Application of Particle Swarm Optimization in Optimizing Stereo Matching Algorithm's Parameters for Star Fruit Inspection System

Saidatul Nizan Nasroddin, Musa Mohd Mokji, Tan Kok, Amar Faiz Zainal Abidin, Rahman Amirulah, Nur Anis Nordin, Saipol Hadi Hasim, Hamzah Zakaria, Jefery Hassan, Hazriq Izzuan Jaafar, and Khairuddin Osman

***Abstract*—**This paper reports the finding of the experimentation of the Particle Swarm Optimization in optimizing the stereo matching algorithm's parameters for the star fruit inspection system. The star fruit inspection system is built by CvviP Universiti Teknologi Malaysia. While the stereo matching algorithm used in the experiment is taken from the Matlab library. Each particle of Particle Swarm Optimization in the search pace repsents a set of candidate numerical value of the stereo matching's parameters. The fitness function for this application is the sum of absolute error of the gray scale value of both images. Based on this information, the particles will improve its position in the search space by moving towards its best record and the swarm best record. The process repeated until the maximum iteration met. The result indicates that there is potential application of Particle Swarm Optimization in stereo matching's parameters tuning.

***Keywords*—** particle swarm optimization, fruit inspection, stereo matching algorithm, swarm intelligence.

I. Introduction

STAR fruit is one of the most popular fruits exported by Malaysia. From 1965, Federal Agricultural Marketing Authority (FAMA) is given the authority to regulate the quality of the star fruit exported by Malaysia [1]. The great interest among the importers of Malaysia's star fruit becomes a great motivation to increase the yield and quality of the star fruit for export. One of the areas where the process can be improved is the fruit inspection [2]. Universiti Teknologi Malaysia Computer Vision, Video and Image Processing Research Group (CvviP) had successfully invented an automatic star fruit grading system. Now, CvviP try to extend the system by experiment the application of stereo vision in star fruit inspection. Disparity map obtained from the stereo vision can be used to find the size of the star fruit. In this article, the authors will explain the implementation of Particle Swarm Optimization (PSO) in optimizing the stereo matching algorithm's parameters. Based on the result obtained, application of PSO improves the selection of parameters in stereo matching algorithm, although further study is required for practical implementation.

Khairuddin Osman, Saipol Hadi Hasim, Musa Mohd Mokji, Tan Kok, Rahman Amirulah, Faculty of Electrical Engineering, Universiti Teknologi Malaysia, 81300 Johor Bahru, Malaysia.

Saidatul Nizan Nasroddin, Hamzah Zakaria, Jefery Hassan, Department of Electrical Engineering, Politeknik Ibrahim Sultan, 81700 Pasir Gudang, Malaysia.

Nur Anis Nordin, Faculty of Business Management, Universiti Teknologi MARA, 40000 Shah Alam, Malaysia.

Amar Faiz Zainal Abidin, School of Science & Technology, Wawasan Open University, 10500 Pulau Pinang, Malaysia.

Hazriq Izzuan Jaafar, Faculty of Electrical Engineering, Universiti Teknikal Melaka, 76100 Durian Tunggal, Malaysia.

II. Methodology

The main objective of this project is to obtain disparity value of the star fruit from two images taken from the hardware setup. The proposed approach consists of two main parts: hardware and software implementation. Section II.A will explain the hardware implementation of the proposed approach. The hardware implementation covered the hardware setup and image acquisition model. In Section II.B will explain the software implementation.

A. Hardware Implementation

Fig. 1 shows the hardware setup of the proposed approach. The hardware setup consists of several components: vision housing structure, camera holder, camera, conveyor, conveyor structure, vision housing, lamp, mirror, slider, personal computer platform, and personal computer.

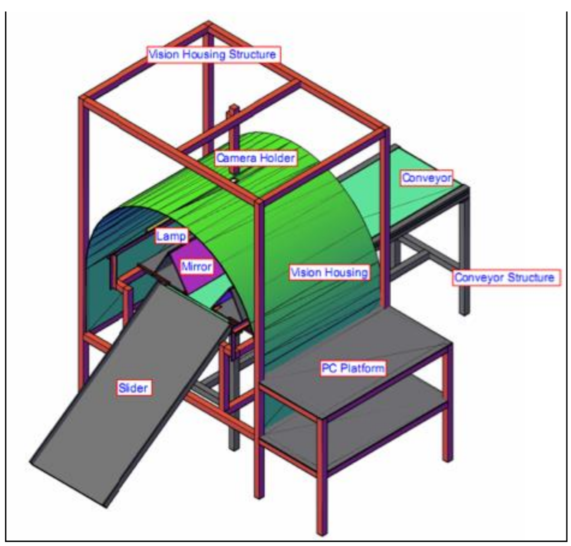

Fig. 1 Hardware setup of the proposed approach

The vision housing structure functions as the supporting housing structure of the inspection system. This to ensure the inspection system has a firm structure, which is practical for heavy-duty use. The camera holder holds and fixed the camera location for images acquisitions. The position of the camera holder can be move in all of three axis. Camera as we known, is use to acquire images of the star fruit. The camera use here is Canon which is connected to the personal desktop computer for images acquisition. Conveyor is use to move the star fruit to a fixed points for image acquisitions. This to ensure that both images taken have then fixed position of x-axis. Conveyor structure is a supporting structure of the conveyor. Lamp is use to provide lighting to the star fruit for image acquisitions. Unfortunately, the lighting of the lamp is not carefully design, thus produces uneven illumination on the conveyor belt. Personal computer is use to acquire images and performs disparity computation. The personal computer platform is the location where the personal computer is placed. Inspected fruits will be transport into a box via the slider which located at the end of the inspection system. The actual image is as shown in Fig. 2.

As shown in Fig. 2, the inspection system uses only one camera. The main advantage of having a single camera is that the cost of the inspection system is relatively cheaper. The main disadvantage of a single camera system is that the time taken to acquire two images increased. Dual images acquisition with a single camera vision can be done by model shown in Fig. 3. Firstly, the star fruit is position at the desired location and the first image acquired. Then, the conveyor will move the star fruit according to the desired baseline value. Lastly, the second image acquired before going into the software implementation.

The single camera modeling as shown in Fig. 4 where Z is the coordinate of z axis of the object in the real world, f is the focal point, d is the disparity in pixel, α is the size of the pixel sensor, and T is the baseline. $m_1$andm_2 are the center location of the star fruit in two locations. Thus, Z can be obtained from Equation 1

$$Z = \frac{fB}{d\alpha} \qquad (1)$$

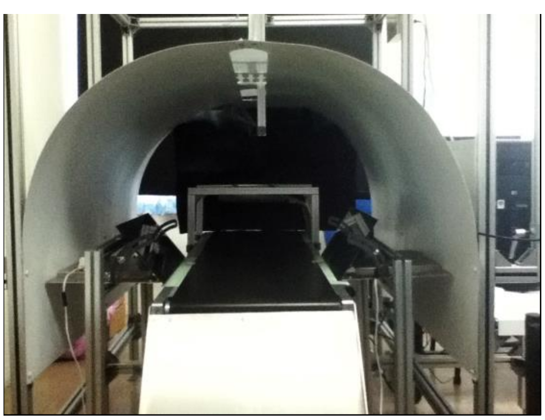

Fig. 2 Actual image of the proposed hardware setup

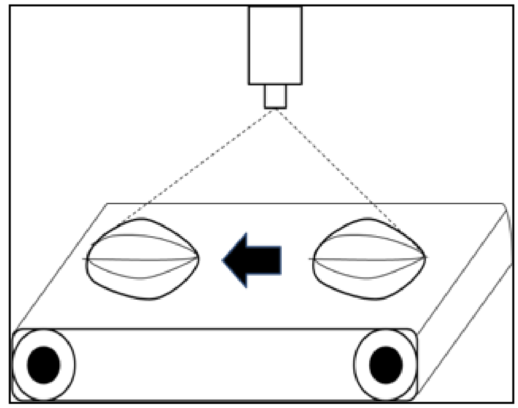

Fig. 3 Dual images acquisition using single camera

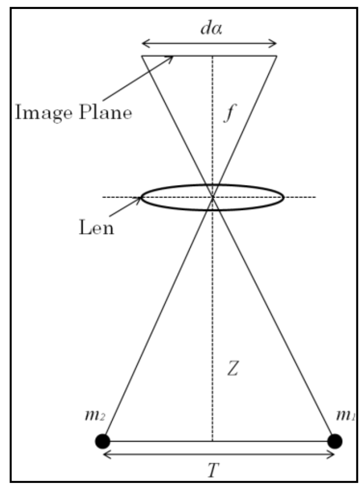

Fig. 4 Single camera modeling for stereo vision

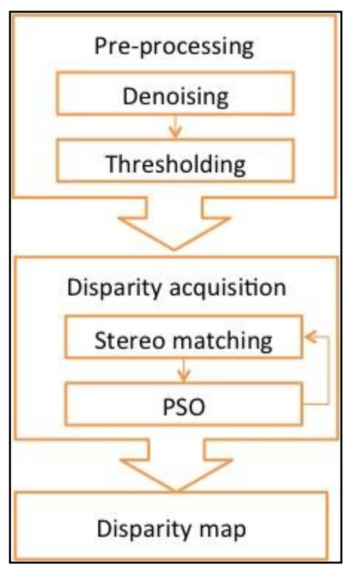

Fig. 5 Steps in software implementation

B. Software Implementation

In software implementation the system proposed consists of three main steps which are pre filtering, disparity computation, and mapping the disparity. This is as illustrated in Figure 5. The first step is the pre-processing process. The main objective of this process is to prepare images for stereo matching. There are two processes involved in pre-processing process: de-noising and thresh-holding. The objective of the de-noising step is to reduce noise of the image due to uneven lighting, or illumination. Illumination can be roughly estimate as Equation 2

$$I_i = I_{bl} - I_{bw} \quad (2)$$

Where I_i is intensity of image with illumination, $I_{bl}$is intensity of background image with lamp on, and $I_{bw}$is light intensity of background with lamp off. Thus, the denoise object image, I_t can be obtained by Equation 3.

$$I_t = I_o - I_i \quad (3)$$

Where I_O is intensity of object image with illumination.

Then, the thresh-holding step is done to get the region of interest. In other words, this process separates background from the object. Here the threshold value is set at 10% of the maximum value of the Red-Green-Blue (RGB) color scheme. The main objective is to reduce computation time of the stereo matching algorithm by computing disparity for region of interest only, instead of the entire image. Also, the background can be easily flag out with unreliable disparity computation if using disparity function in MATLAB R2012a. Then, the region of interest (ROI) is converted from RGB color scheme into gray scale. This is because, the stereo matching algorithm use in this project only compare the gray level between the two pixels.

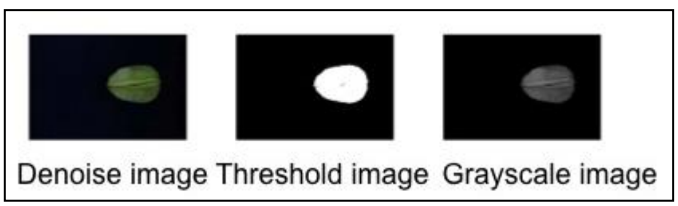

Fig. 6 (a) De-noise image. (b) Threshold image. (c) Gray scale image

The second process is the disparity acquisition where it consists of the stereo matching algorithm and optimization stereo matching algorithm using PSO. The stereo matching algorithm here is the "Sum of Absolute Difference" (SAD) local stereo matching algorithm provided by MATLAB R2012a [3]. The algorithm starts by computing a measure of the contrast of the image using Sobel filter. Then disparity is computed based on block matching with SAD as the matching cost and "Winner-Take-All" rule. There are seven parameters involved in the stereo matching algorithm, which are: contrast threshold (s_1), block size (s_2), disparity range (s_3 for minimum value and s_4 is the maximum value), texture threshold (s_5), uniqueness threshold (s_6), and distance threshold (s_7). Description of these parameters can be obtained from MATLAB R2012a [3].

The parameters in the stereo matching algorithm are problem specific; we can use PSO to optimize these parameters. Algorithm 1 is the adaptation of the generic Particle Swarm Optimization algorithm for optimizing the parameters. The particle position can be modeled as Equation 4.

$$\boldsymbol{x} = [s_1, s_2, s_3, s_4, s_5, s_6, s_7]^T \quad (4)$$

Where s_1 to s_7 bounded by constraints shown in Equation 5 to Equation 10.

$$0 < s_1 \leq 1 \quad (5)$$

$$5 \leq s_2 \leq 255,\ s_2 = 5 + 2n,\ n = 0,..,125,\ s_2 < I_X,\ s_2 < I_Y \quad (6)$$

$$\frac{s_4 - s_3}{16} = n, n \in \mathbb{N}, s_3 \in \mathbb{Z}, s_4 \in \mathbb{Z}, s_3 < s_4,\ s_3 < I_X,\ s_4 < I_X \quad (7)$$

$$0 \leq s_5 \leq 1 \quad (8)$$

$$s_6 \in \mathbb{N} \quad (9)$$

$$s_7 \in \mathbb{N} \quad (10)$$

Algorithm 1: PSO Algorithm for parameters tuning in Stereo Matching Algorithm

Initialize all particles with a random position and velocity in the search space based on model in Equation 4 and bounded by constraints (Equation 5 to Equation 10)
while stopping condition not met
  for each particle **do**
    Calculate the fitness of the particles using Equation 20
    if particle fitness better than previous $\boldsymbol{pbest}$ **then**
      Set particle fitness value as new$\boldsymbol{pbest}$
    end if
    if particle fitness value better than the current $\boldsymbol{gbest}$ **then**
      Set fitness value as the new$\boldsymbol{gbest}$
    end if
  end for
  for each particle **do**
    Update particle velocity according to Equation 18
    Update the particle position according to Equation 19
    Perform correction if the updated particle position does not meet the constraint requirement
  end for
end while
Present $\boldsymbol{gbest}$ solution

For binary version of PSO, each parameters is represented using 10 bits as shown in Equation 11.

$$\boldsymbol{x} = \begin{bmatrix} s_{1,b=1} & \cdots & s_{1,b=10} \\ \vdots & \ddots & \vdots \\ s_{7,b=1} & \cdots & s_{7,b=10} \end{bmatrix} \quad (11)$$

Each particle of the parameter t, p_t can have a discrete value from 0 to 1023 as shown in Equation 12.

$$p_t = 0,1,2,..,1023 \quad (12)$$

As the constraints of each parameter are different to each other, then each parameter has different binary representation as shown in Equation 13 to Equation 19.

$$s_1 = \frac{(1+p_{s_1})}{1024} \quad (13)$$

$$s_2 = 5 + \left\lceil \frac{250 \times (1+p_{s_2})}{1024} \right\rceil, s_2 < I_X, s_2 < I_Y \quad (14)$$

$$s_3 = 16 \times (512 - p_{s_3}), s_3 < I_X \quad (15)$$

$$s_4 = 16 \times p_{s_4}, s_4 < I_X \quad (16)$$

$$s_5 = \frac{p_{s_5}}{1023} \quad (17)$$

$$s_6 = p_{s_6} \quad (18)$$

$$s_7 = p_{s_7} \quad (19)$$

The fitness function use to calculate the fitness is the entire region of interest (the star fruit) which as shown in Equation 20.

$$\sum_{(x,y)\in ROI} |I_L(x,y) - I_R(x + d_{ROI}, y)| \quad (20)$$

The objective of the implementation of PSO is to find the values of the parameters that minimize the value in Equation 12. By assumption, this should give the best image as the sum of difference between the intensity of the pixels of the two images is at minimum level. After the particles update their position in line 14, Algorithm 1, the new solutions, they represented might invalid due to the constraints given by the parameters. This can be solve by correcting the particle position to the nearest feasible solution.

The last process is to display the disparity map. The disparity map is display using color tone where red indicates a high displacement between the two images, while bright blue indicates there is a small displacement between the two images. As shown in Equation 1, high disparity indicates that the object is near with the camera, vice versa

III. Result And Discussion

Figure 7 and Figure 8 show the raw images taken from the images acquisition system. It can be seen that there still uneven lighting or illumination from the two images.

Fig. 7 Example of the left-camera image

Fig. 8 Example of the right-camera image

The disparity obtained from the two images above as shown in Fig. 9. Each computation takes around 10 seconds to complete (excluding PSO). It can be seen that the basic stereo matching algorithm does not perform well especially around the edges: boundary region of interest and the slanted region. This is because a window-based block matching assumes constant depth within the window. Poor lighting also is one of the causes of poor performance of the stereo matching algorithm. Uneven lighting between the two images makes the stereo matching algorithm difficult to match the pixels of the images. This is because the intensity of both images varies according to its location due to the illumination.

As stated earlier, parameters in stereo vision matching algorithm can be optimized using any global optimization algorithm such as PSO. Although, time taken to find the optimized parameters is long in general, optimization algorithm usually need to be done once only. Table 1 listed the parameters are used to study the performance of PSO and Binary PSO (BPSO). In this experiment, the inertia weight use is a linear decreasing inertia weight which as Equation 21.

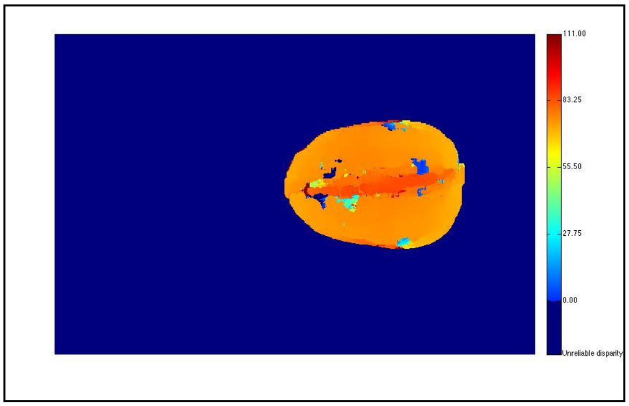

Fig. 9 Disparity map obtained from the proposed approach

$$\omega = \omega_{max} - (\frac{\omega_{max} - \omega_{min}}{t_{max}} \times t) \quad (21)$$

where ω_{max} and ω_{min} represent the maximum and minimum value of the inertia weight. t_{max} is the maximum iteration.

Fig. 9 shows the result obtained by PSO while Fig. 10 shows the disparity map obtained by BPSO implementation, ceteris paribus.

TABLE I
PSO AND BPSO PARAMETERS

Parameters	Quantity
Number of agents	50
Number of iterations	500
Number of computations	10
Inertia weight	0.9 → 0.4
Cognitive component	1.42
Social component	1.42

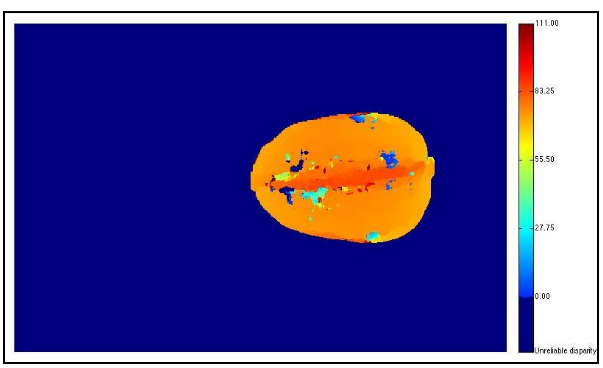

Fig. 10 Disparity map obtained using BPSO

Visually, there is not much difference between these two disparity maps. Disparity map obtained from PSO has less bad disparity matching compared to the BPSO. This is because PSO proposed a solution with a larger window compared to BPSO. There also moment where both algorithms PSO and BPSO provide unacceptable solution. This can be minimized by guiding the algorithms by setting the best found parameters by human as one of the particle position solution.

The selection of inertia weight, social coefficient, and cognitive coefficient values are based on the recommendation of several literatures [4-7]. The selection of number of agents, and number of iteration are based on the experimental result which shown in Table I.

In term PSO and BPSO performance, the iteration taken for BPSO to convergence and obtained optimized solution is faster than PSO. But if we compared using time, there is no much difference between these two algorithms. Both algorithms took around 10 hours to complete a simulation. Note that the simulation is done using 1.73GHz Intel Centrino Core 2 Duo with 2GB RAM. The reason is because even BPSO convergence faster, but the algorithm is more complex than PSO. Thus, BPSO is more computational expensive than PSO which leads to additional time required per iteration. Table II indicates the experimental result obtained from PSO and BPSO for ten computations.

TABLE II
EXPERIMENTAL RESULT OBTAINED FROM PSO AND BPSO

Criteria	PSO	BPSO
Best found fitness	3725248	3817866
Average fitness	4254618	4541112

IV. CONCLUSION

This paper shows the potential application of PSO in tuning parameters of stereo matching algorithm. The experiment result indicates there is a potential of the application of PSO or other optimization strategies in tuning parameters in stereo matching algorithms. Further study can be done by experimenting different variations of optimization strategies.

REFERENCES

[1] R. Amirullah, M. M. Mokji. Z. Ibrahim, "Starfruit Color Maturity Classification Using Cr as Feature," in *Proceeding of Signal-Image Technology and Internet-Based Systems*, 93-97, 2010.
[2] M. M. Mokji, S. A. R. A. Bakar, "Starfruit Grading Based on 2-Dimensional Color Map," in *Proceeding of Regional Postgraduate Conference on Engineering and Science*, 203-206, 2006.
[3] Matlab R2012a. disparity() function. URL: http://is.gd/fvTcA9 [Accessed 29 May 2012].
[4] S. Naka, T. Genji, T. Yura, Y. Fukuyama, "Practical Distribution State Estimation using Hybrid Particle Swarm Optimization," In IEEE Power Engineering Society Winter Meeting, vol. 2, pp. 815-820, 2001.
[5] A. Ratnaweera, S. Halgamuge, H. Watson, "Particle Swarm Optimization with Self-Adaptive Acceelration Coefficients," in Proceeding of the First International Conference on Fuzzy Systems and Knowledge Discovery, pp. 264-268, 2003.
[6] P. N. Suganthan, "Particle Swarm Optimizer with Neighborhood Operator," in *Proceedings of the IEEE Congress on Evolutionary Computation*, pp. 1958–1962, 1999.
[7] H. Yoshida, Y. Fukuyama, S. Takayama, S., Y. Nakanishi, "A Particle Swarm Optimization for Reactive Power and Voltage Control in Electric Power Systems Considering Voltage Security Assessment," In *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, vol. 6, pp. 497–502, 1999.