WIT: Window Intensity Test Detector and Descriptor

T.W.U.Madhushani*, D.H.S.Maithripala†, J.V.Wijayakulasooriya‡

*Postgraduate and Research Unit, Sri Lanka Technological Campus, CO 10500, Sri Lanka.

†Department of Mechanical Engineering, University of Peradeniya, KY 20400, Sri Lanka

‡Department of Electrical and Electronic Engineering, University of Peradeniya, KY 20400, Sri Lanka

Abstract—**In this paper, we present a novel algorithm for feature point detecting and matching. Algorithm is robust to image noise and computationally simple, hence it is more suitable for on-board processing robotics applications. We introduce two repeatability measurements, that can be used in training phase to optimize parameters of the algorithm. Proposed detection algorithm is free from extracting multiple points around the same feature point. We suggest a technique to overcome the effect of scale variation in detection stage, for a compensation of computational simplicity. In matching stage, we directly utilize resulting data from detection, which makes the overall process much faster. We procure a technique for matching point error suppression, for a compensation of computational simplicity. Simulation results provided in the paper illustrate the accuracy and efficiency of the proposed algorithm.**

I. INTRODUCTION

Vision can be considered as the most popular sensing technique when it comes to navigation of robots, who tend to suffer from uncertainties in operating environment. The reason for this popularity is; for a wide variety of application purposes, vision sensing systems are capable of procuring more information about the terrain, than amalgamation of other sensing techniques. This research area can be mainly divided in to two widely researched streams; visual odometry (VO) [1] and visual localization and mapping (V-SLAM) [2]. Identifying corresponding points of two images can be considered as the basic step for both these fields. In spite of the fact that many efficient methods [3], [4], [5], [6], [7], [8] have been developed, addressing this problem, it still continues to demand more and more accurate and computationally simple algorithms. Since a wide class of robotics applications benefit from on-board processing, this demand tends to grow rapidly. In this paper, we present a novel algorithms for identifying feature points and matching.

Over decades, researchers have presented different types of detectors for identifying edges [9] and corner points [4], [10], [11]. Most of them rely on extracting an unique image point, using reliable techniques of comparisons among neighboring pixels. To reduce the effect of noise, first step of most of these algorithms is, Gaussian blurring. Laplacian of Gaussian, Difference of Gaussian and Histograms are commonly incorporated mathematical concepts in these algorithms. However, above mentioned approaches tend to increase the computational complexity. FAST [11] has taken a fairly different approach, while incorporating the concept of enthalpy, to detect feature points; this reduces the computational cost of the algorithm. Nevertheless, almost all feature point detection algorithms suffer from, extracting multiple points around an unique point or a corner.

Different types of feature point descriptors [12] are also present in state of the art. Main concerns in this research area can be identified as, simplifying the computational complexity and reducing the memory usage. To achieve these common goals, researchers widely incorporate concepts of dimensionality reduction such as, Principle Component Analysis [12], [13] and Linear Discriminant Analysis [14]. However, it is required to calculate the full descriptor before applying reduction techniques. BRIEF [15] descriptor bypasses this necessity, by using binary strings approach. ORB [6], a modified combination of FAST and BRIEF, consists of a detector and a descriptor. This is a widely used algorithm in many applications, due to its computational simplicity, reliability and efficiency.

In section II we present a novel algorithm for feature point detection. WIT detector: Window Intensity Test detector. Our approach is based on comparing average pixel intensities of rows and columns of a chosen small image window; therefore Gaussian blurring step becomes redundant, which reduces number of computations substantially, while procuring sufficient robustness for image noise. We incorporate a different definition for a feature point, which based on repeatability, than conventional definition, which based on uniqueness. This doesn't diminish reliability or accuracy of WIT in any way, because algorithm is capable of capturing essential features of a proficient detector. In section II-A we present two different repeatability measuring methods. One method is for testing the repeatability of individual feature points and other is for testing repeatability of all detected feature points as a whole. These methods can be utilized for tuning parameters of the proposed detector. To our knowledge this is faster than state of the art algorithms. To demonstrate the versatility of our algorithm we present a comparison with FAST, which considered to be the fastest detection algorithm for feature point detection, in section II-B. Another appealing quality of WIT is, that it doesn't extract multiple points around a feature point; WIT gives at most one feature point per window. We present resulting images of simulations to corroborate that. Furthermore, we suggest a method for addressing the effect of scale variations, for a compensation of computational simplicity.

We present a novel descriptor for feature point matching in section III. WIT descriptor: Window Intensity Test descriptor. In order to reduce computational cost, we directly utilize resulting data from WIT detector. In section III-A we present a method to achieve increment in accuracy, for a compensation of computational simplicity, by suppressing non-matching feature points. To illustrate the performance of the descriptor, we present data on percentage accuracy for three sets of benchmark images in section III-B. We use ground truth for calculating percentage values. In section IV we conclude that considering overall performance WIT outperforms most of the state of the art detection and matching algorithms in speed, while maintaining a appreciable accuracy. In appendix we provide all original images, which were used for simulations.

Algorithm presented in this paper can be exploited for applications, which benefit from on-board processing, such as UAVs and micro robots.

II. FEATURE POINT DETECTION

Consider a $n \times n$ pixel window of a gray scale image that we will denote by P. We assume that each pixel can take an matrix where

theinteger value betweeni^{th} element is the row sum of the$0 - (2^N - 1)$. Let $r(P)$ be the columni^{th} row of

P and let $c(P)$ be the column matrix where the i^{th} element is the column sum of the i^{th} column of P. Let $r^{-}(P)$ and $c^{-}(P)$ be the 1^{st} left circular shifts of $r(P)$ and $c(P)$ respectively.

Let $e^r(P) = r(P) - r^{-}(P)$ and $e^c(P) = c(P) - c^{-}(P)$. Here, Letwe consider first $n-1$ entries of matrices $e_r(P)$ and $e_c(P)$

$S_r(P) = \{i \mid |e_{ri}| > t\}$, $S_c(P) = \{i \mid |e_{ci}| > t\}$. where, t is threshold intensity value.

Let,

$$\alpha = i \in S_{\max r}(P)|e_{ri}|, \beta = i \in S_c(P)|e_{ci}|.$$

We define the unique pixel point $(\alpha,\beta) \in S^r(W) \times S^c(W)$ to be the feature point in W. Note that, feature point is a point, which can be uniquely identified and repeated across consecutive images. In conventional definition, a feature point is an unique image point. Two questions need to be answered about the effectiveness of this scheme: i) given a generic image will there exist a n and t such that there are sufficiently many feature points, ii) given that a feature point is detected what is the probability of detecting the same point in a subsequent image. The answer to the first question depends on how generic the feature point is. The answer to the second question depends on how robust the test is on images corresponding to parallel translations of the camera.

Consider the space, Ω, of all possible patch images. There are $n^{\Omega} = n^2! \, (2^N)^{n_2}$ such images. Denote by A the set of possible images that will give the answer yes to the test described above and let n_A be the number images in it. Then the probability of getting a positive answer to the test for an arbitrary image w is

$$p(w \in \mathcal{A}) = \frac{n_A}{n_\Omega}.$$

Two imagesequivalent if and only if $w_i, w_j \in W_{Wj}$in a windowcorresponds to an image of the W are said to be same scene taken from a purely translated camera. Denote by $[w]$ the equivalent class of w in Ω. The test is 100% repeatable if $[w] \in A$ for all $w \in A$.

Considering row sum and column sum acts as an averaging method for pixel intensities. This causes reduction in image noise and makes any kind of blurring at the initial stage redundant. This method posses the advantage of not getting multiple feature points around the same point because it gives only one or no feature point for a window. Window size can be chosen in a way that we get 500 or more windows per image. Threshold intensity value should be chosen according to the overall intensity values of the image. For results in paper, we have used $t = 150$ when $n = 10$ and $t = 300$ when $n = 20$.

A. Repeatability

Consider two arbitrary unknown camera poses i and j. The test carried out on some window W in the i^{th} image gives a positive answer indicating the identification of a feature point. contains this feature point in the arbitraryThat is, it tells us that $w_i \in A$. Let w_{jj}thbe the image thatcamera pose. We

are interested in estimating the probability of findinggiven that $w_i \in A$. That is we are interested in finding$w_j \in A$

$$P(w_j \in \mathcal{A} \,|\, w_i \in \mathcal{A}) = \frac{P(w_j \in \mathcal{A} \cap w_i \in \mathcal{A})}{P(w_i \in \mathcal{A})}.$$

if Two imagesw_j corresponds to an image of the same scene, that gives$w_i, w_j \in \Omega$ are said to be equivalent if and only w_i, taken from a purely translated camera. Denote by $[w]$ the equivalent class of w. Then the test will detect a feature point in two consecutive images with 100% certainty if and only if

Let $n([w_i])$

be the number of images in$Aw) \in$ A, is given byimpliesbe the number of images in$[w] \in \bar{A}$ andA$\cap w \in A[w_i]_c$.

Then,implies$[wP_i][($and$ww]_j \in\in$ A$\cap n$A$(\bar{}$A$\cap_c$. $[ww_{ii}]) \in$

$$P(w_j \in \mathcal{A} \cap w_i \in \mathcal{A}) = \frac{n(\mathcal{A} \cap [w_i])}{n([w_i])}.$$

Hence we have

$$P(w_j \in \mathcal{A} \,|\, w_i \in \mathcal{A}) = \frac{n(\mathcal{A} \cap [w_i])}{n([w_i])} \frac{n_\Omega}{n_A}.$$

Let p be an arbitrary pixel in the image. There exists n^2 possible windows which include p as a point in image window. If the test gives same point p as the feature point for two or more such windows we claim point p is repeatable. For a feature point we define a measure *atomistic certainty of repeatabilit* as the ratio between number of windows which give p as the feature point and number of windows which include p as a point. This gives an indication about repeatability of
each feature point. This is somewhat similar to the notion of cornerness.

Consider the case where we shift all windows of the image by same pixel distance. We define another measure *holistic certainty of repeatability* as the ratio between number of feature points which are repeated, and number of all feature points. This gives an indication about repeatability of feature points when considering the image as a whole; not individual windows. This can be used to predict the repeatability percentage of feature points and utilized for parameter tuning; window size n and intensity threshold value t for a training image set from an application area.

B. Simulation results

In state of the art, FAST is considered to be the fastest feature point detection algorithm. We present a comparison of feature point detection time between FAST and WIT for a publicly available image sequence containing 30 images. Used order of the image sequence is, bikes, graffiti, trees, ubc and wall. (http://www.robots.ox.ac.uk/ vgg/research/affine)

Here we set threshold values so that both algorithms detect approximately same number of points. In all simulation results we compare execution time using MATLAB 2015b.

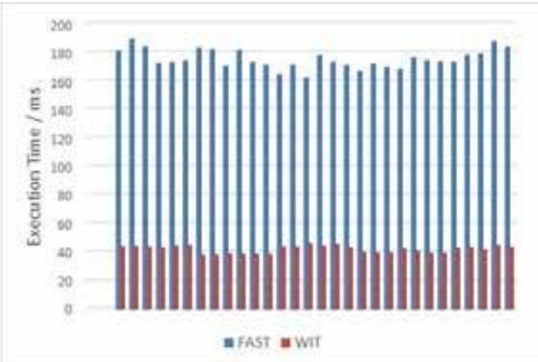


Fig. 1: Execution Time

Fig. 2: Number of Points

Following table shows a comparison of feature point detection and repeatability between FAST and WIT. The given execution time is the average feature point detection times for two images in a set. Here, we set threshold values for two algorithms in a way that, both algorithms detect approximately similar number of points from same image. Results are given for three different sets of images.

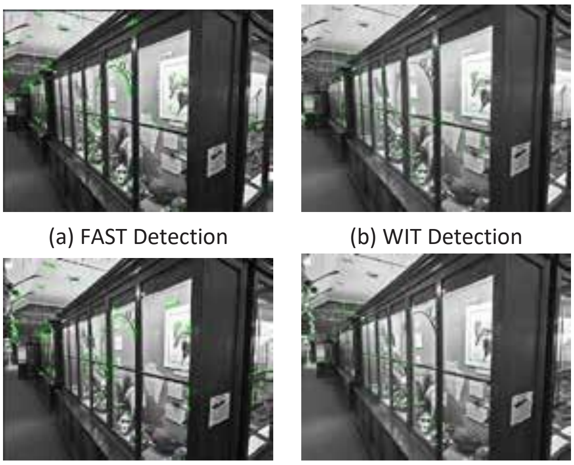
In order to calculate predicted percentage (predicted percentage of repeatability), we shift all image windows in first image by a random pixel distance (all image windows are shifted by the same pixel distance) and perform the test. Predicted percentage is the percentage of repeated feature points from originally detected feature points.

Feature Point Detecting Results		
Image	FAST	WIT

Indoor 1	113	103
Indoor 2	108	92
Repeated Points	44	46
Execution Time /(ms)	420.2	76.7
Repeated Percentage	37.3	44.7
Predicted Percentage	-	46.8
Lena 1	103	100
Lena 2	27	56
Repeated Points	18	25
Execution Time /(ms)	395.8	65.6
Repeated Percentage	17.5	25
Outdoor 1	114	104
Outdoor 2	105	112
Repeated Points	62	61
Execution Time /(ms)	389.4	57.2
Repeated Percentage	54.4	58.7
Predicted Percentage	-	61.1

Resulting images of simulations are shown from figure (3)- figure (5). We chose image sets corresponding to three different application areas; an indoor environment, human face and outdoor environment. Figure (3) illustrates performance of the algorithm in an indoor environment. These images are rich with corner points and edges. Images (1-a) and (1-b) corresponds to resulting images from FAST and WIT detector respectively. Images (1-c) and (1-d) corresponds to performance of FAST and WIT on a translated version of previous image. Images used in figure (4) are well known benchmark images. Here, considered second original image has a scale variation with respect to first original image. Although WIT detector performs better than FAST, results show that, WIT detector suffers from scale variations. This can be remedied using different scaled window sizes, with a compensation of computational simplicity. Figure (5) shows performance of FAST and WIT detector in an outdoor environment, which is very likely to encounter in robotic navigation in the field. Images (3-a) and (3-b) corresponds to resulting images from FAST and WIT detector respectively. Images (3-c) and (3-d) corresponds to performance of FAST and WIT detector on a translated version of previous image.

Note that, for all images, FAST has detected multiple points around corners, while WIT gives exactly one feature point corresponding to a corner.



(a) FAST Detection (b) WIT Detection

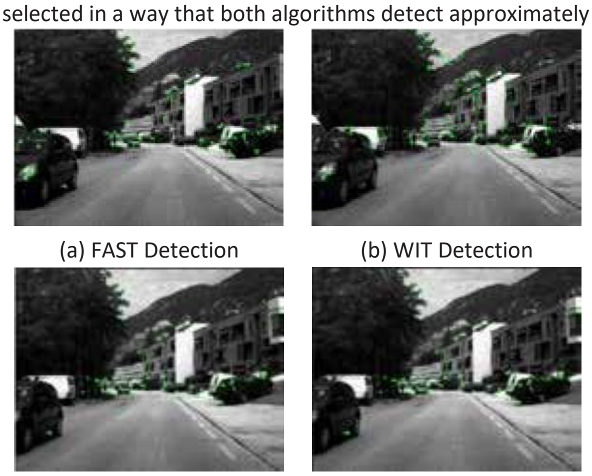
(c) FAST Detection (d) WIT Detection

Fig. 3: Detected feature points in an indoor environment using algorithms FAST and WIT. For a better comparison, threshold values have been selected in a way that both algorithms detect approximately similar number of points from same image.

(a) FAST Detection (b) WIT Detection

(c) FAST Detection (d) WIT Detection

Fig. 4: Detected feature points in a human face using algorithms FAST and WIT. For a better comparison, threshold values have been selected in a way that both algorithms detect approximately similar number of points from same image.



(a) FAST Detection (b) WIT Detection

(c) FAST Detection (d) WIT Detection

Fig. 5: Detected feature points in an outdoor environment using algorithms FAST and WIT. For a better comparison, threshold values have been selected in a way that both algorithms detect approximately similar number of points from same image.

III. FEATURE POINT MATCHING

Consider a detected feature point p in the first image. Lets denote the corresponding window as P. Suppose the point p corresponds to i^{th} row and j^{th} column of the window. Lets denote point p as (α,β). Suppose $r(P)$ and $c(P)$ are respective row sum and column sum of the image window. For all detected feature points of second image we select windows such that detected feature point corresponds to i^{th} row and j^{th} column of windows. Which means, we select a new window for each feature point in second image, such that position of feature point in the window can be denoted as (α,β). We perform the test for all new windows and discard the windows, which fail to detect the point corresponds to i^{th} row and j^{th} column as the feature point. Lets denote row sum and column sum of each window of second image as $\hat{r}(P)$ and $\hat{c}(P)$.

Let,

$$\hat{e}_r(P) = r(P) - \hat{r}(P)$$
$$\hat{e}_c(P) = c(P) - \hat{c}(P).$$

For a candidate matching feature point, let $s(P)$ be a scalar, corresponds to number of elements in $\hat{e}_r(P)$ and $\hat{e}_c(P)$ which satisfy the condition $-t_1 < e < t_1$. Where, t_1 is window threshold intensity value. We denote a vector S_w, which contains all $s(P)$ of candidate matching feature points, correspond to a feature point in first image. If $max\ S_w > t_2$, where t_2 is the threshold value for number of matching rows and columns of windows, we consider the corresponding point to be the matching feature point. For results presented in this paper, we have chosen $t_1 = 100$ and t_2 can be chosen to be the window size.

A. Error point suppression and increasing number of match-ing points

In this section we describe a method to suppress false matching points and increase total number of matching points, for a compensation of computational simplicity. After detecting feature points of first image we shift all windows of the second image by a random shift and detect feature points. For new point set we perform the point matching test same as previous case. After sufficient number of window shifts we decide matching points to be point pairs which give higher number of times as matching points.

B. Simulation results

In this section, we illustrate performance of WIT descriptor, by presenting number of matched points and percentage accuracy for previously considered image sets.

Feature Point Matching Results	
Image	WIT
Indoor 1 & Indoor 2	
Matched Points	14
Correct Percentage	100
Lena 1 & Lena 2	
Matched Points	11
Correct Percentage	54.5
Outdoor 1 & Outdoor 2	
Matched Points	18
Correct Percentage	77.7

Feature points matched resulting images are shown from figure (6)-(8). WIT Descriptor performs well with translated images, but this suffers from scale variations and rotations. Figure (6) and (8) illustrates performance of WIT descriptor with translated indoor and outdoor images respectively. Figure (7) shows performance of WIT descriptor for images of a human face with scale variation.

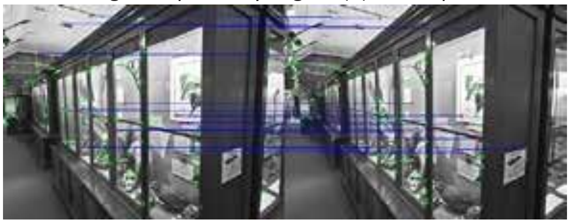


Fig. 6: Matched feature points in an indoor environment using WIT descriptor. Feature points have been identified using WIT detector. Points marked by green color are unmatched feature points

IV. CONCLUSIONS

This paper presents a novel algorithm, which can be used to detect and match feature points accurately and efficiently. The

Fig. 7: Matched feature points in a human face using WIT descriptor. Feature points have been identified using WIT detector. Points marked by green color are unmatched feature points

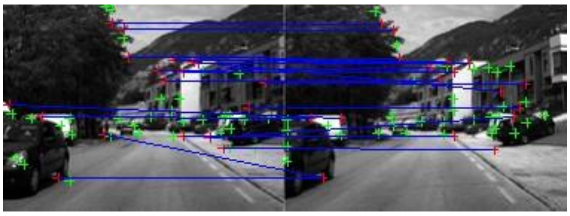


Fig. 8: Matched feature points in an outdoor environment using WIT descriptor. Feature points have been identified using WIT detector. Points marked by green color are unmatched feature points

process is based on an averaging method of intensity values of small image windows. This quality makes the initial Gaussian blurring step, which is used to reduce the effect from image noise, redundant. The paper suggests two methods for parameter tuning in detection phase and an error suppression method in matching phase. These additional processes can be utilized, at the expense of computational simplicity. Considering overall performance of simulation results it can be concluded that WIT outperforms most of the state of the art detection and matching algorithms in speed, while maintaining a appreciable accuracy

Appendix

In this section we provide all original images, that are used for simulations. Figure (9) and (11) contain images with translations. Figure (10) contains images with scale variation.

Fig. 9: Indoor environment

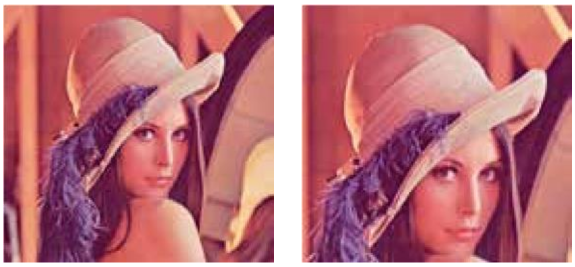


Fig. 10: Human face

Fig. 11: Outdoor environment

References

[1] D. Scaramuzza and F. Fraundorfer, "Visual odometry [tutorial]," *IEEE Robotics and Automation Magazine*, vol. 18, Des 2011.
[2] J. Fuentes-Pacheco, J. Ruiz-Ascencio, and J. Rendn-Mancha, "Visual simultaneous localization and mapping: a survey," *Artificial Intelligence Review*, vol. 43, p. 5581, Jan 2015.

[3] D. Lowe, "Distinctive image features from scale-invariant keypoints," *ComputerVision and Image Understanding*, vol. 20, pp. 91–110, 2004.
[4] C. Harris and M. Stephens, "A combined corner and edge detector," in *Alvey Vision Conference*, 1988, pp. 145–151.
[5] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," in *In European Conference on Computer Vision,*, May 2006.
[6] C. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: an efficient alternative to sift or surf," in *International Conference on Computer Vision*, Nov 2011, pp. 2564–2571.
[7] M. Agrawal, K. Konolige, and M. Blas, "Censure: Center surround extremas for realtime feature detection and matching," in *European Conference on Computer Vision*, Oct 2008.
[8] S. Smith and J. Brady, "Susana new approach to low level image processing," *International Journal of Computer Vision*, vol. 23, pp. 45– 78, May 1997.
[9] W. McIlhagga, "The canny edge detector revisited," *International Journal of Computer Vision*, vol. 91, pp. 251–261, Feb 2011.

[10] T. Shi and T. Tomasi, “Good features to track,” in *IEEE Conference on Computer Vision Pattern Recognition*, June 1994.
[11] E. Rosten and T. Drummond, in *Machine learning for high-speed corner detection*, 2006, pp. 430–443.
[12] K. Mikolajczyk and C. Schmid, “A performance evaluation of local descriptors,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, pp. 1615 – 1630, Oct 2005.
[13] Y. Ke and R. Sukthankar, “Pca-sift: a more distinctive representation for local image descriptors.” in *Computer Vision and Pattern Recognition*, July 2004.
[14] G. Hua, M. Brown, and S. Winder, “Discriminant embedding for local image descriptors.” in *International Conference on Computer Vision*, 2007.
M. Calonder, V. Lepetit, C. Strecha, and P. Fua, “Brief: Binary robust independent elementary features,” in *In European Conference on Computer Vision,*, 2010, pp. 778–792.