

A GPU-accelerated (implemented by CUDA) Ortho-rectification Software based on RPC model (CudaOrthRPC)

Tengfei Long¹, Jiao Weili Jiao¹, Guojin He¹

¹ Institute of Remote Sensing and Digital Earth, under the Chinese Academy of Sciences, No.9 Dengzhuang South Road, Haidian District, Beijing 100094, China, longtf@radi.ac.cn

KEY WORDS: open source, Ortho-rectification, RPC model, GPU, L1LS

ABSTRACT: Ortho-rectification is one of the most essential, as well as time-consuming processing for satellite and aerial images. Some commercial softwares, such as PCI Geomatics and ENVI, provide GPU-accelerated ortho-rectification module, but they are expensive. On the other hand, although ortho-rectification is available through many open source applications (e.g. GDAL, OSSIM or Orfeo Toolbox), the existing solutions are not efficient enough. In this paper, an open source ortho-rectification software which is accelerated by GPU device will be introduced. This software is based on RPC (Rational Polynomial Coefficient) model and implemented in C++ and CUDA, thus is called “CudaOrthRPC”. The L1-normalized least squares (L1LS) method is included in the software, which not only makes it possible to directly refine the RPCs with several GCPs, but also makes it possible to robustly estimate the terrain-dependent RPCs with a small amount of GCPs. Consequently, one can use one or several accurate ground control points (GCPs) to refine the vendor-provided RPCs and achieve high geometric precision, or use tens of GCPs to calculate terrain-dependent RPCs without any prior orientation information about the images.

1. INTRODUCTION

Ortho-rectification is one of the most essential processing for satellite and aerial images as the effects of imaging perspective and relief must be corrected before the images can be used for practical applications, e.g., mosaicing into large map, change detection, etc. However, the process of ortho-rectification is very computational expensive for large satellite images. Some commercial softwares, such as PCI Geomatics and ENVI, provide GPU-accelerated ortho-rectification module, but they are expensive. On the other hand, although ortho-rectification is available through many open source applications (e.g. GDAL, OSSIM or Orfeo Toolbox), the existing solutions are not efficient enough. In this paper, an open source ortho-rectification software which is accelerated by GPU device will be introduced. This software is based on RPC (Rational Polynomial Coefficient) model and implemented in C++ and CUDA, thus is called “CudaOrthRPC”. RPC model is a general sensor model which is widely used for high resolution satellite images. One can use one or several accurate ground control points (GCPs) to refine the vendor-provided RPCs and achieve high geometric precision, or use tens of GCPs to calculate terrain-dependent RPCs without any prior orientation information about the images. As far as we know, it is the first open source, general and GPU-accelerated ortho-rectification software.

2. FEATURES OF CudaOrthRPC

The current version of CudaOrthRPC software is 1.0 (a beta version), which includes the following attractive features:

- Bias compensation in image space using affine transformation
- Refine RPCs using L1LS estimation
- Directly compute RPCs from GCPs using L1LS estimation, without an initial RPC model
- Load specific RPC file to get RPC model
- Save the computed or refined RPC model to a specific file
- GPU-accelerated ortho-rectification
- The input image can be most of the image formats supported by GDAL software

Bias compensation in image space using affine transformation is a common approach to correct the vendor-provided RPC model, and it is widely used in practical applications. However, for those satellite images do not contain an initial RPC model, bias compensation approach is not feasible.

L1LS estimation (Long et.al, 2015) is a robust approach to calculate or refine RPCs from a small amount of GCPs, and refining RPCs by L1LS estimation commonly resulting better accuracy than bias compensation approach. L1LS estimation is included in CudaOrthRPC, and it makes RPC model more powerful.

GPU-acceleration make the process of ortho-rectification tens of times faster.

Nevertheless, the current version of CudaOrthRPC software also has some limitations:

- Only WGS-84 coordinate system is supported
- Only Longitude/Latitude projection and UTM projection are supported
- The spatial reference system of DEM data should be WGS-84 coordinate system and Longitude/Latitude projection
- The data type of DEM data must be unsigned integer (16-bit)
- Only cubic interpolation method is implemented
- Only support NVIDIA GPU card

2.1 RPC model

The RPC model is a mathematical model, which describes the object-to-image space transformation. In order to improve the numerical stability of the equations, image coordinates and object coordinates are both normalized to the range of -1.0 to 1.0. The RFM is given as equation (1),

$$\begin{cases} l = \frac{N_l(X, Y, Z)}{D_s(X, Y, Z)} \\ s = \frac{N_s(X, Y, Z)}{D_s(X, Y, Z)} \end{cases} \quad (1)$$

where l and s are normalized coordinates of image points in image space, while X , Y and Z are the normalized coordinates of ground points in object space,

$$N_l(X, Y, Z) = a_0 + a_1Z + a_2Y + a_3X + a_4ZY + a_5ZX + \dots + a_{18}Y^3 + a_{19}X^3,$$

$$D_l(X, Y, Z) = b_0 + b_1Z + b_2Y + b_3X + b_4ZY + b_5ZX + \dots + b_{18}Y^3 + b_{19}X^3,$$

$$N_s(X, Y, Z) = c_0 + c_1Z + c_2Y + c_3X + c_4ZY + c_5ZX + \dots + c_{18}Y^3 + c_{19}X^3,$$

$$D_s(X, Y, Z) = d_0 + d_1Z + d_2Y + d_3X + d_4ZY + d_5ZX + \dots + d_{18}Y^3 + d_{19}X^3,$$

a_i, b_i, c_i, d_i ($i=0,1,\dots,19$) are the RPC parameters.

2.2 Compute RPC model by LILS estimation

Although it is a nonlinear model, the RFM can be transformed into a linear model by a simple deformation, and the deformed linear model is shown as equation (2),

$$\begin{cases} N_l(X, Y, Z) - lD_l(X, Y, Z) = 0 \\ N_s(X, Y, Z) - sD_s(X, Y, Z) = 0 \end{cases} \quad (2)$$

When n GCPs are used to solve the equation (2), the error equations can be written in the vector form as equation (3),

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon} \quad (3)$$

where

$$\mathbf{y} = (l_1, l_2, \dots, l_n, s_1, s_2, \dots, s_n)^T,$$

$$\boldsymbol{\beta} = (a_0, \dots, a_{19}, b_1, \dots, b_{19}, c_0, \dots, c_{19}, d_1, \dots, d_{19})^T,$$

$\boldsymbol{\varepsilon}$ is the error vector,

$$\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)^T,$$

$$\mathbf{x}_i = (1, X_i, Y_i, \dots, Z_i^3, -l_i X_i, -l_i Y_i, \dots, -l_i Z_i^3,$$

$$1, X_i, Y_i, \dots, Z_i^3, -s_i X_i, -s_i Y_i, \dots, -s_i Z_i^3)^T,$$

(l_i, s_i) is the normalized line and sample coordinates of the i th GCP,

(X_i, Y_i, Z_i) is the normalized ground coordinates (longitude, latitude and height) of the i th GCP,

$$i = 1, 2, \dots, n.$$

According to formula (3), solving RPCs is essentially a multiple linear regression problem, and the estimated value of $\boldsymbol{\beta}$ may be obtained by ordinary least square (OLS) method, and it is shown as formula (4)

$$\beta = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y} \quad (4)$$

Due to the strong correlation between the coefficients, $\mathbf{X}^T \mathbf{X}$ is usually ill-posed. In this case, direct inverse of the matrix may not yield accurate and stable results, and some ridge parameter is ordinarily brought in to strengthen the solution. However, ridge estimation still requires numerous GCPs, and the solution may be oscillatory if the observations are insufficient.

On the other hand, by introducing a L1-norm regularizer, the OLS problem as equation (3) becomes a L1-norm regularized least squares (L1LS) as equation (5),

$$\min_{\beta} \|\mathbf{X}\beta - \mathbf{y}\|_2^2 + \lambda \|\beta\|_1 \quad (5)$$

L1LS estimation makes some of the elements in β zeroes, and not only the correlation between the coefficients is alleviated, but also much less GCPs are required to estimate β stably. Consequently, the terrain-dependent RPC model becomes practical with the help of L1LS.

2.3 Bias compensation by affine transformation

If the initial RPC model is available, bias compensation in image space can be utilized to achieve an accurate RPC model, with the help of several GCPs. Affine transformation is the most commonly used compensation model when perform bias compensation in image space, as shown in equation (6),

$$\begin{cases} a_0 + a_1 l + a_2 s = \frac{N_l(X, Y, Z)}{D_s(X, Y, Z)} \\ b_0 + b_1 l + b_2 s = \frac{N_s(X, Y, Z)}{D_s(X, Y, Z)} \end{cases} \quad (6)$$

where $a_0, a_1, a_2, b_0, b_1, b_2$ are affine transformation parameters.

Bias compensation in image space can refine the initial RPC model with a small amount of (or without) GCPs. However, the accuracy of affine transformation in image space is limited as the bias of the initial RPC model may be more complicated than simple affine distortion in image space for some images.

2.4 Refine RPC model by L1LS estimation

Bias compensation in image space can refine the initial RPC model with a small amount of (or without) GCPs. However, the accuracy of affine transformation in image space is limited as the bias of the initial RPC model may be more complicated than simple affine distortion in image space for some images. Actually, some potential biases, such as the possible lens distortions and CCD line distortions, the effect of atmospheric conditions, etc., may not be modeled by simple transformations in image space or ground space. Consequently, direct refinement of the RPC model can be useful to achieve better accuracy than conventional bias compensation in image space or object space. Given an initial RPC model, a correction vector of the RPCs can be calculated from formula (7) or (8) using additional GCPs

$$\mathbf{y} = \mathbf{X}(\beta^0 + \Delta\beta) + \varepsilon \quad (7)$$

$$\mathbf{y}' = \mathbf{X}\Delta\beta + \varepsilon \quad (8)$$

Where

$\beta^0 = [a_0^0, \dots, a_{19}^0, b_1^0, \dots, b_{19}^0, c_0^0, \dots, c_{19}^0, c_1^0, \dots, c_{19}^0]^T$ is the initial value of RPCs,

$\Delta\beta = [a_0^\delta, \dots, a_{19}^\delta, b_1^\delta, \dots, b_{19}^\delta, c_0^\delta, \dots, c_{19}^\delta, c_1^\delta, \dots, c_{19}^\delta]^T$ is the correction vector of RPCs.

By applying the L1-norm regularization, only a small number of GCPs (even a single GCP) are needed to obtain $\Delta\beta$.

3. TECHNIC FLOW OF CudaOrthRPC

3.1 General workflow of CudaOrthRPC

CudaOrthRPC generally includes two processing stages, i.e. RPC model calculation and image ortho-rectification, and Figure 1 shows the general workflow of CudaOrthRPC software.

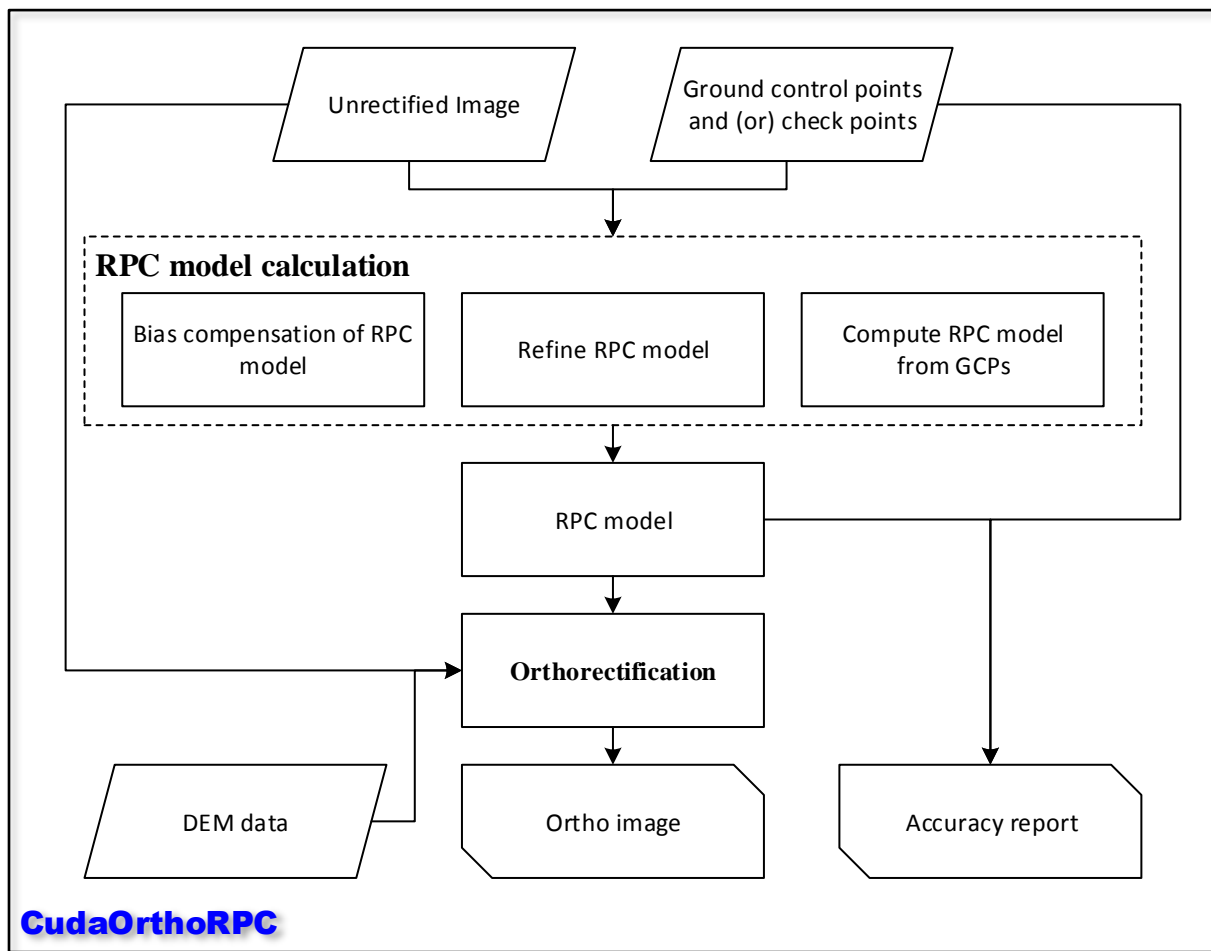


Figure 1 Workflow of CudaOrthRPC software

As shown in Figure 1, the input data includes an unrectified image, GCPs (and check points) and DEM data, while the output data includes the ortho-rectified image and accuracy report. In the step of RPC model calculation, there are three different approaches to achieve an accurate RPC model:

1. Bias compensation in image space using affine transformation
2. Refine RPC model using L1LS estimation
3. Compute RPC using L1LS estimation

If an initial RPC model is available, all of the three approaches can be applied. Bias compensation is the most commonly used approach, and it always yields acceptable results which are not bad. Refining RPC model using L1LS estimation usually produce more accurate results than bias compensation, especially when plenty GCPs are available. Computing RPC using L1LS estimation is not recommended when an initial RPC model is available.

If an initial RPC model is not available, only the third approach can be applied. Note that computing RPC using L1LS estimation required only some well-distributed GCPs but no prior information about the image. Commonly, around 30 well-distributed GCPs are sufficient to stably estimate a terrain-dependent RPC model, but more GCPs usually produce better results.

3.2 Workflow of ortho-rectification module

The process of ortho-rectification is very time consuming, especially when the data volume of satellite image is large. The ortho-rectification module move the most computational steps of ortho-rectification process onto GPU device, i.e. projection converting, ground coordinates to image coordinates converting via RPC model, and interpolation. Figure 2 shows the workflow of ortho-rectification module.

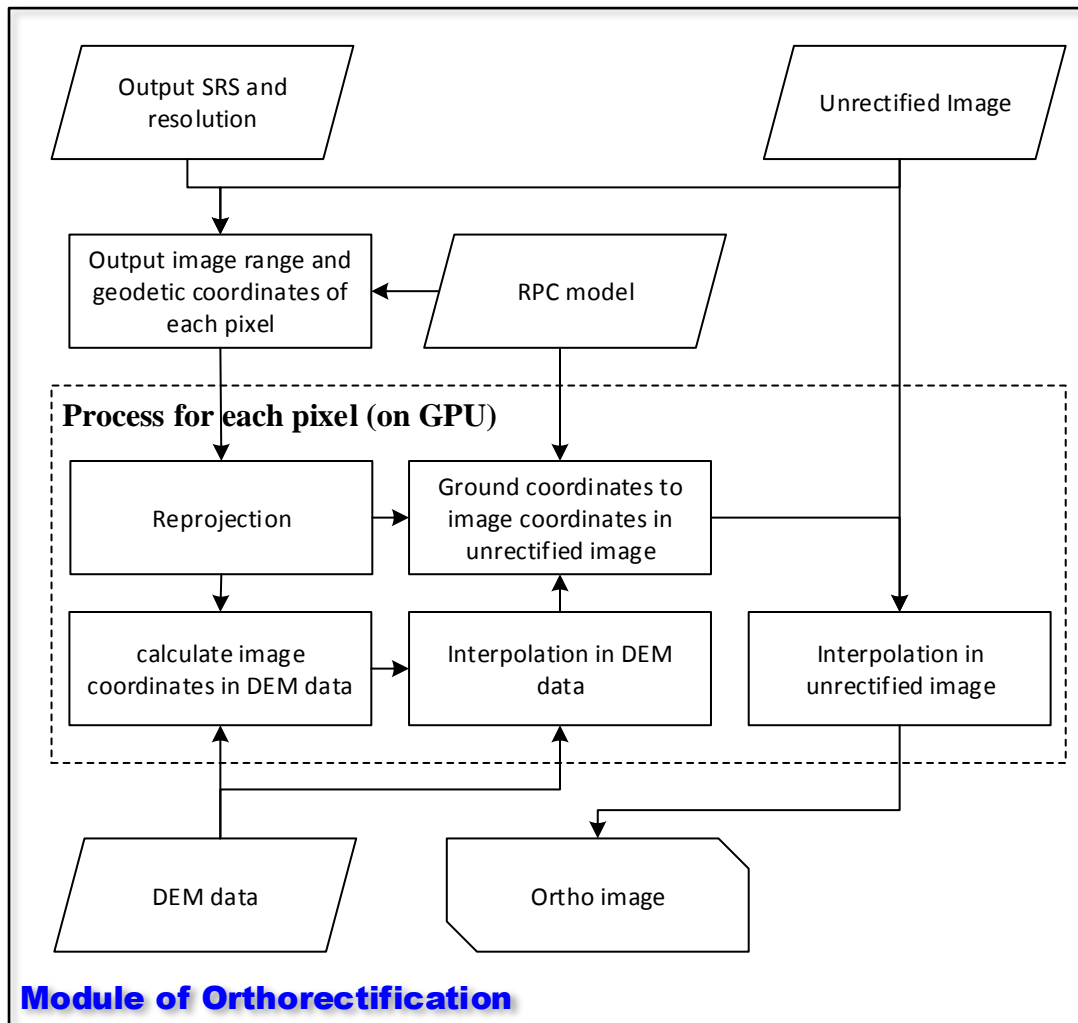


Figure 2 Workflow of Ortho-rectification

As shown in Figure 2, the input data or information include the spatial reference system and spatial resolution of the output image, RPC model, and unrectified image and DEM data. The output data is the ortho-rectified image.

Firstly, the spatial reference system and spatial resolution of the output image, RPC model, and the boundary (or area of interesting) of unrectified image are used to calculate the range of the output image as well as the geodetic coordinates of each pixel of the output image. Then for each pixel, the geodetic coordinates (longitude, latitude, but height) are converted to the image coordinates in DEM data to interpolating the elevation value (height), and the longitude, latitude and height are converted to the image coordinates in the unrectified image via RPC model. Finally, this pixel in the output image is filled with the value interpolated from the unrectified image.

4. EXPERIMENTAL RESULTS

4.1 Accuracy of RPC model calculation

BIAS COMPENSATION V.S. RPC REFINEMENT

Two groups of satellite images are used to evaluate the performance of bias compensation using affine transformation and RPC refinement using L1LS estimation.

Test 1

The test data is a scene of GF-1 (GaoFen-1 of China) MSS image in Fujian, China, whose spatial resolution is 8m

and the range of elevation is 270m~1482m. The ortho-rectified GF-1 PAN image (spatial resolution is 2m) in the same district is used as reference image, and 181 tie points are automatically collected. Different numbers of tie points (1~80) are selected from all the 181 tie points as GCPs, while the rest are used as check points. For each pair of GCPs and check points, the two approaches are applied to correct the vendor-provided RPC model, and the root mean square errors of the check points are noted down, as shown in Figure 3. Note that the initial RPC model of GF-1 MSS image is provided by the vendor.

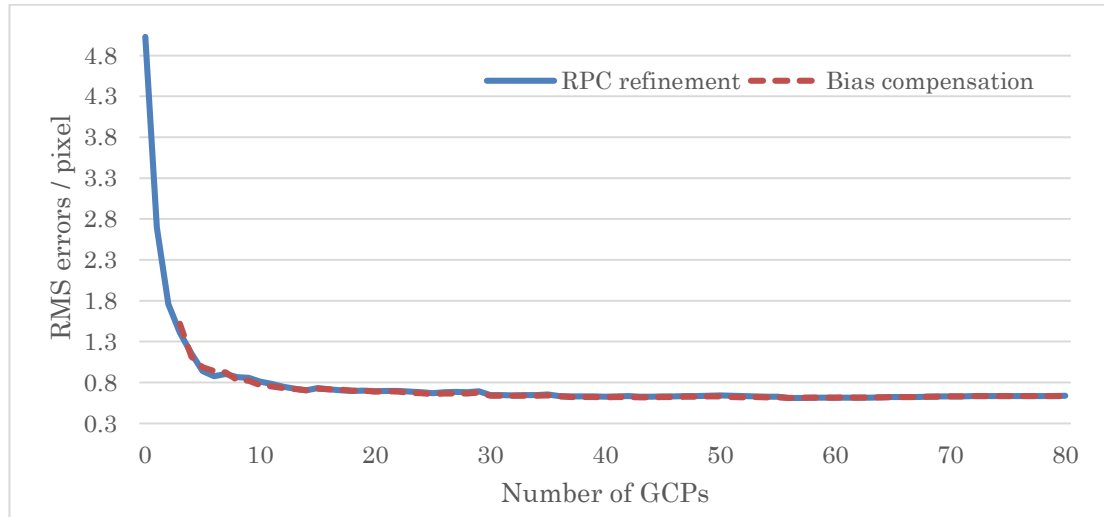


Figure 3 Accuracy report of GF-1 MSS image using two approaches with different number of GCPs

Test 2

The test data is a scene of CBERS-4 MUX image in Liaoning, China, whose spatial resolution is 20m and the range of elevation is 8m~592m. The ortho-rectified Landsat-8 image (spatial resolution is 15m) in the same district is used as reference image, and 456 tie points are automatically collected. Different numbers of tie points (1~80) are selected from all the 456 tie points as GCPs, while the rest are used as check points. For each pair of GCPs and check points, the two approaches are applied to correct the vendor-provided RPC model, and the root mean square errors of the check points are noted down, as shown in Figure 4. Note that the initial RPC model of CBERS-4 MUX is generated from the rigorous sensor model via terrain-independent approach.

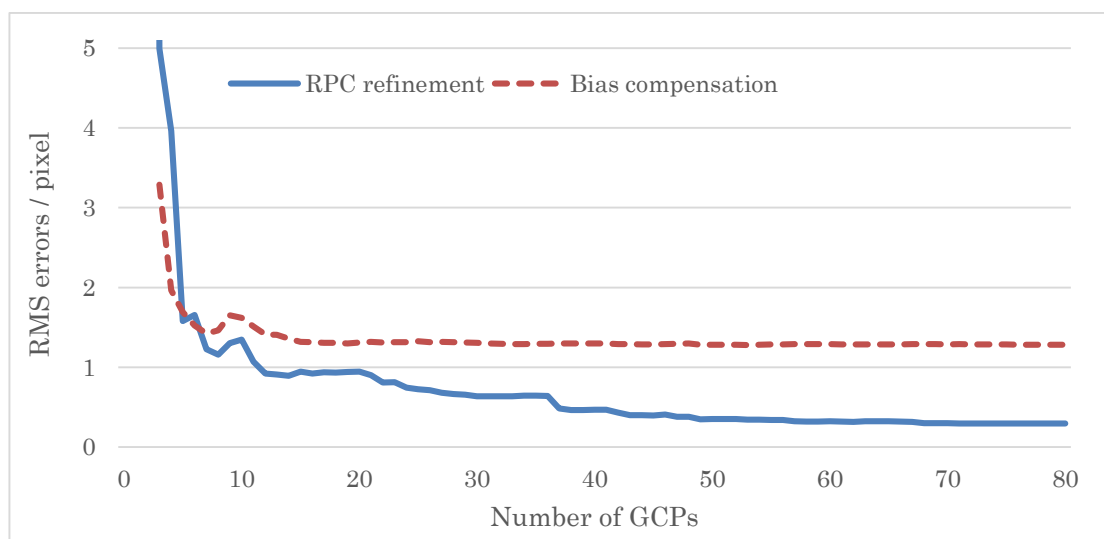


Figure 4 Accuracy report of CBERS-4 MUX image using two approaches with different number of GCPs

From Figure 3 and Figure 4, one can see that the two approaches have similar performance for the data set of GF-1

MSS image, while RPC refinement has much better performance than bias compensation when more than 10 GCPs are used. The reason is that the rigorous sensor model of CBERS-4 MSS is not accurate enough (actually, the attitude data for the CBERS-4 sensor is not available), and the error of the generated RPC model is much more complicated than affine transformation.

COMPUTE RPC MODEL USING L1LS ESTIMATION

The test data is a scene of HJ-1-B (Chinese Huanjing satellite) image in Qinghai, China, whose spatial resolution is 30m and the range of elevation is about 958m~4158m. This image is a product of level 2, and neither the rigorous sensor model nor the initial RPC model is available, thus we have to directly compute the terrain-dependent RPC model from GCPs. The reference image is the ortho-rectified Landsat-5 images in the same district, and 178 tie points are automatically collected. Then, 100, 80, 60, 40, 30, 20 and 10 tie points are evenly selected as GCPs, and the remaining (78, 98, 118, 138, 148, 158 and 168) tie points are used as check-points. Two different approach, i.e. ridge estimation and L1-norm-regularized least squares (L1LS) estimation, are used to compute RPCs, respectively. The accuracy reports are shown as Table 1, containing the root-mean-square (RMS) errors of GCPs and check points.

Table 1 Accuracy report of HJ-1-B image using two approaches with different numbers of GCPs

Approach	GCP	Check point	GCP RMS error / pixel			Check point RMS error / pixel		
	Num	Num	sample	line	plan	sample	line	plan
Ridge	100	78	0.38	0.65	0.75	0.5	0.56	0.75
L1LS			0.44	0.72	0.84	0.61	0.65	0.89
Ridge	80	98	0.35	0.64	0.73	0.65	0.79	1.02
L1LS			0.47	0.79	0.92	0.57	0.76	0.95
Ridge	60	118	0.28	0.61	0.67	0.94	0.84	1.26
L1LS			0.46	0.81	0.93	0.56	0.87	1.03
Ridge	40	138	0.2	0.33	0.38	2.02	2.09	2.9
L1LS			0.44	0.79	0.9	0.62	1.02	1.19
Ridge	30	148	0.04	0.13	0.13	2.79	4.04	4.91
L1LS			0.36	0.64	0.73	0.65	1.19	1.36
Ridge	20	158	0.01	0.01	0.01	58.28	42.1	71.89
L1LS			0.3	0.55	0.63	0.99	1.14	1.51
Ridge	10	168	0	0	0	516.59	462.97	693.69
L1LS			0.11	0.23	0.26	1.73	2.57	3.09

According to Table 1, one can see that:

- When the number of GCPs is much greater than 39 (80-GCP case), ridge estimation provides reliable results, similar as the results of L1LS estimation.
- When the number of GCPs is not much greater than 39 (40-GCP case), however, ridge estimation performs badly at check points. Despite the high fitting accuracy at GCPs, the estimated model can be oscillatory between exact-fit values.
- L1LS performs robustly when different numbers of GCPs are used. Even in the 10-GCP case, this method provides practical results (the maximum residuals of ICPs are less than 2 pixels). In this sense, it is possible to perform accurate ortho-rectification without knowing the rigorous sensor model or the initial RPC model using a small amount of GCPs.

4.2 Efficiency of CudaOrthRPC

The following are the major hardware on which the experiments are performed:

CPU: Intel(R) Xeon(R) W3550 3.07GHz, memory: 6G

GPU: Quadro 2000, display memory: 1G

COMPARING WITH CPU VERSION

A scene of Landsat-5 image and a scene of Spot-5 PAN image are used to perform the comparison tests. Three

different approaches, including rigorous sensor model (RSM) in CPU, RPC model in CPU and RPC model in GPU, are used, and the results are shown in Table 2.

Table 2 Comparison of CPU version and GPU version using Landsat-5 and Spot-5 images.

Data	Bands	Image size (pixel×pixel)	Data amount (MB)	Time (sec.)		
				RSM CPU	RPC CPU	RPC GPU
Landsat-5	7	6871×5733	260	101.34	34.59	2.13
Spot-5 PAN	1	24000×24000	562	1853.23	103.37	9.24

From Table 2, one can see that rigorous sensor model in CPU is the most time consuming, and RPC model can sharply reduce the computation, especially for Spot-5 PAN image. In addition, GPU-acceleration further shortens the processing time.

COMPARING WITH COMMERCIAL SOFTWARE-PCI GEOMATICS

Commercial software PCI Geomatics 2013 and CudaOrthRPC are both used to ortho-rectify, and several Chinese satellite images, including GF-1, GF-2, HJ-1 and CBERS-4, are used as test data. The information of the test data is shown in Table 3, and the test results are shown in Figure 5.

Table 3 The information about the test data

Test ID	Data	Bands	Image size (pixel×pixel)	Data amount (MB)
1	GF-1WFV	4	12000×13400	1256
2	GF-1PAN	1	18192×18000	624
3	GF-1 MSS	4	7300×6908	384
4	GF-2 PAN	1	29200×27619	1576
5	HJ-1	3	12000×12000	411
6	CBERS-4 P5	1	12000×12000	140
7	CBERS-4 P10	3	6000×6000	103
8	CBERS-4 MUX	3	6000×6000	103

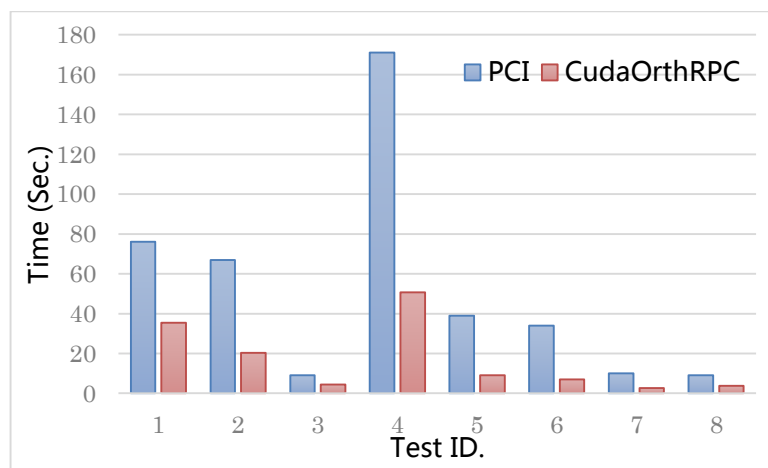


Figure 5 Time consuming comparison of ortho-rectification using PCI and CudaOrthRPC

Figure 5 shows that CudaOrthRPC outperforms PCI Geomatics 2013 in all the cases. Note that PCI geomatics automatically builds image pyramid after the ortho-rectification process finishes. Building image pyramid may increase the consumed time of PCI, but the increased time is much less than which is spent by the ortho-rectification process.

5. CONCLUSIONS

This paper introduced a GPU-accelerated ortho-rectification software based on RPC model (CudaOrthRPC). This software provides a robust approach to compute terrain-dependent RPC model from limited numbers of GCPs without any prior information about the image. When the initial RPC model is available, both bias compensation in image space using affine transformation and direct RPCs refinement using L1LS estimation can be applied with the help of a small amount of GCPs, yielding accurate RPC model. Thanks to the computing capability of GPU device, CudaOrthRPC is able to finish the ortho-rectification process of large remotely sensed images in tens of, or even several seconds.

REFERENCES

- Fraser, C. S., Hanley, H. B., 2003. Bias Compensation in Rational Functions for IKONOS Satellite Imagery. *Photogrammetric Engineering & Remote Sensing*, 69 (1), 53–57.
- Long, T., W. Jiao, and G. He. 2015. “RPC Estimation via L1-Norm-Regularized Least Squares (L1LS).” *IEEE Transactions on Geoscience and Remote Sensing* 53 (8): 4554–4567.
- Tao, C., Hu, Y., 2001. A Comprehensive Study on the Rational Function Model for Photogrammetric Processing. *Photogrammetric Engineering & Remote Sensing*, 67 (12), 1347–1357.