Comparison of Keypoint Detection Methods for Indoor and Outdoor Scene Recognition

Submitted: 2nd October 2017; accepted: 12th December 2017

Urszula Libal, Łukasz Łoziuk

DOI: 10.14313/JAMRIS_4-2017/32

Abstract:

We describe an experimental study, based on several million video scenes, of seven keypoint detection algorithms: BRISK, FAST, GFTT, HARRIS, MSER, ORB and STAR. It was observed that the probability distributions of selected keypoints are drastically different between indoor and outdoor environments for all algorithms analyzed. This paper presents a simple method for distinguishing between indoor and outdoor environments in a video sequence. The proposed method is based on the central location of keypoints in video frames. This has lead to a universally effective indoor/outdoor environment recognition method, and may prove to be a crucial step in the design of robotic control algorithms based on computer vision, especially for autonomous mobile robots.

Keywords: *machine vision, keypoint, UAVs, scene recognition*

1. Introduction

Correct determination of the sensed environment by a multitasking robot is crucial to the success of its control algorithms. For various kinds of environments, different sets of sensors and devices can be used (e.g., IR scanner for indoor and GPS for outdoor) to support autonomous control and other methods of determining the trajectory of the mobile robot. Most of the models and algorithms that work well in outdoor environments work poorly inside buildings [8]. The simplest method for distinguishing a robot environment is to divide it into interior and outdoor areas. It is clear that the weather conditions and the nature of the obstacles encountered by the robot on its path can be very different between indoor and outdoor environments [19]. Striving to create autonomous mobile robots forces us to supply them with control algorithms broad enough to be able to cope with a variety of different environments. The division of environments into indoor [3], [4] and outdoor [1] does not exhaust all the possibilities. However, this may be used as an initial recognition step, allowing the robot to switch between two dedicated control algorithms, better-adapted to open and closed spaces, respectively.

This paper focuses on demonstrating the differences between probabilities of keypoint locations for indoor and outdoor areas on the basis of characteristic points identified by the algorithms: BRISK, FAST, GFTT, HARRIS, MSER, ORB and STAR for video sequences. The difference in the central position of characteristic points (on video frames) between the interior and the exterior environment provides a strong basis for the development of a simple and effective method for distinguishing the environment surrounding the robot (based on video footage).

For land-based robots, many methods have been proposed for the detection and avoidance of obstacles. However, for aerial robots (such as unmanned aerial vehicles(UAVs)) there are still many challenges left to be solved. For UAVs, avoiding obstacles is more difficult, because they operate in 3D space, whereas for land-based robots, whose movement can be simplified to the 2D plane, obstacle detection is made simpler. Since UAVs have a limited carrying capacity, they are not able to be equipped with as many sensors as land-based robots in order to detect obstacles, for example, laser scanners [2]. Computer vision systems provide a good solution to this problem, because the video cameras are lightweight and energy efficient. In contrast to scanning sensors such as Lidar and sonar, cameras offer higher resolution and noise immunity. Land-based robots can also disengage the drive and stop for a long analysis of the environment. However, UAVs can not stay in fixed position for a long time, because it adversely affects their flight time.

In contrast to land-based robots, aerial robots operate in several different environments. In urban areas, the following types of neighborhoods, shown in Figs.1–4, may be encountered:

Indoors, inside buildings – Characterized by enclosed, often rectangular spaces containing obstacles, including mobile ones can appear from all directions. There may also be mirrors, and windows adding to the complexity object recognition systems

Fig. 1. Indoors

Streets – Streets are characterized by having many moving obstacles. However, the street plane can serve

as a reference to identify moving objects in the image, and thanks to such an assumption, many automotive obstacle detection algorithms can be used.

Fig. 2. Street

Urban canyons – Obstacles are usually located on the sides, e.g. buildings, trees, and the occasional obstacle in front. A robot must also avoid colliding with power lines and streetlights.

Fig. 3. Urban canyon

The space above the city – Compared to other environments, there are relatively few obstacles and they are usually static, e.g. radio antennas, tall buildings, chimneys.

Fig. 4. The space above the city

There are many specialized obstacle-detection algorithms used to find barriers in certain environments. The effectiveness of the algorithms strongly depends on the area where the mobile robot works. In each of the environments, other types of obstacles occur, so it is necessary to distinguish between the different environments and use dedicated control algorithms adapted to that specific environment.

2. Motivation

Proper scene recognition, whether it involves distinguishing if the robot is located inside or outside a building [19] to a more sophisticated environment analysis, is necessary if the robot loss could prove costly. Correct characterization of the environment allows for the use of dedicated control algorithms. The main objective is thus to avoid colliding with obstacles and damaging the robot or obstacle. It is not necessarily just about avoiding the destruction of the robot (e.g., UAV), but also to eliminate collisions with people, which is important because of the possibility of severe injury by such a flying robot.

In some papers, a completely different approach is presented, allowing the destruction of the robot. One example is training robots to cross a highway collision-free [18]. In this process, a certain number of robots must be discarded (damaged, or run over by vehicles), in order that the otherrobots can acquire the knowledge needed to learn.

3. Data

The analyzed data is a collection of more than 3,000,000 scenes shot from various movies. The videos have a resolution ranging from VGA (640x480) to Full HD (1920x1080). The video frame sequences were shot using various different cameras, with photodetector matrices based on either CCD or CMOS technologies. The videos were recorded at speeds of between 20 and 30 frames per second. They can be divided into two categories: recordings captured on premises (indoor) and those captured in open spaces (outdoor). The scenes are characterized by different lighting and camera movement dynamics.

Video sequences, shot in closed areas depict interior spaces such as museums or apartments. It should be noted that most of the movies filmed in a closed area are characterized by camera movements through areas such as corridors, where the orientation of the camera is close to that of the typical orientation of the human eye when walking through a room. Movies, shot in open areas, come from cameras mounted on drones (UAVs). The movies were recorded during drone flights over fields, roads, and mountains. We made sure that parts of the drone do not enter the frame All videos were recorded by cameras with lenses pointed in the direction of the drone flight. Of the 3,000,000 video frames, 1,200,000 video frames depict indoor areas, whilst 1,800,000 depict outdoor areas.

The diversity of the video material serves as a basis for constructing a simple and rapid method for determining and distinguishing between indoor and outdoor environments captured by cameras.

4. Keypoints Selection Algorithms

The following seven keypoint selection methods were tested: BRISK, FAST, GFTT, HARRIS, MSER, ORB and STAR. The keypoint pixels, detected by each of the listed algorithms, are shown on exemplary frames in

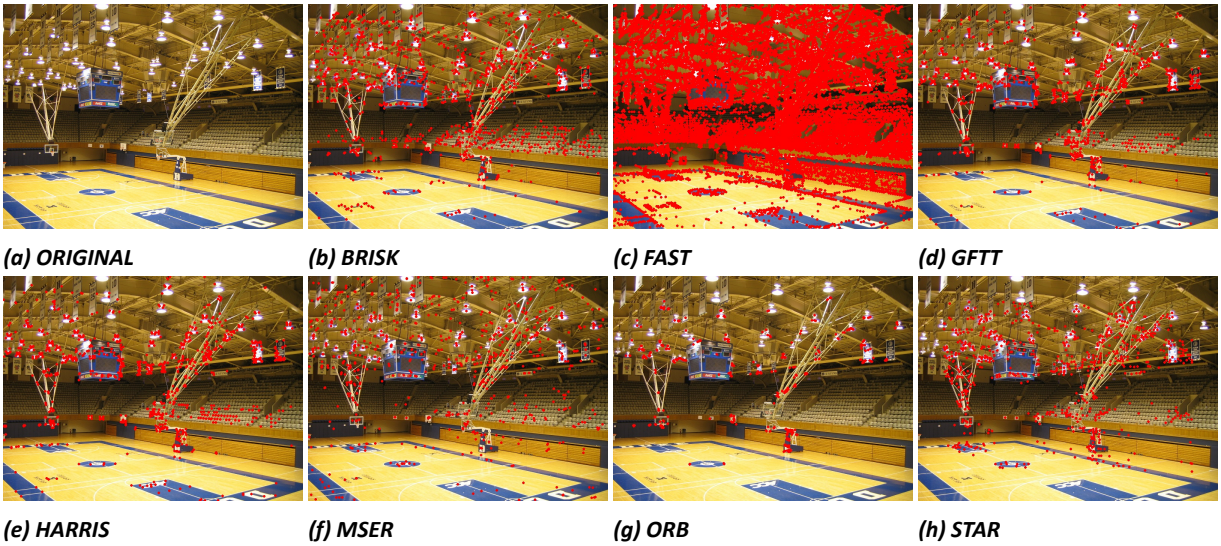


Fig. 5. Keypoints on exemplary frame for indoor environment

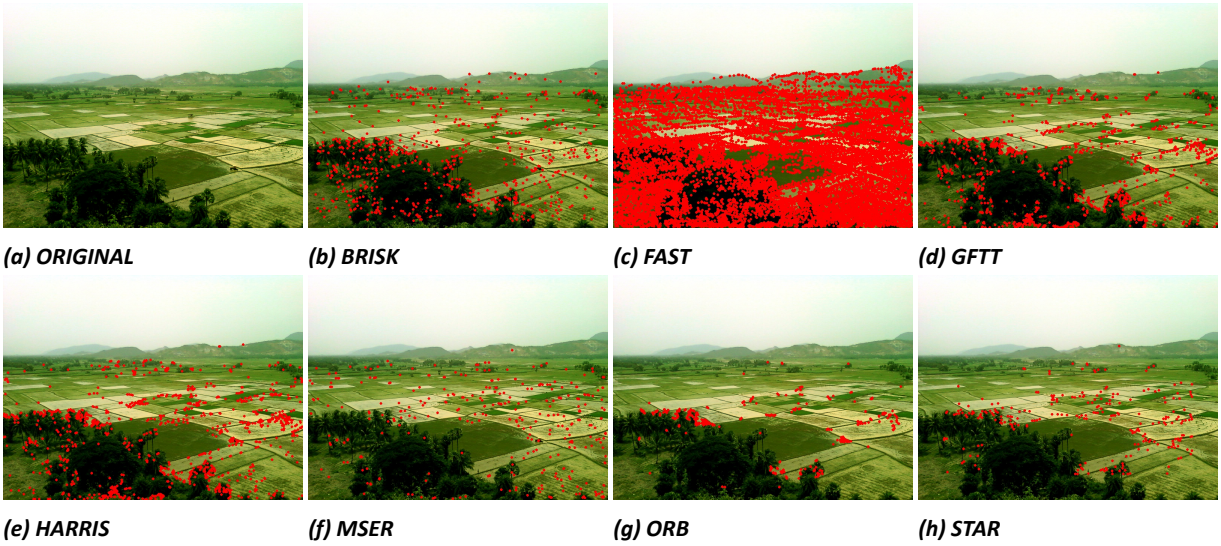


Fig. 6. Keypoints on exemplary frame for outdoor environment

Figures 5 and 6. The original pictures are presented in Figure 5a (indoor scene–a covered sports hall for basketball) and in Figure 6a (outdoor scene–a landscape with fields in the foreground and mountains in the background).

All algorithms compared in this paper were implemented using the Open CV [6] library. The software [17] contains all if the previously mentioned methods for keypoint selection in video sequences. The program associated with this paper is freely available via github [17]. A short description of each method is given below,

1) BRISK
The Binary Robust Invariant Scalable Keypoints method was proposed by Leutenegger *et al.* [9]. The algorithm is a modification of the BRIEF [5] (Binary Robust Independent Elementary Features) method, which uses binary strings for corner detection and can be computed by performing simple intensity difference tests. The output of every test is either zero or one, and the result is appended to the end of the string. In contrast to the BRIEF method, BRISK relies on a circular sampling pattern from which it computes brightness comparisons to form a binary descriptor string. The BRIEF method is not invariant to large, in-plane rotations. However the BRISK method handles simple in-plane rotations very well.
2) FAST
Features from Accelerated Segment Test is another corner detection method, published by Rosten and Drummond [11]. The method is the most computationally efficient algorithm among corner detectors, leading to extremely fast execution times.
3) GFTT
The Good Features To Track detector was presented by Shi and Tomasi [10]. This method is based on the calculation of the eigenvalues and eigenvectors of the deformation matrix. If both eigenvalues are small, then the feature does not vary much in any direction and is designated a flat region (bad feature). If one eigenvalue is

much larger than the other, then the feature varies mainly in one direction, i.e., an edge (bad feature). If both eigenvalues are large, then the feature varies significantly in both directionsi.e., a corner (good feature).

4) HARRIS
The HARRIS feature detector took its name from the surname of one of its originators, Harris and Stephens [12]. This algorithm is the oldest of those analyzed in this paper, having first been published in 1988. The HARRIS identifies similar regions among images by selecting and thresholding autocorrelated patches. A high positive response function value determines that it is a corner region, negative an edge region, and a small value determines a flat region. The detection method is similar to GFTT, instead of eigenvalues, the value of the response function is used.

5) MSER
The Maximally Stable Extremal Regions is a method that finds blobs in images, and was introduced by Matas et al [13, 14]. This algorithm is invariant to affine transformations of the image intensities.

6) ORB
The Oriented FAST and Rotated BRIEF is a fast feature detector, proposed by Rublee *et al.* [15]. It is based on both the visual descriptor BRIEF [5] (Binary Robust Independent Elementary Features) and the FAST [11] (Features from Accelerated Segment Test) keypoint detectors. The ORB algorithm is robust to in-plane rotations and it uses a nearest neighbor search instead of random sampling.

7) STAR
The CenSurE (Center Surround Extrema) feature detector, published by Agrawal, Konolige and Blas [16], has been implemented in the Open CV library [6], where it got designated a new name—the STAR—and some minor changes were applied, e.g., circular shapes were replaced with approximations. It is designed as a multiscale detector with masks of different sizes.

5. Keypoint Distribution

5.1. Location

The location of central keypoints is measured with the median. For every set of keypoints

$$kp = \{(x_1, y_1),\ (x_2, y_2),\ \ldots,\ (x_n, y_n)\}, \tag{1}$$

detected on a fixed frame, medians for the vertical and the horizontal coordinates are calculatedas

$$M_x(kp) = \text{median}\,(x_i), \tag{2}$$

$$M_y(kp) = \text{median}\,(y_i). \tag{3}$$

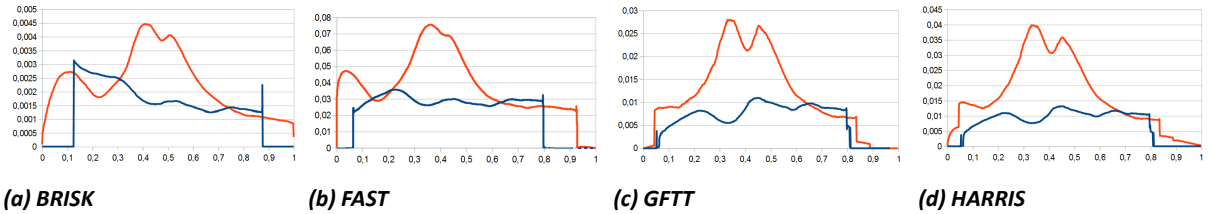
The median is used instead of the mean value, because the distributions for all analyzed algorithms are

Tab. 1. Average characteristics per frame for indoor scenes

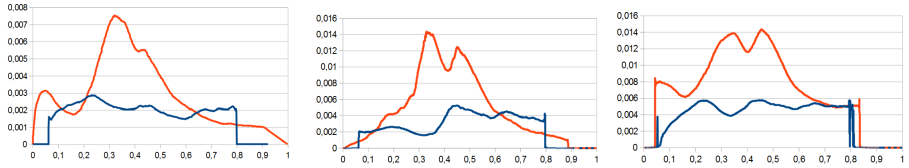
INDOOR						
Algorithm	Keypoint Center		Keypoint St Dev		Execution Time [s]	No. of Keypoints
	M_x^I	M_y^I	S_x^I	S_y^I		
BRISK	0.410	0.691	0.299	0.153	0.466	435.2
FAST	0.445	0.666	0.331	0.189	0.003	5824.8
GFTT	0.446	0.692	0.308	0.164	0.019	904.0
HARRIS	0.405	0.695	0.292	0.139	0.018	507.9
MSER	0.407	0.691	0.330	0.186	0.104	285.5
ORB	0.424	0.670	0.256	0.122	0.009	498.8
STAR	0.411	0.691	0.270	0.132	0.026	256.6

Tab. 2. Average characteristics per frame for outdoor scenes

OUTDOOR						
Algorithm	Keypoint Center		Keypoint St Dev		Execution Time [s]	No. of Keypoints
	M_x^O	M_y^O	S_x^O	S_y^O		
BRISK	0.464	0.776	0.415	0.102	0.526	4053.2
FAST	0.474	0.771	0.437	0.105	0.022	51705.7
GFTT	0.448	0.769	0.398	0.095	0.079	1000.0
HARRIS	0.451	0.770	0.396	0.095	0.072	932.2
MSER	0.494	0.786	0.443	0.110	0.352	1497.1
ORB	0.399	0.770	0.353	0.081	0.045	499.3
STAR	0.449	0.783	0.379	0.093	0.137	1517.1

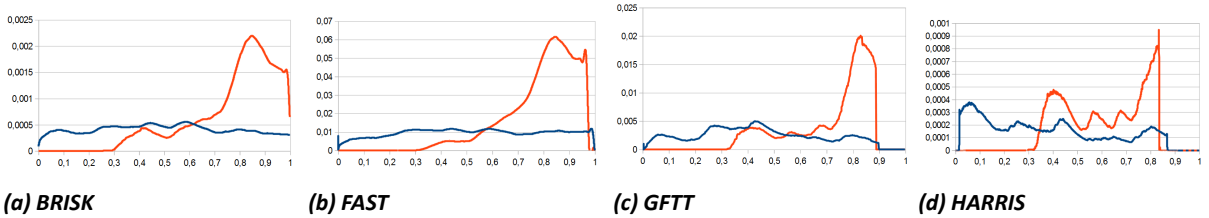


(a) BRISK (b) FAST (c) GFTT (d) HARRIS

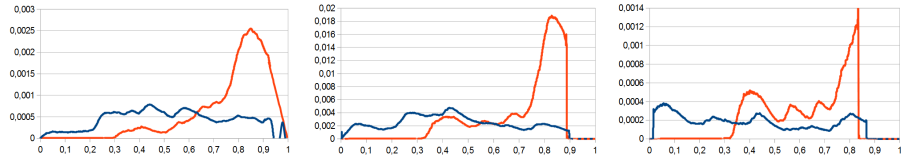


(e) MSER (f) ORB (g) STAR

Fig. 7. Marginal histograms of keypoint position (x,y) for indoor scenes. Average keypoint occurrence by frame for every relative pixel position (x,y): x – blue line, y – red line



(a) BRISK (b) FAST (c) GFTT (d) HARRIS



(e) MSER (f) ORB (g) STAR

Fig. 8. Marginal histograms of keypoint position (x,y) for outdoor scenes. Average keypoint occurrence by frame for every relative pixel position (x,y): x - blue line, y - red line.

skewed, or have more than one mode (see Figures 7-8). In such situations, the mean value does not point to a central location and so the median is more accurate.

For one type of environment, the vertical coordinates M_y of central keypoints take values on the same level for all algorithms. However, the vertical median M_y values are significantly different for indoor and outdoor video sequences: higher relative pixel position for outdoor and lower for indoor. High relative pixel positions are obtained for pixels placed in the lower section of frame. For every frame, the pixel in the upper left corner is the origin, (0,0). A relative scale is used, because videos of a wide range of resolutions were analyzed.

5.2. Dispersion

The dispersion of keypoints $kp = \{(x_i, y_i)\}_{i=1}^{n}$ on a frame is measured by the standard deviation of the median:

$$S_x(kp) = \frac{1}{n}\sum_{i=1}^{n}(x_i - M_x)^2, \tag{4}$$

$$S_y(kp) = \frac{1}{n}\sum_{i=1}^{n}(y_i - M_y)^2. \tag{5}$$

The standard deviations (horizontal, S_x, and vertical, S_y) show the dispersion from central points, obtained by the median (M_x, M_y). The lowest deviation is observed for the y-coordinate for outdoor video sequences. This means that the y-coordinates of keypoints for outdoor scenes are highly concentrated on the median indicated by M_y. The concentration of keypoints in the lower part of the video frame is characteristic of outdoor lanscapes. This attribute is shown in Figures 6b-6h and also in Figure 8.

6. Algorithm Performance

6.1. Number of Keypoints per Frame

The average numbers of keypoints for all executed algorithms are presented in Table 1 (for indoor) and Table 2 (for outdoor). The largest number of keypoints was detected by the FAST algorithm, for both environment types. This can also be observed in Figures 5c and 6c. The large number of selected points by the FAST algorithm coincides with the shortest average computation time.

6.2. Execution Time per Frame

As mentioned earlier, the FAST algorithm needed the shortest average time to calculate keypoints. The

second fastest algorithm is the ORB (Oriented FAST and Rotated BRIEF) algorithm. HARRIS and GFTT perform calculations in a comparably short time. The BRISK algorithm proved to be the slowest procedure. All execution times are shown in Tables 1 and 2.

7. Classification Problem for Scene Recognition

Scene recognition can be defined as a two-class classification problem, with class 1 (indoor) and class 2 (outdoor). The significant differences for keypoints distributions, shown in this paper, should enable the discrimination of both of these classes. The central keypoint locations are placed in lower sections on frames for videos taken outside, and in central sections for videos taken inside. Most of the selected keypoints for outdoor landscapes occur in the bottom half of frame. For indoor scenes, keypoints can be found on upper part of frame as well. This property brings hope that the classes (indoor and outdoor) are distinguishable.

We performed the classification based on the following decision rule:

$$\Psi(kp) = \begin{cases} I \text{ (indoor), if} \\ |M_y(kp) - M_y^I| \leq |M_y(kp) - M_y^O|, \\ O \text{ (outdoor), if} \\ |M_y(kp) - M_y^I| > |M_y(kp) - M_y^O|. \end{cases} \quad (6)$$

A frame represented by a set of keypoints kp is classified to class $I-indoor$, when the median vertical pixel position $M_y(kp)$ of keypoints detected in the frame is closer to the median vertical pixel position M_y^I of keypoints for indoor frames (see Table 1). It is classified to class $O-outdoor$, when the median $M_y(kp)$ is closer to the median M_y^O, calculated for outdoor frames (see Table 2). Without loss of generality, we present the rule for the fixed case $M_y^I < M_y^O$ as follows:

$$\Psi(kp) = \begin{cases} I \text{ (indoor), if } M_y(kp) \leq \lambda_{CK}, \\ O \text{ (outdoor), if } M_y(kp) > \lambda_{CK}, \end{cases} \quad (7)$$

where the threshold distinguishing between the two classes is given by

$$\lambda_{CK} = (M_y^O + M_y^I)/2. \quad (8)$$

The proposed classification method, based on central keypoint position, was compared to the Naive Bayes classifier [20,21], since the distributions in both classes are known (estimated by histograms to be precise). Table 3 contains the results for the two fastest keypoint selection methods: FAST and ORB, and one with medium computation time: STAR.

The proposed classifier has surprisingly good performance, attributed to both the simplicity of the rule and the reduction of a high dimensional problem to one dimension (i.e., only vertical components). For the algorithms FAST and ORB, the results are comparable to the Naive Bayes classifier. The decision areas for both classifiers in those cases were similar, as indicated by thresholds λ_{CK} and λ_{NB} (see Figure 9). For

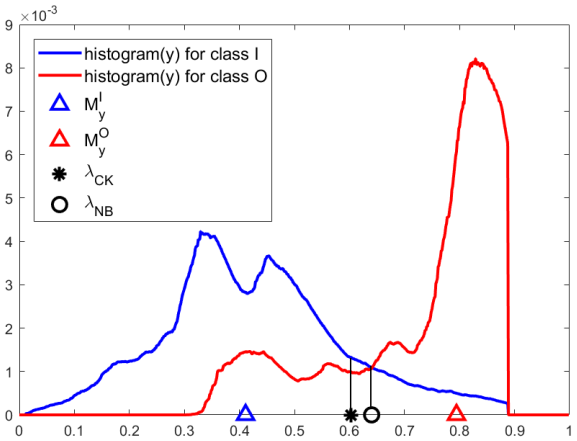


Fig. 9. Example of threshold λ_{CK} for vertical components y of keypoints (x, y) detected with the ORB algorithm.

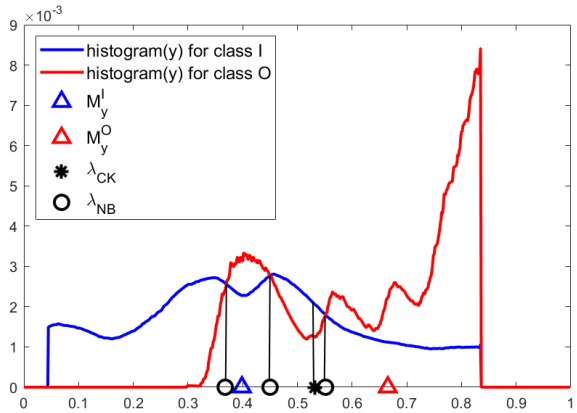


Fig. 10. Example of threshold λ_{CK} for vertical components y of keypoints (x, y) detected with the STAR algorithm.

the STAR algorithm, the results differ between the two classification methods, because for the Naive Bayes classifier, four decision areas were created, whereas for the central keypoint method, only two are created (see the border points λ_{NB} in Figure 10).

8. Conclusions

The results presented in Figures 7 and 8 cover the full collection of 3 million scenes, divided into two groups: indoor and outdoor. For closed areas, all investigated algorithms: BRISK, FAST, GFTT, HARRIS, MSER, ORB and STAR produced quite similar distributions of keypoints location in an average video frame (see Figure 7). The same was observed for open areas (see Figure 8). This means that the distribution of keypoints for each environment (indoor and outdoor) is practically independent from the algorithm that has been used. The method for distinguishing between indoor and outdoor environments, on the basis of characteristic points obtained from video sequences, may be considered as objective and reliable, regardless of the algorithm used for determining the keypoints. Statistical analysis showed that the probability density function (as estimated by histograms–see Figures 7-

Tab. 3. Classification performance of proposed central keypoint method in comparison to Naive Bayes classifier.

Algorithm	Central Keypoint Method			Naive Bayes		
	precision	recall	F-score	precision	recall	F-score
FAST	0.8954	0.7837	0.8358	0.9086	0.7715	0.8345
ORB	0.7918	0.8588	0.8240	0.7647	0.8916	0.8233
STAR	0.6765	0.7367	0.7053	0.8327	0.6243	0.7136

8) of keypoints for an average video frame captured inside buildings has a maximum in the middle of the image. Characteristic points appear in random places on the whole picture, but most often they occur in the middle of the image. This is connected to the fact that the person filming usually directs the optical center of the camera in the direction of a large number of characteristic points, eg., at the end of the corridor or at objects such as sculptures, paintings, faces, etc. The probability density function of the location of the average characteristic points in open areas has a maximum at the bottom of the image. This is connected to the fact that for open spaces, most of the upper part of frame is the sky. The upper part of video frames usually has a small number of characteristic points, in contrast to the lower part. The bottom part of the image contains the land and buildings and the number of characteristic points for all algorithms is significantly higher–assuming a horizontal orientation of the camera.

For all of the algorithms analyzed, the median vertical M_y calculated for outdoor environments is higher, with respect to the M_y for indoor environments, as would be expected. In contrast, there is no significant difference in the median position calculated horizontally M_x by the different algorithms for determining the characteristic points. Therefore, the distinction between indoor and outdoor environments should be based on the vertical position of the characteristic points centers (M_x, M_y): in the middle of the video frame for indoor, at the bottom of the video frame for outdoor–this can be explained by the camera positioning on the horizon line.

In addition to the distinction between scenes, an important aspect is also the calculation speed. In real-life applications, such distinction has to be done in real time, and therefore, the comparison of the average computation time needed to determine the keypoints for different algorithms was performed. The FAST algorithm was shown to be the fastest, and gives the highest number of keypoints (see Tables 1-2). The second fastest algorithm was the ORB, which is a modification of the FAST method. In contrast, the slowest methods include the MSER and BRISK algorithms.

9. Future Work

Probability distributions for indoor and outdoor environments describing keypoints location–such as medians M_x, M_y (central pixel position), and standard deviations S_x, S_y were calculated for every frame. This leads to outlying central keypoints positions if the number of keypoints is noticeably smaller than for the average frame. To avoid this problem, one can analyze not only one frame in every step, but a subsequence of frames. This approach is often applied for moving object analyses (see, e.g., Foresti and Micheloni [7]).

The results of this paper are reduced only to the application of the average occurrence distinguishing between indoor and outdoor scenes. It would be very interesting to continue the this line of research, and investigate more kinds of scenes, e.g., those mentioned in the introduction.

Software

Open source software [17], together with a graphical user interface, written in python, is attached to this paper. The software was created and used by authors for keypoint selection in video sequences.

REFERENCES

[1] S. Srinivasan, L. Kanal, "Qualitative landmark recognition using visual cues", *Pattern Recognition Letters*, vol. 18, no.11-13, 1997, 1405–1414. DOI: 10.1016/S0167-8655(97)00142-6.

[2] M. Bosse, R. Zlot, "Keypoint design and evaluation for place recognition in 2D lidar maps", *Robotics and Autonomous Systems*, vol. 57, no.12, 2009, 1211–1224. DOI: 10.1016/j.robot.2009.07.009.

[3] P. Espinace, T. Kollar, N. Roy, A. Soto, "Indoor scene recognition by a mobile robot through adaptive object detection", *Robotics and Autonomous Systems*, vol. 61, no. 9, 2013, 932–947. DOI: 10.1016/j.robot.2013.05.002.

[4] A. Vailaya, A.K. Jain, H. Zhang, "On image classification: city images vs. landscapes", *Pattern Recognition*, vol. 31, no. 12, 1998, 1921-1935. DOI: 10.1016/S0031-3203(98)00079-X.

[5] M. Calonder, V. Lepetit, C. Strecha, P. Fua, "BRIEF: Binary Robust Independent Elementary Features". In: *European Conference on Computer Vision* (ECCV), Heraklion, Crete, Greece, September 5-11, Proceedings - Part IV, 778–792, 2010. DOI: 10.1007/978-3-642-15561-1_56.

[6] Itseez. Open CV Library. `http://docs.opencv.org/modules/features2d/doc/common_interfaces_of_feature_detectors.html`, Accessed: 2017-04-30.

[7] G.L. Foresti, C. Micheloni, "Real-time video-surveillance by an active camera", *Ottavo Convegno Associazione Italiana Intelligenza Artifi-*

*ciale (AI*IA)– Workshop sulla Percezione e Visione nelle Macchine*, Universita di Siena, 2002. DOI: 10.1.1.19.4723.

[8] A. Quattoni, A. Torralba, "Recognizing indoor scenes". In: I*EEE Conference on Computer Vision and Pattern Recognition*, IEEE Computer Society, Los Alamitos, CA, USA, 413–420, 2009. DOI: 10.1109/CVPRW.2009.5206537.

[9] S. Leutenegger, M. Chli, R.Y. Siegwart, "BRISK: Binary Robust invariant scalable keypoints". In: *IEEE International Conference on Computer Vision* (ICCV), 2548–2555, 2011. DOI: 10.1109/ICCV.2011.6126542.

[10] J. Shi, C. Tomasi, "Good features to track", *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (CVPR),1994, 593–600. DOI: 10.1109/CVPR.1994.323794.

[11] E. Rosten, T. Drummond, "Fusing Points and Lines for High Performance Tracking", *IEEE International Conference on Computer Vision* (ICCV), 17-20 October 2005, Beijing, China, 1508–1515, 2005. DOI: 10.1.1.451.4631.

[12] C. Harris, M. Stephens, "A Combined Corner and Edge Detector". In: *Alvey Vision Conference*, (AVC), Manchester, UK, September, 1-6, 1988. DOI: 10.5244/C.2.23

[13] J. Matas, O. Chum, M. Urban, T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions", *Image Vision Computing*, vol. 22, no. 10, 2004, 761-767. DOI: 10.1016/j.imavis.2004.02.006.

[14] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, L. Van Gool, "A Comparison of Affine Region Detectors", *International Journal of Computer Vision*, vol. 65, no. 1– 2, 2005, 43–72. DOI: 10.1007/s11263-005-3848-X.

[15] E. Rublee, V. Rabaud, K. Konolige, G. Bradski, "ORB: An efficient alternative to SIFT or SURF", *IEEE International Conference on Computer Vision*, IEEE Computer Society, Los Alamitos, CA, USA, 2564– 2571, 2011. DOI: 10.1109/ICCV.2011.6126544.

[16] M. Agrawal, K. Konolige, M.R. Blas, "CenSurE: Center Surround Extremas for Realtime Feature Detection and Matching", *European Conference on Computer Vision (ECCV)*, Marseille, France, October 12–18, 2008, Proceedings – Part IV, 102–115, 2008. DOI: 10.1007/978-3-540-88693-8_8.

[17] Ł. Łoziuk. Video Analysis Algorithms – Software. https://github.com/bitcoinsoftware/video AlgorithmsAnalysis, Accessed: 2017- 08-30.

[18] A. Lawniczak, B. Di Stefano, J. Ernst, "Stochastic Model of Cognitive Agents Learning to Cross a Highway", *Stochastic Models, Statistics and Their Applications*, Springer Proceedings in Mathematics and Statistics, 122:319–326, 2015. DOI: 10.1007/978-3-319-13881-7_35.

[19] D. Lopez De Luise, G. Barrera, S. Franklin, "Robot Localization Using Consciousness", *Journal of Pattern Recognition Research*, 6(1), 2011, 96–119. DOI: 10.13176/11.257.

[20] R.O. Duda, P.E. Hart, D.G. Stork, *Pattern Classification*, 2nd Edition, Wiley-Interscience, 2000.

[21] C.M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006. DOI: 10.1117/1.2819119.