Computers and Electronics in Agriculture 85 (2012) 123–133

Contents lists available at SciVerse ScienceDirect

Computers and Electronics in Agriculture

journal homepage: www.elsevier.com/locate/compag

Acceleration of CT reconstruction for wheat tiller inspection based on adaptive minimum enclosing rectangle

Ni Jiang, Wanneng Yang, Lingfeng Duan, Xiaochun Xu, Chenglong Huang, Qian Liu *

Britton Chance Center for Biomedical Photonics, Wuhan National Laboratory for Optoelectronics-Huazhong University of Science and Technology, 1037 Luoyu Rd., Wuhan 430074, PR China
Key Laboratory of Biomedical Photonics of Ministry of Education, College of Life Science and Technology, Huazhong University of Science and Technology, 1037 Luoyu Rd., Wuhan 430074, PR China

ARTICLE INFO

Article history:
Received 30 June 2011
Received in revised form 11 April 2012
Accepted 13 April 2012

Keywords:
X-ray computed tomography
Adaptive minimum enclosing rectangle
Graphics Processing Unit
Wheat tillers
Real-time inspection

ABSTRACT

Tiller number is highly correlated with grain yield in wheat. Traditional observation of wheat tiller number is still manual. Previously, our group developed a high-throughput system for measuring automatically rice tillers (H-SMART) based on X-ray computed tomography (CT), providing high accuracy for measuring rice tillers. However, the time-consuming reconstruction, which is necessary to generate tomographic images, limits the throughput improvement of system as well as the CT potential for the real-time applications. In order to accelerate the reconstruction process, we present an adaptive minimum enclosing rectangle (AMER) method to reduce the number of reconstructed pixels from the full field of view (FOV) and apply parallel processing using Graphics Processing Unit (GPU). The reconstruction time and speedup with different methods were discussed. Compared to the AMER method, GPU technique improved reconstruction with a higher speedup of approximately 200 times. And the speedup with AMER method was determined by two factors: area ratio of AMER and FOV, and the longest distance between the vertices of the AMER and the rotation center. Besides reconstruction, tiller identification could also be accelerated by AMER. Moreover, the tiller measurement accuracy did not decrease. With the combination of AMER and GPU, the entire tiller inspection time for a pot-grown plant was reduced from about 11870 ms to less than 200 ms. In sum, the optimized method met the requirement of real-time imaging and expanded CT application in plant phenomics and agriculture photonics.

1. Introduction

Wheat is a main world food crop (Li et al., 2000), which supplies one-third of the world's population (Dhanda et al., 2004). The spikes or panicle number is dependent upon the number of tiller per plant (Kuraparthy et al., 2007), while, the number of spike-bearing tillers is highly correlated with grain yield in wheat (Mass et al., 1994; Sharma, 1995). Hence, tiller number, which is an important agronomic character, needs to be observed for wheat research. Traditional observation of wheat tiller number is still manual, which is subjective, labor-intensive and low-reproducibility. In addition, the traditional method cannot satisfy the requirement for accuracy and efficiency when a large number of plants need to be evaluated. Therefore, a high-throughput automatic method is necessary for measuring the tiller number. However, tillers usually overlap in natural condition, which makes it difficult to detect the inner tillers even using multi-angle visible light imaging.

X-ray computed tomography (CT) is a very useful technique in agriculture engineering (Singh et al., 2010; Cubero et al., 2011). It has been applied to analysis pore topology inside grain bulks (Neethirajan et al., 2008), fruit internal quality (Barcelon et al., 1999), internal log properties (Longuetaud et al., 2004; Yu and Qi, 2008), plant structure (Heeraman et al., 1997; Stuppy et al., 2003; Dhondt et al., 2010). In virtue of the penetrability of X-ray, the internal structures of objects can be revealed. Due to the X-ray attenuation within tillers, all tillers can be seen in the transverse section image of the wheat culms, and the tiller number can be determined through image analysis. Nevertheless, the generation of section image needs to scan the objects at hundreds of different angles, and the reconstruction takes very long computation time. So, the application of CT for real-time imaging is limited due to its low speed. It is imperative that the images are computed fast in many applications (Herman, 1995).

Previously, our group developed a high-throughput system for measuring automatically rice tillers (H-SMART) equipped with CT imaging (Yang et al., 2011b). In order to extract more plant phenotyping parameters, the H-SMART would be incorporated with other imaging technologies, such as visible light imaging,

* Corresponding author. Tel.: +86 27 8779 2033; fax: +86 27 8779 2034.
E-mail address: qianliu@mail.hust.edu.cn (Q. Liu).

0168-1699/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.compag.2012.04.004

near-infrared imaging, and thermal infrared imaging. The throughput of H-SMART would be reduced since it will need extra time for these image processing mentioned above. To solve this problem and expand the real-time application of CT, time consumption of CT image processing should be shortened.

Besides the data acquisition speed of CT system, the reconstruction speed is a critical factor which limits the CT to be used in real-time applications. Analytical reconstruction algorithms and iterative algorithms are well-known traditional reconstruction algorithms but their speed performance is not very good especially when there are extensive data to be computed. Some methods such as focus of attention (Benson and Gregor, 2006), air skipping (Lee et al., 2010) and adaptive region of interest (Yang et al., 2011a) have been developed to reduce the amount of data to be reconstructed. Although focus of attention algorithm can avoid a majority of redundant data to be reconstructed, the method contains one time-consuming step of backprojecting all projections. Classification and polygon clipping in air skipping technique is computation-intensive and thus degrades its performance. The region of interest method takes little time, but the effective sinogram region cannot be extracted precisely and may contain redundant pixels. Therefore, it is necessary to develop a new method which can reduce computed data as much as possible at a low time-consumption.

On the other hand, many acceleration techniques based on hardware have been introduced in CT applications. Using the Graphics Processing Unit (GPU) is shown to accelerate the CT reconstruction with a good performance (Xu and Mueller, 2007; Noël et al., 2010; Okitsu et al., 2010). Acceleration via field programmable gate array (Gac et al., 2006) or cell broadband engine (Kachelrieß and Knaup, 2007) is also used for CT reconstruction. The speed of memory operation in GPU is superior to the operation in Central Processing Unit (CPU) and GPU is equipped with the parallel architecture, thus GPU is well suited for CT reconstruction which contains mass of repeated processing. In addition, the NVIDIA common unified device architecture (CUDA) technology simplifies the GPU programming. Compared with other hardware, GPU is popular because of its low-cost, flexibility and programmability.

In this paper, we present a GPU-based CT reconstruction algorithm using an adaptive minimum enclosing rectangle (AMER) method to accelerate CT reconstruction. Moreover, the present method is applied to improve the performance of H-SMART. The objectives of this work were: (1) to devise a fast and effective AMER method to reduce the amount of computation; (2) using GPU parallel computing to accelerate the CT reconstruction; and (3) to test the present method by incorporating it into the H-SMART and analyze the performance. The rest of the paper is organized as follows. Section 2 describes the CT system, filtered back-projection (FBP) algorithm, and the present method combined AMER technique and GPU technique. Section 3 shows the results and demonstrates the effectiveness of the present method. Section 4 is the conclusion.

2. Materials and methods

2.1. The CT system

The CT imaging system of H-SMART consisted of a commercial fan-beam X-ray source (T80-1-60, BMEI Co. Ltd., China), a linear-array X-ray detector (768 detector elements, x scan 0.4f3-205, BMEI Co. Ltd., China) and a rotating platform. The pot-grown plants were delivered to the CT imaging system by an industrial conveyor. When a pot-grown plant reached the rotating platform, it rotated with the platform driven by a servo motor to get the projection data of different angles (step angle 0.85°, covering 336 orientations). The pixel pitch of the detector was 0.4 mm, and the magnification of the system was 1.26. Hence, the pixel size of the section image was approximately 0.32 × 0.32 mm. A computer workstation (HP xw6400, Hewlett–Packard Development Company, USA) controlled the process of image acquisition and completed data processing. Details about the H-SMART system were listed in Yang et al. (2011b).

2.2. FBP reconstruction algorithm

Since FBP reconstruction algorithm was much faster than other CT reconstruction algorithms, it was used for CT reconstruction of crop tiller. FBP algorithm (Kak and Slaney, 1988) consisted of two stages: the preweighted convolution filtering stage and the back-projection stage. Let $P(\beta,a)$ be the projection data at angle β, the preweighted convolution filtering stage was represented by the following equation:

$$\widetilde{P}(\beta,a) = \left(P(\beta,a)\frac{R}{\sqrt{R+a^2}}\right) * h(a) \tag{1}$$

where R was the distance between the X-ray source and the rotation center, a was the coordinate of projection on the detector and $h(a)$ was a convolution filter.

Let $f(x,y)$ be the reconstructed image and was computed through back projection as below:

$$f(x,y) = \int_0^{2\pi} \frac{R^2}{U(x,y,\beta)^2}\widetilde{P}[\beta, a(x,y,\beta)]d\beta \tag{2}$$

where $U(x,y,\beta)$ was the weighted value given by:

$$U(x,y,\beta) = R + x\cos\beta + y\sin\beta \tag{3}$$

And the coordinate a was computed as:

$$a(x,y,\beta) = R\frac{-x\sin\beta + y\cos\beta}{R + x\cos\beta + y\sin\beta} \tag{4}$$

2.3. Image analysis

Assuming the number of projection angles was M, and the size of the reconstruction image was $N \times N$, the FBP reconstruction had the computational complexity of $O(MN^2)$. Thus, the reconstruction time can be reduced by reducing N. Since the preweighted convolution filtering and the backprojection of each pixel were independent in FBP algorithm, the use of GPU for parallel computation was very effective to accelerate the reconstruction process. Here we proposed a method that can identify the AMER region to be reconstructed, load projection data into the GPU memory, and compute the pixel value in parallel.

The overall process of tiller inspection is shown in Fig. 1. In order to remove the noise in the image, the projection was normalized using the Eq. (5) (Yang et al., 2011b):

$$I_N = \log\frac{I - I_D}{I_0 - I_D} \tag{5}$$

where I was the original projection data acquired from the linear X-ray detector; I_0 and I_D were the blank and dark current projection; and I_N was the normalized projection data. The AMER region was identified from full field of view (FOV) via the preprocessed projection. Then, the normalized projection and the parameters of the AMER were loaded into the GPU. After completing reconstruction on GPU, the CT image was read back into CPU for tiller identification.

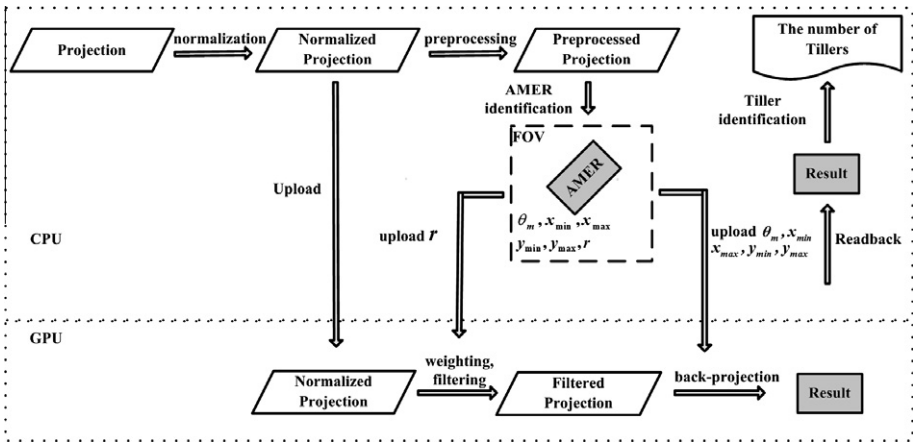

Fig. 1. Overall process of the present method.

2.3.1. AMER for parallel beam CT system

In parallel beam CT system, for each projection view, the beam which passed through the object formed a rectangular region. When considering two orthogonal view angles, the overlap of the two rectangular regions contained all the object points. As it is shown in Fig. 2a, the radial $S_{1S}D_{1S}$ and $S_{1E}D_{1E}$ were the tangents to the object at projection angle θ, while the radial $S_{2S}D_{2S}$, $S_{2E}D_{2E}$ were the tangents to the object at projection angle $\theta + 90°$. $P_1(x_{SS},y_{SS})$, $P_2(x_{ES},y_{ES})$, $P_3(x_{EE},y_{EE})$ and $P_4(x_{SE},y_{SE})$ were the intersections of tangent lines. And the rectangle they formed, which is indicated in red lines in Fig. 2a, was the overlap rectangular region, described as the enclosing rectangle. In order to determine the location of the enclosing rectangle, it was necessary to find the coordinates of $P_1(x_{SS},y_{SS})$, $P_2(x_{ES},y_{ES})$, $P_3(x_{EE},y_{EE})$ and $P_4(x_{SE},y_{SE})$. Fig. 2a illustrates the geometry of the enclosing rectangle for parallel beam projections. t_1 and t_2 denoted the rotated detector position at projection angle θ and $\theta + 90°$, respectively. O was the rotation center defined as the origin, and O_1 and O_2 were the projections of O on the detector at the angle of θ, $\theta + 90°$, respectively. D_{1S}, D_{1E}, D_{2S} and D_{2E} were the corresponding radial projections on the detector. w_{1S} and w_{1E} were the distances from O_1 to D_{1S} and O_1 to D_{1E}, respectively. w_{2S} and w_{2E} were the distances from O_2 to D_{2S} and O_2 to D_{2E}, respectively. w_{1S}, w_{1E}, w_{2S} and w_{2E} were positive if the distance vector was in the same direction as the detector, and negative otherwise. Since w_{1S}, w_{1E}, w_{2S} and w_{2E} can be attained from the projection image, the coordinates of $P_1(x_{SS},y_{SS})$, $P_2(x_{ES},y_{ES})$, $P_3(x_{EE},y_{EE})$ and $P_4(x_{SE},y_{SE})$ were computed in the following way. The lines OO_1 and P_1P_2 intersected at $A(a_x,a_y)$. Let $A_x = (a_x,0)$, $A_y = (0,a_y)$, $X_{SS} = (x_{SS},0)$ and $Y_{SS} = (0,y_{SS})$. According to Fig. 2a, the coordinates of $P_1(x_{SS},y_{SS})$ were computed as:

$$\begin{aligned} x_{SS} &= OA_x + A_xX_{SS} \\ y_{SS} &= OA_y + A_yY_{SS} \end{aligned} \tag{6}$$

where OA_x, OA_y, A_xX_{SS} and A_yY_{SS} were the lengths of the corresponding lines respectively. And the equations for calculating OA_x, OA_y, A_xX_{SS} and A_yY_{SS} were:

$$\begin{aligned} OA_x &= OA\cos\theta = -w_{2S}\cos\theta \\ OA_y &= OA\sin\theta = -w_{2S}\sin\theta \\ A_xX_{SS} &= AP_1\sin\theta = -w_{1S}\sin\theta \\ A_yY_{SS} &= AP_1\cos\theta = w_{1S}\cos\theta \end{aligned} \tag{7}$$

where OA and AP_1 were the lengths of the lines of OA and AP_1, respectively.

By combining Eqs. (6) and (7) together, we got the formula for x_{SS} and y_{SS}:

$$P_1(x_{SS},y_{SS})\begin{cases} x_{SS} = -w_{1S}\sin\theta - w_{2S}\cos\theta \\ y_{SS} = w_{1S}\cos\theta - w_{2S}\sin\theta \end{cases} \tag{8}$$

Analogously, the coordinates of $P_2(x_{ES},y_{ES})$, $P_3(x_{EE},y_{EE})$ and $P_4(x_{SE},y_{SE})$ were derived:

$$\begin{aligned} P_2(x_{ES},y_{ES})&\begin{cases} x_{ES} = -w_{1E}\sin\theta - w_{2S}\cos\theta \\ y_{ES} = w_{1E}\cos\theta - w_{2S}\sin\theta \end{cases} \\ P_3(x_{EE},y_{EE})&\begin{cases} x_{EE} = -w_{1E}\sin\theta - w_{2E}\cos\theta \\ y_{EE} = w_{1E}\cos\theta - w_{2E}\sin\theta \end{cases} \\ P_4(x_{SE},y_{SE})&\begin{cases} x_{SE} = -w_{1S}\sin\theta - w_{2E}\cos\theta \\ y_{SE} = w_{1S}\cos\theta - w_{2E}\sin\theta \end{cases} \end{aligned} \tag{9}$$

And let A_r be the area of the enclosing rectangle it was computed using the coordinates of the $P_1(x_{SS},y_{SS})$, $P_2(x_{ES},y_{ES})$, $P_3(x_{EE},y_{EE})$, and $P_4(x_{SE},y_{SE})$:

$$\begin{aligned} A_r &= |P_1P_2| \times |P_1P_4| = \left|\sqrt{(x_{SS}-x_{ES})^2 + (y_{SS}-y_{ES})^2}\right| \times \left|\sqrt{(x_{SS}-x_{SE})^2 + (y_{SS}-y_{SE})^2}\right| \\ &= |w_{1E}-w_{1S}| \times |w_{2E}-w_{2S}| \end{aligned} \tag{10}$$

where P_1P_2 and P_1P_4 were the lengths of the lines of P_1P_2 and P_1P_4 respectively, and $|.|$ denoted absolute value.

Considering all the projection couples that were perpendicular to one another, we used the enclosing rectangle with minimum area as the region of interest to be reconstructed.

2.3.2. AMER for fan beam CT system

Fig. 2b illustrates the geometry of the enclosing rectangle for fan beam projections. For the fan beam CT system, the X-ray (S) and detector points formed a triangle instead of a rectangle, while the overlap of the triangles was an anomalous quadrangle ($A_1A_2A_3A_4$, which is represented as blue lines in Fig. 2b). Construct two pair of lines of support for $A_1A_2A_3A_4$ through all four extreme points in the t_1 and t_2 directions. These lines formed an enclosing rectangle ($P_1P_2P_3P_4$, which is represented as solid red lines in Fig. 2b) for $A_1A_2A_3A_4$. Similarly, we defined the enclosing rectangle as the region of interest. Radial S_1O_1 intersected $A_1A_2A_3A_4$ at W_{2S} and W_{2E}. Radial S_2O_2 intersected $A_1A_2A_3A_4$ at W_{1S} and W_{1E}. Compared with the parallel beam geometry, w_{1S}, w_{1E}, w_{2S} and w_{2E} corresponded to the distances from O to the intersections W_{1S}, W_{1E}, W_{2S} and W_{2E} respectively. According to Fig. 2b, the coordinates of S_1, S_2, W_{1S}, W_{1E}, W_{2S} and W_{2E} were $(-R\cos\theta, -R\sin\theta)$, $(R\sin\theta, -R\cos\theta)$, $(-w_{1S}\sin\theta, w_{1S}\cos\theta)$, $(-w_{1E}\sin\theta, w_{1E}\cos\theta)$, $(-w_{2S}\cos\theta, -w_{2S}\sin\theta)$ and $(-w_{2E}\cos\theta, -w_{2E}\sin\theta)$, respectively. Using the coordinates of S_1 and W_{1S}, the equation for the line S_1D_{1S} was expressed in a two-point form as:

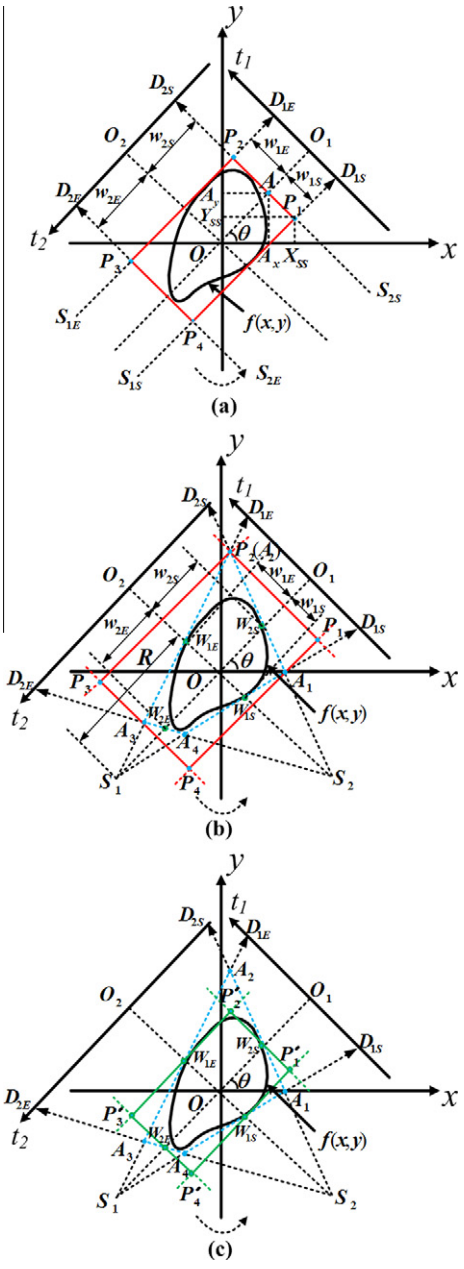

Fig. 2. Geometric diagram of the enclosing rectangle at a projection angle: (a) parallel beam; (b) fan beam; and (c) fan beam approximation.

$$y - w_{1S}\cos\theta = \frac{-R\sin\theta - w_{1S}\cos\theta}{-R\cos\theta + w_{1S}\sin\theta}(x + w_{1S}\sin\theta) \tag{11}$$

Similarly, the equation for the line S_2D_{2S} was expressed as:

$$y + w_{2S}\sin\theta = \frac{-R\cos\theta + w_{2S}\sin\theta}{R\sin\theta + w_{2S}\cos\theta}(x + w_{2S}\cos\theta) \tag{12}$$

Based on Eqs. (11) and (12), the coordinates (x_1,y_1) of the intersection A_1 were computed using Eq. (13):

$$A_1(x_1,y_1)\begin{cases} x_1 = \frac{R}{R^2+w_{1S}w_{2S}}[-w_{1S}(R-w_{2S})\sin\theta - w_{2S}(R+w_{1S})\cos\theta] \\ y_1 = \frac{R}{R^2+w_{1S}w_{2S}}[w_{1S}(R-w_{2S})\cos\theta - w_{2S}(R+w_{1S})\sin\theta] \end{cases} \tag{13}$$

The coordinates for A_2, A_3 and A_4 were derived analogously. Using the coordinates of these points and the direction of the enclosing rectangle (denoted as θ), the equations for the edges of the enclosing rectangle were derived. For instance, the line P_1P_4, which had slope $\tan\theta$ and passed through the point A_1, was expressed in point-slope form as:

$$y - y_1 = \tan\theta(x - x_1) \tag{14}$$

Finding the intersections of the four edges, the coordinates for P_1, P_2, P_3 and P_4 were calculated as:

$$P_1(x_{SS},y_{SS})\begin{cases} x_{SS} = -\frac{R(R-w_{2S})}{R^2+w_{1S}w_{2S}}w_{1S}\sin\theta - \frac{R(R+w_{1E})}{R^2+w_{1E}w_{2S}}w_{2S}\cos\theta \\ y_{SS} = \frac{R(R-w_{2S})}{R^2+w_{1S}w_{2S}}w_{1S}\cos\theta - \frac{R(R+w_{1E})}{R^2+w_{1E}w_{2S}}w_{2S}\sin\theta \end{cases}$$

$$P_2(x_{ES},y_{ES})\begin{cases} x_{ES} = -\frac{R(R-w_{2S})}{R^2+w_{1E}w_{2S}}w_{1E}\sin\theta - \frac{R(R+w_{1E})}{R^2+w_{1E}w_{2S}}w_{2S}\cos\theta \\ y_{ES} = \frac{R(R-w_{2S})}{R^2+w_{1E}w_{2S}}w_{1E}\cos\theta - \frac{R(R+w_{1E})}{R^2+w_{1E}w_{2S}}w_{2S}\sin\theta \end{cases}$$

$$P_3(x_{EE},y_{EE})\begin{cases} x_{EE} = -\frac{R(R-w_{2S})}{R^2+w_{1E}w_{2S}}w_{1E}\sin\theta - \frac{R(R+w_{1E})}{R^2+w_{1E}w_{2E}}w_{2E}\cos\theta \\ y_{EE} = -\frac{R(R-w_{2S})}{R^2+w_{1E}w_{2S}}w_{1E}\cos\theta - \frac{R(R+w_{1E})}{R^2+w_{1E}w_{2E}}w_{2E}\sin\theta \end{cases}$$

$$P_4(x_{SE},y_{SE})\begin{cases} x_{SE} = -\frac{R(R-w_{2S})}{R^2+w_{1S}w_{2S}}w_{1S}\sin\theta - \frac{R(R+w_{1E})}{R^2+w_{1E}w_{2E}}w_{2E}\cos\theta \\ y_{SE} = \frac{R(R-w_{2S})}{R^2+w_{1S}w_{2S}}w_{2S}\cos\theta - \frac{R(R+w_{1E})}{R^2+w_{1E}w_{2E}}w_{2E}\sin\theta \end{cases} \tag{15}$$

And the area of the enclosing rectangle was computed as:

$$A_r = |P_1P_2|\times|P_1P_4| = \left|\sqrt{(x_{SS}-x_{ES})^2+(y_{SS}-y_{ES})^2}\right|\times\left|\sqrt{(x_{SS}-x_{SE})^2+(y_{SS}-y_{SE})^2}\right|$$
$$= \frac{(1+\frac{w_{1E}}{R})(1-\frac{w_{2S}}{R})}{\left(1+\frac{w_{1E}w_{2S}}{R^2}\right)^2\left(1+\frac{w_{1S}w_{2S}}{R^2}\right)\left(1+\frac{w_{1E}w_{2E}}{R^2}\right)}|w_{1E}-w_{1S}|\times|w_{2E}-w_{2S}| \tag{16}$$

It was noticed that the computation of enclosing rectangle for fan beam CT system was more complicated than that for parallel beam CT system. In the H-SMART, R was 3800 while the maximum of w was 384, so the maximum angle between lines S_iD and S_iO was approximately 5°. To simplify the calculation, the radial lines were regarded as parallel lines. As shown in Fig. 2c, by constructing two lines paralleled to line t_2 and passed through point W_{1S} and point W_{1E} respectively and two lines paralleled to line t_1 and passed through point W_{2S} and point W_{2E} respectively, four points called P'_1, P'_2, P'_3 and P'_4 were determined. The rectangle with vertices $P'_1P'_2P'_3P'_4$ (solid green lines in Fig. 2c) was approximately equivalent with the enclosing rectangle $P_1P_2P_3P_4$ (solid red line in Fig. 2b). In this way, the enclosing rectangle was computed approximately using Eqs. (8) and (9) instead of using Eq. (15). With simplified computation, the enclosing rectangle may not contain the objects completely. In addition, the edge blurring in projection image may cause deviation in locating w, leading to a smaller computed enclosing rectangle and consequently some tillers were beyond the border of the reconstruction image. Therefore, there was a need for the enclosing rectangle expansion. Through the preliminary experiments, the rectangle was designed to expand outward 10 pixels. Since the enclosing rectangle was determined by w_{1S}, w_{1E}, w_{2S}, w_{2E} and θ, these variables were defined as the parameters of the enclosing rectangle.

2.3.3. Projection image preprocessing

To get the parameters of the enclosing rectangle, the projection image should be preprocessed. Since the detector had 768 detector units and the number of projection angles was 336, the size of the projection image was 336 × 768. An example of the preprocessing of the projection image is illustrated in Fig. 3. Fig. 3a is a projection image after normalization. Applying a fixed threshold which was determined by preliminary experiments, the projection image was segmented into a binary image, as shown in Fig. 3b. Whereas the binary image contained noises it was necessary to extract the object regions. Since the objects were rotated over 180°, the regions which belonged to the objects would pass through the position of the rotation center (shown as red dash line in the Fig. 3) in

the projection image. Therefore, it was feasible to extract the object regions by detecting the connected regions along the position of the rotation center. The extracting algorithm was based on stack data structure. The detailed steps were as follows:

Step 1: Initialize an empty stack.
Step 2: Along the column which represented the position of the rotation center, the projection image pixels were scanned from top to bottom. If a non-labeled foreground pixel p was scanned, it was labeled and the procedure jumped to Step 3 to scan the connective region of the labeled pixel. Otherwise, the procedure jumped to Step 5.
Step 3: Scan the neighbors of the pixel p. If a neighbor pixel was a non-labeled foreground pixel, it was labeled and pushed in the stack. When the stack was not empty, remove the top element q on the stack and use q to substitute p. And iteratively execute this step until the stack was empty.
Step 4: The procedure jumped to Step 2 to continue the scan along the column.
Step 5: Remove labeled regions whose areas were smaller than a predefined, fixed threshold. The remained labeled regions were the objects.

The extracted objects image of Fig. 3b is shown in Fig. 3c. Scanning the image from left to right, the first point and the last point on each row was acquired. To ensure all the active data was included in the AMER, both the first point and the last point were expanded 10 pixels. Moreover, the pixels between the first point and the last point were filled to facilitate AMER extraction. Fig. 3d shows the image of Fig. 3c after expansion and filling.

2.3.4. AMER identification

After preprocessing, the parameters of the enclosing rectangle were attained from the projection image and the AMER was determined by the following procedure:

Step 1: For each projection angle θ, calculate the distances from the start point and end point to the position of the rotation center, i.e. w_{1S} and w_{1E}.
Step 2: Similarly, calculate w_{2S} and w_{2E} for projection angle $\theta + 90°$.
Step 3: Calculate the area of the enclosing rectangle A_r at angle θ using Eq. (10).
Step 4: Identify the minimum enclosing rectangle with the minimum area among all angles.
Step 5: Calculate the vertices' coordinates of the minimum enclosing rectangle in the reconstructed image using Eqs. (8) and (9).

In the projection image space as shown in Fig. 4a, the No. 0 of the projection displacement (x-coordinate of the image) represented the projection position of the rotation center and the detector was oriented to the right. Fig. 4a illustrates the geometric diagram for the parameters of the minimum enclosing rectangle. As shown in the image, the distance parameters w_{1S}, w_{1E}, w_{2S} and w_{2E} were attained in the projection image.

The minimum enclosing rectangle calculated with these parameters was overlaid onto the FOV reconstruction image, as shown in Fig. 4b. It was noticed that the minimum enclosing rectangle was not always with sides parallel to the axis of image. Let θ_m be the projection angle of the minimum enclosing rectangle. To make image storage easy and reduce computation complexity, the coordinate system of the reconstruction image space was rotated by taking every projection angle minus θ_m during the backprojection stage. In this way, the orientation of AMER was horizontal or vertical. Fig. 4c shows the results of AMER and FOV reconstruction image after rotation. Let x_{min}, x_{max}, y_{min} and y_{max} be the value for the left side, right side, top and bottom of the rotated AMER, and r be the longest distance between the vertices of the rotated AMER and the rotation center O, the projection data which lay between No. $-r$ and No. r of the projection displacement, i.e. the so-called active region between two dash-dot lines in Fig. 4a, contained all the projection data of AMER. According to the FBP algorithm, it was viable to execute the convolution filtering merely on the active region and the backprojection operation merely on execute on the AMER region.

2.3.5. GPU based FBP reconstruction using AMER

According to Eqs. (1) and (2), there was no data dependency among different projections in the preweighted convolution filtering stage as well as among different pixels in the backprojection stage. Therefore, it was able to parallelize both the stages of reconstruction. Fig. 5 shows the pseudo-code of the GPU-based reconstruction algorithm. In Fig. 5, Algorithm 1 was the function which ran on the CPU for kernels invocation. Algorithm 2 and 3 were the kernels which ran on the GPU to execute preweighted convolution filtering and backprojection, respectively.

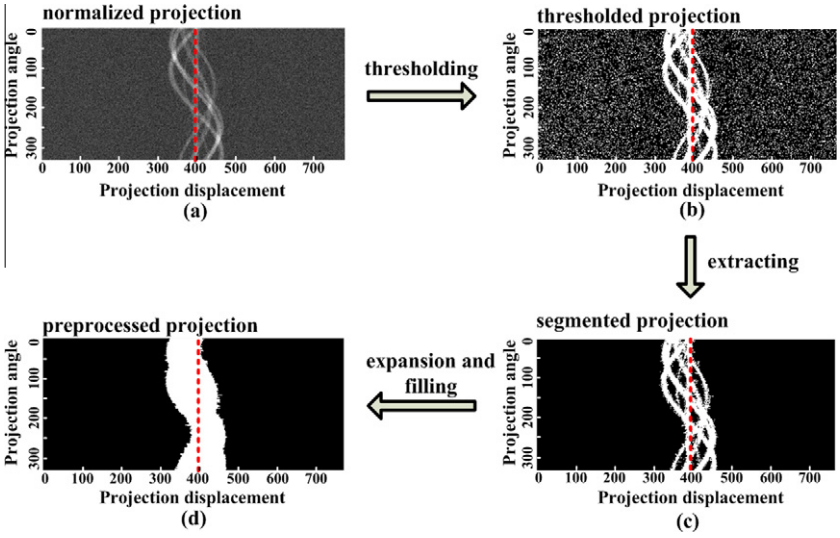

Fig. 3. Steps of the projection image preprocessing: (a) the normalized projection image; (b) the binary image of (a); (c) object regions extracted from (b); and (d) expansion and filling result.

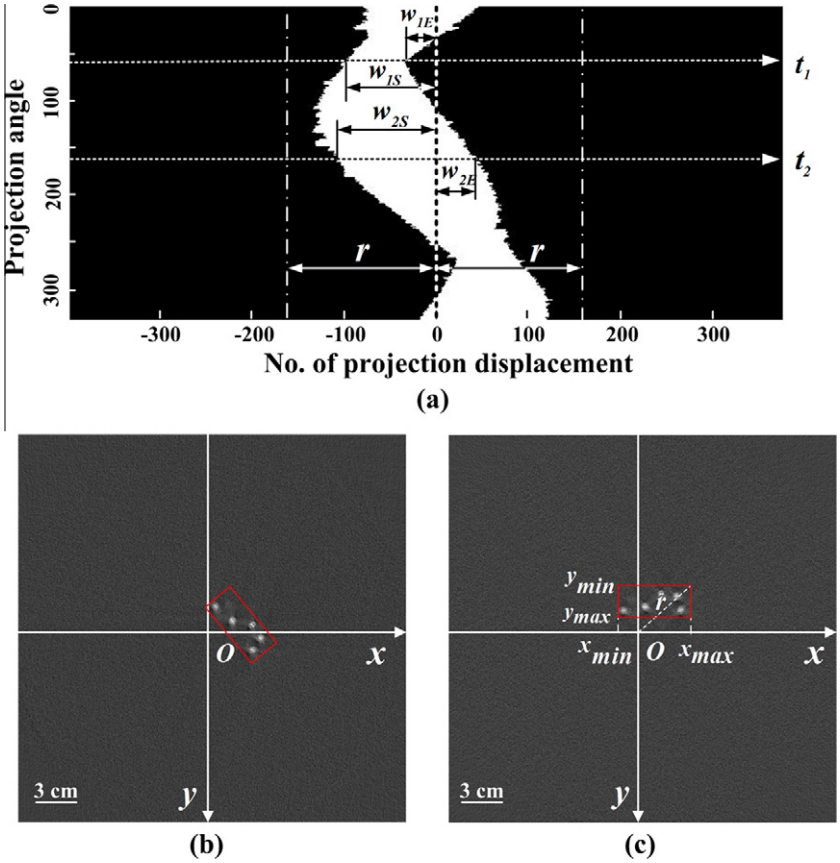

Fig. 4. Geometric diagram for AMER (the red rectangle) identification: (a) the parameters of the AMER in the projection image; (b) the AMER in reconstructed image before rotation; and (c) the AMER in reconstructed image after rotation. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Eq. (2) indicated that it did not need data exchange between pixels during the preweighted convolution filtering stage. However, each pixel in the same row referred all the projection data in the row and the convolution filter. Thus, the parallelism scheme for the preweighted convolution filtering was designed in the following way. The projection data was divided into each row and then was assigned to different thread blocks for the preweighted convolution filtering. In addition, the shared memory was used to store the projection data and the filter (line 26–28) for fast access. Let *c* be the corresponding column index of the rotation center in the projection image, the left boundary index (*LI*) and the right boundary index (*RI*) of the active region was attained (line 7 and 8). As it was mentioned above, only the projection data in the active region was performed (line 33). The size of a thread block (*blockNum*) was assigned to 256. Therefore, a thread was processing several pixels in the row, depending on the number of pixels in a row, as the two loops (line 21–39) shown in Algorithm 2.

While in the backprojection stage, the size of thread block was 16×16, and the blocks were organized into a two-dimensional grid. In this way, each pixel in the AMER region was assigned to different threads. The if condition in line 42 insured that only the pixels in the AMER region were performed. To hide the latency of addressing and get maximum memory bandwidth, the filtered projection data was stored in the texture memory (line 12) and the sine and cos values of projection angles was stored in constant memory (line 13) instead of the global memory.

2.3.6. Tiller identification

The tiller identification algorithm was illustrated in Yang et al. (2011b). However, there may be some leaves in the section images. If the leaf area was no less than the tiller area, it cannot be removed by the removing small particles operation. Therefore, we added a processing step to remove the leaves. Considering the leaves were always narrow, we used the length–width ratio as well as the area to identify them. After removing small particles, we performed two pre-defined thresholds to the length–width ratio of each individual object. The non-touching objects and the touching objects were classified by applying the lower threshold. If the length–width ratio of a touching object was larger than the upper threshold and its area was smaller than an area threshold, it was identified as a leaf and removed. All the thresholds used in the processing were determined through preliminary experiments. Because there was little parallelism in this stage, GPU technique was not suitable for tiller identification.

2.4. Algorithm performance test

As introduced above, the tiller inspection procedure of the present method consisted of projection normalization, AMER identification (contained sinogram preprocessing and AMER calculation), reconstruction and tiller identification. The present method was a combined AMER- and GPU-based method. To analysis the acceleration performance of different techniques, we compared the GPU-based method, the AMER-based method, and the AMER- and GPU-based method with the conventional CPU-based method respectively. The tiller inspection time of different methods was:

$$\begin{aligned} T_{CPU} &= T_N + T_{R.CPU.FOV} + T_{TI.FOV} \\ T_{GPU} &= T_N + T_{R.GPU.FOV} + T_{TI.FOV} \\ T_{AMER} &= T_N + T_{AI} + T_{R.CPU.AMER} + T_{TI.AMER} \\ T_{AMER+GPU} &= T_N + T_{AI} + T_{R.GPU.AMER} + T_{TI.AMER} \end{aligned} \tag{17}$$

```
(a) Algorithm1 FBP()
1: Initialize float h[2N-1]; // N: the detectors number
2: float CosValue[M], SinValue[M]; // M: The projection angle number
3: for i = 0 to M – 1 do
4:    CosValue[i] = cos (i - θm);
5:    SinValue[i] = sin (i - θm);
6: end for
7: LI = c – r;
8: RI = c + r;
9: Copy projection P to global memory;
10: Copy array h to global memory;
11: WeightConvFilterKernel(P, R, h, LI, RI);
12: Bind P as a texture;
13: Copy CosValue[M] and SinValue[M] to constant memory;
14: Copy f to global memory;
15: BackProjectionKernel(f, P, R, c, xmin, xmax, ymin, ymax, θm);
16: Copy f back to CPU memory;
(b) Algorithm2 WeightConvFilterKernel (P, R, h, c, LI, RI)
17: __shared__ float p [N];      // N: the detectors number
18: __shared__ float h'[2N-1];
19: nTile_p = N / blockNum;   // blockNum: the size of a thread block
20: nTile_Q = (RI – LI + 1) / blockNum;
21: for i = 0 to nTile_p do    // unroll
22:    a ← displacement(blockID, threadID);
23:    if a < N
24:    a' = a - c;
25:    weightValue = R / sqrt(R * R + a' * a');
26:    p[a] = weightValue * P[a, blockID];
27:    h'[N–1– a] = h[N–1– a];
28:    h'[N–1+ a] = h[N–1+ a];
29:    end if
30: end for
31: for i = 0 to nTile_Q do   // unroll
32:    a ← displacement(blockID, threadID) + LI;
33:    if a <= RI
34:    for j = 0 to N – 1 do   // unroll
35:       temp += p[j] * h'[N – 1 – j + a];
36:    end for
37:    P[a, blockID] = temp;
38:    end if
39: end for
(c) Algorithm3 BackProjectionKernel (f, P, R, c, xmin, xmax, ymin, ymax, θm)
40: x ← displacement(blockID, threadID) + xmin;
41: y ← displacement(blockID, threadID) + ymin;
42: if x <= xmax && y <= ymax
43:    for i = 0 to M do      // unroll
44:       U = R + x * CosValue[i] + y * SinValue[i];
45:       a = R * (y * CosValue[i] – x * SinValue[i]) / U;
46:       a' = a – c;
47:       if 0 <= a' < M
48:          temp += P[a', i] / U / U;
49:       end if
50:    end for
51:    f'[blockID, threadID] = temp;
52: end if
```

Fig. 5. Pseudo-code of the overall GPU algorithm: (a) the algorithm that runs on the CPU for kernels invocation; (b) the kernel that runs on the GPU to execute preweighted convolution filtering; and (c) the kernel that runs on the GPU to execute backprojection.

where T_{CPU}, T_{GPU}, T_{AMER} and $T_{AMER+GPU}$ were the tiller inspection time of the CPU-based method, the GPU-based method, the AMER-based method, and the AMER- and GPU-based method respectively, T_N was the normalization time, T_{AI} was the AMER identification time, $T_{R.CPU.FOV}$ and $T_{R.GPU.FOV}$ were the reconstruction time of the CPU-based method and the GPU-based method respectively, $T_{R.CPU.AMER}$ and $T_{R.GPU.AMER}$ were the reconstruction time of the AMER-based method and the AMER- and GPU-based method respectively, $T_{TI.FOV}$ was the tiller identification time of the CPU-based method and the GPU-based method, and $T_{TI.AMER}$ was the tiller identification time of the AMER-based method and the AMER- and GPU-based method. It was noticed that different methods had different effect on different stages. Therefore, the total reconstruction time (including the AMER identification time), the tiller identification time and the total tiller inspection time were measured to analyze the impact of each technique. The speedup of different methods was relative to the CPU-based method.

To evaluate the accuracy of the present method, the tiller numbers measured using the method were compared against the numbers measured using the conventional CPU-based method and the numbers counted by skilled workers.

The wheat samples used in this paper were composed of 200 pots of Zhengmai 9023 and 43 pots of Huamai 2533 (at the flowering stage). All of the experiments were performed on the workstation configured with 2.3 GHz main frequency, 3 GB memory, four CPU processors and an NVIDIA GeForce 9800 512 MB graphics card.

3. Results and discussion

3.1. Computation time

3.1.1. Reconstruction time of different acceleration techniques

To clarify the impact of different techniques on reconstruction, all the samples were tested. We selected 30 samples randomly for listing the reconstruction time and the speedup of different methods, as shown in Table 1. The GPU technique reduced reconstruction time with a speedup of approximately 200 times, while AMER technique achieved several tens of times higher performance. Therefore, GPU technique played a main role in the improvement of reconstruction speed. By combining AMER and GPU, the reconstruction time was reduced from about 11,500 ms to tens of ms.

Note that the speedup differed in different images with the AMER technique. The reason was that the sizes of the AMER for each image may be varied. Fig. 6 shows the relationship between the speed ratio of reconstruction and the area ratio. As shown in Fig. 6, for all the 243 pots of the test samples, the speed ratio for the total reconstruction process decreased as the area ratio of the AMER and the FOV was larger. For all the test samples, the area ratio ranged from 0.0047 to 0.2751. The speed ratio ranged from 3 to 38 when only AMER technique was used (shown in Fig. 6a), and ranged from 169 to 1276 when both AMER technique and GPU technique were used (shown in Fig. 6b).

It was worth mentioning that the speed ratio of the AMER- and GPU-based technique was lower than the GPU-based speed ratio when the area ratio was larger than 0.2. This was because T_{AI} would be longer than $T_{R.GPU.FOV}$ minus $T_{R.GPU.AMER}$, when area ratio was larger than 0.2. It meant that the AMER technique would make no sense compared with GPU technique, when the proportion of AMER area was large. Generally, the proportion of AMER area was small for wheat plants in this work, so applying the AMER technique was always effective on acceleration. While in other applications, it was necessary to analyze whether the objects region was smaller than the FOV region.

3.1.2. Influencing factors of AMER-based reconstruction time

Fig. 6 shows that the speed ratio decreases with the increase of the area ratio of the AMER and the FOV. It was observed that there was a fluctuation in the speed ratio at the same area ratio. This was because the total reconstruction time consisted of the AMER identification time (T_{AI}), the weighted operation time (T_w), the convolution filtering time (T_f) and the backprojection time (T_b). The area of AMER merely determined the pixels which should be backprojected. However, the weighted operation used all the projection data, and the amount of convolution filtering pixels was determined by r, which was the longest distance between the vertices of the rotated AMER and the rotation center. In other words, T_{AI} and T_w were fixed, while T_f and T_b were affected by r and area ratio, respectively.

In order to further analyze the effect of the parameter r, three area ratio levels were selected from the range of area ratio and the speed ratio and r value of each sample at each level were recorded. For reliable trend analysis, three different rating area ratios (0.01, 0.04 and 0.1) with a relatively more samples were selected. The results are shown in Fig. 7. The lines in the image were the trendlines of speed ratios. It was inferred that the speed ratio of

Table 1
Reconstruction time and speedup of different acceleration methods over CPU-based method for 30 pots of wheat plants.

No.	CPU-based method	GPU-based method		AMER-based method		AMER- and GPU-based method	
	Time (ms)	Time (ms)	Speedup	Time (ms)	Speedup	Time (ms)	Speedup
1	11485	58	198	515	22	12	957
2	11486	58	198	3491	3	64	179
3	11485	58	198	754	15	17	676
4	11487	59	195	1046	11	20	574
5	11485	57	201	1250	9	27	425
6	11485	58	198	535	21	13	883
7	11486	58	198	3841	3	68	169
8	11485	58	198	688	17	15	766
9	11485	58	198	627	18	16	718
10	11486	58	198	472	24	10	1149
11	11486	58	198	1255	9	24	479
12	11486	57	202	773	15	17	676
13	11486	58	198	1002	11	13	884
14	11485	58	198	468	25	12	957
15	11485	58	198	740	16	18	638
16	11486	58	198	1091	11	24	479
17	11487	58	198	1173	10	25	459
18	11486	57	202	1366	8	29	396
19	11485	58	198	2364	5	47	244
20	11485	57	201	2011	6	44	261
21	11485	58	198	1241	9	26	442
22	11487	58	198	865	13	20	574
23	11486	58	198	1481	8	30	383
24	11485	59	195	1283	9	27	425
25	11485	58	198	1605	7	33	348
26	11486	58	198	2272	5	44	261
27	11487	57	202	1750	7	39	295
28	11487	57	202	1667	7	26	442
29	11485	57	201	677	17	16	718
30	11485	57	201	1125	10	27	425

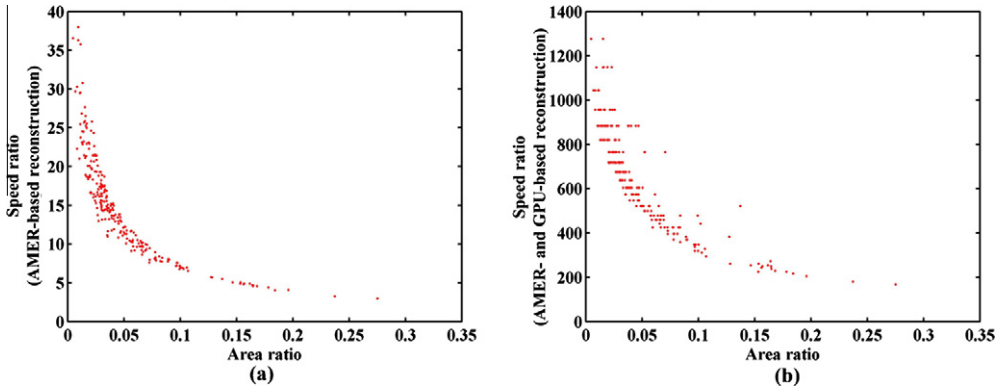

Fig. 6. Relationship between the speed ratio of reconstruction and the area ratio: (a) the AMER-based method; and (b) the AMER- and GPU-based method.

reconstruction decreased with the increase of r at the same area ratio. What's more, the influence of r on the reconstruction speed increased as the area ratio decreased. The reason was that the proportion of T_f was increased when T_b was reduced significantly.

3.1.3. Tiller identification acceleration with the AMER technique

There should be some differences between $T_{TI.FOV}$ and $T_{TI.AMER}$. As the size of the reconstruction image got smaller by the AMER identification, the time consumption of tiller identification was reduced. The tiller identification time and the speedup for the same 30 pots of wheat plants used in Table 1 are shown in Table 2. The AMER technique led to considerable improvement in tiller identification, and the area ratio of the AMER and the FOV also had effect on the speed ratio of tiller identification (Fig. 8). The speed ratio decreased with the increase of the area ratio. As the area ratio ranged from 0.0047 to 0.2751, the speed ratio ranged from 4 to 86. From Fig. 8 it was noticed that the speed ratio varied at the same area ratio. That was because except for AMER size, the time consumption of tiller identification was also determined by other factors such as tiller number, number of touching objects and number of foreground pixels et al.

3.1.4. Tiller inspection time of different acceleration techniques

As mentioned above, GPU technique improved reconstruction with a higher speedup comparing to the AMER method, while the AMER technique had an extra acceleration for tiller identification. To analyze the impact of each technique on tiller inspection, T_{CPU}, T_{GPU}, T_{AMER} and $T_{AMER+GPU}$ were measured. Table 3 shows the tiller inspection time and the speedup of different methods for the above 30 pots of wheat plants. The AMER technique achieved at least 3 times higher performance than the CPU-based method, and the GPU technique achieved about 34 times higher performance, which was a little better than the AMER technique. The entire image processing time for tiller inspection was further reduced by combining the two techniques. For each test sample, the entire tiller inspection time was reduced from approximately 11870 ms

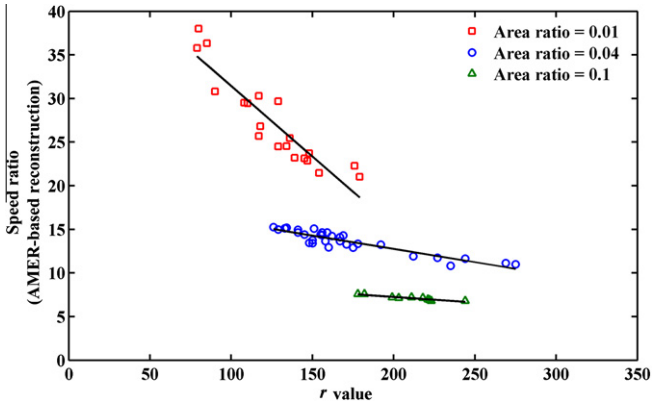

Fig. 7. Relationship between the speed ratio of reconstruction and the r value at the same area ratios.

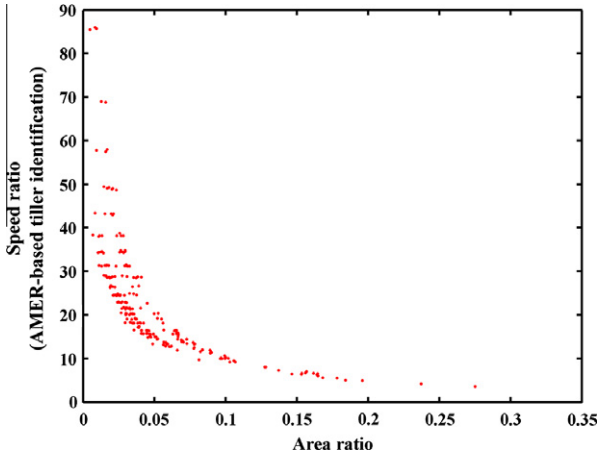

Fig. 8. Relationship between the speed ratio of tiller identification and the area ratio.

Table 2
Tiller identification time and speedup for 30 pots of wheat plants with AMER technique.

No.	Without AMER-based method	AMER-based method	
	Time (ms)	Time (ms)	Speedup
1	347	7	50
2	349	84	4
3	348	17	20
4	343	12	29
5	346	20	17
6	346	13	27
7	347	96	4
8	345	8	43
9	343	13	26
10	343	7	49
11	343	24	14
12	344	12	29
13	348	17	20
14	343	10	34
15	349	13	27
16	347	21	17
17	347	24	14
18	349	30	12
19	346	52	7
20	347	43	8
21	345	22	16
22	347	16	22
23	345	29	12
24	346	27	13
25	348	34	10
26	349	52	7
27	349	38	9
28	351	35	10
29	346	12	29
30	349	23	15

Table 3
The tiller inspection time and speedup of different acceleration methods over CPU-based method for 30 pots of wheat plants.

No.	CPU-based method	GPU-based method		AMER-based method		AMER- and GPU-based method	
	Time (ms)	Time (ms)	Speedup	Time (ms)	Speedup	Time (ms)	Speedup
1	11864	347	34	553	21	51	233
2	11868	349	34	3607	3	180	66
3	11865	348	34	803	15	59	201
4	11860	343	35	1090	11	64	185
5	11863	346	34	1302	9	85	140
6	11864	346	34	581	20	58	205
7	11866	347	34	3968	3	197	60
8	11862	345	34	728	16	59	201
9	11859	343	35	673	18	62	191
10	11862	343	35	511	23	53	224
11	11860	343	35	1312	9	81	146
12	11862	344	34	817	15	62	191
13	11863	348	34	1051	11	68	174
14	11862	343	35	510	23	55	216
15	11866	349	34	786	15	65	183
16	11865	347	34	1143	10	82	145
17	11865	347	34	1230	10	82	145
18	11868	349	34	1428	8	97	122
19	11862	346	34	2448	5	131	91
20	11865	347	34	2086	6	119	100
21	11862	345	34	1296	9	80	148
22	11867	347	34	913	13	74	160
23	11861	345	34	1542	8	91	130
24	11866	346	34	1343	9	88	135
25	11868	348	34	1672	7	101	118
26	11866	349	34	2356	5	129	92
27	11868	349	34	1820	7	109	109
28	11867	351	34	1734	7	93	128
29	11865	346	34	721	16	64	185
30	11864	349	34	1180	10	81	146

to less than 200 ms with the AMER- and GPU-based method. The results illustrated that the AMER- and GPU-based method was an optimization method for tiller inspection.

3.2. Measurement accuracy

The present method was implemented on the H-SMART for the measurement of wheat tiller number. Since it had been proved that the section image quality was not compromised when using GPU for CT reconstruction (Xu and Mueller, 2007; Noël et al., 2010), here we only analyzed the results of wheat tillers measurement using the conventional CPU-based method and the AMER- and GPU-based method against the manually measurement results. Fig. 9 shows a comparison of the three methods. The plot of the measurement results for Zhengmai 9023 was given in Fig. 9a while the plot of the measurement results for Huamai 2533 was given in Fig. 9b. The overlapping of the curves indicated that the differences of different methods were not significant and the automatic measurements accuracy was high. It was observed that there was a slight difference between the two automatic measurements, which was due to the rotation of AMER. As the section image reconstructed using the AMER was rotated by a θ_m angle, there would be interpolation that may cause gray value difference of each corresponding pixels in the two section images.

The mean absolute error (MAE defined by Eq. (18)) of the manual and automatic measurements of tiller number were calculated, respectively. With the CPU-based method, the MAE for the two

wheat varieties was 0.15 and 1.33, respectively. With the AMER- and GPU-based method, the MAE for the two wheat varieties was 0.11 and 1.26, respectively:

$$\mathrm{MAE} = \frac{1}{n} * \sum_{i=1}^{n} |x_{ai} - x_{mi}| \tag{18}$$

where x_{ai} and x_{mi} were the automatically measured tiller number and the manually measured tiller number for the ith sample, respectively, and n was the number of samples.

The MAE of Huamai 2533 was higher because there were more small tillers with low X-ray absorption because of water shortage. The pixel values of these tillers in the section image were close to background, therefore, they may not be identified. Linear regression of manual measurement and automatic measurement (using the AMER- and GPU-based method) for Zhengmai 9023, Huamai 2533 are shown in Fig. 10a and b, respectively. The results indicated that the automatic measurement with the AMER- and GPU-based method was highly correlated with manual measurement and there was a well linear relationship between them.

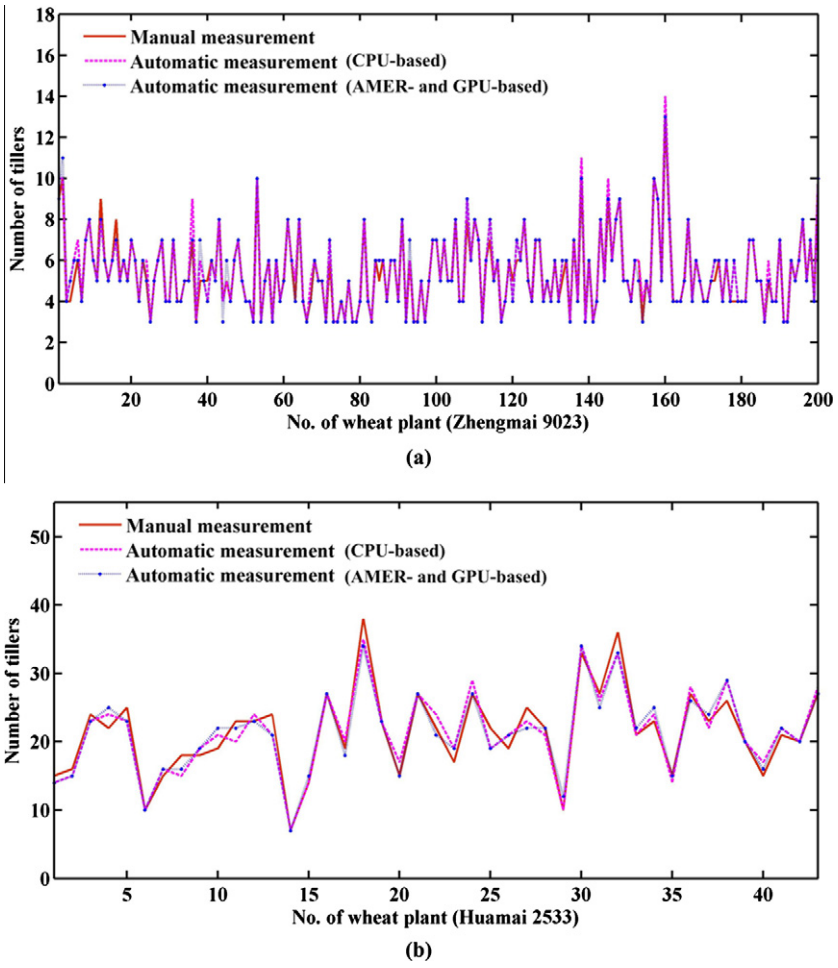

Fig. 9. Comparisons of manual measurement, automatic measurement with the CPU-based method and automatic measurement with the AMER- and GPU-based method for the two varieties: (a) Zhengmai 9023; and (b) Huamai 2533.

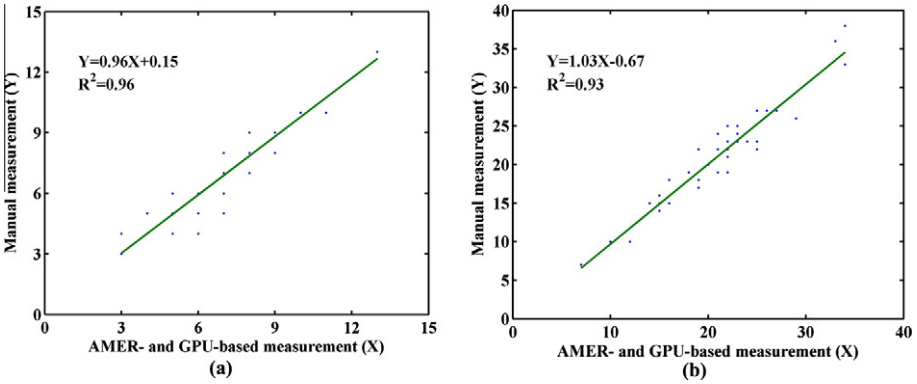

Fig. 10. Scatter plot for the AMER- and GPU-based measurement (X) and manual measurement (Y) of tiller numbers: (a) Zhengmai 9023; and (b) Huamai 2533.

4. Conclusions

In this paper, we presented an AMER method to reduce the computation amount of CT reconstruction. Combined with the GPU technique, the method was capable of speeding up the reconstruction of wheat section in H-SMART about hundreds of times faster than the conventional CPU-based method. Moreover, the method was able to speed up the tiller identification about tens of times without reducing the measurement accuracy of tiller number. With the acceleration of reconstruction, the H-SMART performance was improved, which excavated the potential of integrating other imaging technologies into the facility. In addition, this method had the potential of being applied to three-dimensional CT imaging systems with some modifications. The improvement of the method got more significantly as the area ratio of the AMER and the FOV got smaller, and the method will be more effective with a larger volume. In sum, this method expanded CT application in plant phenomics and agriculture photonics.

Acknowledgements

We thank the Huazhong Agricultural University for their assistance in sample preparation. This work was supported by the Key Program of the Natural Science Foundation of Hubei Province (Grant No. 2008CDA087), and the Program for New Century Excellent Talents in University (No. NCET-10-0386).

References

Barcelon, E.G., Tojo, S., Watanabe, K., 1999. X-ray computed tomography for internal quality evaluation of peaches. Journal of Agricultural Engineering Research 73, 323–330.

Benson, T.M., Gregor, J., 2006. Three-dimensional focus of attention for iterative cone-beam micro-CT reconstruction. Physics in Medicine and Biology 51, 4533–4546.

Cubero, S., Aleixos, N., Moltó, E., Gómez-Sanchis, J., Blasco, J., 2011. Advances in machine vision application for automatic inspection and quality evaluation of fruit and vegetables. Food and Bioprocess Technology 4, 487–504.

Dhanda, S.S., Sethi, G.S., Behl, R.K., 2004. Indices of drought tolerance in wheat genotypes at early stages of plant growth. Journal of Agronomy and Crop Science 190, 6–12.

Dhondt, S., Vanhaeren, H., Loo, D.V., Cnudde, V., Inzé, D., 2010. Plant structure visualization by high-resolution X-ray computed tomography. Trends in Plant Science 15 (8), 419–422.

Gac, N., Mancini, S., Desvignes, M., 2006. Hardware/software 2D–3D backprojection on a SoPC platform. In: Proceedings of the 2006 ACM Symposium on Applied, Computing, pp. 222–228.

Heeraman, D.A., Hopmans, J.W., Clausnitzer, V., 1997. Three dimensional imaging of plant roots in situ with X-ray computed tomography. Plant and Soil 189, 167–179.

Herman, G.T., 1995. Image reconstruction from projections. Real-time Imaging 1, 3–18.

Kachelrieß, M., Knaup, M., 2007. Hyperfast parallel-beam and cone-beam backprojection using the cell general purpose hardware. Medical Physics 34 (4), 1474–1486.

Kak, A.C., Slaney, M., 1988. Principles of Computerized Tomographic Imaging. IEEE, New York, USA, pp. 75–93.

Kuraparthy, V., Sood, S., Dhaliwal, H.S., Chhuneja, P., Gill, B.S., 2007. Identification and mapping of a tiller inhibition gene (tin3) in wheat. Tag Theoretical and Applied Genetics 114, 285–294.

Lee, B., Lee, H., Shin, Y.G., 2010. Fast hybrid CPU- and GPU-based CT reconstruction algorithm using air skipping technique. Journal of X-ray Science and Technology 18, 221–234.

Li, Y., Zu, Y., Chen, H., Chen, J., Yang, J., Hu, Z., 2000. Intraspecific responses in crop growth and yield of 20 wheat cultivars to enhanced ultraviolet-B radiation under field conditions. Field Crops Research 67, 25–33.

Longuetaud, F., Leban, J.M., Mothe, F., Kerrirn, E., Berger, M.O., 2004. Automatic detection of pith on CT images of spruce logs. Computers and Electronics in Agriculture 44, 107–119.

Mass, E.V., Lesch, S.M., Francois, L.E., Grieve, C.M., 1994. Tiller development in salt-stressed wheat. Crop Science 34 (6), 1594–1603.

Neethirajan, S., Jayas, D.S., White, N.D.G., Zhang, H., 2008. Investigation of 3D geometry of bulk wheat and pea pores using X-ray computed tomography images. Computers and Electronics in Agriculture 63, 104–111.

Noël, P.B., Walczak, A.M., Xu, J., Corso, J.J., Hoffmann, K.R., Schafer, S., 2010. GPU-based cone beam computed tomography. Computer Methods and Programs in Biomedicine 98, 271–277.

Okitsu, Y., Ino, F., Hagihara, K., 2010. High-performance cone beam reconstruction using CUDA compatible GPUs. Parallel Computing 36, 129–141.

Sharma, R.C., 1995. Tiller mortality and its relationship to grain yield in spring wheat. Field Crops Research 41, 55–60.

Singh, C.B., Choudhary, R., Jayas, D.S., Paliwal, J., 2010. Wavelet analysis of signals in agriculture and food quality inspection. Food and Bioprocess Technology 3, 2–12.

Stuppy, W.H., Maisano, J.A., Colbert, M.W., Rudall, P.J., Rowe, T.B., 2003. Three-dimensional analysis of plant structure using high-resolution X-ray computed tomography. Trends in Plant Science 8 (1), 2–6.

Xu, F., Mueller, K., 2007. Real-time 3D computed tomographic reconstruction using commodity graphics hardware. Physics in Medicine and Biology 52, 3405–3419.

Yang, W., Xu, X., Bi, K., Zeng, S., Liu, Q., Chen, S., 2011a. Adaptive region of interest method for analytical micro-CT reconstruction. Journal of X-ray Science and Technology 19, 23–33.

Yang, W., Xu, X., Duan, L., Luo, Q., Chen, S., Zeng, S., Liu, Q., 2011b. High-throughput measurement of rice tillers using a conveyor equipped with X-ray computed tomography. Review of Scientific Instrument 82, 025102-1–025102-7.

Yu, L., Qi, D., 2008. Analysis and processing of decayed log CT images based on multifractal theory. Computers and Electronics in Agriculture 63, 147–154.