British Journal of Mathematics & Computer Science
9(1): 40-61, 2015, Article no.BJMCS.2015.186
ISSN: 2231-0851

SCIENCEDOMAIN *international*
www.sciencedomain.org

A Proposed Method to Improve Quality of Image Using Panorama Technology for WSNs

The Nguyen Manh[1*], Phat Nguyen Huu[1] and Vinh Tran-Quang[1]

[1]School of Electronics and Telecommunications, Hanoi University of Science and Technology, Hanoi, Vietnam.

Article Information

DOI: 10.9734/BJMCS/2015/17632
Editor(s):
(1) Victor Carvalho, Polytechnic Institute of Cávado and Ave, Portuguese Catholic University and Lusiada University, Portugal.
Reviewers:
(1) Anonymous, Jordan.
(2) Stepan Bilan, State Economy and Technology University of Transport, Kiev, Ukraine.
(3) Anand Nayyar, Department of Computer Applications & IT, KCL Institute of Management and Technology, Punjab, India.
(4) Anonymous, Universiti Sains, Malaysia.
Complete Peer review History: http://www.sciencedomain.org/review-history.php?iid=1142&id=6&aid=9236

Method Article

Received: 20 March 2015
Accepted: 22 April 2015
Published: 13 May 2015

Abstract

Aims: Applying an image processing algorithm (image mosaic) for Wireless Sensor Networks (WSNs) and detect the position of the object of the picture.
Methodology: The method that we propose includes four steps: Detecting key points, matching key points and cutting overlaps, finding Homography matrix, and combining with RANdom Sample Consensus (RANSAC). In WSNs, a number of nodes are large. We use distributed processing on nodes by using multi-path routing and cutting overlaps. The energy consumption of the system will be reduced and balanced since lifetime of nodes is longer.
Results and Conclusion: The results show that the proposed method improves not only quality of images but also energy consumption and lifetime of networks.

Keywords: Panorama; combining; image stitching; SIFT; RANSAC; homography; ghost; detecting.

1 Introduction

Wireless Sensor Network (WSN) is a system of wireless sensors usually random development of special conditions without depending on human to monitor changes in the environment, to detect the abnormality in the works or to collect information about a region in the defined period. To communicate each other, the nodes need to organize themselves in a stable environment with a reserve energy source.

**Corresponding author: manhthe9x@gmail.com;*

In the network, power of nodes is limited. WSNs have a lot of nodes, it is not difficult to deploy in a large area. Besides, we will obtain images with different angles and locations at the sink node. Using this feature, we propose to apply image processing algorithm (image mosaic) for WSNs. The basic idea of this algorithm is to reconstruct a large image from small images to create a clear picture for surveillance applications.

Recently, many applications require image transmission over WSNs. However, it is rather impossible to capture a high quality image with a limited resource of node. Besides, the current angle of the cameras is limited at about 110°×90° compared to the human eye perspective of reaching 200º×135º [1]. Therefore, one image will not store much information. To solve this problem, we perform to combine several images that have the part of object or landscape to form a larger panoramic image known as Panorama image [2] with the angle can be up to 360°.

The objective of the paper is to rebuild the large image by a system of sensor nodes and determine the location of objects in images with high accuracy. Because of limited resources of sensor (its energy and processor), this objective is a major challenge. Therefore, this paper will propose the Panorama algorithm for image restoration in the context of optimizing the energy consumed on networks.

Other application of Panorama in sensor networks is to detect the position of the object of the image. We assume that the object appears in both images but we cannot determine the exact object from a single image. Therefore, detecting the location of the object of the image is not correct. In the case, the stitching Panorama will result in sharper objects and identify objects more accurately.

The remainder of this paper is organized as follows. Section II introduces the Panorama technique. Section III introduces the network and energy consumption models. In Section IV, we evaluate the proposed scheme and analyze the results. The last section presents the conclusion and directions for future work.

2 Methodology 1: Panorama Technology

Panorama image technique includes the following steps:

Step 1: Extracting the features between the two images
Step 2: Matching key points
Step 3: Cutting parts of two identical photos
Step 4: Calculating the matrix of Homography
Step 5: Robust Homography Estimation using RANSAC
Step 6: Combining the two images together

2.1 Extracting the Features

2.1.1 Detecting key points

The features points of image are pixels that contain more information than the neighboring pixels. Using the points will improve the size of image by points more succinct, and thus reducing the searching space in the image processing applications.

We use the algorithms to find invariant features scale invariant feature transform (SIFT) points [3]. The feature does not change when rotating, stretching or changing the light intensity of the image. In this step, we perform several small steps as follows:

2.1.1.1 *Scale-invariant feature detection*

Attractions with SIFT features are compatible with the local extreme difference of Gaussian filter (DoG) at different rates [3,4]. The scale space of an image is defined as a function $L(x, y, k\sigma)$ that is described as follows:

$$L(x, y, k\sigma) = G(x, y, k\sigma) * I(x, y), \tag{1}$$

where $G(x, y, k\sigma)$ is variable scale Gaussian, I(x, y) is an input image, and $*$ is the convolution multiplication between x and y. We have:

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}, \tag{2}$$

The local extreme is identified as maximum or minimum of the image. "Difference of Gaussian" (difference of Gaussian lines) (DOG) is performed by two Gaussian blurs on images with each command as different blurring radius as shown in Fig. 1, and then subtracting them together to give the results as shown in Fig. 2.

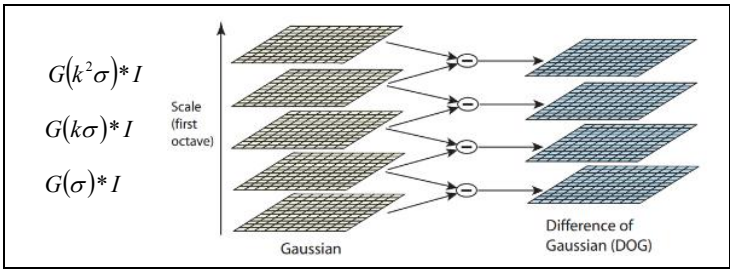

Fig. 1. Scale-space octaves [3]

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \tag{3}$$

For each octave of scale space, the initial image is repeatedly convolved with Gaussians to produce the set of scale space images shown on the left. Adjacent Gaussian images are subtracted to produce the difference-of-Gaussian images on the right. After each octave, the Gaussian image is down-sampled by a factor of 2, and the process is repeated [3].

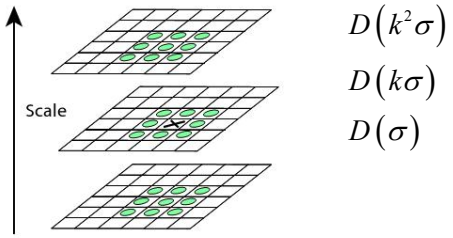

Fig. 2. Three DOGs [3] $\left(D(k^2\sigma), D(k\sigma), D(\sigma)\right)$

Each pixel of DOG is compared with its eight neighboring pixels in the same proportion and nine adjacent neighbors in the rate immediately which stands before and after it. If that pixel reaches the minimum value or the maximum, it will be selected as the key points.Each key point will be selected after evaluating whether or not retainingits.

2.1.1.2 Removing key points that have low contrast

We uses the Taylor expansion (up to the quadratic terms) of the scale-space function, D(x, y,σ), shifted so that the origin is at the sample point:

$$D(x) = D + \frac{\partial D^T}{\partial x} x + \frac{1}{2} x^T \frac{\partial^2 D}{\partial x^2} x, \tag{4}$$

where D and its derivatives are evaluated at the sample point and $x = (x, y, \sigma)^T$ is the offset from this point.

We minimize to get true location of extreme as follows

$$\hat{x} = -\frac{\partial^2 D^{-1}}{\partial x^2} \frac{\partial D}{\partial x}. \tag{5}$$

Then, we reject points with bad contrast if $D(\hat{x}) < 0.03$.

2.1.1.3 Removing key point that is located along the its edges because it does not hold stability when noising and rotating image

Using Trace and Determinant of Hessian, we have

$$He = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}, \tag{6}$$

Where *He* is Hessian matrix, D_{xx} is second partial derivative in the x direction and D_{xy} is the mixed partial second derivative in the x and y directions.

To check ratio of principal curvatures, we use the equation

$$\frac{Tr(He)^2}{Det(He)} < \frac{(r+1)^2}{r}, \tag{7}$$

where r=10 [3].

2.1.2 Orientation assignment

The step is described in Fig. 3. In Fig. 3 (a), key point location is determined as extreme location and key point scale is the scale of the DOG image. In Fig. 3 (b), we perform to calculate gradient of Gaussian image in two cases, gradient magnitude and gradient orientation as follows:

$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2} \tag{8}$$

where m(x,y) is the gradient magnitude.

$$\theta(x,y) = \tan^{-1}\left(\frac{L(x,y+1) - L(x,y-1)}{L(x+1,y) - L(x-1,y)}\right) \tag{9}$$

where $\theta(x, y)$ is orientation

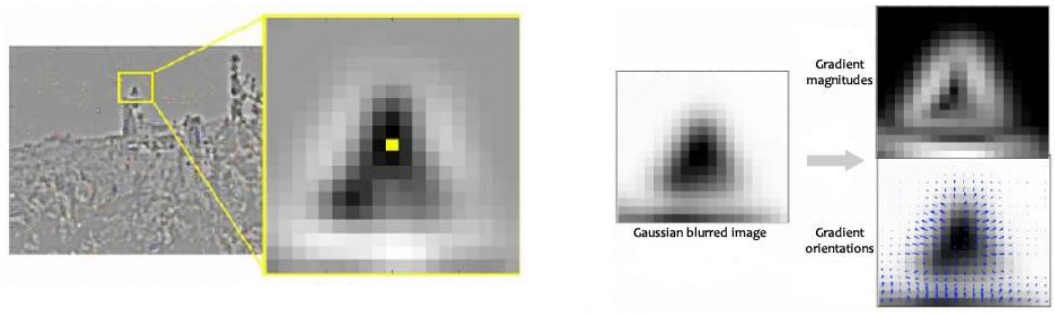

Fig. 3. Orienting for key points [3]
(a) orientation collection region and (b) gradient magnitude and gradient orientation.

2.1.3 Keypoint descriptor

A keypoint descriptor is created by computing the gradient magnitude and orientation at each image that has sample point in a region around the keypoint location, as shown on Fig. 4. They are weighted by a Gaussian window indicated by the overlaid circle. These samples are then accumulated into orientation histograms summarizing the contents over 4×4 subregions, as shown the length of each arrow corresponding to the sum of the gradient magnitudes near that direction within the region. Arrays that contain synthetic gradient are created as follows:

$$number\ of\ dimensions = 8\ (dimensions) \times (4 \times 4) = 128 \tag{10}$$

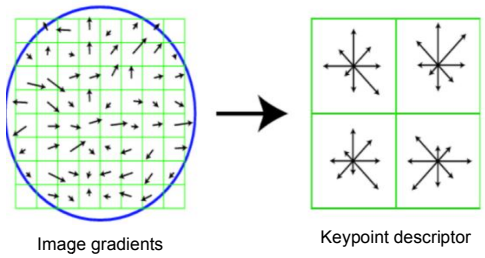

Fig. 4. Key point descriptor [3]

2.2 Matching Key Points [5]

In the step, we compare of key-point descriptors among the images. Two points are considered similarly if the Euclidean distance between two points is the smallest, and the ratio between the closest distance and the second closest distance is less than a given threshold. We chose the threshold as 0.8 [3].

Euclidean distance is calculated as follows [6]:

$$\begin{aligned} d(p,q) &= \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \ldots + (p_n - q_n)^2} \\ &= \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2}, \end{aligned} \tag{11}$$

Where

n = 128 for each point is characterized by 128 dimensions.

2.3 Cutting Parts of Two Identical Photos

The implementation of the Panorama algorithm according to the diagram is shown in Fig. 5. The downside is performed when the input image overlapping appears the "ghosting". Besides, the calculation volume is larger than other methods.

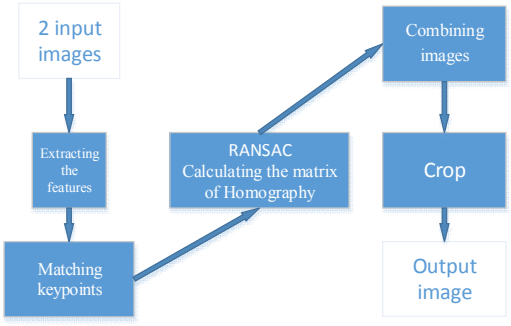

Fig. 5. Block Diagram old [7]

Output image is:

$$I_{out}(x,y) = I_1(x,y) + I_0(x,y) + H * \left(I_2(x,y) + I_0(x,y) \right) \tag{12}$$

where

$I_1(x,y) + I_0(x,y)$ and $I_2(x,y) + I_0(x,y)$ are two input images.

$I_0(x,y)$ is the overlap between the two pictures. ***H*** is matrix of Homography. In our proposal, we remove the overlap between the two pictures before combining them as shown in Fig. 6.

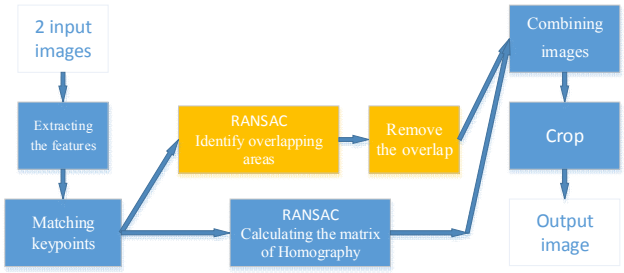

Fig. 6. Block diagram new

In the figure, output image is calculated as follows:

$$I_{out}(x,y) = I_1(x,y) + I_0(x,y) + H * I_2(x,y) \tag{13}$$

2.3.1 Identify overlapping areas

After finding the similarities (matching key points), overlapping region between the two pictures is the region where contains many similarities. However, there are still misidentified points called outlier noises.

In the step, we use RANSAC ("*RANdom SAmple Consensus"*) algorithm [11] to determine the exact overlapped region between the two images, and eliminating adverse effects of outlier noise and limiting errors.

Fig. 7. Ghosting in red area

The idea of the algorithm performs as follows:

From the set of input data that have two types of data, "inlier" and "outlier", in which "inlier" the data is not noisy and "outlier" is the data noise, we carry out calculating and finding the best model for the set of data. The calculation and selection the best model will be repeated *k* times, where *k* values are chosen large enough to ensure that the probability *p* (typically 0.99) of the random sample data set does not contain "outlier".

Algorithm 1: Identifying overlapping areas

1: ***while****i< k* ***do***
2: Choose an image area on the second image
3: Count the number of similarities in that area
4: Calculate the number m satisfying the conditions set inlier inlier score greater than 80% of the total number of similarities found
5: ***if*** *inlier >max (inlier)* ***then***
6: max(inlier) = inlier and the overlapping areas is identified

$$7: k = \frac{\log(1-p)}{\log(1-w^n)} \tag{14}$$

where w is the ratio between the total number of inlier points. We select 80%

8: end if
i=i+1;
9: end while

2.3.2 Remove the overlap

Remove the overlap on the first image.

2.4 Calculating the Matrix of Homography [8]

Homography is a mathematical definition. That is shift using projection geometry, or it is the combination of pairs of points in perspective projection in other way. Real image in 3D space can be changed by mapping

space through Homography transformation matrix (***H***). The projection matrix transforms through Homography do not guarantee the size and angle of the reference objects, but guarantee their rate as follows:

$$\boldsymbol{X'= HX,} \tag{15}$$

where ***H*** is calculated as direct linear transform DLT (Direct Linear Transformation).

In the inhomogeneous coordinates, Eq. (15) can be rewritten as follows:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{16}$$

where $h_9 = 1$ because space is 2D.

The first line is divided into the third line and the second line is also divided into to the third line, we obtain the following expression:

$$-h_1 x - h_2 y - h_3 + \left(h_7 x + h_8 y + h_9\right) u = 0 \tag{17}$$

$$-h_4 x - h_5 y - h_6 + \left(h_7 x + h_8 y + h_9\right) v = 0 \tag{18}$$

Eqs. (17) and (18) can be written as follows:

$$A_i h = 0 \tag{19}$$

where:

$$A_i = \begin{bmatrix} -x & -y & -1 & 0 & 0 & 0 & ux & uy & u \\ 0 & 0 & 0 & -x & -y & -1 & vx & vy & v \end{bmatrix}, \tag{20}$$

$$\text{and } h = \left(h_1\ h_2\ h_3\ h_4\ h_5\ h_6\ h_7\ h_8\ h_9\right)^T, \tag{21}$$

For each pair of corresponding points, we have two expressions. Therefore, we only need 4 pairs of points to determine the matrix ***H*** [9,10] for our matrix (***A***), and we have:

$$A = U \sum V^T = \sum_{i=1}^{9} s_i u_i v_i^T, \tag{22}$$

where s_i is the single value and s_i is arranged by descending order. Therefore, s_9 is the smallest value and the value of h_i is the final value of the v_i column as follows : $Ah = 0$,

$$A = UDV^T = U \begin{bmatrix} d_{11} & \cdots & d_{19} \\ \vdots & \ddots & \vdots \\ d_{91} & \cdots & d_{99} \end{bmatrix} \begin{bmatrix} v_{11} & \cdots & v_{19} \\ \vdots & \ddots & \vdots \\ v_{91} & \cdots & v_{99} \end{bmatrix}^T, \tag{23}$$

$$h = \frac{[v_{19}, \cdots, v_{99}]}{v_{99}}, \tag{24}$$

Finally, we divide results for v_{99} and have h_{99}= 1.

2.5 Robust Homography Estimation using RANSAC [11]

Algorithm 2: Selecting algorithm H matrix:

1: ***while*** *i< k* ***do***
2: Choose 4 random pair similarities
3: Calculate Homography matrix H_{tmp} *based on 4 points as described above*
4: Calculate the distance d of the set of pairs of similarities as follows

$$d_i = d\left(\vec{X}_i', \mathrm{H}_{tmp}\vec{X}_i\right) + d\left(\vec{X}_i', \mathrm{H}_{tmp}\vec{X}_i\right) \tag{25}$$

5: Calculate the number of m "inlier" point pairs satisfied the conditions d_i*< threshold*
6: ***if*** *inlier >max(inlier)* ***then***
7: max(inlier) = inlier and H = H_{tmp} *with*

$$k = \frac{\log(1-p)}{\log(1-\mathrm{w}^n)} \tag{26}$$

where w is the ratio between the total number of "inlier" points. We choose the valuable as 0.5

8: *end if*
i=i+1;
9: end while

2.6 Proposed Method to Combine Images

After Homography matrices are calculated, the final step that is to create the Panorama image that is blended two images together. The basic idea of the step is to use a centered image, then to use the matrix to project the image Homography left to center the image plane as shown in Fig. 8

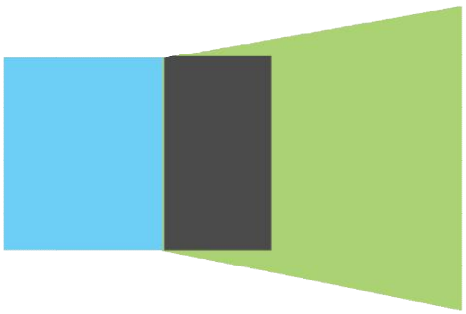

Fig. 8. Combining two images

In the Fig. 8, the image plane used as reference is on the left, and the right picture is projected onto the image plane of the first image using Homography matrix. The green part is the private part of each picture, and the black part is the common part of two images. After combining images, we find that the size of the

image is not rectangular, and they have distortion in the image on the right. We therefore propose two steps to solve the problem as follows:

Instep 1: When we rotate the camera capturing the pictures, we also translate camera horizontally.
Instep 5: After combining images, we conduct the cropping rectangular to have a more completed image.

3 Methodology 2: Detect the Position of the Object of the Picture

3.1 Method of Implementation

In the step, we perform the steps as follows:

- Creating Panorama image from two input images
- Detecting object in image
- Identifying focal point and grounding point of the object
- Calculating the position in real space.

3.1.1 Create Panorama image from two input images

In the step, we perform similar to part 2

3.1.2 Detect object in image

We use SIFT algorithm to compare the sample object with the object in panorama image, shown as Part 2.2 and 2.3.

Then we find a set of N points similarity (x_i , y_i)

3.1.3 Determine the focus and exposure earth grounding point of the object

Focal point is the point in center of the object calculated as follows:

$$l_{w1} = \frac{\sum_{i=1}^{N} x_i}{N}, \tag{27}$$

where l_{w1} is an averaged coordinates x of the set of similarity.

Exposure earth grounding point is the point exposure ground of the object calculated as follows:

$$l_{h1} = \max(y_i) \tag{28}$$

3.1.4 Calculate the position in real space

We assume that we have:

h_h is the size of the sensor vertically,

h_w is the size sensor horizontally

f is the lens focal,

h_0 is the height of the camera,

Elevation angle β_h of the camera at the time of shooting, and

Rotation angle β_w of the camera at the time of shooting as shown in Figs. 9 and 10.

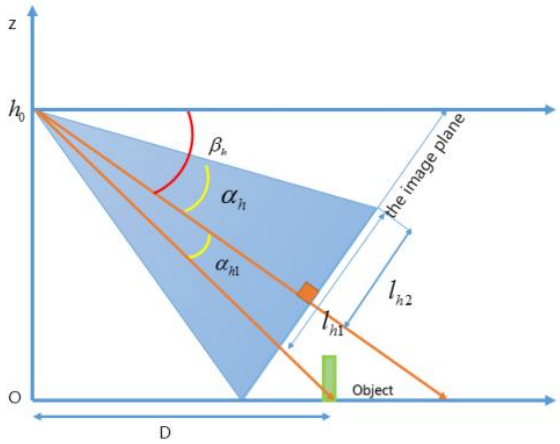

Fig. 9. Camera in vertical planes

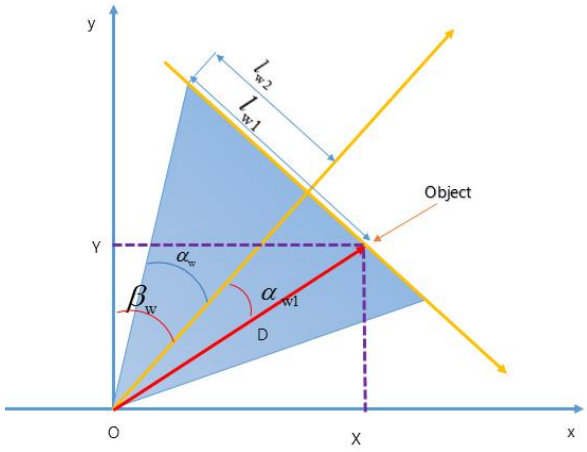

Fig. 10. Camera in the horizontal plane

Vertical angle of the camera is calculated as follows:

$$\alpha_h = \arctan(\frac{h_h}{2f}) \tag{29}$$

and horizontal angle is

$$\alpha_w = \arctan(\frac{h_w}{2f}) \tag{30}$$

$$(l_{h1} - l_{h2}) \times \cot(\alpha_{h1}) = l_{h2} \times \cot(\alpha_h) \tag{31}$$

$$\alpha_{h1} = \arctan\left(\tan(\alpha_h) \times \frac{(l_{h1} - l_{h2})}{l_{h2}} \right) \tag{32}$$

The distance from the camera to the object is calculated by the following formula:

$$D = h_0 \times \cot(\beta_h + \alpha_{h1}) \tag{33}$$

$$l_{w2} \times \cot(\alpha_w) = (l_{w1} - l_{w2}) \times \cot(\alpha_{w1}) \tag{34}$$

$$\alpha_{w1} = \arctan(\tan(\alpha_w) \times \frac{(l_{w1} - l_{w2})}{l_{w2}}). \tag{35}$$

The coordinates of the object coordinate system (O_{xy}) is calculated by the following formula:

$$X = D \times \sin(\beta_w + \alpha_{w1}) \tag{36}$$

$$Y = D \times c\cos(\beta_w + \alpha_{w1}) \tag{37}$$

4 Simulation Results

To design the algorithm, we use Matlab program. Input images have taken from real sensor with different sizes and angles. Initial energy of node is 12 joules. We divide the algorithm into five main steps: Extracting the features, matching key points, RANSAC, cutting parts of two identical photos, combining images and cropping.

To evaluate the energy consumption, we use the wireless communication energy model presented in [12-14]. The energy consumed when transmitting and receiving one bit are:

$$e_{tx} = \begin{cases} \varepsilon_{elec} + \varepsilon_{fs} d^2, d < d_0 \\ \varepsilon_{elec} + \varepsilon_{mp} d^4, d \geq d_0 \end{cases}, \tag{38}$$

And energy consumed when receiving one bit are

$$e_{rx} = \varepsilon_{elec}, \tag{39}$$

where d_0 is the close-in referential distance or threshold distance, which is determined based on measurements close to the transmitter, and d is the distance between two nodes (transmitter and receiver nodes) in WSNs. ε_{elec} is the energy consumption of the transceiver electron, and $\varepsilon_{fs} d^2$ and $\varepsilon_{mp} d^4$ are the energy consumptions for transmitting in free and multi-path space, respectively.

To estimate energy consumption, we use Panorama combining model. The energy consumptions for creating panorama image (E_{Pn}) is

$$E_{Pn} = E_{Ex} + E_{Fs} + E_C + E_H + E_{Oh} + E_{Cb} \tag{40}$$

where E_{Ex}, E_{Fs}, E_c, E_H, E_{Oh}, vàE_{Cb} are the energy consumptions for extracting the features, matching key points, cutting parts of two identical photos, calculating the matrix of Homography, rousting Homography Estimation using RANSAC, combining images.

Software used to measure the energy consumption is Joule meter [15].

We implemented the algorithm in two cases: multipath space distributed processing on nodes and concentrating processing on source node to consider the energy balance of the network. We obtained some results follows.

4.1 Results 1

4.1.1 Extracting the features

Fig. 11 shows the result of step 1. The features are shown by the blue vector, expressing the direction and magnitude.

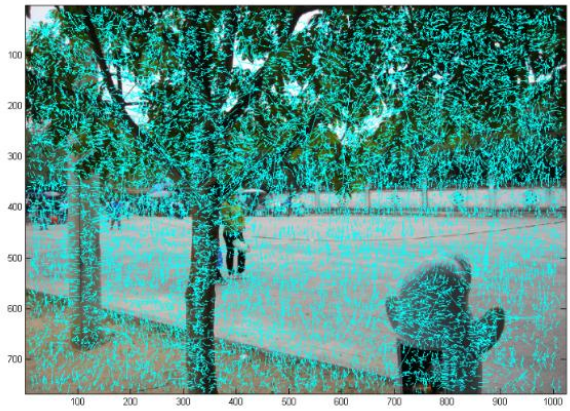

Fig. 11. Expressing the features by the vector

4.1.2 Matching key points

Fig. 12 shows the results of step 2, that looksfor the similarities between the two pictures. The similarities are connected by blue lines. We recognize that there are misidentified similarities, and henceusing RANSAC is the best way to calculate the matrix *H*, and it is not affected by the misidentified similarities.

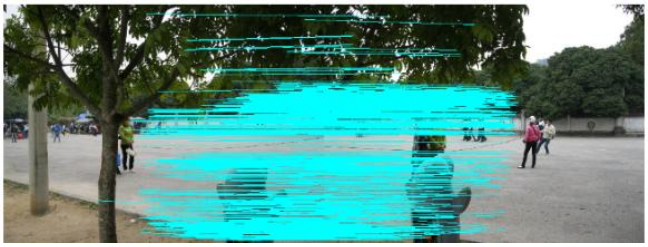

Fig. 12. Connecting the similarities by straight lines

4.2 Results *2*

We identify overlapping area (red area), and then we cut it from photo 1. We recognize that size of photo after cut overlaps is smaller than size of first photo before cut overlaps as shown in Figs. 13 and 14.

Fig. 13. Photo 1 before cutting overlaps

Fig. 14. Photo 1 after cutting overlaps

4.3 Result 3

4.3.1 Robust homography estimation using RANSAC

Fig. 15 shows results obtained in step 3 of our algorithm. We need 4 pairs of points to determine the matrix ***H.*** After using RANSAC, we have the best matrix ***H*** with four pairs of the best keypoint.

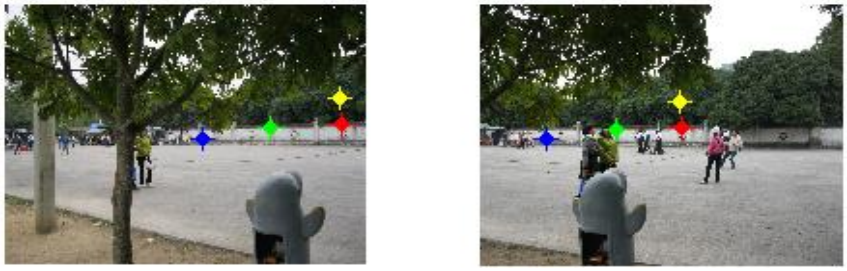

Fig. 15. Four pairs of best keypoint green, blue, yellow, red

4.3.2 Combining the two images together

Fig. 16 shows the results achieved were no ghosting.

In the last step, we perform to combine in two cases. In first case, we conduct the camera that has an angle as 30º, and in other cases we use translational camera. The results are shown in Figs. 17 and 18. We found that the output image is distortion because the second image is multiplied by the matrix ***H***.

Fig. 16. No ghosting

Fig. 17. The camera that has an angle as 30º

Fig. 18. Translational camera

4.3.3 Cropping

To overcome this drawback as shown in Fig. 17, we propose the solution that is to crop image as shown in Fig. 19.

4.4 Results 4: Detecting the Position in Real Space

4.4.1 Set up

We use model as follows:

Sensor size: 1 / 2.3 inch

h_h=4.55 mm is the size of the sensor vertically. h_w = 6.17 mm is the size of the sensor horizontally.

f =5 mm the lens focal. h_0 = the height of the camera.

Elevation angle β_h of the camera at the time of shooting is 0^0 and rotation angle β_w of the camera at the time of shooting is 0º.

4.4.2 Results

In the scene, we perform to detect object in real environment and obtain several.

The object in each image is blurred as shown in Fig. 20. The object in panorama image as shown in Fig. 21 is clearer than object in each image so we can detect object's position more exactly. Object is detected by central point and ground point as shown in Fig. 22 results locate objects shown in Table. 1.

Fig. 19. Image cropped panorama

Fig. 20. Object in each image

Fig. 21. Object in the panorama image

Table 1. Results locate objects and axes located at the camera

Coordinates	Using image 1	Using image 2 rotate 30^0	Using panorama	Location
X	1.152	0.956m	1.026	1m
Y	3.120m	2.914m	3.065	3m

Fig. 22. The central (red) and ground point (yellow) of the object

4.5 Energy Consumption

4.5.1 Energy consumption changes with different angles

In this section we use the input image with three different sizes, each size taking the case: translational camera, camera angles between two shots are 15º, 30º, 45º as shown in Fig. 23.

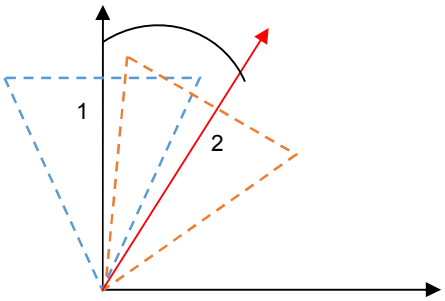

Fig. 23. Camera angles between two shots

The results of energy processing are shown in Table 2.

4.5.2 Energy consumption in wireless sensor networks

In this section, we simulate wireless sensor network with 100 nodes, random positions on the 300x500m as shown in Fig. 24. Initial energy of node is 12J. Number of source node is 10. We use two input images with 320x240pixels. We stop simulation when 10 nodes are dead.

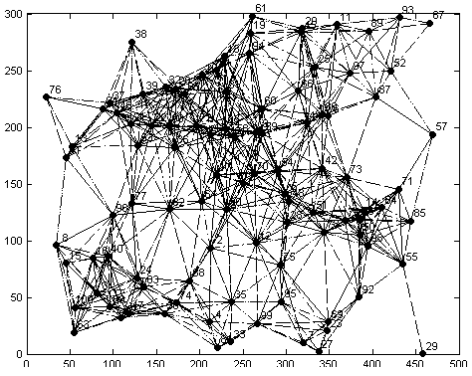

Fig. 24. Scene of Wireless sensor network

Table 2. The energy consumption of each node on the network size and different angles

	Step 1: SIFT			Step 2: Calculate H, RANSAC			Step 3: Combining the images			Step 4: Crop		
Angles	**Size**	**Process**	**Transfer**	**Size**	**Process**	**Transfer**	**Size**	**Process**	**Transfer**	**Size**	**Process**	**Transfer**
	pixel	**mWh**	**J**	**pixel**	**mWh**	**J**	**pixel**	**mWh**	**J**	**pixel**	**mWh**	**J**
Translational	160x120	1.9785	0.0829	177x124	0.0063	0.1778	177x124	0.3074	0.1422	163x109	0.3016	0.0384
15	160x120	2.2185	0.0829	224x136	0.0834	0.2145	224x136	0.4434	0.1974	159x110	0.3116	0.0378
30	160x120	1.8862	0.0829	251x141	0.1274	0.2358	251x141	0.3074	0.2293	226x110	0.0802	0.0537
45	160x120	1.9985	0.0829	476x241	0.3514	0.5785	479x241	0.4822	0.7449	235x110	0.2477	0.0558
Translational	320x240	6.1017	0.3318	367x173	0.0394	0.6061	367x273	0.5222	0.4907	340x230	0.3165	0.1689
15	320x240	6.4494	0.3318	448x274	0.7957	0.8621	488x274	0.6139	0.8191	310x230	0.3265	0.1540
30	320x240	6.1940	0.3318	637x280	6.7425	1.1023	637x280	0.6724	1.1558	573x240	0.3577	0.2970
45	320x240	6.8894	0.3318			0.3318						
Translational	640x480	132.732	1.3271	764x543	0.0434	3.1193	764x543	2.1026	2.6882	638x532	0.5493	0.7331
15	640x480	112.312	1.3271	856x504	0.0874	3.1909	856x505	0.5946	2.7975	630x470	0.3435	0.6396
30	640x480	129.632	1.3271	950x527	0.1513	3.4899	950x527	0.5946	3.2442	630x470	0.3016	0.6396
45	640x480	132.272	1.3271	1151x544	0.1751	4.0320	1151x544	3.0737	4.0574	630x470	0.4684	0.6396

We simulate multipath space distributed processing on nodes and concentrated processing on source node in two cases, cutting overlaps and uncutting overlaps.

Results of energy calculations are shown in Table 3, Table 4 and Figs. 25, 26, and 27

Table 3. The energy consumption of each node on the network

Step		Image 1 pixel	Image 2 pixel	Energy J	Time s	Times	Energy/times J	Energy/bit J
1	Uncut	320x240	320x240	1080.7	239.711	100	2.39711	1.301E-06
	Cut	320x240	320x240	1080.7	239.711	100	2.39711	1.301E-06
2	Uncut	320x240	320x240	207.2	21.782	100	2.072	5.621E-07
	Cut	320x240	320x240	207.2	21.782	100	2.072	5.621E-07
3	Uncut	468x240		1122.5	180.901	200	5.6125	2.082E-06
	Cut	468x240		994.7	133.688	200	4.9735	1.845E-06

Table 4. Results of the energy consumption of sensor network

Mode	Round	Time	Energy	Energy/round
Distributed, uncut	45	124.565	350.8683	7.797074
Distributed, cut	47	130.072	329.6537	7.013908
Concentrate, uncut	10	32.717	107.9423	10.79423
Concentrate, cut	10	35.851	101.5523	10.15523

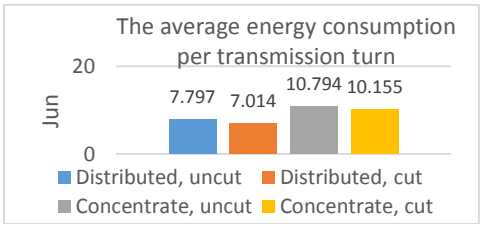

Fig. 25. The average energy consumption per transmission round

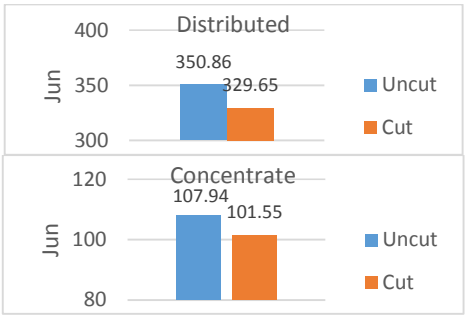

Fig. 26. Energy consumption of network

Through Figs. 25, 26 and 27, we found that since multipath space distributes processing on nodes, wireless sensor networks exist longer; consume less energy than concentrate processing on source node. Also by the cutting overlaps, energy consumption is less than uncut overlaps.

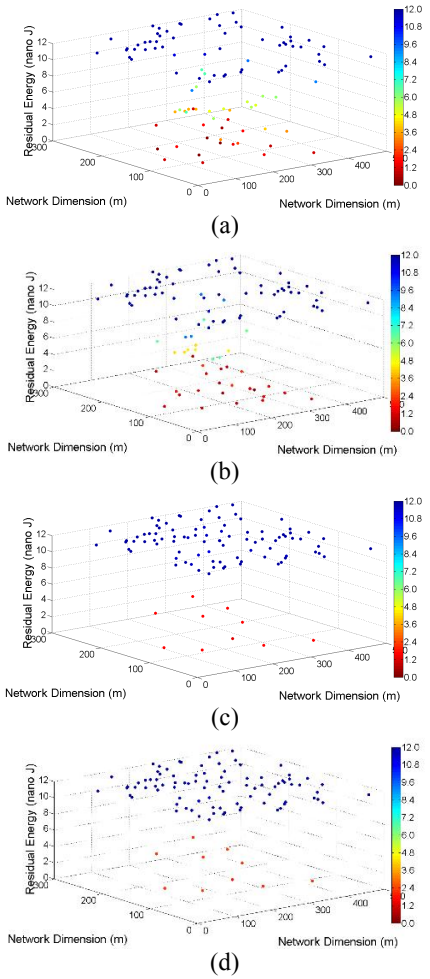

Fig. 27. Distribution of energy in wireless sensor networks

a. Distributed, uncut; b. Distributed, cut; c. Concentrated, uncut; d. Concentrated, cut

4.6 Discussion

4.6.1 Extracting the features

SIFT algorithm is the superior in accuracy due to less affected by the direction, intensity of light, noise, etc. of the input image. Therefore, it helps to calculate the Homography matrix more accurate, and it limits errors when performing image combining. However, the disadvantage of this algorithm is still slow speed.

However, the speed is slow because of the volume of processing large, multi-step.

4.6.2 Cutting overlaps

In the step, we recognize that “ghosting” problem has been removed.

4.6.3 Combining two images

In the step, we show two solutions for the translational camera cropping and we find that solutions translational Panorama camera is less distorted image, almost no need to crop. However, a small observation angle depends on the distance by translating camera for the amount of image information in Panorama. To solve the problems, we perform algorithm with the camera angle and cropping photos Panorama observations of much larger.

4.6.4 Detecting the position in real space

Objects can be put on flat surface. If surface is tilted, you have to know before tilt angle.

4.6.5 Energy Consumption

Through the implementation results in Figs. 4 and 5, we can see the energy consumption of the system will be reduced when implementing distributed processing on the network and cutting overlaps.

Although the input images are the same size but energy consumption is different since each pair of images have different overlap, and hence the number of similarities between the two images is different volume to be processed. The lager the overlapping is, the greater energy consumption is.

When comparing the energy consumption between multipath spaces distributed processing on nodes and concentrated processing on source node, we found distributed processing nodes is less, and therefore lifetime is longer.

When comparing the energy consumption between two cases: Cutting overlaps and uncut overlaps, we found consumption energy is less when cutting overlapped regions. However, energy consumption of cutting overlaps and transfer are much less than energy consumption to transfer original image. Therefore, volume calculated for the following steps will reduce.

The reduction of energy consumption when transmitting multi-path, distributed processing and cutting overlapped regions will help to balance the energy consumption for WSNs.

5 Conclusion and Future Work

This paper focuses on technical research combining Panorama technology from two input images. In each step, we present fundamental theoretical basis, then offer in depth assessment and implement of techniques to apply in Panorama image combining. However, the performance of algorithm is not yet optimized. Therefore, we will optimize techniques to improve time and accurate while rebuilding image, perform to combine the Panorama technology with more than two input images, search algorithm to solve the problem such as image unmatched pair, image edge blurring. In the future, we will use Panorama technology to improve the quality of image for searching, tracking and detecting objects applications.

Acknowledgments

This research is funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 102.04-2012.06.

Competing Interests

Authors have declared that no competing interests exist.

References

[1] Alexander T, Marieke H, Mirela K, Nico D. Effects of field of view on human locomotion. Proc. SPIE 6955, Head- and Helmet-Mounted Displays XIII: Design and Applications, 69550H; 2008.

[2] Oleksijczuk D. The First Panoramas: Visions of British Imperialism; 2011.

[3] David L. Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision. 2004;60(2):91-110.

[4] Michael D, Mortimer A. Molecular expressions microscopy primer: Digital image processing - difference of gaussians edge enhancement algorithm. Olympus America Inc. and Florida State University.

[5] Jgou H, Douze M, Schmid C. Searching with quantization: Approximate nearest neighbor search using short codes and distance estimators. Technical Report RR-7020, INRIA; 2009.

[6] Michel D, Elena D. Encyclopedia of Distances; 2009.

[7] Matthew B, David L. Automatic Panoramic image stitching using invariant features. International Journal of Computer Vision. 2007;74(1):59-73.

[8] Elan D. Homography Estimation. Carleton University; 2007.

[9] Richard H, Andrew Z. Multiple view geometry in computer vision. 2th ed. Cambridge University Press; 2003.

[10] Melzer T. SVD and its Application to Generalized Egienvalue problem. 2004;1-15.

[11] Martin F, Robert B. Random sample consensus: A paradigm for model fitting with application to image analysis and automated cartography. Communications of the ACM. 1981;24:381–395.

[12] Heinzelman WB, Chandrakasan AP, Balakrishnan H. An application-specific protocol architecture for wireless microsensor networks. IEEE Transactions on Wireless Communications. 2002;1(4):660–670.

[13] Lu Q, Luo W, Ye X. Collaborative in-network processing of LT based image compression algorithm in WMSNs. In Proceedings of the 1st International Workshop on Education Technology and Computer Science (ETCS '09). 2009;839–843.

[14] Lu Q, Luo W, Wang J, Chen B. Low-complexity and energy effcient image compression scheme for wireless sensor networks. Computer Networks. 2008;52(13):2594–2603.

[15] Joulemeter. Available:http://research.microsoft.com/en-us/projects/joulemeter (Accessed 8 April 2015).

Peer-review history:
The peer review history for this paper can be accessed here (Please copy paste the total link in your browser address bar)
www.sciencedomain.org/review-history.php?iid=1142&id=6&aid=9236