ISSN 1870-4069

Free Form Object Recognition Module using A-KAZE and GCS

Karen Lizbeth Flores-Rodríguez, Felipe Trujillo-Romero

Universidad Tecnológica de la Mixteca,
División de Estudios de Posgrado, Oaxaca,
Mexico

karenflores350@hotmail.com, ftrujillo@mixteco.utm.mx

Abstract. This paper presents an object recognition module development. This module uses a local feature approach to identify keypoints in free form objects and an unsupervised artificial neural network (ANN) to associate the nearest ones and get clusters of each object learned. The module uses A-KAZE feature descriptor and Growing Cell Structure (GCS) ANN. The module is validated using an own data base, with twenty real objects and twenty different images each one. Here is presented a variety of experiments using from five to fifteen trainning images per object and the rest of them for evaluation. This method gets good results with 100% of discrimination between objects and up to 80% of correct classification.

Keywords: A-KAZE, growing cell structure, free form object recognition, local features.

1 Introduction

There are plenty of works referents to object recognition showing good results [1–4]. These works prove classification and object recognition is an issue solved for a lot of computer vision methods. These tools can use external features (size, signatures, polygonal approximations) or inner features (color, gray levels, texture). Additionally, the feature classification can be doing with different approach like ANN, support vector machines, statistical methods, etc.The choice of one of these methods depends of the problem, application or data base to work with. One of the current challenges is to get a method who can be used in real time, be faster and not need extra processing resource to be applied. Thinking in a human environment like an office or a house, where a human do his daily activities, it can be find several different objects. There are objects like phones, scissors, staplers, etc. These object are complex to modeling mathematically.

Then, if the goal is recognize them, a good solution is an object recognition method considering free form objects. The most methods like that use local feature to describe the objects in a faster way [5–8]. These methods, detects visually distinctive points in images called: interest points, salient points, keypoints or corner points. The detection includes the keypoints scale, orientation

and description making them a strong feature to learn and recognize an object. The most popular of them are Scale Invariant Feature Transform [9] (SIFT) and Speeded Up Robust Feature [10] (SURF). Recently, in [11], it was develop KAZE, a feature detector and descriptor algorithm comparable to SIFT but with an increase computational cost disadvantage compared to SURF. KAZE evolved to A-KAZE [12] showing an excellent compromise between speed and execution compared to BRISCK [13], ORB [14], SURF, and SIFT. The keypoints alone can be classified with a nearest neighbors algorithm. But talking about free form object where the all 3D form is taking into account and there are a lot of keypoints describing only one object, should be used an optimized method to cluster the data. The best choice is an ANN, whom is an approach with a high popularity and a way for develop adaptive coefficients to make decision function by series of patterns training presentations. Actually, there are a lot of ANN variants and is hard enough to choice one that fully solve a particular problem. In the free form object recognition module, the patterns to be classified are those obtained by the method A-KAZE and it is chosen the self-organizing GCS [15] for classification task which have the advantage to increase and decrease dynamically his form during the training phase.

There are many successful object recognition models who take local features into objects representations. These models can often be broken down into two steps: a coding step, of the most representative features in a scene, and an association step, which summarizes the features over nearest neighborhoods or some similar clusters. Several coding combinations and association schemes have been made. The bag of words or bag of features it will be made more popular, works like [16–18] use it with supervised or unsupervised classification obtaining result about 84% to 91.4%. ANN supervised and unsupervised have been used too, like in [19], the back–propagation algorithm with local signatures and self-organizing maps obtained a 75% correct classification. With unsupervised low level local descriptors [20], a model based on bayes, PCA and combining SIFT-PCA [21] make more distinctive the representation. These obtained a 95% classification. The unsupervised feature learning with convolutional ANN [22], [23] and [24], and local descriptors using SIFT obtained correct classification about 80%, 85.8%, and 92.5%. Another using the same model [25] but with images RGB-d obtained a 93.23%. Cellular ANN and SIFT had been used obtained 90%. The disadvantage is to have large data set and large time. Also many of them use data base like COIL-100, CALTHEC-101 and others. The free form object recognition module contribution is: (i) the automatically data association obtained from A-KAZE with the ANN self-organizing GCS, (ii) the speed because use only a few views, (iii) discriminate well the objects between them, and (iv) easy to be implemented. A performance analysis is carried to prove using an ANN model and keypoints can be applied in a real time module, remains consistent, be fast and be accurate. To evaluate the approach it is considered data cross performances, confusion matrix and receiver operating characteristics (RoC) curve analysis.

2 A-KAZE

A-KAZE refers to a method based on KAZE but faster because the dramatic speed-up introduced by Fast Explicit Diffusion (FED) schemes and the low computational demand and storage by Modified-Local Difference Binary (M-LDB). In this section is briefly described the algorithm from [12].

Firstly, define a evolution times set to build the nonlinear scale space in a O octaves and S sub-levels series mapped to pixels scale σ with

$$\sigma_i(o,s) = 2^{o+s/S}, o \in [0 \dots O-1], s \in [0 \dots S-1], i \in [0 \dots M], \tag{1}$$

where M is the total filtered images number. Then the discrete scale levels set is converted to time units

$$t_i(o,s) = \frac{1}{2}\sigma_i^2, i = \{0 \dots M\}. \tag{2}$$

The input images are convolved with a standard deviation Gaussian to reduce noise and artefact and to computed the contrast factor as the gradient histogram 70%. With both input images and contrast factor the FED scheme starts, the pyramidal approach algorithm 1 and inner cycle algorithm 2.

Algorithm 1: Pyramidal FED, nonlinear diffusion filtering

Data: Image L_0, contrast λ, τ_{max} and evolution times t_i
Result: Set of filtered images L_i, $i = 0...M$

for $i = 0 -> M-1$ **do**
- 1. Compute diffusivity $A(L^i)$
- 2. FEDcycle time $T = t_{i+1} - t_i$
- 3. Number of inner steps n
- 4. Compute step size τ_j
- 5. Set Prior $L^{i+1,0} = L^i$ $L^{i+1} =$ **FEDCycle**$(L^{i+1,0}, A(L^i), \tau_j)$ **if** $o_{i+1} > o_i$ **then**
  - Downsample L_{i+1} with mask $(\frac{1}{4}, \frac{1}{2}, \frac{1}{4})$
  - $\lambda = \lambda 0.75$

Algorithm 2: FED Cycle

Function FEDCycle $(L^{i+1,0}, A(L^i), \tau_j)$ **for** $j = 0 -> n-1$ **do**
- $L^{i+1,j+1} = (I + \tau_j A(L^i))L^{i+1,j}$

return $L^{i+1,n}$

The feature detection uses the Hessian determinant for each images filtered L_i in the nonlinear scale space. The differential multi-scale operators set are normalized with a scale factor

$$L^i_{Hessian} = \sigma^2_{i,norm}(L^i_{xx}L^i_{yy} - L^i_{xy}L^i_{yx}). \tag{3}$$

The concatenated Scharr filters is used for computing the second order derivatives with step size $\sigma_{i,norm}$. At each evolution level i, the detector response is check, if is higher than a pre-defined threshold and if is a maxima in a 3x3 pixels window. Then, the 2D keypoint position is estimated with sub-pixel accuracy.

The feature description use M-LDB with the derivatives computed in the feature detection step to compute an average approximation of the same areas in the intensity and gradient images. Finally, the descriptor vector of length 64 is obtained.

3 Growing Cell Structure

Next is briefly describe the unsupervised GCS model from [15]. The model is a Kohonen's self-organizing network variant. The main advantage over existing approaches is the model ability to automatically find a suitable network structure and size, achieve through a controlled growth process that also includes occasional units removal. The network dynamics is summarized in the next algorithm 3 and 4.

Algorithm 3: GCS

Data: ε_b best matching, ε_n neighboring and λ steps

Start: k-dimensional simplex $V = R^b$

while *(≠ desired network size)* **do**

  AdaptationSteps

Determine q: $h_q \geq h_q \quad (\forall\, c \in A)$

Look q largest distance neighbor f: $\| w_f - w_q \| \geq \| w_c - w_q \| \; (\forall\, c \in N_q)$

Insert cell r between q and f.

Initialize r: $w_r = 0.5(w_q + w_f)$

Redistribute counter: $\Delta\tau_c = \frac{|F_c^{(new)}| - |F_c^{(old)}|}{|F_c^{(old)}|}\tau_c$

Initialize new cell: $\tau_r = -\sum_{c \in N_r} \Delta\tau_c$

After insertion, check $\hat{p_i} < \eta$

Cells remove: $\hat{p} = \tilde{p} \sum_{c \in A} \tilde{f}_c$

Algorithm 4: GCS: **AdaptationSteps**

for *adaptationsteps* $= 0 - > \lambda$ **do**

  Choose an input signal ξ according to $P(\xi)$

  Locate the best matching unit $s = \phi_w(\xi)$.

  Increase matching:

  $\Delta w_s = \varepsilon_b(\xi - w_s)$

  $\Delta w_c = \varepsilon_n(\xi - w_c) \quad (\forall\, c \in N_s)$

  Increment the signal counter of s:

  $\Delta\tau_s = 1$

  Decrease all signal counters by a fraction α in the network A:

  $\Delta\tau_c = -\alpha\tau_c \quad (\forall\, c \in A)$

4 Object Recognition Module

The free form object recognition module is proposed like the described in the Figure 1. The module is divided in two main phases: *Leraning* and *Recognition*.

In *Learning* phase, it is used different object view images as input. Using A-KAZE extracts keypoints, process the data and pass trough the unsupervised GCS for classification. Each class is reserved into a data base with a label (name of the object). The *Recognition* phase, uses only one object view image as input.

As in the *Learning* phase, it is used A-KAZE to extract keypoints, process the data and pass trough GCS evaluation to obtain the object label.

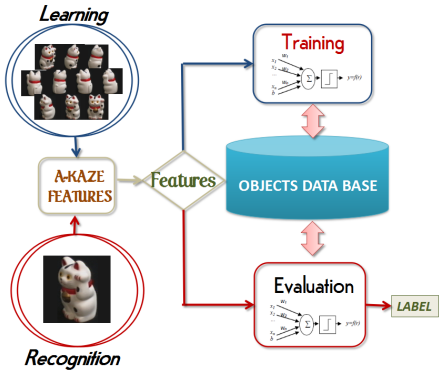

Fig. 1. Object recognition module blocks diagram.

4.1 Module Description

After extract the keypoints from the images and before pass trough GCS, the data is processed. A-KAZE gets a big data from an image and all the data can be reduced to an histogram. This is proved with the process explanation next:

1. There are four images from an object in different views: front, back, left and right. For each view all the keypoints are extracted.
2. Each view is compared with the right. The Figure 2 presents the 50 best keypoints coincidences connected by a line.

The above shows, that images from one view to other, in an object, share many features. If the object has more images it will have more coincidences between views.

3. All the keypoints are summarized into an histogram per image from the previous object. The keypoints homogenization summarized the view essence.

The histogram keeps the information about an object view because it is learned all the keypoints whom are part of that view. In Figure 3 (a), the four histograms are overlayed to show the similitude between them and the graph errors from each comparison is shown in Figure 3 (b). Also, a comparison between the object 2 in Table 4 and the actual object is carried out and the result are shown in

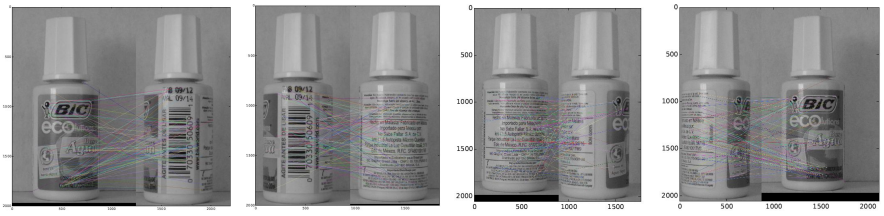

Fig. 2. Four images from an object in different views: front, back, left and right view. Graphical representation of keypoints coincidences.

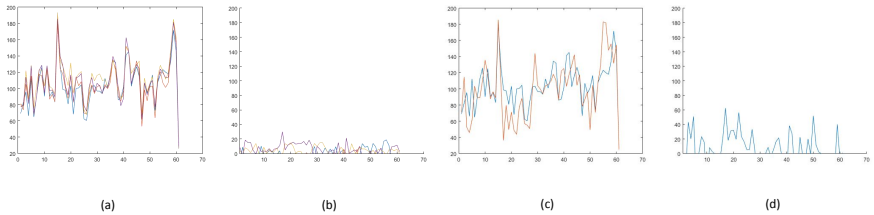

Fig. 3. (a) Histograms per image view, (b) Error between views, (c) Two different object histogram, (d) Error graph.

Figure 3 (c) and in Figure 3 (d) it is observed that the error between them is bigger.

4. Once obtained the histograms per object, these are the GCS input. The GCS automatically do a nearest neighbor association of them.
5. The histograms belongs to an object will be classified into a neurons group. Therefore, there will be more than one neuron per object.

Algorithm 5: Object recognition module. *Training*

Data: $\boldsymbol{I}$ images, $\boldsymbol{N}$ objects, $\boldsymbol{L}$ images per object, $\boldsymbol{E}$ object label
Result: ***classes(N)*** Object classes.

for $n \leftarrow 1$ ***to*** N **do**
  for $l \leftarrow 1$ ***to*** L **do**
    keypoints =A-KAZE($\boldsymbol{I(n,l)}$)
    H(n,l)= $\frac{\Sigma keypoints}{Totalkeypoints}$
classes($\boldsymbol{N,E}$)=GCS($\boldsymbol{H}$)

Algorithm 6: Object recognition module. *Evaluation*

Data: $\boldsymbol{I}$ image
Result: ***class(I)*** object class and $\boldsymbol{E}$ object label

keypoints =A-KAZE($\boldsymbol{I}$)
H($\boldsymbol{I}$)= $\frac{\Sigma keypoints}{Totalkeypoints}$
class($\boldsymbol{I,E}$)=GCS($\boldsymbol{H}$)

4.2 Algorithm Training and Classification

The free form object recognition module implementation is as follows. There are *Training* and *Evaluation* tasks. In *Training* task, as is shown in algorithm 5, the input are the images, object number, images number and the object label.

The module accurate depends of the images number per object. Although, in some cases, only five images are enough to learn an object. Using A-KAZE the keypoints are extracted and the histograms per image are builded. All the histograms are sent to ANN GCS to classification. Each class is saved in a data base with a label (object name). The *Evaluation* task, as is shown in algorithm 6, receives only one image per object to recognize as input. As in the *Training* task, uses A-KAZE to extract all the keypoints and builds the image histogram. Then, the histogram is sent to evaluation with the ANN trained. The ANN delivers the nearest class and the recognized object label.

5 Experiments and Results

The free form object recognition module validation is as next. It is used an own five real objects data set shown in Table 4. Each of them with twenty views images to different sizes. The experiments are from 5 to 15 images to training and the rest for evaluation. The main parameters of each experiments, described in Table 1 are: training images number, evaluation images, ANN max neurons numbers and training steps. The neurons and epoch values are chosen to be between 100 minimum and 500 maxima depends of the training images number. In GCS $\varepsilon_b = 0.05$, $\varepsilon_n = 0.005$ and $\lambda = NumOfSamples$ adaptation steps for all the experiments. The experiments takes randomly images from the data base for each object in each experiment.

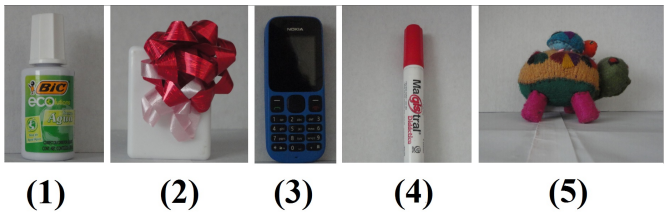

Fig. 4. Real objects data set. (1) corrector, (2) gift, (3) cell-phone, (4) marker, (5) turtle.

5.1 Results

The experiments results are shown in a confusion matrix for each one and RoC curve analysis for the classification. The confusion matrix is shown in Table 2,

Table 1. Main parameters per experiment.

Experiment	Training	Evaluation	Neurons	Epochs
First	5	5	100	100
Second	5	1	100	100
Third	5	15	200	200
Fourth	10	10	200	500
Fifth	15	5	200	500

these show a consistent classification mistaking the two last objects in each experiment. The training and evaluation performance was done in MATLAB 2015 in a PC with Intel(R) Core(TM) i7-4770 3.40GHz processor, with 12 GB RAM, Windows 8 (64) bits OS.

Table 2. Experiments confusion matrix.

First					
	1	**2**	**3**	**4**	**5**
1	5	0	0	0	0
2	0	5	0	0	0
3	0	0	5	0	0
4	2	0	0	3	0
5	0	0	0	0	5

Second					
	1	**2**	**3**	**4**	**5**
1	10	0	0	0	0
2	0	10	0	0	0
3	0	0	10	0	0
4	4	0	0	6	0
5	1	0	0	0	9

Third					
	1	**2**	**3**	**4**	**5**
1	5	0	0	0	0
2	0	5	0	0	0
3	0	0	5	0	0
4	2	0	0	3	0
5	0	0	0	0	5

Fourth					
	1	**2**	**3**	**4**	**5**
1	1	0	0	0	0
2	0	1	0	0	0
3	0	0	1	0	0
4	1	0	0	0	0
5	0	0	0	1	0

Fifth					
	1	**2**	**3**	**4**	**5**
1	13	0	0	1	1
2	0	15	0	0	0
3	0	0	15	0	0
4	4	0	0	10	1
5	1	1	2	0	11

Table 3. Experiments results.

Experiment	(%)	Time (sec)
First	92	0.351
Second	80	0.426
Third	85.33	1.246
Fourth	90	2.344
Fifth	92	3.457

However, the experiments percentages and the performance training time presented in Table 3, exhibit two best with 92% correct classification, the rest up to 80% correct classification. The performance training time depends of

the neurons quantity which at the same time depends of the images quantity. Considering the results and only the neurons quantity, the training time increase with a lineal approximation like (4):

$$time = 0.0063(neurons) - 0.2395. \tag{4}$$

The increase in time remaind small from a neuron quantity to other. If the images training increase to 20 the neurons should increase to 700 and the time will be 4.1705 s approximately. Remembering that are 20 images per each of the five objects, then, the module is considering faster.

Table 4. Operating characteristics per experiment.

Object	TP	TN	FP	FN	Sensitivity	Specificity	1-Specificity
(1)	34	129	15	2	0.944	0.896	0.104
(2)	36	143	1	0	1	0.993	0.007
(3)	36	142	2	0	1	0.986	0.014
(4)	22	142	2	14	0.611	0.986	0.014
(5)	30	142	2	6	0.833	0.986	0.014

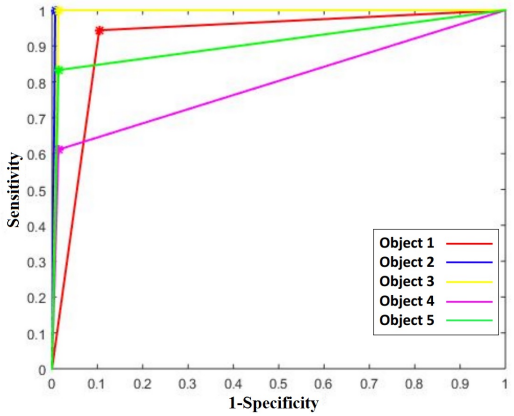

Fig. 5. RoC curve of each object for all the experiments.

Additionally, the sensitivity and specificity experiments analysis is obtained using the true positive (TP), true negatives (TN), false positives (FP) and false negatives (FN). The values are in Table 4, corresponding to all the data classify in the five experiments, it means 180 images total. These operating characteristics can be reformulated slightly and then presented graphically as shown below in Figure 5. The plots shows true positive against the false positive rate for the different possible classification result. These curves demonstrates,

the tradeoff between sensitivity and specificity (any increase in sensitivity will be accompanied by a decrease in specificity). The curves follow the left-hand border and the RoC space top border, showing the test accuracy. The object 1, 4 and, 5 curves are closer to the 45-degree RoC space diagonal but still closer to the RoC space top border. These are less accurate but enough to do a good classification.

6 Conclusions

This paper presented a free form object recognition module development based on local features and ANN. The features identify free form objects interest points and are used as ANN input to cluster the nearest. It was used A-KAZE and GCS.

The GCS network uses from 100 to 500 neurons maxima in training. The module was validated using an own real object data base with five objects and twenty images per object. The ANN was trained with different images number per object. The obtained results are good ones with 100% classification rate and up to 80% recognition. It means the module can discriminate very well the objects. There are some confusions in the evaluation because it is necessary to give the system images with more features from object views. Future work is: (i) to increase the objects number, (ii) mix more similar objects to see the module performance, and (iii) to implement it in a humanoid robot NAO to do service tasks.

Acknowledgement. Acknowledgements to CONACYT for the economic scholarship support to get a master's degree 584190/571271.

References

1. Pereyra, M., Destefanis, E.: Reconocimiento de objetos para control de calidad mediante la descomposición en figuras geométricas. In: 2 Congreso Nacional de Ingeniería Informática/Sistemas de Información (2014)
2. Gelsi, L., Di Caro, C.A., Fernández, H.A.: Reconocimiento rápido de objetos. Anales AFA, 22, pp. 95–97 (2011)
3. Gómez, J.D.: Proceso de reconocimiento de objetos asistido por computadora, aplicando gases neurales y técnicas de data mining. Revista Científica y Tecnológica de la Facultad de Ingeniería. pp. 12–1 (2007)
4. Roth, P. M., Winter, M.: Survey of appearance-based methods for object recognition. Technical Report: ICG-TR-01/08, Graz University of Technology, Austria (2008)
5. Albanesi, B., Funes, N., Chichizola, F.: Reconocimiento de objetos en video utilizando sift paralelo. In: XVI Congreso Argentino de Ciencias de la Computación. (2010)
6. Salamanca, S., Adán, A., Cerrada C., Adán, M., Merchán, P., Pérez, E.: Reconocimiento de objetos de forma libre a partir de los datos de rango de una vista parcial usando cono curvaturas ponderadas. Revista Iberoamericana de Automática e Informática Industrial, pp. 94–106 (2007)

7. Chen, H., Bhanu, B.: 3d free-form object recognition in range images using local surface patches. Pattern Recognition Letters. 28, pp. 1252–1262 (2007)
8. Guo, Y., Beccamoun, M., Sohel, F., Liu, M., Wan, J.: 3D object recognition in cluttered scenes with local surface features: a survey. IEEE Transactions on Pattern Analysis and Machine Intelligence (2013)
9. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision (2004)
10. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: SURF: Speeded up robust features. Computer Vision and Image Understanding (CVIU), 110-3, pp. 346–359 (2008)
11. Alcantarilla, P. F., Bartoli, A., Davison, A. J.: Kaze features. ECCV, Springer-Verlag Berlin Heidelberg, pp. 214–227 (2012)
12. Alcantarilla, P. F., Nuevo, J., Bartoli, A.: Fast explicit diffusion for accelerated features in nonlinear scale spaces. In: British Machine Vision Conference (BMVC) (2013)
13. Leutenegger, S., Chli, M., Siegwart, R. Y.: BRISK: Binary robust invariant scalable keypoints. In: IEEE Intl. Conf. on Computer Vision (ICCV), pp. 2548–2555 (2011)
14. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: Orb: an efficient alternative to siftt or surf. In: IEEE Intl. Conf. on Computer Vision (ICCV), pp. 2564–2571 (2011)
15. Fritzke, B.: Growing cell structures, a self-organizing network for unsupervised and supervised learning. Neural Networks, 7-9, pp. 1441–1460 (1994)
16. Boureau, Y. L., Bach, F., LeCun, Y., Ponce J.: Learning mid-level features for recognition. In: Computer Vision and Pattern Recognition (CVPR), IEEE Conference (2010)
17. Oursland, J.: Real-time scale invariant object recognition using an articial neural network. In: Proceedings of the ISCA 25th International Conference on Computers and Their Applications, CATA (2010)
18. Sudholt S., Rothacker L., Fink G. A.: Learning local image descriptors for word spotting. Document Analysis and Recognition (ICDAR), pp. 651–655 (2015)
19. Ros, J., Laurent, C., Lefebvre, G.: Cascade of unsupervised and supervised neural networks for natural image classification. In: Image and Video Retrieval: 5th International Conference, CIVR, pp. 92–101 (2006)
20. Osendorfer, C., Bayer, J., Urban, S., Van der Smagt, P.: Unsupervised feature learning for low-level local image descriptors. CoRR (2013)
21. Ken, Y., Sukthankar, R.: Pca-sift: a more distinctive representation for local image descriptors. Computer Vision and Pattern Recognition, CVPR, 2 pp. 506–513 (2004)
22. Dosovitskiy, A., Fischer, P., Sprinenberg, J. T., Riedmiller, M., Brox,T.: Discriminative unsupervised feature learning with exemplar convolutional neural networks. CoRR, abs/1406.6909 (2014)
23. Paulin, M., Douze, M., Harchaoui, Z., Mairal, J., Perronnin, F., Schmid, C.: Local Convolutional Features with Unsupervised Training for Image Retrieval. In: IEEE International Conference on Computer Vision, ICCV, (2015)
24. Dong, J., Soatto, S.: Domain-Size Pooling in Local Descriptors: DSP-SIFT. CoRR (2014)
25. Jhuo, I. H., Gao, S., Zhuang, L., Lee, D. T., Ma, Y.: Unsupervised Feature Learning for RGB-D. IEEE Transactions on Image Processing, 24 pp. 4459–4473 (2015)