

CCF-Based Awareness in Agent Model ABGP

Zhongzhi Shi^{1(✉)}, Jinpeng Yue^{1,2}, Gang Ma^{1,2}, and Xi Yang¹

¹ Key Laboratory of Intelligent Information Processing, Institute of Computing Technology,
Chinese Academy of Sciences, Beijing 100190, China

{shizz,mag,yangx}@ics.ict.ac.cn, yuejp@163.com

² University of Chinese Academy of Sciences, Beijing 100049, China

Abstract. ABGP is an agent cognitive model, which combines external perception and internal mental state of agents together. The model consists of 4-tuple <Awareness, Belief, Goal, Plan>. For environment awareness a framework named Coding and Combining Features (CCF) is proposed in this paper. The CCF framework combines feature descriptors together based on the compact coding. In the CCF framework firstly compact codes are learned to be robust to complex image content variations. Secondly, the compact codes from multiple information sources are combined by multi-kernel learning to generate the final feature. The experiment results show us that CCF method achieves a better performance in the identification task.

Keywords: Compact coding · Awareness · Agent model ABGP

1 Introduction

Cyborg intelligence is dedicated to integrating artificial intelligence with biological intelligence by tightly connecting machines and biological beings [1]. The key issue is how to collaborate between machines and biological beings? In our project entitled Cognitive Computational Model for Brain Machine integration we use multi-agent collaboration to simulate their behaviors.

We consider that under practical environment the agent should be rational. Rational agents have an explicit representation for their environment and objectives they are trying to achieve. Rationality means that the agent will always perform the most promising actions to achieve its objectives. For a rational agent faced with a complex natural scene, how to get knowledge from scenes to drive their actions? Most agent designer maybe have a common view that either create a virtual scene or set some single inflexible rules for agent to recognize surrounding.

As an internal mental model of agent, BDI model has been well recognized in philosophical and artificial intelligence area. Bratman's philosophical theory was formalized by Cohen and Levesque [2]. In their formalism, intentions are defined in terms of temporal sequences of agent's beliefs and goals. Rao and Georgeff have proposed a possible-worlds formalism for BDI architecture [3]. The abstract architecture they proposed comprises three dynamic data structures representing the agent's beliefs, desires, and intentions, together with an input queue of events. The update operations

on beliefs, desires, and intentions are subject to respective compatibility requirements. These functions are critical in enforcing the formalized constraints upon the agent's mental attitudes. The events the system can recognize include both external events and internal events.

A cognitive model for multi-agent collaboration should consider external perception and internal mental state of agents. Awareness is knowledge created through interaction between an agent and its environment. Endsley pointed out awareness has four basic characteristics [4]:

- a) Awareness is knowledge about the state of a particular environment.
- b) Environments change over time, so awareness must be kept up to date.
- c) People maintain their awareness by interacting with the environment.
- d) Awareness is usually a secondary goal—that is, the overall goal is not simply

For internal mental state of agents we can consult BDI model which was conceived by Bratman as a theory of human practical reasoning. Its success is based on its simplicity reducing the explanation framework for complex human behavior to the motivational stance [5]. This means that the causes for actions are always related to human desires ignoring other facets of human cognition such as emotions. Another strength of the BDI model is the consistent usage of folk psychological notions that closely correspond to the way people talk about human behavior.

In terms of above considerations we propose a cognitive model ABGP for multi-agent collaboration through 4-tuple <Awareness, Belief, Goal, Plan>[6]. Awareness is an information pathway connecting to the world (including natural scenes and other agents in multi-agent system). Beliefs can be viewed as the agent's knowledge about its setting and itself. Goals make up the agent's wishes and drive the course of its actions. Plans represent agent's means to achieve its goals. If we expect the agent constructed by ABGP model has an ability to sense directly the natural scenes, a new awareness pathway must to be proposed.

In computer vision, visual feature extraction is vital for object identification, image retrieval, scene classification, etc. These features are complement and should be synthesized together in computer vision applications. For example, edge and texture features are suitable for landscape and building scene classification; shape features are suitable to recognize objects such as apples. However, existing applications mainly focus on one kind of features such as the widely used SIFT [7] texture feature. To improve the feature discriminability, we put forward that complement features corresponding to different reacting regions of brain should be binding together to simulate the brain.

Wu [8] proposed a method to bundle two complement feature detecting methods together, i.e. SIFT [7] and MSER [9]. Xie [10] also bundles two detectors together. However, these two methods are detector binding methods, not considering the description aspects. The description of visual features is vital for the analysis of visual contents. Many of such methods have been proposed to code features into more compact representations, such as binary codes.

For environment awareness we propose a framework named CCF (Coding and Combining Features) to combine feature descriptors together based on the compact coding. In the CCF framework we take advantage of complement feature extracting methods, compact coding methods and multi-kernel hashing together. We encode

features into compact binary codes by an effective Locality Preserving Coding method and combine the codes together by multi-kernel hashing. The combined feature achieves a good trade-off between discriminability, robust and learning efficiency. And thus our method achieves a better performance in the experiments.

In previous work we have described agent model ABGP[6]. In this paper we describe especially on environment awareness. This paper is organized as follows. Briefly introduction of agent model ABGP is given in Section 2. Section 3 describes the coding and combining features framework. The experimental results are put into Section 4. Section 5 concludes with a discussion on environment awareness of rational agents.

2 Agent Model ABGP

In computer science and artificial intelligence, agent can be viewed as perceiving its environment information through sensors and acting environment through effectors [11]. A cognitive model for rational agent should especially consider external perception and internal mental state. The external perception as a knowledge is created through interaction between an agent and its world. A cognitive model ABGP (shown as in Figure 1) is represented as a 4-tuple framework as $\langle \text{Awareness; Belief; Goal; Plan} \rangle$, where awareness is an information pathway connecting to the world and a relationship between agents. Awareness is defined as 2-tuple $MA = \langle \text{Element, Relation} \rangle$ [6].

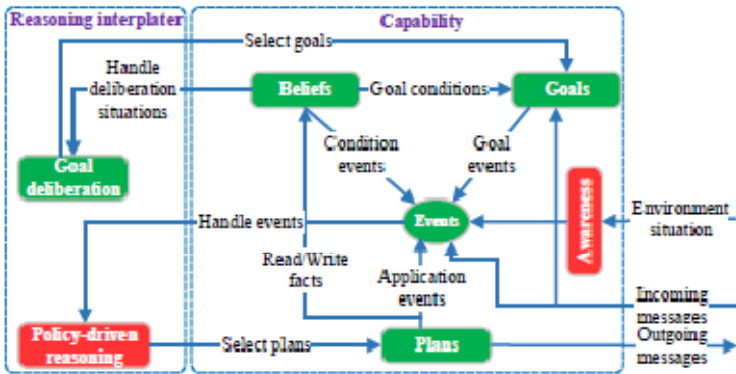


Fig. 1. Agent model ABGP

Belief can be viewed as the agent’s knowledge about its environment and itself. Goal represents the concrete motivations that influence an agent’s behaviors. Plan is used to achieve agent’s goals. Policy can be seen as a state transition function, and multi-agent system changes their state based on policies.

In this paper, we only discuss how to identify who is participating. In order to identify an object we propose a coding and combining features framework described in next section.

3 Coding and Combining Features Framework

3.1 Introduction of the Framework

The CCF Framework process an image into a set of codes, and then bundles the codes together to accomplish recognition or identification tasks. The model is composed of three steps:

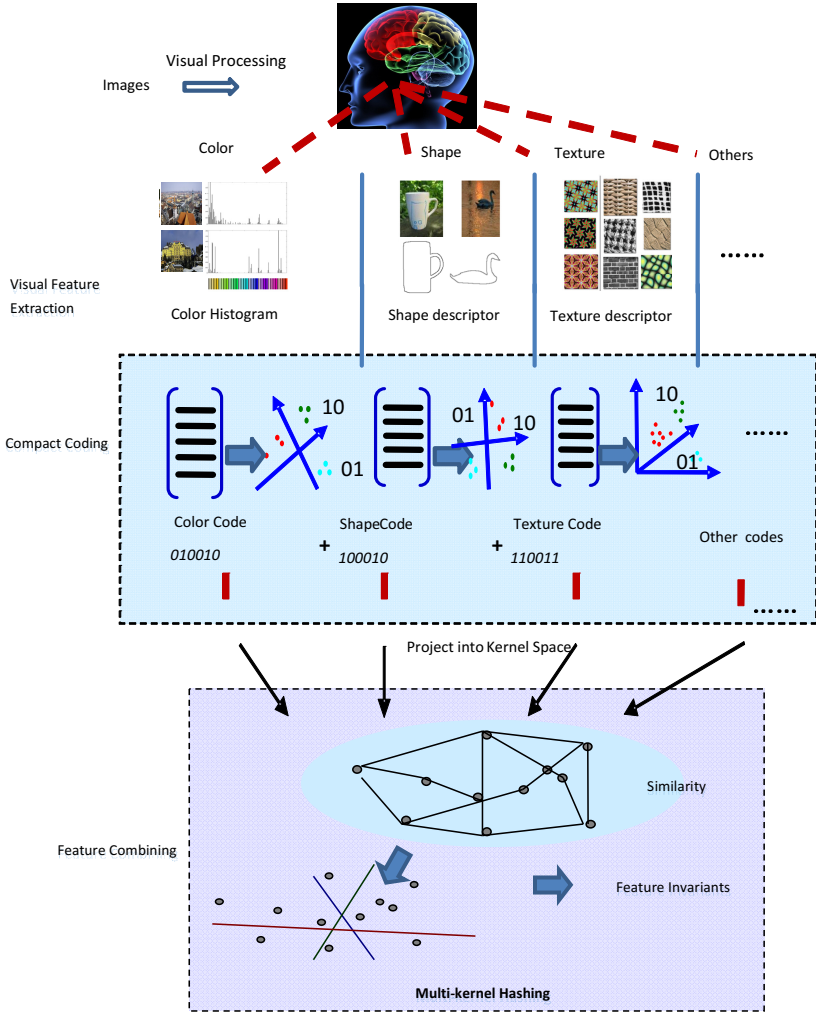


Fig. 2. Coding and Combining Features Framework

1. *Visual Feature Extraction.* Visual features such as color histogram, Gabor filters, intensity histogram, etc. are widely used in content based image analysis. In Figure 2, there are only three different features referred as samples. Actually, most of existing feature descriptors can be adopted in the model to simulate the visual processing of brain.

2. *Compact Coding.* After various features are extracted, we compact them into efficient and effective codes. To preserve as much information as possible during the coding process, we propose a novel coding method which will be introduced in detail in the next section.
3. *Learning Combined Features.* The codes generated in the compact coding step are then combined by multi-kernel hashing to make a more discriminative feature representation. Since different features such as color, texture impose different information, they are complement and effective in different situations.

3.2 Compact Coding

In this step, our object is not only to encode the features to a more robust and compact codes, but also to bind the features that are visually similar but change slightly in visual contents. We seek a binary coding approach that preserves the visual similarity relationships under the encoding Hamming space. Specifically, we propose a Locality Preserving Coding (LPC) method to learn robust binary codes by exploring the underlying manifold structure of training samples, with effective and efficient optimization.

The objective in this step is to code the visually similar patches into similar binary codes. In this section, we use affine transformation to simulate patches with similar contents but different in scales, slight transitions, and small content variations, as is shown in Figure 3. The other transformations such as lighting, Gaussian noise, scaling, etc can also be used to simulate visually similar patches. Then binary codes are learned on the patch by exploring the manifold structure of patch descriptors.

We aim to preserve the manifold structure of visual similar features when compacting descriptors to binary codes. Let $X=\{x_1, \dots, x_n\}$ be a set of d -dimensional visual descriptors. Suppose we want to get a k -bit code y_i of x_i , then k hash functions leading to k Hamming embeddings are needed. We use linear projection and threshold to encode X , and then binary code is computed using the following equation:

$$y_i = \text{sign} (P x_i^T + t) \quad (1)$$

where P is a $k*d$ matrix and t is a $k*1$ vector.

The projection matrix P is the key of embedding and threshold t is the key of binarization. To encode the visually similar descriptors into the same binary code, we compute P by solving an optimization problem and utilizing the visual relationships between descriptors.

Given a matrix W with weights characterizing the similarity of two image patches, we want to learn a P to preserve the local structure, which satisfies the following:

$$\text{minimize} : \sum_{ij} \left\| y'_i - y'_j \right\|^2 w_{ij} \quad (2)$$

where y' is the data projected by P , i.e. $y'=P x^T$ and w_{ij} is defined as (3) or (4) :

$$w_{ij} = \begin{cases} \exp(-\|x_i - x_j\|^2 / \beta), & \text{If } x_i \text{ and } x_j \text{ are visually similar} \\ 0, & \text{Otherwise} \end{cases} \quad (3)$$

$$w_{ij} = \begin{cases} 1, & \text{If } \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ are visually similar} \\ 0, & \text{Otherwise} \end{cases} \quad (4)$$

The similarity learning method is the spirit of LPP and P can be obtained by solving a generalized Eigen value problem as in [12]. Then we concatenate the generated codes to the final decision code. The similarity between two objects can be measured by the distance between their final decision codes. A smaller distance means the two objects are more similar.

3.3 Learning Combined-Feature

In this section, the codes of different features are combined by multi kernel hashing. Suppose we are given a set of N training binary codes $x^m = [x_1^m, x_2^m \dots x_N^m]$ with M different visual features. The hash codes Y with p bits are learned to preserve the manifold structure of visual similar features by multi-kernel hashing. For each feature, let K_{ij} denote the multi-kernel function between the i -th and j -th data, which is linearly combined by kernels for each feature K_{ij}^m . Let $\varphi(\cdot)$ denote the embedding function of the p -th multi-kernel function, the hyperplane vector V_p in the kernel space can be represented as a combination of L landmarks Z_l embedded in corresponding kernel space:

$$V_p = \sum_{l=1}^L W_{lp} \varphi(Z_l), l = 1, \dots, L \quad (4)$$

The p -th binary code after multi-kernel hashing is:

$$y_p = \text{sign}(V_p^T \varphi(\cdot) + b_p) \quad (5)$$

Then the final combined code after multi-kernel hashing of the compact code $x = [x_1, \dots, x_m]$ is:

$$y = \text{sign}(W^T [K(x, Z_1), \dots, K(x, Z_L)]^T + b)$$

The formulation forces the learned hash functions to preserve the given similarity as much as possible, by optimizing both hyperplane vectors W :

$$\begin{aligned} \min & \frac{1}{2} \sum_{i,j=1}^N S_{ij} \|Y_i - Y_j\|_2 + \lambda \|V\|_F^2 \\ \text{s.t.} & \sum_{i=1}^N Y_i = 0, \frac{1}{N} \sum_{i=1}^N Y_i Y_i^T = 0 \end{aligned} \quad (6)$$

The optimization problem can be solved by the relaxed problem and details can be found in [13].

The CCF framework can be applied to the object retrieval and recognition task. Take retrieval for example, the similarity between two objects can be measured by the distance of their final decision codes. The hamming distance is used to measure the similarity. A smaller distance means the two objects are more similar in the manifold structure.

4 Experimental Results

In our CCF framework, compact coding and multi-kernel hashing to combine features are two key steps. Firstly, we evaluate the independent compact coding step. Secondly, we evaluate the multi-kernel hashing by combining two different features.

4.1 Compact Coding Evaluation

Dataset. We simulate the affine transformations of images selected from INRIA Holiday, utilize SIFT to detect features and generate visually similar image patches. There are totally 10k groups, some of which are illustrated in Figure 3. Each group of image patches is similar in vision but changes in computer representations. The objective of compact coding is to encode one group of patches into the same binary code. We select 5k groups randomly for training and the rest for testing.

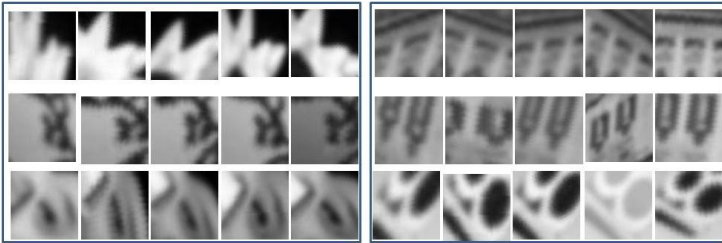


Fig. 3. Samples of evaluation dataset

We compare our compact coding method with LSH (Locality Sensitive Hashing) [14], SH (Spectral Hashing)[15], LDAHash (Linear Discriminant Analysis Hash) [16] which are coding methods widely used in multimedia retrieval task. The experiment conducts image patch retrieval using these methods and our coding method, the results are shown in Figure 4. The LPH refers to our method using equation (3), while LPH-1 refers to the method using equation (4).

Figure 4 uses recall and rate-of-codes-retrieved curve to evaluate the retrieval task. A higher recall means more correct similar patches are found, which indicates the image patch retrieval accuracy. The rate of codes retrieved indicates the noises contained in the final results. And thus a lower rate means lower noises and the number of incorrect patches is smaller. We can conclude from the figure that our compact coding method achieves a better performance in this task. The reason is that our coding method not only utilizes the supervised visual information, but also considers the manifold structure of data.

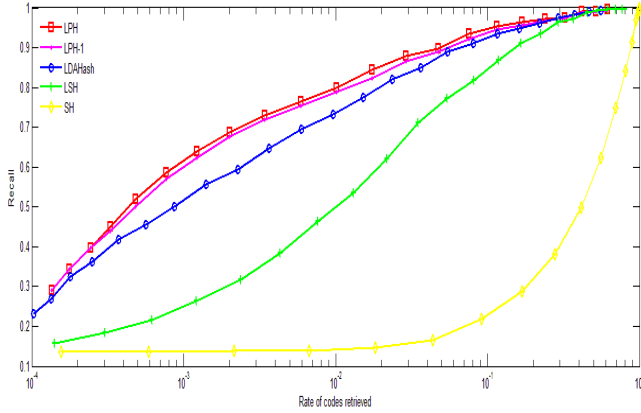


Fig. 4. Comparison of compact coding methods

4.2 Feature Combining Evaluation

Then we evaluate the combined feature on the Ukbench [17] image retrieval dataset, which contains 10,200 images including groups of 4 similar images. In this experiment, we combine texture histogram and color histogram together based on the compact coding. The texture feature is extracted by computing texture histogram from image patches, and the dimensionality is 80. The color feature is extracted by computing quantized color histogram and the dimensionality is 27. And the compact code is of 64-bit, which consumes much less memory.

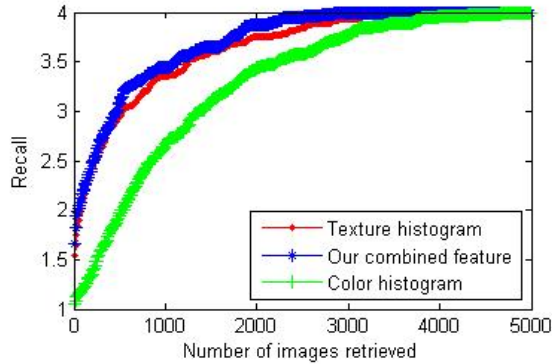


Fig. 5. Combined feature evaluation

From Figure 5, we can see that the combined feature achieves a higher recall than the color histogram and the texture feature. The combined feature utilizes two complementary features and benefit from compact coding, and thus features generated by our CCF framework is more discriminative and robust. Our method is effective with much less memory consumption, which can be used in large-scale applications.

We also use the CIFAR-10 dataset [18] to evaluate multi-kernel hashing based on compact coding, which consists of 60k 32×32 color images in 10 classes with 6k images per class. There are 50k training images and 10k test images. For each image, two features are extracted to evaluate our coding and combining framework, i.e. GIST feature of 384-D and bag of visual words of 300-D quantized from dense SIFT features of 8×8 patches with 4 space overlap. The GIST feature and SIFT features are extracted, compactly coded by LPC and then combined by multi-kernel hashing. The combining method will be evaluated in this experiment.

From the Figure 6 we can see that the precision and recall of our CCF method outperforms LSH with both 16bits and 32bits.

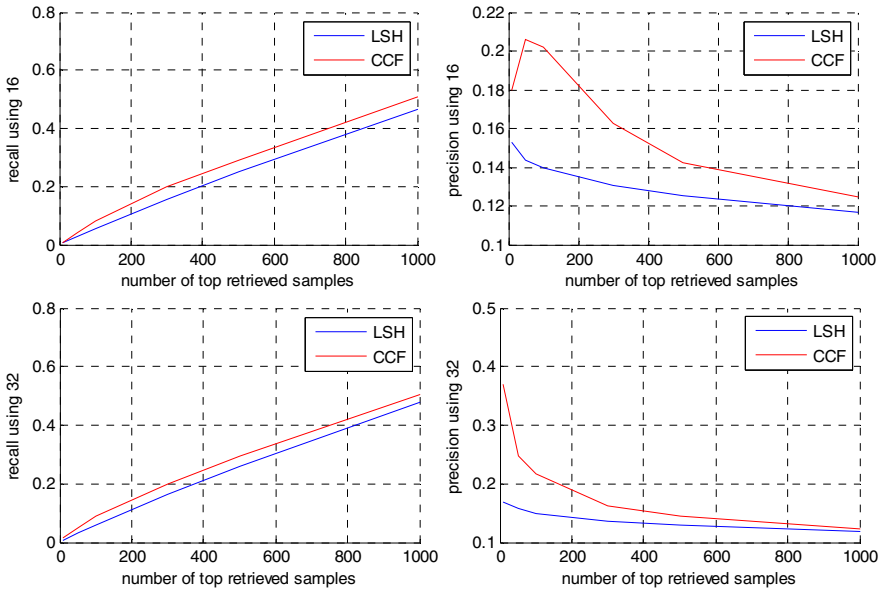


Fig. 6. Precision and recall comparison results

5 Conclusions

We propose a CCF novel feature combining framework to generate a robust and discriminative feature representation with effective coding technique. In the framework, features are encoded into compact codes which represent features with much more information and less units. Then the compact codes are combined to the final discriminative and robust feature by multi-kernel hashing.

The CCF framework is applied to environment awareness for agent model ABGP, which combines external perception and internal mental state. The CCF framework also can be applied many other feature extraction methods. Actually, the weights of different features can be utilized and learned from training samples.

Acknowledgment. This work was supported by the National Program on Key Basic Research Project (973 Program) (No. 2013CB329502), National Natural Science Foundation of China (61035003), National Science and Technology Support Program (2012BA107B02) .

References

1. Wu, Z., Pan, G., Zheng, N.: Cyborg Intelligence. *IEEE Intelligent Systems* **28**(5), 31–33 (2013)
2. Cohen, P.R., Levesque, H.J.: Intention is choice with commitment. *Artificial Intelligence* **42**(2–3), 213–361 (1990)
3. Rao, A.S., Georgeff, M.P.: Modeling rational agents within a BDI-architecture. In: Allen, J., Fikes, R., Sandewall, E. (eds.) *Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning*. Morgan Kaufmann Publishers, San Mateo (1991)
4. Endsley, M.: Toward a Theory of Situation Awareness in Dynamic Systems. *Human Factors* **37**(1), 32–64 (1995)
5. Pokahr, A., Braubach, L.: The active components approach for distributed systems development. *IJPEDS* **28**(4), 321–369 (2013)
6. Shi, Z., Zhang, J., Yue, J., Yang, X.: A Cognitive Model for Multi-Agent Collaboration. *International Journal of Intelligence Science* **4**(1), 1–6 (2014)
7. Lowe, D.: Distinctive image features from scale invariant keypoints. *IJCV* **60**(2), 91–110 (2004)
8. Wu, Z., Ke, Q., Isard, M., Sun, J.: Bundling Features for Large Scale Partial-Duplicate Web Image Search. In: *CVPR 2009* (2009)
9. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide baseline stereo from maximally stable extremal regions. In: *BMVC*, pp. 384–393 (2002)
10. Xie, H., Gao, K., Zhang, Y., Li, J., Liu, Y., Ren, H.: Effective and efficient image copy detection based on GPU. In: *Kutulakos, K.N. (ed.) ECCV 2010 Workshops, Part II. LNCS*, vol. 6554, pp. 338–349. Springer, Heidelberg (2012)
11. Shi, Z., Wang, X., Yue, J.: Cognitive Cycle in Mind Model CAM. *International Journal of Intelligence Science* **1**(2), 25–34 (2011)
12. He, X., Niyogi, P.: Locality preserving projections. In: *Neural Information Processing Systems*. MIT Press (2003)
13. Liu, X.L., He, J.F., Liu, D., Lang, B.: Compact kernel hashing with multiple features. In: *MM 2012* (2012)
14. Datar, M., Immorlica, N., Indyk, P., et al.: Locality-sensitive hashing scheme based on p-stable distributions. In: *Proceedings of the 20th annual symposium on Computational geometry (SCG)*, pp. 253–262 (2004)
15. Weiss, Y., Torralba, A., Fergus, R.: Spectral hashing. In: *Proceedings of Neural Information Processing Systems (NIPS)*, Canada, pp. 1–8 (2008)
16. Strecha, C., Bronstein, A., Bronstein, M., et al.: LDAHash: Improved Matching with Smaller Descriptors. *IEEE Trans on PAMI* **34**(99), 1–9 (2012)
17. Nister, D., Stewenius, H.: Scalable Recognition with a Vocabulary Tree. *Computer Vision and Pattern Recognition* (2006)
18. <http://www.cs.toronto.edu/~kriz/cifar.html>