A learning based feature point detector

A. Verichev[1]

[1]Samara National Research University, 34 Moskovskoe Shosse, 443086, Samara, Russia

Abstract

We propose a learning-based image feature points detector. Instead of giving an explicit definition for feature point we apply the methods of machine learning to infer it inductively using a representative training set. This allows for a flexible tuning of the proposed detector to a specific problem that is described by a training set of desired responses. To increase feature points' repeatability and robustness to various image transformations the feature space of the learning algorithm includes raw image moments and image moment invariants. Experiments demonstrate high flexibility in tuning the detector to a specific task, acceptable repeatability of the feature points and robustness to various image transformations.

Keywords: image feature points; image feature points detector; image moments; image moment invariants; machine learning

1. Introduction

Feature point is a piece of information which is relevant to solving a certain application-related computational task. Feature points find their use in numerous applications such as image stitching, stereo correspondence, locating and tracking of a moving object, object detection and recognition, and others [1, 2]. The ubiquitous usage of feature points is a direct consequence of their properties [3]:

- *Repeatability*: Given two images of the same object or scene, a high percentage of the features detected on the scene visible in both images should be found in both images.
- *Informativeness*: The intensity patterns underlying the detected features should show a lot of variation.
- *Locality*: The features should be local, so as to reduce the probability of occlusion and to allow simple model approximations of the geometric and photometric deformations between two images.
- *Quantity*: The number of detected features should be sufficiently large, such that a reasonable number of features are detected even on small objects.
- *Accuracy*: The detected features should be accurately localized.
- *Efficiency*: The detection of features in a new image should allow for time-critical applications.

Algorithms and methods that detect image feature points by making local decisions are called feature points detector. An abundance of image feature points detectors is known, most of which are based on a certain criterion - a heuristics that implicitly defines what a term feature point constitutes. Generally these heuristics can be classified into three categories [4]:

- Gradient-based: A majority of image feature points detectors is based on computation of gradients of intensity function, for example Förstner [5], Harris [6], Shi-Tomasi [7].
- Template-based: Feature points are found by comparing the intensity of surrounding pixels with that of center pixels which is governed by some template. The well-known template-based detectors are SUSAN [8], FAST [9], AGAST [10].
- Contour-based: A feature point is defined as the intersecting point of two adjacent edge lines, examples are DoG-curve [11], ANDD [12].

However, formulating a heuristics for an image feature points detector requires a well-formed application-dependent definition of the term feature point, which in turn requires some level of expertise in the application domain. Moreover, a strictly stated criterion, although sharpening performance, diminishes its flexibility to adjust to a particular problem, which renders all the possible usages outside the destined application moot.

The goal of this work is to dispense with defining the term feature point altogether and focus on the properties we wish the feature points to possess. With that goal in mind we resort to machine learning methods. Image raw moments and image moment invariants are used along with some other local characteristics of image points to form a feature space of a learning algorithm. The detector is trained to solve a specific problem on a relevant and carefully collected training set. This effectively defines the term feature point implicitly, since it's inductively inferred from the training examples.

The proposed method is described in full detail in section 2, along with the learning algorithm, its feature space and the procedures for collecting training and test sets. Evaluation criteria of a trained detector's performance and the results of experimental evaluation are described in section 3. We conclude with a discussion of these results.

2. Proposed method

The proposed learning-based feature points detector is based on the idea of transforming detection task into a classification task as suggested in [13], which boils down to training the detector's classifier on a set of the desired responses.

2.1. Feature space

The first step towards constructing our detector is to define the classifier's feature space, which is an $\mathbb{R}^{15}$ vector space. Each pixel of an image $I[x, y]$ is mapped to a certain vector in this feature space using a locally defined operator $P^{9\times9} \to \mathbb{R}^{15}$, where $P = \{n: 0 \le n < 256\}$ is a set of intensities of a grayscale image. The features of the feature space are described below.

The first two features are standard deviation of a standardized local area, ϕ_1, and standard deviation divided by the norm of the local area, ϕ_2:

$$\phi_1 = \sqrt{\frac{1}{80}\sum_{i=-4}^{4}\sum_{j=-4}^{4}\frac{1}{n^2}(I[x+i, y+j] - \bar{I})^2}, \qquad \phi_2 = \frac{\phi_1}{n}, \tag{1}$$

where norm n and local mean $\bar{I}$ are defined:

$$\bar{I} = \frac{1}{81}\sum_{i=-4}^{4}\sum_{j=-4}^{4} I[x+i, y+j],$$

$$n = \sqrt{\sum_{i=-4}^{4}\sum_{j=-4}^{4}(I[x+i, y+j])^2}.$$

The use of these features is motivated by their sensitivity to monotonous and textured areas.

The next four features are chosen to be central image moments of a local image area: $\phi_{t+3} = \mu_{tt}$, $0 \le t \le 3$. The central moments are defined [14]:

$$\mu_{ij} = \sum_{k=-4}^{4}\sum_{l=-4}^{4} k^i \cdot l^j \cdot \frac{1}{81} I[x+k, y+l]. \tag{2}$$

To induce invariance to rotation transformations the following Hu invariant image moments and Flusser moments are used [15, 16]:

$$\begin{aligned}
\phi_7 &= \mu_{20} + \mu_{02},\\
\phi_8 &= (\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2,\\
\phi_9 &= (\mu_{30} - 3\mu_{12})^2 + (3\mu_{21} - \mu_{03})^2,\\
\phi_{10} &= (\mu_{30} + \mu_{12})^2 + (\mu_{21} + \mu_{03})^2,\\
\phi_{11} &= (\mu_{30} - 3\mu_{12})(\mu_{30} + \mu_{12})[(\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2] + (3\mu_{21} - \mu_{03})(\mu_{21} + \mu_{03})[3(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2],\\
\phi_{12} &= (\mu_{20} - \mu_{02})[(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2] + 4(\mu_{30} + \mu_{12})(\mu_{21} + \mu_{03}),\\
\phi_{13} &= (3\mu_{21} - \mu_{03})(\mu_{30} + \mu_{12})[(\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2] - (\mu_{30} - 3\mu_{12})(\mu_{21} + \mu_{03})[3(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2],\\
\phi_{14} &= \mu_{11}[(\mu_{30} + \mu_{12})^2 - (\mu_{03} + \mu_{21})^2] - (\mu_{20} - \mu_{02})(\mu_{30} + \mu_{12})(\mu_{03} + \mu_{21}).
\end{aligned} \tag{3}$$

Moments calculation is an intensive computational task that requires of a lot of operations. To reduce the number of arithmetical operations we apply the recursive method of moments calculation based on the use of integer factorial polynomials [16].

The last feature that characterizes misalignment of centre of local area and its centre of mass is defined:

$$\phi_{15} = \sqrt{(x_c - x)^2 + (y_c - y)^2}, \tag{4}$$

where $x_c = \mu_{10}/\mu_{00}$ and $y_c = \mu_{01}/\mu_{00}$.

The set of the features ϕ_i, $1 \le i \le 15$, defined by (1) - (4), with a usual addition and scalar multiplication operations form the feature vector space.

2.2. Tuning the detector

2.2.1. Collecting a training set

Tuning the detector requires a training set that consists of the desired detector's responses. Depending on the application there are various ways the set can be obtained:

- manually, involving experts of the domain;
- automatically, using well-known feature points detectors such as Harris or Canny;
- combining the two.

In case there is a human involvement of any kind it is inevitable for a training set to contain a so called training noise [17]. Besides, in a typical scenario a number of feature points is small compared to the other points. To alleviate these negative effects the neighbouring points of the feature points can be considered feature points as well.

Provided an application requires high level of robustness to certain transformations, a training set can be enlarged to contain the so called virtual examples [18]. To this end every image used to form a training set is transformed according to some transformation. Since the parameters of that transformation are known, the elements of the original image can be mapped onto the transformed image, which makes it possible to extract feature vectors of the points of the transformed image that correspond to the feature points of the original image. These new feature vectors are the virtual examples that convey information about various effects the transformation have on the feature vectors.

2.2.2. Training a classifier

With a training set at hand we can pose and solve a supervised learning problem. Since the number of the feature vectors in a training set is typically quite large we chose to apply nonparametric probability density estimation approach. Let $D = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$ denote training set, where $\boldsymbol{x}_i$ is a feature vector, y_i is its label, $y_i \in \{C_1, C_2\}$. C_1 corresponds to feature points and C_2 corresponds to the other points. Then, an estimation of conditional probability density function is defined as follows:

$$\hat{p}(\boldsymbol{x}|C_i) \propto \sum_{j=1}^{N}[y_j = C_i]\, K\left(\frac{\|x - x_j\|}{h}\right), \tag{5}$$

where K is a kernel function, h is kernel's width parameter. By the Bayes' Theorem:

$$\hat{p}(C_i|\boldsymbol{x}) \propto \hat{p}(\boldsymbol{x}|C_i) \cdot \hat{\pi}_i, \tag{6}$$

where $\hat{\pi}_i$ is an estimate of prior probability of i^{th} class:

$$\hat{\pi}_i = \frac{1}{N}\sum_{j=1}^{N}[y_j = C_i]. \tag{7}$$

Define a *characteristic function* of a feature point $l(\boldsymbol{x})$:

$$l(\boldsymbol{x}) = ln\big(\hat{p}(C_1|\boldsymbol{x})\big) - ln\big(\hat{p}(C_2|\boldsymbol{x})\big). \tag{8}$$

In order to smooth the detector's response we filter the characteristic function $l(\boldsymbol{x})$ using a local peak filter. The peak filter suppresses non-maximal values in a local 3×3 neighbourhood of the point $\boldsymbol{x}$:

$$\tilde{l}(\boldsymbol{x}) = \begin{cases} l(\boldsymbol{x}),\ l(\boldsymbol{x}) > l(\boldsymbol{g}) + \delta \quad \forall \boldsymbol{g} \in W \setminus \{\boldsymbol{x}\} \\ 0,\ otherwise \end{cases}, \tag{9}$$

where W is a set of all feature vectors from the local neighbourhood, $\delta$is some threshold.

From (8) and (9) we infer the decision rule:

$$y(\boldsymbol{x}) = \begin{cases} C_1,\ \tilde{l}(\boldsymbol{x}) > t = ln\left(\frac{\hat{\pi}_2}{\hat{\pi}_1}\right) \\ C_2,\ otherwise \end{cases}. \tag{10}$$

3. Experimental evaluation

3.1. Experimental setup

To experimentally evaluate the proposed detector we built a set of images. The set contains a series of 10 overlapping images of 6 different scenes, 60 images in total. Figure 1 shows three images of one of these scenes. Each of the 6 groups of images was split in relation 8:2 to form training set D and test set C, respectively. We chose to use Harris [6] corner detector to detect feature points. The training set was enlarged by the virtual examples as described in section 2.2.1 and the transformations that were applied are described in section 3.3.

Fig. 1. Example images of a scene.

3.2. Evaluation of training accuracy

Let $V = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$ be training or test set. The primary criterion of detector's performance on the set V is its *accuracy*:

$$A(V) = \frac{1}{N}\sum_{i=1}^{N}[y(\boldsymbol{x}_i) = y_i]. \tag{11}$$

Besides the accuracy two more criteria are used: *precision P* and *recall R* [19]. Precision is the fraction of relevant instances over the retrieved instances, while recall is the fraction of relevant instances among the retrieved ones over the total number of relevant instances in the set.

Let FP, FN and TP denote false positives, false negatives and true positives, respectively. Then,

$$\begin{aligned} FP(V) &= \sum_{i=1}^{N}[y(\boldsymbol{x}_i) = C_1] \cdot [y_i = C_2\ ], \\ FN(V) &= \sum_{i=1}^{N}[y(\boldsymbol{x}_i) = C_2] \cdot [y_i = C_1\ ], \\ TP(V) &= \sum_{i=1}^{N}[y(\boldsymbol{x}_i) = y_i]. \end{aligned} \tag{12}$$

Precision and recall are defined:

$$\begin{aligned} P(V) &= \frac{TP(V)}{TP(V)+FP(V)}, \\ R(V) &= \frac{TP(V)}{TP(V)+FN(V)}. \end{aligned} \tag{13}$$

The proposed detector was first trained on the training set. Accuracy, precision and recall were evaluated on the training set D and test set C. The results are shown in table 1. Taking into account a fairly large size of the sets, the data suggests an adequate quality of training.

3.3. Repeatability evaluation of the detector

As mentioned in introduction, repeatability is one of the most important properties of the feature points. Along with its importance, repeatability allows for an objective and qualitative evaluation. Hence, we used repeatability to evaluate the performance of the proposed detector.

Table 1. Accuracy, precision and recall of the trained detector .

	$A(D)$	$P(D)$	$R(D)$
Training set, D	0.997	0.905	0.960
Test set, C	0.9766	0.730	0.580

The procedure for repeatability evaluation is outlined below.

- An original image is used to find a set of feature points P_o.
- The original image is transformed by one of the transformations (cf. the next list below).
- The transformed image is used to find a set of feature points P_t.
- Since parameters of the transformation are known, coordinates of the points P_o of the original image can be mapped onto the transformed image. Thus, the points in the set P_o are mapped onto the transformed image, forming a set P_m.
- The sets P_m and P_t are matched. Two points $a \in P_m$ and $b \in P_t$ are considered equal if $a \in V_\varepsilon(b)$, $\varepsilon = 2.0$.

- As a result of the comparison performed in the previous step we find three sets of points: P_{TP} are the points found on both sets, P_{FP} are new points that were not found on the original image but were found on the transformed image, P_{FN} are the missed points that were found on the original image and were not found on the transformed image. The cardinalities of these sets are, respectively, TP, FP and FN values of the proposed detector. These values are used to calculate the detector's accuracy, precision and recall.

To evaluate repeatability we used the following transformations of the images:

- rotation by angle α, $-45° \leq \alpha \leq 45°$, α is increased by 3°;
- sub-pixel shift by t, $0.25 \leq t \leq 0.75$, t is increased by 0.05;
- scaling by s, $0.5 \leq s < 1.5$, s is increased by 0.1

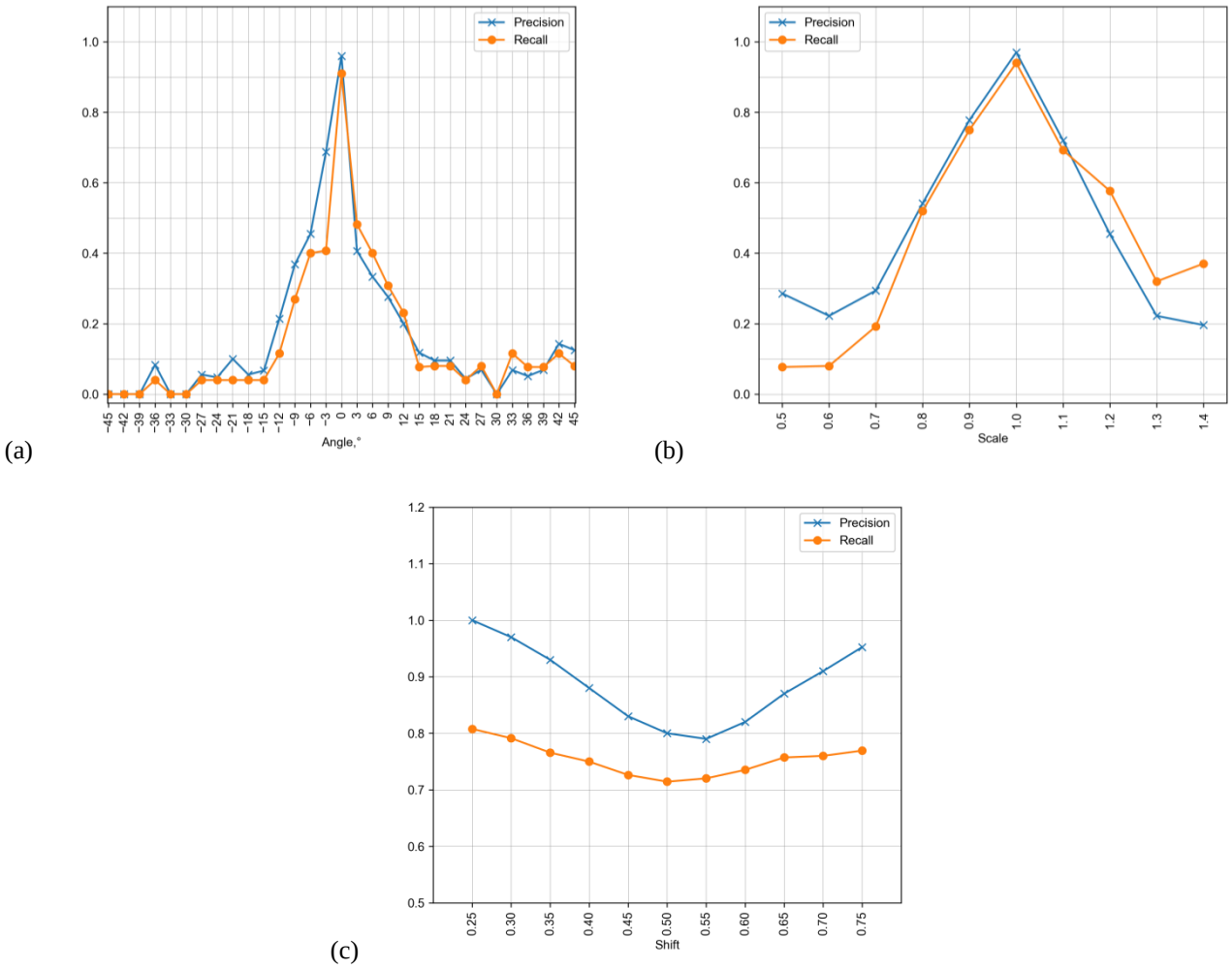

Fig. 2. Repeatability of the detector evaluated for various transformations: (a) rotation, (b) scaling, (c) translation.

The results of the repeatability evaluation of the proposed detector that was trained on the training set D are shown on fig. 2. The detector's performance can be considered adequate on rotated images for $-9° < \alpha < 9°$ and on scaled images for $0.8 \leq s \leq 1.2$. The performance on shifted images is high for the whole range of the parameter t.

4. Conclusion

In this paper we investigated a relatively new approach to feature point detection. Contrary to the standard approach to the problem, we didn't formulate any heuristics-based definition of the term feature point but tried to infer it inductively using the methods of machine learning and a representative training set. This enabled us to tune the proposed detector to a specific problem at hand. The results of the experimental evaluation of the detector verify that such a tuning is in fact possible. Moreover, the detector showed acceptable robustness to rotation and scaling transformation, and high robustness to sub-pixel shift transformation. This suggests a great potential of the learning-based approach to feature points detection.

Acknowledgements

The reported study was funded by RFBR according to the research project №17-29-03190-ofi.

References

[1] Szeliski R. Computer Vision: Algorithms and Applications. London: Springer, 2011; 812 p.
[2] Denisova AY, Myasnikov VV. Anomaly detection for hyperspectral imaginary. Computer Optics 2014; 38(2); 287–296.
[3] Tuytelaars T, Mikolajczyk R. Local invariant feature detectors: a survey. Foundations and trends® in computer graphics and vision 2008; 3(3): 177–280. DOI: 10.1561/0600000017.
[4] Li Y, Wang S, Tian Q, Ding X. A survey of recent advances in visual feature detection. Neurocomputing 2015; 149: 736–751. DOI: 10.1016/j.neucom.2014.08.003.
[5] Förstner W, Gülch E. A fast operator for detection and precise location of distinct points, corners and centres of circular features. Proc. ISPRS intercommission conference on fast processing of photogrammetric data 1998; 281–305.
[6] Harris C, Stephens M. A combined corner and edge detector. Alvey vision conference 1988; 15(50): 147–151.
[7] Shi J, Tomasi C. Good features to track. Proc. Intl Conf. on Comp. Vis. and Pat. Recog (CVPR) 1994; 593–600.
[8] Smith SM, Brady J.M. SUSAN – A new approach to low level image processing. International Journal of Computer Vision 1997; 23(1): 45–78. DOI: 10.1023/A:1007963824710.
[9] Rosten E, Drummond T. Machine learning for high-speed corner detection. European Conference on Computer Vision 2006; 430–443. DOI: 10.1007/11744023_34.
[10] Mair E, Hager GD, Burschka D, Suppa M, Hirzinger G. Adaptive and generic corner detection based on the accelerated segment test. European conference on Computer Vision 2010; 183–196. DOI: 10.1007/978-3-642-15552-9_14.
[11] Zhang X, Wang HA, Smith WB, Ling X, Lovell BC, Yang D. Corner detection based on gradient correlation matrices of planar curves. Pattern Recognition 2010; 43(4): 1207–1223. DOI: 10.1016/j.patcog.2009.10.017.
[12] Shui PL, Zhang WC. Corner detection and classification using anisotropic directional derivative representations. IEEE Transactions on Image Processing 2013; 22(8): 3204–3218. DOI: 10.1109/TIP.2013.2259834.
[13] Chernov AV, Myasnikov VV, Sergeyev VV. Fast Method for Local Image Processing and Analysis. Pattern Recognition and Image Analysis 1999; 9(2): 237–238.
[14] Flusser J, Suk T. Pattern recognition by affine moment invariants. Pattern Recognition and Image Analysis 1993; 26(1): 167–174. DOI: 10.1016/0031-3203(93)90098-H.
[15] Hu MK. Visual pattern recognition by moment invariants. IRE transactions on information theory 1962; 8(2): 179–187. DOI: 10.1109/TIT.1962.1057692.
[16] Myasnikov VV. Constructing efficient linear local features in image processing and analysis problems. Automation and Remote Control 2010; 72(3): 514–527. DOI: 10.1134/S0005117910030124.
[17] Theodoridis S. Machine learning: a Bayesian and optimization perspective. San Diego: Academic Press, 2015; 1062 p.
[18] Alpaydin E. Introduction to machine learning. Cambridge: MIT press, 2014; 584 p.
[19] Hastie T, Tibshirani R, Frieman J. Elements of statistical learning: data mining, inference, and prediction. London: Springer, 2011; 745 p.