Scientific Journal Impact Factor (SJIF): 1.711

e-ISSN: 2349-9745
p-ISSN: 2393-8161

International Journal of Modern Trends in Engineering and Research
www.ijmter.com

Interactive Multimodal Visual Search on Mobile Device

Miss. Sushmita Bankar[1], Prof. D. B. Kshirsagar[2]
[1,2]*Computer Department, SRES Collage of Engineering, Kopargaon*

Abstract— The proposed method describes a novel multimodal interactive image search system on mobile devices. The system, the Joint search with ImaGe, Speech, And Word Plus (JIGSAW+), takes full advantage of the multimodal input and natural user interactions of mobile devices. It is designed for users who already have pictures in their minds but have no precise descriptions or names to address them. By describing it using speech and then refining the recognized query by interactively composing a visual query using exemplary images, the user can easily find the desired images through a few natural multimodal interactions with his/her mobile device.

Keywords - Mobile visual search, Exemplar images, multimodal search, interactive search, mobile device.

I. INTRODUCTION

Image search is a hot topic in both computer vision and information retrieval with many applications. The traditional desktop image search systems with text queries have dominated the user behavior for a quite long period. However, while on the go, more and more consumers use phones or other mobile devices as their personal concierges surfing on the Internet. Along this trend, searching is becoming pervasive and one of the most popular applications on mobile devices. The bursting of mobile users puts forward the new requests for image retrieval.

Compared with text search, map search, and photo-to-search, visual (image and video) search is still not very popular on the phone, though image search has become a common tool on the PC since 10 years ago, with which the user can input text query to retrieve relevant images. A main reason why such image search applications are not popular on mobile device is that the existing image search applications do not perfectly accommodate to the mobile and local oriented user intent. Due to this, the search results are rarely useful and the user experience on the phone is not always enjoyable. First of all, typing is a tedious job on the phone no matter whether a tiny keyboard or a touch screen is used. Even though voice queries are available on some devices, there are still many cases that semantic and visual intent can hardly be expressed by these descriptions for search. For example, in a common image search task, the user might have already conceived of the general idea of expected pictures such as color configurations and compositions. However, the user usually has to pick up ideal images amidst much more irrelevant results.

In such cases where irrelevant images spoil the results and ruin the user experience, visual-aided tools can largely boost the relevance of search results and the user experience. Let's further consider such a scenario in which the user has no idea of the name of a restaurant but can only describe its particular appearance, such as "a Chinese restaurant with red door, two stone lions, and many red pillars in front;" or even in another totally different situation where the user wants to find

"an oil paint of a man with straw hat." The common thing shared in both situations is that only with a scene or general picture in the user's mind, the user doesn't have the title or name of the target. Such kinds of searches are not easy under present text-based search condition. But with the help of visual aids, which can search for images based on not only text but also image content, these tasks can be much easier. As a result, a powerful image search system with visual aids is desired.

In [10] the authors build a Sketch2Photo system that uses simple text-annotated line sketch to automatically synthesize realistic images. They also employ text and sketch to search for templates which are then stitched on a background to generate a montage. However, their work focuses on image composing instead of image retrieval. Inspired by these works, in this paper [7], a multimodal mobile search system is designed to do visual search.

1.1 Related Work

As a mobile visual search system, the most related works include many multimedia search apps available on mobile devices. Table 1 summarizes the recently developments in mobile phone applications for multimedia search.

As the speech recognition became mature, phone applications using speech recognition rapidly grows recently. The most representative application is Apple Siri [9], which combines speech recognition; natural language understanding and knowledge-based searching techniques.

Table 1. Recent Mobile Visual Search Applications

App	Features
Goggles [20]	product, cover, landmark, namecard
Digimarc Discover [10]	print, article, ads
Point and Find [30]	place, 2D barcode
SnapTell [37]	cover, barcode
SnapNow [26]	MMS, email, print, broadcast
Kooaba [36]	media cover, print

Photo-to-search applications also became pervasive on mobile phones. Such applications enable users to search for what they see by taking a photo on the go. As it is summarized in Table 1, Google Goggles, Point and Find, and Snaptell are good examples in this field. These applications search for the exact partial duplicate images in their database and provide the users with related information of the query images. However, the search is only available for some vertical domains, such as products, landmarks, CD covers, and etc., where the partial duplicate images of the query image have been indexed in their database.

In academic circles, there is not a big difference from industry. Re- searchers for mobile search also focus mainly on photo-to-search techniques. The research topics include visual feature design, database indexing, and transmission. Traditional features such as MPEG-7 image signature, SIFT [8], Speeded Up Robust Feature (SURF) [6], and Oriented FAST and Rotated BRIEF (ORB) are widely used in such visual search systems because of their invariance to illumination, scale and rotation. Moreover, compressed visual descriptors are proposed to accommodate the limited processing speed and narrow bandwidth on the phone. Chandrasekhar et al. discussed their compression as well as proposed a new feature of Compressed Histogram of Gradients (CHoG) [12]. It can quantize and encode gradient histogram with Huffman and Gagie trees to produce very low bit-rate descriptors. Besides, various systems with optimized technique and better indexing are developed to search for landmarks, books, CD covers [11] etc. In [10], different local descriptors are compared in a system of CD cover search. SIFT is widely accepted as the best performed feature and CHoG has an advantage in low-bit transmission. Bernd Girod et al. gave a comprehensive overview

of photo-to-search in [1], from the architecture of an efficient mobile system to the framework of an image recognition algorithm. Other techniques are also used in visual search such as barcodes and OCR.

In both industry and academic circles, it is found that there are few works on mobile image search. As a result, the JIGSAW+ system differs to existing mobile visual search systems in that it represents a new visual search mode by which the mobile users can naturally formulate visual queries to search for images on the go.

II. METHODOLOGIES USED

In this method, a different image representation and indexing technique is introduced. To distinguish from the previous version of JIGSAW, this system is noted with the new techniques as JIGSAW+. Objective is to build a system that is robust with respect to resolution changes, dithering effects, color shifts, orientation, size, and location, not only of the whole image, but of its individual objects as well. An image is first over-segmented into small pieces. In each piece, color moments are extracted. Besides, Scale-Invariant Feature Transform (SIFT) features are extracted in some regions of interest. With large amount of features from different pieces, a codebook is established by performing hierarchical clustering on sampled features. Each feature can be assigned to the nearest clustering center as its code. Thus, each feature can be represented as a word, and a bag-of-word model is initiated. The incidence of different words is used as a factor to calculate image similarity. Moreover, one million Flickr [3] images are indexed in the database so that tag, GPS location and other descriptions are available for the user's further information. The experiment proved that these multi-exemplar search methods outperformed all the other techniques and the system created better user experience in searching for images. The overview of the multimodal visual search system (JIGSAW+) is shown in Fig. 1.

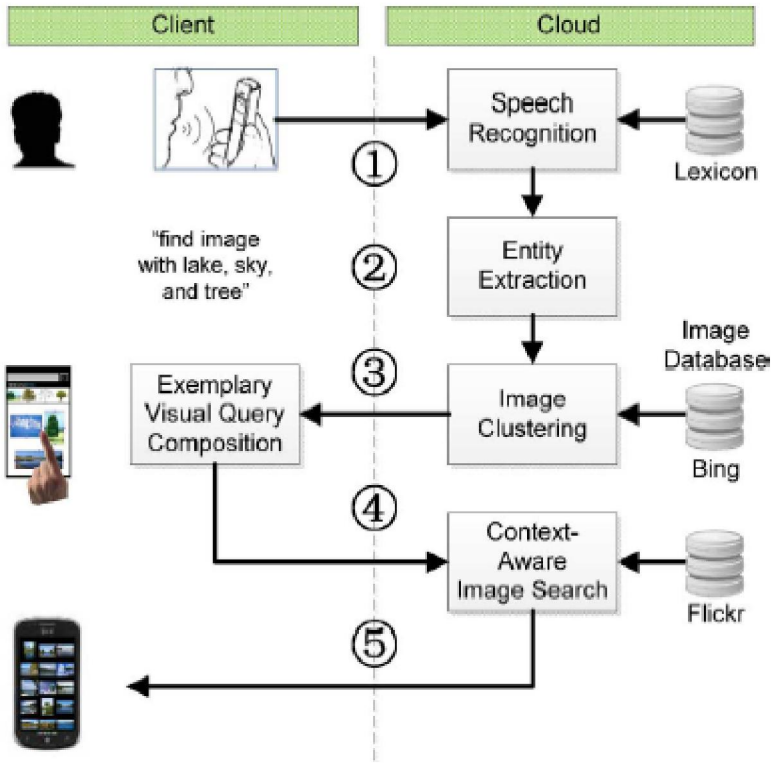

Figure 1. The architecture of proposed multimodal visual search system.

2.1. Entity from Speech

The whole search interaction begins with the natural language understanding. In the JIGSAW+ system, a simple entity extraction strategy is adopted to handle this problem. The entity

extraction from speech can be divided into two steps: 1) speech recognition, and 2) key entity extraction.

After the voice is captured by the phone, it is first sent to a commercial online speech recognition engine to translate the recorded audio stream into a text string. A Hidden Markov Model (HMM) and N-gram-based engine is able to handle both natural sentences and phrase fragments. However, it is preferred to use natural sentences like "find an image with iron tower on the grass" instead of separate phrases like "iron tower, grass". Although a common text-based image retrieval engine can perform better with unnatural phrase segments as "iron tower" and "grass," a whole sentence is easier to be recognized by speech recognizer and much more natural for the user. Actually, an entire sentence is also easier to be recognized by speech engine because the speech engine uses Markov-based algorithm to recognize the words within audio stream, a complete sentence can be recognized with higher accuracy. For example, it's hard to decide whether to choose "abbey" or "obey" if only a single word is uttered. But within a sentence, the language model can choose the suitable words according to the context and part of speech.

In the JIGSAW+ system, once the text is available, the core entities within the text will be extracted. Usually, such word extraction is done by either key word extraction or meaningless word filter. These strategies are troublesome and uncertain. A simple way to extract only meaningful words in the sentence is adopted. A dictionary can be first automatically established according to WordNet, which is a famous published hierarchical word database disclose the hierarchical relationships among English words.

2.2 Visual Query Formulation

Different from traditional text-based image search system, in the JIGSAW+ image search system, the user can use a visual query to further articulate the user's visual intent. In the JIGSAW+ system, to formulate a visual query, the user need only to choose one or more exemplary images in our gallery to compose a collage according to the user's visual intent. Because the user doesn't have proper exemplary images on the phone, the system will first automatically suggest some visual exemplars according to the extracted keywords from the user's speech. Limited to the wireless bandwidth and small screen, the exemplary images cannot be too many. It is impractical to allow users to manually select exemplary images in a lot of candidates queried from the Internet on the phone. And it is unreasonable to give only several top images by text retrieval. As a result, clustering is better to be applied to the images retrieved by each keyword. Many images are clustered by their content and only a few cluster centers will be provided to the user as visual exemplary images. The user will then make a collage-like composite image using some favorable exemplary images as pieces of building blocks. Compared with line sketch and color pattern, the JIGSAW+ search modal can be sometimes an easier complement so that the user intent can be better transferred. Fig. 2 demonstrates the chosen few of exemplary images.

2.2.1. Visual Exemplar Generation

A visual exemplar database should be ready to provide exemplary images to the user according to their keywords. As a result, plenty of exemplary images are generated offline. Given a keyword, a set of images are first downloaded from the Internet by inquiring a image search engine. Usually, top results from a Internet search engine for simple concepts are reliable than many other resources (e.g., Flickr). It is observed that the top results always well cover the concepts, so the top 200 images are used to abstract representative images. These images are then clustered into several groups according to their visual content. The influence of the background in images can seriously damage the performance of the content-based clustering, so the background should be removed before clustering. Since the candidate visual exemplars are queried with single concept, it is assumed that these images are with single objects and simple backgrounds. As a result, a simple salient region

detector is applied to these candidate exemplars to locate the object area. To further remove the influence of background and to precisely pick out the foreground part, GrabCut is adopted to segment the foreground object from the image. To describe the foreground object, bag-of-word (BOG) model is used. Finally, a visual words histogram is established for an image using the bag-of-word model. The value of each entry is the sum of weights of corresponding visual words.

With this vector representation, the cosine similarity is calculated between each pair of images:

$$sim(f1, f2) = \frac{(f1, f2)}{\sqrt{(f1, f1)(f2, f2)}} \quad \ldots (2.1)$$

Where *(f1, f2)* means the inner product of two image vectors *f1* and *f2*.

Using this similarity measure, affinity propagation (AP) algorithm [4] is adopted to cluster the candidate images into several groups. AP is an unsupervised clustering method which can group data automatically into moderate number of classes. In this implementation, this usually generates about 20 clusters. The obtained clusters are then sorted according to the size of the group in descending order. The centers of the top clusters are selected as exemplars for this keyword. Fig. 2 shows some exemplary images generated by the clustering process.

Fig. 2 describes an example of processing a visual query with multiple exemplars C_q from an initial query $T = \{T_q\}^k_{q=1}$ with multiple entities T_q. Each entity corresponds to one exemplary image in the composite visual query. There are two recognized entities: T1 = "sky" and T2 = "grass". For each entity, there is a list of candidate exemplary images. The user can select any image in each list. The selected exemplary images are denoted as I1 and I2.

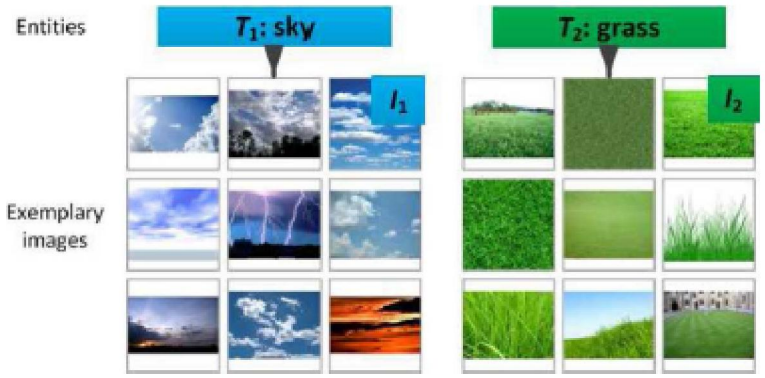

Figure 2. An example of processing a visual query with multiple exemplars.

Because it will take a while to download candidate images from the Internet, the visual exemplar generation is performed offline. A list of exemplars of possible keywords can be kept in the initial database, as stated in Section of Entity Extraction from Speech, while the list can also dynamically grow with user's demands.

2.2.2 Composite Visual Query Generation

For each keyword, the system offers to the user with several exemplary images according to the clustering results. The user can choose one image under each keyword and put it onto the canvas region on the phone screen. The user can both reposition and resize the exemplary images until satisfied. Fig. 3 gives some examples of the composite visual query with different number of exemplary images. Take the second visual query for example; composite visual query is made up by the selected image I1 and I2, where the position and the size (R1 and R2) of each exemplary image

on the canvas are adjusted by the user. Finally, by choosing and positioning single or multiple exemplary images, a composite visual query is generated by the user. Each sub-image in the query is named as a visual exemplar. As shown in Fig. 3, each exemplar carries the information of position R_q represented by its center's relative coordinate $x_q = (x_q, y_q)(0 <= x_q, y_q <= 1)$, and its size its width and height $(w_q, h_q)(0 <= w_q, h_q <= 1)$. Together with text T_q and exemplary image I_q, the visual exemplar is a triplet $C_q = \{T_q, I_q, R_q\}$. The whole query is a set of K exemplar(s) $Q = \{C_q\}^k_q = 1$.

Figure 3. Some examples of the visual queries.

III. VISUAL SEARCH BY MULTIPLE EXEMPLARS

3.1 Problem Formulation

The objective of visual search by multiple exemplars is to search for images using composite query $Q = \{C_q\}$. The retrieved images should be consistent with the query in following standards:

1. The image should contain the entities in the user's query.
2. The entities in the image should be similar to the exemplary images the user chooses.
3. The position of single entity or the relative layout of multiple entities in the image should consist with the composite visual query.

The images are first segmented into irregular regions as did in another image retrieval framework. A bag-of-word model is used to represent the regions. Then a multi-exemplar region-based image search system is developed to solve the problem. The relative position is considered in the search process instead of absolute position. And a better merging scheme is used to achieve more reasonable results. The rest of this section describes the details of the JIGSAW+ visual search method.

3.2 Visual Representation of Images

As a content-based image retrieval system, a series of feature should be first extracted from images, where color features and texture features are widely used. Instead of the sensitive absolute

position matching in previous papers, a strategy to consider relative positions between exemplars is also developed.

3.2.1 Color Feature Extraction:

Before color feature extraction, the images are over-segmented using algorithm by Felzenszwald and Huttenlocher described in [2]. This algorithm has been widely used in image content analysis. This algorithm is easy to use and has advantage in speed, though other methods may bring about better results. Different from object segmentation, these over-segmentation methods use graph based algorithms that segments an image into many homogeneous regions. In Felzenszwald and Huttenlocher's algorithm, each node in the graph stands for a pixel in the image, with undirected edges connecting its adjacent pixels in the image. The weight of each edge between two pixels reflects their similarity. The partition is carried out by merging two segments if the dissimilarity between them is less than the dissimilarity inside either of them. In this implementation, the recommended parameters are used, such that k = 500, Amin = 20 and σ = 0.5. A Gaussian smoothing is applied to the image with the parameter of σ. Parameter k is how dissimilar each piece can be. Pieces that are smaller than Amin pixels are merged. Moreover, the similarity of RGB-color space is used instead of gray level, so that inside each piece the color is close. After over-segmentation, color moments and positions are extracted for each piece. Although color histogram is widely used as image feature and proves effective, in our case, since the color within each piece is almost the same, it is assumed that it conform a Gaussian distribution. As a result, it is enough to take the mean, covariance and 3rd order moment of each channel of RGB color within the piece as its color descriptor. Thus each piece is represented in a 9-dimensional vector. In addition, since the position of the region is crucial for the following retrieval process, the center coordinate is also obtained for each piece. Besides, the size of each piece is considered as the weight of the corresponding descriptor for the image.

3.2.2 Texture Feature Extraction:

Beside the homogeneous regions, some unique texture regions are also extracted as secondary feature. The most widely used local feature of SIFT are used to capture local texture patterns in the image. To reduce computing and noise, only prominent regions are used instead of homogeneous regions used in color feature extraction, or grid-based dense sampling. Prominent regions are detected using Difference of Gaussian (DoG) detector. By doing this, the size of each region can be also optimized via scale selection. For speed consideration, features with small scales are dropped which are not always reliable. In hope of better grasping the texture information and reducing false matches, the scale is enlarged by three times when extracting SIFT features. Its area is also used as the weight for each SIFT descriptor.

3.2.3 Descriptor Quantization:

Bag-of-word model is used for both color and SIFT features to handle the uncertainty of the number of descriptors in the image. Codebooks are generated separately for color and SIFT feature, so that each descriptor is quantized to a visual code. One million features are sampled from Flickr images and clustered them into N clusters using hierarchical k-means. The cluster centroids form the codebook with N visual words. After all the descriptors are assigned to their nearest visual words, an image is represented by a series of visual words. Finally, both color and texture features are concatenated into one feature vector with their weights voted.

3.3 Multiple-Exemplar Relevance

After the composite visual query is submitted by the user, a search process starts at the server end. Images are favored which contain all the exemplars as well as spatially consist with the exemplars' geometry structure.

In the first stage, all the images in the database are matched with the exemplary images. Following the popular image search scheme, visual word and histogram intersection are used to measure the similarity of images. For each exemplar, the image visual word histogram is first normalized by its percentage of exemplar area on the canvas, so that $||f_q||_1 = w_q h_q$. For an image I in the database, the histogram f is normalized so that $||f||_1 = 1$. The existence of each visual word is evaluated by the intersection of f_q and f, yielding an ingredient vector:

$$s = f_q \cap f \quad \ldots (3.1)$$

which means, in each dimension, the component in s is assigned by the minimum component in f_q and f. By accumulating the all the components in s, the existence score is obtained as:

$$S_q = \sum_{n=1}^{N} s_n \quad \ldots (3.2)$$

where s_n are components in ingredient vector s.

Indeed, for one entity, single exemplary image can be extended to multiple exemplary images if multiple-exemplar selecting is enabled for one entity, or the system automatically uses multiple exemplary images in candidate exemplary images. While the position and size of one entity is fixed on the canvas, multiple exemplary images can be selected that better represent the entity instead of a single exemplary image. For example, the user can select both red sky and blue sky as exemplary images for one entity "sky" at the same time. In this situation, the system just goes through all the provided exemplars and uses the best existence score.

After the existence scores for all exemplars are obtained for image I, these scores are merged by their geometric mean:

$$s = \sqrt[Q]{\prod_{q=1}^{Q} S_q} \quad \ldots (3.3)$$

Where q = 1…Q, and Q is the total number of the entities. This merging scheme is used because it imposes a strong coexistence requirement so that images with all high existence are preferred. If one of the existence score sq is small, the merging yields a small value as well.

Beside the existence, the existed position of each exemplar in image I should also be consistent with the composite visual query. The position of occurrence of each exemplar is used to check the spatial consistency. The relative center position of each visual word component in image I is saved as $0 <= y_n <= 1 (n = 1,\ldots,N)$. Thus, the position of the object can be estimated by

$$y = \sum_{n=1}^{N} \frac{s_n y_n}{S_q} \quad \ldots (3.4)$$

For visual query with single entity, the distance between exemplar position x and object position y is used to obtain a measure of spatial similarity

$$p = exp\{-||x - y||_2\} \quad \ldots (3.5)$$

If the object occurs in the identical position as the user demand, this spatial similarity will get the maximum value of 1, which means the position is exact the same. If the object is far from the demanded location, the spatial similarity is small.

While the query contains multiple exemplars, the spatial consistency between each pair of exemplars is calculated by the cosine of the angle between two pairs. For a pair of entities i and j, the spatial consistency is obtained by

$$p_{i,j} = cos(x_i - x_j, y_i - y_j) \quad \ldots (3.6)$$

The formula measures the relative spatial consistency instead of absolute occurrence positions, which is continuous and stable to the variance of query's layout and, in many cases, better represent the user's intent. The smallest value of spatial consistencies among all pairs is used as a final penalty coefficient $p = \min_{i,j}(p(i, j))$.

Finally, the visual similarity between the image and visual query is combined by:

$$sim_v(Q, I) = p.s \quad \ldots (3.7)$$

This similarity reflects both the existence and the spatial consistency of all the exemplary images.

3.4 Indexing and Retrieval

Million-scale images are indexed by the extracted visual words in a reverted index. Each image is represented as a series of visual codes first, and then a table mapping visual codes to images is constructed. The same N visual codes are followed by N lists of entries. Each entry keeps the information of a unique piece of an image including image ID, center point location and weight.

Since the amount of visual similar but semantic dissimilar images is overwhelming and hard to control, the keywords are used first to mark all images that are relevance to any keyword by text-based retrieval. Many web images like Flickr images have text information like tags, titles, and descriptions. A text stemmer is used to remove the commoner morphological and in flexional endings from words. The stemmed tags are indexed into an inverted-file index with tf - idf (term frequency-inverse document frequency) weights, so that we can obtain another similarity between image and query from text information as $sim_t(Q, I)$. The similarity between Q and I can be calculated as follows:

$$sim(Q, I) = sim_v(Q, I).sign(sim_t(Q, I)) \quad \ldots (3.8)$$

Where sign(x) is a binary function which is 1 if x > 0, otherwise 0. Finally, sim(Q, I) is used to rank images in our database and show the top-ranked images in the result page. However, if there are too few text results, the visual similarity $sim_t(Q, I)$ can be directly used to rank the images.

Algorithm 1: The multi-part region-based matching algorithm

1. Similarities and positions: Retrieve the similarity for each exemplary image in the reverted file; estimate the position by averaging the positions of matched visual words.
2. For multi-exemplar entities, find the max similarity.
3. Merge the similarities from different entities by eq. 3.3.
4. Obtain the special consistency score by eq. 3.5 and eq. 3.6.
5. Combine the merged similarity and position consistency.

IV. CONCLUSION

In this paper, an interactive mobile visual search system is introduced which allows the users to formulate their search intent through natural multimodal interactions with mobile devices. The system represents the first study on mobile visual search by taking the advantages of multimodal and multitouch functionalities on the phone. The JIGSAW+ provides a game-like interactive image search scheme with composition of multiple exemplars. The visual query generated by the user can be effectively used to retrieve similar images by this method.

REFERENCES

[1] D. Chen, N. Cheung, R. Grzeszczuk, Y. Reznik, G. Takacs, S. Tsai, B. Girod, V. Chandrasekhar and R. Vedantham, "Mobile visual search", IEEE Signal Process. Mag., 2011.

[2] P. Felzenszwalb, and D. Huttenlocher, "Efficient graph-based image segmentation", Massachusetts Institute of Technology, Cornell University, 2004.

[3] Flickr. http://www.ickr.net/.

[4] B. Frey and D. Dueck, "Clustering by passing messages between data points", Department of Electrical and Computer Engineering, University of Toronto, 10 Kings College Road, Toronto, Ontario M5S 3G4, Canada., 2007.

[5] M. Gales and S. Young, "The application of hidden markov models in speech recognition", Foundations and Trends in Signal Processing, Vol. 1, 2008.

[6] T. Tuytelaars, H. Bay and L. Van Gool, "SURF: Speeded-up robust features", Proc. ECCV, Belgium, 2008.

[7] Tao Mei, Jingdong Wang, Houqiang Li, Yang Wang and Shipeng Li, "Interactive multimodal visual search on mobile device", IEEE transactions on multimedia, 2013.

[8] David G. Lowe, "Distinctive image features from scale-invariant keypoints", International Journal of Computer Vision, 2004.

[9] Siri. http://www.apple.com/iphone/features/siri.html.

[10] P. Tan, A. Shamir, T. Chen, M.-M. Cheng and S. M. Hu, "Sketch2photo: Internet image montage", 2009.

[11] A. Lin, G. Takacs, S. S. Tsai, N. M. Cheung, Y. Reznik, R.Grzeszczuk, V. Chandrasekhar, D. M. Chen and B.Girod, "Comparison of local feature descriptors for mobile visual search", Proc. IEEE Int. Conf. Image Process., 2010.

[12] D. Chen, S. Tsai, R. Grzeszczuk, V. Chandrasekhar, G. Takacs and B. Girod, "CHoG: Compressed histogram of gradients a low bit-rate feature descriptor", in Proc. IEEE Conf. Comput. Vis. Pattern Recogn, 2009.

[13] J. Wang, H. Li, Y.Wang, T. Mei and S. Li, "JIGSAW: Interactive mobile visual search withmultimodal queries", in Proc. ACMMultimedia, 2011.

