Analysis of Compact Features for RGB-D Visual Search

Alioscia Petrelli[1(✉)], Danilo Pau[2], and Luigi Di Stefano[1]

[1] University of Bologna, Bologna, Italy
{alioscia.petrelli,luigi.distefano}@unibo.it
http://vision.deis.unibo.it
[2] ST Microelectronics, Agrate Brianza, Italy
danilo.pau@st.com
http://www.st.com

Abstract. Anticipating the oncoming integration of depth sensing into mobile devices, we experimentally compare different compact features for representing RGB-D images in mobile visual search. Experiments on 3 state-of-the-art datasets, addressing both category and instance recognition, show how Deep Features provided by Convolutional Neural Networks better represent appearance information, whereas shape is more effectively encoded through Kernel Descriptors. Moreover, our evaluation suggests that learning to weight the relative contribution of depth and appearance is key to deploy effectively depth sensing in forthcoming mobile visual search scenarios.

Keywords: RGB-D visual search · Binary hash codes · Deep learning

1 Introduction

Nowadays almost any mobile device is equipped with an high-resolution camera and constantly connected to the Internet. This fosters development and increasing diffusion of a variety of mobile visual search tools, such as Google Goggles, Amazon Flow, CamFind, Vuforia, and WeChat Image Platform. A mobile visual search engine allows the user to easily gather information about the objects seen in the camera field of view. Purposely, she/he would just snap a picture to have the mobile device computing a representation of the image which is sent to a remote server and matched into a database to recognize image content and report back relevant information. Such scenario has been made real by the fertile research on mobile visual search [6,9,12] as well as by sensor miniaturization, which enables inexpensive integration of cameras into smartphones and tables. Alongside these progresses, the advances on 3D sensing have lead to the availability of affordable and effective RGB-D cameras, such as the Microsoft Kinect or Creative Senz3D, and, predictably, will enable depth sensing on mobile devices in the near future. Indeed, a number of solutions aimed at enhancing mobile devices with depth sensing capabilities do already exist. *Occipital* has recently released

© Springer International Publishing Switzerland 2015
V. Murino and E. Puppo (Eds.): ICIAP 2015, Part II, LNCS 9280, pp. 14–24, 2015.
DOI: 10.1007/978-3-319-23234-8_2

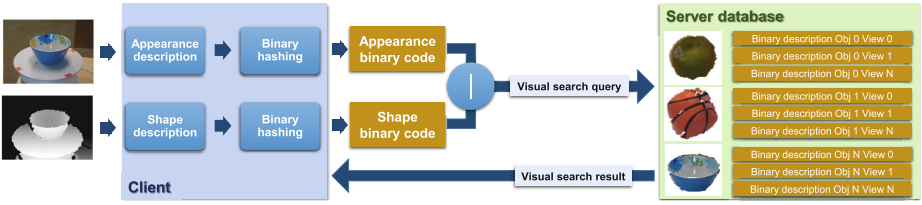

Fig. 1. Visual search architecture deployed to investigate on RGB-D features.

the *Structure Sensor*[1], a structured light depth camera that can be clipped onto a tablet. In [22], *Pelican Imaging*[2] introduced a camera array that captures light fields and synthesizes a range image, the camera being small enough to be embedded into next generation smartphones. The *HTC One (M8)* smartphone, released by *HTC* in 2014, integrates a 2-megapixel depth sensor and provides the *Dual Lens SDK* to foster the development of 3D applications on Android. Google has given green light to *Project Tango*[3], that is shipping to researchers and programmers a prototype tablet equipped with 3D sensing capabilities and up-to-date APIs. The foreseeable advent of depth sensing on mobile devices at a significant scale may pave the way to a new generation of mobile applications. In particular, we are interested in investigating on whether and how mobile visual search architectures may benefit of depth sensing capabilities.

A fundamental requirement of any mobile visual search architecture deals with compactness of the description sent to the server, so as to guarantee a satisfying user experience even in case of limited bandwidth or congestion of the network. Moreover, research on binary codes is not limited to mobile visual search but pertains the entire field of content-based image retrieval. As a matter of fact, compact and binary descriptors are key to efficient storage and matching in databases comprising millions of images. Thus, several approaches to either conceive compact image descriptors or compress existing ones have been proposed in literature [4,5,7,13]. However, research on compact representation has addressed only RGB images thus far. To the best of our knowledge, the only works that address compact description of depth information are [18], which is focused on 3D point clouds, and [19], which, instead, deals with RGB-D images. Both papers, though, propose local descriptors without addressing the issue of obtaining a compact global representation of the image.

To fill this lack, in this paper we consider next generation mobile visual search scenarios and propose an investigation on how to encode both appearance and depth information to obtain compact binary codes that properly describe RGB-D images. More precisely, within a visual search pipeline that allows to exploit both color and depth data, we analyze different image description approaches and carry out an experimental comparison aimed at evaluating their relative merits and limits.

[1] http://structure.io
[2] http://www.pelicanimaging.com
[3] www.google.com/atap/projecttango

2 Visual Search Architecture

Fig. 1 depicts the architecture we deployed to evaluate different image description approaches for the task of mobile RGB-D visual search. Given an RGB-D image acquired by a mobile device, the pipeline independently process the appearance and shape channels at client side, so to produce compact binary codes that are concatenated and sent to the server. Each binary code is obtained by a two step process that computes first a global encoding of the whole image and then creates the binary description through a similarity-preserving hashing stage. At server side, the received binary code is matched against a database of descriptions in order to find the most similar image.

2.1 Image Description

For global encoding of the RGB and depth images we considered the established paradigm dealing with aggregation of local features. Accordingly, local features are first extracted and described, then they are globally encoded through the *Fisher Kernel* [20] algorithm. Moreover, we considered an approach based on deep neural networks so as to address both hand-crafted and learned features.

SIFT: As a baseline local description approach we use SIFT[4] [16], which detects keypoints through DoG and produces descriptions of length $D = 128$. We apply SIFT on intensity images without any preprocessing, whereas depth images are rescaled in the range [1, 255] reserving the 0 value for denoting invalid depths. As to isolate depths belonging to the searched object, we modeled the distribution of depths of database images as a gaussian, then we linearly rescaled depths falling on less than $2 \times \sigma$ from the gaussian mean and saturated all the others. Then, the *Fisher Kernel*[5] method is deployed to aggregate SIFT features into a global representation of the entire image. Fisher kernels has been introduced to combine the power of discriminative classifiers with the ability of generative models to handle representations comprising a variable number of measurement samples. The encoding vector is the gradient of the sample log-likelihood with respect to the parameters of the generative model, which, intuitively, can be seen as the contribution of the parameters to the generation of the samples. Perronnin et al. in [20] applied Fisher kernels to image classification by modeling visual vocabularies by *Gaussian mixture models* (GMM). In our setup, the parameters are the mean and covariance (assumed diagonal) of each of the N_G components of the mixture. Thus, global encodings have length $2 \times D \times N_G$. According to our experiments, best results are obtained with a number of components as small as $N_G = 3$.

Dense SIFT: To investigate on whether uniform sampling of features may turn out more beneficial than keypoint detection to visual search applications, we compute SIFT descriptors on 16×16 patches sampled across a regular grid. Then, densely computed descriptors are aggregated via *Fisher Kernel*. As $N_G = 1$ turns

[4] SIFT features are computed by the OpenCV implementation.
[5] We use the Fisher Kernel implementation available in the VLFeat libray.

out here the best choice for the number of components, global encodings of RGB and depth images both have length $2 \times D$.

Kernel Descriptors: Given the excellent results reported on a variety of RGB-D recognition tasks, we have considered the *RGB-D Kernel Descriptors* introduced in [1,2]. Kernel descriptors are a generalization of descriptors based on orientation histograms, such as SIFT and HOG, which may suffer from quantization errors due to binning. Kernel descriptors overcome this issue by defining the similarity between two patches through kernel functions, referred to as *Match Kernels*, that average out across the continuous similarities between pairs of pixel attributes within the two patches. Local description is performed on patches sampled across a regular grid, with each patch represented by a 200-dimensional feature vector. Finally, local features are condensed into a global description by *Fisher Kernel* ($N_G = 2$). The authors propose 8 types of kernel descriptors by defining match kernels for different patch attributes such as intensity and depth gradient, local binary patterns and object size. In our experiments we used the C++ implementation made available online by the authors, which permits to apply 4 types of Kernel Descriptors. In particular, appearance information is described by kernels dealing with *Intensity Gradients* and *Color*, while shape information is captured by kernels based on *Depth Gradients* and *Spin Images*.

Deep Features: In [10], Gupta et al. address the problem of globally encoding an RGB-D image through a Convolutional Neural Network (CNN) architecture. Purposely, they exploit the so called "AlexNet" proposed in [14], that processes a 256×256 RGB image and can produce a 4096-dimensional feature vector as output of the last hidden layer. Besides describing the RGB image, the authors of [10] deploy the HHA representation to map the depth image into three channels: *H*orizontal disparity, *H*eight above ground and *A*ngle between local surface normal and inferred gravity direction. Accordingly, AlexNet is also fed with the HHA representation as if it were an RGB image. The authors ground this approach on the hypothesis that RGB and depth images share common structures due to, for example, disparity edges corresponding to object boundaries in RGB images. Moreover, the authors perform fine tuning of AlexNet based on HHA data. Our experiments indicate that slightly better results can be achieved by feeding the hashing stages with the 100 *Principal Components* of the 4096-dimensional vectors computed by both the RGB and HHA networks.

2.2 Binary Hashing

Among the several hashing approaches proposed in the last years, we considered the state-of-the-art *Spherical Hashing* (SH) method [11], which has been reported to result peculiarly effective on large datasets. Let N_b be the number of bits comprising the binary description. At training time, SH represents the data with a set of N_b hyperspheres and choose the value of the $i-th$ bit depending on whether the feature vector falls inside or outside the $i-th$ hypersphere. To determine the centers and radii of the hyperspheres, an iterative optimization process is performed so to achieve balanced partitioning of descriptions for each

hashing function as well as independence between any two of them. We applied the iterative process on 1% of the training samples, such percentage turning out adequate to train SH. Furthermore, we do not exploit the *Spherical hashing distance* proposed in [11], as our experiments did not show any improvement with respect to the standard Hamming distance.

2.3 Matching

As illustrated in Fig. 1, the appearance and shape binary codes are juxtaposed to form the final binary code. This is sent to the server to be matched against a database of stored binary codes using the Hamming distance together with the weighted k-NN search approach introduced in [8]. To speed-up the search for the k-NNs, the server side database is efficiently indexed by the *multi-probe LSH* scheme proposed in [17].

3 Experimental Evaluation

This section reports the results of the experimental analysis we performed to determine the merits and limits of the considered features. For each dataset, we split the images so as to reserve a portion as the training set used to estimate the GMM required by Fisher Kernel, to find the principal components of the Deep Features extracted by the CNNs and to train SH. After that, we describe each image of the training set with the trained pipeline and build the index used in the server-side matching stage. Finally, we describe all the test images and calculate the rate of them correctly recognized in the training set. This procedure is repeated 10 times splitting differently the training and test sets and, eventually, the attained recognition rates are averaged. To compare the different types of features, we execute the pipeline by considering either the appearance information extracted from the RGB image only or the shape information extracted from the depth image only or we fuse the two kinds of information concatenating their binary codes (see again Fig. 1). For each configuration, we run the pipeline while varying the length of the final binary code from 32 to 1024 bits and plot the attained mean recognition rates as a function of the code length. In the case of kernel descriptors, both for appearance and shape description, we compute the two available kernel descriptors, perform the hashing separately and then juxtapose the resulting binary codes.

3.1 Datasets

The evaluation concerns 3 state-of-the-art datasets of household objects: the *RGB-D Object Dataset*, *CIN 2D+3D* and *BigBIRD*. The former two datasets share a two-level category/instance structure that allows us to evaluate our framework on both category and instance recognition tasks, whereas BigBIRD consists of object instances not partitioned into categories.

The **RGB-D Object Dataset** [15] is nowadays the de-facto standard for evaluating and comparing visual recognition systems relying on RGB-D sensing. For each of the 300 household objects composing the dataset, a set of acquisitions from different vantage points has been collected and segmented from the background so as to gather a total of 41,877 RGB-D images. Each object belongs to one of 51 categories based on the WordNet hierarchy. As for instance recognition, we chose the *Alternating Contiguous Frames* methodology [15].

The **CIN 2D+3D** dataset [3] consists of 18 categories, which in turn include about 10 instances each. The objects, placed on a turntable, have been acquired from 36 vantage points by rotating the turntable by 10°upon each acquisition. In [3], the authors propose a procedure aimed at evaluating simultaneously the ability to recognize both instances and categories. However, similarly to standard methodology defined with the RGB-D Object Dataset, we test the performance for the two tasks of category and instance recognition separately. Thus, for category recognition, we select a tenth of the instances for each category as test set and train the pipeline on the remaining ones. Likewise, for instance recognition. we split a different tenth of the views of each instance and uses it as test set whereas the remaining acquisitions are used for training. As suggested by the authors, we discard the "Perforator" and "Phone" categories from the evaluation as they do not include a sufficient number of instances. Instead, we do not aggregate "Fork", "Spoon" and "Knife" into the "Silverware" super-category.

The **BigBIRD** dataset [21] comprises 600 views of 125 object instances, including mostly supermarket products. The dataset includes quite challenging instances as most of them are box-shaped products recognizable only by their package textures, which sometimes are very similar (e.g. as in the case e.g. "pop secret butter" and "pop secret light butter") or distinguishable just due to color. As reliable segmentation masks are not provided for 11 objects (the majority of them being transparent bottles), we discarded them from the data used in our experiments. As the authors do not suggest a methodology to evaluate the dataset, for each of the 10 trials, we randomly select 100 acquisitions and split them so as to perform testing on a tenth of them and training based on the others.

3.2 Results

The results of our experimental evaluation are reported in Fig 2. Firstly, the charts reveal that an encoding based on SIFT keypoints (the green curves in the figure) is not effective within our visual search architecture as it provides the lowest recognition rates in all but the experiment dealing with appearance-only description on BigBIRD. Better results are scored by methods leveraging on densely computed local descriptors. Indeed, if SIFT is applied to patches extracted across a regular grid, the recognition rate raises substantially (red plots), especially in category recognition experiments (first 2 rows of the figure). Overall, the best performance are provided by representations based on Kernel Descriptors and Deep Features. Accordingly, in the remainder of the discussion we will mostly focus on these two approaches. We start by commenting the behavior of representations based on appearance information only (first column

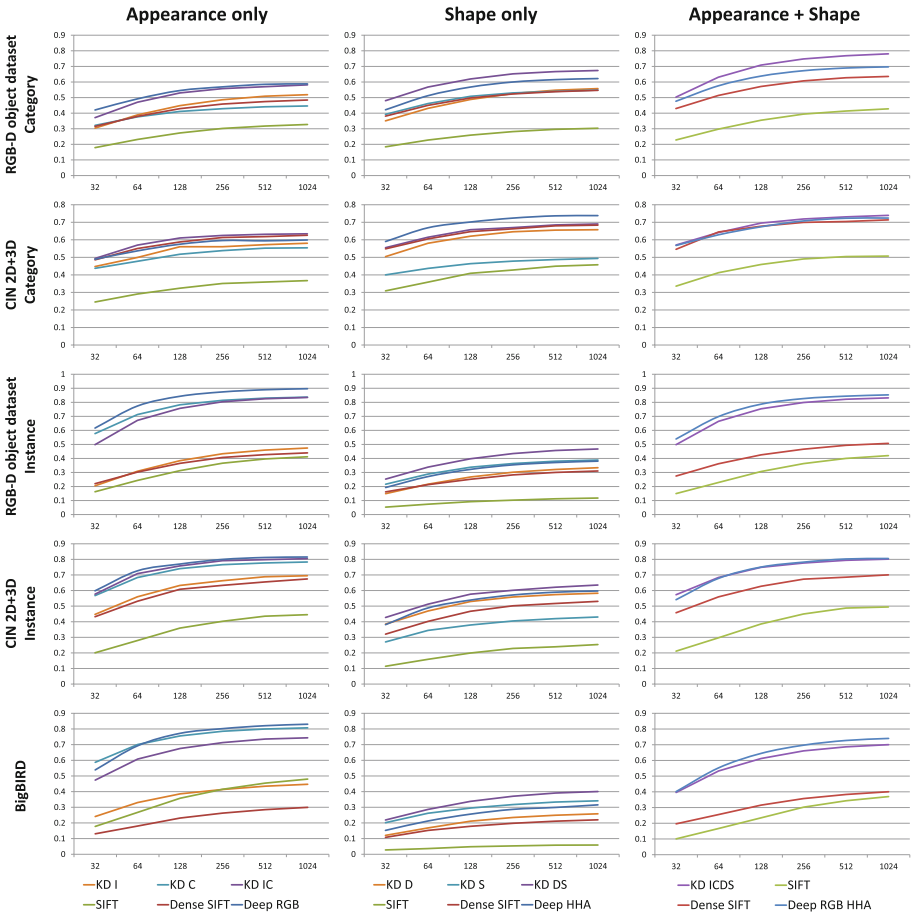

Fig. 2. The charts are organized as a table, the rows dealing with the different datasets and recognition tasks (first 2 rows: category recognition, last 3 rows: instance recognition) and the 3 columns reporting, respectively, the results obtained with appearance-based descriptions only, shape-based descriptions only and fusion of appearance and shape. Each chart reports the recognition rate as a function of the length in bits of the binary code. The different curves are identified by the legend underneath columns. Kernel Descriptors (KD) based on Intensity gradients, Color, Depth gradients and Spin Images are labeled as I, C, D and S respectively.

of Fig 2) and address the impact of the two types of Kernel Descriptors first. The charts report the recognition rates yielded by Kernel Descriptors based on either intensity gradients or color as orange and cyan curves respectively, whereas purple curves deal with the performance attained assigning half of the binary code to the former and half to the latter. In category recognition experiments, both Kernel Descriptors contribute significantly to the recognition ability of the pipeline, so that their synergistic deployment ends up in improving the recognition rate, as

Fig. 3. Examples of BigBIRD objects distinguishable by colour and texture only.

perceivable more clearly in the case of the RGB-D Object dataset. On the other hand, in case of instance recognition experiments, color seems the main cue that allows for telling apart objects in the considered datasets. This is particularly noticeable in the BigBIRD dataset, as deploying half of the binary code to represent intensity gradients turns out even detrimental with respect to spending all bits to encode color. This can be ascribed to the nature of the dataset that, as already pointed out, consists mainly of boxes and bottles distinguishable by color features only (a few examples are shown in Fig. 3). The comparison between Kernel Descriptors and Deep Features (the blue plots in the charts) highlights how, with the exception of category recognition on the CIN 2D+3D dataset, the latter approach provides quite consistently higher recognition rates.

As for the experiments addressing representation of shape information only (second column of Fig 2), it is unclear which Kernel Descriptor allows for encoding more effectively the depth channel between that relying on Depth Gradients and on Spin Image, which represented by the orange and cyan curves respectively. Nonetheless, it is clear that fusing the two contributions by splitting the code bits evenly (purple curve) does increase the recognition rates insomuch as to outperform Deep Features in 4 out of the 5 experiments. This vouches as the two types of kernel descriptors are complementary and thus the recognition ability of the pipeline can benefit significantly of their synergistic deployment. Looking now at the first two columns, it seems quite evident how shape is more relevant than appearance in category recognition experiments, the opposite being the case of instance recognition, appearance being definitely the primary cue to tell apart the different objects comprising the considered datasets.

The third column of charts in Fig 2 reports the recognition rates attained by exploiting jointly the appearance and shape information provided by RGB-D images. In the task of category recognition (first 2 rows), Kernel Descriptors (purple curve) provide the best performance whereas Deep Features (blue curve) turn out more effective in distinguishing object instances (last 3 rows). This can be explained by observing that Kernel Descriptors seem more effective to encode shape information that, in turn, is more relevant to the task of category recognition, whereas Deep Features better capture the appearance information that is key to effective instance recognition. In Table 1 we summarize the results shown in Fig 2 by highlighting the approaches providing the best performance when deploying either appearance or shape information only (first 2 columns). Furthermore, the last column of the table reports the configuration yielding the highest recognition rate when both kinds of information are available. In the

Table 1. Summary of the results reported in Fig 2. For each dataset and both types of experiment, the first two columns highlight the method providing the best recognition rate in case either only appearance or only shape information is deployed for image representation. Then, the last column highlights the approach yielding the highest possible recognition rate assuming that both kinds of information are available.

	Appearance	Shape	Best
RGB-D Object Dataset - Category	Deep RGB	KD DS	KD ICDS
CIN 2D+3D - Category	KD IC	Deep HHA	KD ICDS
RGB-D Object Dataset - Instance	Deep RGB	KD DS	Deep RGB
CIN 2D+3D - Instance	Deep RGB	KD DS	Deep RGB
BigBIRD	Deep RGB	KD DS	Deep RGB

case of category recognition, exploiting both appearance and shape information is beneficial as the best configuration involves the combined use of all Kernel Descriptors. Conversely, for the task of instance recognition, our evaluation suggests to simply discard the shape contribution for the available code bits would be best spent to encode the RGB image only by Deep Features. Puzzled by the above finding, we devised an additional type of instance recognition experiment, whereby the bits of the binary codes are no longer split evenly between appearance and shape but, instead, according to a varying ratio. We run the experiments setting the description length to 1024 bits (i.e. the lengthiest considered in Fig 2) while deploying Deep Features to encode the RGB image and Kernel Descriptors (Depth Gradient and Spin Image) to encode the depth image, i.e. the best approaches to represent appearance and shape respectively. In Fig 4 we report the obtained recognition rates: as expected, peak performance are reached with a high ratio of code bits deployed to represent appearance. Interestingly, though, the best performance are never achieved by allocating the totality of the binary code to appearance information, but, rather, by splitting properly code bits between appearance and shape. In particular, with CIN 2D+3D the best recognition rate is reached by allocating 1/4 of the binary code to shape, while

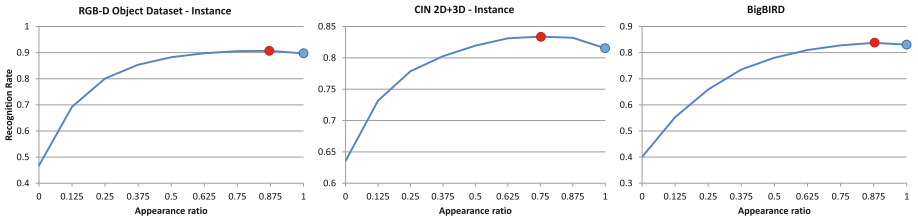

Fig. 4. Instance recognition experiments with a varying relative contribution of appearance (Deep Features) and shape (Kernel Descriptors). The horizontal axis indicates the ratio of bits of the binary code deployed to encode appearance. Accordingly, the performance of the best methods in Table 1 are denoted by blue dots (all bits encode appearance by Deep Features). The best recognition rates attainable by splitting code bits unevenly between appearance and shape are highlighted by red dots.

the optimal ratio is 1/8 for both the RGB-D Object dataset as well as BigBIRD. Indeed, a shape-to-appearance ratio of about 1/8 would provide better performance than disregarding shape with all the considered datasets. Hence, proper deployment of the depth channel associated with RGB-D images may contribute to improve instance recognition performance even in scenarios where texture and color provide the primary cues to tell objects apart.

4 Conclusion and Future Work

Our analysis on image features for RGB-D mobile visual search reveals that an approach based on Kernel Descriptors or Deep Features followed by Spherical Hashing can provide an effective and very compact image encoding. In particular, Deep Features computed through Convolutional Neural Networks seem the best choice to represent appearance, whereas shape information is better captured by Kernel Descriptors. In category recognition scenarios, both RGB and depth information contribute notably to ascertain the class to which a query object does belong. Instead, in instance recognition tasks, our experiments highlight how appearance features, like texture and colour, are key to tell apart the specific object instances stored into the database, whereas depth furnishes a limited, though still informative, contribution. Indeed, an approach based on simply juxtaposing the two representations does not take into account the different discriminative power that the two cues may convey in diverse scenarios. Hence, devising suitable strategies to learn and deploy the relative prominence of appearance and depth in diverse settings is among the key research issues to be addressed in order to leverage on depth sensing in forthcoming mobile visual search scenarios. We are currently investigating on a learning-to-rank approach aimed at deploying the weights provided by two separate k-NN classifiers associated with appearance and depth in order to better judge candidates according to the specific distinctiveness of the two cues for any query. The architecture has been ported on a *Samsung Galaxy Tab Pro 10.1* equipped with a *Structure Sensor* for the acquisition of the depth image. The pipeline, deploying the four types of Kernel Descriptors and trained on the *RGB-D Object Dataset*, spends, on average, 550 ms for producing the binary code and 2 ms to perform the matching.

References

1. Bo, L., Ren, X., Fox, D.: Kernel descriptors for visual recognition. In: Advances in Neural Information Processing Systems, vol. 23, pp. 1–9 (2010)
2. Bo, L., Ren, X., Fox, D.: Depth kernel descriptors for object recognition. In: Intelligent Robots and Systems (2011)
3. Browatzki, B., Fischer, J.: Going into depth: Evaluating 2D and 3D cues for object classification on a new, large-scale object dataset. In: International Conference on Computer Vision Workshops (2011)
4. Calonder, M., Lepetit, V., Strecha, C., Fua, P.: BRIEF: binary robust independent elementary features. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part IV. LNCS, vol. 6314, pp. 778–792. Springer, Heidelberg (2010)

5. Chandrasekhar, V., Makar, M., Takacs, G., Chen, D., Tsai, S.S., Cheung, N.M., Grzeszczuk, R., Reznik, Y., Girod, B.: Survey of SIFT compression schemes. In: International Conference on Pattern Recognition (2010)
6. Chandrasekhar, V., Takacs, G., Chen, D.M., Tsai, S.S., Makar, M., Girod, B.: Feature matching performance of compact descriptors for visual search. In: Data Compression Conference (2014)
7. Chandrasekhar, V., Takacs, G., Chen, D.M., Tsai, S.S., Reznik, Y., Grzeszczuk, R., Girod, B.: Compressed Histogram of Gradients: A Low-Bitrate Descriptor. International Journal of Computer Vision (2011)
8. Dudani, S.A.: The Distance-Weighted k-Nearest-Neighbor Rule. Transactions on Systems, Man, and Cybernetics, 325–327 (1976)
9. Girod, B., Chandrasekhar, V., Chen, D.M., Cheung, N.M., Grzeszczuk, R., Reznik, Y., Takacs, G., Tsai, S.S., Vedantham, R.: Mobile visual search. IEEE Signal Processing Magazine, 61–76, July 2011
10. Gupta, S., Girshick, R., Arbeláez, P., Malik, J.: Learning rich features from RGB-D images for object detection and segmentation. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014, Part VII. LNCS, vol. 8695, pp. 345–360. Springer, Heidelberg (2014)
11. Heo, J.P., Lee, Y., He, J., Chang, S.F., Yoon, S.E.: Spherical hashing. In: Conference on Computer Vision and Pattern Recognition, pp. 2957–2964 (2012)
12. Ji, R., Duan, L.Y., Chen, J., Yao, H., Yuan, J., Rui, Y., Gao, W.: Location Discriminative Vocabulary Coding for Mobile Landmark Search. International Journal of Computer Vision, 290–314 (2011)
13. Johnson, M.: Generalized descriptor compression for storage and matching. In: British Machine Vision Conference, pp. 23.1-23.11 (2010)
14. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1–9 (2012)
15. Lai, K., Bo, L., Ren, X., Fox, D.: A large-scale hierarchical multi-view rgb-d object dataset. In: International Conference on Robotics and Automation, pp. 1817–1824 (2011)
16. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vision **60**(2), 91–110 (2004)
17. Lv, Q., Josephson, W., Wang, Z., Charikar, M., Li, K.: Multi-probe LSH: efficient indexing for high-dimensional similarity search. In: International Conference on Very Large Data Bases (2007)
18. Malaguti, F., Tombari, F., Salti, S., Pau, D., Di Stefano, L.: Toward compressed 3D descriptors. In: International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission, pp. 176–183, October 2012
19. Nascimento, E.R., Oliveira, G.L., Campos, M.F.M., Vieira, A.W., Schwartz, W.R.: BRAND: a robust appearance and depth descriptor for RGB-D images. In: International Conference on Intelligent Robots and Systems, pp. 1720–1726, October 2012
20. Perronnin, F., Dance, C.: Fisher kernels on visual vocabularies for image categorization. In: Conference on Computer Vision and Pattern Recognition (2007)
21. Singh, A., Sha, J., Narayan, K.S., Achim, T., Abbeel, P.: BigBIRD: a large-scale 3D database of object instances. In: International Conference on Robotics and Automation, pp. 509–516 (2014)
22. Venkataraman, K., Lelescu, D., Duparr, J., McMahon, A., Molina, G., Chatterjee, P., Mullis, R.: PiCam: an ultra-thin high performance monolithic camera array. In: Siggraph Asia (2013)