International Journal of Computer and Information Technology (ISSN: 2279 – 0764)
Volume 05 – Issue 01, January 2016

Artificial Bee Colony for False Match Filtering in 3D Reconstruction

Milena Lazarova* and Zoran Kostov
Dept. Computer Systems
Technical University of Sofia
Sofia, Bulgaria
*Email: milaz [AT] tu-sofia.bg

Abstract— **The paper discusses the problem for 3D scene reconstruction and presents a general overview of the steps and approaches utilized for image based 3D model generation. A metaheuristics approach based on Artificial Bee Colony (ABC) is suggested to be applied for false match filtering instead of RANSAC or similar methods. The algorithm is based on ABC with Big Valley Landscape Exploration and is applied for finding false feature match by considering them as a possible local extrema in the problem search space. The suggested solution uses a population based heuristics that combines the global exploration of the search space with diversification and evaluation of the local optima. The approach allows fast and efficient feature correspondence refinement. The 3D reconstruction utilizing the suggested metaheuristics based false match filtering is experimentally evaluated using two test datasets with different number of images.**

Keywords-3D reconstruction; false match filtering; metaheuristics, artificial bee colony, big valley landscape exoloration

I. INTRODUCTION

The ability to recover three dimensional (3D) structure of the world around us is natural for the human vision system. The generation of a 3D model of existing objects based on data gathered by non-contact sensors has many applications including virtual reality, video games and movies, cultural heritage preservation, industrial and architecture object modelling, object recognition and analysis. Solving the problem of 2D to 3D projection requires additional data in order to recover the missing depth information [1, 5]. Active devices utilization such as either laser scanner or range camera can support the reconstruction by generation of a 3D point cloud that later to be used for 3D model generation [2, 3]. On the other side image based modelling does not require expensive and specialized devices still providing possibility to recover 3D models only using 2D images. Restoring a three dimensional scene structure by extracting information from one or several images is recently a widely studied research problem [4, 5].

The paper discusses the problem for 3D scene reconstruction and presents a general overview of the steps and approaches utilized for image based 3D model generation. A metaheuristics approach based on Artificial Bee Colony (ABC) is suggested to be applied for false match filtering instead of RANSAC or similar methods for feature pair refinement. Metaheuristics provide a fast and prominent solution for hard optimization problems giving a near optimal solution in reasonable time [19, 20, 21]. The algorithm is based on ABC with Big Valley Landscape Exploration [23] and is applied for finding false feature matches by considering them as a possible local extrema in the problem search space. The suggested solution uses a population based heuristics that combines the global exploration of the search space with a local diversification and evaluation of the local optima. The 3D reconstruction with the suggested false match filtering is experimentally evaluated using two test datasets with different number of images.

II. 3D RECONSTRUCTION

The world around us is a three dimensional. When pictured as a 2D image the one of the dimensions is omitted. Recovering a 3D volumetric model of an object that is captured by a camera requires the missing depth of the scene to be restored.

The image based visual 3D modelling framework can be generally regarded as comprising the following steps [5]:

- image acquisition: in general the higher quality images are taken the more successful and accurate is the 3d reconstruction;
- image calibration: intrinsic parameters such as focal-length and external parameters like relative rotation/translation among cameras capturing the separate images are estimated;
- 3D point cloud generation: estimate a multi-view correspondences between corresponding feature points in the images;

- mesh model generation: construct a 3D model based on the 3D point cloud using computer geometry techniques;
- texture mapping: builds the correspondence between facets of 3D geometry model and patches in the input images.

Depending on the number of images that are used there are different approaches suggested in the literature based on either single image [6], stereo pair of images [7] or a multi view set of images [8, 9, 10].

The multi view approach requires at least two images to be analysed for finding a 3D positions and the quality of reconstruction generally increases with the number of captured images. The 3D reconstruction by multi view images is based on matching the locations of features in at least two images and generally the larger number of matching features the better quality of the reconstruction. Several capturing parameters are required for precise 2D to 3D coordinates calculation: external camera parameters (camera position and rotation) and internal camera parameters (camera lens).

There are different algorithms for 3D reconstruction that uses various approaches but can be divided into two types: algorithms for sparse and algorithms for dense 3D reconstruction. Both types differ mainly in the number of the images that are used as input data and the number of recovered 3D points. Sparse 3D reconstruction requires fewer images to be able to calculate the relevant 3D points. However, with the increase in the number input images, the number of reconstructed points and thus the quality of the reconstruction proportionally increases. Sparse 3D reconstruction approach is based on calculation of less number of 3D points compared to dense 3D reconstruction. The algorithms for dense 3D reconstruction algorithms use multiple images to be able to calculate 3D points. The planes of the sensors of the two cameras that capture two adjacent images should lie almost in the same plane in order to obtain reliable results.

Three general subtasks can be outlined for solving the 3D point cloud generation no matter which particular reconstruction approach is applied:

1. Identify features in each of the input images.

2. Search for features that match in the input image pairs.

3. Find the corresponding 3D coordinates of the points based on the features matching (using rectification and triangulation).

The final results is a set of points with coordinates in world space which allows a depth map to be created and a 3D model of the input scene to be produces.

A 3D coordinate of a point in world space is calculated based on the 2D coordinates of the corresponding projected points in two or more of the input images. There are various algorithms for finding local image features that are reliable for solving the correspondence problem in a variety of conditions such as scale changes, camera rotation and perspective distortions [11]. The features detectors may rely upon corner, blob, region and edge detection. The scale invariant blob detector and the corresponding algorithm for descriptor (SIFT) is one of the most prominent based on finding local extrema of the Laplacian of Gaussians (LoG) of the image [12]. Another option is to utilize the Speeded Up Robust Features (SURF) that uses an integer approximation of the determinant of Hessian blob detector estimated by a precomputed integral image and a descriptor based on the sum of the Haar wavelet response around the point of interest [13].

After the feature points in each of the input images are detected a correspondence should be found between them based on feature descriptor similarity. As better are the feature matches that are calculated as more correct are the calculated 3D coordinates and more relevant 3D model will be reconstructed. One of the approaches to establish feature correspondence uses exhaustive search by brute force algorithm. For each two features the similarity is estimated based on a Euclidean distance:

$$d(p,q) = \sqrt{(p_1 - q_1)^2 + \ldots + (p_{128} - q_{128})^2} \qquad (1)$$

where p_i and q_i are the SIFT feature descriptors of feature p and feature q respectively (i = 1, 2, …, 128) and d is the feature similarity estimated using Euclidean distance.

Two more constraints are added as a requirement in order to increase the quality of matches found [12]:

- a correspondence is only established if both two features are found to be best match to each other, i.e. each of them is the nearest neighbour for the other;
- a correspondence is only established if the similarity to the nearest neighbour is significantly smaller than the dissimilarity to the second nearest neighbour.

The accuracy of the 3D reconstruction depends very much on the detection of reliable features and the establishment of correct feature correspondence. Before using the feature matches to calculate the 3D coordinates a match refinement stage is recommended. False matches might be marked due to differences in imaging conditions as illumination changes as well as deviation due to the viewpoint change in the multiple views of the input scene. The detection and filtering of false matches is aimed to separate “good matches" from “bad matches". There are several approaches for false match removal that are generally based on RANSAC [14] or variants that attempt to improve it [15, 16, 17, 18]. RANSAC is based on fitting a model to the best feature points from the data sets by iteratively random sampling minimal data subsets. The algorithm uses an iterative procedure to determine inliers (good matches) that fit the model and outliers which cannot be fitted to it.

The various RANSAC improvements and suggestions for noise match filtering are all not deterministic and iterative in their nature thus making them hard to run in parallel.

III. ARTIFICIAL BEE COLONY ALGORITHM FOR FALSE MATCH FILTERING

A metaheuristic is a general framework for heuristics in solving hard computational problems [19, 20]. Metaheuristics are high level strategies for exploring search spaces by using different methods providing dynamic balance between diversification and intensification of the searched space [21]. The metaheuristics approaches can be classified depending on several criteria as follows [22, 23]:

- trajectory based search vs. population based search;
- nature-inspired vs. non-nature inspired approaches;
- dynamic vs. static objective function;
- one vs. various neighbourhood structures;
- memory usage vs. memory less methods.

The successful implementation of a metaheuristic approach on a given optimization problem provides balance between exploitation of the accumulated search experience and the exploration of the search space to identify regions with high quality solutions in a near-optimal way. Therefore the metaheuristic approaches can be an effective alternative to RANSAC for finding the incorrect feature matches and thus reducing the noise in the final result of the 3D reconstruction.

A. Artificial Bee Colony

The Artificial Bee Colony (ABC) is one of the newest metaheuristics that is motivated by the intelligent behaviour of honey bees [23]. ABC is a population-based search procedure that can be used for solving optimization problems by combining both global exploration and local search of the solution space for finding near optimal solution to the problem in a reasonable time. The population comprises several individuals (artificial bees) that are aimed to discover foods positions with high nectar amount in a multidimensional search space. The goal is to evaluate the food sources and at the end to provide the optimal solution to the problem solved as being the food source of highest nectar. The individuals are divided in three groups: employed and onlooker bees are searching for a food source in the space based on a global and local experience while the scouts choose the food sources randomly without using experience. ABC system combines local search methods, carried out by employed and onlooker bees, with global search methods, managed by onlookers and scouts, attempting to balance exploration and exploitation process [24].

B. Artificial Bee Colony with Big Valley Landscape Exploration

As the name suggests the ABC with Big Valley Landscape Exploration is a modification of the original algorithm that uses a structure called "landscape" [24]. The landscape can be described as a structure of points in the search space that are generated by a heuristic operator for a neighbourhood of a search space in view of the given objective function. A landscape consists of many local optimum or false peaks that change by applying particular heuristic operators.

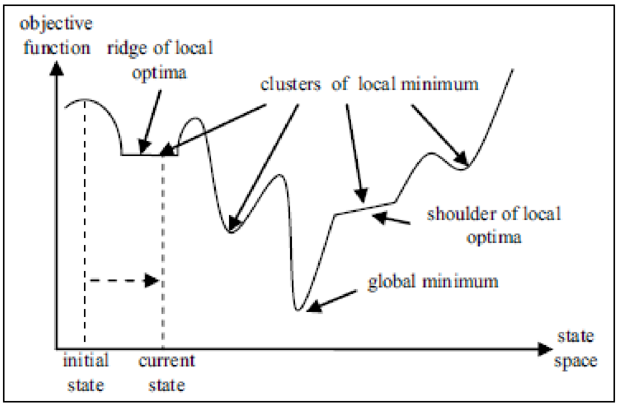

Figure 1. Big Valley Landscape Concept

The existence of such a local optimum often interferes with finding the global optimum. However the different landscape structures generated by several heuristic operators can support the search for the global optimum rather than prevent its detection. One such structure is the landscape "Big Valley" in which the local optima occur close together in clusters (fig. 1).

The Big Valley Landscape structure advises for the new food point selection by the scout bees based on previous local optima instead of random selection in the search space.

C. Feature match refinement using ABC with Big Valley Landscape Exploration

The ABC with Big Valley Landscape Exploration algorithm is aimed at solving optimization problems and obtaining a near optimal solution. In order to apply it for the feature match filtering and refinement stage as part of the multi view 3D reconstruction problem the following assumption are suggested:

- the optimization goal is to find the best false match that will be removed from the set of the calculated feature correspondences;
- each feature match is considered as a local optimum and thus a possible false match;
- the global optimum represents the solution of the optimization problem that is the highest dissimilarity of the set of all feature correspondences;
- the big valley landscape modification of the ABC algorithm applied for the feature match filtering has a goal of to find as many local optimum while looking for the biggest one;

The local optima are divided into two types. The first one are local optima which have been a good food source and bees have left after exploring them that is they represent false feature correspondence that is confirmed and will be removed from the set. The second type of local optima are the one that were visited by the bees but are evaluated as not being a good food source that is they are confirmed as a good match.

ABC with Big Valley Landscape Exploration algorithm uses a similar procedure as the one for the ABC (fig. 2). The algorithm starts by finding the initial sources of food for the bee colony (GenerateInitialSolutions). At this step the scouts randomly choose possible solutions of the optimization problem that is our case false feature match pairs.

The set of all feature correspondences to be filtered is divided equally between all scouts for exploring them as local optima that in our case is based on comparison of the given match with several randomly selected matches. The feature pairs that are selected as potential errors are further explored and optimized.

One iteration of the ABC with Big Valley Landscape Exploration algorithm comprises a search for food sources (Forage) and notification for the source found (PerformWaggleDance).

Forage is the procedure in search of food where a bee tries to further optimize a solution from the previous iteration that is a possible false match is verified whether to be accepted as a food source. The Forage step of the algorithm represents the exploration of the local space around a possible solution in order for it to be improved. The local search of the food sources in ABC with Big Valley Landscape Exploration algorithm requires adding a short memory for the bees that provides possibility to take into account the last few steps of solution optimization based on the feature match comparisons. The memory is implemented through a list that prevents repeating of comparisons between feature pairs explored at the last several iterations. Thus in Forage the comparison between matches is uniquely applied by the bees.

The local optima are stored in a list WL. At each iteration, employed bees chose one local optimum solution and explores the local space around it evaluated it by comparison with the current best solution based on the feature dissimilarity value (ObserveAndSelectDance) (fig. 3).

procedure BCBV
  $C_{max\_best} \leftarrow \infty$
  $N_{iter} \leftarrow 0$
  GenerateInitialSolution( )
  while N_{iter} <> N_{max} ***do***
    for each forager bee f_i ***do***
      if *WL*<>{ }
        f_i.ObserveNSelectDance( )
      end if
      $f_i.C_{max} \leftarrow f_i$.Forage( )
      if $f_i.C_{max} < C_{max\_best}$
        $C_{max\_best} \leftarrow f_i.C_{max}$
        f_i.PerformWaggleDance( )
      end if
    end for
    $N_{iter} \leftarrow N_{iter}+1$
  end while
end procedure BCBV

Figure 2. The ABC with Big Valley Landscape Exploration algorithm

The probability that a bee will follow a food source of another bee is given as P_{follow} and is estimated depending on the values of the fitness of the solutions explored by the given bee at the previous iteration Pf_i and the overall fitness Pf_{WL} of the local optima in the list WL:

$$Pf_i = \frac{1}{C^i_{max}} \tag{2}$$

$$Pf_{WL} = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{C^i_{max}} \tag{3}$$

$$P_{follow} = \begin{cases} 0.6, & Pf_i < 0.9 * Pf_{WL} \\ 0, & otherwise \end{cases} \tag{4}$$

where C^i_{max} is the feature pair dissimilarity calculated for the bee *i* and *n* is the total number of local solutions in the WL.

In order to provide exhaustive search for all the local optima by the discussed ABC with Big Valley Landscape Exploration algorithm the number of visits of each feature pair as a possible false match is taken into account. The local solution is removed from the WL list after a predefined maximum number of unsuccessful visits are made thus providing another area in the search space to be explored. Removing a solution from the WL list marks it as a false match depending on it current fitness. That particular solution is also put in a list of pairs not be used for fitness comparison with the other local optima.

IV. EXPERIMENTAL FRAMEWORK AND RESULTS

The described 3D reconstruction approach using a feature refinement based on ABC with Big Valley Landscape Exploration is implemented in C++ using OpenCV library.

The algorithm is verified using two test data sets with three version of each dataset containing different number of images [26]: "Temple" data set with 312 (full), 47 (ring) and 16 (sparse) images and "Dino" data set 363, 48 and 16 images respectively.

Step 1: $P_{follow} \leftarrow 0.00$
Step 2: $S_x \leftarrow$ solution found at previous iteration.
Step 3: If $Pf_i < 0.90 * Pf_{WL}$
  $P_{follow} \leftarrow 0.60$
Step 4: Generate a random number $r \leftarrow [0,1]$.
Step 5: If $r < P_{follow}$
  $S_x \leftarrow$ Pick a solution from *WL* using a dance selection strategy.
Step 6: Return S_x.

Figure 3. The algorithms for ObserveNSelectDance

The results for the point cloud generated for the sparse, ring and full Temple dataset are shown on fig. 4, fig. 5 and fig. 6 respectively. Table I presents the results obtained for the two datasets at each step of the 3D reconstruction providing the number of feature points detected, the number of feature matches, the number of false matches and the number of the 3D points calculated for each set of images.

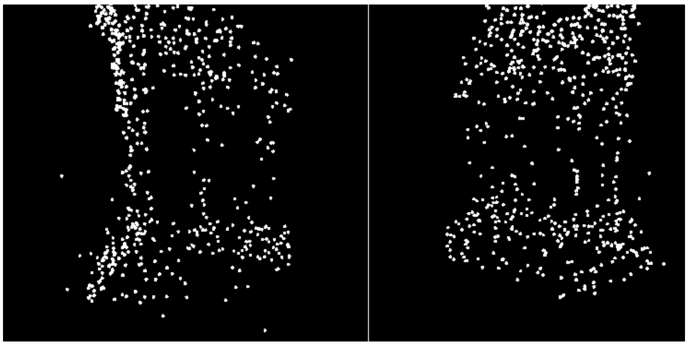

Figure 4. Point cloud for Temple data set with 16 images

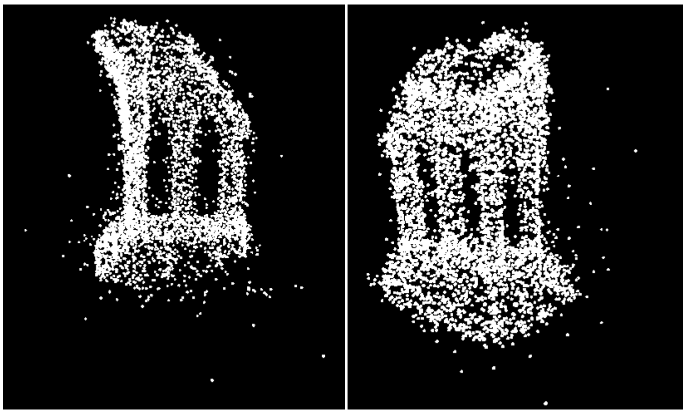

Figure 5. Point cloud for Temple data set with 47 images

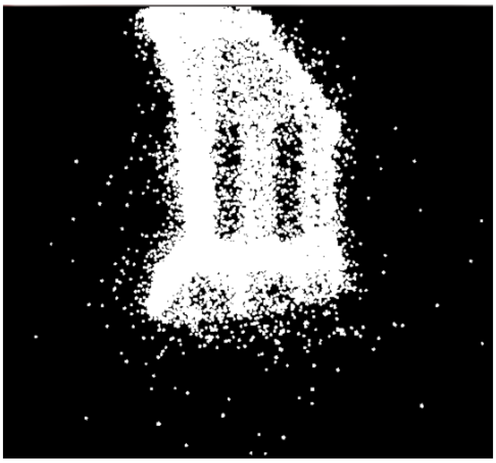

Figure 6. Point cloud for Temple data set with 312 images

The results for the relevant number of feature points, feature matches, false matches and 3D point using SURF or SIFT for feature detection are given in table II. The computational time required for each of the 3D reconstruction steps is shown in table III.As seen from the results the number of the input images used for the 3D recovery process influences the results. The bigger the number of the input images, more feature points are detected and respectively the total number of the reconstructed 3D points in the cloud also increases.

For both tested image datasets the ABC algorithm provides relevant reduction of the number of the detected feature points thus allowing the reconstructed 3D point cloud to comprise a clustered version of only the most reliable feature matches.

By detection and filtering of the false matches the next steps of the 3D recovery are refined both in terms of the time required and the accuracy achieved. The processing and estimation of the 3D coordinates based only on the sustainable feature correspondences reduce the computations and improves the rectification and triangulation results.

The quality of the resulting 3D model as well as the speed of image dataset processing is very sensitive to the feature detection stage. The comparison of using SURF and SIFT for feature detection shows that SURF is faster and due to the utilization of integral image and improved construction of the feature descriptors. But on the other hand the number of features detected using SURF increases more than twice compared to SIFT thus also increasing the number of feature correspondences to be verified and filtered.

TABLE I. RESULTS OF 3D RECONSTRUCTION FOR THE TEST DATASETS

Data set	Number of images	Number of feature points	Number of feature matches	Number of false matches	Number of 3D points
Temple Sparse	16	13 479	780	306	620
Dino Sparse	16	6 523	1 020	253	683
Temple Ring	47	39 973	21 884	3 399	6 221
Dino Ring	48	19 474	18 107	4 152	3 107
Temple Full	312	243 736	330 222	180 440	33 184
Dino Full	363	139 884	444 364	268 912	15 762

TABLE II. RESULTS FOR THE 3D RECONSTRUCTION WITH SURF AND SIFT FEATURE DETECTOR ALGORITHM

Data set	Algo-rithm	Number of feature points	Number of feature matches	Number of false matches	Number of 3D points
Temple Ring	SIFT	16 794	15 150	1 560	2 433
	SURF	39 973	21 884	3 399	6 221
Temple Full	SIFT	104 511	263 455	72 824	13 570
	SURF	243 736	330 222	180 440	33 184

TABLE III. TIME REQUIRED FOR THE EACH OF THE 3D RECONSTRUCTION STAGES

Data set	Feature detection [sec]	Feature match [sec]	3D point cluster [sec]	Triangu-lation [sec]	Total time [sec]
Temple Sparse	25.7	3.7	0.1	0.1	29.6
Dino Sparse	18.3	1.4	0.1	0.2	20.0
Temple Ring	75.8	49.8	38.1	3.5	168.0
Dino Ring	54.7	23.5	16.8	3.6	98.7
Temple Full	457.0	3 028.0	4 578.0	80.0	8 144.0
Dino Full	390.0	1 826.0	3 554.0	143.0	5 913.0

V. CONCLUSION AND FUTURE WORK

The problem for 3D scene reconstruction by multi view images requires a precise detection of feature points and feature correspondence so that a point cloud of 3D points to be accurately calculated and used for precise 3D scene model. As the number of images in the input data set grows, the computational time for each of the stages of the 3D reconstruction pipeline increases. Thus the feature detection and feature matching stages become very important both for the final results from the reconstruction and the performance parameters in in terms of the computational time required. The suggested metaheuristics approach based on Artificial Bee Colony (ABC) with Big Valley Landscape Exploration provides a fast heuristics based approach for false match filtering that is an alternative to the RANSAC or similar methods for feature pair refinement. It utilizes a population based metaheuristics that combines the global exploration of the search space with a local diversification and evaluation of the local optima. The suggested solution has a general advantage over other possible approaches in terms of possibilities for speed up and decrease of the required time using a parallel computational model.

REFERENCES

[1] J. Xiao, “3D reconstruction is not just a low-level task: retrospect and survey”, MIT9.S912: What is Intelligence?, Technical Report, 2012.

[2] F. Remondino, S. El-Hakim, “Image-based 3D modelling: a review”, The Photogrammetric Record, Vol.21, №115, Sept. 2006, pp. 269÷291.

[3] F. Blais, “Review of 20 years of range sensor development”, Journal of Electronic Imaging, Vol.13, №1, 2004, pp. 231–240.

[4] R. Hartley, A. Zisserman, Multiple view geometry in computer vision, Cambridge University Press, 2003.

[5] J. Li, E. Li, L. Xu, Y. Chen, Visual 3D Modeling from images and videos, Intel Labs China, June 3, 2010.

[6] A. Saxena, S. Chung, A. Ng, “3-D depth reconstruction from a single still image”, International Journal of Computer Vision, Vol. 76, 2008, pp. 53÷69.

[7] D. Scharstein, P. Szeliski, “A taxonomy and evaluation of dense two-frame stereo correspondence algorithms”, International Journal of Computer Vision, Vol.47, №1, 2002, pp. 7÷42.

[8] S. Seitz, B. Curless, J.Diebel, D.Scharstein, “A comparison and evaluation of multi-view stereo reconstruction algorithms”, Proc, of IEEE Conference on Computer Vision and Pattern Recognition, Vol. 1, June 2006, pp. 519÷528.

[9] D. Forsyth, J.Ponce, Computer vision: a modern approach, New York: Prentice Hall, 2003.

[10] R. Hartley, A. Zisserman, Multiple view geometry in computer vision, Cambridge University Press, March 2004.

[11] T. Tuytelaars, K. Mikolajczyk, Local invariant feature detectors: a survey, Now Publishers Inc., 2008.

[12] D. Lowe, “Distinctive image features from scale-invariant keypoints”, International Journal of Computer Vision , Vol.60, №2, 2004, pp. 91÷110.

[13] H. Bay, T. Tuytelaars, L. Van Gool, “SURF: speeded up robust features”, Proc. of 9th European Conference on Computer Vision, Graz, Austria, May 2006, pp. 404÷417.

[14] M. Fischler, R. Bolles, “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography”, Communications of the ACM, Vol.24, №6, 1981, pp. 381÷395.

[15] P. Torr, A. Zisserman, “MLESAC: a new robust estimator with application to estimating image geometry”, Comput Vision and Image Understanding, Vol.78, No.1, 2000, pp. 138–156.

[16] O.Chum, J. Matas, “Matching with PROSAC – progressive sample consensus”, Proc. of IEEE Conf. on Computer Vision and Pattern Recognition, June 2005, Vol. 1, pp. 220÷226.

[17] H. Gao, B. Li, C. Wu, C. Zhou, “False matches filtering methods for binocular vision”, Proc. of 7th IEEE World Congress on Intelligent Control and Automation, June 2008, pp. 8477÷8482.

[18] M. May, M. Turner, T. Morris, “Analysing false positives and 3D structure to create intelligent thresholding and weighting functions for SIFT features”, Lecture Notes in Computer Science Vol. 7087, pp. 190÷201.

[19] F. Glover, G. Kochenberger, Handbook of metaheuristics, Springer Science & Business Media, 2003.

[20] J. Dréo, A. Pétrowski, P. Siarry, E. Taillard, Metaheuristics for hard optimization, Springer-Verlag, Berlin, 2006.

[21] C. Blum, A. Roli, “Metaheuristics in combinatorial optimization: overview and conceptual comparison”, ACM Computing Surveys, Vol.35, №3, 2003, pp.268÷308.

[22] M. Birattari, L. Paquete, T. Stiitzle, K. Varrentrapp, “Classification of metaheuristics and design of experiments for the analysis of components”, 2001.

[23] I. BoussaïD, J. Lepagnot, P. Siarry, “A survey on optimization metaheuristics”, Information Sciences, Vol. 237, 2013, pp. 82÷117.

[24] D. Karaboga, B. Basturk, “A powerful and efficient algorithm for numerical function optimization: Artificial Bee Colony (ABC) algorithm”, Journal of Global Optimization, Vol.39, №3, 2007, pp. 459÷471.

[25] L. Wong, C. Puan, M. Low, Y. Wong, C. Chong, “Bee colony optimisation algorithm with big valley landscape exploitation for job shop scheduling problems”, International Journal of Bio-Inspired Computation, Vol.2, №2, 2010, pp. 85÷99.

[26] S. Seitz, B. Curless, J. Diebel, D. Scharstein, R. Szeliski, Multiview stereo datasets, http://vision.middlebury.edu/mview/data/