Displayed Object Recognition for Smartphone Interaction

Karim Kadar
Keio University
Yokohama, Japan
karim@hvrl.ics.keio.ac.jp

François de Sorbier
Keio University
Yokohama, Japan
fdesorbi@hvrl.ics.keio.ac.jp

Hideo Saito
Keio University
Yokohama, Japan
saito@hvrl.ics.keio.ac.jp

Abstract

In the last few years, communication between man and virtual world has been made easier with the apparition of smartphones. This paper presents a method allowing a smartphone user to recognize from any viewpoint a 3D model displayed on a screen. The model is selected from a 190 objects database and displayed rotating along the vertical axis. This method can be used to allow interaction with the recognized model on the smartphone or to retrieve information about the object. Curvature scale space based contour recognition and color matching are used to identify the captured object. Evaluation experiments show a recognition success rate of 92% on a one hundred photographs data set.

1 Introduction

Smartphones have attracted much attention from the public with an always easier access to information. Used as means of greater interaction with virtual information, they often require a capture of a particular design (bar code, AR marker, etc.) or knowledge of the displayed information to gather the required recognition data. We present a method allowing to recognize a 3D object displayed on a screen based on its contour and simple color information captured from the smartphone.

Several methods can allow recognition of an object from different points of view. Feature points can be detected in the texture of the image in a reliable way as seen with Mikolajczyk et al. [1], Lowe [2], or Bay et al. [3] works, where objects could be represented by improved bag of features [4]. In texture based methods however, features would be impacted in case of noise (luminance, reflections on the display, etc.) and objects would present changing features during their rotation as the virtual light source is fixed in space. Other methods based on Zernike moments or Histogram of Oriented Gradients such as the work of Ansary et al. [5] or Aono and Iwabuchi [6] are used in 3D search engines to find models based on a 2D picture. These methods unfortunately present a low recognition rate when only one result is returned. They draw their strengths from retrieving similar objects from the database, lowering the accuracy to recognize one particular object.

To avoid these issues we chose a contour based recognition method, more specifically a method derived from an earlier work by Lee and Drew [7] based on a modified curvature scale space [8]. This recognition process allows building eigenspaces from related contours and thus easily linking shapes from different viewpoints together to represent an entire object.

This method allows interacting with a display system without establishing any connection with it, but only with a simple and efficient use of the camera attached to a smartphone. Usually, such an interaction is achieved through special designs such as bar codes or QR codes. Being based on an object recognition method, we do not need to use such markers. We can instead directly use the displayed object for interaction purposes. The method could be used to recognize and obtain information on objects advertised on television just by photographing them. It could also be applied to digitalized art pieces displayed in museums to deepen interaction between visitors and art.

The rest of the paper is organized as follows. In Section 2 we present the environment of the system. Section 3 explains the methods used to process the input image. Section 4 details the database creation with eigenspaces and color information. Section 5 presents the complete recognition process. We detail the application, show the experimental results and discuss them in Section 6 and finally draw the conclusion of our work and future developments in Section 7.

2 System Environment

The system is composed of a server, a smartphone, and an unknown number of screens displaying 3D objects. Both server and smartphone are communicating through a wireless LAN network. The server holds the database of all 3D models but does not have any information about the actual display arrangement, as opposed to similar applications such as the Touch Projector [9]. A user takes a picture of any of the displays with minor constraints: screen is in focus and in the center of the photograph. The picture is then sent to the server and used as the input for recognition. The result of the processing is sent back to the smartphone for further use, in our case displaying the 3D model of the recognized object as shown in Fig. 1.

Figure 1. Experimental setting

3 Input Image Processing

The input photograph sent on the server by the smartphone must be processed before attempting recognition. First, the image displayed on the screen is recovered from the photograph, then both the outer contour of the object and its colors are extracted. The shape information is transcribed in a curvature scale space image which is vectorized and undergoes a phase correlation process before being used for object recognition.

3.1 Perspective correction

The first step in processing the smartphone's input photograph is to detect and extract the screen aimed at by the user in order to correct the perspective. The high contrast between the displayed image and the borders of the screen allows an easy detection of at least the shape of the main screen. All detected contours are then approximated to simpler polygons using the Douglas-Peucker algorithm. The screen is defined as the biggest quadrangle centered in the image whose angles are relatively close to 90 degrees and that is not the border of the image. The optimal capture position currently is under 30 degrees away from the normal of the screen in any direction. The final quadrangle undergoes a perspective correction through simple homography to ensure that the 3D object is not deformed, followed by a white balance of colors to allow future color matching.

Extracting the screen also allows the user to take photographs from further away, thus reducing the noise due to visible pixels and refresh rate.

3.2 Contour and color extraction

The recognition algorithm uses a single contour curve to describe an object from a certain viewpoint. The background from the recovered displayed image is not homogeneous but presents quite low gradients, allowing us to highlight the object's main contours using a Sobel Derivatives algorithm. The external contour of the biggest object is retrieved from the highlights and then slightly dilated to smooth sharp angles. All contour lengths are homogenized to a set number of points. An example of extracted contour from an input photograph is presented in Fig. 2.

The pixels found inside the contour are used to create a color matrix representing the frequency of colors present in the photographed object. This result M can be assimilated to a 2D color histogram along the Hue and Saturation color scales. Using M, the main color of the input object is determined amongst seven possible (red, yellow, green, cyan, blue, magenta, grey scale colors).

3.3 Curvature scale space image

In order to use the contour effectively we first generate a curvature scale space (CSS) image [8] from the previous result. The shape obtained is described with a closed parametrized curve $L(t)$, which is smoothed successively by convolution with a Gaussian of standard deviation $\sigma \in [5, 40]$ with a 0.25 step. The curvature of the smoothed curve is calculated as $K_\sigma(t)$ for each σ value. The CSS image can be explained as a binary graph with t as the x dimension and σ as the y dimension. When a curvature zero-crossing point is found (i.e. when $K_\sigma(t) = 0$ and $\dot{K}_\sigma(t) \neq 0$) the corresponding point (t, σ) is drawn in the CSS image. The final result is an image describing the main curves of the contour.

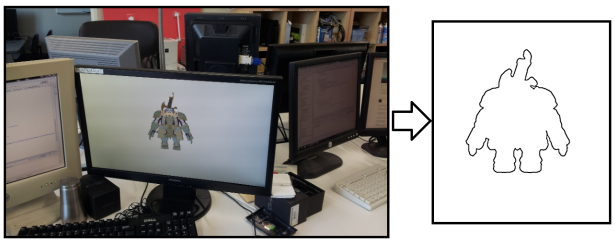

Figure 2. Perspective correction & contour extraction

As the contours extracted from photographs can be subject to noise, the small bumps on the shape are not taken into account in the final image by setting the first convolution step at $\sigma = 5$ instead of 0.25.

3.4 Transformation of scale space image

The CSS image as it is is unfortunately rotation and mirroring dependent. Combined with the fact that a raw CSS image is unnecessarily heavy, we need to transform it into a more adapted format [7].

3.4.1 Marginal-sum feature vectors:

First, the CSS image 2D information is vectorized by summing the binary image's pixel values along both rows and columns. Let r be the vector of summed rows (of size n: the length of the contour) and c the vector of summed columns (of size m: the number of smoothing levels). This operation allows to reduce the data size from a $n \times m$ matrix to a $n + m$ vector. The contour data extracted from the input image is now represented by $x = [r, c]^T$.

3.4.2 Phase correlation:

The current vector x is still rotation and mirroring dependent. Such transformations of the input image would cause a translation of the contour's starting point along the curve and with it a circular shift of the CSS image along the t axis. In the new representation, only r would be impacted.

The solution adopted by Lee and Drew [7] to solve this issue is to perform a phase correlation on r. This can be achieved mathematically with

$$\tilde{r} = |F^{-1}(|F(r)|)|$$

where F represents a 1D Discrete Fourier Transform.

The final representation of the input image used for the recognition process is $x = [\tilde{r}, c]^T$.

4 Eigen-CSS and database processing

The used database consists of 190 3D models, most from the object databank [10] and each rendered on a

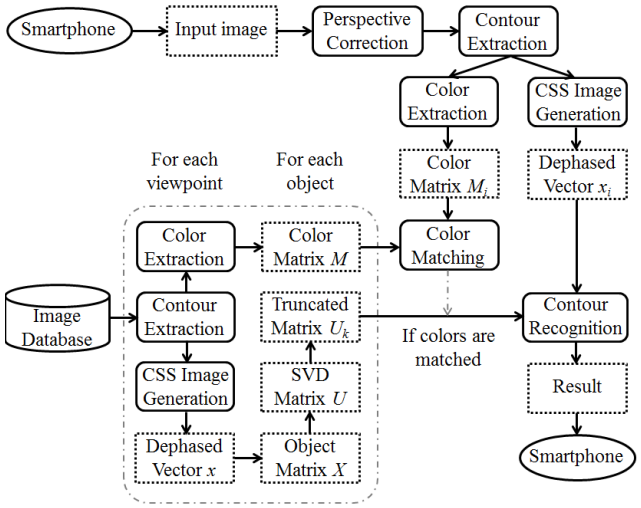

Figure 3. Object recognition process

white background under 12 different viewpoints - one every 30 degrees of a vertical axis rotation. Each set of images is processed into a specific eigenspace associated to the corresponding object.

A vector $x_i = [\tilde{r}, c]^T$ is created for each viewpoint i of an object using the previously detailed methods (from 3.2 to 3.4). All viewpoints are then assembled in a matrix $X = [x_1, x_2, .., x_{12}]$ gathering all the processed data about the contour of the object.

A Singular Value Decomposition is performed on the input matrix X producing USV from which only the matrix of eigenfeatures U will be used to represent the object. Given its important size $(m+n) \times (m+n)$, U is truncated by keeping only its first k columns. The result U_k is a fair approximation of U with modest $(m+n) \times k$ dimensions.

In addition to U_k, the objects are also assigned several characteristics throughout their processing:

- The object is declared "circular" if at least one of its viewpoints presents a curve without any curvature zero-crossing point.

- The object is assigned a list of main colors appearing on different viewpoints, seven colors are represented along the Hue/Saturation scale: red, yellow, green, cyan, blue, magenta, and grey.

- The object is assigned a 2D color histogram representing the frequency of HSV colors appearing in the contour from all viewpoints.

5 Object Recognition

The image recognition process, as described in Fig. 3, can be divided into two parts: color matching and contour recognition.

The color matching consists of finding in the database all objects that shares most of its colors with the input image. The color matrix M_i of the photograph is compared to the color matrix M of each of the objects: only if all but ϵ colors from M_i are found in M will the object go through contour matching.

To recognize the contour of the object displayed on the photographed screen, we look for the database entry that gives the best reconstruction of the input vector x through its approximated eigenspace. We thus look for the object that minimizes the Euclidean distance D where $D^2 = \|x - U_k U_k^T x\|^2$.

Once the most fitting object for both colors and contour is found, the result is sent to the smartphone which in turn displays the 3D model associated with the picture taken.

6 Evaluation

6.1 Experiments

The application to be experimented on in this section works as follows. A random object taken from the database is displayed from a random viewpoint. The user takes a picture of the screen that is sent to the server. The image is then processed as explained in section 5. The result is returned to the smartphone and the associated 3D model is displayed allowing the user to interact with it directly through the touch screen. At the moment the models are pre-loaded on the smartphone but could easily be sent by the server. If any information on the recognized object is available, it can be displayed as well.

Before testing the application, we determined during a self evaluation the best parameters to use for the contour recognition process. The experiments showed best results for an incrementation step σ of 0.25 and a truncation level k of 13.

6.2 Results

We evaluated the success rate of the system with several datasets: the perfect database images, rotated ones, and more importantly a set of 100 smartphone photographs of screens displaying randomly selected objects from the database. On the smartphone input, the automatic white balance and exposure settings have a strong impact on the colors in the photograph making precise matching unsuccessful as some information is purely lost. In that regard, a simpler matching is used for now where the main color of the input is determined between seven (red, yellow, green, cyan, blue, magenta, and grey scale colors) and the contour recognition process only takes into account similarly colored objects. Some of the objects present an important loss of color information: dark brown becoming blue or black, grey becoming cyan or blue, etc. Thus objects with more than 30% of dark or grey colors are not undergoing color matching. The table below shows the actual results.

Table 1: Experiment results for several datasets and color support settings.

Dataset	#Images	Color	Recognition
Database images	2260	None	96.75%
Database images	2260	Full	99.43%
Rotated images	2260	None	84.47%
Rotated images	2260	Full	95.44%
Photographs	100	None	88%
Photographs	100	Simple	92%

Figure 4. Difference in illumination highly impacting SURF but not our method.

We then compared our algorithm to the local feature recognition method SURF. On the same photograph dataset, SURF also recognized 92% of the images in comparable times. However, reflection on the display or different illumination of the 3D model highly impacts SURF's performance while it does not change our results as long as the contour is still visible and the color of the light source does not change. An example of such a case is presented in Fig. 4.

All experiments where carried out on a 2.81GHz Quad core machine. The pre-processing of an image (sections 3.2 to 3.4) takes an average of 76 ms. The recognition process over the entire database with a non optimized code takes in average 363 ms without the color matching, and 34 ms with the full color matching. This system processing time is not taking into account the client/server networking capability to transfer pictures.

6.3 Discussion

In this subsection, we will discuss some of the characteristics of this methods which must be taken into account when pondering whether or not to use it.

On the one hand, depending on the smartphone used (here a Samsung Galaxy SII), the built-in parameters and the quality of the camera system can impact on the color recognition process. In our case, the smartphone captured different colors than the ones displayed resulting in colors being modified from the ground truth: dark greens becoming black, pink becoming red, grey becoming cyan or blue etc. We are currently working on solving this issue by registering only lightly affected and thus recognizable colors and by classifying input and objects by main apparent color.

Also, some objects are fully convex and have no curvature zero-crossing points along certain viewpoints. That would be the case for a wheel or a mural clock for example. These objects are unrecognizable using only the contours. We don't recommend using this method for a database highly populated by such objects.

On the other hand, this method is highly adaptable to any other kind of 3D object database. For more fidelity of recognition from unregistered viewpoints, the user can increase the number of views in the database. This decision should preferably be taken when creating the database images for the first time.

7 Conclusion and Future Works

The approach presented in this paper presents large benefits for a robust recognition of displayed object captured by a smartphone. It unlocks new ways to interact with a display system without having information on it or using special markers. The correction and extraction of the contour allows to negate the impact of texture blurring and noisy features while conserving enough information to recognize the object in most cases. Moreover the combination of Eigen CSS and color matching improves and quickens the process.

The tests performed on the random sample from the 190 objects large database give a success rate of 92% from the smartphone photographs. On the used dataset, the color matching allowed for two more images to be recognized and for faster processing time. Thus the applications detailed in the introduction are compatible with this approach.

For future works based on this approach, it is recommended to calibrate the degraded input colors in order to be able to tighten the color matching requirements. This should allow for more accurate object categorization and even faster processing times. A texture based recognition method could also supplement this approach only for the processing of non characteristically curved objects such as ellipses or circles. A multi resolution database could also be taken into consideration to allow pictures to be taken from even further.

Acknowledgements

This work was partially supported by MEXT/JSPS Grant-in-Aid for Scientific Research(S) 24220004.

References

[1] Mikolajczyk, K., & Schmid, C.: "Scale & affine invariant interest point detectors," *International journal of computer vision*, vol.60, no.1, pp.63–86, 2004.

[2] Lowe, D.: "Object recognition from local scale-invariant features," *Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol.2, pp.1150–1157, 1999.

[3] Bay, H., Tuytelaars, T., & Gool, L.: "SURF : Speeded Up Robust Features," *Computer VisionECCV*, pp.404–417, 2006.

[4] Jégou, H., Douze, M., & Schmid, C.: "Improving Bag-of-Features for Large Scale Image Search," *International Journal of Computer Vision*, vol.87, no.3, pp.316-336, 2009.

[5] Ansary, T. F., Vandeborre, J.-P., & Daoudi, M.: "3D-Model search engine from photos," *Proceedings of the 6th ACM international conference on Image and video retrieval*, 2007.

[6] Aono, M., Iwabuchi, H.: "3D Shape Retrieval from a 2D Image as Query," *Proceedings of APSIPA Annual Summit & Conference*, 2012.

[7] Drew, M. S., Lee, T. K., & Rova, A.: "Shape retrieval with eigen-CSS search," *Image and Vision Computing*, vol.27, no.6, pp.748–755, 2009.

[8] Abbasi, S., Mokhtarian, F., & Kittler, J.: "Curvature scale space image in shape similarity retrieval," *Multimedia Systems*, vol.7, no.6, pp.467–476, 1999.

[9] Boring, S., Baur, D., Butz, A., Gustafson, S., & Baudisch, P.: "Touch projector: mobile interaction through video," *Proceedings of the 28th international conference on Human factors in computing systems*, pp. 2287–2296, 2010.

[10] Tarr, M. J.: The Object Databank. Available from http://stims.cnbc.cmu.edu/Image%20Databases/