

PARALLEL ALGORITHMS FOR EFFECTIVE CORRESPONDENCE PROBLEM SOLUTION IN COMPUTER VISION*

© 2017 г. S.A. Tushev, B.M. Sukhovilov

South Ural State University

(Russian Federation, 454080 Chelyabinsk, 76 Lenin avenue)

E-mail: science@tushev.org, sukhovilovbm@susu.ru

Received: 01.05.2017

We propose new parallel algorithms for correspondence problem solution in computer vision. We develop an industrial photogrammetric system that uses artificial retroreflective targets that are photometrically identical. Therefore, we cannot use traditional descriptor-based point matching methods, such as SIFT, SURF etc. Instead, we use epipolar geometry constraints for finding potential point correspondences between images. In this paper, we propose new effective graph-based algorithms for finding point correspondences across the whole set of images (in contrast to traditional methods that use 2-4 images for point matching). We give an exact problem solution via superclique and show that this approach cannot be used for real tasks due to computational complexity. We propose a new effective parallel algorithm that builds the graph from epipolar constraints, as well as a new fast parallel heuristic clique finding algorithm. We use an iterative scheme (with backprojection of the points, filtering of outliers and bundle adjustment of point coordinates and cameras' positions) to obtain an exact correspondence problem solution. This scheme allows using heuristic clique finding algorithm at each iteration. The proposed architecture of the system offers a significant advantage in time. Newly proposed algorithms have been implemented in code; their performance has been estimated. We also investigate their impact on the effectiveness of the photogrammetric system that is currently under development and experimentally prove algorithms' efficiency.

Keywords: computer vision, photogrammetry, correspondence problem, parallel algorithms, maximum clique problem, epipolar geometry.

FOR CITATION

Tushev S.A., Sukhovilov B.M. Parallel Algorithms for Effective Correspondence Problem Solution in Computer Vision. *Bulletin of the South Ural State University. Series: Computational Mathematics and Software Engineering*. 2017. vol. 6, no 2. pp. 49–68. DOI: 10.14529/cmse170204.

Introduction

Effective matching between images of a 3D object points (also known as point correspondence problem) is one of the key problems in computer vision [1]. There are several ways to solve the correspondence problem, such as feature detection, a method based on the epipolar geometry restraints, and the combination of these two methods. Feature detection is the most popular approach for solving the point correspondence problem. These methods analyze images and look for *features*, such as corners, ridges, contrast points etc. A numerical value, called *descriptor*, is calculated from the image data based on some vicinity of the feature. This approach allows us to find point correspondences almost instantaneously by matching the numerical values of descriptors, as local feature detectors are robust against most image transformations (caused by movement of the camera) given that lighting conditions remain the same

* The paper is recommended for publication by the Program Committee of the International Scientific Conference "Parallel Computational Technologies (PCT) 2017".

and the camera position does not change drastically. Another advantage is the fact that feature detection does not require any knowledge of the camera relative position.

Unfortunately, feature detection methods only work well in stable lighting conditions. If the light source moves, features might alter their appearance, structure of shadows, or disappear. This leads to changes in numerical values of the descriptors that make it impossible to find matches. Harris detector [2], SIFT [3] and SURF [4] are among the most popular and widely used feature detectors and descriptors.

It is also possible to solve correspondence problem using epipolar geometry [1, 5]. As it is shown in Section 1.1, if we choose an image point in the first image, the corresponding image point in the second image may only lie on the epipolar line [1]. Epipolar line can be easily calculated, provided that we know camera intrinsic parameters with the relative position and orientation of a pair of cameras. Thus, it is possible to narrow the search area to a single epipolar line (for a pair of images), or even to the crossing point of two epipolar lines (for a triplet of images). An approach based on epipolar geometry does not have any special requirements to lightning conditions of the image contents; however, it requires cameras' parameters and their relative position to be known. Fraser in [5] describes various approaches of finding point correspondence based on epipolar geometry in 2- and 3-dimensional space. He also states that computational difficulty grows rapidly when the number of images is increased.

A group of special (typically industrial) photogrammetric systems uses retroreflective targets (to identify the key points of the object being measured) and a flash to ensure that those targets are easily distinguishable in an industrial environment that often has insufficient lighting conditions. This directed light drastically changes the structure of shadows in the image, thus changing the values of feature descriptors. In this case, point correspondence problem can only be solved by means of epipolar geometry. Unfortunately, in practice the point may lie slightly out of the calculated epipolar line due to some degree of uncertainty in estimated relative camera position or due to camera's manufacturing imperfections. Thus in real-life applications it is better to search for a point in some δ -vicinity of the epipolar line. Moreover, as the number of images and targets increases, the search area may contain more other points or artefacts (such as glares or reflections mistakenly recognized by the system as retroreflective targets), so the correspondence problem becomes non-trivial. As we show in Sections 1.2 and 1.3, the exact solution of point correspondence problem belongs to clique problem in a multipartite graph and has exponential complexity.

We develop an industrial photogrammetric system that uses retroreflective targets [6, 7], so we have to use epipolar geometry to solve the point correspondence problem. Our system should work on a mid-high level laptop. Moreover, due to specifications, the whole process of the 3D reconstruction (from uploading images from the camera to obtaining accurate 3D coordinates of the targets) should take no more than 5-6 minutes. This requires us to develop some new effective point matching algorithms based on epipolar geometry, which would allow us to solve the problem in acceptable time. To achieve this, we use parallelization along with an iterative scheme that adjusts the accuracy for estimation of targets and cameras' positions. This allows using heuristic clique finding algorithms (particularly but not exclusively their parallel versions) that finally lead us to an exact solution of the point correspondence problem in acceptable time.

It is impossible to parallelize the whole problem of 3D reconstruction due to its specific nature. However, we may significantly increase overall efficiency of the process by performing

parallel data processing at several stages, thanks to the widespread parallel architecture (nowadays most of the mid-high level laptops contain multi-core CPUs, as well as discrete GPUs).

In this paper we will cover different aspects of the problem. We start with considering the theoretical solution of the point correspondence problem by the means of epipolar geometry (Sections 1.1-1.3) and estimating its complexity (Section 3.1). Section 1.4 describes our iterative scheme that allows us to use near-polynomial time clique finding algorithms that are described in section 1.5. Section 2 covers our software implementation of the algorithms. Test results are provided in Section 3. Our findings are summarized in the final section, “Conclusion and future work”.

Related work. Point matching is one of the key tasks in computer vision. Most works rely on having photometrically distinct features that allow computing a descriptor from image data [8, 9]. SIFT [3] and SURF [4] are among the most popular and widely used methods. We can name BRISK [8] and FREAK [11] among the developments of the recent years. There are also works that consider hybrid approach, such as [12]

However, due to the usage of photometrically identical targets, we should rely solely on epipolar geometry. One of the most fundamental works that cover multiple aspects of computer vision, including epipolar geometry, is [1]. There are two primary approaches to finding point correspondences via epipolar geometry: so-called clustering method, which operates within 3D space, and plane-based methods. Various techniques that utilize the clustering method are considered in [13, 14]. Among the papers that describe the plane-based approach, we can name both classical works by Maas [15, 16], Zhang et al. [17], as well as recent papers, such as [18, 19]. Fraser in [20] describes “presently adopted approaches for close-range photogrammetric network orientation, along with three categories of processing for 3D point determination”.

Modern photogrammetry systems, such as V-STARS [21] or Agisoft PhotoScan [22] are likely to use similar approaches to finding point correspondences. However, due to commercial considerations, very little information concerning the internal target detection mechanisms, point matching algorithms etc. is available. Among the few works that consider various aspects of photogrammetry systems we can name [5] as the closest one to our photogrammetric system.

In this paper, we propose new effective graph-based algorithms that utilize epipolar geometry for finding point correspondences across the whole set of images. This is, to the best of our knowledge, a new approach; other techniques that utilize epipolar geometry for point matching are based on 2, 3 or 4 images [15, 23].

An extension of 4-images approach to an arbitrary number of images has been reported by Dold and Maas in [16], although they state high computational complexity of their method that “grows exponentially with the number of images and would hardly be tolerable in an application with 18 images”. However, our method, despite utilizing similar principles of epipolar geometry, is different: we use a graph as a mathematical model of point correspondence problem, so that each clique in the graph represent a potential matched point across all images in the set. Thus we can “automatically” get all point matches, by choosing the largest disjoint cliques from the graph. We also consider additional information such as clique weight to choose the best candidates for establishing point correspondence. While we encountered the same exponential computational complexity within our superclique approach, we developed another approach that allows to find exact point correspondences across the whole range of images in a reasonable time. It includes our new parallel graph-based algorithms along with an iterative

scheme. Thus it makes it possible to find an exact solution of the point correspondence problem for hundreds of images in a fast and efficient way.

Another approach that utilizes epipolar geometry, tree hierarchy, graph theory and clustering was reported in [24].

1. Theory

1.1. Epipolar geometry and point correspondence

Epipolar geometry describes the so-called stereoscopic pair – two cameras with known relative pose and orientation that observe the same three-dimensional object.

Suppose that point X is simultaneously observed by two cameras: the left camera with optical center O_L and the right camera with optical center O_R (Fig. 1). The real (three-dimensional) point X is projected on the left and right image planes as X_L and X_R correspondingly. The baseline, which connects cameras' centers O_L and O_R , intersects with image planes in points e_L and e_R , which are known as epipoles [1].

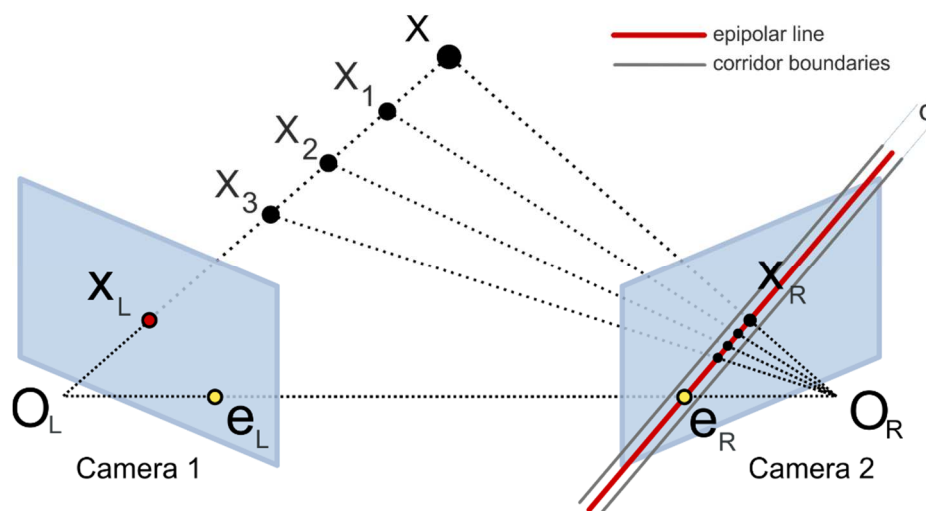


Fig. 1. Epipolar geometry and epipolar corridors

Apparently, each 3D point X has its own epipolar plane that goes through baseline and through X (thus through X_L as well). The intersection of the epipolar plane with the right image plane forms an epipolar line that goes through X_R .

Given a case where real 3D coordinates of X are unknown (for instance, processing a set of images when we have only two-dimensional coordinates of X_L), we cannot unambiguously locate X_R on the right image due to the loss of depth when projecting the point to an image plane. However, it is still possible to narrow the search area of the corresponding point X_R from the whole image to epipolar line.

An epipolar line (in the form of $ax + by + c = 0$) that corresponds to X_L in the right image can be described as (1), according to [25]:

$$e = F \cdot X_L, \quad (1)$$

where e is the column vector of coefficients a , b and c , X_L is the column of homogeneous point coordinates in the left image, F is a fundamental matrix calculated for the given pair of images by camera intrinsic parameters K and camera relative pose and orientation, which are represented by essential matrix E [1]:

$$F = K^{-1} \cdot E \cdot K, \quad (2)$$

Essential matrix E can be calculated as (3):

$$E = \begin{bmatrix} 0 & -tx(3) & tx(2) \\ tx(3) & 0 & -tx(1) \\ -tx(2) & tx(1) & 0 \end{bmatrix} \cdot R, \quad (3)$$

where tx is the normalized translation vector between the cameras' centers. Given the pair of translation vectors t_l and t_r which define location of the cameras' centers in some global coordinate system, we can calculate tx from (4) and (5):

$$tx = \frac{tx'}{\sqrt{\|tx'\|}}, \quad (4)$$

$$tx' = t_r^T - R \cdot t_l^T, \quad (5)$$

where R is the rotation matrix between the two cameras. It can be found by (6) from a pair of rotation matrices A_l и A_r in some global coordinate system:

$$R = A_r \cdot A_l^T. \quad (6)$$

It is evident that epipolar line on the right image can go through several different image points, thus making the point correspondence problem to be non-trivial.

In order to use the epipolar geometry, camera poses and orientations should be known before making calculations. In our photogrammetric system we use so-called *coded targets* that provide us with the global coordinate system for all images [26].

Unfortunately, real cameras are not as perfect as their mathematical models are. In the right picture, the point may lie slightly off the calculated epipolar line due to some degree of uncertainty in the estimated relative camera position or due to the camera's manufacturing imperfections. Thus in practice it is better to use the *epipolar corridor* of width d that is more likely to contain the point.

Switching from epipolar lines to epipolar corridors brings even more nearby candidate image points into the correspondence search area. Moreover, some of the images may contain extra artefacts inside of the corridor (such as glares or reflections mistakenly recognized by the system as retroreflective targets). Thus, point matching algorithm should be robust and insensitive to various disturbances and artefacts.

1.2. Epipolar geometry and point correspondence

We propose to use the multipartite graph $G=\{V;E\}$ as a mathematical model of the point correspondence problem. Graph G consists of a set of image points V (graph *vertices*) and a set of possible correspondences between points E (graph *edges*). A pair of points is considered adjacent if each point of the pair lies within the epipolar corridor calculated for the opposite point of the pair. Each single image forms a separate part of the graph (because vertices that represent points in the same image cannot be adjacent), so the whole graph becomes k -partite, where k is the number of images.

The algorithm for building the graph from epipolar data is given below. For each pair of images we calculate fundamental matrix F using (2) — (6). Then we process each pair of image points in the “*left*” and “*right*” images of the pair. For each point of the pair we calculate epipolar line on the “*opposite*” image (the one that the point does not belong to) and Euclidian distance to this line for the other point of the pair. When the mean pixel distance to the

epipolar lines (7) is not greater than the half of the allowed width of the epipolar corridor, two vertices in G become adjacent with the edge weight w , equal to (7):

$$w = \frac{1}{2} \left(\frac{|e_l \cdot p_r|}{\sqrt{e_l(1)^2 + e_l(2)^2}} + \frac{|e_r \cdot p_l|}{\sqrt{e_r(1)^2 + e_r(2)^2}} \right), \quad (7)$$

Here e_l, e_r are the column vectors of coefficients (1) for epipolar lines in the left and the right image correspondingly; p_l, p_r are homogeneous pixel coordinates of image points in the left and the right images (8):

$$p = \begin{bmatrix} x_{px} \\ y_{px} \\ 1 \end{bmatrix}, \quad (8)$$

This algorithm has a large number of independent calculations and possesses high cyclomatic complexity, which makes it a good candidate for parallelization. We describe our parallel implementation of this algorithm in Section 2.1.

1.3. Epipolar geometry and point correspondence

The exact, theoretical solution of the point correspondence problem is a superclique – a clique in a supergraph, a graph where each vertex represents a clique in graph G (a clique is a subset of graph where every two distinct vertices are adjacent).

At the first step, we should initially find all maximal cliques in G with the number of vertices greater than or equal to some pre-defined value t (which is used to filter out artefacts). Each clique $C_i = \{V_i\}$ represents a possible 3D candidate point; its vertices V_i represent possible images of this point in various pictures that satisfy epipolar restraints. Besides the set of vertices, each clique obtains its unique identifier.

At the next step, we build a supergraph – graph S with vertices C_i . A pair of cliques C_i and C_j are adjacent in S if they do not have any common vertices V_i (i.e. they are disjoint: $C_i \cap C_j \equiv \{V_i\} \cap \{V_j\} = \emptyset$)

Superclique is the maximum clique of S that consists of disjoint cliques $\{C_i\}$ and represents the final set of 3D points. At this step each C_i from superclique represents an actual 3D point identified across all the pictures in the set. The superclique is the exact solution of the graph-based point correspondence problem. This approach requires two steps.

We have introduced t before as a parameter that controls filtering various artefacts, such as glares, blinks etc. It is assumed that those disturbances are visible in no more than $t-1$ pictures in the series. If a bright point that resembles retroreflective target is visible in t or more images, it can be considered as an additional 3D feature that can be used to improve accuracy of the computations. The value of t partially depends on shape of object being measured. In our work we typically use $t = 4 \dots 6$.

Unfortunately, the usage of the superclique approach in practice is limited due to computational complexity. Nowadays both the maximum and the maximal clique problems are considered to be NP-complete [27]. While listing all maximal cliques in G is hard, finding maximum clique in S is much harder (as S may contain much more edges and vertices than G). We provide some experimental data and estimations in Section 3.1.

According to our research, at the current level of technology it is impossible to use the superclique approach for practical tasks, except for cases with low number of targets and cameras that are not of any practical industrial interest.

1.4. Exact solution of the problem using an iterative scheme

Our photogrammetric system must perform measurements in a specified amount of time. Our tests show that the superclique approach requires large amounts of time, which exceed the acceptable limits even for small-sized datasets.

In order to meet the time requirements we develop an iterative scheme for finding point correspondences that uses approximate superclique finding methods, triangulation of 3D points with filtering of outliers, bundle adjustment [28] of point and cameras' parameters with backprojection techniques that allow finding new point images in the pictures. This scheme makes it possible to use approximate clique finding algorithms of near-polynomial time complexity at every single iteration. Eventually, this scheme provides the exact solution of the point correspondence problem (and estimation of their 3D coordinates) in a reasonable amount of time.

1.5. Parallel local graph-based algorithm for finding correspondence points

We propose a new parallel algorithm for finding point correspondence. The key idea of this algorithm is building a number of small *local* graphs instead of processing one large *global* graph. This is the further development of our PLG family of algorithms, which are described in [29], with the consideration of gathered experience and criticism.

General idea. Most algorithms of this family process the whole set of point images in series. At each iteration, algorithm builds a small-sized local graph for the current image point, which becomes the *seeding* vertex, or seeding point (*SP*). All other image points that lie within the corresponding epipolar corridors in all other pictures are also added to the local graph as vertices, adjacent to the seeding point (*NP*). The corresponding edge weight is calculated by equation (7). In the next step, the algorithm analyses epipolar geometry of the newly added point images and adds more edges to the graph between existing vertices. Some members of the PLG family even expand the graph at this step by adding new vertices that are connected to *NP* (but not to *SP*). Finally, the algorithm finds the maximum clique in the graph. Vertices that form this maximum clique represent the same 3D point across all pictures [12].

An intuitive (geometrical) explanation of the working principles of the point matching algorithms that are based on local graphs can be formulated as the following:

1. An image point (*SP*) is picked in a certain way from the set of all image points across all pictures;
2. Epipolar lines in all other images are calculated for *SP*;
3. All other image points from other pictures that lie within some δ -vicinity of the epipolar lines are considered as potential corresponding points;
4. New epipolar line across all other images are calculated for each of the potential corresponding points;
5. In each image, we choose an image point that lies at the crossing point of the maximum number of epipolar lines (or in some δ -vicinity of the crossing point);
6. The set of chosen image points represents the same 3D point with the maximum degree of probability (which is proven by the experiments).

Our method conceptually resembles point matching between a triplet of images [1, 5], however, it takes more data to consider (we analyze all pictures that contain the necessary

scene fragment). Thus, potentially, it may achieve more accurate point matching than triplet-based methods. Our experiments have proven this method to be robust and accurate.

Different algorithms in PLG family pick the seeding point in a different way. At present, we consider the algorithm called PLG4 to be the most optimal. At every iteration, it picks the vertex with the highest degree in the global graph (i.e. the image point with the maximum number of candidate image points in the proximity of its epipolar lines), which has not been included to any cliques before, as the seeding vertex for the current local graph. The local graph in PLG4 must contain only vertices adjacent to *SP*. We consider this algorithm as more accurate rather than PLG1 or PLG2 described in [29], thus our parallel point correspondence algorithm will be based upon these principles.

Our parallel point correspondence algorithm. Most algorithms in PLG family work with the converging set of image points (because image points that have formed a clique are excluded from consideration). This feature makes data to be dependent from the previous step, thus making parallelization of this algorithm more difficult. Besides, the exclusion of image point from further analysis it could potentially lead to the loss of other cliques of the same size, but with lesser summary weight of edges, which are more preferable.

```

parallel_for_each (point  $\in$  2DPoints)
   $G_p := \emptyset$ 
  for_each neighbour  $\in$  point.Targets
     $G_p \leftarrow$  vertex(neighbour)
     $G_p \leftarrow$  edge(point  $\leftrightarrow$  neighbour)
  end_for_each
  for_each (neighbour  $\in$  point.Targets)
    for_each (link  $\in$  neighbour.Targets)
      if ( $G_p \ni$  link as vertex)
         $G_p \leftarrow$  edge(link  $\leftrightarrow$  neighbour)
      endif
    end_for_each
  end_for_each
  grouped_point := find_maximum_clique_of_min_weight( $G_p$ )
  if (size(grouped_point)  $\geq$  t )
    { cliques }  $\leftarrow$  grouped_point
  endif
end_parallel_for_each
  cliques := sort_by_size (cliques)
  foreach clique  $\in$  cliques
    foreach clique2  $\in$  cliques[clique...end]
      if (clique  $\cap$  clique2  $\neq \emptyset$ )
        cliques := cliques \ clique2
      endif
    end_for_each
  end_for_each

```

Fig. 2. Parallel PLG algorithm pseudo-code

In our new algorithm, we decided not to exclude the image points that have formed cliques from further processing, so the input dataset does not converge with time. This not only simplifies parallelization of algorithms, but also allows creating a brand new approach based on

local graphs as well. A sequential implementation of this approach is also possible, although it is far less effective.

Therefore, our parallel algorithm for finding point correspondence consists of two stages. At the first stage, we build local graphs for *all* image points in a parallel. The algorithm also tries to find maximum clique in each local graph that it builds. Once the parallel stage is complete, we refine the resulting set of cliques to get rid of the clique intersections. Our new algorithm is provided in pseudo-code given in Fig. 2.

`find_maximum_clique_of_min_weight` procedure searches for the maximum clique in G_p , choosing the one with lesser summary edge weight in case of two cliques of the same size being compared. From practical point of view, clique with lesser summary edge weight has its vertices located closer to the corresponding epipolar lines, thus being more preferable over the clique with greater summary edge weight.

2. Implementation

We implement both epipolar-corridors-to-graph and the point correspondence algorithms in C++11. We use Microsoft Visual Studio 2015 C++ Compiler to build our code.

In the current version of our software implementation of algorithms, we use Microsoft C++ Concurrency Runtime\Parallel Patterns Library for the purpose of parallelization. Our choice is based on the requirements for photogrammetric software being developed: it must run under MS Windows on a mid-high level laptop. Current software only uses CPU cores for performing calculations. We consider tailoring our algorithms to GPU architecture as one of the directions for the future work.

Other modules of our photogrammetric system that are not described in this paper are implemented with either GNU Octave or C++.

2.1. Implementation of parallel algorithm for graph building from epipolar geometry

Initially, we had been implementing epipolar-corridors-to-graph algorithm with the use of OpenCV 3.1 Library [30] for matrix operations (because OpenCV is the inner standard of our photogrammetric system). However, profiling showed that OpenCV matrix performance is low, so we chose to use Eigen 3.2.8 library [31] for matrix operations which gives us extra speedup.

Our parallel algorithm is given in pseudo-code in Fig. 3.

We modify the initial algorithm described in Section 1.2 by splitting it into several steps. First, we build a batch of tasks (a pair of images + corresponding fundamental matrix) in a parallel. Then we process this batch (as well in a parallel) by calculating (7) for each pair of image points from current pair of images and writing down pairs that lie within the corridor. Finally, we build a graph (in a form of edge list, as required by design rules) based on the data from the second step.

These modifications make calculation of fundamental matrices to be independent from the processing of image point pairs. Besides that, it reduces cyclomatic complexity of algorithm from 4 nested cycles to 3 and gives us more potential for parallelization.

```

pairs<i,j> := all_possible_pairs(1...Nimages)
parallel_for_each(pairs)
  F := equations (2) - (6)
  tasks ← <i, j, F>
end_parallel_for_each
parallel_for_each(tasks)
  for_each( pl ∈ image_points(i) )
    er = F · pl
    for_each( pr ∈ image_points(j) )
      el = F · pr
      if ( equation (7) ≤ ec_halfwidth )
        graph ← edge(pl, pr)
      endif
    end_for_each
  end_for_each
end_parallel_for_each

```

Fig. 3. Parallel algorithm for graph building from epipolar geometry

2.2. Details on implementation of parallel algorithm for point correspondence problem

We implement our newly proposed graph-based point correspondence algorithm as a part of clique finding module, which also implements other clique finding algorithms described in [29].

The pseudo-code of the new algorithm is given above in Fig. 2.

In order to find the maximum clique, we use modified Konc and Janězič [32] Maximum Clique Algorithm, also known as MCQDyn. We use its reference implementation [33] as the basis of function `find_maximum_clique_of_min_weight` in Fig. 2. Our modifications of this algorithm are described in the next Section 2.3.

Our clique finding module also contains implementation of other sequential point correspondence algorithms described in [29]. We use those algorithms as competitors to our new parallel algorithm. Most of those algorithms (i.e. PLG4) also use MCQDyn to find the maximum clique. However, other sequential algorithms, such as CE, find *all* maximal cliques in a graph. In order to do this, we use Tomita's variation [34, 35] of Brohn–Kerbosch algorithm [36] with pivoting. Our code is based on Ozaki's implementation [37] that uses adjacency lists built upon `unordered_set` from Boost [38].

2.3. Implementation of the modified Konc-Janězič algorithm for the maximum clique with minimal summary edge weight

Most maximum clique algorithms are only capable of finding clique with the maximum number of vertices. In tasks related to epipolar geometry it is also important to consider *the summary edge weight of a vertex set* (which is also called *clique weight*). Edge weight defines how far the vertex is from its corresponding epipolar line, thus clique weight shows how close this vertex set lies to its corresponding epipolar lines. Given two cliques of the same size, we should prefer the one that has lesser summary edge weight.

In the text below we name key differences of our implementation of Konc-Janězič maximum clique algorithm (also known as MCQDyn) that find maximum clique with minimal weight from its reference implementation [33].

We introduce additional data structure for keeping edge weights. It is implemented as `unordered_map<pair<int,int>,double>`, which has search time complexity of $O(1)$. For working with clique weight, it is also necessary to implement the corresponding `clique_weight` function.

Immediately after exiting the recursion call, the reference version of the algorithm checks whether the newly found candidate clique Q can unseat current maximum clique $QMAX$ by the following criterion: $|Q| > |QMAX|$. Our modified criteria is given in Fig. 4. It also skips cliques with the size lower than t (*threshold*), which is used for filtering out artefacts:

```

if  $|Q| \geq$  threshold
  if  $|Q| > |QMAX|$  OR
     $|Q| = |QMAX|$  AND clique_weight(|Q|) < clique_weight(|QMAX|)
       $QMAX := Q$ 
  endif
endif

```

Fig. 4. Modified QMAX unseat criterion

3. Experimental results

Test machine A is equipped with Intel Core i7-6700K processor running at 4.0 GHz (4 cores, 8 threads) and 48 GBs of RAM. We as well estimate integral algorithm efficiency (section 3.4) on machine B, which is equipped with Intel Core 2 Duo CPU running at 2.66 GHz (2 cores, 2 threads) and 4 GBs of RAM. Except where otherwise noted, we provide average running time for a series of $n=10$ launches with the same input data as “running time of algorithm”. All datasets can be downloaded from <https://github.com/tushev/pplgx-sample-data>.

3.1. Estimation of complexity for superclique approach

In order to estimate time complexity for superclique approach, we use one of our test series with 74 pictures and 122 spatially dense retroreflective targets. We obtain graph G with 1 048 vertices and 19 542 edges with edge density of 3,56%. The corresponding *supergraph* S consists of 16 356 vertices и 77 140 880 edges with edge density of 57,67%. It took 74,4 seconds on test machine A just to build S 's adjacency matrix in RAM (with the total program's memory consumption of 6,97 Gb). In comparison, the previous step that found 16 356 maximal cliques in G took only 0,138 seconds. We were unable to find superclique as we had to terminate superclique discovery step after more than 2 hours of computations.

Thus, at the moment, the superclique approach may only be of theoretical interest. It is unsuitable for most practical photogrammetric applications (except for the cases with small number of cameras and targets) due to high time complexity.

3.2. Experimental results for parallel graph building algorithms

Tab. 1 provides average running times for three variants of the epipolar-corridors-to-graph algorithm: classic sequential implementation that uses OpenCV library for matrix operations, parallel implementation with OpenCV and parallel implementation with Eigen library for matrix operations.

Table 1

Running time of graph building algorithms from epipolar geometry, s.

#	# of pictures	Total images of points	Avg. images of points per picture	parallel-cv	parallel-eigen	classic
1	23	2186	95	0,8658	0,3831	5,0344
2	30	1288	42	0,5966	0,3606	2,0730
3	89	896	10	0,4149	0,2776	0,3608

We use three different photographic sessions as input data. The data given in Tab. 1 suggest that our parallel OpenCV implementation runs faster than sequential implementation on dense sessions (like datasets 1&2), while the overheads slow it down on the small-sized sparse dataset 3. Nevertheless, the Eigen-based parallel implementation always runs faster because of better performance of matrix operations. All the tests are performed on machine A.

3.3. Experimental results for point correspondence algorithms for synthetic data

We develop a simulation model of our photogrammetric system to estimate efficiency of point correspondence algorithms. This simulator includes graph generator, which allows to form synthetical graphs with the specified number of “pictures” taken, number of targets and number of artefacts. By varying these parameters, we obtain graphs with different number of vertices, edges and different edge density.

Tab. 2 contains average runtime of point correspondence algorithms on different synthetic graphs on machine A.

Table 2

Running time of point correspondence algorithms, s.

Graph #	Vertices	Edges	Graph edge density	Images	PPLGx, s	PLG4, s	CE, s
1	2 188	71 024	2,97%	23	0,207	0,266	0,444
2	2 175	19 722	0,83%	23	0,024	0,842	0,162
3	2 188	69 284	2,90%	23	0,191	0,27	0,439
4	2 978	42 786	0,97%	20	0,107	0,212	0,338
5	11 112	375 414	0,61%	53	2,936	7,734	4,457
6	20 590	981 122	0,46%	71	14,488	15,507	16,461
7	32 739	1 609 460	0,30%	83	30,332	219,215	40,253

We denote the new parallel point correspondence algorithm as PPLGx, which is described together with sequential PLG4 algorithm in Section 1.5. A sequential CE point matching algorithm is described in [29].

The data in Tab. 2 suggest that our new algorithm is faster in all cases. However, this data does not reflect overall matching efficiency of proposed approaches and should only be used to compare the execution time between algorithms. To estimate overall integral matching efficiency we should test our algorithms as a part of the photogrammetric system.

3.4. Estimation of integral algorithm efficiency

As shown in Section 1.4, our photogrammetric system uses iterative scheme for finding point correspondence that refines data accuracy on each iteration. We have chosen the following variables as the key performance indicators: the total running time, the number of 3D points identified, the mean reprojection error, the total number of point images identified and the number of iterations, denoted as N .

We have picked two test datasets that represent two kinds of situations we may encounter while performing photogrammetric reconstruction. Tab. 3 contains experimental results for a typical photogrammetric session, which is representative for most measurement procedures in our practice.

Table 3

Integral efficiency of the photogrammetric system (“typical” session)

Algorithm	Avg. runtime (Machine A), s	Avg. runtime (Machine B), s	# 3D points	# 2D points	N	Mean reproj. error, px
PPLGx	41,08	135,55	161	1701	17	0,396974
PLG4	45,63	139,92	161	1701	17	0,396974
CE	45,18	141,51	161	1701	17	0,396974

Tab. 4 contains experimental results for our most computationally hard case as of the current date with high spatial density of retroreflective targets. Such situations are not common, but they contain a very large number of candidate points and therefore are the most challenging ones. They also may lead to varying results due to different principles lying beneath matching algorithms.

Table 4

Integral efficiency of the photogrammetric system (“extreme” session)

Algorithm	Avg. runtime (Machine A), s	Avg. runtime (Machine B), s	# 3D points	# 2D points	N	Mean reproj. error, px
PPLGx	187,14	502,59	253	2371	29	0,307259
PLG4	240,39	557,44	253	2379	45	0,305532
CE	269,73	597,10	251	2362	36	0,307569

We use parallel epipolar-geometry-to-graph algorithm with PPLGx and its sequential version with PLG4 and CE.

As we see from Tables 3 & 4, our new parallel algorithm (PPLGx) is the fastest in both cases. Also, in the “extreme” case, it converges the system much earlier than other point correspondence algorithms, and finds the same number of 253 3D points as PLG4. However, in this case, PLG4 identifies 0,33% more image points and gains slightly less reprojection error (around 0,0017 pixels).

Note on overall system accuracy. The overall measurement error of the photogrammetric system mostly depends on final bundle adjustment procedure and the accuracy of measurements of pixel coordinates of the circle targets in the images [26]. The main goal of the point correspondence algorithm is to identify as many targets in the images as possible. Thus, if any of the point correspondence algorithms manages to find more points than the other, the result measurement error of the photogrammetric system (represented in this case by mean

reprojection error) decreases (as shown in the data from Tab. 4). Otherwise, if all algorithms converge to the same result (as in Tab. 3), the final bundle adjustment procedure gains the same accuracy. In this case, the fastest algorithm becomes the most preferable one.

This allows us to create a kind of “switch” in our software, which detects the complexity of the given session and asks the user whether to perform fast but slightly less accurate calculations, or precise but 20-25% slower calculations. If the given session is identified as “typical”, the system always chooses the fast parallel point correspondence algorithm.

Conclusion and future work

In this paper we describe how epipolar geometry may be used to find point correspondences when other methods such as feature detectors are not applicable. We introduce a multipartite graph as mathematical representation of the system and describe how to construct it from epipolar geometry. We also propose parallel implementation of graph building algorithm and estimate its efficiency over sequential implementation.

We propose a new parallel graph-based algorithm for finding point correspondence, which is based on the idea of local graphs of small size. Our tests prove that the new algorithm is faster than other sequential point matching algorithms, both on synthetic data and as a part of the photogrammetric system. Also, it is accurate and it produces the same results for all sparse cases as the other algorithms. However, for spatially dense cases, it may lead to slightly different results with negligible reprojection differences. We believe that fine-tuning of the other parts of the photogrammetric system may mitigate these differences and regard this as one of the directions for the future work.

We consider tailoring our algorithms to GPU architecture as the future work as well.

The work was supported by Act 211 Government of the Russian Federation, contract № 02.A03.21.0011.

References

1. Hartley R., Zisserman A. Multiple View Geometry in Computer Vision. 2nd ed. Cambridge University Press, 2004.
2. Baggio D. et al. Mastering OpenCV with Practical Computer Vision Projects. Packt Publishing, 2012.
3. Bay H., Tuytelaars T., van Gool L. SURF: Speeded Up Robust Features. *Computer Vision and Image Understanding*. 2008. vol. 110, no. 3, pp. 346–359. DOI: 10.1016/j.cviu.2007.09.014
4. Lowe D.G. Object Recognition from Local Scale-Invariant Features. *Proceedings of the Seventh IEEE International Conference on Computer Vision*. 1999. vol. 2, pp. 1150–1157. DOI: 10.1109/ICCV.1999.790410
5. Fraser C.S. Innovations in Automation for Vision Metrology Systems. *Photogrammetric Record*. 1997. vol. 15, no. 90, pp. 901–911.
6. Sukhovilov B. M., Grigorova E. A., Development of a Photogrammetric System for Measuring the Spatial Coordinates of Structural Elements of the Frame of a Low-Floor Tram. *Nauka JuUrGU. Materialy 67-j Nauchnoj Konferencii. Sekcii Jekonomiki, Upravlenija i Prava*. [Science of SUSU. Materials of the 67th Scientific Conference. Section

- of Economics, Management and Law]. Chelyabinsk, Publishing of the South Ural State University, 2015, pp. 458–463. (in Russian)
7. Tushev S.A., Sukhovilov B.M.. Some Ways to Improve the Performance of Automatic Calibration of Digital Cameras. *Molodoj Issledovatel: Materialy 2-J Nauchnoj Vystavki-Konferentsii Nauchno-Tekhnicheskikh i Tvorcheskikh Rabot Studentov*. [Young Researcher: Materials of the 2nd Scientific Exhibition-Conference of Scientific, Technical and Creative Works of Students]. Chelyabinsk, Publishing of the South Ural State University, 2015, pp. 434–439. (in Russian)
 8. Hartmann W., Havlena M., Schindler K. Recent Developments in Large-Scale Tie-Point Matching. *ISPRS Journal of Photogrammetry and Remote Sensing*. 2016. vol. 115, pp. 47–62. DOI: 10.1016/j.isprsjprs.2015.09.005
 9. Remondino F., Spera M. G., Nocerino E., Menna F., Nex F. State of the Art in High Density Image Matching. *The Photogrammetric Record*. 2014. vol. 29(146), pp. 144–166. DOI: 10.1111/phor.12063
 10. Leutenegger S., Chli M., Siegwart R. BRISK: Binary Robust invariant scalable keypoints. *Proceedings of the 2011 International Conference on Computer Vision (ICCV '11)*. IEEE Computer Society, Washington, DC, USA, 2011. pp. 2548–2555. DOI: 10.1109/ICCV.2011.6126542
 11. Ortiz R. FREAK: Fast Retina Keypoint. *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (CVPR '12)*. IEEE Computer Society, Washington, DC, USA, 2012. pp. 510–517. DOI: 10.1109/CVPR.2012.6247715
 12. Fraser C. S., Cronk S. A Hybrid Measurement Approach for Close-Range Photogrammetry. *ISPRS Journal of Photogrammetry and Remote Sensing*. 2009, vol. 64(3), pp. 328–333. DOI: 10.1016/j.isprsjprs.2008.09.009
 13. Leung C., Lovell B. C. 3D Reconstruction through Segmentation of Multi-View Image Sequences. *Proceedings of the 2003 APRS Workshop on Digital Image Computing*. 2003. pp. 87–92. Available at: <http://espace.library.uq.edu.au/view/UQ:10960> (accessed: 29.11.2016)
 14. Qiqiang F., Guangyun L. Matching of Artificial Target Points Based on Space Intersection. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. Beijing 2008, vol. XXXVII, part B5, pp. 111–114.
 15. Maas, H.-G. Complexity analysis for the establishment of image correspondences of dense spatial target fields. *International Archives of Photogrammetry and Remote Sensing*. 1992. vol. 29, pp. 102–107.
 16. Dold J., Maas, H.-G. An Application of Epipolar Line Intersection in a Hybrid Close-Range Photogrammetric System. *International Archives of Photogrammetry and Remote Sensing*. 1994. vol. 30(5), pp. 65–70.
 17. Zhang Z., Deriche R., Faugeras O., Luong Q. T. A Robust Technique for Matching Two Uncalibrated Images through the Recovery of the Unknown Epipolar Geometry. *Artificial Intelligence*. 1995. vol. 78(1–2), pp. 87–119. DOI: 10.1016/0004-3702(95)00022-4
 18. Arnfred J. T., Winkler S. A General Framework for Image Feature Matching Without Geometric Constraints. *Pattern Recognition Letters*. 2016. vol. 73, pp. 26–32. DOI: 10.1016/j.patrec.2015.12.017

19. Takimoto R.Y., De Castro Martins T., Takase F.K., De Sales Guerra Tsuzuki M. Epipolar Geometry Estimation, Metric Reconstruction and Error Analysis from Two Images. *IFAC Proceedings*. 2012. vol. 2012;14, pp. 1739–1744. DOI: 10.3182/20120523-3-RO-2023.00098
20. Fraser C. Advances in Close-Range Photogrammetry. *Photogrammetric Week 2015*. 2015. pp. 257–268.
21. V-STARS – Geodetic Systems, Inc. Available at: <https://www.geodetic.com/v-stars/> (accessed: 29.01.2017).
22. Agisoft PhotoScan. Available at: <http://www.agisoft.com/> (accessed: 29.01.2017).
23. Maas H.-G. Automatic DEM Generation by Multi-Image Feature Based Matching. *International Archives of Photogrammetry and Remote Sensing*. 1996. vol. 31, pp. 484–489.
24. Bhowmick B., Patra S., Chatterjee A., Madhav Govindu V., Banerjee S. Divide and Conquer: A Hierarchical Approach to Large-Scale Structure-from-Motion. *Computer Vision and Image Understanding*. 2017. vol. 157, pp. 190–205. DOI: 10.1016/j.cviu.2017.02.006
25. Forsyth D., Ponce J. *Computer Vision: A Modern Approach*. 2nd ed. Pearson, 2011.
26. Sukhovilov B.M., Sartasov E.M., Grigorova E. A. Improving the Accuracy of Determining the Position of the Code Marks in the Problems of Constructing Three-Dimensional Models of Object. *Proceedings of the 2nd International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM)*, Chelyabinsk, Russia, 2016. IEEE Xplore Digital Library. pp. 1–4. DOI: 10.1109/ICIEAM.2016.7911682
27. Bomze I., Budinich M., Pardalos P., Pelillo M. The Maximum Clique Problem. *Handbook of Combinatorial Optimization*, 1999. pp. 1–74.
28. Triggs B., McLauchlan P., Hartley R., Fitzgibbon A.. Bundle Adjustment — A Modern Synthesis. *Vision Algorithms: Theory & Practice*. 2000. Vol 1883. pp. 298–372. DOI: 10.1007/3-540-44480-7_21
29. Tushev S.A., Sukhovilov B.M.. Effective Graph-Based Point Matching Algorithms. *Proceedings of the 2nd International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM)*, Chelyabinsk, Russia, 2016. IEEE Xplore Digital Library. pp. 1–5. DOI: 10.1109/ICIEAM.2016.7911628
30. OpenCV (Open Source Computer Vision Library): Available at: <http://opencv.org/> (accessed: 29.11.2016).
31. Eigen: C++ Template Library For Linear Algebra: Available at: <http://eigen.tuxfamily.org/> (accessed: 29.11.2016).
32. Konc J., Janežič D. An Improved Branch and Bound Algorithm for the Maximum Clique Problem. *MATCH Communications in Mathematical and in Computer Chemistry*. 2007. vol. 58, pp. 569–590
33. Konc J., Janezic D. New C++ Source Code of the MaxCliqueDyn Algorithm: Available at: <http://insilab.org/maxclique/> (accessed: 29.11.2016).
34. Tomita E., Tanaka A., Takahashi H. The Worst-Case Time Complexity for Generating All Maximal Cliques and Computational Experiments. *Theoretical Computer Science*. 2006. vol. 363, no. 1, pp. 28–42. DOI: 10.1016/j.tcs.2006.06.015
35. Conte A. Review of the Bron-Kerbosch Algorithm and Variations. Available at: <http://www.dcs.gla.ac.uk/~pat/jchoco/cliقة/enumeration/report.pdf> (accessed: 29.11.2016)

36. Bron C., Kerbosch J. Algorithm 457: Finding All Cliques of an Undirected Graph. *Communications of the ACM*, 1973. vol. 16, no. 9, pp. 575–577. DOI: 10.1145/362342.362367
37. Ozaki K. smlly:find_cliques – a Pivoting Version of Bron-Kerbosch Algorithm. Available at: <https://gist.github.com/smlly/1516622> (accessed: 29.11.2016)
38. Boost C++ libraries.: Available at: <http://www.boost.org/> (accessed: 29.11.2016).

УДК 004.92, 004.021

DOI: 10.14529/cmse170204

ПАРАЛЛЕЛЬНЫЕ АЛГОРИТМЫ ДЛЯ ЭФФЕКТИВНОГО ПОИСКА СООТВЕТСТВУЮЩИХ ТОЧЕК В ЗАДАЧАХ КОМПЬЮТЕРНОГО ЗРЕНИЯ

© 2017 г. С.А. Тушев, Б.М. Суховилов

*Южно-Уральский государственный университет**(454080 Челябинск, пр. им. В.И. Ленина, д. 76),**E-mail: science@tushev.org, sukhovilovbm@susu.ru*

Поступила в редакцию: 01.05.2017

В настоящей статье предложены параллельные алгоритмы для поиска соответствующих точек в задачах компьютерного зрения. Разрабатываемая коллективом авторов фотограмметрическая система основана на использовании искусственных световозвращающих мишеней, идентичных по фотометрическим параметрам. В связи с этим традиционные методы поиска соответствий на основе вычисления дескрипторов (SIFT, SURF, и др.) неприменимы; фотограмметрическая система использует методы, основанные на эпиполярной геометрии. В настоящей статье предложены эффективные алгоритмы поиска соответствий между точками по всей совокупности снимков (в отличие от классических методов, использующих 2-4 снимка), основанные на графах. Приведено точное двухшаговое решение задачи через суперклику графа потенциальных соответствий; показана невозможность практического нахождения суперклики в реальных задачах в связи с вычислительной сложностью. Предложена эффективная параллельная реализация алгоритма формирования графа на основе эпиполярных ограничений, а также быстродействующий параллельный эвристический алгоритм поиска клик в данном графе. Применение итерационной схемы с обратным проецированием точек, отсевом выбросов и уравниванием координат точек и положений камер через метод связей позволяет в итоге получать точное решение задачи с использованием эвристического алгоритма поиска клик на каждой итерации. Предложенная архитектура системы дает значительный выигрыш во времени. Разработаны программные реализации описанных алгоритмов. Выполнена сравнительная оценка эффективности и производительности предложенных алгоритмов применительно к разрабатываемой фотограмметрической системе, экспериментально подтверждена эффективность предлагаемых решений.

Ключевые слова: компьютерное зрение, фотограмметрия, поиск соответствующих точек, параллельные алгоритмы, нахождение максимальной клики, эпиполярная геометрия

ОБРАЗЕЦ ЦИТИРОВАНИЯ

Tushev S.A., Sukhovilov B.M. Parallel Algorithms for Effective Correspondence Problem Solution in Computer Vision // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2017. Т. 6, № 2. С. 49–68. DOI: 10.14529/cmse170204.

Литература

1. Hartley R., Zisserman A. Multiple View Geometry in Computer Vision. 2nd ed. Cambridge University Press, 2004.
2. Baggio D. et al. Mastering OpenCV with Practical Computer Vision Projects. Packt Publishing, 2012.

3. Bay H., Tuytelaars T., van Gool L. SURF: Speeded Up Robust Features // Computer Vision and Image Understanding. 2008. Vol. 110, No. 3, P. 346–359. DOI: 10.1016/j.cviu.2007.09.014
4. Lowe D.G. Object recognition from local scale-invariant features // Proceedings of the Seventh IEEE International Conference on Computer Vision. 1999. Vol. 2, P. 1150–1157. DOI: 10.1109/ICCV.1999.790410
5. Fraser C.S. Innovations in automation for vision metrology systems // Photogrammetric Record. 1997. Vol. 15, No. 90, P. 901–911.
6. Суховилов Б. М., Григорова Е. А. Разработка фотограмметрической системы измерения пространственных координат элементов конструкций каркаса низкопольного трамвая // Наука ЮУрГУ [Электронный ресурс]. Материалы 67-й научной конференции. Секции экономики, управления и права. Челябинск: Издательский центр ЮУрГУ, 2015. С. 458–463.
7. Тушев С.А., Суховилов Б.М. Некоторые способы повышения производительности автоматической калибровки цифровых камер// Молодой исследователь: материалы 2-й научной выставки-конференции научно-технических и творческих работ студентов. Челябинск: Издательский центр ЮУрГУ, 2015. С. 434–439.
8. Hartmann W., Havlena M., Schindler K. Recent developments in large-scale tie-point matching // ISPRS Journal of Photogrammetry and Remote Sensing. International Society for Photogrammetry and Remote Sensing, Inc. (ISPRS), 2016. Vol. 115. P. 47–62.. DOI: 10.1016/j.isprsjprs.2015.09.005
9. Remondino F. et al. State of the art in high density image matching // The Photogrammetric Record. 2014. Vol. 29, No 146. P. 144–166. DOI: 10.1111/phor.12063
10. Leutenegger S., Chli M., Siegwart R.Y. BRISK: Binary Robust Invariant Scalable Keypoints. DOI: 10.1109/ICCV.2011.6126542
11. Ortiz R., Alahi A., Vanderghenst P. FREAK: Fast retina keypoint // Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2012. P. 510–517. DOI: 10.1109/CVPR.2012.6247715
12. Fraser C.S., Cronk S. A hybrid measurement approach for close-range photogrammetry // ISPRS Journal of Photogrammetry and Remote Sensing. Elsevier B.V., 2009. Vol. 64, No 3. P. 328–333.
13. Leung C., Lovell B.C. 3D Reconstruction through Segmentation of Multi-View Image Sequences // Proceedings of the 2003 APRS Workshop on Digital Image Computing. 2003. P. 87–92. URL: <http://espace.library.uq.edu.au/view/UQ:10960> (дата обращения: 29.11.2016)
14. Qiqiang F., Guangyun L. Matching of artificial target points based on space intersection // The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. 2008. Vol. XXXVII. P. 111–114.
15. Maas H.-G. Complexity analysis for the establishment of image correspondences of dense spatial target fields // International Archives of Photogrammetry and Remote Sensing. 1992. Vol. 29. P. 102–107.
16. Dold J., Maas, H.-G. An Application of Epipolar Line Intersection in a Hybrid Close-Range Photogrammetric System // International Archives of Photogrammetry and Remote Sensing. 1994. Vol. 30(5), P. 65–70.

17. Zhang Z. et al. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry // *Artificial Intelligence*. 1995. Vol. 78, No 1–2. P. 87–119. DOI: 10.1016/0004-3702(95)00022-4
18. Arnfred J.T., Winkler S. A general framework for image feature matching without geometric constraints // *Pattern Recognition Letters*. Elsevier B.V., 2016. Vol. 73. P. 26–32. DOI: 10.1016/j.patrec.2015.12.017
19. Takimoto R.Y. et al. Epipolar geometry estimation, metric reconstruction and error analysis from two images // *IFAC Proceedings Volumes (IFAC-PapersOnline)*. 2012. Vol. 14, No 1. P. 1739–1744. DOI: 10.3182/20120523-3-RO-2023.00098
20. Fraser C. *Advances in Close-Range Photogrammetry* // *Photogrammetric Week 2015*. 2015. P. 257–268.
21. V-STARS – Geodetic Systems, Inc. URL: <https://www.geodetic.com/v-stars/> (дата обращения: 29.01.2017).
22. Agisoft PhotoScan. URL: <http://www.agisoft.com/> (дата обращения: 29.01.2017).
23. Maas H.-G. Automatic DEM generation by multi-image feature based matching // *International Archives of Photogrammetry and Remote Sensing*. 1996. Vol. 31. P. 484–489.
24. Bhowmick B. et al. Divide and conquer: A hierarchical approach to large-scale structure-from-motion // *Computer Vision and Image Understanding*. Elsevier Inc., 2017. Vol. 157. P. 190–205. DOI: 10.1016/j.cviu.2017.02.006
25. Forsyth D., Ponce J. *Computer Vision: A Modern Approach*. 2nd ed. Pearson, 2011.
26. Суховилов Б.М., Сартасов Е.М., Григорова Е.А. Повышение точности определения положения кодовых марок в задачах построения трехмерных моделей объектов // *Пром-инжиниринг: труды II международной научно-технической конференции*. Челябинск: Издательский центр ЮУрГУ, 2016. С. 363–366.
27. Bomze I., Budinich M., Pardalos P., Pelillo M. The Maximum Clique Problem // *Handbook of Combinatorial Optimization*, 1999. P. 1–74.
28. Triggs B., McLauchlan P., Hartley R., Fitzgibbon A.. *Bundle Adjustment — A Modern Synthesis* // *Vision Algorithms: Theory & Practice*. 2000. Vol. 1883. P. 298–372. DOI: 10.1007/3-540-44480-7_21
29. Тушев С.А., Суховилов Б.М. Эффективные алгоритмы поиска соответствий точек на снимках на основе графов // *Пром-инжиниринг: труды II международной научно-технической конференции*. Челябинск: Издательский центр ЮУрГУ, 2016. С. 464–468.
30. OpenCV (Open Source Computer Vision Library): URL: <http://opencv.org/> (дата обращения: 29.11.2016).
31. Eigen: C++ template library for linear algebra: URL: <http://eigen.tuxfamily.org/> (дата обращения: 29.11.2016).
32. Konc J., Janežič D. An improved branch and bound algorithm for the maximum clique problem // *MATCH Communications in Mathematical and in Computer Chemistry*. 2007, P. 569–590
33. Konc J., Janezic D. New C++ source code of the MaxCliqueDyn algorithm: URL: <http://insilab.org/maxclique/> (дата обращения: 29.11.2016).
34. Tomita E., Tanaka A., Takahashi H. The worst-case time complexity for generating all maximal cliques and computational experiments // *Theoretical Computer Science*. 2006, Vol. 363, No. 1, P. 28–42. DOI: 10.1016/j.tcs.2006.06.015

35. Conte, A. Review of the Bron-Kerbosch algorithm and variations. URL: <http://www.dcs.gla.ac.uk/pat/jchoco/cliقة/enumeration/report.pdf> (дата обращения: 29.11.2016)
36. C. Bron, J. Kerbosch. Algorithm 457: finding all cliques of an undirected graph // Communications of the ACM. Sep 1973, Vol. 16, No. 9, P. 575–577. DOI: 10.1145/362342.362367
37. Ozaki K. smlы:find_cliques – a pivoting version of Bron-Kerbosch algorithm. URL: <https://gist.github.com/smlы/1516622> (дата обращения: 29.11.2016)
38. Boost C++ libraries.: URL: <http://www.boost.org/> (дата обращения: 29.11.2016).

Тушев Семен Александрович, аспирант, высшая школа экономики и управления, «Южно-Уральский государственный университет (национальный исследовательский университет)» (Челябинск, Российская Федерация)

Суховилов Борис Максимович, д.т.н, с.н.с., высшая школа экономики и управления, «Южно-Уральский государственный университет (национальный исследовательский университет)» (Челябинск, Российская Федерация)