An improved GMS-PROSAC algorithm for image mismatch elimination

Panpan Zhao\textsuperscript{a}, Derui Ding\textsuperscript{a}, Yongxiong Wang\textsuperscript{a} and Hongjian Liu\textsuperscript{b}

\textsuperscript{a}Department of Control Science and Engineering, University of Shanghai For Science and Technology, Shanghai, China; \textsuperscript{b}School of Mathematics and Physics, Anhui Polytechnic University, Wuhu, China

\begin{abstract}
Image matching usually plays a critical role for visual simultaneous localization and mapping (VSLAM). However, the resultant mismatches and low calculation efficiency of current matching algorithms reduces the performance of VSLAM. In order to overcome above two drawbacks, an improved GMS-PROSAC algorithm for image mismatch elimination is developed in this paper by introducing a new epipolar geometric constraint (EGC) model with a projection error function. This improved algorithm, named as GMS-EGC-PROSAC, is made up of the traditional GMS algorithm and the improved PROSAC algorithm. First, the GMS algorithm is employed to obtain a rough matching set and then all matching pairs in this set are sorted according to their similarity degree. By selecting some matching pairs with the highest similarity degree, the parameter of the EGC model is obtained. By resort to the calculated parameter, the improved PROSAC algorithm can be carried out to eliminate false matches. Finally, the real-time and the effectiveness are adequately verified by executing some contrast experiments. Our approach can not only quickly eliminate mismatches but also get more high-quality matching pairs.
\end{abstract}

\section{1. Introduction}

Real-time Visual Simultaneous Localization and Mapping (VSLAM) (Fraundorfer & Scaramuzza, 2011, 2012; Mur-Artal, Montiel, & Tardos, 2015; Mur-Artal & Tardos, 2017) and 3D reconstruction (Bourke, 2012; Izadi et al., 2011; Rebecq, Gallego, Mueggler, & Scaramuzza, 2017) have become a research hotspot in recent years. It is well known that image matching plays an important role in this area (Masoud & Hoff, 2016). Some essential requirements of image matching algorithm involve real-time, accuracy as well as robustness (Wang, Zhu, Qin, Jiang, & Li, 2018; Zheng, Hong, Zhang, Li, & Li, 2016). Up to now, there is still a wide performance gap between slow (but robust) feature matchers and the much faster (but often unstable) real-time solutions. Generally speaking, image matching algorithm usually consists of four main steps. First, we should pick up some `interest points’ at distinctive locations in an image, such as blobs, corners, and T-junctions. Then, we can obtain the descriptor of the neighbourhoods of every interest point by utilizing a feature vector (Nepomuceno, Martins, Amaral, & Riveret, 2017; Rajaratnam, Gomm, Yu, & Abdelhadi, 2017). Such a descriptor should be distinctive and, at the same time, robust to noise, detection errors, and geometric and photometric deformations. Next, these descriptor vectors must be matched between different images. The matching is often based on a distance between the vectors, e.g. the Mahanalobis or Euclidean distance (Bian et al., 2017; Jordan & Zell, 2016). Finally, mismatch elimination should be dealt with to obtained origin match by implementing above steps.

In recent years, the image matching algorithm with point feature has been widely deployed because of its simple extraction operation, flexible matching and less time. The typical point feature matching algorithms include Scale Invariant Feature Transform (SIFT) (Lowe, 2004) and Speeded Up Robust Features (SURF) (Bay, Tuytelaars, & Gool, 2006), to just name a few. SIFT algorithm (Lowe, 2004) has good adaptability to scale change, rotation and illumination change, but has large computation, long time consuming and no real-time performance. SURF (Bay et al., 2006) feature matching algorithm usually reduces the dimension of the feature descriptor with intent to improving the matching efficiency. Unfortunately, the real-time performance of above algorithm is far from the requirement of VSLAM for mobile robots. By adopting convolutional neural networks (Wang, Chen, Qiao, & Snoussi, 2018; Zabidi et al., 2017) to train the learned invariant feature transform (LIFT) descriptors, a new algorithm has been developed which not only adapts to real-time performance but also improves the matching efficiency.
developed in Morel and Yu (2009) to improve matching accuracy. But, it is still a time consumption one. Very recently, an oriented matching algorithm (called as ORB Rublee, Rabaud, Konolige, & Bradski, 2011) composing features from accelerated segment test (FAST) and rotated BRIEF feature (Rublee et al., 2011) discloses the rapidity in detecting feature points and matching while achieving the rough real-time performance for sparse feature point detection. For obtained match feature points in VSLAM of mobile robots, we found that there are a large number of false matches, which leads to the inaccuracy of the pose estimation and the poor robustness.

It is very crucial to eliminate false matches while retaining the matching quality in VSLAM. Typical approaches include both the Fast Library for Approximate Nearest Neighbours (FLANN) algorithm (Muja & Lowe, 2009) and Random Sample Consensus (RANSAC) (Fischler & Bolles, 1987) algorithm. FLANN is dependent on an artificial threshold value and RANSAC cannot accommodate the sheer number of false matches in the set of all nearest-neighbour matches. Recently, Bian et.al. has proposed a grid based motion statistics (GMS) algorithm (Bian et al., 2017) to distinguish some false and true matches. It should be pointed out that such an algorithm only get a rough match set by eliminating some false matches of two images about the same 3D scene. In other words, there still exist some false matches. As false matches do not satisfy epipolar geometry constraints (EGCs) (Hartley & Zisserman, 2003; Kushnir & Shimshoni, 2014), it is possible to develop a new PROSAC algorithm (Chum & Matas, 2005) by fusing an EGC model to further eliminate false matches. EGC, which reveals the relationship between two views, is a basic concept in image geometry. Specially, suppose a point \( P \) in the three-dimensional space can be projected into two image planes, and its projection is denoted as \( p \) and \( p' \) in the left and right images, respectively. This constraint shows that \( p' \) must be located in the \( p \) corresponding polar line. In the end, it can be utilized to get more desired quality matches for VSLAM.

Responding the above discussion, our paper is concerned with the design of novel mismatch elimination algorithm via uniting both GMS algorithm and PROSAC algorithm. We endeavour to deal with the following challenges: how can we introduce the concept of epipolar geometry constraint and then utilize it to further eliminate mismatches, and how can we design and carry out a series of experiments to verify the effectiveness of proposed algorithm. To this end, first, we use the ORB algorithm (Rublee et al., 2011) to describe feature points and get a rough matching set with the help of hamming distance method. Then, a new algorithm via the epipolar geometric constraint is designed to get a large number of high-quality matching pairs. In comparison with the traditional ORB algorithm (Rublee et al., 2011) combined with FLANN (Muja & Lowe, 2009) and RANSAC (Fischler & Bolles, 1987), the proposed algorithm should possess higher real-time performance, higher matching precision and more number of high-quality matches. Our main contributions can be highlighted as follows: (1) A new epipolar geometric constraint model, combining with a projection error function, is employed to further eliminate mismatches; (2) an improved GMS-PROSAC algorithm, named as GMS-EGC-PROSAC algorithm, is put forward to enhance the real-time performance and the matching precision, and to increase the number of high-quality matching; and (3) in comparison with the GMS, ORB, SIFT, SURF algorithms with a standard ratio-test, the effectiveness of proposed algorithm is adequately demonstrated by a series of experiments.

The rest sections are organized as follows. In Section 2, we introduce the ORB image matching algorithm including both the ORB feature detection and the ORB feature descriptor. In Section 3, we briefly summarize the GMS algorithm and then put forward to an improved PROSAC algorithm by adding an epipolar geometric constraint. Its flow chart and the algorithm steps are provided in details. In Section 4, we verified the validity of our algorithm through experiments. Finally, a conclusion is drawn in Section 5.

### 2. ORB Image Matching Algorithm

In this section, we introduce the ORB image match algorithm, which consists of oriented FAST and rotated BRIEF feature match algorithms.

#### 2.1. Oriented FAST

Oriented FAST is commonly obtained by adding a direction information to the well-known original FAST proposed by Rosten and Drummond (2006). It should be pointed out that FAST has neither the direction invariance nor the scale invariance. Firstly, an image pyramid (Klein & Murray, 2008) should be built for oriented FAST in order to achieve the scale invariance, and then the FAST corner is detected at each level of such a pyramid. Furthermore, the direction of feature points can be obtained by calculating the centre of mass of image blocks via the grey matter method. In what follows, specific steps of the oriented FAST are shown with the help of an example, see Figure 1. In this figure, the right sub-figure shows a disassociated Bresenham ring around the centre corner candidate \( p \), where the red squares are the pixels used in the FAST corner detection and the arc plotted by dashed line.
passes through 12 contiguous pixels which are brighter or darker than the candidate \( p \).

1. **The detection of FAST corners.** We start by detecting FAST points in the image. In the above template, we can obtain a circle of 16 pixels around the centre corner candidate \( p \) via drawing a half-diameter of three circumference. In order to determine whether \( p \) is a feature point or not, the following corner response function is adopted

\[
N = \sum_{q \in \text{circle}(p)} |G_q - G_p| > \theta_d
\]  

where \( G_q \) means the grey value of 16 pixels on the ring; \( \text{circle}(p) \) stands for a collection of 16 pixels on the Bresenham ring at the centre of \( p \); and \( N \) is the number of pixels, whose greyscale difference with the pixel point \( p \) is greater than a given threshold \( \theta_d \) (taking 20\% of \( G_p \) in our paper). When \( N \) consecutive pixels are satisfied, the candidate \( p \) is called a FAST feature point. Especially, \( N \) is set as 12 in our paper.

2. **Adding a direction of feature points.** The intensity centroid (Rosin, 1999) is introduced to determine the direction of the feature points. We usually assume that the intensity centroid is deviated from its geometry centre. Under the framework of the grey matter method, the moment \( M_{ij} \) of a patch \( B \) in Chum and Matas (2005) is defined as follows:

\[
M_{ij} = \sum_{(x,y) \in B} x^i y^j I(m), \quad i,j \in \{0,1\}
\]  

where \( m = (x, y) \) stands for a pixel point, that is, the coordinate of a pixel in patch \( B \), and \( I(m) \) is its intensity. According to the calculated moments, we can obtain the centroid \( c \) of the patch \( B \) by resort to

\[
c = \left( \begin{array}{cc} M_{10} & M_{01} \\ M_{00} & M_{00} \end{array} \right)
\]  

Denoting the point \( p \) as the origin point and connecting point \( \bar{p} \) with the obtained centroid \( c \), we can calculate the orientation of such a patch \( B \):

\[
\theta = \text{atan}2(M_{01}, M_{10})
\]  

where \( \text{atan}2 \) is the quadrant-aware version of arctan. Up to now, the feature points can be fixed an orientation so that we can describe the information around it via BRIEF. It should be pointed out that such a description possesses the rotation invariance.

### 2.2. Rotated BRIEF feature descriptor algorithm

Rotated BRIEF is a critical part of the ORB feature match algorithm by adding the rotation invariance based on the original BRIEF algorithm (Calonder, Lepetit, Strecha, & Fua, 2010). Now, let us introduce it from the original BRIEF algorithm. Firstly, by performing a set of binary intensity tests of an image patch (Zitnick, 2010), a bit string description can be obtained.

Define the binary intensity test \( \tau \) in a smooth image patch \( B \) as:

\[
\tau(p; m, s) = \begin{cases} 1, & I(m) < I(s) \\ 0, & I(m) \geq I(s) \end{cases}
\]  

where \( I(m) \) is the intensity of pixel \( p(m) \) (denoted as \( p \) for convenience) with the coordinate description \( m = (x, y) \). For various distribution of tests considered in Zitnick (2010), a Gaussian distribution around the centre of patch \( B \), one of the best performers, is exploited in our paper. In what follows, choosing a vector length \( n \), we carry out the \( \tau \) test for the \( n \) neighbour vectors for each feature point. First, the neighbour location is denoted as \( (x_i, y_i) \) \((i = 1, 2, \ldots, n)\). For the purpose of analysis, these points are then described as a matrix \( S \) (corresponding to a two-frame image) with dimension \( 2 \times n \)

\[
S = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \\ y_1 & y_2 & \cdots & y_n \end{bmatrix}
\]

Implement the oriented FAST to obtain the patch orientation \( \theta \) and then get the rotation matrix \( R_\theta \)

\[
R_\theta = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}
\]

Executing the rotation invariance to two-frame image \( S \) results in

\[
S_\theta = R_\theta S = \begin{bmatrix} x'_1 & x'_2 & \cdots & x'_n \\ y'_1 & y'_2 & \cdots & y'_n \end{bmatrix}
\]

Therefore, we have the following improved \( \tau \) test for the BRIEF algorithm with rotation invariance

\[
g_\theta(p, \theta) = \sum_{i=1}^{n} 2^{i-1} \tau(p; m', s'), \quad (m', s') \in S_\theta
\]
We need to point out that the rotated BRIEF descriptors can not only save the storage space and but also reduce the matching time. In addition, it is convenient to ORB feature matching with hamming distance.

3. Our approach: GMS-EGCPROSAC

As shown in Figure 2, using hamming distance to Brute-force (Lin et al., 2014) match will produce a large number of mismatches, which leads to a big obstacle for the subsequent visual localization and mapping work. In order to improve the accuracy of matching and obtain as many good matches as possible, we propose a union mismatching elimination algorithm, named as GMS-EGCPROSAC. Such an algorithm makes up of the grid based motion statistics (GMS) (Bian et al., 2017) and the improved PROSAC algorithms (Chum & Matas, 2005) with epipolar geometric constraints.

To be specific, a rough matching set is first generated via GMS algorithm, and then the false matches in such a matching set will be further eliminated by utilizing a new improved PROSAC algorithm, which embeds an epipolar geometric constraint model with a projection error fusion.

3.1. GMS algorithm

GMS algorithm is essentially a matching statistical constraint model. For a pair of images taken from different views of the same 3D scene, a feature correspondence implies that a pixel (i.e. feature point) in one image is identified as the same point in the other image. If targets in scene move, their neighboring pixels and features will move together. Motion smoothness guarantees that a small neighbourhood around a true match shows the same 3D location. Likewise, the neighbourhood around a false match views geometrically different 3D locations.

Consider an image pair \( \{B_a, B_b\} \) with \( M \) and \( N \) feature points for each image and assume \( M \geq N \). The set of all matching pairs can be obtained according to the hamming distance from \( B_a \) to \( B_b \). For the purpose of convenience, it is denoted as \( \chi = \{x_1, x_2, \ldots, x_N\} \) with \( x_i = \{p_i, p'_i\} \) where \( p_i \) and \( p'_i \) come from images \( B_a \) and \( B_b \), respectively. The goal of GMS algorithm is to divide \( \chi \) into two sets: the true and false sets of matching pairs by utilizing the local support of each matching. In addition, remove false matches to get a new matching set \( \bar{\chi} \). For this purpose, denote \( S_i \) as the matching statistical constraint scoring:

\[
S_i = (R(\chi_i)) - 1, \quad \chi_i \in \chi
\]

where \( \langle \cdot \rangle \) stands for the count operation on matching pairs, \( R(\chi_i) \) stands for the neighbourhood around the pair \( \chi_i \), and \( S_i \) is the number of other matching pairs in neighbourhood \( R(\chi_i) \), which has removed the matching itself via subtracting ‘1’. We can judge if a matching pair is correct by comparing \( S_i \) with the threshold \( \beta \) obtained by experiments.

In order to enhance the robustness of this calculation model, an \( N \times N \) neighbourhood (see Figure 3 with \( N = 9 \) for an example) is selected to calculate the score of each small neighbour area. In this case, the sum of scores is calculated via (10):

\[
S_{ab} = \sum_{k=1}^{N} (R(\chi_{kj})) - 1, \quad \chi_{kj} \in \text{Area}(a^k, b^k) \quad \text{(11)}
\]

Additionally, selecting an \( N \times N \) neighbourhood in experiments as a score template, we can get the threshold \( \beta = \alpha \sqrt{\delta} \), where \( \alpha \) is a suitable parameter (\( \alpha = 6 \) for Figure 3) and \( \delta \) is the average number of feature points present in per single grid-cell in Figure 3. In addition, executing the same operation for all matching pairs in \( \chi \), and removing all false matching pairs for \( S_{ab} \geq \beta \) to produce a new one \( \bar{\chi} \).

3.2. Improved PROSAC algorithm

In light of the traditional RANSAC algorithm (Fischler & Bolles, 1987), the main idea of PROSAC algorithm (Chum & Matas, 2005) is to (1) sort the feature matching point

![Figure 2. ORB initial matching figure. Each horizontal colour line in the figure represents a correct match and the slash means false match.](image)

![Figure 3. The score template with 9 neighbors in GMS algorithm.](image)
pairs according to their similarity degree and then produce a suitable set of interior points; (2) to obtain the corresponding parameters in algorithm template with the help of parts of interior points; and (3) to execute template for all matching point pairs. Obviously, the correct set of interior points plays an important role in VSLAM and the mismatch pairs inevitably deteriorate the matching performance. In what follows, we will provide an improved PROSAC algorithm (Chum & Matas, 2005) by fusing a novel epipolar geometric constraint (EGC) model to achieve more precise elimination of mismatches.

To be specific, in the first step, we apply the GMS algorithm to obtain a rough matching point set, and then calculate and sort the similarity degree of each matching pair in this set. In the second step, we use \( N \) matching pairs with the highest similarity degree to calculate the epipolar geometric constraint model, and then employ the obtained model to estimate remaining matching pairs in the matching point set. There are two tasks: (1) eliminating the outer points that do not meet the geometric constraint model and (2) obtaining a large number of good matching pairs via updating model with an iteration form.

The EGC model is essentially the geometric projection relations of two views, or the geometry of the intersection of the image plane and the plane beam, whose axis is the baseline (the line joining the camera centres). Suppose that a point \( P \) in the three-dimensional space can be projected in two image planes: \( p_1 \) in the left image and \( p_2 \) in the right image. The epipolar geometry constraint shows that, for the projection point \( p_1 \), if the projection point \( p_2 \) lies in the corresponding polar line, we can reduce the number of match pairs, increase the matching accuracy, improve the speed of delete mismatches, and strengthen the real-time performance of the algorithm. The mathematical equation for polar geometric constraints can be expressed as:

\[
Q : \begin{bmatrix} x_1 K^{-1} p_1 \end{bmatrix} = 0
\]

where \( E \) is the essential matrix and \( K \) is the camera internal reference obtained by calibration. \( x_1 = [u_1, v_1, 1]^T \) and \( x_2 = [u_2, v_2, 1]^T \) are normalized coordinates of points \( p_1 \) and \( p_2 \), respectively. Because the type under different scales is equivalent, one has

\[
(u_1 \quad v_1 \quad 1) \begin{bmatrix} e_1 & e_2 & e_3 \quad e_4 \quad e_5 \quad e_6 \quad e_7 \quad e_8 \quad e_9 \end{bmatrix} \begin{bmatrix} u_1 \\ v_1 \\ 1 \end{bmatrix} = 0
\]

In addition, stretching \( E \) as a vector \( \tilde{e} \) leads to

\[
\tilde{e} = (e_1 \quad e_2 \quad e_3 \quad e_4 \quad e_5 \quad e_6 \quad e_7 \quad e_8 \quad e_9)^T
\]

Then, with the help of this vector, the epipolar geometry constraint (12) can be written as a linear form relating to \( \tilde{e} \):

\[
[u_1 u_2 \quad u_1 v_2 \quad v_1 u_2 \quad v_1 v_2 \quad u_1 \quad u_2 \quad v_1 \quad v_2 \quad 1] \tilde{e} = 0
\]

Obviously, we can employ eight matching pairs to calculate the essential matrix \( E \): \( E \in \mathbb{R}^{3 \times 3} \), and we can evaluate other matching pairs in matching set \( \tilde{\chi} \) to get high quality matching pairs. Specially, by resort to the essential matrix \( E \), arbitrary feature points \( p_i \) in the previous frame image can be mapped in one point \( p'_i \) in its corresponding polar line in the next frame. Define the projection error function

\[
d_s(E, p_i, p'_i) = p_i - E p'_i
\]

If \( d_s \) is less than a threshold \( d_T \) predetermined by experiments (\( d_T = 5 \) in our paper), it can be regarded as an expected high quality matching pair, all of which consist of the support match set \( T \). The specific steps of the improved PROSAC algorithm are as follows.

1. Obtain a matching point set \( \tilde{\chi} \) by GMS algorithm, and then sort the matching point pair according to their similar distance.
2. Give the threshold \( d_T \) (\( d_T = 5 \) in our paper). Select 8 matching pairs with the highest similar degree in the matching set \( \tilde{\chi} \), and calculate the essential matrix \( E \) to get the epipolar geometric constraint model \( Q \).
3. According to the obtained essential matrix \( E \), calculate the projection error function via (15) for all data in matching set \( \tilde{\chi} \). When the error \( d_s < d_T \), it is a high quality matching pair and then achieve the support match set \( T_k \).
4. If the element of the current support match set \( T_k \) is greater than the last one, then compute the essential matrix \( E \) like step (2), and update the support match set \( T_k \) and the iteration number \( k \).
5. If \( k \) is greater than \( k_{\text{max}} \), \( k_{\text{max}} \) can be obtained by (16), then exit and can obtain the best inliers set \( T^* \). Specially, \( k_{\text{max}} \) can be calculated by

\[
k_{\text{max}} = \frac{\log(1 - t)}{\log(1 - w^D)}
\]

where \( t \) is a confidence coefficient (usually taking 0.95), \( D = 8 \) is the minimum number of samples, and

\[
(u_1 \quad u_2 \quad u_1 v_2 \quad v_1 u_2 \quad v_1 v_2 \quad u_1 \quad u_2 \quad v_1 \quad v_2 \quad 1) \tilde{e} = 0
\]

 Obviously, we can employ eight matching pairs to calculate the essential matrix:

\[
\begin{bmatrix}
 u_1^1 u_2^1 & u_1^1 v_2^1 & u_1^2 u_2^2 & u_1^2 v_2^2 & u_1^3 u_2^3 & u_1^3 v_2^3 & u_1^4 u_2^4 & u_1^4 v_2^4 & u_1^5 u_2^5 & u_1^5 v_2^5 \\
 u_2^1 u_2^1 & u_2^1 v_2^1 & u_2^2 u_2^2 & u_2^2 v_2^2 & u_2^3 u_2^3 & u_2^3 v_2^3 & u_2^4 u_2^4 & u_2^4 v_2^4 & u_2^5 u_2^5 & u_2^5 v_2^5 \\
 \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 u_8^1 u_2^8 & u_8^1 v_2^8 & u_8^2 u_2^8 & u_8^2 v_2^8 & u_8^3 u_2^8 & u_8^3 v_2^8 & u_8^4 u_2^8 & u_8^4 v_2^8 & u_8^5 u_2^8 & u_8^5 v_2^8 \\
\end{bmatrix}
\times \begin{bmatrix}
 e_1^T \\
 e_2^T \\
 \vdots \\
 e_8^T \\
\end{bmatrix} = 0
\]

With the calculated matrix \( E \), we can evaluate other matching pairs in matching set \( \tilde{\chi} \) to get high quality matching pairs. Specially, by resort to the essence matrix \( E \), arbitrary feature points \( p_i \) in the previous frame image can be mapped in one point \( p'_i \) in its corresponding polar line in the next frame. Define the projection error function

\[
d_s(E, p_i, p'_i) = p_i - E p'_i
\]

If \( d_s \) is less than a threshold \( d_T \) predetermined by experiments (\( d_T = 5 \) in our paper), it can be regarded as an expected high quality matching pair, all of which consist of the support match set \( T \). The specific steps of the improved PROSAC algorithm are as follows.

1. Obtain a matching point set \( \tilde{\chi} \) by GMS algorithm, and then sort the matching point pair according to their similar distance.
2. Give the threshold \( d_T \) (\( d_T = 5 \) in our paper). Select 8 matching pairs with the highest similar degree in the matching set \( \tilde{\chi} \), and calculate the essential matrix \( E \) to get the epipolar geometric constraint model \( Q \).
3. According to the obtained essential matrix \( E \), calculate the projection error function via (15) for all data in matching set \( \tilde{\chi} \). When the error \( d_s < d_T \), it is a high quality matching pair and then achieve the support match set \( T_k \).
4. If the element of the current support match set \( T_k \) is greater than the last one, then compute the essential matrix \( E \) like step (2), and update the support match set \( T_k \) and the iteration number \( k \).
5. If \( k \) is greater than \( k_{\text{max}} \), \( k_{\text{max}} \) can be obtained by (16), then exit and can obtain the best inliers set \( T^* \). Specially, \( k_{\text{max}} \) can be calculated by

\[
k_{\text{max}} = \frac{\log(1 - t)}{\log(1 - w^D)}
\]

where \( t \) is a confidence coefficient (usually taking 0.95), \( D = 8 \) is the minimum number of samples, and

\[
(u_1 \quad u_2 \quad u_1 v_2 \quad v_1 u_2 \quad v_1 v_2 \quad u_1 \quad u_2 \quad v_1 \quad v_2 \quad 1) \tilde{e} = 0
\]
is the ratio of the inliers to the number of $\bar{\chi}$ in the initial state, it can be defined as: $w = D / V$, $V$ is the number of the matching set $\bar{\chi}$.

By calculating the geometrical constraint model, we can eliminate some points that do not satisfy the epipolar geometric constraint model and get better matching point pairs. Based on GMS algorithm and the improved ROSAC algorithm, we provide the following GMS-EGCPROSAC mismatching elimination algorithm.

### 3.3. GMS-EGCPROSAC mismatches elimination algorithm

In the above subsection, we develop an improved PROSAC algorithm, named as EGCPROSAC algorithm, in order to eliminate mismatches in a GMS matching set via embedding a novel epipolar geometric constraint model into the traditional PROSAC algorithm. In addition, fusing the implementation foundation based on GMS matching set, the whole union algorithm can be called as GMS-EGCPROSAC mismatches elimination algorithm, whose implementation framework is shown in Figure 4 and the flow is provided in ‘Algorithm 1: GMS-EGCPROSAC Algorithm’.

![Figure 4. Implementation framework of GMS-EGCPROSAC algorithm.](image-url)

#### Algorithm 1 GMS-EGCPROSAC Algorithm

**Input:** One pair of matched images \{Ba, Bb\} and the projection error threshold.

1. initialize $k = 0$, $T^* = 0$
2. Calculate the feature points $p_1, p'_1$ from $B_a$ and $B_b$ via (1).
3. For obtained $p_1, p'_1$, calculate their descriptors $d_1$ and $d_2$ with the help of (9).
4. The resultant match with hamming distance obtains a rough match set $\chi = \{X_1, X_2, \ldots, X_N\}$.
5. Divide images $B_a$ and $B_b$ by $3 \times 3$ grids, and calculate the threshold $\beta$ and $\delta$ in the score template.
6. Calculate the matching score $S_{ab}$ via (11) for each $3 \times 3$ grid.
7. if $S_i \geq \beta$ then
8. obtain a GMS matching set $\bar{\chi} = \{X_1, X_2, \ldots, X_M\}$
9. end if
10. select $D$ matching pairs with the highest similar degree in $\bar{\chi}$
11. calculate $k_{max}$ with via (16) using $w = D / V$
12. calculate the essential matrix $E$ via (12);
13. while $k < k_{max}$ do
14. calculate projection error $d_s$ via (15) of the rest matching pairs in $\bar{\chi}$
15. if $d_s < d_T$ then
16. obtain the support matching set $T_k$
17. if $|T_k| > |T^*|$ then
18. $T^* = T_k$
19. Update the essential matrix $E$ in $T^*$ via (12);
20. Update $k_{max}$ via (16) using $w = |T^*| / V$
21. end if
22. end if
23. $k = k + 1$
24. end while

**Output:** best inliers set $T^*$

### 4. Experiments

In this section, we evaluate the proposed approach based on two popular data sets: Cabinet (Strecha, Hansen, Gool, Fua, & Thoennessen, 2008) and TUM (Sturm, Engelhard, Endres, Burgard, & Cremers, 2012). The picture of two data sets is collected by the visual sensor along the real track. TUM includes six video sequences with the characteristics of wide-baselines, low-texture as well as blur. Cabinet, a subset of TUM, can permit the separate analysis in low-texture scenes. In our experiments, the computer is with an Intel Core i7-5557U and 16 Gib RAM, and the analysis is executed in comparison with GMS algorithm (Bian et al., 2017) and other state-of-the-art image match algorithm like traditional ORB (Rublee et al., 2011), SURF (Lowe, 2004) and SIFT (Bay et al., 2006).
4.1. The comparison between our approach and GMS algorithm

Figure 5 plots the picture of ‘rgbd_dataset_freiburg2_flowerbouquet_brownbackground’ in TUM (Sturm et al., 2012). There is a certain angle between these two images and the picture size is 640 × 480 in our experiment. The matching effect is clearly plotted via the colour lines in Figure 5 and the match results of these two algorithms are shown in Table 1. Specially, the initial matching pairs are obtained directly via the ORB algorithm (Rublee et al., 2011) and then mismatch pairs are preliminarily removed via FLANN for GMS and RANSAC for our scheme. In addition, both the GMS algorithm (Bian et al., 2017) and our improved algorithm are, respectively, executed to catch much more suitable matching pairs. We can find from Table 1 that the proposed algorithm can effectively increase the matching accuracy at the expense of a little longer time compared with the result via the GMS algorithm. Our algorithm can eliminate some mismatches which do not conform to the polar geometric constraint model, and thereby the matching precision can be increased and the quality of matching can be improved.

4.2. Our approach compare to some traditional picture matching algorithm

In the following experiment, the effectiveness of our algorithm is shown from the number of high-quality matching pairs, the matching accuracy as well as the running time in comparison with traditional algorithms. When guaranteeing the same matching accuracy, our method can get more high quality matching pairs.

4.2.1. Matching quantity and matching precision

The five group images from the TUM and Cabinet data sets are arranged according to their matching complexity. Firstly, applying these two algorithms, 500 initial matching pairs produced directly by the ORB algorithm is plotted in Figure 6(a). Obviously, there exist a lot of false matching pairs. In what follows, the resultant eliminating false ones are, respectively, depicted in Figures 6(b) and 6(c) by adopting the traditional ORB with FLANN-RANSAC and our algorithm, where colour horizontal colour lines stand for matching pairs with high-quality. The number of obtained good matching pairs and the matching accuracy are listed in Table 2, where the matching accuracy here is the ratio of the number of rough ones and good ones. We can observe from the four, eight and ten columns that the number of high-quality matching pairs obtained by our algorithm is more than that via the traditional ORB, whose all matching pairs are contained in our results. In addition, the number of obtained high quality matching pairs achieves 1 to 2 times more than the ones by traditional algorithms, and thereby the accuracy of VSLAM can be effectively improved. Additionally, the accuracy of our algorithm is slightly higher than the traditional algorithm at the same time.

4.2.2. Matching time

Image match algorithm requires not only the matching accuracy but also the matching real-time. In this experiment, the running times of our algorithm and several classical matching algorithms are recorded in Table 3. From this table, we can easily find that the running speed of our algorithm is about 0.5 times faster than that of traditional ORB (Rublee et al., 2011), 30 times faster than that of SURF (Lowe, 2004), and even 40 times faster than one of SIFT (Bay et al., 2006). Through the above comparison,
Figure 6. The initial match figure and mismatches elimination figure of different algorithms. Each horizontal colour line in the figure represents a correct match, and the slash represents false match. (a) Initial matching diagram, (b) Traditional ORB algorithm and (c) Our algorithm.

Table 2. Comparison results of matching quality of our algorithm and the traditional ORB algorithm.

| Image group | Initial matches | Rough matches by using ORB and FLANN | Good matches | Matching accuracy (%) | Initial matches | Rough matches by using ORB and GMS | Good matches | Matching accuracy (%) | Containing the number of matches |
|-------------|-----------------|--------------------------------------|--------------|-----------------------|-----------------|-------------------------------------|--------------|-----------------------|----------------------------------|
| 1           | 500             | 79                                   | 69           | 87.3                  | 500             | 206                                 | 180          | 87.4                  | 69                               |
| 2           | 500             | 241                                  | 182          | 75.6                  | 500             | 325                                 | 248          | 76.3                  | 182                              |
| 3           | 500             | 324                                  | 307          | 94.7                  | 500             | 469                                 | 445          | 94.9                  | 307                              |
| 4           | 500             | 337                                  | 311          | 92.3                  | 500             | 456                                 | 422          | 92.5                  | 311                              |
| 5           | 500             | 368                                  | 330          | 89.6                  | 500             | 417                                 | 376          | 90.2                  | 330                              |

it is verified that the real-time performance of our developed algorithm is greatly improved for image matching, which is beneficial to the VSLAM work in future.

In summary, the above experimental results show that the real-time of the improved algorithm is obviously better than the traditional ORB algorithm, even an order of magnitude faster than the classical SIFT and SURF image matching algorithms. The matching accuracy and the number of obtained high quality matching pairs are better than the ORB algorithm. Therefore, the effectiveness is adequately verified for the developed algorithm in our paper.
Table 3. Comparison results of the matching time of several different algorithms.

| Image group | Our approach | Traditional ORB algorithm | SURF | SIFT |
|-------------|--------------|---------------------------|------|------|
| 1           | 58.8         | 70.4                      | 3525.5 | 4827.7 |
| 2           | 52.5         | 86.5                      | 2768.3 | 4535.1 |
| 3           | 51.4         | 87.7                      | 3847.9 | 4970.6 |
| 4           | 47.5         | 75.4                      | 2203.5 | 3453.3 |
| 5           | 48.9         | 86.8                      | 3193.5 | 4050.0 |
| Average value | 51.82       | 81.36                     | 3107.7 | 4367.3 |

5. Conclusions

In this paper, we have developed an improved algorithm of image mismatches elimination by introducing a new epipolar geometric constraint (EGC) model. Such an algorithm adequately absorbs the advantages of both the traditional GMS algorithm and the PROSAC algorithm. By resort to the calculated epipolar geometric constraint model, our algorithm has been carried out to eliminate false matches while guaranteeing the matching accuracy and the number of good matching pairs. In addition, the real-time and effectiveness have been adequately checked by comparing with other algorithms in different scenarios. Future topics would include the camera pose in visual odometry in light of our algorithm developed in this paper.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was supported in part by the National Natural Science Foundation of China under Grants 61503001, the Shanghai Rising-Star Program of China under Grant 16QA1403000, the Natural Science Foundation of Shanghai under Grant 18ZR1427000, and the Program for Capability Construction of Shanghai Provincial Universities under Grant 15550502500.

References

Bay, H., Tuytelaars, T., & Gool, L. V. (2006). Surf: Speeded up robust features. In Proceedings of the 2006 European conference on computer vision (pp. 404–417). Berlin, Heidelberg: Springer.

Bian, J. W., Lin, W. Y., Matsushita, Y., Yeung, S.-K., Nguyen, T.-D., & Cheng, M.-M. (2017, July). Gms: Grid-based motion statistics for fast, ultra-robust feature correspondence. Proceedings of the 2017 IEEE conference on computer vision and pattern recognition (pp. 2828–2837), Honolulu, HI.

Bourke, R. (2012). Automatic 3D reconstruction: An exploration of the state of the art. GSTF Journal on Computing, 2(3), 71–75.

Calonder, M., Lepetit, V., Strecha, C., & Fua, P. (2010). Brief: Binary robust independent elementary features. Proceedings of the 2010 European conference on computer vision (pp. 778–792), Heraklion, Crete, Greece.

Chum, O., & Matas, J. (2005). Matching with PROSAC-progressive sample consensus. Proceedings of the 2005 IEEE computer society conference on computer vision and pattern recognition (Vol. 1, pp. 220–226), San Diego, CA.

Fischler, M. A., & Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM, 24(6), 381–395.

Fraundorfer, F., & Scaramuzza, D. (2011). Visual odometry. Part I: The first 30 years and fundamentals. IEEE Robotics and Automation Magazine, 18(4), 80–92.

Fraundorfer, F., & Scaramuzza, D. (2012). Visual odometry. Part II: Matching, robustness, optimization, and applications. IEEE Robotics and Automation Magazine, 19(2), 78–90.

Hartley, R., & Zisserman, A. (2003). Multiple view geometry in computer vision. New York, USA: Cambridge University Press.

Izadi, S., Kim, D., Hilliges, O., Molyneaux, D., Newcombe, R., Kohli, P., . . . Fitzgibbon, A. (2011). KinectFusion: Real-time 3D reconstruction and interaction using a moving depth camera. Proceedings of the 24th annual ACM symposium on user interface software and technology (pp. 559–568), Santa Barbara, CA.

Jordan, J., & Zell, A. (2016). Ground plane based visual odometry for RGBD-Cameras using orthogonal projection. IFAC-PapersOnLine, 49(15), 108–113.

Klein, G., & Murray, D. (2008). Improving the agility of keyframe-based SLAM. In Proceedings of the 2008 European conference on computer vision (pp. 802–815). Berlin, Heidelberg: Springer.

Kushnir, M., & Shimshoni, I. (2014). Epipolar geometry estimation for urban scenes with repetitive structures. IEEE Transactions on Pattern Analysis and Machine Intelligence, 36(12), 2381–2395.

Lin, W.-Y. D., Cheng, M.-M., Liu, J., Yang, H., Do, M. N., & Torr, P. (2014). Bilateral functions for global motion modeling. In Proceedings of the 2014 European conference on computer vision (pp. 341–356). Cham: Springer.

Lowe, D. G. (2004, November). Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, 60(2), 91–110.

Masoud, A., & Hoff, W. (2016). Segmentation and tracking of nonplanar templates to improve VSLAM. Robotics and Autonomous Systems, 86, 29–56.

Morel, J.-M., & Yu, G. (2009, April). ASIFT: A new framework for fully affine invariant image comparison. SIAM Journal on Imaging Sciences, 2(2), 438–469.

Muja, M., & Lowe, D. G. (2009, February). Fast approximate nearest neighbors with automatic algorithm configuration. Proceedings of the fourth international conference on computer vision theory and applications (pp. 331–340), Lisboa, Portugal.

Mur-Artal, R., Montiel, J. M. M., & Tardos, J. D. (2015). ORB-SLAM: A versatile and accurate monocular SLAM system. IEEE Transactions on Robotics, 31(5), 1147–1163.

Mur-Artal, R., & Tardos, J. D. (2017). Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. IEEE Transactions on Robotics, 33(5), 1255–1262.

Nepomuceno, E. G., Martins, S. A. M., Amaral, G. F. V., & Riveret, R. (2017). On the lower bound error for discrete maps using associative property. Systems Science and Control Engineering: An Open Access Journal, 5(1), 462–473.
Rajarathinam, K., Gomm, J. B., Yu, D., & Abdelhadi, A. S. (2017). An improved search space resizing method for model identification by standard genetic algorithm. *Systems Science and Control Engineering: An Open Access Journal, 5*(1), 117–128.

Rebecq, H., Gallego, G., Mueggler, E., & Scaramuzza, D. (2017, November). EMVS: Event-based multi-view stereo-3D reconstruction with an event camera in real-time. *International Journal of Computer Vision, 1*–21.

Rosin, P. L. (1999, February). Measuring corner properties. *Computer Vision and Image Understanding, 73*(2), 291–307.

Rosten, E., & Drummond, T. (2006). Machine learning for high-speed corner detection. *Proceedings of the 2006 European conference on computer vision* (pp. 430–443), Graz, Austria.

Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011). ORB: An efficient alternative to SIFT or SURF. *Proceedings of the 2011 international conference on computer vision* (pp. 2564–2571), Barcelona, Spain.

Strecha, C., Hansen, W. V., Gool, L. V., Fua, P., & Thoennessen, U. (2008). On benchmarking camera calibration and multi-view stereo for high resolution imagery. *Proceedings of the 2008 IEEE conference on computer vision and pattern recognition* (pp. 1–8), Anchorage, AK.

Sturm, J., Engelhard, N., Endres, F., Burgard, W., & Cremers, D. (2012). A benchmark for the evaluation of RGB-D SLAM systems. *Proceedings of the 2012 IEEE/RSJ international conference on intelligent robots and systems* (pp. 573–580), Vilamoura, Portugal.

Wang, T., Chen, Y., Qiao, M., & Snoussi, H. (2018, February). A fast and robust convolutional neural network-based defect detection model in product quality control. *The International Journal of Advanced Manufacturing Technology, 94*(9–12), 3465–3471.

Wang, K., Zhu, T., Qin, Y., Jiang, R., & Li, Y. (2018). Matching error of the iterative closest contour point algorithm for terrain-aided navigation. *Aerospace Science and Technology, 73*, 210–222.

Zabidi, A., Yassin, I. M., Hassan, H. A., Ismail, N., Hamzah, M. M. A. M., Z. I. Rizman, & Abidin, H. Z. (2017). Detection of asphyxia in infants using deep learning convolutional neural network (CNN) trained on Mel frequency cepstrum coefficient (MFCC) features extracted from cry sounds. *Journal of Fundamental and Applied Sciences, 9*(3S), 768–778.

Zheng, S., Hong, J., Zhang, K., Li, B., & Li, X. (2016). A multi-frame graph matching algorithm for low-bandwidth RGB-D SLAM. *Computer-Aided Design, 78*, 107–117.

Zitnick, C. L. (2010). Binary coherent edge descriptors. In *Proceedings of the 2010 European conference on computer vision* (pp. 170–182), Berlin, Heidelberg: Springer.