An automatic detection algorithm for the maize kernel row number based on its convex characteristics

Xinhui Wu, Jing Li

Anyang Institute of Technology,
Anyang 455000, Henan, China

Junqiao Song

Anyang Academy of Agricultural Sciences,
Anyang 455000, Henan, China

Abstract
This paper introduces a new detection algorithm to calculate the maize kernel row number automatically. The algorithm is based on convex characteristics of the ear cross section. The convex hull of the contour of an ear cross section is the same as that of the cross section itself. We use an improved 2-d OTSU algorithm as the threshold segmentation algorithm to obtain the binary image of the ear cross section and its exact contour. Afterward we adopt Quickhull algorithm to calculate the convex hull of the contour and fill it with the pixel value the same as that of the ear cross section. The binary image of the ear cross section is subtracted from the convex hull image to obtain the convex deficiency image, the connected component of which is labeled to achieve the number of the kernel row. Particularly, a new hybrid optimization algorithm is introduced to the image segmentation. By experiments, we detected 500 samples. The one-sample detection time was 0.427s and the precision rate was 100%. The algorithm proposed can reach the results more quickly and more accurately.
Key words: MAIZE KERNEL ROW NUMBER, AUTOMATIC DETECTION, CONVEX HULL, HYBRID OPTIMIZATION ALGORITHM, THRESHOLD SEGMENTATION, CONNECTED COMPONENT.

Introduction

Maize is almost grown on every continent mainly in temperate latitudes (40 degrees North to 50 degrees South), except Antarctica. It is a multipurpose crop used for food, feed and industrial materials. Besides, maize production plays an important role in food production and food security in the world. Dai Jingrui, Chinese Academy of engineering, explained that breeding new varieties of maize accounts for 40 to 50 percent of all the factors for the increasment of maize yield. In other words, the quality of maize seeds directly relates to the yield and quality of maize. However, breeding new varieties of maize needs a great quantity of high quality seeds. [1, 2]

The kernel row number is one of the maize's important agronomic characters. Maize with different varieties has different kernel row numbers. But the kernel row numbers of maize with the same variety

would differ if provided different growth conditions. Accurately calculating the kernel row number is very important for maize breeding, cultivation and new variety DUS testing (DUS: Distinctness, Uniformity and Uniformity). [3, 4, 5]

In the main producing district around world, the age of artificial counting is gradually replaced. However, the existing automatic detection algorithms on kernel row number using the method belonging to image processing or machine vision are sort of shortage. For example, in an ear cross section image, distances between points on its outer contour and its mass center are different. The kernel row number can be calculated by determining the number of minimum values or maximum values of the distances. This algorithm reported usually need huge computation and its accuracy was over 90% when testing. Another algorithm reported utilized the method of egde labeling and discrete curvature calculation which had an accuracy above 90% when testing. Generally, the utilization of the ear cross section's shape features determines the algorithm's speed and accuracy. [6, 7]

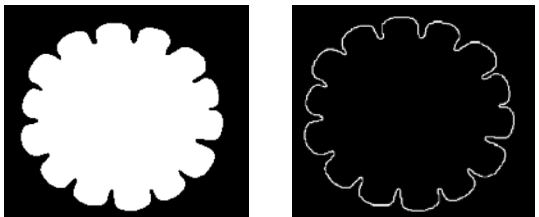
This paper introduces a new kernel row number detection algorithm to overcome the disadvantages of the existing methods. We found that a maize ear cross section image has convex characteristics. Find the convex hull image of the ear cross section. And subtracting the ear cross section from the convex hull image is able to obtain the kernel gap image showing the gaps between any two adjacent kernels. The number of kernel gaps is the number of the kernel row itself. Finally, the connected component of the kernel gap image can be labeled to achieve the number of the kernel row. This algorithm is based on the convex characteristics of the ear cross section, uses the convex hull algorithm and an improved image segmentation algorithm, and takes advantage of the connected component operation. This original approach and effective algorithm can make the kernel row number detection quicker and more accurate.

Theoretical analysis and algorithm design

Put a maize ear cross section on the black background. Let the ring light illuminate the maize ear. The camera is fixed in the center of the ring light and captures the maize ear cross section image. The original maize ear cross section image is shown in figure 1.

Figure 1. Original image of a maize ear cross section

Contour feature image of the ear cross section is achieved by separating the ear cross section from the background using the image segmentation algorithm. Firstly, transform the original ear cross section image to gray image in order to reduce the image data amount when processing. Secondly, a combination algorithm with median filter algorithm and wiener filtering algorithm is utilized to undo noise in the gray image. Thirdly, we use the 2-dimensional OTSU algorithm to split the image, keep the background black (pixel value equaling to 0), and fill the edge and the inner part with white color (pixel value equaling to 1). After completing these steps, we get the binary contour feature image of the ear cross section, shown in figure 2.



(a) Contour feature image (b) Edge line

Figure 2. Contour feature image of the ear cross section

According to the concept of convex hull, In mathematics, in a real vector space V, C is a given set. The convex hull or convex envelope of the set C is the smallest convex set S which contains C.

$$S = \bigcap_{\substack{C \subseteq K \subseteq V \\ K \text{ is convex}}} K \qquad (1)$$

The convex hull of the set C can be formed by a linear combination of all its points $(c_1, c_2, c_3, \ldots, c_n)$. See formula 2.

$$S = \{\sum_{j=1}^{n} t_j c_j \mid c_j \in C, \sum_{j=1}^{n} t_j = 1, t_j \in [0,1]\} \quad (2)$$

In the Euclidean plane, if and only if the segment $\overline{PQ}$ belongs to the set S when both Point P and Point Q belong to S, the set S is convex. Or a set of points is defined to be convex if it contains the line segments connecting each pair of its points. In other words, when C is a bounded subset of the plane, the convex hull S may be visualized as the shape formed by a rubber band stretched around C. [8]

To calculate the convex hull image of figure 2 is to find a convex hull of the ear cross section, which is a convex polygon enclosing all pixels of the ear cross section (in figure 3), and to fill the convex polygon with the value 0, which is the value of the pixels in the ear cross section. Actually, from the characteristics of convex hull, the convex hull of the ear cross section is the same with the convex hull of its edge. The edge of the ear cross section is in figure 2(b). Therefore, we can calculate the convex hull image of figure 2(b) instead, which is able to reduce computation points and save operation time.

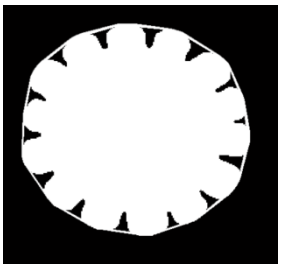


Figure 3. Convex characteristics of the ear cross section

2-dimensional Quick Hull algorithm is known for computing the convex hull for a finite set of points in Euclidean plane. It uses a divide and conquer approach. Its average case complexity is considered to be $O(n\log(n))$, whereas in the worst case it approximately equals to $O(n^2)$. Suppose the set of pixels on the edge line in figure 2(b) is denoted as the set C, and the set of pixels on the convex hull of C is denoted as the set S. The algorithm for this problem can be broken down to the following steps: [9, 10, 11]

(1) Find two points with minimum and maximum x coordinates, denoted by N_1 and N_2. Those are bound to be on the convex hull.

(2) Use the line connecting N_1 and N_2 to divide the set C in two subsets of points, C_1 and C_2. Judging every point belonging to C is in C_1 or C_2 by traversing method.

(3) Determine the point, on one side of the line (or in C_1), with the maximum distance from the line. Denote the point by P_1. The three points N_1, N_2 and P_1 form a triangle. The points inside of this triangle can not be part of the convex hull and can be ignored in the next steps.

(4) For the points N_1 and P_1, repeat step 2 to step 3. For the points N_2 and P_1, repeat step 2 to step 3. This step will be processed recursively. Until no more points are left, the recursion has come to an end.

(5) All the points that are used to form the judging triangles are the vertexes of the convex hull. The convex hull of C can be achieved by connecting all the vertexes.

Theoretically, the complexity of this Quick Hull algorithm is to be $O(m\log(n))$, where n is the pixels' number of the edge in figure 2(b) and m is the vertexes' number of the convex hull. Find the convex hull image of the ear cross section edge, and fill the inside part of the convex hull with the value which is the same as that of the pixels in the ear cross section. Figure 4 shows the result.

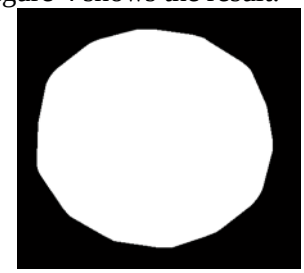


Figure 4. Convex hull image of the ear cross section

Subtracting figure 2(a) from figure 4, we obtain figure 5, convex deficiency image of the ear cross section. The white districts are defined as connected component. Obviously, the white parts with large area are gaps

between any adjacent kernels. According to the characteristic of ear cross section, the gap number is equal to the kernel number or the kernel row number. However there are many other white parts with very tiny area.

The tiny white districts in figure 5 are formed because of the rough and uneven surfaces of the kernels. Here we use a threshold value A_t. If the area of a white district is less than A_t, the white district must be eliminated by assigning the value of every pixel in the district to be zero. Therefore we have to label each white part and delete the domain whose area is less than A_t.

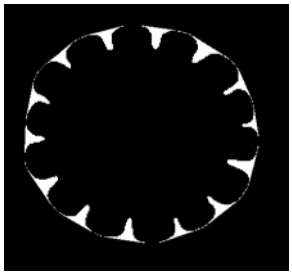


Figure 5. Convex deficiency image of the ear cross section

The following steps explain the algorithm in detail. [12, 13, 14, 15]

(1) Scanning the input binary image progressively. Continuous pixels with white color are formed a sequence, which is called a Run. Record the coordinate of start point, the coordinate of end point and the row number. Label the Run use a integer number starting at 1.

(2) Use i to represent the image row under scanning. Suppose a Run belonging to Row i ($i>1$). The overlap judgement is based on comparing the 8-Neighbor points of the center pixel.

(i) If there are no overlap domains between the Run under labeling and the Runs in Row $i-1$, denote the Run under labeling by a new number.

(ii) If there is only one Run in Row $i-1$ overlapping the Run under labeling, denote the Run under labeling by the mark of the overlapping one.

(iii) If there are more than one Run in Row $i-1$ overlapping the Run under labeling, denote the Run under labeling by the minimum number which represents one of the overlapping Runs in Row $i-1$. Furthermore, write the marks of the overlapping Runs as equal - label list to declare that these Runs are connected.

(3) Transform the equal - label lists into equal - label sequences. Relabel the Run represented by the labels in a equal - label sequence with the smallest number in the sequence because all the Runs label ed by the sequence are connected.

(4) Calculate the areas of the labeled Runs one by one. Denote one-twentieth the difference between maximum area and minimum area by A_t. Traverse all the labeled Runs. If the Run has an area less than A_t, eliminate the Run label and make all its pixels' value to be zero. Otherwise, retain the Run and relabel it using a number after the former Run.

Completing the steps above, the connected component who has an area less than A_t is eliminated from the image, shown in figure 6. Meanwhile, all the connected components left are labeled. The kernel row number is the maximum label number of the connected components.

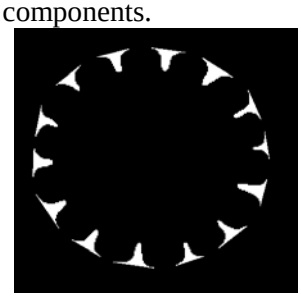


Figure 6. Kernel gaps in the convex deficiency image

To sum up, the entire process of the kernel row number automatic detection can be illustrated by figure 7.

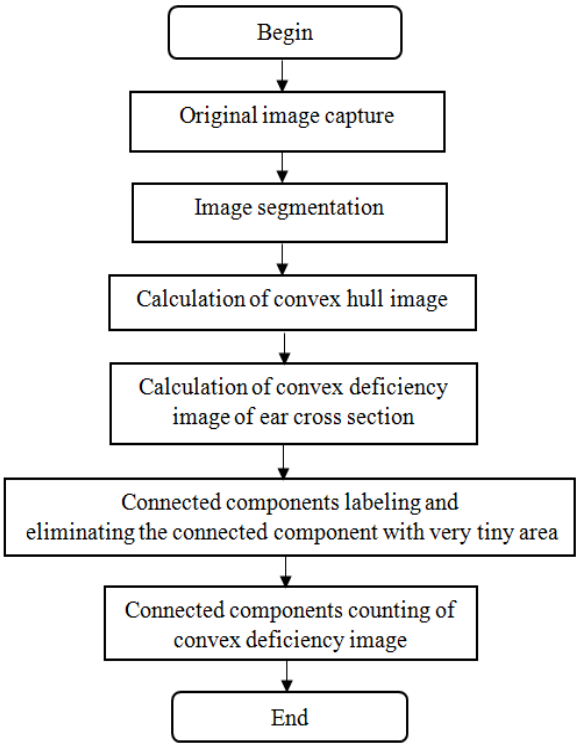


Figure 7. Flow chart of kernel row number automatic detection

The most important step is the image segmentation. The reason is that the image segmentation, which is design to obtain the contour image of the ear cross section, is the key to calculate the following convex hull image. Besides, if the image segmentation is inaccurate, it's almost impossibility to achieve the true kernel gaps image by calculating the convex deficiency image. Therefore the image segmentation algrithm determines the success or failur of the kernel row number automatic detection. In this paper, we proposed a new hybrid optimization algorithm to improve the 2-d OTSU image segmentation algorithm, which will be introduced in section 3.

Image segmentation algorithm

In image processing, OTSU algorithm is an automatic algorithm used to perform clustering-based image threshold. It can reduce a gray level image to a binary image. The algorithm assumes that the image contains two classes of pixels following bi-modal histogram, which are foreground pixels and background pixels. It is to calculate the optimum threshold in order to separate the two classes so that their intra-class variance is minimal. However, traditional OTSU has a sensitive-to-noise shortcoming. 2-d OTSU algorithm was proposed to overcome the disadvantage. It utilizes both the gray level information and the spatial correlation information within the neighborhood of each pixel. Therefore 2-d OTSU has a powerful anti-noise ability. [16, 17]

Suppose the size of the image in pixel is $M \times N$. Use function $F(x, y)$ to represent the image, and its gray level is L. Therefore, $1 \le x \le M$ and $1 \le y \le N$. Calculate the mean gray value in a $r \times r$ region of each pixel to form a smooth image denoted by $G(x, y)$, which has the same gray level value L. Assume that Num_{ij} is the number of positions where both Pixels with gray level value i in $F(x, y)$ and Pixels with gray level value j in $G(x, y)$ lie. Num_{ij} can be calculated using formula 3.

$$\begin{cases} Num_{ij} \\ = \sum_{x=1}^{M} \sum_{y=1}^{N} \delta[F(x, y) - i] \delta[G(x, y) - j] \\ x = 0, 1, 2, ..., L-1 \\ y = 0, 1, 2, ..., L-1 \end{cases} \quad (3)$$

And then the probability P_{ij} of i and j appearing together can be calculated by formula 4.

$$\begin{cases} P_{ij} = \dfrac{Num_{ij}}{MN} \\ 0 \le P_{ij} < 1, \ \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} P_{ij} = 1 \end{cases} \quad (4)$$

Suppose the threshold vector (s, t) separate 2-dimensional histogram into 4 parts. For the background pixels and the pixels inside the target image, their gray level values are similar as their neighbor mean gray values. Whereas, for the edge pixels of the target, their gray level values and their neighbor mean gray values have a big difference. Assume there are two classes O and B, representing the target and the background. These two classes have different probability density function, written as $P_O(s, t)$ and $P_B(s, t)$. Then the mean vector of Class O, Class B and the 2-dimensional histogram itself are denoted relatively by formula 5.

$$\begin{cases} \boldsymbol{\mu}_O(s,t) = [\boldsymbol{\mu}_{OO}(s,t), \boldsymbol{\mu}_{OB}(s,t)]^T \\ \boldsymbol{\mu}_B(s,t) = [\boldsymbol{\mu}_{BO}(s,t), \boldsymbol{\mu}_{BB}(s,t)]^T \\ \boldsymbol{\mu} = [\boldsymbol{\mu}_O(s,t)^T, \boldsymbol{\mu}_B(s,t)^T]^T \end{cases} \quad (5)$$

Therefore the separation matrix of O and B is able to calculated by formula 6.

$$\boldsymbol{S}(s,t) = \boldsymbol{P}_O(s,t)[\boldsymbol{\mu}_O(s,t) - \boldsymbol{\mu}][\boldsymbol{\mu}_O(s,t) - \boldsymbol{\mu}]^T + \boldsymbol{P}_B(s,t)[\boldsymbol{\mu}_B(s,t) - \boldsymbol{\mu}][\boldsymbol{\mu}_B(s,t) - \boldsymbol{\mu}]^T \quad (6)$$

Define the summation of all elements on the primary diagonal is the trace of the matrix $\boldsymbol{S}(s,t)$, written as $\mathrm{Tr}[\boldsymbol{S}(s,t)]$. The trace of the matrix $\boldsymbol{S}(s,t)$, can be used to measure the separability between Class O and Class B. Finally, the segmentation criterion to determine the most suitable threshold is given by formula 7, which means that $[s_{best}, t_{best}]$ belongs to the set of arguments which maximize the absolute value of function $\mathrm{Tr}[\boldsymbol{S}(s,t)]$.

$$[s_{\text{best}}, t_{\text{best}}] = \arg \max_{1<s<L-1} \max_{1<t<L-1} |\mathrm{Tr}[\boldsymbol{S}(s,t)]| \quad (7)$$

Traditional method to solve formula 7 is enumeration method. Though it is able to help us get an accurate result, its computing complexity $\mathrm{O}(L^4)$ is extremely high. Most of the present methods trying to reduce the computing complexity are exchanging accuracy for speed. This paper proposed a new hybrid optimization algorithm which is based on Gene Algorithm(GA), inserting Particle Swarm Optimization(PSO) and Simulated Annealing Algorithm(SAA) when choosing new gene groups. This algorithm combining the advantages of these three algorithms can reduce the calculation time with a computing complexity no more than $\mathrm{O}(L^2)$ without sacrificing the accuracy. The flow chart of the entire algorithm is in figure 8.

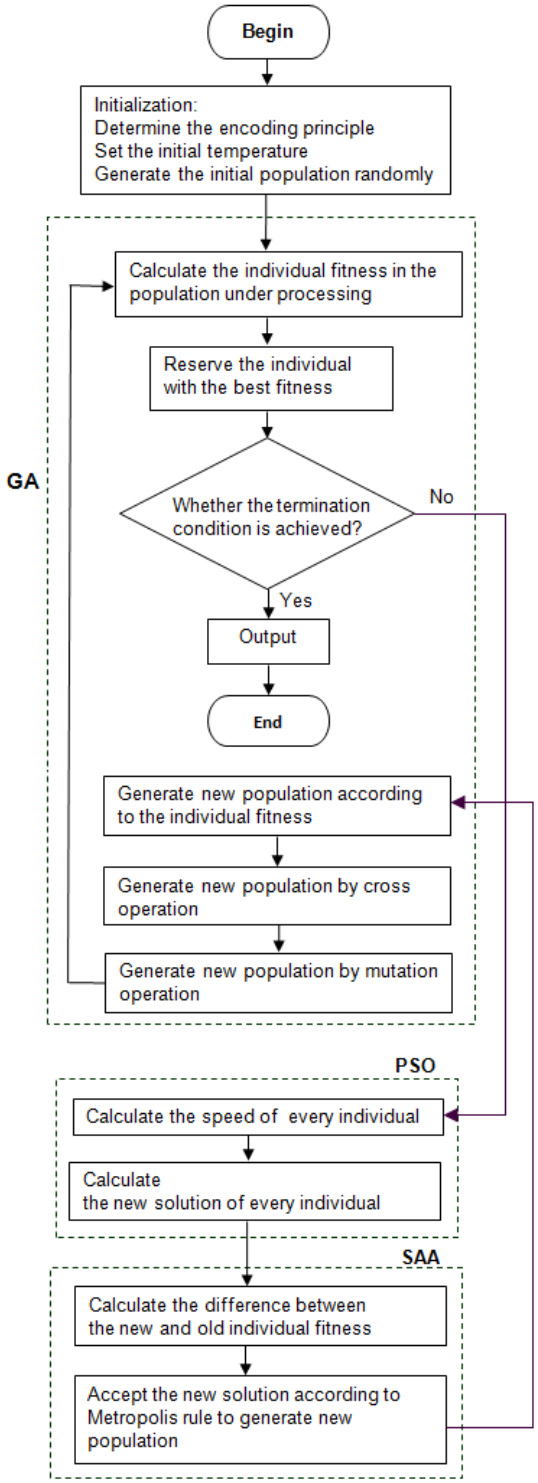


Figure 8. Flow chart of the hybrid optimization algorithm

The important points requiring further discussion are as follows. [18, 19, 20, 21]

(1) Encoding principle

Float-point encoding is adopted in this paper. That is, the gene of every individual is represented by a floating point number with a fixed value range. This encoding method is suitable for a wide range searching and is able to find results with high accuracy. For this problem the chromosome is constructed as $\boldsymbol{X}=[s,t]$ whose length is 2. Seeing in formula 7, $1<s<L-1$ and $1<t<L-1$.

(2) Fitness function

The fitness function plays an very important role when generating new population according to the individual fitness. GA requires that the fitness function can get a positive result for the practical input parameters. For 2-d OTSU algorithm of the image segmentation, the object is to find the best values of s and t which can maximize the absolute value of $\mathrm{Tr}[\boldsymbol{S}(s,t)]$. Therefore, the fitness function is defined as formula 8.

$$C(s,t)=|\mathrm{Tr}[\boldsymbol{S}(s,t)]| \tag{8}$$

(3) Termination condition

Usually, the termination condition of GA is that the maximum fitness or average fitness reaches a steady state or has a very tiny increasment which is set as the judgment value within the scope of the iteration. Suppose N_g is the total number of iterations and n_{g} ($j=2$、.....、N_{g}) represents the iteration being handled. Moreover, we use M_{pop} to represent the gene number in a population and adopt $\mathrm{Pop}_{\mathrm{ng}}(\boldsymbol{X}_{\mathrm{ng1}},\boldsymbol{X}_{\mathrm{ng2}},\boldsymbol{X}_{\mathrm{ng3}},...,\boldsymbol{X}_{\mathrm{ng}M_{\mathrm{pop}}})$ to denote the n_{g}th population. $C_{\max n_{\mathrm{g}}}$ is the maximum fitness in the n_{g}th population and $C_{\max n_{\mathrm{g}}-1}$ is the maximum fitness in the ($n_{\mathrm{g}}-1$)th population. We set a very tiny value C_{chosen} as the judgment value, which is fixed as 0.001 from the real situation in this paper. The termination condition for the problem here is set as $C_{\max n_{\mathrm{g}}}-C_{\max n_{\mathrm{g}}-1}<C_{\mathrm{chosen}}$.

(4) Calculation of the speed for every individual

According to the principle of PSO algorithm, we set the size of the particle swarm the same as the size M_{pop} of the population in GA. The dimensions of a single particle is 2-d on account of the variables number in formula 7. Adopt $\boldsymbol{X}_{\mathrm{np}}=[s_{\mathrm{np}},t_{\mathrm{np}}]$ to represent the n_{p}th particle and $\boldsymbol{v}_{\mathrm{np}}=[v_{\mathrm{np1}},v_{\mathrm{np2}}]$ to represent the speed of the n_{p}th particle in the swarm under handling. Then the n_{g}th swarm can be written as $\mathrm{Swm}_{\mathrm{ng}}(X_{\mathrm{np1}},X_{\mathrm{np2}},X_{\mathrm{np3}},...,X_{\mathrm{np}M_{\mathrm{pop}}})$. We know for the n_{g}th swarm, $\boldsymbol{X}_{\mathrm{npi}}=\boldsymbol{X}_{\mathrm{ngi}}$ ($1\le \mathrm{i}\le M_{\mathrm{pop}}$). Assume the best position of a particle in the swarm under handling is $\boldsymbol{P}_{\mathrm{np}}=[p_{\mathrm{np1}},p_{\mathrm{np2}}]$ and the global best position of a particle is written as $\boldsymbol{P}_{\mathrm{p}}=[p_{\mathrm{p1}},p_{\mathrm{p2}}]$. Therefore, for every single particle the speed will change as the formula 9.

$$\begin{aligned}\boldsymbol{v}_{\mathrm{np}}=\boldsymbol{v}_{\mathrm{np}}&+a_1\times\mathrm{rand}()\times(\boldsymbol{P}_{\mathrm{np}}-\boldsymbol{X}_{\mathrm{np}})\\&+a_2\times\mathrm{rand}()\times(\boldsymbol{P}_{\mathrm{np}}-\boldsymbol{X}_{\mathrm{np}})\end{aligned} \tag{9}$$

In formula 9, a_1 and a_2, two non-negative numbers, are accelerated factors. Usually $0\le a_1\le 4$ and $0\le a_2\le 4$. Here we choose $a_1=a_2=2$. The sign rand() represents the random function whose value belongs to [0,1].

(5) Calculation of the new solution for every individual

Complying with the PSO algorithm, the new solution of each particle in the swarm under handling can be calculated according to formula 10.

$$x'_{\mathrm{np}}=x_{\mathrm{np}}+v_{\mathrm{np}} \tag{10}$$

(6) The rules of accepting the new solution for every individual in SAA

Metropolis proposed a method to accept a new value for a variable by a certain probability. For this problem, assume E is the internal energy at temperature T. The old state of a particle is x_{np}, and the new state of the particle is x'_{np}. When the state of the particle x_{np} changes to x'_{np}, its internal energy changes to $E+\Delta E$ correspondingly. Thus, we

should suppose $E = -C(x_{np})$ and $\Delta E = C(x_{np}) - C(x'_{np})$. If $\Delta E < 0$, x'_{np} is accepted. Otherwise x'_{np} will be accepted in a probability $Pr = e^{-\Delta E / kT}$, where k is the Boltzmann constant.

(7) Generate new population according to the individual fitness

In this step, the particles in the swarm under handling have become the retained x_{np} and the accepted x'_{np} in a certain probability. However, the new population $\mathrm{Pop}'_{ng}(...)$ of the n_gth iteration is asked for a further operation. Putting all the particles of $\mathrm{Swm}'_{ng}(...)$ and all the genes of $\mathrm{Pop}_{ng}(...)$ together, we obtain $2M_{pop}$ individuals. Calculate the fitness of every individual and choose the best M_{pop} ones to generate the new population $\mathrm{Pop}'_{ng}(...)$.Then the following cross operation and mutation operation are based on the population $\mathrm{Pop}'_{ng}(...)$.

Experimental Verification

To verify the improvement of the detection algorithm proposed by this article, 500 original image of ear cross section were detected. The final statistic results of experiments are listed in table 1. And the average detection time of an ear was 0.427s and the precision rate was 100%.

Table 1. Statistic result of experiments

Sample number	Precision rate	Average running time
500	100%	0.427s

In comparision, we have detected the 500 samples using a former automatic algorithm by calculating the distances between the edge and the mass center of the ear cross section and counting the number of minimum values of the distances. The precision rate of the former algorithm was only 98.22%, less than that of this algorithm. Besides, the average running time of the former algorithm was 2.434s, more than 5 times that of this algorithm.

Conclusions

This paper introduced a new detection algorithm to automatically calculate the kernel row number. We proposed a new threshold segmentation method which introduced a hybrid optimization algorithm into 2-d OTSU to obtain the binary contour feature image of an ear cross section. This hybrid optimization algorithm was based on Gene Algorithm (GA). When choosing new gene groups, it inserted Particle Swarm Optimization (PSO) and Simulated Annealing Algorithm (SAA). According to the characteristics of convex hull, we obtained the convex hull image of the ear cross section by calculating the convex hull of the contour of the section with the two-dimension Quickhull algorithm. The kernel gap image showing the gaps between any two adjacent kernels was achieved by subtracting the ear cross section from the convex hull image. The tiny districts in kernel gap image which were formed from the rough and uneven surfaces of the kernels were eliminated using the connected component labeling method. During the method, we labeled each kernel gap and deleted the domain whose area was less than a determinate tiny threshold. Finally we detected kernel row numbers of 500 samples to examine the entire algorithm designed in this paper. In comparison, we also detected the same 500 samples using a former popular automatic algorithm whose precision rate was 98.22% and average running time was 2.434s. However, using the algorithm in this paper, the average detection time of a sample was 0.427s and the precision rate was 100%. The statistic results indicated that the algorithm proposed in this paper was more accurate and had a higher speed.

Acknowledgment

This research was financially supported by Scientific and Technological Research Funding of Henan Province in China(No. 142102210232). And it is also funded by Anyang Institude of Technology and Anyang Academy of Agricultural Sciences in China. Here in particular, I want to give my deepest gratitude to my working partner Jing Li. It is exactly her ideas and efforts that help me successfully complete this paper.

References

1. Pedro M.R. Mendes-Moreira, et al. (2014) Is ear value an effective indicator for maize yield evaluation?

Field Crops Research, Vol. 161, p.p.75-86.
2. Robert Williams, Lourenco Fontes Borges, Myrtille Lacoste, et al.(2012) On-farm evaluation of introduced maize varieties and their yield determining factors in East Timor. Field Crops Research, vol. 137, p.p.170-177.
3. Zhang Huanxin, Weng Jianfeng, Zhang Xiaocong, et al. (2014) Genome-wide Association Analysis of Kernel Row Number in Maize. ACTA AGRONOMICA SINICA, 40(1), p.p.1-6.
4. Agustina Amelong, et al. (2015) Predicting maize kernel number using QTL information. Field Crops Research, Vol. 172, p.p.119-131.
5. Peter Bommert, Namiko Satoh Nagasawa, David Jackson (2013) On-farm evaluation of introduced maize varieties and their yield determining factors in East Timor. Nature Genetics, Vol. 45, p.p.334-337.
6. Zhenghong Yu, Zhiguo Cao, Xi Wu, et al. (2013) Automatic image-based detection technology for two critical growth stages of maize: Emergence and three-leaf stage. Agricultural and Forest Meteorology, Vol. 174-175, p.p.65-84.
7. Han Zhong-zhi, Yang Jinzhong (2010) Vision Research on the Machine of Counting Ear Rows in Maize. Journal of Maize Sciences, 18(2), p.p.146-148.
8. Marco Di Summa (2015) A short convex-hull proof for the all-different system with the inclusion property. Operations Research Letters, 43(1), p.p.69-73.
9. Ming Zeng, Yu Yang, Jinde Zheng, Junsheng Cheng. (2015) Maximum margin classification based on flexible convex hulls. Neurocomputing, 149(B), p.p. 957-965.
10. Mehdi Soltanifar, et al. (2013) On efficiency in convex hull of DMUs. Applied Mathematical Modelling, 37(4), p.p. 2267-2278.
11. Yong-Joon Kima, Myung-Soo Kim, Gershon Elber (2014) Precise convex hull computation for freeform models using a hierarchical Gauss map and a Coons bounding volume hierarchy. Computer-Aided Design, Vol. 46, p.p. 252-257.
12. J. Iverson, C. Kamath, G. Karypis (2015) Evaluation of connected-component labeling algorithms for distributed-memory systems. Parallel Computing, Vol. 44, p.p.53-68.
13. Y-Chuang Chen, et al. (2014) Maximally local connectivity and connected components of augmented cubes. Information Sciences, Vol. 273, p.p.387-392.
14. Oleksandr Kalentev, Abha Rai, Stefan Kemnitz, et al. (2011) Connected component labeling on a 2D grid using CUDA. Journal of Parallel and Distributed Computing, 71(4), p.p. 615-620.
15. Lifeng He, Yuyan Chao, Kenji Suzuki, et al. (2009) Fast connected-component labeling. Pattern Recognition, 42(9), p.p.1977-1987.
16. Akira Mizushima, Renfu Lu (2013) An image segmentation method for apple sorting and grading using support vector machine and Otsu's method. Computers and Electronics in Agriculture, Vol. 94, p.p.29-37.
17. Xiangyang Xu, Shengzhou Xu, Lianghai Jin, Enmin Song (2011) Characteristic analysis of Otsu threshold and its applications. Pattern Recognition Letters, 32(7), p.p.956-961.
18. Marcela Quiroz-Castellanos, et al. (2015) A grouping genetic algorithm with controlled gene transmission for the bin packing problem. Computers & Operations Research, Vol. 55, p.p.52-64.
19. Hai Min, et al. (2015) An Intensity-Texture model based level set method for image segmentation. Pattern Recognition, vol. 48(4), p.p.1547-1562.
20. Limin Zhang, et al. (2015) A new particle swarm optimization algorithm with adaptive inertia weight based on Bayesian techniques. Applied Soft Computing, Vol. 28, p.p.138-149.
21. Lars Junghans, Nicholas Darde (2015) Hybrid single objective genetic algorithm coupled with the simulated annealing optimization method for

building optimization. Energy and Buildings, Vol. 86, p.p.651-662.