Anthony Vetro
Mitsubishi Electric Research Labs

Mobile Visual Search: Architectures, Technologies, and the Emerging MPEG Standard

Bernd Girod and Vijay Chandrasekhar
Stanford University

Radek Grzeszczuk
Nokia Research Center

Yuriy A. Reznik
Qualcomm

Modern-era mobile phones and tablets have evolved into powerful image- and video-processing devices, equipped with high-resolution cameras, color displays, and hardware-accelerated graphics. They are also equipped with location sensors (GPS receiver, compass, and gyroscope), and connected to broadband wireless networks, allowing fast information transmission and enabling a class of applications that use the phone's built-in camera to initiate search queries about objects in the user's visual proximity (see Figure 1). Such applications can be used, for example, for identifying and comparing products and for finding information about movies, CDs, real estate, printed media, or art. First deployments of mobile visual-search systems include Google Goggles (see http://www.google.com/mobile/goggles/), Nokia Point and Find (see http://www.pointandfind.nokia.com), Kooaba (see http://www.kooaba.com), and Snaptell (see http://www.snaptell.com).

Mobile image-based retrieval applications pose a unique set of challenges. What part of the processing should be performed on the mobile client, and what part is better carried out at the server? On the one hand, you can simply transmit a JPEG image and offload the rest of the processing to the server. But an image transmission of this kind could take anywhere from a few seconds to a minute or more over a slow wireless link. On the other hand, image analysis, extraction of salient image features, and even full image-based retrieval from a small database can be done now on mobile devices in a much shorter time.

In Figure 2, we show three possible client–server architectures.

- In Figure 2a, the mobile device transmits a query image to the server. Image-based retrieval is carried out entirely on the server, including an analysis of the query image.
- In Figure 2b, the mobile device analyzes the query image, extracts features, and transmits feature data. The retrieval algorithms run on the server using the transmitted features as the query.
- In Figure 2c, the mobile device maintains a cache of the database and performs image matching locally. Only if a match is not found, the mobile device sends a query request to the server.

In each case, the retrieval framework must adapt to stringent mobile system requirements. The processing on the mobile device must be fast and economical in terms of power consumption. The size of the data transmitted over the network must be as small as possible to minimize network latency and thus provide the best user experience. The algorithms used for retrieval must be scalable to potentially very large databases, and capable of delivering accurate results with low latency. Further, the retrieval system must be robust to allow reliable recognition of objects captured under a wide range of conditions,

Editor's Note

Sophisticated visual-search and augmented-reality applications are beginning to emerge on mobile platforms. Such applications impose unique requirements on latency, processing, and bandwidth, and also demand robust recognition performance. This article describes a new standard that is being developed to address the functionality and interoperability needs for mobile visual-search applications.

—Anthony Vetro

1070-986X/11/$26.00 © 2011 IEEE

Published by the IEEE Computer Society

including different distances, viewing angles, and lighting conditions, or in the presence of partial occlusions or motion blur.

Mobile image-based retrieval technologies

Most successful algorithms for image-based retrieval today use an approach that is referred to as *bag of features* (BoF) or *bag of words* (BoW).[1,2] The BoW idea is borrowed from text document retrieval. To find a particular text document, such as a webpage, it's sufficient to use a few well-chosen words. In the database, the document itself can likewise be represented by a bag of salient words, regardless of where these words appear in the document. For images, robust local features that are characteristic of a particular image take the role of visual words. As with text retrieval, BoF image retrieval does not consider where in the image the features occur, at least in the

Figure 1. A snapshot of an outdoor mobile visual-search system. The system augments the viewfinder with information about the objects it recognizes in the image taken with a phone camera.

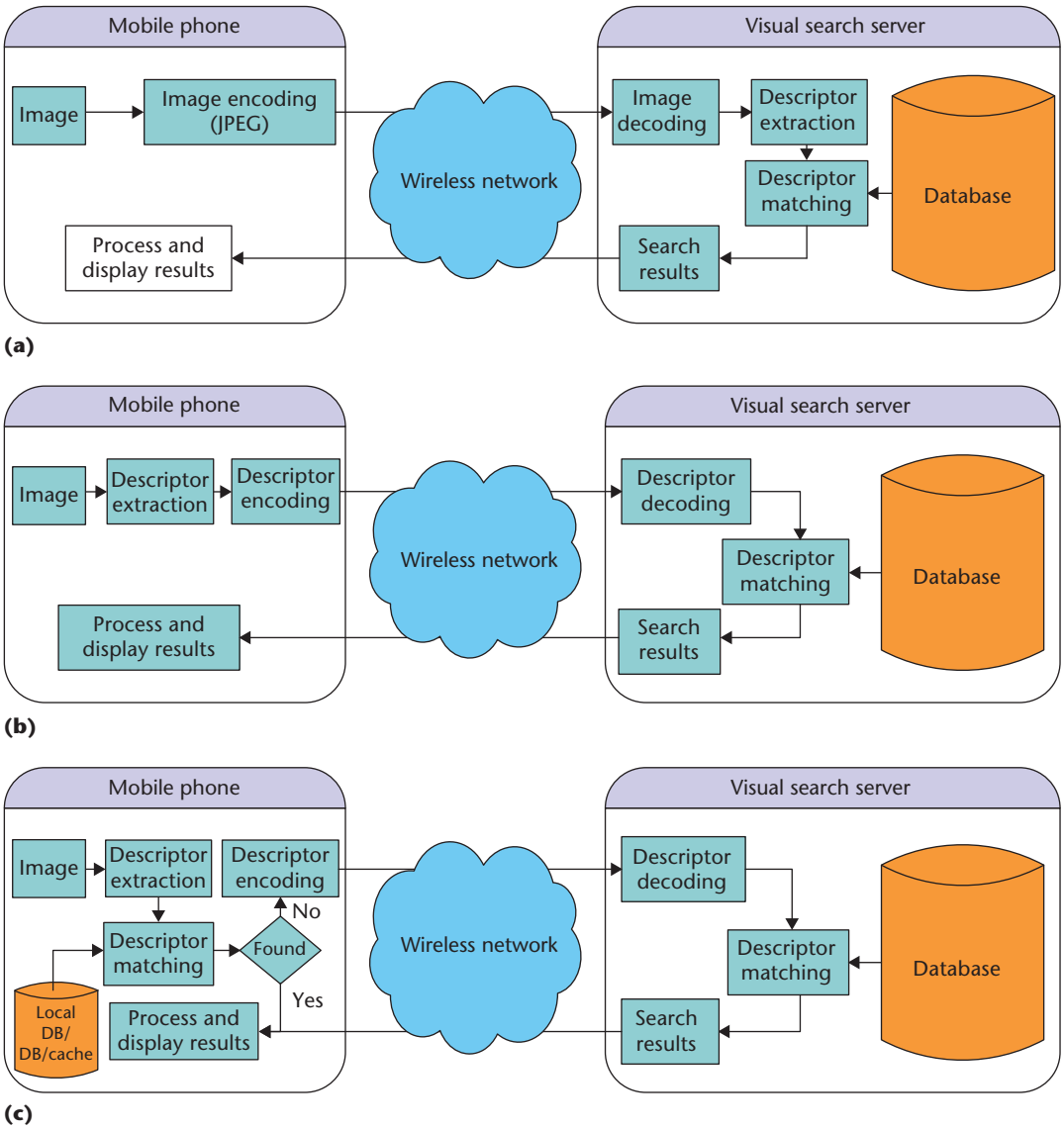

Figure 2. Mobile visual search architectures. (a) The mobile device transmits the compressed image, while analysis of the image and retrieval are done entirely on a remote server. (b) The local image features (descriptors) are extracted on a mobile phone and then encoded and transmitted over the network. Such descriptors are then used by the server to perform the search. (c) The mobile device maintains a cache of the database and sends search requests to the remote server only if the object of interest is not found in this cache, further reducing the amount of data sent over the network.

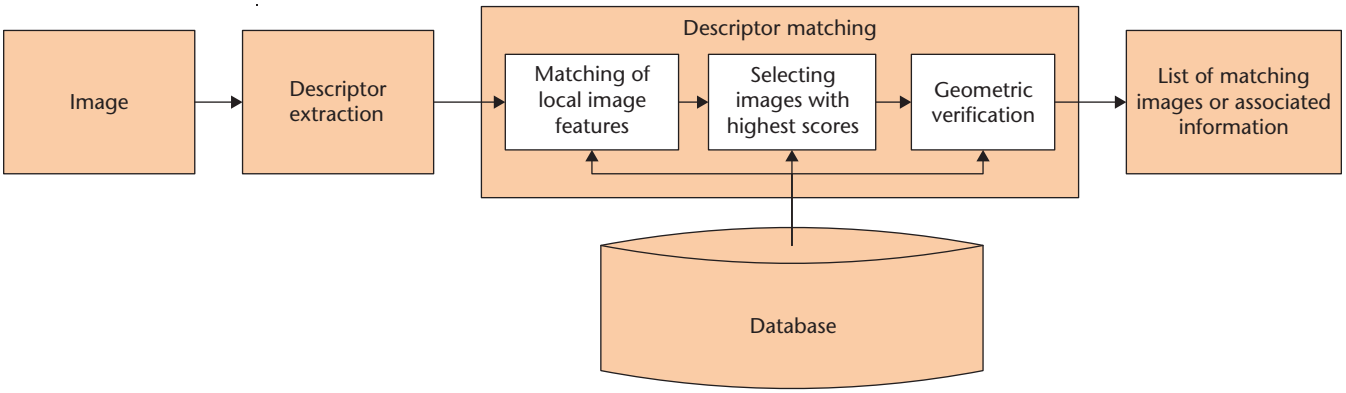

Figure 3. Pipeline for image retrieval. First, local image features and descriptors are extracted from the query image. Such descriptors are then matched against descriptors of images stored in the database. The images that have many features in common with the query image are then selected. The geometric verification step rejects matches with feature locations that cannot be plausibly explained by a change of viewing position.

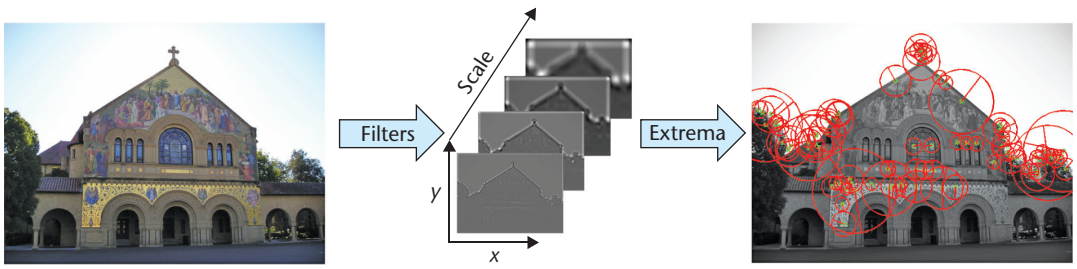

Figure 4. Interest point detection. (a) Starting with a query image, to achieve scale invariance, interest points (blobs and extrema points) are typically detected at different scales using (b) a scale-space pyramid. In (c) red circles indicate the scale of each key point, and connecting lines show dominant directions of gradients.

initial stages of the retrieval pipeline. However, the variability of features extracted from different images of the same object makes the problem much more challenging.

A typical pipeline for large-scale image-based retrieval is shown in Figure 3. First, local features (or descriptors) are extracted from the query image. The set of local features is used to assess the similarity between query and database images. To be useful for mobile search applications, individual features should be robust against geometric and photometric distortions encountered when the user takes the query photo from a different viewpoint. Next, query features are matched to features of images stored in the database. Typically, this is accomplished using special index structures, allowing fast access to lists of images containing matching features. On the basis of the number of features they have in common with the query image, a short list of potentially similar images is selected from the database. To these images, further examination is applied, including a geometric verification step that looks for a coherent spatial pattern between features of the query image and the features of the candidate database image to ensure that the match is correct.

We next discuss each block in the image-retrieval pipeline in more detail, focusing on algorithms suitable for use in mobile client–server recognition systems. For a more in-depth look at these technologies, interested readers are referred to our recent review paper.[3]

Feature extraction

The feature-extraction process typically starts by finding salient interest points in the image. For robust image matching, such interest points need to be repeatable under perspective transformations (such as scale changes, rotation, and translation) and lighting variations. To achieve scale invariance, interest points are typically computed at multiple scales using a scale-space pyramid (see Figure 4).

Computing Compressed Histogram of Gradients Image Features

The patches around interest points at different scales are normalized and oriented along the most dominant gradient. To achieve high discriminating ability, each patch is divided into several spatially localized bins, as shown in Figure A. The joint (d_x, d_y) gradient histogram in each spatial bin is then computed. The compressed histogram of gradients (CHoG) histogram binning exploits the typical skew in gradient statistics that are observed for patches extracted around key points. Finally, histograms of gradients from each spatial bin are quantized and compressed to obtain a highly compact feature descriptor. To assess similarity between CHoG features, information distance measures, such as Kullback-Leibler divergence or Jeffrey divergence, can be used.

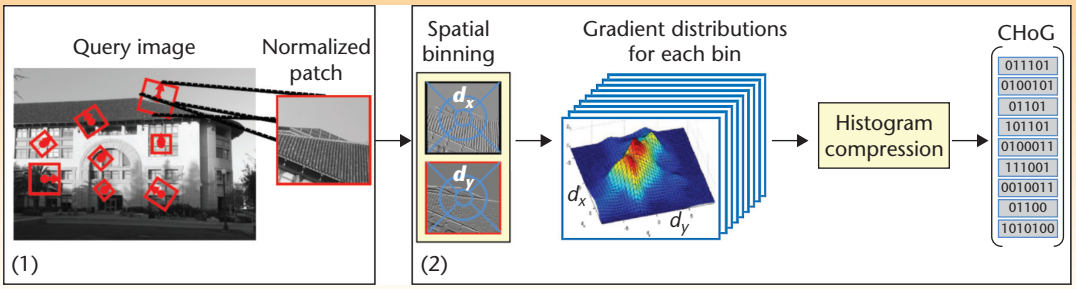

Figure A. (1) Patch extraction computation of a (2) compressed feature descriptor.

To achieve rotation invariance, the image patch around each interest point is oriented in the direction of the dominant gradient. The gradients in each patch are further normalized to make them robust to illumination changes. Such normalized patches are then used to compute visual word descriptors suitable for search.

Several robust feature descriptors have been proposed in the literature. Best known is the Scale Invariant Feature Transform (SIFT) algorithm, introduced by Lowe in 1999.[4] SIFT employs a difference-of-Gaussian image pyramid to find interest points, and produces 128 dimensional vectors of parameters of gradient distributions in each patch as feature descriptors. Another popular design is Speeded Up Robust Features (SURF), by Bay et al.[5] SURF uses a simpler Hessian blob detector and allows several additional optimizations, making it feasible to compute in near-real-time on mobile devices.[6]

Both SIFT and SURF descriptors show good discriminating properties. However, they are not compact. For example, a set of SIFT descriptors for 1,000 features in a query image requires about 128 Kbytes. This is comparable to the size of a query image compressed by JPEG.

For mobile visual-search systems that transmit or locally store image features, more compact descriptors are needed. The compressed histogram of gradients (CHoG) descriptor[7] offers one such design. CHoG is easy to compute and uses only about 60 bits per feature, on average. In other words, it's over 17 times smaller than SIFT. Nevertheless, its retrieval performance is as good as or better than SIFT. Further details about CHoG can be found in the "Computing Compressed Histogram of Gradients Image Features" sidebar and in the original paper.[7]

Feature indexing and matching

For a large database of images, matching the query image against every database image using pairwise feature-matching methods is infeasible. Instead, it's customary to use a data structure that returns a shortlist of the database candidates most likely to match the query image. The shortlist might contain false positives, as long as the correct match is included. Slower pairwise comparisons can subsequently be performed on just the shortlist of candidates rather than the entire database.

Many data structures have been proposed for indexing local features in image databases. Lowe uses approximate nearest neighbor search of SIFT descriptors with a best-bin-first strategy.[4] Sivic and Zisserman use a BoF model. The BoF codebook is constructed by *k*-means clustering of a training set of descriptors.

Vocabulary Tree-Based Retrieval

A vocabulary tree (VT) for a particular database is constructed by performing hierarchical k-means clustering on a set of training feature descriptors representative of the database. Initially, k large clusters are generated for all the training descriptors. Then, for each large cluster, k-means clustering is applied to the training descriptors assigned to that cluster, to generate k smaller clusters. This recursive division of the descriptor space is repeated until there are enough bins to ensure good classification performance. Figure B1 shows a VT with only two levels, branching factor $k = 3$, and $3^2 = 9$ leaf nodes. In practice, VT can be much larger, for example, with height 6, branching factor $k = 10$, and containing $10^6 = 1$ million nodes.

The associated inverted index structure maintains two lists for each VT leaf node, as shown in Figure B2. For a leaf node x, there is a sorted array of image identifiers $i_{x1}, \ldots, i_{xNx}$ indicating which N_x database images have features that belong to a cluster associated with this node. Similarly, there is a corresponding array of counters $c_{x1}, \ldots, c_{xNx}$ indicating how many features in each image fall in same cluster.

During a query, the VT is traversed for each feature in the query image, finishing at one of the leaf nodes. The corresponding lists of images and frequency counts are subsequently used to compute similarity scores between these images and the query image. By pulling images from all these lists and sorting them according to the scores, we arrive at a subset of database images that is likely to contain a true match to the query image.

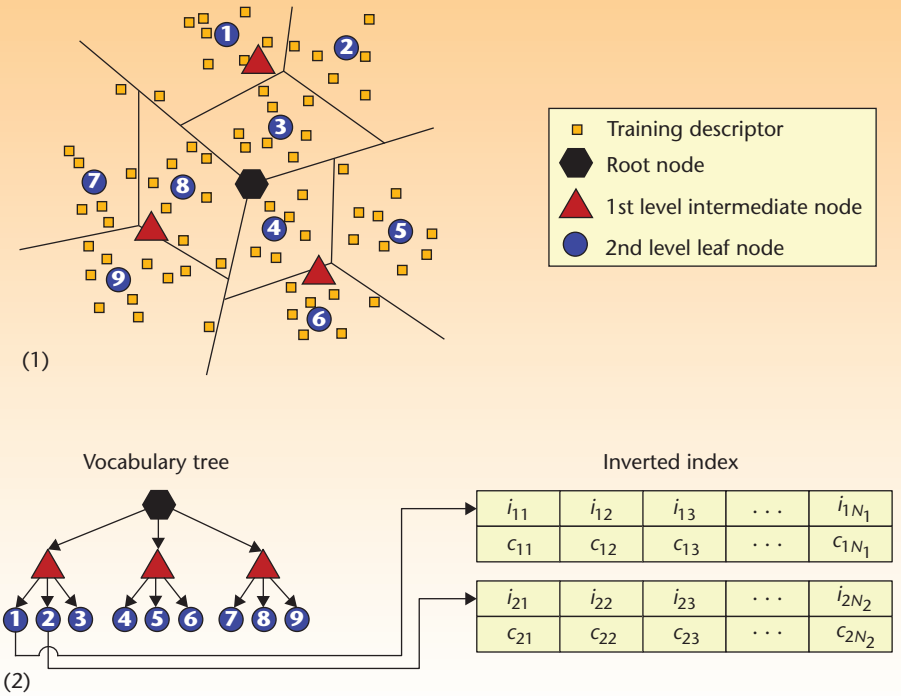

Figure B. (1) Vocabulary tree and (2) inverted index structures.

During a query, scoring the database images can be made fast by using an inverted index associated with the BoF codebook. To generate a much larger codebook, Nister and Stewenius use hierarchical k-means clustering to create a vocabulary tree (VT).[2] Additional details about a VT can be found in the ''Vocabulary Tree-Based Retrieval'' sidebar. Several alternative search techniques, such as locality-sensitive hashing and various improvements in tree-based approaches, have also been developed.[8-11]

Geometric verification

Geometric verification typically follows the feature-matching step. In this stage, location information of features in query and database images is used to confirm that the feature matches are consistent with a change in viewpoint between the two images. This process is illustrated in Figure 5. The geometric transform between the query and database image is usually estimated using robust regression techniques such as random sample consensus

(Ransac)[12] or the Hough transform.[4] The transformation is often represented by an affine mapping or a homography.

The geometric verification tends to be computationally expensive, which is why it's only used for a subset of images selected during the feature-matching stage. Several additional techniques can be used to speed up this process further. Other work investigates how to optimize steps in Ransac.[13] Jegou et al. use weak geometric consistency checks based on feature-orientation information.[9] Yet another technique, employing a fast geometric reranking step before the Ransac, is described in other work.[14]

Figure 5. Geometric verification: pairwise feature matches are checked for consistency with a geometric model. True feature matches are shown in red. False feature matches are shown in green.

Performance of mobile visual-search systems

What performance can we expect for a mobile visual-search system using today's commonly available wireless networks, mobile phones, and servers? To answer this question, we examine the experimental Stanford Product Search (SPS) system,[6] which can be configured to operate in two modes, corresponding to the client–server architectures shown in Figures 2a and 2b. In *send image* mode (architecture in Figure 2a), we transmit the query image to the server and all operations are performed on the server. In *send features* mode (architecture in 2b), we process the query image on the phone and transmit compressed query features to the server.

Figure 6. Example image pairs from the dataset. A clean database picture (top) is matched against a real-world picture (bottom) with various distortions.

For evaluation, we use a database of one million CD, DVD, and book cover images, and a set of 1,000 query images with a range of photometric and geometric distortions.[6] We show examples of such images in Figure 6. For the client, we use a Nokia 5800 mobile phone with a 300-MHz CPU. For the recognition server, we use a Linux server with a Xeon E5410 2.33-GHz CPU and 32 Gbytes of RAM. We report results for both 3G and WLAN networks. For 3G, the data-transmission experiments are conducted in an AT&T 3G wireless network, averaged over several days, with a total of more than 5,000 transmissions at indoor locations where such an image-based retrieval system would typically be used.

In Figure 7 (next page), we compare retrieval accuracy achieved by different schemes. When we send the image, we set different quality levels of JPEG to achieve different rates. When we transmit SIFT or CHoG image features we achieve different rates by varying numbers of features transmitted. SIFT features are uncompressed at 1,024 bits, while CHoG features are compressed, using about 60 bits each. We define classification accuracy as the percentage of query images correctly retrieved as a top match. We observe that in highest rate modes, we approach 96 percent classification accuracy for our challenging query images and database. We also note that transmission of CHoG features significantly outperforms the other schemes in terms of data usage.

We show end-to-end latency measured for different networks and different modes of the SPS system in Figure 8. When communication is done over Wi-Fi, transmission of images is fast, and most time is spent on recognition done on the server. However, when communication is done over 3G, network latency becomes the bottleneck. In this situation, there is a significant benefit in sending compressed features, which reduces system latency

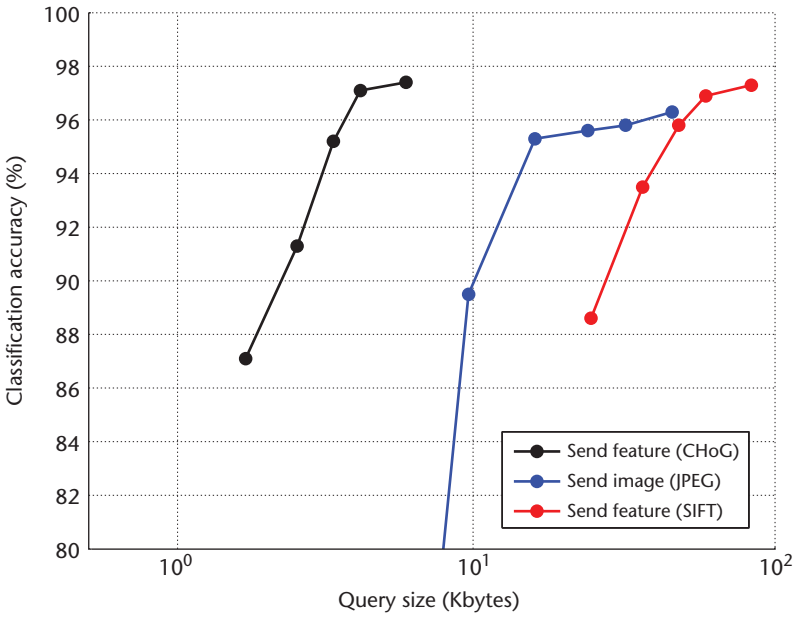

Figure 7. Comparison of different schemes with regard to classification accuracy and query size. CHoG descriptor data is an order of magnitude smaller compared to JPEG images or uncompressed SIFT descriptors.

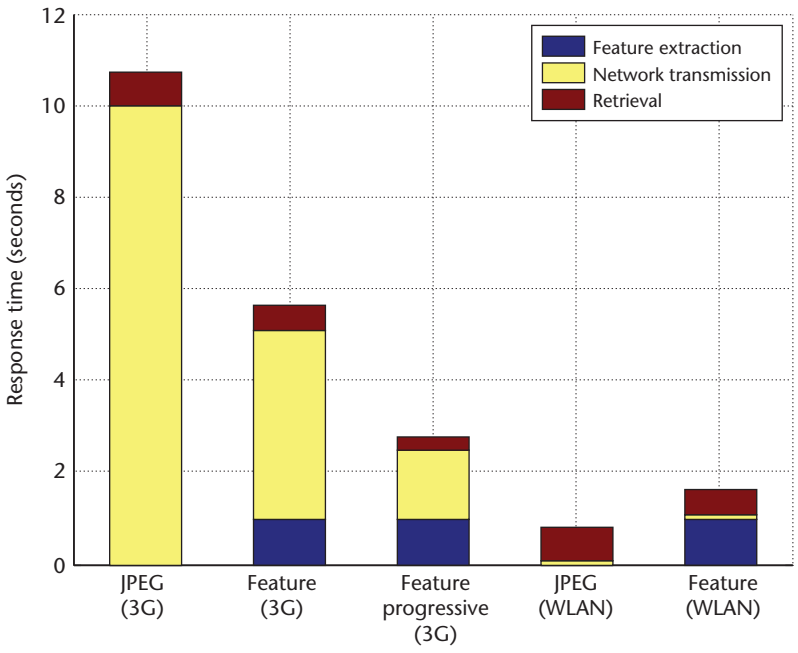

Figure 8. End-to-end latency for different schemes. Compared to a system transmitting a JPEG query image, a scheme employing progressive transmission of CHoG features achieves approximately four times the reduction in system latency over a 3G network.

by approximately a factor of two. Moreover, transmission of features allows yet another optimization: it's possible to use progressive transmission of image features, and let the server execute searches on a partial set of features, as they arrive.[15] Once the server finds a result that has sufficiently high matching score, it terminates the search and immediately sends the results back. The use of this optimization reduces system latency by another factor of two.

Overall, the SPS system demonstrates that using the described array of technologies, mobile visual-search systems can achieve high recognition accuracy, scale to realistically large databases, and deliver search results in an acceptable time.

Emerging MPEG standard

As we have seen, key component technologies for mobile visual search already exist, and we can choose among several possible architectures to design such a system. We have shown these options at the beginning, in Figure 2. The architecture shown in Figure 2a is the easiest one to implement on a mobile phone, but it requires fast networks such as Wi-Fi to achieve good performance. The architecture shown in Figure 2b reduces network latency, and allows fast response over today's 3G networks, but requires descriptors to be extracted on the phone. Many applications might be accelerated further by using a cache of the database on the phone, as exemplified by the architecture shown in Figure 2c.

However, this immediately raises the question of interoperability. How can we enable mobile visual search applications and databases across a broad range of devices and platforms, if the information is exchanged in the form of compressed visual descriptors rather than images? This question was initially posed during the Workshop on Mobile Visual Search, held at Stanford University in December 2009. This discussion led to a formal request by the US delegation to MPEG, suggesting that the potential interest in a standard for visual search applications be explored.[16] As a result, an exploratory activity in MPEG was started, which produced a series of documents in the subsequent year describing applications, use cases, objectives, scope, and requirements for a future standard.[17]

As MPEG exploratory work progressed, it was recognized that the suite of existing MPEG technologies, such as MPEG-7 Visual, does not yet include tools for robust image-based retrieval and that a new standard should therefore be defined. It was further recognized

Table 1. Timeline for development of MPEG standard for visual search.

When	Milestone	Comments
March, 2011	Call for Proposals is published	Registration deadline: 11 July 2011 Proposals due: 21 November 2011
December, 2011	Evaluation of proposals	None
February, 2012	1st Working Draft	First specification and test software model that can be used for subsequent improvements.
July, 2012	Committee Draft	Essentially complete and stabilized specification.
January, 2013	Draft International Standard	Complete specification. Only minor editorial changes are allowed after DIS.
July, 2013	Final Draft International Standard	Finalized specification, submitted for approval and publication as International standard.

that among several component technologies for image retrieval, such a standard should focus primarily on defining the format of descriptors and parts of their extraction process (such as interest point detectors) needed to ensure interoperability. Such descriptors must be compact, image-format independent, and sufficient for robust image-matching. Hence, the title Compact Descriptors for Visual Search was coined as an interim name for this activity. Requirements and Evaluation Framework documents have been subsequently produced to formulate precise criteria and evaluation methodologies to be used in selection of technology for this standard. The Call for Proposals[17] was issued at the 96th MPEG meeting in Geneva, in March 2011, and responses are now expected by November 2011. Table 1 lists milestones to be reached in subsequent development of this standard.

It is envisioned that, when completed, this standard will

- ensure interoperability of visual search applications and databases,
- enable a high level of performance of implementations conformant to the standard,
- simplify design of descriptor extraction and matching for visual search applications,
- enable hardware support for descriptor extraction and matching in mobile devices, and
- reduce load on wireless networks carrying visual search-related information.

To build full visual-search applications, this standard may be used jointly with other existing standards, such as MPEG Query Format, HTTP, XML, JPEG, and JPSearch.

Conclusions and outlook

Recent years have witnessed remarkable technological progress, making mobile visual search possible today. Robust local image features achieve a high degree of invariance against scale changes, rotation, as well as changes in illumination and other photometric conditions. The BoW approach offers resiliency to partial occlusions and background clutter, and allows design of efficient indexing schemes. The use of compressed image features makes it possible to communicate query requests using only a fraction of the rate needed by JPEG, and further accelerates search by storing a cache of the visual database on the phone.

Nevertheless, many improvements are still possible and much needed. Existing image features are robust to much of the variability between query and database images, but not all. Improvements in complexity and compactness are also critically important for mobile visual-search systems. In mobile augmented-reality applications, annotations of the viewfinder content simply pop up without the user ever pressing a button. Such continuous annotations require video-rate processing on the mobile device. They may also require improvements in indexing structures, retrieval algorithms, and moving more retrieval-related operations to the phone.

Standardization of compact descriptors for visual search, such as the new initiative within MPEG, will undoubtedly provide a further boost to an already exciting area. In the near

term, the availability of standard data sets and testing conditions is expected to foster competition and collaboration to develop a best-of-breed solution. Longer term, the existence of a widely accepted standard will provide the certainty for industry to make major investments in mobile visual-search technology that will lead to a broad deployment. **MM**

References

1. J. Sivic and A. Zisserman, ''Video Google: A Text Retrieval Approach to Object Matching in Videos,'' *Proc. IEEE Int'l Conf. Computer Vision* (ICCV), IEEE Press, 2003.
2. D. Nister and H. Stewenius, ''Scalable Recognition with a Vocabulary Tree,'' *Proc. IEEE Conf. Computer Vision and Pattern Recognition* (CVPR), IEEE Press, 2006.
3. B. Girod et al., ''Mobile Visual Search,'' *IEEE Signal Processing Magazine,* vol. 28, no. 4, 2011, pp. 61-76.
4. D. Lowe, ''Distinctive Image Features from Scale-Invariant Keypoints,'' *Int'l J. Computer Vision,* vol. 60, no. 2, 2004, pp. 91-110.
5. H. Bay, T. Tuytelaars, and L. Van Gool, ''SURF: Speeded Up Robust Features,'' *Proc. European Conf. Computer Vision* (ECCV), Springer, 2006.
6. S.S. Tsai et al., ''Mobile Product Recognition,'' *Proc. ACM Multimedia,* ACM Press, 2010.
7. V. Chandrasekhar et al., ''Compressed Histogram of Gradients: A Low Bit Rate Feature Descriptor,'' *Int'l J. Computer Vision,* vol. 94, 2011, pp. 1-16.
8. H. Jegou, M. Douze, and C. Schmid, ''Hamming Embedding and Weak Geometric Consistency for Large Scale Image Search,'' *Proc. European Conf. Computer Vision* (ECCV), Springer, 2008.
9. J. Philbin et al., ''Lost in Quantization—Improving Particular Object Retrieval in Large Scale Image Databases,'' *Proc. IEEE Conf. Computer Vision and Pattern Recognition* (CVPR), IEEE Press, 2008.
10. O. Chum, J. Philbin, and A. Zisserman, ''Near Duplicate Image Detection: Min-Hash and TF-IDF Weighting,'' *Proc. British Machine Vision Conf.* (BMVC), BMVA, 2008; http://www.bmva.org/bmvc/2008/papers/119.pdf.
11. X. Zhang et al., ''Efficient Indexing for Large-Scale Visual Search,'' *Proc. IEEE Int'l Conf. Computer Vision* (ICCV), IEEE Press, 2009.
12. M.A. Fischler and R.C. Bolles, ''Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography,'' *Comm. ACM,* vol. 24, no. 6, 1981, pp. 381-395.
13. O. Chum, J. Matas, and J.V. Kittler, ''Locally Optimized Ransac,'' *Proc. German Assoc. Pattern Recognition Symp.* (DAGM), Springer, 2003.
14. S.S. Tsai et al., ''Fast Geometric Re-Ranking for Image Based Retrieval,'' *Proc. IEEE Int'l Conf. Image Processing* (ICIP), IEEE Press, 2010.
15. V. Chandrasekhar et al., ''Low Latency Image Retrieval with Progressive Transmission of CHoG Descriptors,'' *Proc. ACM Int'l Workshop Mobile Cloud Media Computing,* ACM Press, 2010.
16. *USNB Contribution: On Standardization of Mobile Visual Search,* MPEG input document M17156, MPEG Requirements Group, Jan 2010.
17. *Compact Descriptors for Visual Search: Call for Proposals, Applications, Scope and Objectives, Requirements, and Evaluation Framework,* MPEG output documents N12038, N11529, N11530, N11531, N12039, MPEG Requirements Group, July 2010-Mar. 2011; http://mpeg.chiariglione.org/working_documents.htm.

Contact author Bernd Girod at bgirod@stanford.edu.

Contact editor Anthony Vetro at avetro@merl.com.