



**ARTICLE**

# Stroke Based Painterly Rendering with Mass Data through Auto Warping Generation

Taemin Lee<sup>1</sup>, Beomsik Kim<sup>2</sup>, Sanghyun Seo<sup>3</sup> and Kyunghyun Yoon<sup>4,\*</sup>

<sup>1</sup>Davinci SW Education Institute, Chung-Ang University, Seoul, 06974, South Korea

<sup>2</sup>School of Computer Science and Engineering, Chung-Ang University, Seoul, 06974, South Korea

<sup>3</sup>College of Art & Technology, Chung-Ang University, Gyeonggi-do, 17546, South Korea

<sup>4</sup>School of Computer Science and Engineering, Chung-Ang University, Seoul, 06974, South Korea

\*Corresponding Author: Kyunghyun Yoon. Email: khyoon@cau.ac.kr

Received: 23 June 2021 Accepted: 13 September 2021

## ABSTRACT

Painting is done according to the artist's style. The most representative of the style is the texture and shape of the brush stroke. Computer simulations allow the artist's painting to be produced by taking this stroke and pasting it onto the image. This is called stroke-based rendering. The quality of the result depends on the number or quality of this stroke, since the stroke is taken to create the image. It is not easy to render using a large amount of information, as there is a limit to having a stroke scanned. In this work, we intend to produce rendering results using mass data that produces large amounts of strokes by expanding existing strokes through warping. Through this, we have produced results that have higher quality than conventional studies. Finally, we also compare the correlation between the amount of data and the results.

## KEYWORDS

Painterly rendering; stroke based rendering; image mass data; stroke warping; non-photorealistic rendering

## 1 Introduction

Non-realistic rendering is a technique that simulates the techniques expressed by artists, as opposed to photorealistic rendering. With the development of realistic rendering technology, all expressions became possible, and the desire for these artistic expressions began to increase. Painting rendering, a field of non-realistic rendering, refers to a technology that receives two-dimensional images or three-dimensional model data and renders them. In pictorial rendering, there is a style transfer technique [1–6] with basic units in pixels and a stroke-based rendering [7–12] with images. Because stroke-based rendering proceeds with the base unit as a stroke image, the result depends on the stroke used.

In the case of stroke-based rendering, it has the advantage of being able to easily produce the desired results using a brush in the DB. If you have Gogh's brush stroke, it means you can use it to paint Gogh's style. For real paintings, the shape of the brush strokes that make up the object is



determined along the lines of the object, so it is drawn as intended by the artist. In other words, there are many different forms of lines. However, traditional stroke-based rendering techniques represent images using fewer DBs, leading to object distortion. Fig. 1 shows the problem caused by the absence of DB. Since there is no DB for the bridge part of the sunglasses marked with a red circle, this part can be omitted in the rendering stage, resulting in object distortion.



**Figure 1:** Example of rendering results that has distortion compare with original input image (As we can see the red circled area, the frame is broken on the sunglasses)

There are two main ways to improve rendering results. First, when rendering brushes in DB and using them, they are directly deformed to fit the line. Second, increasing the number of brushes in the DB to cover the shape of most edges. We propose a method that can generate brush strokes with a focus on the latter. The following conditions require satisfaction when producing a new brush stroke. First, the shape of the brush stroke should be as natural as the actual human drawing. Second, the texture of the brush stroke must be realistic. Third, the shape of the stroke that can be generated reflects the user's intention and must be various shapes. Fourth, it should also be applicable to databases with low diversity. To this end, we applied image-warping technology. Image warping applied to generation methods is the task of deformation, processing, and reproduction of input images, which can be deformed and extended in multiple directions. That is, it is the same as drawing on elastic materials, moving certain parts of the material, and fixing them. In addition, we propose an improved brush stroke judgment algorithm to minimize the omission or distortion of objects in the input image with the brush stroke database we hold. The improved judgment algorithm determines the optimal brush with parameter information used in the production of brush strokes as well as color conformation through color distances. Through the production of brush strokes, our DB has mass data based on less image data. Our work improves the quality of rendering results by selecting brushes that match the rendering results among many brushes.

The content of the paper is described as follows. Section 2 describes existing painterly rendering studies. It also describes the areas that can be developed and used in this study, referring to existing warping techniques. Section 3 describes stroke generation methods and rendering methods. Stroke generation methods enable passive and automatic generation. An improved rendering system is used to select the appropriate brush from the increased DB. In Section 4, we compare the rendering results of Section 3 and show improved rendering results through increased DB. Finally, we summarize our method and conclude our results and future development.

## 2 Related Work

Hertzmann [13] focused on not only straight lines but also various shapes of brush stroke. In this paper, an anti-aliased cubic B-spline is used for render paintings as a curved brush stroke.

Furthermore, paintings are created by stacking layer-specific images after making different brush sizes for each layer. The biggest significance here is an extension of the shape of the brush stroke from a straight line to a curved shape.

There is a paper that uses a method of generating brush strokes is a stroke profile that is created by drawing directly. Seo et al. [14,15] obtained black emulsified pigments by drawing and capturing them on translucent glass plates that had been shot from the back. Afterwards, five-shape brush strokes were generated by eliminating noise generated during capture through pre-processing. As a way of expanding the database, the five brush strokes obtained were transformed into images such as rotating and resizing. This approach has the advantage of being able to obtain exactly what users want.

Another study generating new brush strokes has been studied how to use Texture Distortion and Image warping. Lu [16] proposes a method to synthesize existing brush strokes to generate new brush strokes. The method obtains a centerline for the beginning to end of the brush stroke and samples it with pixels at regular intervals. When sampling, specify a slightly wider angle than the outer angle of the brush stroke, so that it has a small margin. Information about the sample is stored by coordinating the left outer line, the right outer line, and the center line with respect to the center. Then, the centerline of the desired brush stroke shape, the left outer line, and the right outer line are established. Finally, we place samples of existing brush strokes like the coordinates of the shape we want to generate as image warping. The texture of the resulting brush stroke is improved through texture distortion. These methods have the advantage of having high degrees of freedom in the shape of brush strokes that can be generated.

Hertzmann [17,18] devised a layered grid-based stroke generation method to represent omitted objects. Layered grids find incomplete canvas and maximum error points in each grid cell. The image gradient information is then used to paint a new stroke where the difference is significant. This method is a very efficient way of representing pictorial images. Furthermore, it is borrowed from most stroke-based pictorial rendering algorithms.

Several methods of appropriately positioning objects for representation have also been studied using various generated brush strokes in the above methods. Hertzmann first proposed a multi-layered grid algorithm that automatically expresses multiple stroke sizes. The algorithm has actually algorithmized the drawing process and can be expressed in various sizes based on color difference values. For example, a large background or area is expressed using a large stroke. And parts that need to be detailed or have a small area are expressed with small brush strokes. In this way, rendering results similar to real paintings can be obtained, and this process has been widely adopted in pictorial rendering.

Further to Hertzmann's proposed method, Seo et al. [14] proposes a more improved method. Seo et al. utilized pre-made brush stroke textures in details such as the size of the brush and the starting position of the brush. The spacing of the lattice for each brush size is smaller than the size of the brush, so that there is no empty space as much as possible. The position of the stroke is determined randomly within the grid cell at a time for each brush size without being affected by the intermediate results drawn on the canvas. These methods have the advantage of saving money for the process of finding the maximum error position, where the difference between canvas and input images is significant.

All existing studies [17–20] have been used to improve the quality of results using traditional brush strokes or to draw styles for specific artists. In recent years, existing studies [9,21] have proceeded to express the results differently by adding sensitivity or changing lighting. We would

like to propose a study that expands the brush stroke database to improve the results of traditional stroke based rendering. Through the expansion of the database, we propose a painterly rendering system using a large amount of data, i.e., mass data, as in recent studies [22–25].

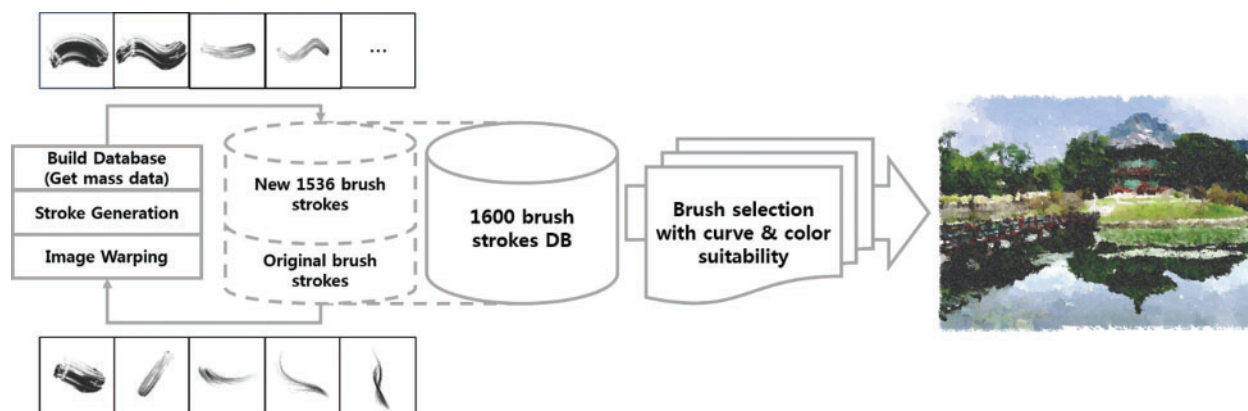
### 3 System

#### 3.1 System Overview

Existing brush stroke rendering studies determine the quality of the results, depending on the number of databases. The more databases, the more diverse expressions you can make, and the less, the simpler the expression. If the database is small, the smaller database is supplemented by image conversion, such as rotating the stroke or resizing it. However, it was not easy to overcome this limitation by modifying it based on a database. In this work, we propose a study that increases the amount of brush stroke databases through image warping, thereby increasing the number of brush shapes in a small database. In our work, we compare these diverse generations of data, and we also work on finding brushes that can be used frequently in paintings. This can be seen as a task to select data from mass data.

The method proposed in this paper can also be applied to databases with low diversity and can generate brush strokes of various shapes simply and quickly, which can quickly generate mass data. Furthermore, the newly produced stroke by applying image warping keeps the texture of the existing stroke intact, so no loss of information occurs. It can also be applied to low-diversity brush stroke databases and can generate a variety of shapes as intended by the user. Image warping applied to generation methods is the task of deformation, processing, and reproduction of input images, which can be deformed and extended in multiple directions.

Finally, we present a novel brush judgment process to effectively represent the brush strokes generated. As a judgment method, we compare the color conformation and shape conformation of images and strokes to select suitable brushes. The overall overview of this paper is shown in Fig. 2.

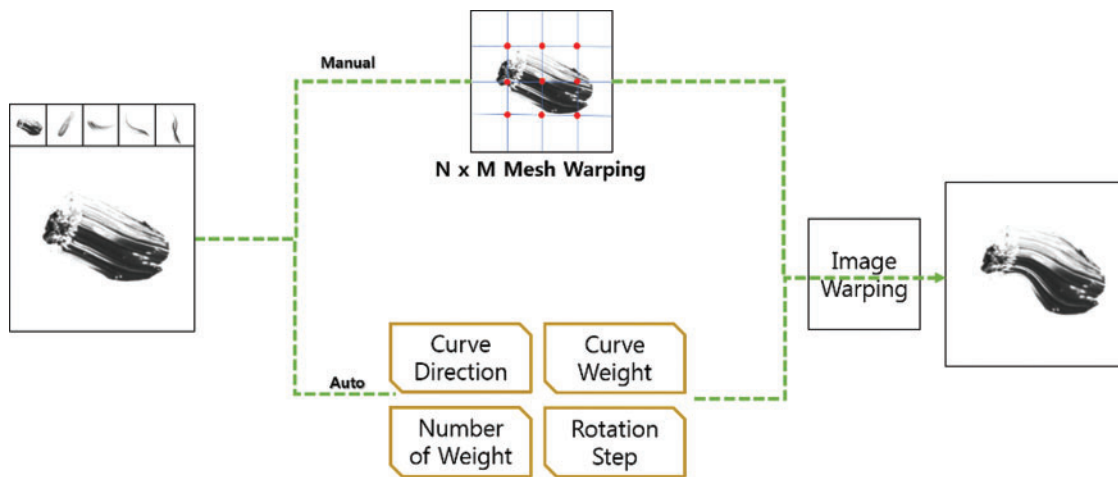


**Figure 2:** Overall overview (various brush strokes are generated by manual/auto warping. Painterly rendering results are generated by these brush strokes with curve & color suitability)

#### 3.2 Method of Generating Brush Strokes

We generate brushes in two forms of generation. One is Manual Mode, which creates shapes directly through mouse and keyboard input, and the other is Auto Mode, which receives multiple

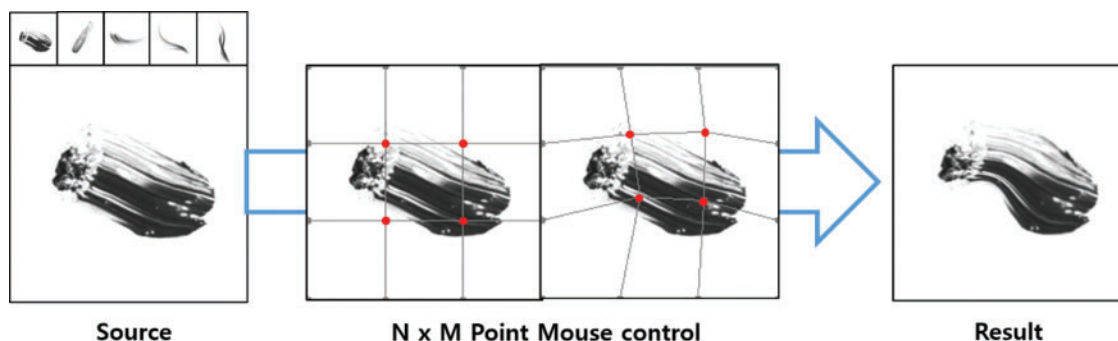
parameters. Parameters used in the automatic generation method are the curve direction (Cd), the curve weight (Cw), the number of curve (Wn), and the rotation step (Rs). The manual generation method uses mesh warping, which adjusts control points with mouse input. The overall overview of how brush strokes are generated is shown in Fig. 3.



**Figure 3:** Process of generating brush strokes (we define 4 parameters for generating various brush strokes automatically. In manual, we set 2 integer parameter N, M which are the point of warping)

The control points applied to the manual generation method are inputted separately by the user and placed at regular intervals. Then, through mouse input, information about the starting points (srcPoints) and target points (dstPoints) required for image wrapping is entered. The manual generation method can visually and real-time check of the generation process of brush strokes whenever the information of the target point is modified. Compared to automatic generation methods, these methods have the advantage of being able to check in real time for the intended shape of the user.

If the user enters the information of the target point incorrectly and does not appear to be intended, it returns the target point to one step before it is turned into unintended information through keyboard input. These methods can improve completeness when determining the shape of a new brush stroke and increase user convenience. Fig. 4 is the result of creating a new shape of brush stroke using a manual generation method.

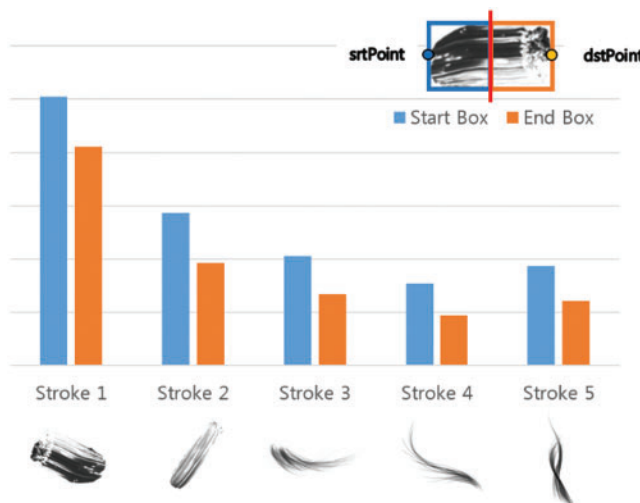


**Figure 4:** Process of Manual generation method (the number of red points is determined by N&M)

### 3.2.1 Curve Direction

There are two possible deflection directions: up and down, relative to the center of the brush stroke. When creating a brush stroke by deflecting it only once, it simply needs to be drawn in the direction up and down. However, when drawing by bending more than twice, the direction of initial bending should be considered depending on the direction of the stroke. Therefore, in order to apply the direction above and below relative to the stroke, it is necessary to know the direction of progress in which the brush stroke is drawn. Therefore, we identify the start point and end point of the brush stroke.

Finding the start and end points is easy considering how to generate a real stroke. When drawing, apply dye to the brush and press it on the canvas to start drawing. And as you brush it when you complete it, you complete the stroke, so the beginning of the stroke is more dyed than the end of the stroke. Therefore, we divide the corresponding portion of the brush stroke in half in the stroke image and compare the histograms of the two regions to obtain the start and end points. The method of obtaining poetic and end points first moves the brush stroke image to the center. Then, a bounding box containing the entire brush stroke is obtained and the bounding box is bisected horizontally to set it to the Start Box at the beginning and the End Box at the end of the brush stroke. In the bounding box, the outer focus is the starting and ending points, which are the legend of Fig. 5. Fig. 5 is a graphical representation of the histogram values of the two boxes set in the brush stroke by the method described above.



**Figure 5:** Histogram comparison results for start and end boxes (calculate a black histogram by dividing left and right based on the midpoint of the bounding box. The 1,4,5 brushes have a large histogram value on the right, and the 2 and 3 brushes have a srtPoint on the left)

### 3.2.2 Curve Weight

In the case of curve weight, the method of granting the shape of the newly created brush stroke to be as similar as possible to the one drawn by humans was anthropologically approached. In a normal person, it is assumed that the arc is drawn only by the movement of the wrist when the upper part of the arm is fixed. On average, a circle can be drawn with a maximum of 130 degrees in the cabinet, 60 degrees when the wrist is moved up and 70 degrees when moved down. That is, assuming that the brush stroke is drawn in the form of a curve with wrist-only motion

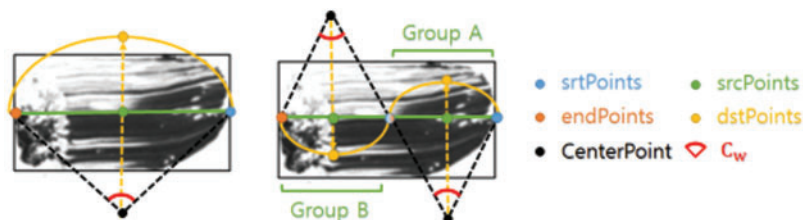
in the same environment as the existing brush stroke, the brush stroke can be drawn in the shape of a circle with a cabinet of 130 degrees.

Thus, the points that make up the line segments with the start and end points of the brush stroke as both endpoints are the starting points (srcPoint) at the time of image warping, with the arc's interior weighted by the curve. Arc's main point is that the curve weight entered by the user is the cabinet, and the first and last points of the starting points are the endpoints of the radius, respectively. The target point (dstPoint) corresponding to the starting point is to start from the center of the arc and meet the arc in a straight line past the current starting point. In this paper, with the above information, curve weights can be established by dividing the curve weights into 11 stages in total from 0 (original) to 10 (130 degrees of interior angle).

### 3.2.3 The Number of Curve

Increasing the number of bends allows multiple places to give curvature weights. In other words, since drawing by deflecting brush strokes is drawing by dividing strokes by the number of deflections, we approach the approach in the same way as single curvature strength. When given a single curvature strength, the target points were obtained by grouping a set of starting points into one group. Therefore, when giving multiple curvature intensities that deflect  $W_n$  times, the starting points are grouped by  $W_n$ . The curvature intensity to deflect  $W_n$  times is also inputted  $W_n$  times. The target points corresponding to the starting point of each group are then obtained in the method of [Section 3.2.2](#).

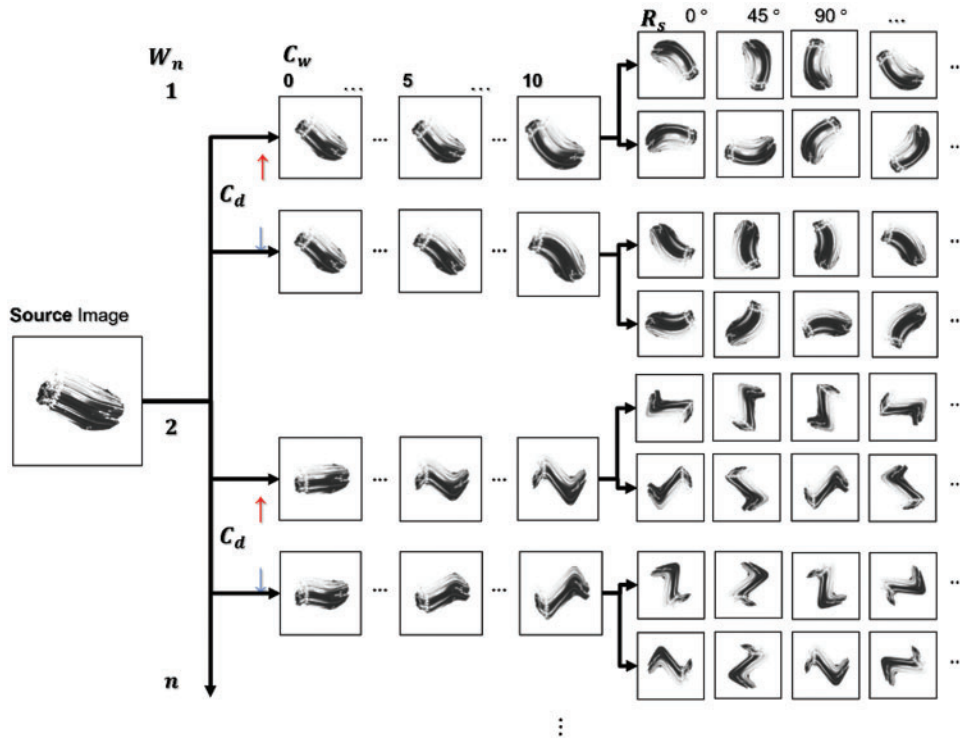
To create a natural brush stroke shape, the first group corresponding to the starting point follows the curve direction entered by the user. The curve direction of the next group is then given in the opposite direction of the previous curve. [Fig. 6](#) shows information on brush strokes at multiple curvature intensities given two curvature intensities and how to obtain a target point corresponding to the starting point.



**Figure 6:** How to set the information for the origin and destination points. Left:  $W_n = 1$ , Right:  $W_n = 2$ , As the  $W_n$  value increases, the number of warping increases.

### 3.2.4 Rotation Step

The database is expanded through image rotation to allow the newly created brush strokes to have multiple orientations. Rotation can be divided and rotated according to the  $R_s$  stage entered by the user from 0 to 360 degrees. [Fig. 7](#) is the process of producing a new brush stroke by suggesting [Section 3.2.1](#) through 3.2.4. After giving up to  $W_n$  number of deflections, the curvature intensity is given by  $C_w$ .



**Figure 7:** New strokes generated by auto generating method from the one original source stroke

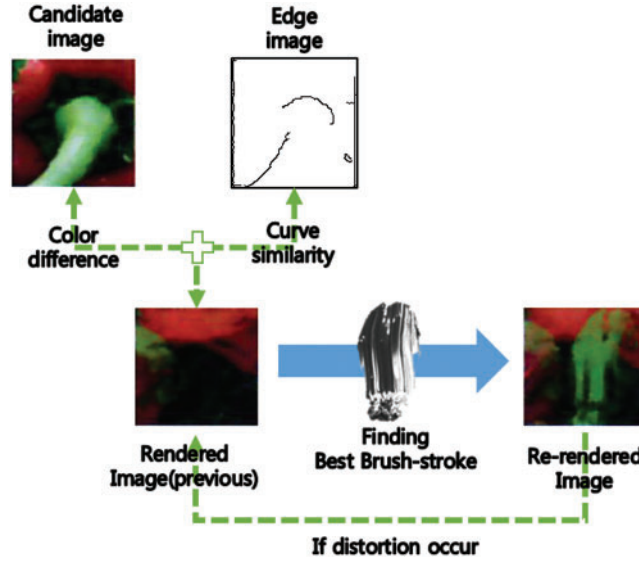
### 3.3 Rendering System with Appropriate Brush Selection

Through the method proposed in this paper, we generate multiple shape brush strokes based on existing brush strokes. Based on the brush strokes generated with the preceding content, we compute the average color distance and curve fit between candidate images and brush strokes to select the optimal brush strokes suitable for representing objects in the input images. The candidate image is the size of the brush stroke, focusing on the try position you want to represent by painting the brush stroke on the input image, and the color you paint on the brush stroke is the color of the probe position on the candidate image. The average color distance is calculated as the average Euclidean distance ( $D(I_1, I_2)$ ) of each RGB color between candidate images ( $I_1$ ) and brush strokes ( $I_2$ ) (Eq. (1)). Curve fit is calculated as curve similarity ( $D(P_d, P_g)$ ) the average Euclidean distance of the target points ( $P_d$ ) and the edge ( $P_e$ ) of the candidate image close to each target point (Eq. (2)). Edge images obtain edges from the input image with FDoG and then use edge-thinning images for accurate distance calculations. The brush stroke-specific score (Bscore) for selecting the most suitable brush stroke in the position where the stroke is to be painted is obtained by multiplying the average color distance and curve fit by weights, respectively, to sum the calculated scores (Eq. (3)). The brush stroke with the lowest score among the calculated scores is the optimal brush stroke. Fig. 8 shows this process.

$$D(I_1, I_2) = \frac{\sum_x \sum_y |I_1(x, y) - I_2(x, y)|}{w * h} \quad (1)$$

$$D(P_d, P_e) = \frac{\sum_s |P_d(s) - P_e(e)|}{e - s + 1} \quad (2)$$

$$B_{score} = w_{col} * D(I_1, I_2) + w_{cur} * D(P_d, P_e) \quad (3)$$



**Figure 8:** Process for finding the best brush-stroke (the color similarity and curve similarity are considered together to find the best brush stroke. Since our work is made with various curved brushes through warping, many brushes produced according to the weight of curve similarity can be used)

## 4 Experimental Result

### 4.1 Development Environment

Our system is built in a MFC in Windows 10 64-bit, Microsoft Visual Studio 2017, and used the Opencv 3.3.0 library. The computer hardware used for implementation and testing is the Intel Core i7-4790 CPU (3.60 GHz) with a standard PC and 32 GB RAM, and the graphics card is the GTX 960.

When generating brush strokes, the deflections were set up and down ( $C_d=2$ ), and the curvature intensity ( $C_w$ ) was given differentially according to 10 steps. The number of deflections ( $W_n$ ) was generated by deflecting up to two times, and the rotation ( $R_s$ ) for database expansion was generated by rotating 45 degrees in eight steps.

### 4.2 Rendering Result

Our work has the advantage of being simpler and faster generation speed than conventional methods [17–20] because it gives multiple weights. Furthermore, more brush strokes could be used than Lu's method [16] due to the varying number of database. We proceed with rendering using a total of 1600 databases, generating an additional 1536 in 64 brush strokes used by existing studies.

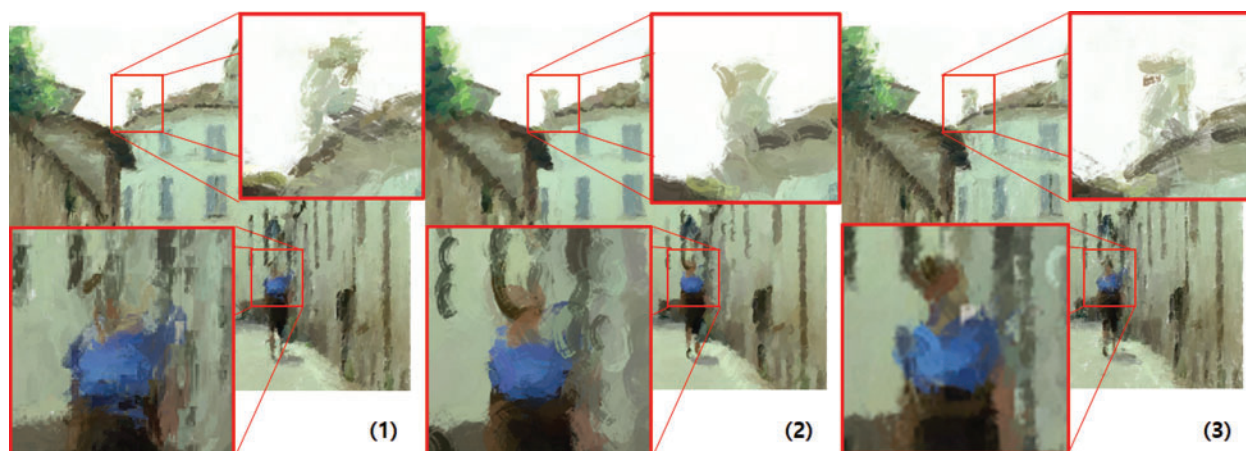
Fig. 9 shows how brushes stick when using only existing data and extended data when rendering a specific location on the same input image. If you clicked on the seat on a cruise ship, the existing database selected a brush stroke that covered the white part of the cruise ship or the blue background. However, in the case of an extended database, the result is a brush stroke that can render only the blackest part possible. As such, the results of rendering after selecting a brush for a particular location show that the non-color of the probe location is less painted than the

conventional brush stroke. Furthermore, we can see improved results in selecting strokes that do not deviate from the borders of the surrounding objects.



**Figure 9:** Results of selecting appropriate brush strokes from existing and expanded database (For the target image on the right, the first row shows a portion of the target image. The middle row is the brushstroke selected when rendering using only existing brushes, and the third row is the images selected from the extended brushstroke DB we generated)

Fig. 10 compares rendering results of existing database(64), newly created database(1536) and expanded database(1600). When rendered into an existing database, the upper body of the woman at the bottom is an object that could not be expressed with an existing brush stroke. In the case of the newly created database, the object was well represented, but only the curved shape was retained, so the tunnel bank, which is a straight line, seems to be poorly represented. The expanded database, combined with the two databases, has all the advantages of each outcome, as can be seen from the upper body of a woman and the chimney. Fig. 11 is a comparison of landscapes that require more detail. The leaves and flower sections are much more detailed and representative compared to the traditional few brushes.

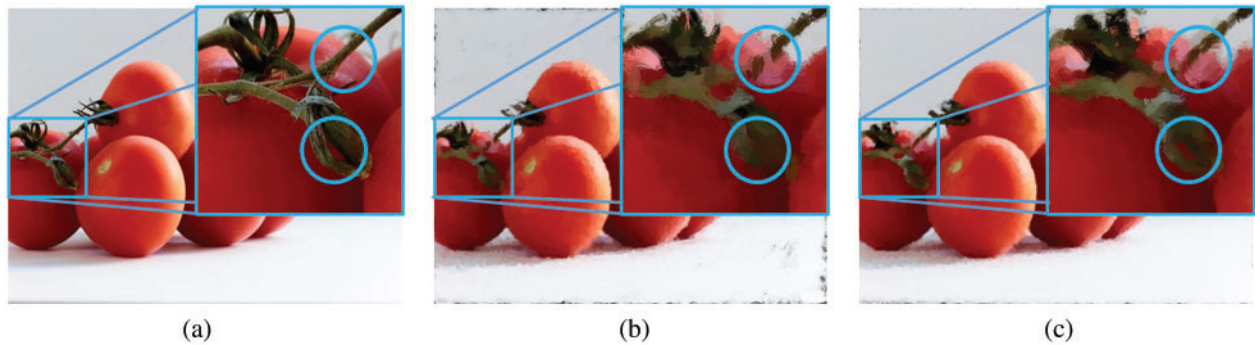


**Figure 10:** Rendering results based on number of brush strokes: (1) uses only 64 brush strokes—the original database, (2) uses 1536 brush strokes—only generated strokes, (3) uses 1600 brush strokes



**Figure 11:** Another rendering results based on number of brush strokes for landscape paintings (Left uses only 64 brush strokes–the original database. Right uses 1600 brush strokes)

The problem of objects being omitted can be solved because various generated brushes can also express details of objects. This can be seen by the comparison in Fig. 12. (a) is the input image, and (b) is the result of rendering by Seo's method [14]. (c) is the result of rendering in the method proposed in this paper. Looking at the blue circle, Seo's method [14] also shows a difference from the original image. However, our rendering results get no distortion.

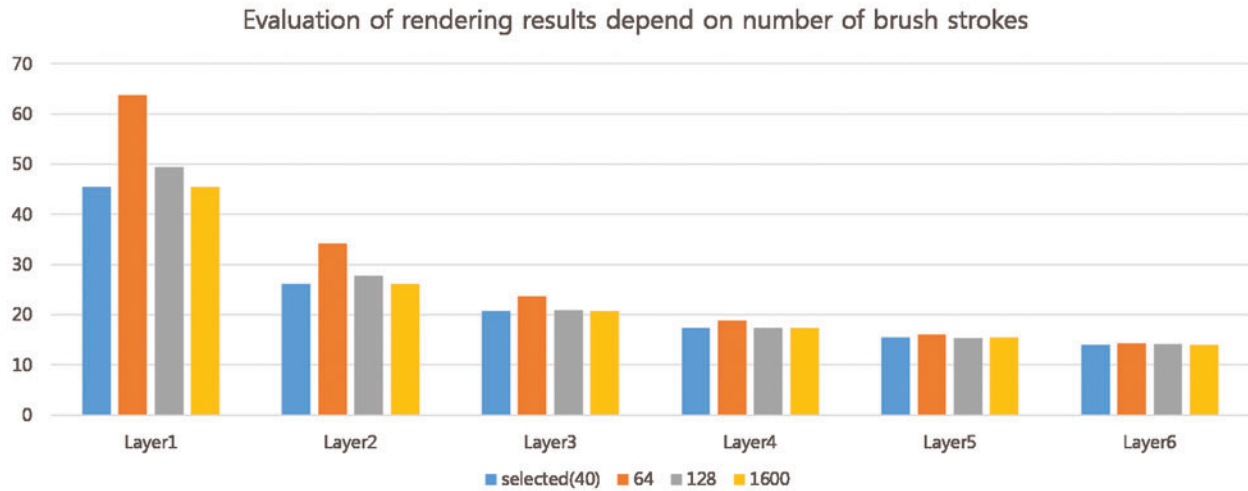


**Figure 12:** Comparison the results with Seo et al. [14] (We can find the difference in blue circles. Our results has no distortion at the tip and corner of the fruit)

Fig. 15 at the end of the paper-shows the results of the 128 stroke databases (turns  $Wn/2$  in  $Cd/2$  directions) with only half of the 64-stroke data previously used in Section 3 (turns  $Wn/2$  in  $Cd/2$  directions to  $Rs/2$  levels) and the database selected from the database. Fig. 13 and Table 1 shows evaluation of rendering results using image color similarity (see Eq. (4)). When the brush stroke is finally stretched, the quality of the results in Layer 6 is close to the original.  $D(I_s, I_r)$  is average Euclidean distance in the RGB color space of the input image and the resulting image.

And we evaluate that the shorter the distance, the more similar the input image, and the lower the score, the higher the completeness.

$$\text{Score}(I_s, I_r) = \frac{D(I_s, I_r)}{D_{max}} \quad (4)$$



**Figure 13:** Chart of rendering results by layer and the number of brush strokes

**Table 1:** Score and spending time of rendering

	Layer 1	Layer 2	Layer 3	Layer 4	Layer 5	Layer 6	Time(s)
Selected (40)	45.49	26.19	20.72	17.35	15.4	13.96	95
64	63.77	34.17	23.65	18.78	15.97	14.24	94
128	49.42	27.72	20.89	17.41	15.31	14.12	200
1600	45.49	26.19	20.72	17.35	15.39	13.96	2450



**Figure 14:** Various results using different textured brush strokes (Left is pastel, right is pencil drawing)

Fig. 14 is the result of applying input brush stroke image warping technology to other types of databases. Figures that the method proposed in this paper can also be applied to other brush stroke databases. The texture of the newly produced brush strokes is preserved and can produce various shapes of brush strokes. Fig. 15 and Fig. 16 are the various results of our rendering system with mass brush strokes–database.

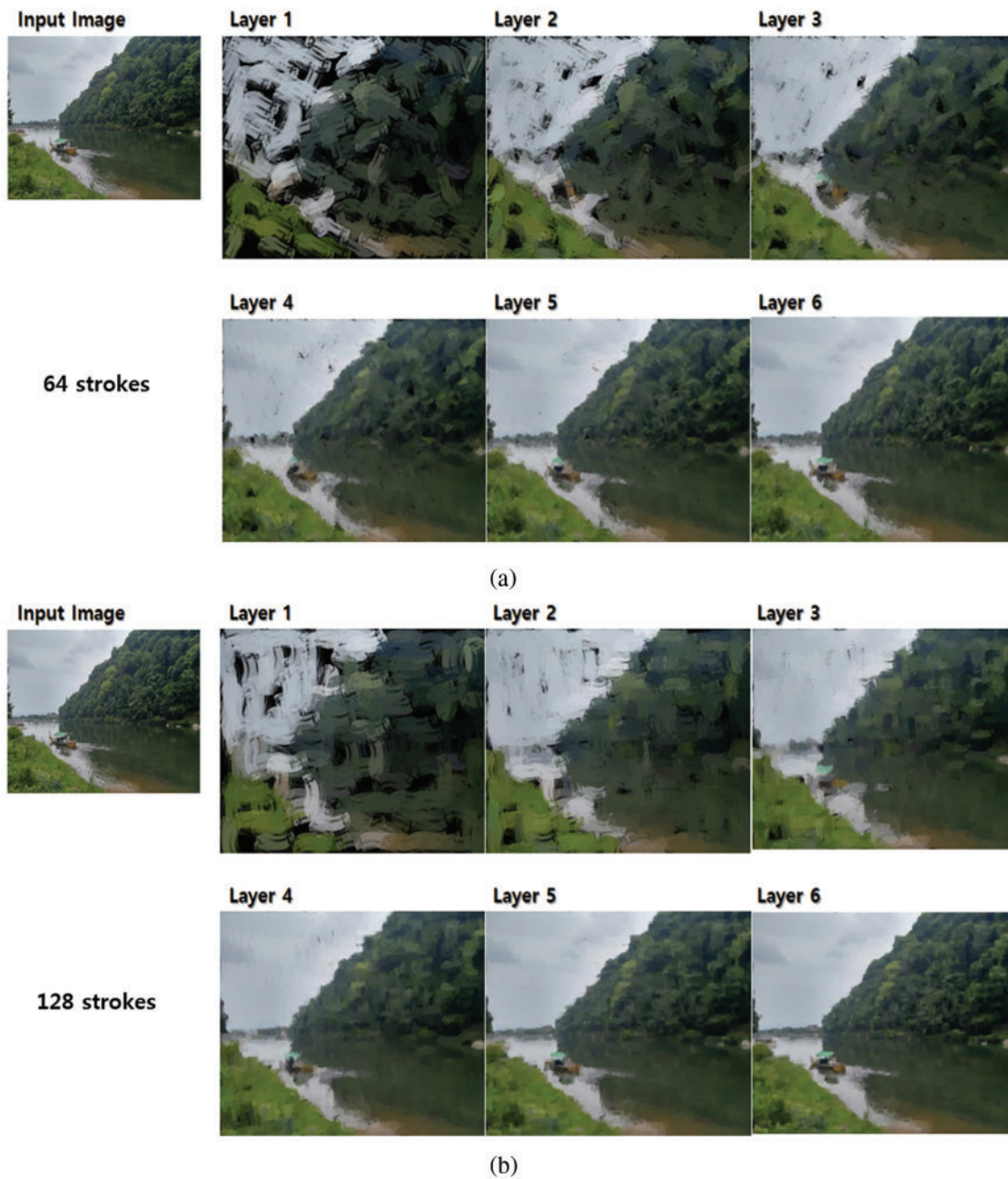
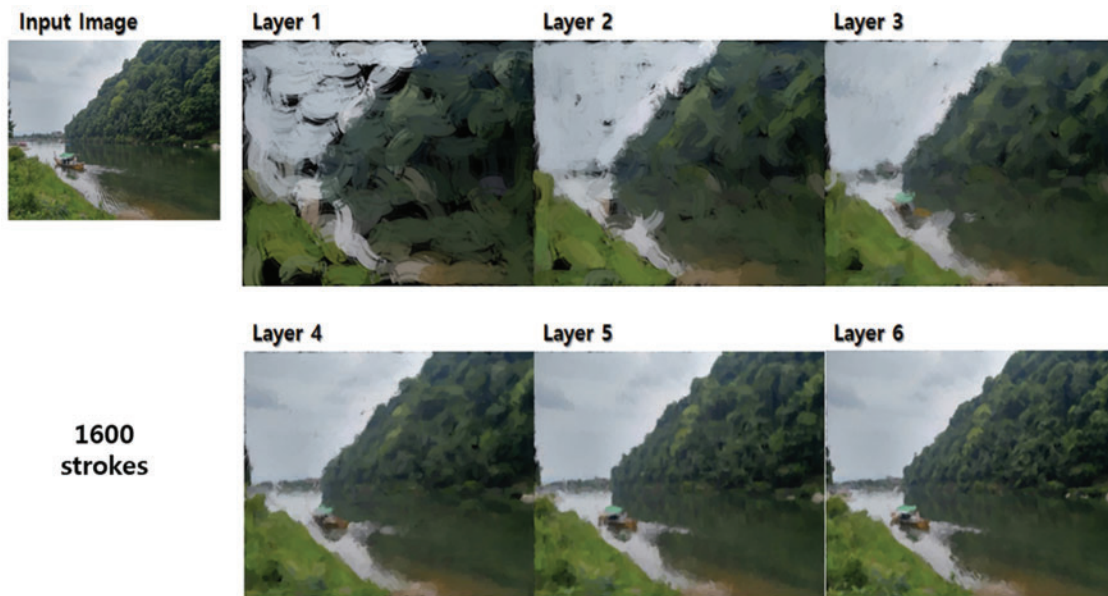
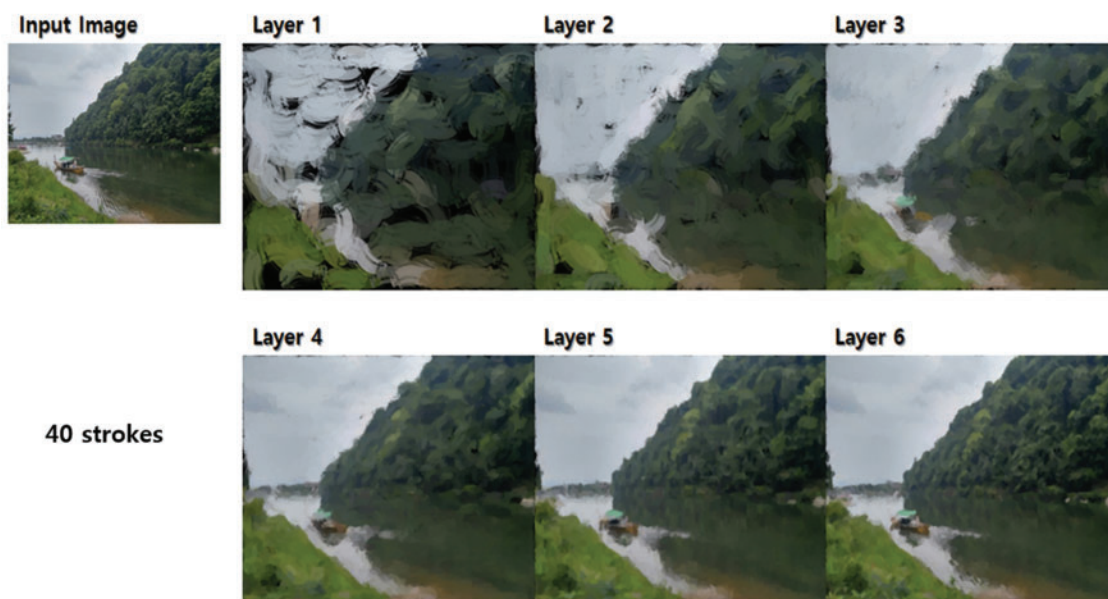


Figure 15: (Continued)



(c)



(d)

**Figure 15:** Comparing rendering results depend on the number of brush strokes (a) rendering results with 64 brush strokes (b) rendering results with 128 brush strokes (c) rendering results with 1600 brush strokes (d) rendering results with selected 40 brush strokes



**Figure 16:** Other results of our system

## 5 Conclusion

Painting works are based on the style that the artist wants. Each artist has a distinctive brush stroke, but it is often necessary to describe the details or modify the brush stroke he or she uses. Real artists can produce results using a variety of brushes at times like this, but in the case of conventional brush stroke rendering, they only work with simple changes such as rotation magnification in their database, resulting in poor quality compared to real works.

The method of generating a new stroke proposed in this paper is faster, simpler, and can be generated with texture preserved than the method of obtaining a stroke by drawing it directly. And since image warping is used, it is possible to produce a variety of shapes in a small number,

unlike stroke synthesis methods. In addition, the brush stroke judgment process through color suitability and shape suitability minimizes the omission of objects. By doing so, many brush stroke mass data produced has been used to produce good results. Furthermore, the parameters used to generate brush strokes were applied to the optimal brush judgment process, resulting in efficient rendering results using only a small number of 1600 brushes.

The contributions of our research can be summarized as follows: First, using simple warping and various parameters, various brushes can be produced. Because there is a limit to collecting brush strokes directly, we propose a technique to expand the database of these collected brush strokes. At this time, a brush stroke was produced in manually and automatically. Secondly, high-quality painterly rendering results were produced through a brush stroke selection algorithm using color and curve similarity. When checking color similarity only, there are parts that are difficult to express, but detailed descriptions are possible considering curve similarity. Finally, through statistical analysis, frequently used ones of the expanded databases could be selected to shorten the generation time of pictorial rendering results. This has the effect of managing mass data.

However, the size of the stroke generated is fixed to the size of the input stroke image, which is inappropriate for generating a wave of strokes when given too many deflections. To this end, we can expect to produce more diverse shape brush strokes considering brush sizes above the input image, and shapes that can be produced by arm movements rather than by considering wrist movements alone.

**Funding Statement:** This research was supported by the Chung-Ang University Research Scholarship Grants in 2017.

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the present study.

## References

1. Ashikhmin, M. (2001). Synthesizing natural textures. *ACM Symposium on Interactive 3D Graphics*, pp. 217–226. USA. DOI 10.1145/364338.364405.
2. Ashikhmin, M. (2003). Fast texture transfer. *IEEE Computer Graphics and Applications*, 23(4), 38–43. DOI 10.1109/MCG.2003.1210863.
3. Bonet, J. S. D. (1997). Multiresolution sampling procedure for analysis and synthesis of texture images. *Proceedings of SIGGRAPH*, vol. 1997, pp. 361–368. USA. DOI 10.1145/258734.258882.
4. Efros, A. A., Freeman, W. T. (2001). Image quilting for texture synthesis and transfer. *Proceedings of SIGGRAPH 2001*, pp. 341–346. USA. DOI 10.1145/383259.383296.
5. Chao, D., Chen, C. L., Kaiming, H., Xiaoou, T. (2015). Image super-resolution using deep convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(2), 295–307. DOI 10.1109/TPAMI.2015.2439281.
6. Chen D., Yuan L., Liao J., Yu N., Hua, G.. (2017). Stylebank: An explicit representation for neural image style transfer. *IEEE Conference on Computer Vision and Pattern Recognition*, USA.
7. Hertzmann, A. (2003). Tutorial: A survey of stroke-based rendering. *IEEE Computer Graphics & Applications*, 23, 70–81. DOI 10.1109/MCG.2003.1210867.
8. Litwinowicz, P. (1997). Processing images and video for an impressionist effect. *Proceeding of SIGGRAPH*, vol. 1997, pp. 407–414. USA. DOI 10.1145/258734.
9. Lee, J. H., Choi, J. I., Seo, S. H. (2020). Emotion-inspired painterly rendering. *IEEE Access*, 8, 104565–104578. DOI 10.1109/Access.6287639.
10. Kang, D. W., Yoon, K. H.. Landscape painterly rendering based on observation. *2017 European Conference on Electrical Engineering and Computer Science*, Switzerland. DOI 10.1109/EECS.2017.66.

11. Seo, S. H., Yoon, K. H. (2009). A study on stroke based rendering using painterly media profile. *Journal of Korea Multimedia Society*, 12(11), 1640–1651.
12. Lee, H., Seo, S. H. and Yoon, K. H. (2009). A study on saliency-based stroke LOD for painterly rendering. *Journal of KIISE: Computer Systems and Theory*, 36(3), 199–209.
13. Herzmann, A. (1998). Painterly rendering with curved brush strokes of multiple sizes. *Proceedings of the 25th annual conference on Computer Graphics and Interactive Techniques*, pp. 453–460.
14. Seo, S. H., Seo, J. W., Yoon, K. H. (2009). A painterly rendering based on stroke profile and database. *Proceeding of International Symposium on Computational Aesthetics in Graphics, Visualization, and Imaging*, pp. 9–16.
15. Seo, S. H., Yoon, K. H. (2013). A multi-level depiction method for painterly rendering based on visual perception cue. *Multimedia Tools and Applications*, 64(2), 277–292. DOI 10.1007/s11042-012-1036-x.
16. Lu, J. W. (2013). Realbrush: Painting with examples of physical media. *ACM Transactions on Graphics*, 32(4). DOI 10.1145/2461912.2461998.
17. Hertzmann, A. (2001). Paint by relaxation. *Proceeding Computer Graphic International*, pp. 47–54. China. DOI 10.1109/CGI.2001.934657.
18. Hertzmann, A. (2002). Fast paint texture. *Proceedings of the 2nd international symposium on Non-Photorealistic Animation and Rendering*, vol. 161, pp. 91–96. France. DOI 10.1145/508530.508546.
19. Haeberli, P. (1990). Paint by numbers: Abstract image representations. *ACM SIGGRAPH Computer Graphics*, 24(2), 207–214. DOI 10.1145/97880.97902.
20. Hays, J., Essa, I. (2004). Image and video based painterly animation. *Symposium of Non-Photorealistic Animation and Rendering*, pp. 120–133. France. DOI 10.1145/987657.987676.
21. Huang, H., Fu, T. N., Li, C. F. (2011). Painterly rendering with content-dependent natural paint strokes. *Visual Computer*, 27(9), 861–871. DOI 10.1007/s00371-011-0596-5.
22. Subramanian, M., Akleman, E. (2020). A painterly rendering approach to create still-life paintings with dynamic lighting. *ACM SIGGRAPH 2020 Posters*. DOI 10.1145/3388770.3407403.
23. Liu, Z., Yang, C., Rho, S., Liu, S., Jiang, F. (2017). Structured entropy of primitive: Big data based stereoscopic image quality assessment. *IET Image Processing*, 11(10), 854–860, DOI 10.1049/iet-ipr.2016.1053.
24. Liu, A., Liu, X., Wei, T., Yang, L. T., Rho, S. (2017). Distributed multi-representative Re-fusion approach for heterogeneous sensing data collection. *ACM Transactions in Embedded Computing Systems*, 16(3), 1–25. DOI 10.1145/2974021.
25. He, J., Huang, L., Tian, R., Li, T., Sun, C. et al. (2018). Massimager: A software for interactive and in-depth analysis of mass spectrometry imaging data. *Analytica Chimica Acta*, 1015(26), 50–57. DOI 10.1016/j.aca.2018.02.030.