles present. In order to gather data under such circumstances within ALLFlight a broad selection of sensors has been chosen. These include TV and infrared cameras, a 2.5D range measuring ladar system and a 2.5D radar. Originally intended as a obstacle warning radar, this device can also be used for dust and fog penetrating terrain scans and obstacle detection. Thus, among these sensors the 2.5D radar plays a major role.

One of the key aspects within ALLFlight is the integration of all sensors into a comprehensive, easy-to-use pilot assistance system. In order to develop and test data fusion and display methods it is necessary to have ground based simulations for all sensors available that can be used in conjuction with a flight simulator. Over the recent years, DLR has developed a suite of computer graphics based real-time simulators for most of the sensors like millimeter wave radar [14] and lidar/ladar [12, 13]. Within preceding projects and in the initial stages of ALLFlight this suite has already proven useful [17, 16]. Furthermore, existing view simulation software can be utilized to provide images for TV camera and IR camera simulation.

We present an extension of the simulator suite that enables it to produce simulation data like a 2.5D terrain scanning radar. As a base the previous 2D implementation [14] was chosen.

2 Related work

Radar simulation has been an extensive field of studies since the early days of computer hardware. Much of the available work concentrates on the calculation of backscattering diagrams via some numerical approach [9, 4, 15, 18]. Backscattering diagrams provide the integral of the reflected radar energy depending on the view direction for a given object. They are useful mainly for construction purposes, for example, for minimizing the radar cross section of a novel aircraft design. The computation is not yet possible in real time since the radar responses of a given object model has to be collected from various (ideally all) possible view angles.

Furthermore, there have been efforts to generate radar images as they would be observed on a radar screen. There are some non real time applications for such tasks, for example, by De Roo *et al.* [6]. Applications include the simulation of weather radar [10].

2.1 Real time radar viws

A special area of imaging radar simulation is concerned with the real time generation of a radar view of a given scene. Typically, these simulations are integrated in cockpit or tower simulations to give operating staff a realistic impression of the scenario [2]. The generated data can also be used as an input for new EVS algorithms in order to test and evaluate these under almost realistic circumstances. Some research has been carried out in this area (see [1, 5, 3]). Our method was first sketched in [14]. It is based on pioneering work by Doehler and Bollmeyer [7].

2.2 2.5D and 3D simulation

Simulation of 2.5D and 3D radar systems has not yet the same coverage in research literature as more traditional forms of radar. However, over the recent years there have been some approaches to provide such simulations, for example, in order to design proper displays [11].

3 Project ALLFlight

Within the scope of DLR's project ALLFlight a pilot assistance system is being developed which allows an intuitive operation of a manned helicopter from start to landing on unprepared landing sites, see [8]. Low level flights in presence of obstacles and in a degraded visual environment will be realized by the support of different sensor systems (ladar, 2.5D radar, infrared). The objective of this project is to achieve a safe and effective 24 hours all weather operation under low visibility conditions, for example, fog. For this, the pilot is provided with an optimal combination of assistance, consisting of advanced visual and tactile cueing and intelligent control augmentation. Reducing the pilot's workload and increasing his situational and mission awareness are further objectives of this project. The result of this development will be fully automatic flight and landing of a manned helicopter in a confined area.

ALLFlight will also provide curved and unsteady trajectories (in space and time) for all phases of an operational helicopter flight, that is, take-off, low level flight, landing. The trajectory generation incorporates the pilot's cognitive decision processes for trajectory planning in the described scenarios. It is based on a highly accurate digital reproduction of the world surrounding the helicopter, including terrain and obstacle data. For this, a pre-processed terrain data base will be combined with data fusion techniques of real time lidar, radar and infrared data.

The latest handling qualities research will drive the development and maturing of the relevant assistance systems (automation, visual and haptic support) and their coalescence in order to allow the pilot an intuitive following of the generated trajectories or surfaces in dependence of the present flight phase and environment condition.

The integration of all systems and all relevant flight tests will be performed on the FHS (flying helicopter simulator, figure 1) research helicopter.

Figure 1: DLR's EC 135 FHS experimental research helicopter

During the development stage of the project all sensor systems have to be simulated on ground by sensor simulation tools. These tools will ensure that the system does not show any noticeable different behavior, no matter whether the actual sensor or a sensor simulation is connected. Furthermore, data fusion algorithms can be verified before real sensor data recorded during expensive flight tests are available. Therefore sensor simulations are necessary for the success of this project.

4 Radar simulation

The radar device used within ALLFlight is an ICX A130 obstacle warning radar. Consequently, our simulation approach stays as close as possible to the characteristics of this device, although the approach can be generalized to similar radar systems. Central to our approach of simulation are two aspects: First, we aim to provide real-time capabilities in the sense that the simulation can provide scanning data at the same rate as the original sensor. Second, the simulation delivers data in a scan pattern that is identical to the original device. The purpose is to have a simulation that can be used in place of real sensors in order to develop and test algorithms for post-processing and fusing sensor data. Basically, from the viewpoint of computer graphics there are two kinds of sensors: image-like and radar-like.

Image-like sensors produce a sensor image that is much like the result of a camera. Measurements are organized in form of a dot matrix that represents an angle-angle view of the scene from the view point of the sensor. Examples for image-like sensors are optical cameras, infrared cameras and some ladar scanners. One should keep in mind that some advanced sensors provide post-processing capabilities to make the sensor data more accessible to the end user. A ladar sensor, for example, using accurate positional and attitude information could transform its dot matrix image into a sequence of geo-referenced points. However, the underlying data is still the original sensor image.

Radar-like sensors produce a sequence of range-angle measurements. One can interpret these measurements to be points in space described by their azimuth angle, elevation angle and distance range from the sensor. Due to the large vertical beam width of the antenna the elevation angle is lost so that the results are visualized in form of a typical "piece of pie" diagram (PPI scope), see figure 2.

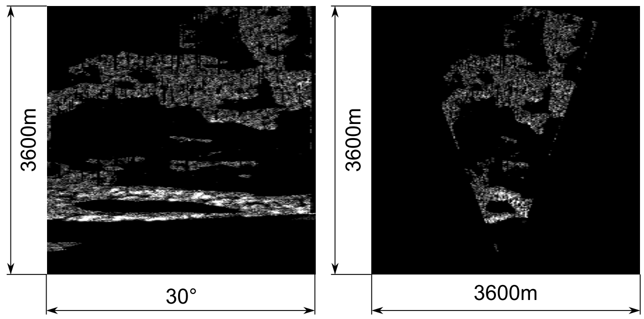

Figure 2: range-angle image, and "piece of pie" range-crossrange image

2.5D radars, sometimes refered to as 3D radars, are a special case where the elevation angle is not lost but recorded with the data. This advantage is bought by a significant slower rate of operation and more moving mechanical parts compared to a traditional 2D radar. A 2.5D radar has a very distinct scanning pattern generated by the complex mechanical movement of the antenna sweep. Therefore, the generated data is not delivered in image frames but rather as a sequence of packages representing measurements along a more or less deformed figure-eight or spiral path.

Simulating radar-like sensors is more difficult on computer graphics hardware since its original purpose is to simulate image-like geometries only.

Early approaches required carrying out crucial stages of the simulation by the computer's CPU [7]. However, since the availability of programable graphics processor pipelines using shaders it is possible to simulate the entire process within the GPU [14]. Using state-of-the-art PC hardware the simulation approach can guarantee data rates comparable to real existing radar devices, thus allowing real-time simulation.

Simulating a 2.5D radar becomes even more complex due to the figure-eight scan path of some systems. Instead of producing a fixed range-angle frame per period one needs to split the simulation in two processes. First, we generate a cube of 3D measurements inside computer memory making use of graphics hardware acceleration. In a second process the in-memory data is rescanned along a path that resembles the movement of the radar antenna. Both processes are executed in parallel making use of modern multi-core CPUs.

4.1 Radar geometry

The typical radar imaging situation is depicted in figure 3.

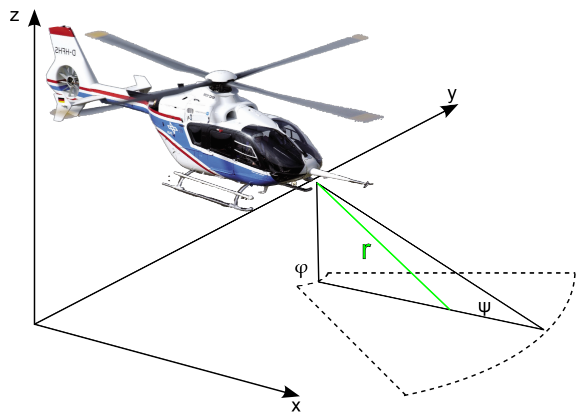

Figure 3: Typical radar imaging situation

The radar system mounted below the helicopter measures object distances by emitting radar pulses and evaluating the respective echoes. Refer to Doehler and Bollmeyer [7] for a more detailed technical description. Here φ is called azimuth angle and ψ is the elevation angle. Note that on a traditional 2D radar screen the scene is projected to an (r, φ)-coordinate system, the *range-angle view*. Thus objects differing by angle ψ only are projected onto the same point and are indistinguishable in the range-angle view. In a 2.5D or 3D radar the elevation angle is not lost and can therefore be utilized for detection purposes. Thus, one "frame" of a 2.5D radar, that is, the results of a complete scan over all possible azimuth and elevation angles, is a three dimensional cube with dimensions azimuth angle, elevation angle and range.

4.2 Implementation

The implementation is based on the graphics framework OpenGL and makes use of contemporary graphics technology like shaders. For shader implementations we use the shader language GLSL.

As of 2010, graphics hardware does not support rendering 3D structures directly to 3D voxel cubes. This would be perfect for radar simulation since one could render the surrounding 3D structures directly to a 3D voxel array (see figure 4). Instead, the voxel cube is constructed interatively by generating each slice, that is, the range angle image for a fixed elevation angle individually.

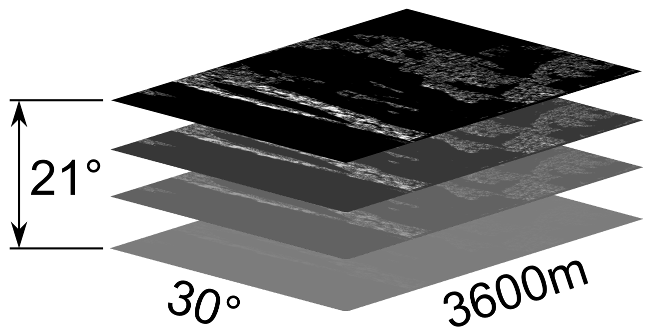

Figure 4: 3D cube of individual range angle slices

For generating the range angle slices we use our previous implementation from [14]:

```
set the field of view in elevation to 1 degree
for each angle PSI in [0 ... 21] do
   generate range angle image for elevation angle PSI
   transfer the result from GPU memory to CPU memory
done
```

The resulting radar cube in memory can then be repacked into data packages directly or be rescanned using the individual scan pattern of the radar device. Each of these two approaches has its individual advantages and problems:

- Sending each generated range angle frame as soon as possible to the client assures that the data is most recent but it is not garanteed that the scan pattern corresponds to the real radar device's.

- Rescanning the cube using the radar device's scan pattern is more realistic but generates a bigger delay between data packages.

Rescanning is carried out by running through the "front plate" of the radar cube, that is, the $21^\circ \times 30^\circ$ array. A series of azimuth-range-angle pairs is generated and for each pair the "beam", that is, the vector of all recorded range values is repacked and sent out. The antenna scan window of the ICX radar-system has an elevation scan height of 21° and the azimuth width is adjustable between 30° and 180°. Its antenna gimbal scans with four beam positions aligned as shown in figure 5.

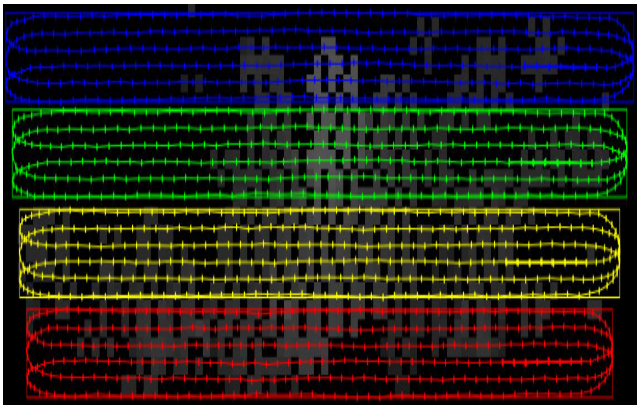

Figure 5: Scan window of ICX radar system with four beams

Figure 6 shows the measured horizontal and vertical beam deflection for beam 1.

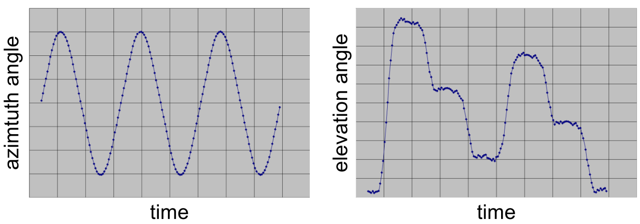

Figure 6: Horizontal and vertical beam deflection

Averaged over time we get the following measured beam elevations in degree:

Beam 0	Beam 1	Beam 2	Beam 3
-10.20	-4.83	0.56	5.76
-5.75	-0.36	5.03	10.24
-7.53	-2.15	3.25	8.45
-9.34	-3.94	1.45	6.65
-6.66	-1.25	4.15	9.34
-8.43	-3.03	2.34	7.58

The change-over time between the particular elevation levels amounts to approximately 120 ms. A complete scan lasts about 1695 ms at an azimuth width of 30°. The distance between different elevation levels is at average 0.9°. In the simulation the deflection of azimuth angle is approximated with a sinus function. The elevation levels are simulated using constant values from the averaged measured real values (see figure 7).

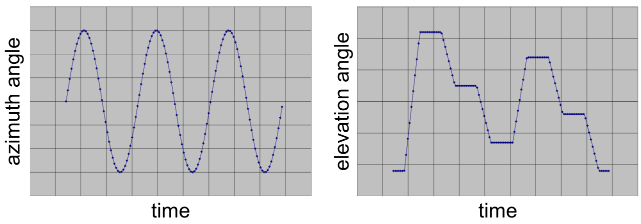

Figure 7: Simulated horizontal and vertical beam deflection

5 Results

Figure 9 shows two sequences of radar slices, the first generated for the scene shown in figure 8, the second for a scene with simulated terrain.

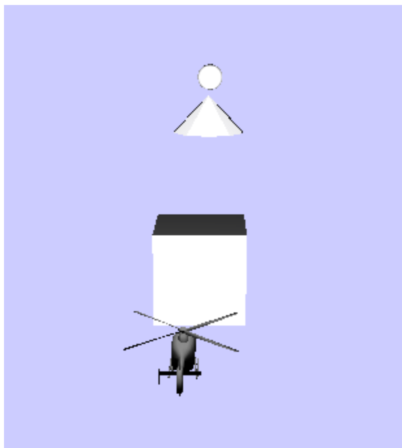

Figure 8: Synthetic test scene with three obstacles

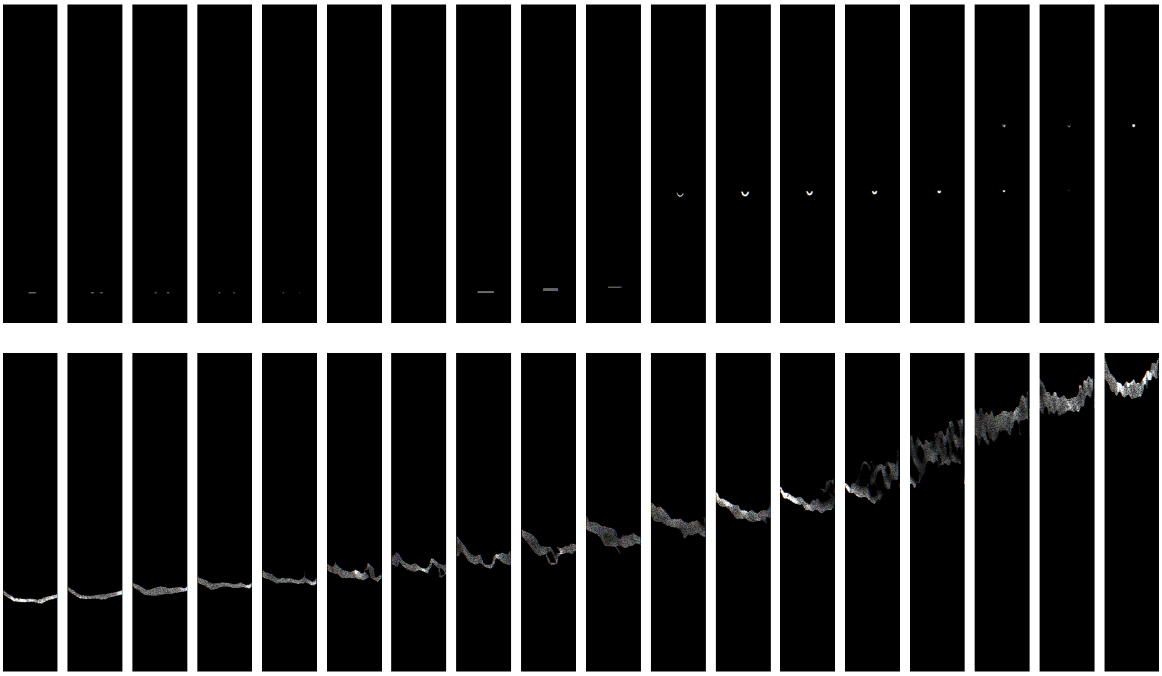

Figure 9: Two sequences of radar slices

In a synthetic scene three obstacles were placed in distances of 200, 800 and 1200 meters ahead of the rotorcraft. The radar simulation was set up for a maximal detection range of 1920 meters. The slices in figure 9 are ordered by elevation angkes from $-10°$ to $+10°$ from the left to right. One can observe that the cube is detected first at lower elevation angles, then the cone and finally the sphere. Since the execution speed mainly depends on the complexity of the scene the method is able to generate individual slices at rates up to 200 Hz on a standard PC. This allows complete 3D scans at a rate of around 5 Hz. However, for a reasonable complex scene rates are closer to 20 Hz, that is, 0.5 Hz for complete scans.

6 Conclusion

We have presented a possibility to extend DLR's existing simulation environment with a simulator module for 2.5D terrain scanning radar data. Being based on a computer graphics hardware implementation approach the simulation allows framerates close to real time operation of a comparable real device. However, for more complex simulation scenes more advanced hardware would be desirable.

References

[1] David J. Allerton and Anthony J. Clare. Real-time modeling of a radar sensor for enhanced vision. In Jacques G. Verly, editor, *Enhanced and Synthetic Vision 2003*, volume 5081/1, pages 39–50. SPIE, 2003.

[2] George L. Bair. Airborne radar simulation, 1996. Camber Corporation, Dallas,Texas,USA; http://www.cambertx.com/pdf/bair-96.pdf.

[3] Timo Balz. Real-time SAR simulation on graphics processing units. In *Proceedings of the 6th European Conference on Synthetic Aperture Radar (EUSAR 2006)*, Dresden, Germany, 2006. VDE, DLR, EADS, FGAN, EADS Astrium.

[4] N. N. Bojarski. A survey of the physical optics inverse scattering identity. *IEEE Trans. On Antennas and Propagation*, 30(5):980–989, 1982.

[5] M. E. Bonjean, F. D. Lapierre, J. Schiefele, and J. G. Verly. Flight simulator with IR and MMW radar image generation capabilities. In Jacques G. Verly and Jeff J. Guell, editors, *Proceedings of the SPIE, Enhanced and Synthetic Vision*, volume 6226, pages 0A.1–0A.13, Jun 2006.

[6] R.D. De Roo, J. Munn, L.E. Pierce, A.Y. Nashashibi, F.T. Ulaby, and G.S. Samples. SPRI: simulator of polarimetric radar images. *IEEE Transactions on Aerospace and Electronic Systems*, 38(1):251–261, 2002.

[7] Hans-Ullrich Doehler and Dirk Bollmeyer. Simulation of imaging radar for obstacle avoidance and enhanced vision. In *SPIE Proceedings*, volume 3088, pages 64–72. SPIE, June 1997.

[8] Hans-Ullrich Doehler, Thomas Lüken, and Robin Lantzsch. ALLFlight - a full scale enhanced and synthetic vision sensor suite for helicopter applications. In *SPIE Defense, Security + Sensing 2009*, volume 7328 of *Proceedings of SPIE*. SPIE press, April 2009.

[9] D. A. McNamara, C. W. I. Pistorius, and J. A. G. Malherbe. *Introduction to the Uniform Geometrical Theory of Diffraction*. Artech House, Boston, Mass, USA, 1990.

[10] D. Meetschen, S. Crewell, P. Grossl, G. Haase, C. Simmer, and A. van Lammeren. Simulation of weather radar products from a mesoscale model. *Phys. Chem. Earth*, 25(10–12):1257–1261, 2000.

[11] Zhang Mo and Han Liu. Simulation of 3D radar display system. *Modern Radar*, 28(2):78–80, 2006. Publication in Chinese.

[12] N. Peinecke, T. Lueken, and B.R. Korn. Lidar simulation using graphics hardware acceleration. In John R. Moore, editor, *Digital Avionics Systems Conference, 2008. DASC 2008. IEEE/AIAA 27th*, pages 4.D.4–1–4.D.4–8, Oct. 2008.

[13] Niklas Peinecke. Simulating airborne sensors using graphics hardware. In *ODAS 2009*. DLR, October 2009.

[14] Niklas Peinecke, Hans-Ullrich Doehler, and Bernd R. Korn. Simulation of imaging radar using graphics hardware acceleration. In *Enhanced and Synthetic Vision*, volume 6957, page 695720, April 2008.

[15] S. M. Rao, D. R. Wilton, and A. W. Glisson. Electromagnetic scattering by surfaces of arbitrary shape. *IEEE Trans. On Antennas and Propagation*, 30(3):409–418, May 1982.

[16] Sven Schmerwitz, Hans-Ullrich Doehler, Niklas Peinecke, and Bernd R. Korn. Stereo radar: Reconstructing 3D data from 2D radar. *Proceedings of the SPIE*, 6957:695703, April 2008.

[17] Sven Schmerwitz, Niklas Peinecke, Hans-Ullrich Döhler, and Bernd Korn. Evaluation of a “stereo” radar approach for terrain reconstruction using synthetic data. In DASC, editor, *27th DASC*, number 225 in 27, 2008.

[18] Yulia V. Zhulina. 3D visualization of radar backscattering diagrams based on OpenGL. *EURASIP Journal on Applied Signal Processing*, 2004(3):358–365, 2004.