A simplified approach to merging partial plane images

Mária Kruláková [1]

This paper introduces a method of image recognition based on the gradual generating and analysis of data structure consisting of the 2D space using the difference from the row and column vectors' sums as an application of the linear algebra. The result of the data processing is a graphical interpretation of the measure of the similarity of the generated results of overlapping of 2 images. Maximal measure of the similarity is a measure for image registration. The study result is to create a list of the images order, in which one follows the other, included in the non-registered set of images that can be used for the final image-stitching

Keywords: *image-stitchining, image recognition, measure of the similarity*

Introduction

The mathematical model used for the study describes the 2D space and the model application can be classified as image recognition method with consequent registration Oldridge et al. (2009), Szeliski and Coughlan (1997). In order to accomplish the final image-stitching, we need to create a mathematical model first Ch-Y.Chen and Klette (1999). The mathematical expressions consist of the pixel values of the image in the coordination system. Then the expressions of two images are processed for overlapping. Mathematical models applied for the data frame were classified into patch-based and feature-based algorithms (Lowe (1999), Pazzi et al. (2010), Chen and Williams (1993)). Applying them we can define the image as graphical interpretation using similarity of the patches and features.

In the linear algebra there is a part devoted to the matrix theory including the algorithms developed in order to find out the rate of similarity based on the linear equations. The process of the algorithms development is based on the assumption that the values generated from the image data frames into the row and column vectors of sums can be applied using them in the formation of the square matrix. The main aim is to determine the similarities based on the estimation out of the differences of the matrixes. In this study the algorithms based on the logical values coming from the data frame of red component applied in the square binary matrices of the image were used for the first stage of the pre-processing of the images.

The RGB image is stored as mxnx3 data array that determines red, green and blue component layer for each pixel separately. The image preprocessing is based on the transformation of the colour structure of the image into the binary data matrix that consists of the red component layer only. Binary images, commonly called bi-level, have each pixel stored as a single bit (0 or 1). Pixel with 0 value is displayed in black that described the occurrence of the red component layer of the image.

These values of the data frame of the red component layer of the image are gradually summed up into the row and column vectors of sums based on the rows and column structures from each data array of image i. Then the components of the vectors are generated and organized into the data matrix Red_{mxi} of the row vector of sums and $Cred_{ixn}$ of the column vector of the sums, where i is defined as number of processed of the data arrays of the images.

The first absolute difference V_{row} is a result of the 2 data arrays with arranged row vectors of sums into the 2 square matrixes V_iRred and $V_{i+1}Rred$. The second absolute difference R_v is a diffrence of the first absolute difference of the data matrix V_{row} and the transponned matrix made of the V_{row} called V^T_{row}. This will define the level of the similarity of the components arranged in rows of the vector after the development of the row vector of the sums $Rrow_j$ from the second absolute difference of the square matrix R_v. The same process is executed for generating of the column vectors of the sums. They will determine the level of similarity of the components arranged in the vector but the column arrangement of the vector's components after the row vector of sums $Crow_j$ from the second absolute difference of the square matrix C_u is created. The arrangement of the components of the row vectors of sums $Rrow_j$ a $Crow_j$ into square matrices and consequent calculation of the differences of the data arrays of the vectors lead to creating data array of the matrix F_{diff} from which we can create definitive row vector of sums $match_j$. This row vector of sums matchj consists of zero values in its components. Based on these values the similarity in two compared data arrays of the images was determined.

[1] *Ing. Mária Kruláková.*, Technical Univerzity of Košice, FBERG, Institute of Control and Informatization of Production Processes, B. Němcovej 3, 040 01 Košice, maria.krulakova@tuke.sk

Definitions and notation

Red_I is a finite set of the Z^2. It is a matrix with m rows and n columns with the pixel values of 0 a 1 of binary image as its elements. For $m \in Z$ there is row m determined as $\{Red\_I(x,y)^2 : y = m\}$, where m is an index of particular row. For $n \in Z$ there is a column n determined as $\{Red\_I(x,y)^2 : x = n\}$, where n is an index of particular column. Sum of the row,$Rred_m$ is a number of the elements of the matrix array *Red_I* generated from the number of the elements of m rows that sums the values of the matrix columns together into the row vector of sums $\mathbf{Rred_m} = \sum_{\mathbf{n} \in \mathbf{Z}} \mathbf{Red\_I_{(mxm)}}$. Sum of the column, $Cred_n$ is a number of the elements of the matrix array *Red_I* generated from the number of the elements of n columns that sums the values of the matrix rows together into the column vector of sums $\mathbf{Cred_n} = \sum_{\mathbf{m} \in \mathbf{Z}} \mathbf{Red\_I_{(nxn)}}$.

Then the $Rred_m$ is the row vector of sums (1) and $Cred_n$ is the column vector of the sums (2) generated from the matrix *Rred_I*.

$$Rred_m = (rred_1, rred_2, rred_3, rred_4, ..., rred_m) \qquad (1)$$

$$Cred_n = (cred_1, cred_2, cred_3, cred_4, ..., cred_n) \qquad (2)$$

The result of the scipt *createComponentR.m* is a matrix $Rred_{mxi}$. The size of this matrix is $[mxi]$. The script *createComponentR.m* has created the row vector of the sums $Rred_m$ (1) as a sum of the binary values of all 4 images processed from the data arrays of the matrices of each image. For example first element of resulting $Rred_{mx}$i is a sum of elements in the first column of the *Red_I*.

$$\mathbf{Rred_{mxi}} = \begin{pmatrix} rred_{1,i-i+1} & rred_{2,i-i+1} & rred_{3,i-i+1} & \dots & rred_{m,i-i+1} \\ rred_{1,i-i+2} & rred_{2,i-i+2} & rred_{3,i-i+2} & \dots & rred_{m,i-i+2} \\ rred_{1,i-i+3} & rred_{2,i-i+3} & rred_{3,i-i+3} & \dots & rred_{m,i-i+3} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ rred_{1,i-i+i} & rred_{2,i-i+i} & rred_{3,i-i+i} & \dots & rred_{m,i-i+i} \end{pmatrix}$$

The result of the scipt *createComponentC.m* is a matrix $Cred_{ixn}$. The size of this matrix is $[ixn]$.
The script *createComponentC.m* has created the column vector of the sums $Cred_n$ (2) as a sum of the binary values of all 4 images processed from the data arrays of the matrices of each image.

$$\mathbf{Cred_{ixn}} = \begin{pmatrix} cred_{i-i+1,1} & cred_{i-i+2,1} & cred_{i-i+3,1} & \dots & cred_{i-i+i,1} \\ cred_{i-i+1,2} & cred_{i-i+2,2} & cred_{i-i+3,2} & \dots & cred_{i-i+i,2} \\ cred_{i-i+1,3} & cred_{i-i+2,3} & cred_{i-i+3,3} & \dots & cred_{i-i+i,3} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ cred_{i-i+1,n} & cred_{i-i+2,n} & cred_{i-i+3,n} & \dots & cred_{i-i+i,n} \end{pmatrix}$$

In case the input image *Red_I* has the size $[mxn]$, and $n > m$, then $k = |m-n|$ and notation for for the same output size of the $Rred_m$ and $Cred_{n-k}$ matrixes, if $[mxn-k]$.

Processing of the row vector of sums is created by script named as *organizeRow.m*. This script processes the row vector of sum in order to create a new square matrix as data array diplayed on (3) shows, where $i = 1$ is the first image of out 4. Square matrix as data array diplayed on (4) represents the second image of out 4, where $i+1 = 2$.

$$\mathbf{V_iRred} = \begin{pmatrix} rred_{1,i} & rred_{1,i} & rred_{1,i} & \dots & rred_{1,i} \\ rred_{2,i} & rred_{2,i} & rred_{2,i} & \dots & rred_{2,i} \\ rred_{3,i} & rred_{3,i} & rred_{3,i} & \dots & rred_{3,i} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ rred_{m,i} & rred_{m,i} & rred_{m,i} & \dots & rred_{m,i} \end{pmatrix} \qquad (3)$$

$$\mathbf{V_{i+1}Rred} = \begin{pmatrix} rred_{1,i+1} & rred_{1,i+1} & rred_{1,i+1} & \dots & rred_{1,i+1} \\ rred_{2,i+1} & rred_{2,i+1} & rred_{2,i+1} & \dots & rred_{2,i+1} \\ rred_{3,i+1} & rred_{3,i+1} & rred_{3,i+1} & \dots & rred_{3,i+1} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ rred_{m,i+1} & rred_{m,i+1} & rred_{m,i+1} & \dots & rred_{m,i+1} \end{pmatrix} \quad (4)$$

Fig. 1 represents an algorithm that describes the process of generating row and columns vector sums into the matrixes $Rred_{mxi}$ a $Cred_{ixn}$. Parameter i determines the number of the preprocessed images.

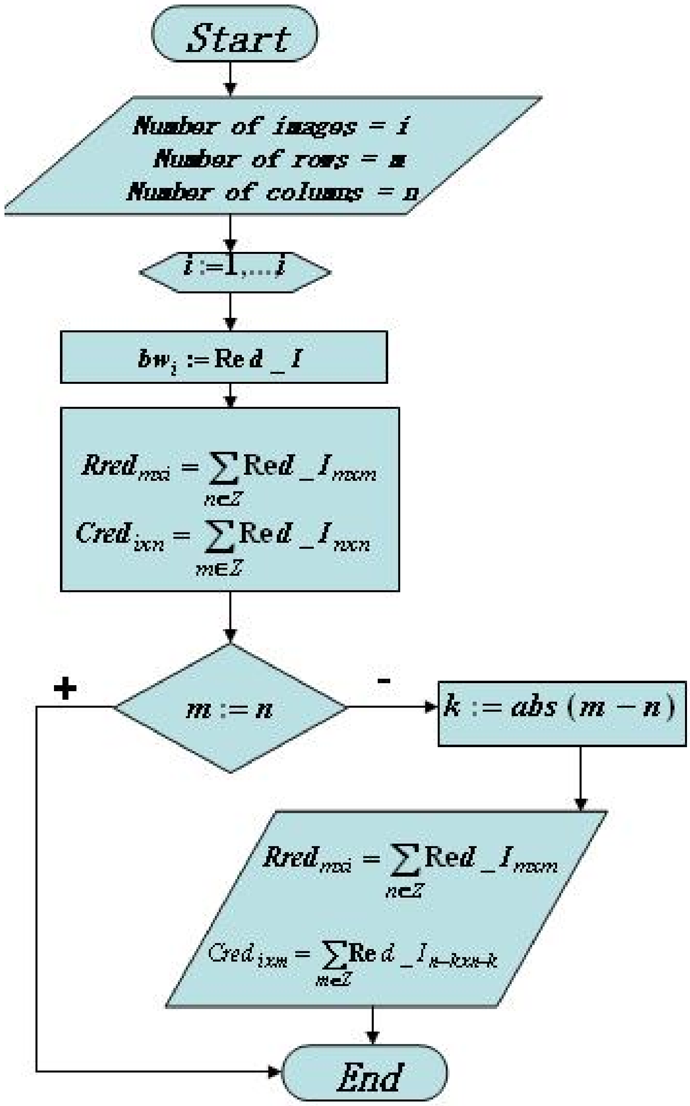

Fig. 1. Generation of the output matrixes $Cred_{ixm}$ and $Rred_{mxi}$.

As the square matrixes V_iRred and $V_{i+1}Rred$ are created out of the row vector of sums of the matrix $Rred_{mxi}$ using (3), in the same way the square matrixes V_iCred and $V_{i+1}Cred$ are created out of the column vector of sums of the $Cred_{ixn}$ matrix using (4). The arrangement of the column vector of sums is generated by script *organizeCol.m*.

The process of how to find the differences between the matrixes includes following steps:

- The first absolute difference V_{row} is made from the 2 data arrays that are generated by the row vectors of the sums with result of the 2 square matrixes V_iRred a $V_{i+1}Rred$ is (5):

$$V_{row} = |V_iRred - V_{i+1}Rred| \tag{5}$$

- The second absolute difference R_v between the matrix V_{row} and V_{row}^T is (6):

$$R_v = |V_{row} - V_{row}^T| \tag{6}$$

- The row vector of sums $Rrow_j$ is generated from the matrix R_v (7):

$$Rrow_j = \sum_{j=1}^{m} R_{v(mxm)} \tag{7}$$

- The column vector of sums $Rcol_j$ is generated from the matrix R_v (8):

$$Rcol_j = \sum_{j=1}^{m} R_{v(mxm)} \tag{8}$$

- Comparing the values of teh row vector of sums (7) and transposed column vector of sums $Rcol_j^T$ (8) it was found out that the vectors consists of the same values of the compared elements of both vectors (9):

$$Rrow_j = Rcol_j^T \tag{9}$$

- The first absolute difference V_{col} is made from the 2 data arrays that are generated by the row vectors of the sums with result of the 2 square matrixes V_iCred a $V_{i+1}Cred$ is (10):

$$V_{col} = |V_iCred - V_{i+1}Cred| \tag{10}$$

- The second absolute difference C_u between the matrix V_{col} and V_{col}^T is (11):

$$C_u = |V_{col} - V_{col}^T| \tag{11}$$

- The row vector of sums $Crow_j$ is generated from the matrix C_u (12):

$$Crow_j = \sum_{j=1}^{m} C_{u(mxm)} \tag{12}$$

- The column vector of sums $Ccol_j$ is generated from the matrix C_u (13):

$$Ccol_j = \sum_{j=1}^{m} C_{u(mxm)} \tag{13}$$

- Comparing the values of teh row vector of sums (12) and transposed column vector of sums $Ccol_j^T$ (13) it was found out that the vectors consists of the same values of the compared elements of both vectors (14):

$$Crow_j = Ccol_j^T \tag{14}$$

- The final absolute difference F_{diff} (15) is a result of the rearrangement of the values from the row vector of sums Row_j (7) generated from the matrix R_v into matrix Row and row vector of sums $Crow_j$ (12), that is generated from matrix C_u into matrix $Column$ (15):

$$F_{diff} = |(Row - Column| \tag{15}$$

- The row vector of the sums $match_j$ was generated from the matrix F_{diff} (16):

$$match_j = \sum_{j=1}^{m} (F_{diff})_{mxm} \tag{16}$$

The measure of similarity was determined according to the most frequent occurrence of the zero values within the vector elements (see Fig.6). All used scripts that are used in the study (*createComponentR.m*, *createComponentC.m*, *organizeRow.m*, *organizeCol.m*, *differenceSum.m*) can be found in the apendix at the end of the paper.

Graphical comparison of the vector values

Fig. 2 shows the list of images that was preprocessed and analysed. The content of the list includes 4 preprocessed binary matrixes corresponding with the 4 bianry images (A, B, C, D) compared in pairs. Fig. 3 and Fig. 4 show the mutual comparison of the vector values between overlapped images *A&B* and *A&C*, Fig. 3 and *B&C* and *C&D* Fig. 4.

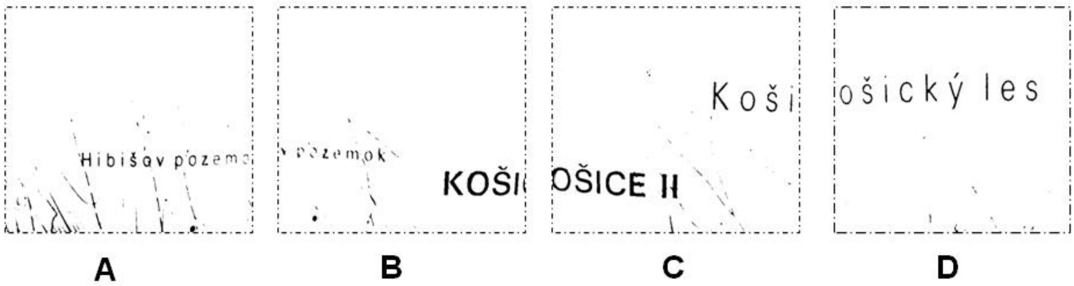

Fig. 2. Generated binary images.

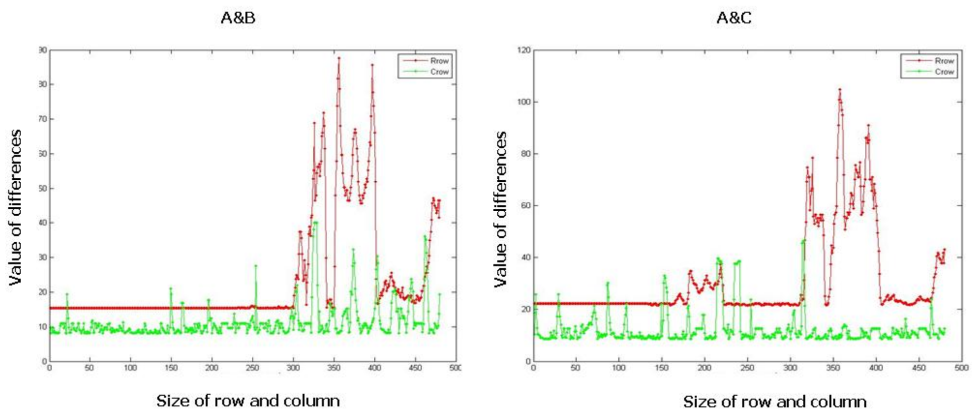

Fig. 3. Comparison of the images A&B and A&C.

The calculation of F_{diff} (15) results in the row vector of sum $match_j$ (16) . This vector includes also zero values that occur when there is the same value in the row and in the column. The frequency of zero values in the row vector of sum $match_j$ is final result of the overlapping of all compared images Fig. 5. It provides the information about the measure of similarity (higher the number of zero values out of pairwise comparison, greater the similarity).

Tab. 1 includes all possible combination of the mutual comparison of overlapping the images (based on pairwise comparison). For each comparison (A&B, A&C, A&D, B&C, B&D, C&D) there is number of zero values found to measure the similarity.

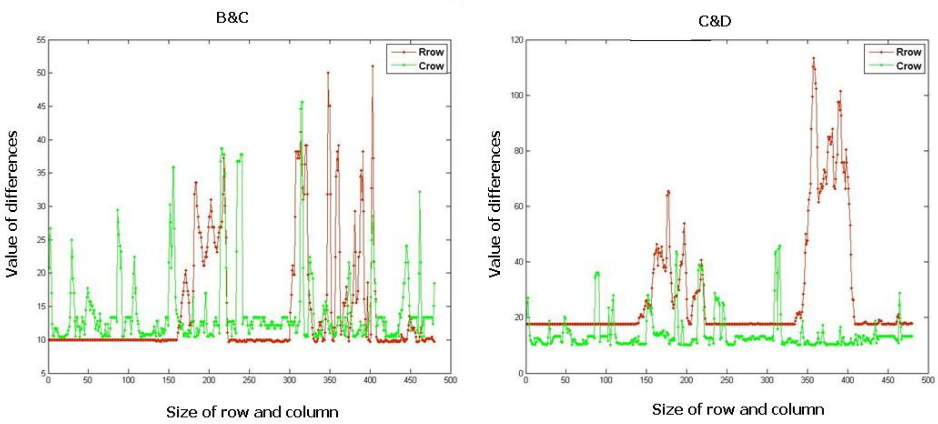

Fig. 4. Comparison of the images B&C and C&D.

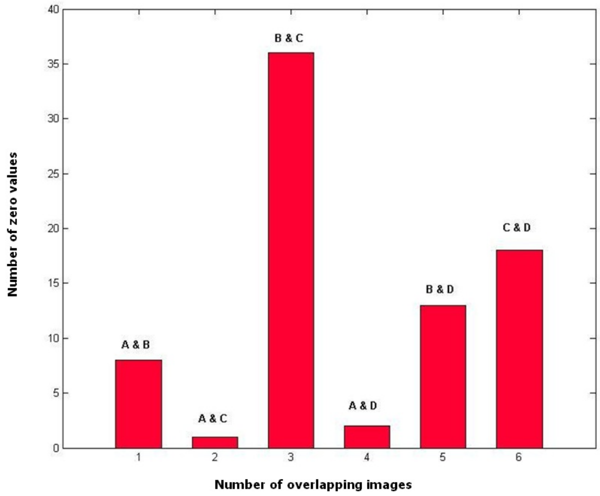

Fig. 5. Result of pairwise comparison (similarity of images)

Tab. 1. Image registration.

Combination of the pairwise comparison	Number of the zero values	The sequence of images
A&B	8	A→ *B*
A&C	1	
A&D	2	
B&C	36	B→ *C*
B&D	12	
C&D	18	C→ *D*

Conclusion

The main goal of the study was to use data processing in graphical interpretation of the measure of the similarity by comparing and assessing how 2 images overlap. In order to fulfill the target we have created algorithms of linear algebra. The result of the image data processing can be seen in Tab. 1. It informs about the measure of the similarity and a list of the images order. It presents image registration as an outcome of the pairwise comparison of 4 images. They were indexed as following: $A \to B \to C \to D$. This type of indexing that is important for image registration. It can be applied in case of more extended list of images with similar data frame for making the final image-stitching faster. The study help us to find algorithms for image processing and the way of creating the sequence of the images, in which one follows the other, that can be practically used for non-registered set of images and for making the final image-stitching faster.

Acknowledgement
This work was supported by the Slovak Research and Development Agency with contract No. APVV-0482-11.

References

Ch-Y.Chen, Klette, R., 1999. Image stitching-comparisons and new techniques, in: Computer Analysis of Images and Patterns. Springer Berlin / Heidelberg. volume 1689 of *Lecture Notes in Computer Science*, pp. 835–835.

Chen, S., Williams, L., 1993. View interpolation for image synthesis, in: Computer Graphics (SIGGRAPH'93), pp. 279–288.

Lowe, D.G., 1999. Object recognition from local scale-invariant features, in: International Conference on Computer Vision, Corfu, Greece. pp. 1150–1157.

Oldridge, S., Miller, G., Fels, S., 2009. Automatic classification of image registration problems, in: LNCS 5815. Springer-Verlag Berlin Heidelberg, pp. 215–224.

Pazzi, R.W., Broukerche, A., Feng, J., Huang, Y., 2010. A novel image mosaicking technique for enlarging the field of view of images transmitted over wireless image sensor networks. Mobile Networks and Applications 15, 589–606.

Szeliski, R., Coughlan, J., 1997. Spline-based image registration. International Journal of Computer Vision 22, 199–218.