Mobile feature-cloud panorama construction for image recognition applications

Miguel Bordallo López, Jari Hannuksela and Olli Silvén
Machine Vision Group
University of Oulu, Finland
Email: miguel.bordallo@ee.oulu.fi

Abstract—Camera-based context retrieval applications are a set of services that provide information about the user environment based on a geo-located picture. Traditionally, to retrieve the information, the server was in charge of computing the features of a taken picture and matching it with the ones present on a database. Our design proposes the substitution of still image capturing for real-time video frame analysis that is able to compute the features required for the image matching on the fly. In our approach, each single video frame shown in the viewfinder is analyzed to extract the relevant features. Only the new features extracted on this single frame are sent to the server along with the registration parameters that compose a feature-cloud panorama. To increase the robustness of the system, only high quality frames are selected for the feature mosaic composition. A moving objects detection stage can discard the undesired detected features. The experiments show that by moving most of the computation to the device, the bandwidth is reduced considerably and the chances of a successful matching improve with the large number of accurate features detectable on a video sequence. The general approach based on feature extraction, registration and object detection for video frames, is achievable at a high frame rate.

I. INTRODUCTION

Camera-based context retrieval applications are a set of services that provide information about the user environment based on a geo-located picture. On a context retrieval application, a user takes a picture from a handheld device that is sent to a server along with some information. Some content is then retrieved from a server and displayed on a mobile device. Traditionally, to retrieve the information, the server was in charge of computing the features of a taken picture and matching it with the ones present on a database. Figure 1 depicts a typical use case of a context recognition application where a tagged building is recognized and relevant information is displayed on the screen.

The main goal of an image recognition application consists on being able to recognize the objects or scenes present in the place where the image is taken. The immediateness of the content retrieval is the most desired feature. Typical proposals for image recognition systems are based on one or several high resolution images that have to be sent to the server for analysis.

The problems that have to be faced when considering the capture of suitable images for the retrieval system are founded on the camera and the environment. These issues, including eventualities such as moving objects across the scene and the motion blur due to involuntary hand shaking or autofocusing problems, cannot be ignored when trying to obtain quality images. Sometimes, a single picture is not enough to perform the recognition and matching properly and several images have to be sent to the server.

Modern mobile communication devices frequently contain built-in cameras allowing users to capture high resolution still images, but at the same time the imaging applications are facing both usability and throughput bottlenecks. The difficulties in taking ad hoc pictures of printed posters or logotypes with multi-megapixel cellular phone cameras on a common use case, illustrate these problems for anyone. The image recognition result can be examined only after several seconds and is often unsuccessful. Therefore, a new picture is needed although the view-finder image had looked good. The process can be a frustrating one with waits and the user not being able to predict the quality of the picture intended to be recognized on beforehand. The problems can be traced to the processor speed and camera resolution mismatch, and application interactivity demands.

Another traceable problem is that a typical 5Mpixels image can use up to 20Mbytes of data when uncompressed. Lossy compression techniques can easily reduce the image data needs to some hundreds kilobytes. However, lossy compression presents certain problems when applied to feature extraction. Compression artifacts might dramatically decrease the quality of feature points if the robustness of them is not taken into account.
Recently, the paradigm for camera-based context retrieval applications has been changed to try to avoid the bandwidth consumption needed for sending complete images, and to reduce the workload of the server when it needs to compute vast amount of pictures sent at the same time. On the most recent approaches [10], the feature extraction stage is moved to the device itself, reducing the workload on the server, and the amount of data to be sent. However, this still presents two main problems, the first one being that the amount of needed features for object recognition is difficult to determine on beforehand. The typical solution is the computation and transmission of an excessive number of feature points that can easily need high bandwidth. The second problem is the incapability of a single still frame of representing properly a three-dimensional world. When facing the recognition of a landscape or scene, a composition of a bigger scene with a wide angle could easily improve the success rate of the image matching process.

To increase the recognition success rate and to provide interactivity, our design proposes the substitution of still image capturing for real-time video analysis that is able to compute on the fly the features required for the image matching. In our approach, each single video frame shown in the viewfinder is analyzed to extract the relevant features. In this context we analyze building feature mosaics of printed documents or scenes from frames selected from VGA resolution (640x480 pixel) video. Our proposal relies on the feature extraction of a large amount of VGA video frames to compose a feature mosaic that will be transmitted sequentially to the server for content retrieval. Furthermore, the user is guided into cooperative interactions that rapidly reduce ambiguities and facilitate identifying the correct target.

II. IMAGE RECOGNITION APPLICATIONS

Simply pointing a phone’s camera at objects around you and finding information about them provides a very intuitive user interface for accessing information and services around your current location. Such a system provides a bridge between the digital and physical worlds.

Context retrieval applications enable people on the move to access relevant information and services on the Internet, simply by pointing their mobile phone camera at real life objects. The most complete context retrieval services combine a mobile augmented reality browser with a web-based content management tool that allows anyone to create and publish augmented reality experiences.

In a context retrieval application, when the device is pointed at an object, the system uses a variety of the phone’s sensors (including the camera and GPS positioning) to evaluate the object. Then, by searching through a database of previously tagged items, the system identifies the object and displays a set of links to associated content and services. Several examples of context retrieval applications are commercially available. Current mobile image retrieval systems on the market include Google Goggles [1], SnapTell [2], Kooaba [4] and Nokia Point and Find [3]. In these applications, several types of objects such as landmarks, 2D-barcodes or movie posters can be recognized.

Some details about the implementation of such applications can be found in the literature. Our work takes grounds in the work by Takacs’ et al. [13], that explain how pointing with a camera provides a natural way of indicating one’s interest and browsing information available at a particular location. The work of Chandrasekhar [7] shows the implementation of CHoG [7] features for its use in image recognition and also compares several feature descriptors in terms of performance and required data transmission [6]. Other interesting works include the papers from Hull [10], presenting a comparison between several mobile image recognition architectures and Tsai [14], that presents the implementation of a complete mobile image retrieval system.

III. SYSTEM DESIGN

Based on our prior work with video analysis and panorama construction, our proposal relies on the feature extraction of a large amount of VGA video frames to compose a feature mosaic that will be sequentially transmitted to the server for content retrieval. Table I depicts some of the advantages of the proposed system compared with a more traditional still image based approach.

<table>
<thead>
<tr>
<th>Field of view</th>
<th>Still image based</th>
<th>Video based</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of features calculated</td>
<td>Limited</td>
<td>Fixed</td>
</tr>
<tr>
<td>Data transmitted</td>
<td>Whole picture or excessive features</td>
<td>Small amount of features per frame</td>
</tr>
<tr>
<td>Recognition of moving objects</td>
<td>Impossible</td>
<td>Done by moving object recognition</td>
</tr>
<tr>
<td>Recognition of quality</td>
<td>Impossible</td>
<td>Done by blur detection</td>
</tr>
<tr>
<td>Feedback on image matching</td>
<td>After the image capture</td>
<td>During the image capture</td>
</tr>
<tr>
<td>Server computational load</td>
<td>High</td>
<td>Low</td>
</tr>
</tbody>
</table>

In our application, most of the computations are moved to the device and computed in real time during the image capturing stage. Figure 2 shows the proposed system design for a feature cloud panorama based context retrieval system.

In our approach, each single video frame shown in the viewfinder is analyzed to extract the relevant features and an image registration stage is performed to construct a feature-cloud panorama. Figure 3 depicts the feature capturing process. Only the new features extracted on a single frame are sent to the server along with the registration parameters that compose a feature cloud.

In addition to interactivity benefits, the use of low resolution video imaging can be defended from purely technical aspects. In low resolution mode the sensitivity of the camera can be better as the effective size of the pixels is larger, reducing
the illumination requirements and improving the tolerance against motion blur. On the other hand, a single-shot high resolution image, if captured properly, could be analyzed and used directly after acquisition, while the low resolution video capture approach requires advanced algorithms and significant processing effort.

IV. APPLICATION IMPLEMENTATION

In the image capturing procedure, the goal is to use the camera mostly in a natural panning manner, enabling to show augmenting information for the user on the screen. In order to retrieve the context of a determined scene, the user focuses the camera on the desired object or target. To demonstrate the generality of the implementation, the user starts panning the camera left and right while capturing a sequence of images. Each image is individually processed to estimate motion. The blurriness of each picture is measured and eventual moving objects are detected. Based on the quality of each individual frame, a selection process takes place. The idea of selection is to consider only good quality frames for the extraction of the recognition features. In this process, each frame is either accepted or discarded. When the server has a sufficient amount of features to recognize the target, a message is shown on the screen, and the user can display the context information. The application flow is showed in Figure 4.

A. Automatic start of image recognition application

Image recognition and all its communications needs have significant latencies that reduce the users impression of interactivity. From the users point of view, the context retrieval process would be most convenient if the device automatically recognizes the information retrieval expectations without demanding manual activation of any application [9].

This situation is subtly different from the above transition to picture capture mode. These uses could be differentiated based on the push of the camera button, but this takes place only at the end of picture capture. The video-based image recognition can be activated whenever the back camera is active, but should not interfere picture capture. Any information retrieved should be presented only on request, or if there is no doubt on the intentions of the user. In our solution, we have counted on the user co-operation and characteristic motion patterns.

The idea is demonstrated in Figure 5. The back and forth panning motion is a request for image recognition service to identify the scene and present information retrieved. If the target is a bar code, its presence is a direct request to fetch product data based on EAN/UPC code.

In this process, the information from three sensors can be employed. Both the frontal and back cameras as well as accelerometers in the device provide information on motion, improving the reliability of context recognition.
B. Interactive capture interface

The key usability challenge of image recognition based context retrieval application is enabling and exploiting interactivity. The user will collaborate on the capture of the images if enough feedback is offered. The capturing process should show the user clear instructions on how to proceed with the capture, based on the analysis of each one of the frames. The user can be guided to avoid the presence of objects that could reduce the performance of the recognition process. The user can also be assessed about the quality of the frames that are being captured based on the estimation of the motion and the amount of blur.

The user guidance employs a technique in which the camera motion is estimated from sequences of image frames with the extraction of CHoG [7] features and using the image registration methodology described in the work of Hannuksela [8]. Based on shutter time and illumination dependent motion blur the user can be informed to move or slow down to capture the required features needed for the recognition of the target. The user can also be asked to back-up or he can return to lower quality regions later in the scanning process. As a result, only features from good partial images are sent to the server.

C. Quality assessment and moving objects

Along with the interactivity, other advantages of the solution can be obtained. To increase the robustness of the system, only high quality frames are selected for the feature cloud composition. The amount of motion blur in the frame is acquired by summed derivatives. This method estimates the images sharpness by summing together the derivates of each row and each column. The result is one single number that expresses the amount of high-frequency detail in the image and is only valid when it is used to compare images [11].

To cope with undesirable artifacts crossing the scene, a moving objects detection stage can discard the undesired detected features. Motion detection is done in a very simple fashion to make the process fast. First, the difference between the current frame and the previous frame is computed. The result is a two-dimensional matrix that covers the overlapping area of the two frames. Then, this matrix is low-pass filtered to remove noise and is thresholded against a fixed value to produce a binary motion map [5] [12].

If the binary image contains a sufficient amount of pixels that are classified as motion, the dimensions of the assumed moving object are determined statistically. The centerpoint of the object is approximated by computing the average coordinates of all moving pixels and the standard deviation of coordinates is used to approximate the dimensions of the object. Figure 6 depicts a reconstructed image with and without a moving objects detection system. When a moving object is detected, the feature points of its corresponding region are ignored and not transmitted to the server.

D. Performance analysis

Together with the qualitative performance, the estimation of the bandwidths for transmission and the computation time estimation is offered on table II. Although a large number of frames have to be analyzed on a video based solution, the transmission of the features is done during the capture and feedback can be obtained from the server at the same time. The bandwidth needed for the system is very small due to the small amount of features transmitted per frame.

<table>
<thead>
<tr>
<th></th>
<th>Compressed Image sending</th>
<th>Still-image Features</th>
<th>Video-based Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total data sent</td>
<td>700 KB</td>
<td>56kB</td>
<td>0.05 KB per frame</td>
</tr>
<tr>
<td>Computation times</td>
<td>100ms (jpeg)</td>
<td>3000 ms</td>
<td>200 ms per frame</td>
</tr>
<tr>
<td>Bandwidth required</td>
<td>All possible</td>
<td>Mid</td>
<td>Very low</td>
</tr>
<tr>
<td>Images needed</td>
<td>1-2</td>
<td>1-5</td>
<td>10-200</td>
</tr>
<tr>
<td>Communication type</td>
<td>Send-receive</td>
<td>Send-receive</td>
<td>Bidirectional</td>
</tr>
</tbody>
</table>

V. CONCLUSIONS

The feature cloud panorama construction improves the chances of a successful matching due to the large number of accurate features detectable on a video sequence. In addition, the image reconstruction process based on video analysis approach offers excellent opportunities to increase interactivity by providing feedback to the user. Our tests show that feature extraction, registration and object detection for VGA frames, is achievable at a high frame rate. Moreover, our experiments show that by moving most of the computation to the mobile device, the bandwidth required is reduced considerably.
REFERENCES


