EE 440 Autumn 2017
Final Project: Image Artistic Stylization
Due: December 12, 2017, 11:59PM

1 Description

The purpose of this project is to show your ability to identify, formulate and solve image processing problems, and perform a demonstration of what you have developed. With what we have learned so far, we are already able to create many artistic effects from real photos. In this project, you will develop a simple **graphic user interface (GUI)** that can **load an image, apply artistic effects and display the input/output**. You can explore interesting image processing possibilities based on the materials taught in this course. We also encourage you to search and discover related literatures and books that benefit the artistic effect you want to achieve.

2 Artistic Effects

You only need to choose one to implement, you are not required to achieve the same results as the example images shown below. Stroke-based effects and water-color effects are relatively difficult (bonus credits). Other effects could be achieved either by applying algorithms in the lecture notes or searching and implementing research papers. You will receive extra points if:

- Implementation is challenging
- Output is outstanding
- Creative ideas
- Interesting combinations or add-ons

We provide several examples that may give you some intuitive ideas in your work:

A. Cartoon Effect

This could be implemented by a bilateral filter, then enhance the edges of the image.

(a) Original Photo (b) Cartoon Effect

Figure 1: Cartoon Effect

B. Illustration Effect

For the illustration effect, we add edge information and enhance the image using high boost filtering so that the content in the image can be viewed with more details.

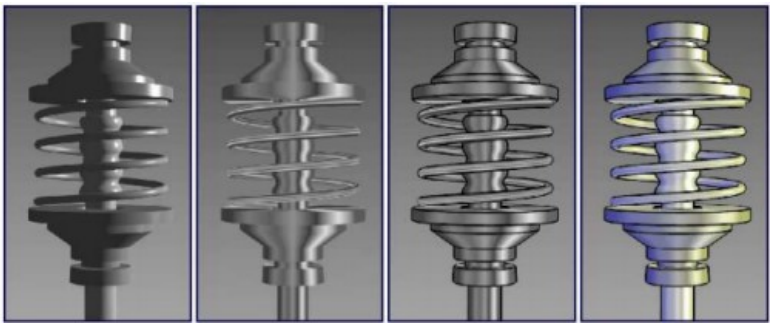

Figure 2: Illustration effect

C. Segmentation effect

The simplest situation is image binarization using Otsu's method. To get more segments, Otsu's method could be applied multiple times. K-means is another easy way to implement the segmentation-based effect.

Figure 3: Segmentation [4]

D. Stippling and Halftoning Effect

Represent an image using small dots. The sizes and colors of the dots could be determined by the local color or gradient of the image. The primitive shape does not need to be limited to dots. It could also be other shapes or mixture of different shapes.

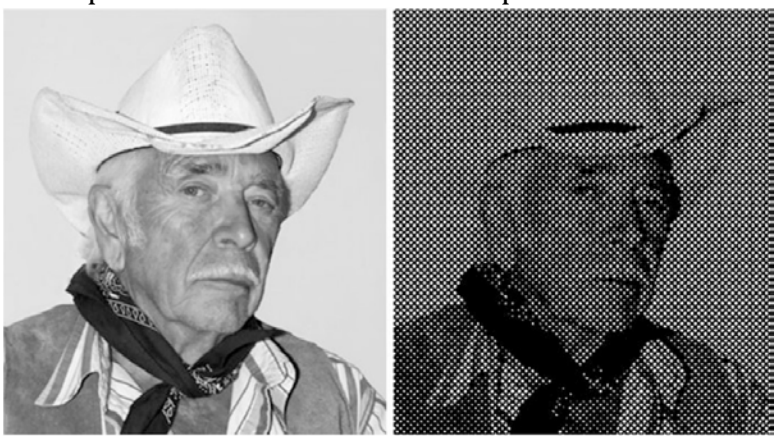

Figure 4: An example of halftoning effect [3]

E. Brush Strokes based Effects

You can analyze the image gradient, then use the gradient information to determine the stroke size, orientation, and shape.

E1. Oil Painting Effect

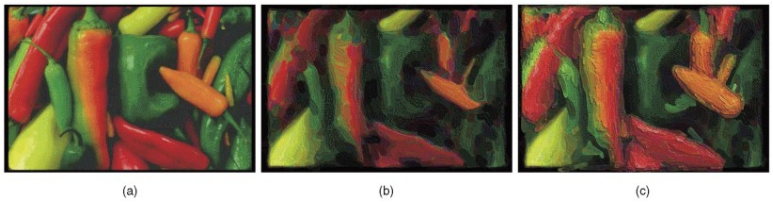

Figure 5: Oil painting effect [2]

E2. Pen and Ink Effect

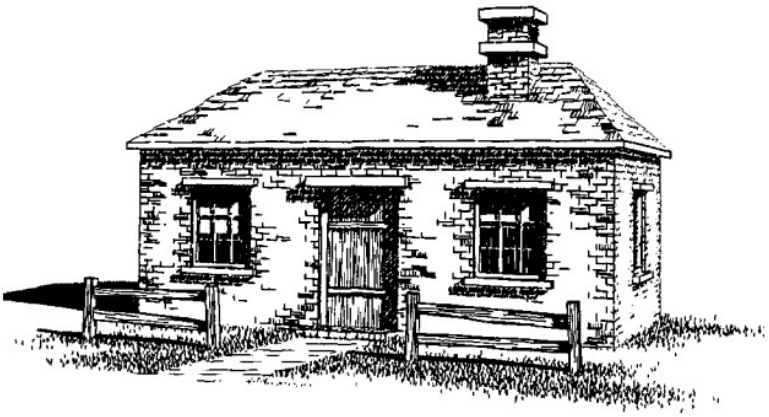

Figure 6: Pen and ink effect [5]

F. Water Color Effect

Color palettes need to be quantified and reduced. Morphological operations such as dilation, erosion, thickening, etc. are also needed.

Figure 7: Water color effect [1]

G. Your own artistic effect

This could be any effect you think is cool, please contact the TA if you plan to implement your own artistic effects.

3 Related Techniques

Here is an **incomplete** list of techniques that might be useful:

For preprocessing:

- Histogram adjustment: e.g. equalization, specification, etc. to adjust the dynamic range
- Adjust RGB color values
- Median filter: e.g., remove image noise
- Gaussian filter: e.g., smooth the image before the gradient calculation
- Image resizing: e.g. create a pyramid of images

For artistic effects:

- Image segmentation
- Gradient analysis: e.g., to determine stroke direction and size
- Edge detection: e.g., emphasize the edge in the cartoon effect
- Image Filtering: e.g., create a cartoon effect using bilateral filtering
- Morphological operation: e.g., opening and closing for water color like effect
- Color quantization: e.g. reduce the number of colors to make the image look artificial
- ...

Useful links:

- An overview of artistic stylization of images and videos
- Slides about Non-photorealistic Rendering (NPR)
- Paper about Stroke based methods
- Pen and ink effect

- A book about image-based artistic stylization

4 GUI Design

The GUI will consist canvas areas where images are displayed, scroll bars for users to adjust the parameters or radio buttons to select different settings (if necessary). Below shows an example of GUI design. Your application does not need to have the same interface. Customize your own GUI depending on what you want to implement.

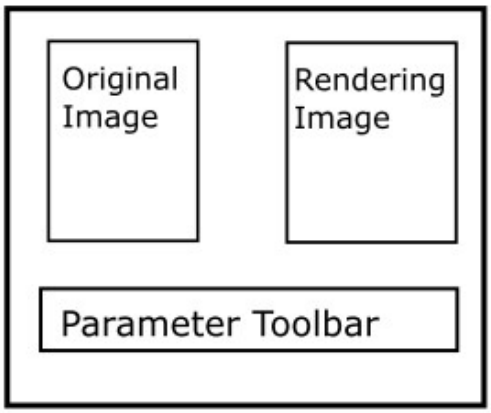

Figure 8: An example GUI

We also attach a few sample code in MATLAB of simple image processing GUI programs. You can use any programming language (MATLAB, PYTHON(PYQT), JAVA, C/C++(QT/C#)). If you want to use external image processing libraries (e.g. OpenCV), please make sure to build a standalone executable upon submission. We encourage you to implement the algorithm in real-time. But depending on the complexity of the algorithm and the implementation language, this is not required.

5 Grading Guideline

You can use any images for testing. You need to include at least 3 images and its artistic results from your algorithm ***in your final report***. There is no standard about how the rendering results should look like. The final grading is based on:

- Functionality (Difficulty) of your algorithm
- GUI design
- Rendering result

6 Submission

Please follow the homework submission guidelines. Be sure to zip all the necessary files including scripts, images in a .zip file. ***Remember to submit your .zip file and .pdf report separately in Canvas.*** In your .pdf project report***, you should contains following sections***:

- Artistic effect you want to achieve
- Details of the algorithm and implementation
- Experimental results and discussions
- Instruction on how to run all functionalities of your program. (We will not be able to grade your project if we don't know how to run your program.)
- Briefly Comments on what you have learned and new features that can be added in the future.

7 Deadline: **12/12 (Tuesday) before 11:59pm (midnight). There will be no extension for the deadline.**

References

[1] Adrien Bousseau, Fabrice Neyret, Jo¨elle Thollot, and David Salesin. Video watercolorization using bidirectional texture advection. In *ACM Transactions on Graphics (ToG)*, volume 26, page 104. ACM, 2007.

[2] Mizuki Kagaya, William Brendel, Qingqing Deng, Todd Kesterson, Sinisa Todorovic, Patrick J Neill, and Eugene Zhang. Video painting with space-time-varying style parameters. *IEEE transactions on visualization and computer graphics*, 17(1):74–87, 2011.

[3] Paul Rosin and John Collomosse. *Image and Video-Based Artistic Stylisation*, volume 42. Springer Science & Business Media, 2012.

[4] Yi-Zhe Song, Paul L Rosin, Peter M Hall, and John P Collomosse. Arty shapes. In *Computational aesthetics*, pages 65–72, 2008.

[5] Georges Winkenbach and David H Salesin. Computer-generated pen-and-ink illustration. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 91–100. ACM, 1994.

You do not need to read those papers to complete this project.