INSAR IMAGING GEOMETRY SIMULATION BASED ON COMPUTER GRAPHICS

Fan Zhang*, Lu Bai, Wen Hong

National Key Laboratory of Microwave Imaging Technology,
Institute of Electronics, Chinese Academy of Sciences,
Lab 1, NO. 19 BEISIHUAN XILU P.O.BOX 2702, BEIJING 100190 P. R. China, zhangfan1998@hotmail.com

WG I/5

KEY WORDS: Photogrammetry, Land, Simulation, Computer, Graphics, Interferometer, SAR

ABSTRACT:

To enhance the computational efficiency and precision of classical interferometric SAR(InSAR) imaging geometry simulation, computer graphics method was firstly introduced into imaging geometry simulation, the OpenGL graphics library and Z-Buffer algorithm were used to complete the shadow computation, the pixel operation and graphics transform method were used to densify the sampling points and complete the coordinate computation. Imaging geometry was verified by the process of echo simulation, imaging and interferometric processing, the experimental results showed that the method has the characteristics of high precision and good efficiency.

1. INTRODUCTION

With the development of InSAR (interferometric synthetic aperture radar) technology， the InSAR data processing algorithm and system design face the challenge of precision and efficiency. Due to the technical complexity and economical restrictions of InSAR flight experiment, the computer simulation becomes the chief method of system design and algorithm verification. In the InSAR simulation domain, echo simulation and image simulation are the main ways. The former always simulates echo according to DEM, scattering model, imaging geometry, and then obtains interferogram after imaging and InSAR data processing; the latter simulates image pair according to reference image, DEM, imaging geometry, and then obtains interferogram after InSAR data processing.

Although the two ways have different solutions, the imaging geometry simulation is necessary for them in slant distance calculation. In the interferometric imaging geometry simulation, two problems shall be considered, one is break point which is caused by resampling in slant distance plane, and the other is geometry distortions that are caused by imaging in slant distance plane. To reduce the break points effect, many algorithms were proposed by researchers, for example densified sampling points, iterated interpolation, fractal interpolation and so on. But the densified sampling points method can reduce the simulation efficiency, the interpolation methods can introduce errors by interpolating functions. Then, the classical geometry distortion simulation consumes much time in shadow calculation for point-by-point comparison.

To enhance the computational efficiency and precision of classical InSAR imaging geometry simulation, CG(computer graphics) method was firstly introduced into imaging geometry simulation, the OpenGL graphics library and Z-Buffer algorithm were used to complete the shadow computation, the pixel operation and graphics transform method were used to densify the sampling points and complete the coordinate computation.

2. SIMULATION ALGORITHM BASED ON COMPUTER GRAPHICS

2.1 Shadow Calculation

The hidden surfaces in CG and the shadow in a SAR image are essentially the same; all of them are inaccessible regions due to sight or microwave obstructions. So the hidden surfaces removal in CG will be used to determine the shadow region in SAR simulated scene.

The implementation process includes two steps: the first step is to show the 3D target model on the computer screen through appropriate graphics transformation, the second step is to remove the hidden surfaces by using removal algorithm.

In graphics transformation, the 3D objects are executed a series of geometric transformations and projected into suitable two-dimension graph on the screen. In that way, the two-dimension graph will be similar to the actual observed scene, the precision of hidden surfaces removal can be assured. These geometric transformations include view transformation, model transformation, perspective transformation and viewport transformation. The view transformation describes the position and direction of the observed point. The model transformation describes the position and direction of object, such as zoom, rotation and shift. The perspective transformation describes the relationship between three dimensions coordinate and two dimensions coordinate, and the orthographic is used in the method. The viewport transformation describes the relationship between the device coordinate and window coordinate. The transformation course is express as:

$$\begin{bmatrix} X & 0 & 0 \\ 0 & Y & 0 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} x & 0 & 0 \\ 0 & y & 0 \\ 0 & 0 & z \end{bmatrix} \begin{bmatrix} Ax & 0 & 0 \\ 0 & Ay & 0 \\ 0 & 0 & Az \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (1)$$

where (x, y, z) is the object coordinate, (X, Y) is the window coordinate, (Ax, Ay, Az) is the zoom coefficients and θ is the rotated angle along X axis of window coordinate.

There some hidden-face removal algorithms in computer graphics, such as floating horizon algorithm, roberts algorithm, z-buffer algorithm and so on. The z-buffer algorithm is one of the simplest hidden-face removal algorithms, and supported by the OpenGL. The technique was originally proposed by Catmull and is a simple extension of frame buffer idea. The z-buffer is a separate depth buffer used to store the z coordinate, or depth, of every visible pixel in image space. In use, the depth of a new pixel to be written to the frame buffer is compared to the depth of that pixel stored in the z-buffer. If the comparison indicates that the new pixel is in front of the pixel stored in the frame buffer, then the new pixel is written to the frame buffer and the z-buffer updated with the new z value. If not, no action is taken. In the application of OpenGL, the z-buffer processing is automatically executed by the computer graphic card, which guarantees the simulation precision and efficiency.

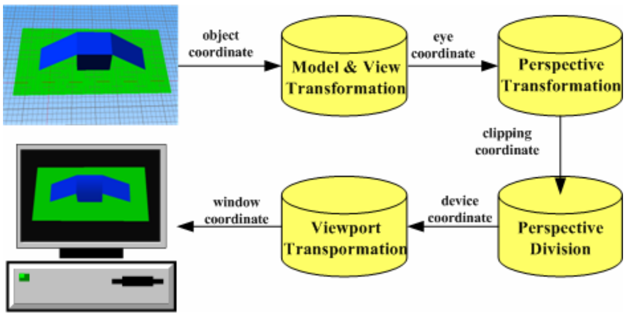

Figure 1: Graphic transformation

2.2 Sampling Point Calculation

The purpose of increasing sampling points is to suppress the break points effect, which is caused by the sampling points lack in the transformation from ground to slant distance plane. The break points effect can bring trouble to phase unwrapping handling. The pixel operation, graphic transformation and inverse transformation are utilized to densify the sampling points.

The first step is to adjust the size of the displayed target on screen for that the pixel numbers of target on screen exceeds the numbers of sampling points in simulation. Only the proper size can guarantee that every ground resolution unit has sufficient sampling points to complete the sampling perfectly in slant distance plane.

The second step is to obtain the three-dimension object coordinate of visible sampling points for slant distance calculation. Through pixel operation in OpenGL, the depth value Z and the screen coordinate (X, Y) can be obtained. The object coordinate (x, y, z) can be deduced according to (X, Y, Z) and the inverse course of implemented graphic transformation.

Finally the 3D target model changes into a heap of dense 3D sampling point array, which will be mapped to slant distance plane by sampling. The sampled point coordinate can be used to calculate the imaging geometry information, which will be used by echo simulation and image simulation. The graphic inverse transformation is expressed as

$$\begin{bmatrix} x & 0 & 0 \\ 0 & y & 0 \\ 0 & 0 & z \end{bmatrix} = \left(\begin{bmatrix} X & 0 & 0 \\ 0 & Y & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & Z \end{bmatrix} \right) \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} 1/Ax & 0 & 0 \\ 0 & 1/Ay & 0 \\ 0 & 0 & 1/Az \end{bmatrix} \quad (2)$$

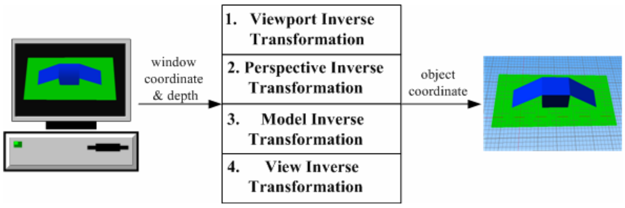

Figure 2: Graphic inverse transformation

3. RESULTS AND ANALYSIS

To verify the geometry simulation precision, a simulation flow that includes geometry simulation, echo simulation, imaging and interferometric processing is designed, and is shown in Figure 3.

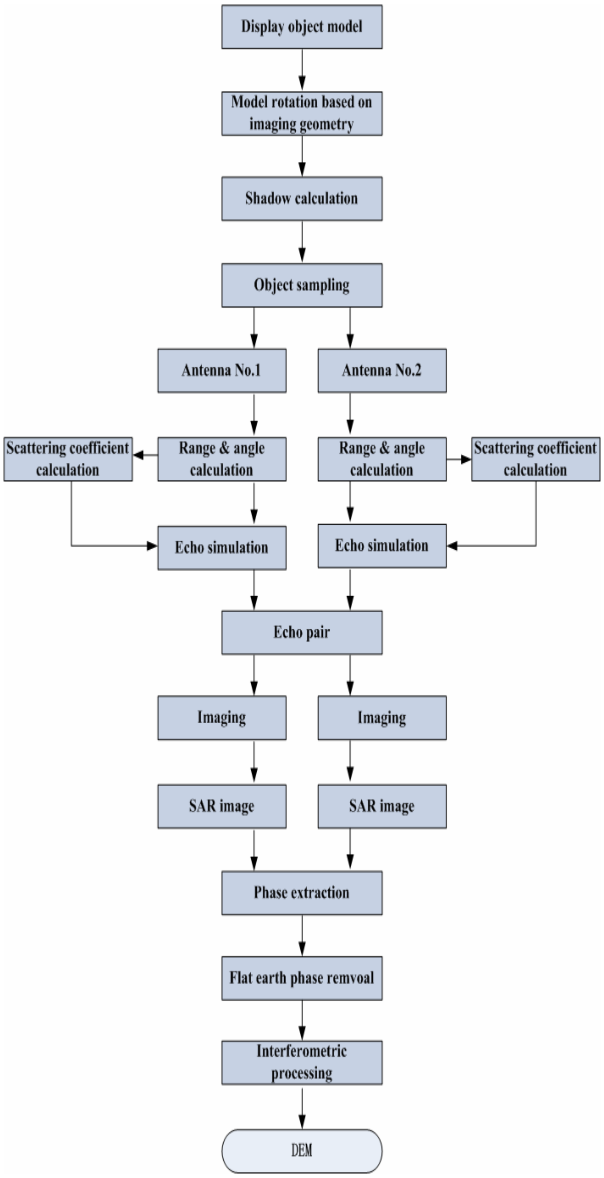

Figure 3: Simulation flow

When the simulation for SAR images is done, interferometric information can be obtained from the complex product of the master and the co-registered slave one. After removing the flat earth phase, the flattened phase is unwrapped using the classical technique proposed by Goldstein. At last, DEM can be estimated from the phase compensated the flat earth phase. The accuracy of the DEM estimation can be used to validate the simulation.

3.1 Interferometric Simulation for Canonical Target

For simplicity, the simulated ideal interferometric phase in flat area is taken as the wrapped flat earth phase. When compensating the flat earth phase, the range information is used so as to estimate the absolute flat earth phase. Finally, the height information can be calculated according to the geometry. All these results are as shown in Fig 4-6.

As shown in Figure 4, a canonical target is selected to verify the InSAR geometry simulation. The interferometric fringes effects in different part are simulated, as shown in Figure 5. The interferometric fringes are dense in frontal slope and sparse in back slope and medium in ground. The foreshortening effect can be found in Figure 5-6. All these effects are the same as the InSAR geometry theory and prove the preciseness of the method.

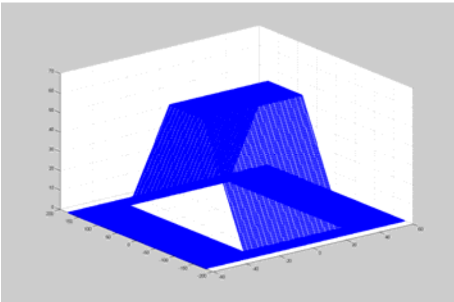

Figure 4: Target points array

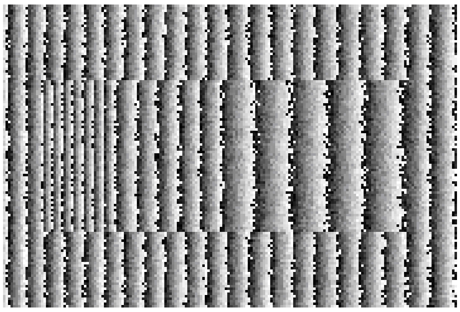

Figure 5: Interferometric fringes

3.2 Analysis on Simulation Precision

To validate the geometry simulation, the DEM calculated from the simulated interferometric phase without noise is compared to the truth. The whole DEM inversion is divided into four steps:
(1) The flat earth phase is removed from the simulated interferomic phase, so that the fringes become flattened. It is good for phase unwrapping.

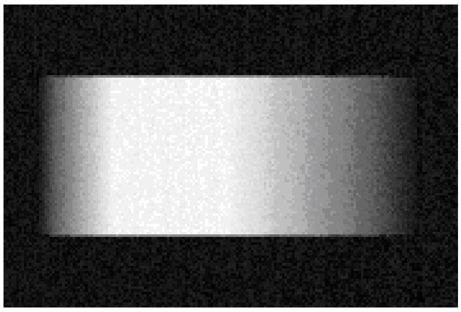

Figure 6: DEM

(2) The flattened phase is unwrapped using the Brunch-cut technique proposed by Goldstein. To avoid the wrong unwrapping route, two brunch-cut are set on the edge in the range direction since there are steep height variations.
(3) The absolute flat earth phase is compensated to the unwrapped phase so that the phase result indicates the range differences directly.
(4) After range differences ΔR is calculated, height information can be obtained from the geometrical relations.

Error analysis is applied on a range profile in the image centre. As shown in Fig. 7-8, the height errors between truth and simulation are less than 3m, which is acceptable for that the sampling gate is 2.5m.

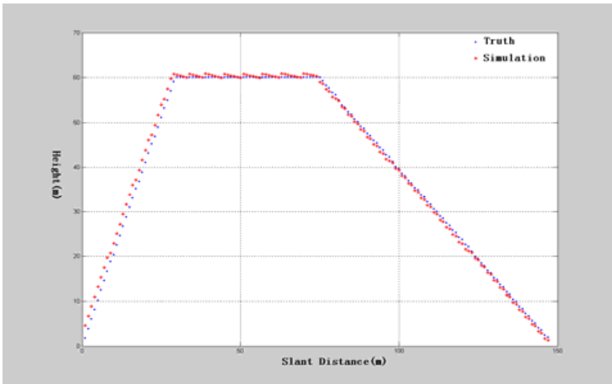

Figure 7: Height information comparison between truth and simulation

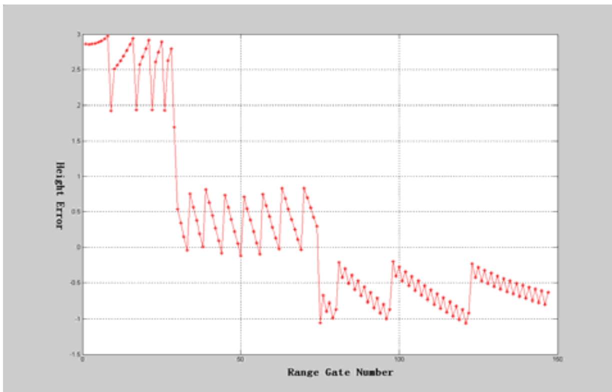

Figure 8: Height errors between truth and simulation

4. CONCLUSIONS

In this paper, a method based on computer graphics is used to simulate the InSAR geometry. The graphic transformation, hidden-face removal algorithm and object coordinate calculation in computer graphics are used to calculate object shadow and densify the sampling points. Compared with traditional shadow calculation method, the method calculates the shadow automatically with computer graphic card. Compared with interpolation method, the method can densify sampling points with graphic transformation in computer graphics. The simulation results show that the method can get high precision DEM data. So this method can be applied in airborne, spaceborne InSAR geometry simulation fields.

REFERENCES

Goldstein R.M., Zebker H.A., Werner C.L., 1988. Satellite radar interferometry: two-dimentional phase unwrapping. Radio Science, 23(4), pp. 713-720.

G.Franceschetti, A.Iodice, 1998. A novel across-track SAR interferometry simulator. IEEE Transaction on Geoscience and Remote Sensing, 36, pp.950-962.

F.D.Vita, et al., 2000. Raw SLC signal simulator for spaceborne interferometric SAR systems. IEEE IGARSS’2000, pp.785-787.

H.Shinohara, T.Homma, et al., 1992. Simulation of interferometric synthetic aperture radar images. IEEE IGARSS’92, pp.1592-1594.

W.Xu, Ian Cumming, 1997. Simulator for repeat-pass satellite InSAR studies. IEEE IGARSS’97, pp.1704-1706.

Zahid Hasan Bawar, Long Teng, Tao Zeng, 2005. Innovative development of spaceborne InSAR simulator. IEEE International Conference on Emerging Technologies’2005, pp.359-364.

Gouinaud C., Pons I., 1996. Use of geometrical SAR simulation for visibility prediction: Application to mission planning and urban study. IEEE IGARSS’96, pp.257-259.

Raney R.K., Runge H., Bamler R., et al., 1994. Precision SAR processing using chirp scaling. IEEE Transaction on Geoscience and Remote Sensing, 32(4), pp.786-799.

David F. Rogers, 2002. Procedural Elements for Computer Graphics, Second Edition. China Machine Press, pp.375-383.

OpenGL Architecture Review Board, et al., 2005. OpenGL Programming Guide, Fourth Edition. Post & Telecom Press, pp.65-105