Evaluating the Robustness of Feature Correspondence using Different Feature Extractors

Shady Y. El-Mashad* and Amin Shoukry*†
*Computer Science and Engineering Department
Egypt-Japan University for Science and Technology (E-JUST)
Alexandria, Egypt
email:{shady.elmashad, amin.shoukry}@ejust.edu.eg
†Computer and Systems Engineering Department
Alexandria University
Alexandria, Egypt

***Abstract*—The importance of choosing a suitable feature detector and descriptor to find the optimal correspondence between two sets of image features has been highlighted. In this direction, this paper presents an evaluation of some well known feature detectors and descriptors; including HARRIS-FREAK, HESSIAN-SURF, MSER-SURF, and FAST-FREAK; in the search for an optimal detector and descriptor pair that best serves the matching procedure between two images. The adopted matching algorithm pays attention not only to the similarity between features but also to the spatial layout in the neighborhood of every matched feature. The experiments conducted on 50 images; representing 10 objects from COIL-100 data-set with extra synthetic deformations; reveal that HARRIS-FREAK's extractor results in better feature correspondence.**

***Keywords*-Features Matching; Features Extraction; Topological Relations; Graph Matching; Performance Evaluation; Quadratic Assignment Problem.**

I. Introduction

Image matching or in other words, comparing images in order to obtain a measure of their similarity, is an important computer vision task. It is involved in many different applications, such as object detection and recognition, image classification, content based image retrieval, video data mining, image stitching, stereo vision, and 3D object modelling. A general solution for identifying similarities between objects and scenes within a database of images is still a faraway goal. There are a lot of challenges to overcome such as viewpoint or lighting variations, deformations, and partial occlusions that may exist across different examples. Furthermore, image matching as well as many other vision applications rely on representing images with sparse number of distinct keypoints. A real challenge is to efficiently detect and describe keypoints, with robust representations invariant against scale, rotation, view point change, noise, as well as combinations of them [1].

Keypoint detection and matching pipeline has three distinct stages which are feature detection, feature description and feature matching. In the feature detection stage, every pixel in the image is checked to see if there is a unique feature at this pixel or not. Subsequently, during the feature description stage, each region (patch) around the selected keypoints is described with a more robust and invariant descriptor which can be used to match against other descriptors. Finally, at the feature matching stage, an efficient search for prospective matching descriptors in other images is made [2].

In the context of matching, a lot of studies have been used to evaluate interest point detectors as in [3], [4], [5]. On the other hand, little efficient work has been done on the evaluation of local descriptors. K. Mikolajczyk and C. Schmid [6] proposed and compared different feature detectors and descriptors as well as different matching approaches in their study. Although this work proposed an exhaustive evaluation of feature descriptors, it is still unclear which descriptors are more appropriate in general and how their performance depends on the interest point detector.

In this paper, a brief discussion of some feature detectors and descriptors is given. They have been selected because of their high performance and low complexity. Unlike other literature reviews which aim to solely compare between feature detectors and descriptors, the present work aims at finding a suitable feature detector and descriptor combination to serve the matching procedure between two images. In addition, the proposed matching algorithm, "Similarity-Topology Matching" [7], is reviewed. Finally, intensive experiments are performed to find the most compatible feature detector and descriptor combination with the proposed matching approach to get a superior performance.

This paper is organized as follow: the selected state-of-the-art feature extractors are briefly described in section 2. The proposed matching approach is reviewed in section 3. Section 4, presents the conducted experiments to evaluate the performance of the matching approach with different feature extraction techniques. Finally, the conclusion of this work and the recommendations for future work are presented in section 5 and 6, respectively.

II. Feature Detection and Extraction

Unlike other literature reviews which aim to solely compare between feature detectors and descriptors, the present work aims at finding a suitable feature detector and descriptor combination to serve the matching procedure between two images. Firstly, a brief discussion of some feature detectors and descriptors relevant to this study is given.

A. Feature Detectors

1) FAST-HESSIAN: HESSIAN is considered as a blob detector [8]. The determinant of the Hessian matrix is used to detect the location of a keypoint as well as its scale. A local maximum of the determinant denotes the existence of a blob. The Hessian matrix $H(x, \sigma)$ at a point $X = (x, y)$ at a scale σ is expressed in (1):

$$H(x, \sigma) = \begin{bmatrix} L_{xx}(x, \sigma) & L_{xy}(x, \sigma) \\ L_{xy}(x, \sigma) & L_{yy}(x, \sigma) \end{bmatrix} \quad (1)$$

Where $L_{xx}(x, \sigma)$ is the convolution of the Gaussian second order derivative with the image I at point x and similarly for $L_{xy}(x, \sigma)$ and $L_{yy}(x, \sigma)$. In order to speed up the calculations in (1), the box filters technique approximates the second order Gaussian derivatives using integral images. Also, the scale space is constructed by increasing the filter size instead of decreasing the image size as shown in (2).

$$det(H_{approx}) = D_{xx}D_{yy} - (0.9D_{xy})^2 \quad (2)$$

2) Maximally Stable Extremal Region (MSER): MSER is considered as a region detector [9]. MSER is a connected component of an appropriately thresholded image. The word extremal indicates the difference in pixels intensities which exist inside and outside the MSER. The maximally stable in MSER describes the property optimized in the threshold selection process. The set of extremal regions (E), the set of all connected components, has a number of properties. Firstly, extremal regions (E) are unchangeable under the monotonic change. Secondly, continuous geometric transformations preserve topology of the regions. Finally, the number of extremal regions is less than the number of pixels, this leads to a preservation of these regions under a broad class of geometric and photometric changes. Implementation Details. The enumeration of the extremal regions (E) is nearly linear in terms of the image pixels number. Initially, the pixels are sorted by their intensities. Then, pixels are marked in the image and a list of connected components and their areas as a function of intensity is sorted by using the union-find algorithm. The maximally stable are determined within the extremal regions as those corresponding to thresholds were the relative area change as a function of relative change of threshold is at a local minimum. In other words, the MSER are the regions inside the image where local binarization is stable over a large range of thresholds.

3) Feature from Accelerated Segment Test (FAST): FAST is considered as a corner detector [10] [11]. FAST algorithm works in the following manner: 1. Given any pixel p in an image, it is required to know whether it is an interest point or not? 2. Set a threshold intensity value T. 3. Assume a circle of 16 pixels around the pixel p. 4. The point is considered an interest point, if there are at least N contiguous pixels (out of the 16 pixels) each having an intensity above or below the intensity at p by the threshold T. To enhance the algorithm speed, first the intensity of pixels numbered 1, 5, 9 and 13 is compared with the intensity of pixel p. If at least three of the four pixel values do not satisfy the condition given in step 4, then this pixel is not an interest point and is rejected. Otherwise, the 16 pixels are checked.

4) HARRIS: HARRIS is considered as a corner detector [12]. It is based mainly on the second moment matrix (the auto-correlation matrix) as shown in (3). This matrix represents the gradient distribution in a local neighbourhood of a point.

$$M = \sigma_D^2 g(\sigma_I) * \begin{bmatrix} I_x^2(x, \sigma_D) & I_x I_y(x, \sigma_D) \\ I_x I_y(x, \sigma_D) & I_y^2(x, \sigma_D) \end{bmatrix} \quad (3)$$

The local image derivatives are computed using Gaussian kernels of scale σ_D. Then the derivatives are smoothed using Gaussian window of scale σ_I. The eigenvalues of this matrix are used to represent the variations in two orthogonal directions in a patch around the point which is defined as σ_I. Corners can be found in an image when the eigenvalues in both directions are large. Harris proposed an evaluation of the cornerness as shown in (4), where $det(M)$ and $trace(M)$ are the determinant and the trace of the matrix (M) respectively.

$$cornerness = det(M) - \lambda trace(M) \quad (4)$$

As the determinant of a matrix is the product of its eigenvalues and the trace is the sum of them. Hence, a corner is detected when both eigenvalues are large. In most of the feature detection survey, Harris corner is considered the most informative and repeatable detector.

B. Feature Descriptors

1) Speeded up Robust Features(SURF): SURF is considered as a fast scale and rotation invariant interest point descriptor for detecting features in an image [8] [13]. It seems that the description of the nature of the underlying image intensity pattern is more distinctive than histogram based approaches. The use of integral images make the descriptor competitive in terms of speed. The descriptor gives the distribution of the intensity content within the interest point neighborhood. This is done on the distribution of first order Haar wavelet responses in x and y directions.

Orientation Assignment: In order to achieve invariance to image rotation each detected interest point is assigned

a reproducible orientation. Haar wavelet responses of size 4σ are calculated for a set of pixels within a radius of 6σ of the detected point, where σ refers to the scale at which the point has been detected. The responses are weighted with a Gaussian centered at the interest point. The dominant orientation is selected by rotating a circle segment covering an angle of 60° around the point. At each position, the x and y-responses within the segment are summed and used to form a new vector. The longest vector lends its orientation to the interest point.

Descriptor based on Sum of Haar Wavelet Responses: The first step consists of constructing a square window centered on the interest point and oriented along the orientation selected in the previous step. The size of this window is 20σ. The window is split up regularly into smaller 4x4 square sub-regions. For each sub-region, haar wavelet responses are calculated at 5x5 regularly spaced sample points. Therefore, each subregion contributes four values to the descriptor vector leading to an overall vector of length 4x4x4 = 64. The resulting SURF descriptor is invariant to rotation, scale, and brightness.

2) Fast Retina Keypoint (FREAK): FREAK is a new descriptor inspired by the Human Visual System and more precisely the retina [14]. FREAK is considered as a fast, compact and robust keypoint descriptor. Based on the sampled retinal patterns; image intensities are compared pair by pair yielding successive binary strings. In order to have a highly structured pattern, the pairs should be wisely chosen so as to reduce the dimensionality of the descriptor. This pattern should mimic the saccadic search of the human eyes.

Retinal sampling pattern: The retinal (circular) sampling grid has been used. It has higher density of points near the center and the density of points decreases exponentially. Each sample point is smoothed to reduce its sensitivity to noise by Gaussian kernels. The exponential change in size and the overlapping receptive fields are considered the main difference against other techniques.

Coarse-to-fine descriptor: The binary descriptor has been constructed by thresholding the difference between every pairs of receptive fields and their corresponding Gaussian kernel. A large descriptor consisting of thousands of pairs is obtained. Unfortunately, most of these pairs are useless in efficiently describing an image. The algorithm used to decide the most relevant pairs is described as follows:

Firstly, a matrix containing all the extracted keypoints is constructed. Each row in the matrix represents a keypoint descriptor made of a combination of all possible pairs. Then, as a high variance is required in order to have a discriminant feature, therefore a rearrangement of the columns with respect to the highest variance is made.

Saccadic search: The idea of the saccadic search has been used by analyze the descriptor in several steps. Searching with the first 16 bytes of the descriptor is used as a primary filter. If the distance is less than a threshold, the search is extended to include the next 16 bytes to get finer information.

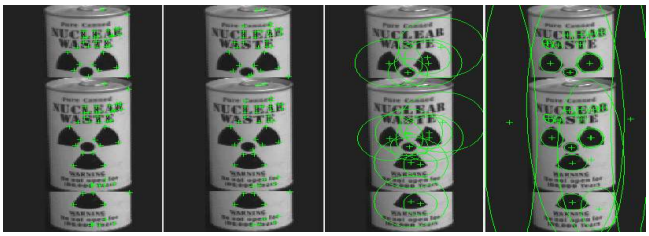


Figure 1. Feature detectors example (from left to right: HARRIS, FAST, HESSIAN, MSER).

Orientation: The estimated local gradients of the selected pairs are aggregated, so as to evaluate how much the keypoints are rotated. Pairs with symmetric receptive fields w.r.to the center are the most suitable for this purpose and should be selected.

Four feature detectors (Fast-Hessian, MSER, Harris and FAST) and two feature descriptors (SURF and FREAK) are illustrated before. The four detectors are considered the fastest detectors. In addition, these detectors have superior accuracy. Four combinations of these detectors and descriptors are used which are FastHessian-SURF, MSER-SURF, FAST-FREAK and Harris-FREAK. These extractors will be used to serve the matching algorithm between two images in the next section. The performance and the complexity are taken into consideration when these combinations are chosen. An example of each feature detector illustrated before is depicted in fig. 1.

III. Proposed Matching Approach

Basically, the conventional matching approaches aim to find the correspondences between features exist in a pair of image. These approaches depend mainly on finding the minimum distance between features (descriptors) in feature space as shown in (5), where D_{ij} is the similarity measure between feature i from the first image and feature j from the second image. X_{ij} is a matching between feature i and feature j, i.e. X_{ij} = 1 if feature i in the 1st image is mapped to feature j in the 2nd image and X_{ij} = 0 otherwise. Note that $X_{ij} \in \{0, 1\}$.

$$Min \;\; F = \sum_{\forall_{i,j}} D_{ij} \; X_{ij} \tag{5}$$

Limitations: The similarity measure between features deals with each feature individually rather than a group of features. Consequently, the minimum distance between features can be misleading in some cases and as a result the performance of the algorithm deteriorates. In other words, the minimum distance criterion has no objection for a feature to be wrongly matched as long as it successfully achieves the minimum distance objective.

We proposed a new matching algorithm called "Similarity-Topology Matching" in [7]. The proposed algorithm pays attention not only to the similarity between features but also to the spatial layout of every matched feature and its neighbors. A new term, describing the neighbourhood/ topological relations between every pair of features has been added $\alpha \sum_{\forall_{i,j,k,l}} X_{ij} \;\; X_{kl} \;\; P_{ij,kl}$. In addition, another term has been added to relax the constraints $\beta \; (Min(m,n) \; - \sum_{\forall_{i,j}} X_{ij}$ as shown below in (6).

$$Min \;\; F = \sum_{\forall_{i,j}} D_{ij} \; X_{ij} + \alpha \sum_{\forall_{i,j,k,l}} X_{ij} \; X_{kl} \; P_{ij,kl} + \beta \; (Min(m,n) \; - \sum_{\forall_{i,j}} X_{ij}) \tag{6}$$

Subject to:

$$\sum_{j=1}^{n} X_{ij} \;\; \leq \; 1 \qquad (a)$$

$$\sum_{i=1}^{m} X_{ij} \;\; \leq \; 1 \qquad (b)$$

The second term in (6) represents a penalty term over all pairs of features. $P_{ij,kl}$ is called a penalty matrix. It is used to penalize matching pairs of features X_{ij} in one image with corresponding pairs X_{kl} in the other image if they have different topologies. It is binary and of $(m \times n, m \times n)$ dimension; where m, n are the number of features in the first and the second images respectively. $P_{ij,kl} = 1$ if the features k, l in the second image have different topology when compared to features i, j in the first image. In other words, if any two features are neighbours to each other in the first image and matched to two features in the second image which are not neighbours to each other or vice versa. Hence a penalty term will be added to this matched pair.

(α) is called a topology coefficient. It indicates how much the matching algorithm depends on the topology between images and it will be adjusted according to the image type. In the experiments, (α) was chosen in a range from 0 to 0.1. The topology term has nearly no impact when the difference of similarities between the features is high.

(β) is called a threshold coefficient. It indicates how much the matching algorithm depends on the features matching threshold. It will be adjusted according to the image type. In the experiments, (β) was chosen in a range from 0 to 0.5.

Constraints: Constraint (a): There exists at most one in every column of x. Constraint (b): There exists at most one in every row of x. These two constraints ensure that every feature should match at most one feature.

Algorithm (1) gives a summary of the proposed local features matching algorithm, which depends not only on the similarity between features but also on the topological relations between them.

Algorithm 1 Similarity-Topology Matching

Input: A pair of images, topology coefficient (α), and threshold coefficient (β).

1) For every image:
   a) Detect local features (select strongest 100);
   b) Extract a descriptor for every feature;
2) For every feature (descriptor) in the 1st image: Calculate the similarity between it and all the features in the 2nd image;
3) Penalize any pair of features that matches to a pair of different topology;
4) Compute the objective function using (6) (features similarity and topological constraints);

Output: List of features correspondences.

IV. Experiments

Unlike other studies that aim to evaluate feature extractors in general, the main purpose of these experiments is to find a robust feature extractor that serves the matching approach described in section III. The experiments have been done using four different feature extractors: HARRIS-FREAK, HESSIAN-SURF, MSER-SURF, and FAST-FREAK. Columbia Object Image Library (COIL-100), has been used in the experiments [15]. COIL-100 is a database of color images which has 7200 images of 100 different objects (72 images per object). Ten objects of the aforementioned dataset have been chosen to perform the experiments. These objects with extra synthetic deformations such as rotation, partial occlusion and heavy noise have been used for this purpose. In addition, a duplication of the same object has been found in the same image with deformations, but one as a whole and one as parts to make the matching more challenging.

Evaluation criterion: For each pair of images, every interest point in image 1 is compared to all interest points in image 2 by comparing their descriptors four times. Every time with different feature extractor (HARRIS-FREAK, HESSIAN-SURF, MSER-SURF, and FAST-FREAK). The detection rate of the best N matches has been calculated to measure the performance. The detection rate (R) is defined as the ratio between the number of correct matches and the number of all possible matches [6].

A Receiver Operating Characteristic (ROC) based criterion has been used to show the detection rates versus the number of most similar matches allowed (N). The ROC curves are shown in table I and fig. 2.

$$\text{R} = \frac{\text{Number of Correct Matches}}{\text{Number of possible Matches}}$$

Table II depicts some experimental results. Each row in this table represents an instance. Each column represents a feature extractor. Lets take a closer look to the first

Table I
THE EXPERIMENTAL RESULTS SUMMARY

	Correct Matches	Possible Matches	Detection Rate
HARRIS-FREAK	704	1100	0.64
HESSIAN-SURF	660	1100	0.6
MSER-SURF	616	1100	0.56
FAST-FREAK	528	1100	0.48

experiment (row). In this experiment, the candidate object is subject to rotation and an exact but partitioned copy of the object is added to the image making the matching process more challenging. The total number of possible matches is 20. The HARRIS-FREAK, HESSIAN-SURF, MSER-SURF, and FAST-FREAK successfully match 14, 12, 12, and 10 features respectively.

From these experiments, the HARRIS-FREAK, HESSIAN-SURF, MSER-SURF, and FAST-FREAK are suitable to serve our matching approach. In addition, the HARRIS-FREAK is the best feature extractor that can be used prior to our matching approach to get more robust feature correspondence.

V. CONCLUSIONS

In this paper, a comparison between HARRIS-FREAK, HESSIAN-SURF, MSER-SURF, and FAST-FREAK is conducted to find an optimal feature detector and descriptor combination for image matching. Test images are synthesized to be tricky enough to challenge the matching process with every feature extractor. It is found that features detected using different detectors perform differently during the correspondence process. HARRIS-FREAK detected features are more capable of surviving during the matching process and results in a more robust feature correspondence.

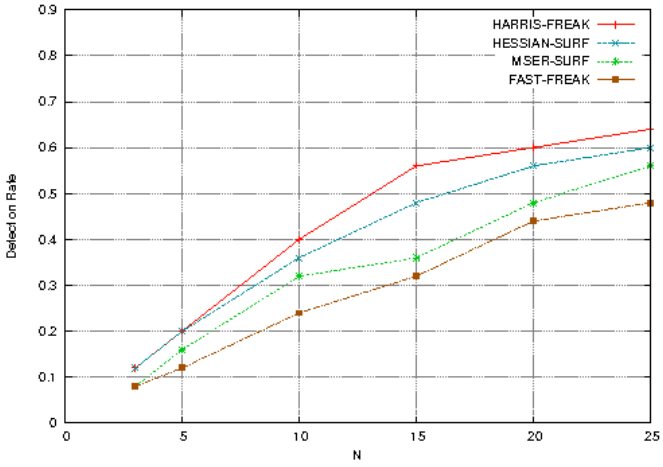


Figure 2. ROC curve for features matching experiments

Based on the experimental results, we can conclude that the robustness of feature detection does not imply the robustness of feature correspondence.

VI. FUTURE WORK

After the proof of concept of the aforementioned approach has been verified as well as finding the optimal features extractor to serve this approach, a lot of work remains to be done in order to generalize the local features matching approach and achieve high degree of robustness and computational efficiency. First, a preprocessing step is required to automatically evaluate the parameters values (alpha, beta) should be done. These values may depend on images size, number of extracted features in each image and images resolution. Second, an optimization of the algorithm to be more computationally efficient should be made without any loss in the algorithm accuracy as this algorithm should be used in real-time applications.

ACKNOWLEDGMENT

This research has been supported by the Ministry of Higher Education (MoHE) of Egypt through a Ph.D. fellowship. Our sincere thanks to Egypt-Japan University for Science and Technology (E-JUST) for guidance and support. I wish to express an extended appreciation to Eng. Islam ElShaarawy for his fruitful discussions and helpful suggestions.

REFERENCES

[1] B. Zitova and J. Flusser, "Image registration methods: a survey," *Image and vision computing*, vol. 21, no. 11, pp. 977–1000, 2003.

[2] R. Szeliski, *Computer vision: algorithms and applications*. Springer, 2011.

[3] K. Mikolajczyk and C. Schmid, "An affine invariant interest point detector," in *Computer VisionECCV 2002*. Springer, 2002, pp. 128–142.

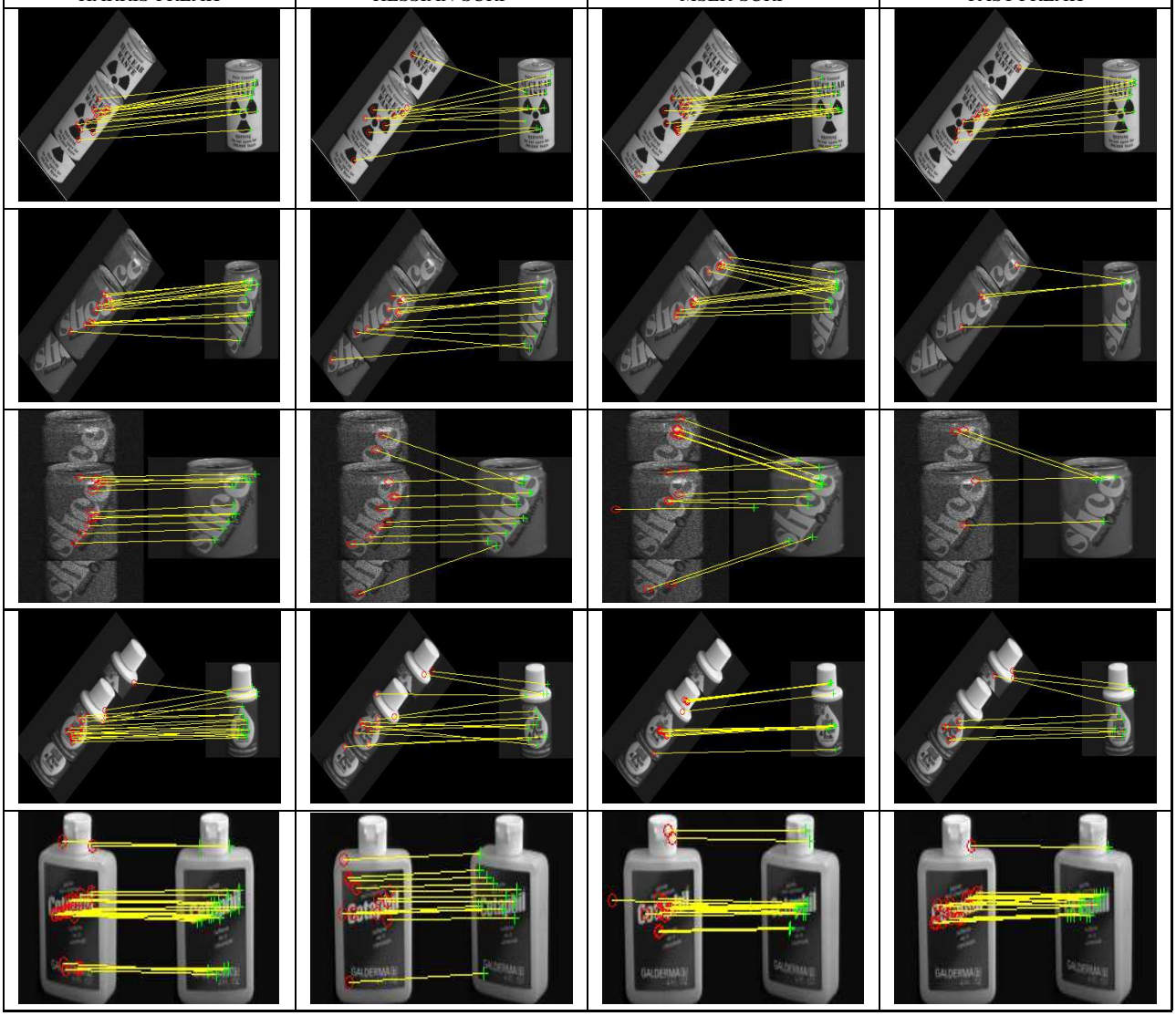
[4] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool, "A comparison of affine region detectors," *International journal of computer vision*, vol. 65, no. 1-2, pp. 43–72, 2005.

[5] C. Schmid, R. Mohr, and C. Bauckhage, "Evaluation of interest point detectors," *International Journal of computer vision*, vol. 37, no. 2, pp. 151–172, 2000.

[6] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 10, pp. 1615–1630, 2005.

[7] S. Y. El-Mashad and A. Shoukry, "A more robust feature correspondence for more accurate image recognition," in *Computer and Robot Vision (CRV), 2014 International Conference on*. IEEE, 2014.

Table II
SOME MATCHING EXAMPLES

HARRIS-FREAK	HESSIAN-SURF	MSER-SURF	FAST-FREAK
			

[8] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *Computer Vision–ECCV 2006*. Springer, 2006, pp. 404–417.

[9] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image and vision computing*, vol. 22, no. 10, pp. 761–767, 2004.

[10] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Computer Vision–ECCV 2006*. Springer, 2006, pp. 430–443.

[11] E. Rosten, R. Porter, and T. Drummond, "Faster and better: A machine learning approach to corner detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 1, pp. 105–119, 2010.

[12] C. Harris and M. Stephens, "A combined corner and edge detector." in *Alvey vision conference*, vol. 15. Manchester, UK, 1988, p. 50.

[13] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Computer vision and image understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[14] A. Alahi, R. Ortiz, and P. Vandergheynst, "Freak: Fast retina keypoint," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 510–517.

[15] S. Nayar, S. Nene, and H. Murase, "Columbia object image library (coil 100)," *Department of Comp. Science, Columbia University, Tech. Rep. CUCS-006-96*, 1996.