Painting Recognition Using Eigenimages

Eva Enns, Mudit Garg, Shahar Yuval
Department of Electrical Engineering, Stanford University

***Abstract*— We propose a method for identifying a painting in an image using eigenimages. We first pre-process the image to remove the background and then correct for skew and scaling. We then apply the method of eigenimages on de-meaned images. The algorithm correctly matched 97% of the images in a test database of 110 images, with an execution time of 3.6 s per image.**

I. Introduction

The increasing prevalence of camera phones and better data connections make it feasible to create phone-based applications to enrich a user's experience in different environments. For example, museum brochures could be replaced by interactive, phone-based guides, which provide information in context with the works being viewed. Such an application would require the identification of an object in an image captured by the user.

There are two broad categories of approaches to this problem of object identification. The first consists of techniques that identify key points in the image and use features around these points for identification [1], [2]. The second is the method of eigenimages, which is popular in distinguishing between objects of a similar type, as in face recognition. This method projects the image from a high dimensional space onto a lower dimensional space. The lower dimensional vector compacts the unique features of the image, and can be more easily compared to a database.

The objective of this paper is to present a technique for identifying a painting captured in an image. For the given setting, we felt that recognition accuracy and speed of execution were the most important considerations. We therefore chose the method of eigenimages because of its low computational requirements and its suitability to the given problem.

This paper is structured as follows. In Section II the required pre-processing steps are described. In Section III the eigenimage method used is explained. In Section IV some performance results are presented, followed by a discussion in Section V. Finally, conclusions regarding this technique are given in Section VI.

II. Pre-Processing

Before the eigenimages technique can be employed, the painting must be scaled and aligned. To accomplish this, our pre-processing algorithm first downsamples the image and removes the background content of the image by creating a mask for the painting. It then identifies the corners of the painting and applies a perspective transform to map the painting into a fixed square at the center of the field of view. This yields paintings of equal dimension and position, which helps eigenimages discriminate between images correctly.

A. Downsampling

Since eigenimages are susceptible to misalignments of the original images, image resolution has to be reduced to the order of our alignment accuracy. This ensures that misalignments are not over represented in the image. Initially we chose a downsampling factor of 3, but found that a downsampling factor of 5 resulted in better performance and reduced run time.

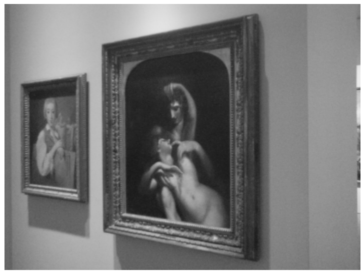
B. Mask Generation

All images in the gallery have either golden or blue frames, and are displayed against a white, well illuminated wall; therefore, there is a significant change in the luminance component of the image at the interface between the wall and the frame. This allows us to extract a mask of the painting region.

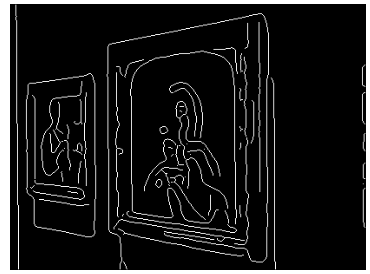
We used a Canny edge detector on the luminance component of the image. We dilated the extracted edges with a square structuring element to fill in regions with many pronounced edges, such as the painting region. Following small region removal, regions were labeled, and the region containing the central pixel was selected as the mask. The generated mask closely followed the frame edge, however, in some cases there were small peninsulas that extended into the wall region. To remove these we eroded twice with the following structuring element:

$$SE = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} \quad (1)$$

This structuring element is particularly suited for removing narrow "bridges" between regions. Thus, any adjacent and unrelated small regions were separated from the painting. By applying small region removal, we were



(a) Original Image



(b) Canny Edge Detector Output

(c) Masked Image

(d) Processed Mask

Fig. 1. (a) Downsampled input image. (b) Edge image from canny edge detector. (c) Initial mask from dilated edge image, small regions removed. (d) Final mask, after 2 erosions with the structuring element described in Eqn 1.

able to eliminate the peninsulas. To regain the proper size of the mask, we dilated twice by a square structuring element.

C. *Corner Identification*

In order to determine the transform required to map the painting into a fixed square, identification of the corners of the painting was required. We achieved this by first considering the bounding box containing the mask. Since we assumed that the paintings were approximately rectangular (with some distortion due to skew), we divided the bounding box into four quadrants. In each of the four quadrants, we computed the point within the mask furthest from the center of the bounding box (Figure 2). These resulting four points were defined to be the corners of the painting.

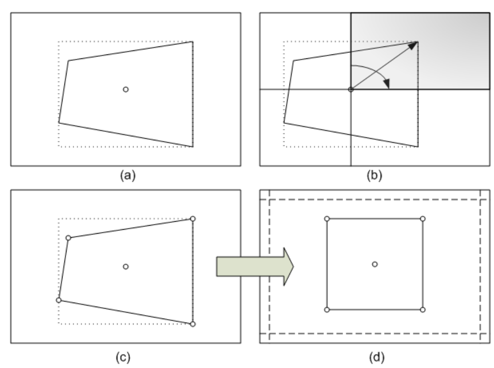


Fig. 2. (a) Bounding box around mask is computed. (b) The furthest point from the bounding box center in each quadrant is identified as a corner. (c) & (d) The four corners are mapped to a square using a perspective transform.

D. *Perspective Transformation*

Using the identified corners, we computed a perspective transform to map those corners to a centered square region. By applying this transform to the image, we corrected for skew, as well as any small rotations or translations of the painting. Through this transformation, we lost the aspect ratio information of the painting; however, the performance of the eigenimages technique was greatly improved with consistent alignment, despite this loss of information. Additionally, we found it is difficult to accurately estimate the aspect ratio of moderately skewed painting. We therefore found that the aspect ratio was not a reliable feature for painting identification when considering skewed paintings.

III. Matching Algorithm

After adjusting for scale and skew in the preprocessing step, we can now use the eigenimage method

(a) Detected Corners

(b) Transformed Image

Fig. 3. (a) Corners detected using the four quadrant method on masked image. (b) Final output image from pre-processing stage.

for painting identification.

To make the images more independent of lighting, we subtracted the average pixel value of each image from itself. The image vectors f_i were then used to create a matrix S as shown in Eqn 2.

$$S = \left(f_1 - \frac{1}{N} \sum f_1, \cdots, f_m - \frac{1}{N} \sum f_m \right) \quad (2)$$

The matrix S is used to compute the eigenimages using the Sirovich and Kirby method.

$$S^H S = \lambda_i v_i \quad \forall i = 1 \cdots M \quad (3)$$

The eigenimages e_i are found using Eqn 4:

$$e_i = S v_i \quad (4)$$

This process is carried out for the luminance and chroma-red channels of the original image. The chroma-blue component did not improve performance for this technique, so it was discarded to save initialization and computation time.

M eigenvectors generated using Eqn 4, and all are stored in two separate databases for the luminance and chroma-red channels.

We then computed the projection of each of the training images on the eigenimage databases and stored them in a projection database. The projection of the test image onto the eigenimages was calculated and then compared with the projection database. According to [3], the Euclidean distance measure gives the best matching rate for the eigenimages method. Therefore, we decided to use the Euclidean distance between the projections of the test image and of the training images. We define the set of all training images corresponding to the same painting as T_k. We measure the distance from the images in T_k to the test image, and average them to get a match score, $\bar{d}_k$. The minimum of these scores is the match.

$$\bar{d}_k = \frac{1}{N} \cdot \sum_{i \in T_k} |\vec{p}_{test} - \vec{p}_i| \qquad N = size(T_k) \quad (5)$$

IV. Results

Using a training set with 3 redundant images of each painting, eigenimages matched almost all the images. Out of a testing set of 110 images which were taken using a Nokia N93 phone, there were only 3 mismatches. This demonstrated the effectiveness of eigenimages when the training and test images were aligned properly.

Following an initial 22 s startup time, images were processes within 3.6 s, on average, which is fairly reasonable for an interactive user experience.

Using the top two closest matching training set images, we defined the following measure to determine the confidence in the match:

$$CONF = \frac{d_2}{d_1} \quad (6)$$

Where d_1 and d_2 were the scores for the top two matches.

We observed that all failures in the testing set used, the confidence measure lower than 1.05.

V. Discussion and Alternatives

Since the performance of eigenimages is highly dependent on successful deskewing, the few failures during the test runs were due to images that were not properly masked and deskewed (Figure 4). This highlights the importance of proper pre-processing before applying eigenimages.

Eigenimages provided good matches whenever the confidence measure was large. Images for which the confidence measure was small could be further processed using an additional method.

Fig. 4. The mask did not remove the mantle place below the image, causing deskewing to fail

Since the images were rich in color information we believed comparing histogram profiles could help discriminate between images for which the eigenimage confidence measure was too small. We used the histogram intersection method (HI), as suggested by [4]. However, the histogram information did not improve our results, and we did not include this method in our final algorithm.

A shortcoming of eigenimages is the dependence on a fairly large database. For this project, even after downsampling by a factor of 5, the eigenimage database was 70 MB large per color component, for a training of set 33 paintings, with 3 redundant images for each. One solution would be to discard some of the less significant eigenimages, as suggested by [3].

VI. Conclusion

This paper presents an accurate and fast method for identifying paintings in an image. We have demonstrated the usefulness of the eigenimage method in for this problem. We have also given a framework for extending this technique to combine eigenimages with other methods.

VII. Division of Labor

All members contributed equally to this project.

References

[1] D. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[2] ——, "Object recognition from local scale-invariant features," *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, vol. 2, 1999.

[3] P. Navarrete and J. Ruiz-del Solar, "Eigenspace-based recognition of faces: Comparisons and a new approach," *Proc. of the Int. Conf. on Image Analysis and Processing ICIAP 2001*, pp. 42–47.

[4] W. Jia, H. Zhang, X. He, and Q. Wu, "A comparison on histogram based image matching methods," *Proceedings of the IEEE International Conference on Video and Signal Based Surveillance*, 2006.