A COMPUTATIONAL FRESCO SKETCH GENERATION FRAMEWORK

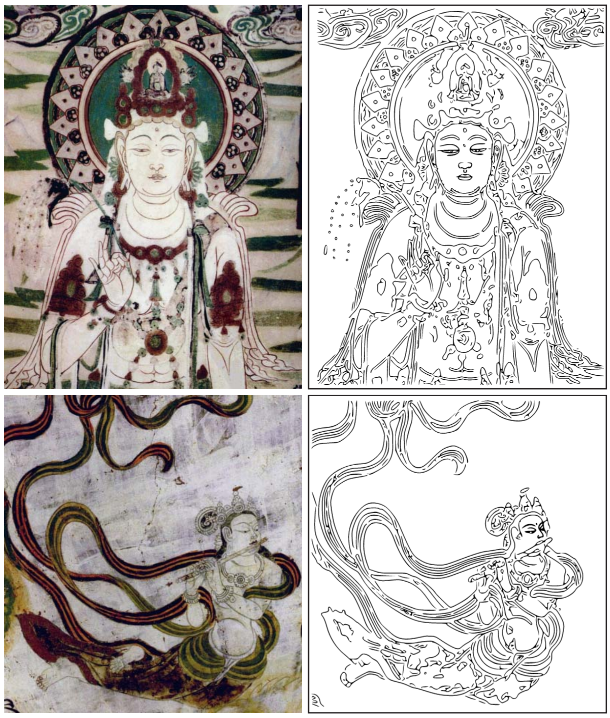
*Jianing He, Shan Wang, Yi Zhang, Jiawan Zhang**

Graphics and Visual Computing Laboratory
Tianjin University, Tianjin, China
jianinghe111@gmail.com, shiny.wangshan@gmail.com, yizhang@tju.edu.cn, jwzhang@tju.edu.cn

ABSTRACT

As a famous cultural wealth, the sketch of fresco is one of the most important art expression forms in the World Heritage. To avoid the damage of natural and human factors, painters can only use photos and videos to depict sketch in most world culture heritage sites, otherwise real frescos. Therefor, a computational method for extracting sketch is helpful and meaningful in sketch copying and researching area. However, existing approaches with the incomplete fresco are not enough to deal with the challenge of sketch extraction. In this paper, we proposes a framework to generate sketch of fresco with frescos as input. To reduce noise and refine detail lines, we adopt hierarchical segmentation technology to extract sketches of different regions respectively, and splice those sketches into an integrated sketch. For replacing the missing content and getting a complete sketch, we provide recommendations from a database of many existing fresco sketches elements for users to select. At last, users can adjust sketch based on vectorization to achieve optimization. Our framework can be used for artists or learners to study painting sketch images and research fresco art. The experimental results demonstrate the effectiveness of our framework.

Index Terms— Fresco; Sketch generation; Structure of fresco; Missing part of fresco



(a) (b)

Fig. 1. The Dunhuang Mogao Grottoes Fresco and the sketch generated by our framework.

1. INTRODUCTION

Fresco (Figure 1(a)), with smooth lines, rich color and unique content, is a famous cultural treasure in the World Heritage. Early sketches of frescos were drawn by artists with reference to real frescos. Afterwards, to avoid the effect of light and human destruction, most famous fresco heritage sites have forbidden people to enter into. Currently, real frescos cannot be seen, and just the photos or videos are provided for painters to copy. Therefore, extracting the sketch (Figure 1(b)) by computational method is urgently needed and meaningful for artists and learners to depict sketch of fresco.

However, with only frescos as input, sketch extraction from fresco is still challenging. At first, due to the damage of the natural and human factors, frescos are missing and uncomplete. Then, even with complete frescos, it is not necessarily able to get the full sketch through existing technologies. ShadowDraw system [1] with a large amount of images data to train, only helps users draw sketch in existing categories, but it is difficult to gain sketch with only a fresco data. Though image segmentation method can extract single fresco, the importance of the extracted lines and contents need to be considered.

*Jiawan Zhang is the corresponding auther, and this work is supported by the National Science and Technology Support Project (2013BAK01B01, 2013BAK01B05).

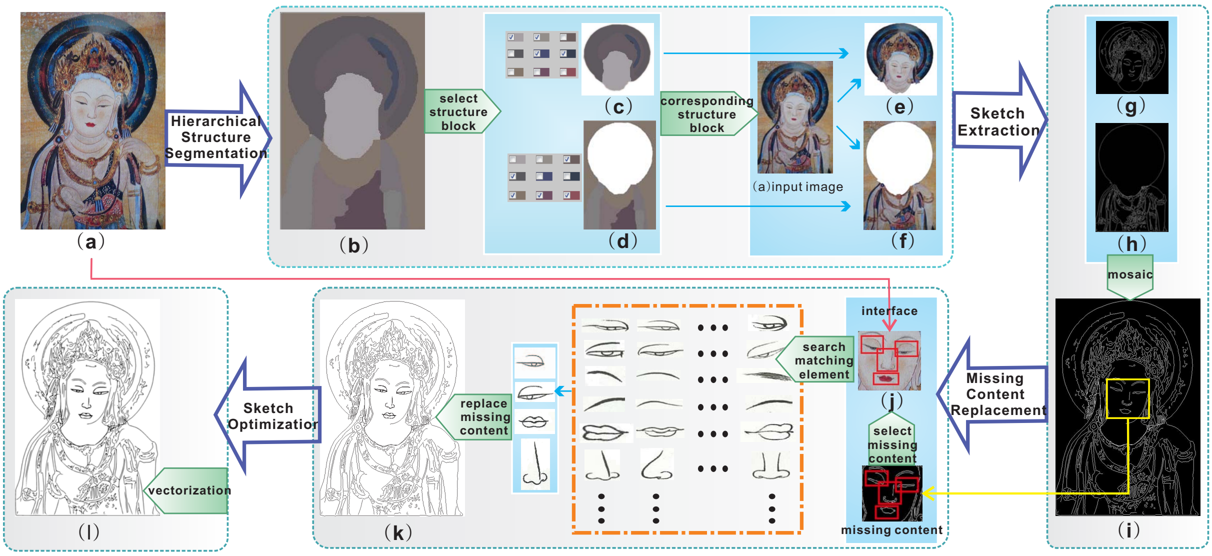


Fig. 2. The flowchart of sketch generation. Input image (a) is segmented to hierarchical structure (b). The users select the color to decide structure block sets (c) and (d), then input image (a) is also divided into (e) and (f) corresponding (c) and (d). After splicing the extraction sketches (g) and (h) into a mosaic sketch (i), users select the missing contents (j) in the interface of (a) and replace them with recommendations. The final image (l) is optimized from (k) using vectorization.

In this paper, we propose a sketch generation framework with a given fresco. We use the term 'sketch' to mean a art form that expresses the fresco content using lines. To minimize global noises and refine local detail lines, we segment fresco into different closed regions, and sketches of different regions are extracted respectively, and we introduce the concept of 'structure block' to represent every segmented region. In order to obtain a complete sketch, we replace missing regions and optimize generation sketch. Our experimental results are worthy for copying and researching. Moreover, they can also be showed in exhibition center or visitor center.

Overall, the contributions of this paper are:

- We propose a tracing and exhibition tool for artists or learners to study painting sketch images and research sketch art and fresco art.
- We design a framework to generate sketch with interaction of users using reliable methods: we present a strategy to get detail lines by extracting hierarchical structure blocks and replacing missing contents, we build a database that includes some elements of sketches drawn by artists, and we provide a vectorization approach to adjust details of sketch.

The rest of this paper is organized as follows. In Section 2, we give an overview of related work. In Section 3, we present the framework with relevant algorithms. The results and evaluation is shown in Section 4. Finally, Section 5 concludes our work.

2. RELATED WORK

Sketch is a rendering technique available to humans, e.g., an interactive sketch recognition system [2] has been proposed to understand how humans depict the visual world using sketching. Sketch is also an art form of painting, e.g., an example-based stroke stylization approach [3] has been introduced for digital painters to generate expressive handwriting. Sketch image can also be extracted in frequency domain, a frame stylized curve model [4] with users'interaction can generate smoother and more multifarious sketch. Yong Jae Lee proposed ShadowDraw system [1] to dynamically adapt to the users drawing and provide real-time feedback of shadow images from thousands of images. ShadowDraw produced more realistically proportioned line drawing than others, but only draw images classified by its database.

Image segmentation techniques [5] in 2004, combining pixel texture information with color information, treated the specified pixel as pixel seed to grow the region in accordance with the principle of level set diffusion. This method achieved better results in Japanese comics. Hierarchical image segmentation [6] introduced the gPb and gPb-owt-ucm algorithms, the contour detector gPb provides equal or better precision for most choice of recall, compared favorably with other leading techniques, and gPb-owt-ucm provides universally better performance than alternative segmentation algorithms. Edge detection algorithms Marr-Hildreth [7], Sobel [8] and Canny [9] have their own advantages in contour extraction process [10], while improved Canny method [11] is more conducive to detect boundary of color image. For fea-

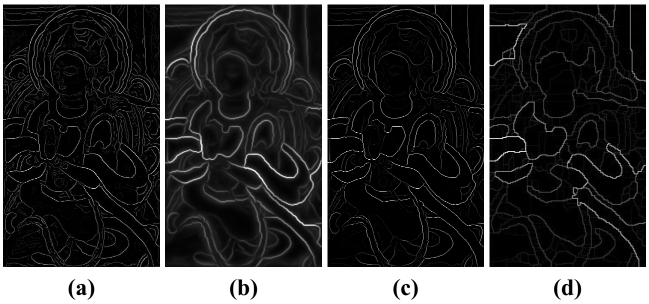


Fig. 3. The feature extraction. From left to right: mPb, sPb, gPb, closed contour.

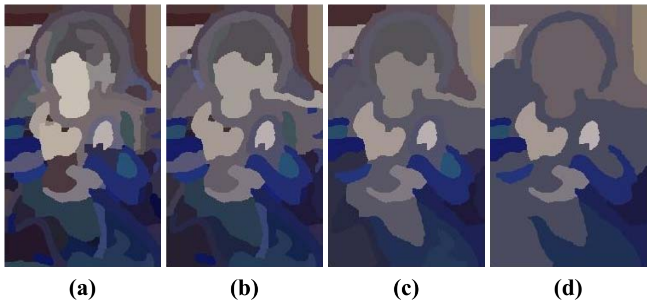


Fig. 4. The color structure image. The threshold T is $0.1, 0.12, 0.15, 0.2$ respectively.

ture extraction, the SIFT method [12] is a classical description approach by using a histogram to represent the difference of image shape and appearance. Recently, the BiCE method [13] is better than SIFT to record the feature of line.

3. THE PROPOSED FRAMEWORK

The goal of our framework is to generate sketch of a given fresco. There are four major parts in this framework (Figure 2). Firstly, we get frescos' structure and structure blocks (Figure 2(b)). Based on users' selection, we divide structure blocks into sets (Figure 2(c)(d)) and obtain corresponding regions (Figure 2(e)(f)) of original images. Then, we extract lines (Figure 2(g)(h)) and splice them into a full sketch (Figure 2(i)). Thirdly, we select and replace the missing contents. Finally, we provide the vectorization operation to optimize the result.

3.1. Hierarchical Structure Segmentation

To get frescos' structure from original image, we use hierarchical image segmentation technology [6]. There are two stages in this method, i.e. feature extraction and closed regions generation. Considering the importance and difference of lines in different regions (most lines may be noise in a region), users can divide structure blocks sets to gain different sketches using different parameters.

Globalized probability of boundary(gPb) based feature extraction gets primary structure contour of image. As the noise of fresco distributes randomly and unfit common Gaussian noise and Saline noise. we extract the gPb feature with texture channel as the fourth channel to control the impact of noise. Different from the method [6] used 17 Gaussian derivatives, we convolves the gray scale of image I (Figure 2(a)) with Gabor wavelet [14] to produce the texture channel, in order to extract more detail feature of structure.

The gPb feature (Figure 3(c)) combines the local contour mPb (Figure 3(a)) that focus on edge and the global sPb (Figure 3(b)) that pays attention to outstanding curve of image:

$$gPb(x, y, \theta) = \Sigma_s \sum_i \beta_{i,s} G_{i,\sigma(i,s)}(x, y, \theta) + \gamma \cdot sPb(x, y, \theta) \tag{1}$$

each pixel $p(x, y)$ is placed with a circular disc, which is split into two half-discs by a diameter at angle θ. The left side of plus sign is mPb feature, s stands for scales, i indexes four feature channels, $\beta_{i,s}$ is the weight of each channels relative contribution, and $G_{i,\sigma(i,s)}$ means the distance of histograms between two halves of a disc. Then we adopt spectral clustering to optimize above mPb and obtain new spectral feature sPb, γ is the weight of sPb.

Closed contour optimization recovers the gPb contours and gain hierarchical segmentation regions by using the Oriented Watershed Transform(OWT) [6] and Ultrametric Contour Map(UCM) [15]. Firstly, the OWT can produce a set of initial regions $E(x, y,)$from the gPb contours signal. Then using an agglomerative clustering procedure, we form these regions into a hierarchy that can be presented by an UCM, the real-valued image obtained by weighting each boundary by its disappearance. The hierarchical segmentation results include closed contours an closed regions for strengthening strong boundary and preserving weak boundary. Different thickness lines represent different hierarchical information (Figure 3(d)).

Then we fill each structure block with the mean color using different threshold T. Figure 4 shows different structure blocks by adjusting T. From the results, we can see that the smaller threshold gains more meticulous and clutter structure, while the larger threshold gains less and rougher structure. We provide a user interface for adjusting threshold to gain the color structure image E (Figure 2(b)).

Structure block sets division provides an interface that users can select structure blocks color to divide structure blocks into sets. The color of structure block instead of block itself to be divided. Some structure blocks with more noise or less lines can be divided into a same set in order to reduce generated lines, and others also can be divided into several sets to gain lines of varying degrees. The number of sets de-

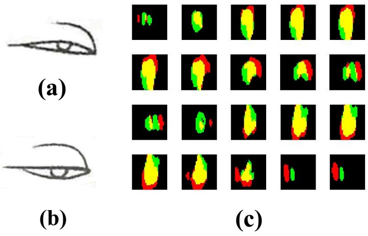


Fig. 5. The BiCE descriptors matching. The red and the green of image (c) represent respectively the eye element (a) and (b).

pends on the users themselves with reference to the noise and content information of structure block. In this paper, We divide the blocks into two block sets (Figure 2(c) and (d)) and gain the corresponding image areas (Figure 2(e) and (f)) in original image.

3.2. Sketch Generation

In this section, we generate the sketch S_k (k=1,2,...) of above each structure block sets and splice all sketches into a full sketch corresponding original image.

Line extraction gains basic sketch of fresco. At first, in order to control noise, we introduce the L_0 minimization method [16] to smooth original image areas (Figure 2(e)and(f)). Then, we choose the improved Canny [11] to detect edge S_k (Figure 2(g)and(h)) using different thresholds. If there are so many structure blocks sets to extract lines, the improved Canny method runs faster than the gPb method.

Sketch mosaic completes a whole sketch image. In accordance with the position in the image I, we merge all sketch images S_k into a whole sketch S (Figure 2(i)).

3.3. Missing Content Replacement

From S, we found some parts contain discontinuous or incomplete lines. Therefore, we build a database of feature descriptor according to existing Dunhung frescos, and provide a function for finding and replacing missing areas with several recommendations after users select missing areas.

Feature descriptors database creation derives from the fact that many contents of fresco repeatedly and similarly appear in history , and some artists have drawn a lot of sketches, such as Dunhung Fresco in China. Then we create a huge database (Figure 2) with partial sketch elements of above contents and their feature descriptors (Binary Coherent Edge) [13].

We collect a larger number sketches painted by artists, and extract the intact elements with specific categories, e.g., eye, mouth and nose. Elements in same category were stored in same size. Then we compute the feature of these elements. The BiCE descriptors describe an edge based on its orientation θ, position (x, y) and length l. Here we adopt a four dimensional histogram $H(x', y', \theta, l)$ to describe every line, where the new coordinate frame (x', y') aggregates position (x, y) with a standard 2D rotation matrix.

To improve the speed of matching, we binarize the subsampled histogram H's value by assigning the value 1 to the descriptors, which are higher than some threshold (the threshold is 0.2), and 0 to others. Then the edge histogram is binarized to encode edge locations, orientations and lengths. Figure 5 shows the descriptors of eyes in fresco. The red and the green of image (c) represent respectively the feature of eye element (a) and (b).

Missing content selection provides an interface for users to pick up the missing contents. We show the original image I and the sketch image S in user interface. The user can identify the differences by comparing S with I and manually select the missing contents caused by fresco itself or extraction process (Figure 2(j)) in I.

Missing content replacement make some parts of sketch more complete. The selected missing regions are smoothed and detected edge by the system. Then the BiCE descriptors gained will match all BiCE descriptors in database. The yellow regions in Figure 5(c) show the matching result between (a) and (b). To generate satisfying sketch, the system recommends several replaced elements of database for users (Figure 6). One of the recommendations may be user's expectation (Figure 2(k)).

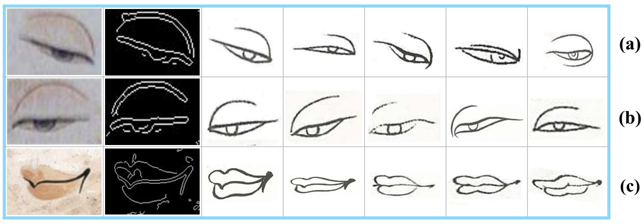


Fig. 6. The recommendations of missing regions. The left column shows the missing regions, the second column shows the edge of the missing regions, the other five columns show the recommendations ranked at the top five. (a)(b)(c) respectively index left eye, right eye and mouse of fresco.

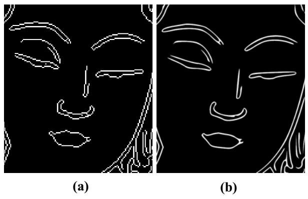
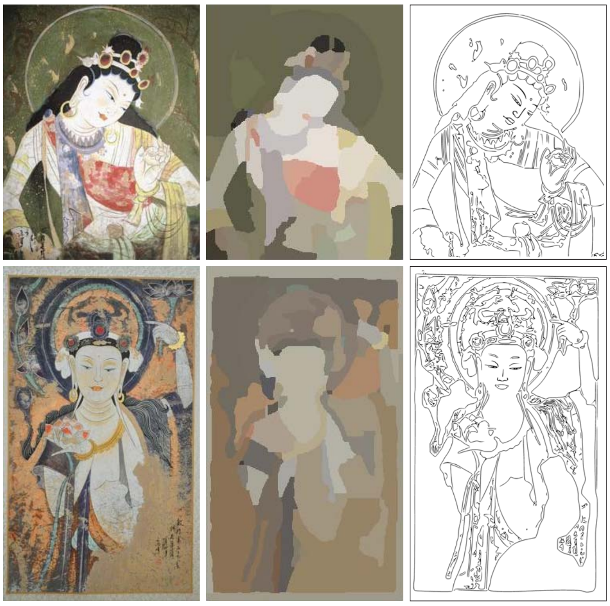


Fig. 7. The comparison between non-vectorization sketch image (a) and vectorization sketch image (b).



(a) (b) (c)

Fig. 8. The results with hierarchical structure. From left to right: original image, structure image and sketch image.

3.4. Sketch optimization

Sketch optimization provides user interface to ensure that sketch can be applied to multi-resolution and modify local sketch. This paper proposes a polygon tracking algorithm [17] to reconstruct vector representation of sketch. This algorithm decomposes edge of sketch image into several paths with black and white regions, and displaces these paths by appropriate geometry figures, finally, transforms the geometry figures into smooth contour. Comparisons between input sketch image and final vector-based sketch image is shown in Figure 7.

At last, a plain sketch (Figure 2(l)) with complete information outputs in our framework.

4. RESULTS AND EVALUATION

We collected Chinese Dunhuang Mogao Grottoes Frescos as experimental material. The input images of system were the original color image of fresco, and the sketch elements in database came from existing copying sketches painted by artists.Figure 8 demonstrates a better result through a hierarchical structure segmentation. We see the main structural contour and content still contain outstanding and detailed information of lines, but the background area with less lines and more noise is removed. In results, as damage and noise, complex lines in main structural blocks are impossible to be avoided. On the contrary, the phenomenon that the color is fading and the lines are not obvious has been prevalent in fresco, there are some incomplete lines in our method, as shown in Figure 9.

(a) (b)

Fig. 9. The results of some incomplete lines. The fill color and line color are closed in each box. In column (a), the boxes at the same location in different row images represent the same content. The same as column (b).

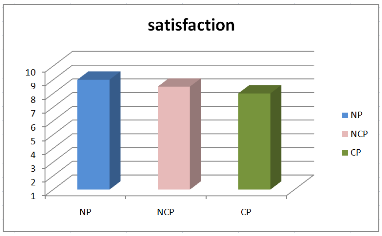


Fig. 10. The satisfaction of users. NP: Non-painting users, NCP: non-computer painters, CP: computer painters.

We have conducted a user study to measure satisfaction of our results. There were three categories of users: non-painting users(NP user); non-computer painting users(NPC user); computer painting users(CP user). The number of each category was thirty and those were chosen randomly from our university to participate in the study. We provided original fresco images and our sketch images to all users, they gave the satisfaction score (1-10, high score means high satisfaction) of our results. To evaluate effectively, we also provided some conditions as a reference, such as the proximity between original fresco and sketch, the importance of remain content, the integrity of lines, the artistry of lines and so on. The result of evaluation is shown in Figure 10.

We can see almost all the investigators are satisfied about our results. Non-painting users felt our results were more convenient to copy fresco. For painter and artists, our results were very realistic and art. Among them, non-computer painting users thought our method was very novel and meaningful for them to paint, and computer painting users thought this method can be applied to bulk for sketches, but there are also some limits and additional operations, such as remove some noise part artificially.

Our framework focus on Dunhuang Mogao Grottoes Frescos, but also apply to other frescoes with obvious feature of lines, even other painting art image. The resulting sketch can be widely used in many fields. For example, as an adjunct of the research and analysis about art pictures, as a learning tool to copy sketch in art world and as a novel exhibition way in Museum of Art.

5. CONCLUSION

In this paper, we have analyzed the process of sketch generation, and proposed a sketch generation framework including: structure extraction, sketch generation, missing content replacement and sketch optimization. The final results are satisfactory for research and application.

For future work, we suggest two aspects for studies:

(1) There are some limitations of our study about fresco itself. For instance, painting style is also an important characteristic to measure in all stages of sketch generation. So the sketch results can be optimized better. One possible way to approach this problem is to create a larger feature database including sketch style feature.

(2) We will invite artists to measure and apply our system to improve our results and obtain better sketch.

6. REFERENCES

[1] Yong Jae Lee, C. Lawrence Zitnick, and Michael F. Cohen, "Shadowdraw: real-time user guidance for freehand drawing," in *ACM SIGGRAPH 2011 papers*, New York, NY, USA, 2011, SIGGRAPH '11, pp. 27:1–27:10, ACM.

[2] Mathias Eitz, James Hays, and Marc Alexa, "How do humans sketch objects?," *ACM Trans. Graph. (Proc. SIGGRAPH)*, vol. 31, no. 4, pp. 44:1–44:10, 2012.

[3] Jingwan Lu, Fisher Yu, Adam Finkelstein, and Stephen DiVerdi, "HelpingHand: Example-based stroke stylization," in *ACM Transactions on Graphics (Proc. SIGGRAPH)*, Aug. 2012, vol. 31, pp. 46:1–46:10.

[4] Hyung W. Kang, Wenjie He, Charles K. Chui, and Uday K. Chakraborty, "Interactive sketch generation," *The Visual Computer*, vol. 21, pp. 821–830, 2005.

[5] Yingge Qu, Tien-Tsin Wong, and Pheng-Ann Heng, "Manga colorization," in *ACM SIGGRAPH 2006 Papers*, New York, NY, USA, 2006, SIGGRAPH '06, pp. 1214–1220, ACM.

[6] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, May 2011.

[7] T.G. Smith Jr., W.B. Marks, G.D. Lange, W.H. Sheriff Jr., and E.A. Neale, "Edge detection in images using marr-hildreth filtering techniques," *Journal of Neuroscience Methods*, vol. 26, no. 1, pp. 75 – 81, 1988.

[8] J Kittler, "On the accuracy of the sobel edge detector," *Image and Vision Computing*, vol. 1, no. 1, pp. 37 – 42, 1983.

[9] John Canny, "A computational approach to edge detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. PAMI-8, no. 6, pp. 679 –698, nov. 1986.

[10] Ehsan Nadernejad, Sara Sharifzadeh, and Hamid Hassanpour, "Edge detection techniques: Evaluations and comparison," *Applied Mathematical Sciences*, vol. 2, no. 31, pp. 1507–1520, 2008.

[11] Peter Meer and Bogdan Georgescu, "Edge detection with embedded confidence," *IEEE Trans. Pattern Anal. Machine Intell*, vol. 23, pp. 1351–1365, 2001.

[12] David G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[13] C. Lawrence Zitnick, "Binary coherent edge descriptors," in *Proceedings of the 11th European conference on Computer vision: Part II*, Berlin, Heidelberg, 2010, ECCV'10, pp. 170–182, Springer-Verlag.

[14] B.S. Manjunath, "Gabor wavelet transform and application to problems in early vision," in *Signals, Systems and Computers, 1992. 1992 Conference Record of The Twenty-Sixth Asilomar Conference on*, oct 1992, pp. 796 –800 vol.2.

[15] Pablo Arbelez, "Boundary extraction in natural images using ultrametric contour maps," in *In Proc. POCV*, 2006.

[16] Li Xu, Cewu Lu, Yi Xu, and Jiaya Jia, "Image smoothing via l0 gradient minimization," *ACM Transactions on Graphics (SIGGRAPH Asia)*, 2011.

[17] Peter Selinger, "Potrace: a polygon-based tracing algorithm," in *In http://potrace.sourceforge.net*, 2003.