Eligible Features Segregation for Real-time Visual Odometry

Hongmou Zhang, Jürgen Wohlfeil, Denis Grießbach, Anko Börner

German Aerospace Center
Rutherfordstr. 2, 12489 Berlin, Germany
Email: (Hongmou.Zhang, Juergen.Wohlfeil, Denis.Griessbach
Anko.Boerner)@dlr.de

Abstract. Stereo vision aided inertial navigation enables accurate self-localization and navigation in unknown environment without the need of positioning systems (e.g. GNSS). The feature extractor is an elementary part of this technology because it extracts the landmarks from the camera images, which are then used for optical navigation. This is why the feature extractor is usually the first module of data processing chain. In most of the cases, the features from the feature extractor are filtered by a non-maximum suppression algorithm. An ideal non-maximum suppression algorithm suppresses "weak" features while keeping "strong" and well-distributed features. Only if the feature extractor is combined with an appropriate non-maximum suppression module, the computer vision system can get reasonably good results. In this paper, we propose a novel non-maximum suppression algorithm. The algorithm does not only provide well-distributed features over the whole image but is also be able to control the maximum number of required features in output, which is very important for real-time system. We apply our framework to the AGAST feature extraction algorithm, and it is very easy to incorporate with other feature extractors. Finally, we combine our algorithm with the *Integrated Positioning System* (IPS) which is developed by the German Aerospace Center (DLR). The comparison of testing results is illustrated.

1 Introduction

In a visual-odometry processing chain, usually the feature extractor is the first module. A reliable and efficient feature extractor is a crucial component for various computer vision applications, such as object tracking, image matching and registration, optical navigation and localization, and so forth. An ideal feature extractor should provide sufficiently strong enough features which are easy to match or to track; on the other hand, the run time of the feature extractor should be as minimal as possible.

During the past decades, many feature extractors have been proposed [1, 2, 3, 4, 5, 6]. However, there are merely few researches that focus on the non-maximum suppression algorithm. In most of these feature extraction algorithms, the extracted

features are merely filtered in its 3×3 neighbor area. This simple feature suppression scheme cannot get reasonable outputs in most of the case. An ideal non-maximum suppression algorithm suppresses "weak" features while keeping "strong" and well-distributed features. Only if the feature extractor is combined with an appropriate non-maximum suppression module, the computer vision system can get reasonably good results.

Based on the above background, in this paper, we focus on the non-maximum suppression algorithm. A novel non-maximum suppression framework is proposed. The algorithm does not only provide well-distributed features over the whole image but is also be able to control the maximum number of required features in output, which is very important for real-time systems. In the experiments, the proposed framework is applied to the *adaptive and generic accelerated segment test* (AGAST) feature extraction algorithms, see [5], and be combined with the *Integrated Positioning System* (IPS) which is developed by the German Aerospace Center (DLR). The experimental results show that the quality of the IPS measurement is significantly improved using the proposed method.

This paper is organized as follows: In Section 2, a review of feature extraction algorithms is given. In Section 3, the details of the proposed non-maximum suppression algorithm is described. Experimental results are presented in Section 4, and Section 5 concludes the paper.

2 Feature Extractor Review

For measuring the geometrical relationship between two camera poses, visual features are required. Over the past decades, many feature extraction algorithms have been proposed.

A well-known corner extraction algorithm is the *Harris algorithm* [1]. The authors describe an autocorrelation method for corner extraction. Considering a block in the image, the *sum of squared differences* (SSD) of pixel intensities can be determined by a small shift of the block in different directions. The first-order Taylor expansion of the SSD cost function leads to the *Harris matrix* which is an approximation of the *Hessian matrix*.

The two eigenvalues of this Harris matrix can be used to indicate whether the block covers a corner or not. If two eigenvalues are both nearly zero, the block is above an *homogenous area* (i.e. with similar intensities); otherwise the block is above an edge or corner. If one eigenvalue is significantly larger than the other, the block is above a corner.

There are several extraction algorithms aiming at an analysis of the Harris matrix. The *Kanade-Lucas-Tomasi* (KLT) [2] feature extractor is one of the famous. The KLT feature extraction algorithm is derived based on the assumption that the intensity of the same object point remains constant in subsequent frames which are grabbed at a short time difference in between. Such assumption is also known as *intensity constancy assumption* (ICA), see [7]. Base on the assumption, a cost function can

be derived. A Harris matrix is obtained by expanding the cost function of its first-order *Taylor series*. A KLT feature can be obtained by checking both eigenvalues of the Harris matrix [2].

The *smallest-univalue-segment-assimilating-nucleus* (SUSAN) method is proposed in [8]. The algorithm defines a circular mask where the center pixel is called the *nucleus*. By sliding the mask over the entire image, the intensity of pixels inside the mask is compared with the nucleus. A *structure* is composed of pixels which have a similar intensity as the nucleus; this structure is called *univalue segment assimilating nucleus* (USAN). It is obvious that a large USAN area describes a homogeneous area. When the mask is located over an edge, the USAN area is about half of the mask size. A corner is indicated by a USAN with a quarter of the mask size. The smaller the USAN area, the higher the probability that the nucleus is a corner.

The *accelerated-segment-test* (AST) method is derived from SUSAN. Instead of checking every pixel on the circular mask, the AST just evaluates pixels located on the circle which is known as *Bresenham circle* [9].

The *features-from-accelerated-segment-test* (FAST) algorithm [10] was the first feature extractor based on the AST method. In FAST, the Bresenham circle of radius 3 is used; this forms a circle which is composed of 16 pixels. The FAST algorithm compares each pixel's intensity on the circle with the center pixel P. If there exist more than S connected pixels on the circle with intensities greater than P's intensity plus a threshold T, or all of them less than P's intensity minus a threshold T, the center pixel is considered to be a feature. T is a user-defined threshold.

Rosten et al. show that S equal 9 has a high efficiency and reliability compared with other values [11]. Figure 1 illustrates the concept. The order of the pixel evaluation is determined by the machine-learning algorithm ID3 [12]. ID3 is a method used to generate a decision tree from a training dataset. FAST needs to be trained on an image dataset from the working environment; then obtain a decision tree to classify each center pixel as a feature point or not.

The FAST algorithm provides an interesting concept for feature extraction. A weakness of this algorithm is the pre-trained decision tree. A fixed decision tree cannot guarantee that each combination of pixels can be checked; this may produce incorrect results. Furthermore, the FAST feature extractor has to be trained again once the working environment has changed. This weakness restricts FAST to work insufficiently correct on computer vision applications such as the IPS which shall work without any prior knowledge of the environment.

To overcome the weakness of FAST, the *adaptive and generic accelerated segment test* (AGAST) feature extractor is proposed in [5].

AGAST is based on the same AST feature criterion as FAST, but uses a different decision tree. AGAST is trained based on a dataset which includes all possible combinations of 16 pixels on the circle. This ensures that the decision tree works in whatever environments. Moreover, AGAST introduces a dynamic-tree switching algorithm which can automatically change the decision trees. One tree is trained for homogeneous areas, and an other one is trained for heterogeneous areas.

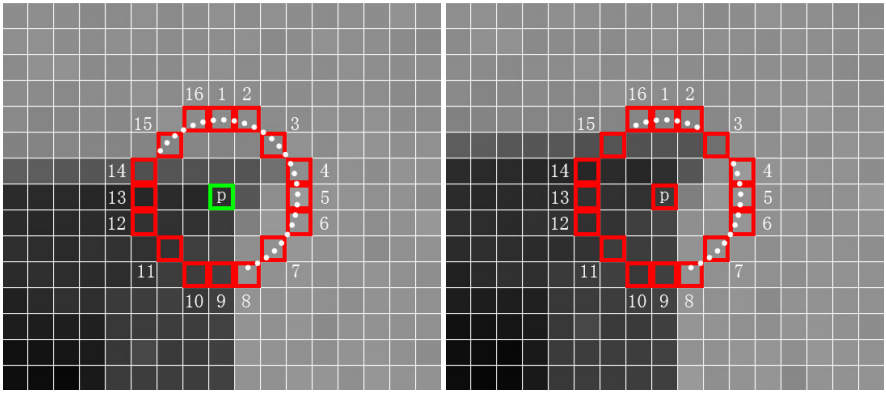

Figure 1: *Left*: The center pixel p is a FAST corner because there exist 10 continuous pixels on the circle satisfying the AST condition. *Right*: The center pixel p is not a FAST corner.

In this way, the performance of AGAST improves for random scenes. By combining these two improvements, AGAST works in any arbitrary environment without any training step. This makes AGAST very promising for IPS and other real-time computer vision applications.

The above-mentioned feature extraction algorithms only extract features that are good in the original scale of the image, but nothing is known about the quality of extracted features at other scales. Because of the variety of working environments and application projects, a scale-invariant feature extractor is desirable.

A very popular multi-scale feature extraction algorithm is the *scale-invariant-feature-transform* (SIFT) algorithm; see [3].

The SIFT method builds an image pyramid; at each level of the pyramid, an image octave is created with the same method as used in the Harris-Laplace algorithm. The *difference of Gaussian* (DOG) images are created by subtracting adjacent images in a Gaussian octave.

A keypoint (i.e. possible feature point) is detected by a local maximum or minimum within 26 adjacent positions in 3D space defined by subsequent layers of the scale space. A filter is applied for selecting only such feature points which are not within a homogeneous area.

Next, the outputted feature points are described by a rotation- and scale-invariant descriptor. An influence area of radius σ (where the feature point was detected in scale space) is selected to analyse the main direction of the feature. After this step, based on the calculated main direction, the influence area is rotated and subdivided into 4×4 windows around the feature point. Inside each window, gradient magnitudes are calculated and be put into an 8-bin gradient histogram, leading to $16 \times 8 = 128$ values. Finally, the resulting 128-dimensional vector is obtained as a descriptor of the feature point.

Since the SIFT performance is in general highly rated, it was rapidly becoming a gold standard in feature detection. Over the past decade, many SIFT-like algorithms were proposed.

Another well-known feature extraction algorithm is given by the *speeded-up-robust-features* (SURF) method; see [4]. SURF follows similar ideas as SIFT but it is significantly faster than SIFT. Instead of building DOG images, SURF uses several masks to detect local maximum or minimum response points. The masks are approximations of Hessian matrices which correspond to different Gauss convolution kernels. The size of the masks depends on the variance of the Gauss kernels. In this way, the scale space can be built with a fixed image but different masks. Same as with SIFT, the output feature point is the local maximum within 26 adjacent positions. The feature descriptor is generated with a similar method as for SIFT. Instead of using a gradient histogram, a *Haar wavelet* method is used. Finally, a 64-dimensional vector is obtained as feature descriptor. In many cases, SURF can obtain a result comparable to SIFT, and it is several times faster than SIFT.

Two further "famous" feature extraction algorithms are the *binary-robust-invariant-scalable-keypoints* (BRISK) method, see [13], and the *oriented-FAST-and-rotated-BRIEF* (ORB) method, see [14]. Both algorithms combine the scale-space concept and the FAST feature extractor idea. The features are detected from images at different scales by FAST.

In BRISK, the main direction of the influence area is calculated by "long-distance" sampling of point pairs. The direction in ORB is obtained by an *intensity-centroid method*. After rotation, all of the 512 "short-distance" sampled point pairs are compared and generate a binary BRISK feature descriptor. In ORB, a steered BRIEF [15] method is adapted; by a combination with the main direction and a predefined pixel comparison order, a binary feature descriptor can be generated.

A comparison between SIFT, SURF and ORB can be found in [16]. In the publication, the feature extractors are evaluated under a visual-odometry framework. The drift between measured trajectories and ground truth is listed and provides a valuable reference for each feature extractor under practical applications.

3 The Non-maximum Suppression Algorithm

In this section, details of the proposed non-maximum suppression algorithm is described. As mentioned in Section 1, in our research, the proposed method is applied to the AGAST feature extractor, but it is very easy to incorporate with other feature extractors.

The stand AGAST adapts a non-maximum suppression algorithm inherited from FAST, using a 3×3 square mask sliding over all features. It suppresses low-rating features in the neighborhood area. However, even after this suppression, more than 600 features per image remain with a threshold of 15.

Although a higher threshold could decrease the number of features, this results in many features close together in structured areas of the image, but no remaining fea-

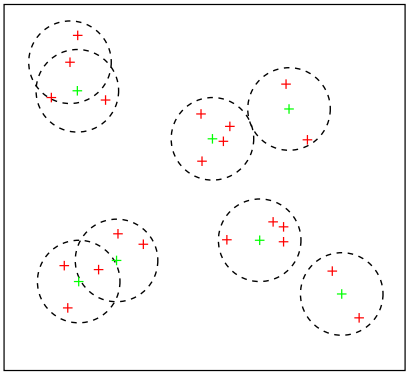

Figure 2: Graphical sketch for feature elimination. The green crosses are the features being kept; the red are suppressed. The radius of the circle is the minimum feature distance; the center point in the top-left circle is suppressed because it has not the highest score in its circle.

tures in less-structured areas. This is suboptimal because the accuracy of the optical navigation strongly depends on a good distribution of features over the whole image. At the same time, the number of features shall be kept low because it unnecessarily increases the processing time needed to track them without a significant improvement in accuracy.

In order to achieve real-time processing, the ideal number of features should be around 100. On the other hand, in an optical navigation application, the images are significantly different during the whole processing. Same with many other feature extractors, in AGAST, the number of features is controlled by a threshold. However, because of the highly non-linear relationship between feature numbers and threshold, the appropriate threshold is difficult to known. On the other hand, in practical optical navigation applications (e.g. IPS), images vary from frame to frame, thus adding new complexities.

In our proposed non-maximum suppression algorithm, a user just needs to define one parameter N_e. This parameter indicates the needed number of features from feature extractor. The algorithm can provide exactly the number of N_e features for the further process (i.e. stereo feature matching).

Altogether, the extended AGAST is defined by the following steps:

1. Use the normal AGAST to extract features with a very low threshold (e.g. 8) and calculate the features scores. (This method is the same as with FAST or AGAST.) This step can result in “many” features.

2. The scores of the features are compared within a circular area. The radius r of the circle is the given minimum feature distance; the feature with the highest score in its circle is kept, others are suppressed. In my tests, r=15 can lead to a good result. Figure 2 illustrates this concept. Because of the low threshold,

usually, more than 500 features remain after the above steps.

3. Divide the image into an $m \times n$ grid, let $m \times n \approx N_e$. Sort the features inside each grid cell based on the score of the features, return the top-one feature in the grid cell for the output feature list.

4. Assume that Step 3 leads to a number of N_g features. In the normal case, some grid cells include no features; that is we have that $N_g < N_e$. Therefore, a sorted global feature list is created by all of the features after Step 2. Mark all of the top-ranked features in the grid cells as *used* in the sorted global list. Start from the highest score of the global list, take the number of N_{us} *unused* features; let $N_{us} + N_g = N_e$. The combination of features from Step 3 and Step 4 is the final output.

The above steps guarantee that the feature extractor always returns the best quality and well-distributed features.

4 Experimental Results

The proposed framework is tested by combining with the IPS. IPS is a low-cost vision-aided inertial navigation which can measure the motion trajectory in unknown environments. The mean error of a measured trajectory is in general much below 1% of the traveled distance. Grießbach et al. show that the 3D error was about 0.65 m for a 410 meters track, see [17].

In the optical navigation process, images vary from frame to frame. If a fixed threshold of feature extractor is used then the number of extracted features may vary over a large range. In order to achieve real-time processing, the ideal number of features should be around 100. Hence, an algorithm is used in the original IPS which can automatically adapt a threshold. By analyzing the number of features in the previous frame, the algorithm calculates a new threshold for the current frame. However, because of the highly non-linear relationship between feature numbers and threshold, the performance of the adaption algorithm is barely satisfactory. By combining with the proposed framework, instead of a threshold, the number of outputs can be specified during processing; an adaption algorithm can directly control the needed feature number. The first test is designed to compare the performance of feature adaption algorithms.

In the original IPS, the threshold T_s of feature extractor is defined by user. The system starts with initialization of T_s, the feature extractor can provide number of N_s features for the next step (i.e. stereo feature matching).

After matching, the number N_{rm} of real matched features is obtained. Then a new threshold T_n can be calculated by Eq. (1). T_n is fed back to the feature extractor to control the feature number of the next frame:

$$T_n = \left(2 - \frac{N_{rm}}{N_{em}}\right) T_s \tag{1}$$

where N_{em} indicates the expected number of remaining features after the stereo-feature-matching step. Figure. 3 (left) illustrates the performance of the original adaption algorithm.

In the proposed framework, a user just needs to define one parameter N_{em}. The system starts with an initialization of N_e as feature number; the proposed framework can provide exactly the number of N_e features for the next step. After matching, the number N_{rm} of real matched features is obtained. The new required feature number N_{ne} can be calculated by Eq. (2). Figure. 3 (right) illustrates the performance of the original adaption algorithm.

$$N_{ne} = \left(2 - \frac{N_{rm}}{N_{em}}\right) N_e \tag{2}$$

From the figure, the proposed method significantly improves the stability of matched feature numbers. The successfully matched feature are used to calculate ego-motion of the system. The number of matched feature can strongly affect the performance of IPS. Therefore, the proposed method increase the robustness of the system.

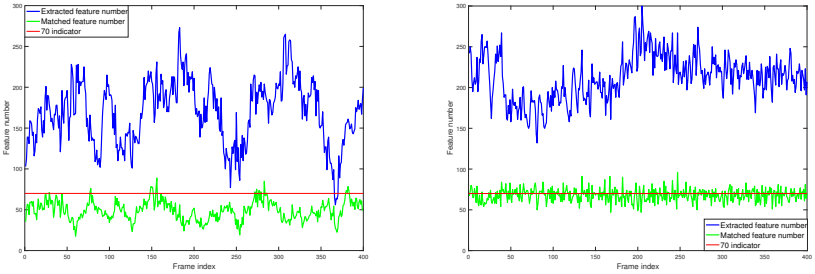

Figure 3: *Left*: Feature number plot of the original method; the blue line is the number of extracted features in each frame, the green line is the number of real matched features. The expected number of matched features is set to 70; this value is indicated by the red line. *Right*: Feature-number plot of the proposed framework.

The second test is to explore the non-maximum suppression algorithm which is working best for IPS, hence getting the most accurate optical navigation results. Therefore the tests are performed with realistic data including the entire processing chain. The performance is evaluated in terms of the accuracy of the resulting trajectory.

First, a dataset is recorded by walking with IPS through a realistic scene, an office building and the surrounding outdoor areas with a length of about 410 meters. Multiple walks have been recorded.

In order to evaluate the accuracy of the resulting trajectory the start and end position of the walk are exactly identical. One complete data sequence from a single walk is called Session. We recorded 8 sessions in total.

In an offline processing step, the IPS application is used to calculate the trajectory.

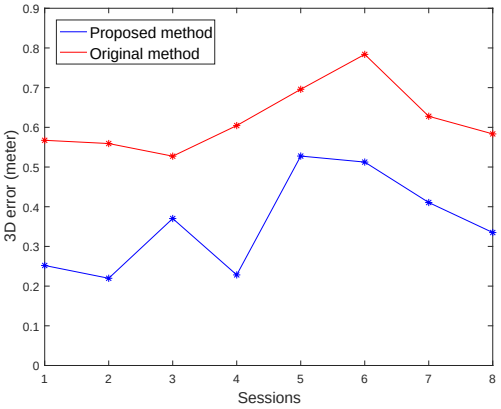

Figure 4: 3D-errors of 8 sessions. The blue line indicates the proposed method, the red line indicates the original AGAST non-maximum suppression algorithm.

Because a RANSAC algorithm is used for the optical navigation, for an identical video sequence and configuration, each run outputs a slightly different trajectory due to the random component. To get high accurate testing results, we run IPS application 20 times for each session for original IPS and the IPS combined with proposed framework respectively. More details of the test can be found in [18]. The average 3D error is calculated and be shown in Fig. 4. The results show the outstanding performance of the proposed method.

5 Conclusion and Outlook

In this paper, a novel non-maximum suppression framework is proposed. During the study, the framework is applied to the AGAST feature extraction algorithm, and it is very easy to incorporate with other feature extractors. The experimental results show that by combining with the proposed method, the number of features can be precisely controlled. This can increase the robustness of the whole system. At the same time, by using the proposed framework, the 3D error of the measured trajectory by IPS is significantly decreased. These prove that the proposed method is very productive.

Our future works will address the uncertainties of features. IPS is a Kalman filter based system; this means the uncertainties of features must be modeled and be handled via uncertainties propagation steps. The feature extraction and matching results are affected by image noise. The key task is to model the image noise and propagate the noise to the uncertainties of features.

References

[1] C. Harris and M. Stephens. A Combined Corner and Edge Detector. Alvey Vision Club, 1988.

[2] J. Shi and C. Tomasi. Good features to track. In *, 1994 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94*, 1994.

[3] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60, 2004.

[4] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *Computer vision ECCV 2006*. Springer, 2006.

[5] Elmar Mair, Gregory D. Hager, Darius Burschka, Michael Suppa, and Gerhard Hirzinger. Adaptive and generic corner detection based on the accelerated segment test. In *Computer Vision ECCV 2010*. Springer, 2010.

[6] Jean-Michel Morel and Guoshen Yu. ASIFT: A new framework for fully affine invariant image comparison. *SIAM Journal on Imaging Sciences*, 2, 2009.

[7] Reinhard Klette. *Concise Computer Vision*. Undergraduate Topics in Computer Science. Springer London, 2014.

[8] S. M. Smith and J. M. Brady. SUSAN - A New Approach to Low Level Image Processing. *International Journal of Computer Vision*, 1995.

[9] J. Bresenham. An incremental algorithm for digital plotting. In *Proc. ACM Natl. Conf.*, 1963.

[10] Edward Rosten and Tom Drummond. Fusing points and lines for high performance tracking. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2. IEEE, 2005.

[11] Edward Rosten, Reid Porter, and Tom Drummond. Faster and better: A machine learning approach to corner detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32, 2010.

[12] J. Ross Quinlan. Induction of decision trees. *Machine learning*, 1, 1986.

[13] Stefan Leutenegger, Margarita Chli, and Roland Yves Siegwart. BRISK: Binary robust invariant scalable keypoints. In *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011.

[14] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB: an efficient alternative to SIFT or SURF. In *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011.

[15] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. Brief: Binary robust independent elementary features. *Computer Vision ECCV 2010*, 2010.

[16] Hsiang-Jen Chien, Chen-Chi Chuang, Chia-Yen Chen, and Reinhard Klette. When to use what feature? SIFT, SURF, ORB, or A-KAZE features for monocular visual odometry. In *Image and Vision Computing New Zealand (IVCNZ), 2016 International Conference on*. IEEE, 2016.

[17] Denis Grießbach, Dirk Baumbach, and Sergey Zuev. Stereo-Vision-Aided Inertial Navigation for Unknown Indoor and Outdoor Environments. In *2014 International Conference on Indoor Positioning and Indoor Navigation*, 2014.

[18] Hongmou Zhang, Jürgen Wohlfeil, and Denis Grießbach. Extension and evaluation of the AGAST feature detector. In *XXIII ISPRS Congress annals 2016*, volume 3, 2016.