CHANGING WEATHER OF IMAGE SEQUENCES TAKEN FROM A VIDEO CAMERA

RYUUICHI KAWASAKI, KAZUFUMI KANEDA, HIDEO YAMASHITA
Faculty of Engineering, Hiroshima University, Japan
{kawa, kin, yama}@eml.hiroshima-u.ac.jp

ABSTRACT

In generating scenes for driving simulators and movies, it is at times desirable to alter the weather conditions of image sequences taken from a video camera. The paper proposes a method for generating a rainy scene from an image sequence taken on a clear day. The proposed method segments an image into three kinds of area: sky, road, and all the other surrounding area, including buildings and trees. Each area is processed in different ways to generate the rainy scene. Blue sky is replaced with a cloudy sky. Road surface tone is modified to that of a wet road. The intensity distribution in the surrounding area is adjusted to that of shadow. A scene taken from a vehicle driver's seat is applied to demonstrate the usefulness of the proposed method.

KEYWORDS

Image manipulation, Segmentation, HLS color space, Filter

1. INTRODUCTION

The real possibility exists for losing a chance to film a rainy scene on location due to the filming schedule. It is therefore useful to generate scenes under a desired weather condition from image sequences taken in different kinds of weather conditions. The goal of our research is to develop a method for altering weather conditions of image sequences taken from a video camera. As an initial step, the paper proposes a method for generating a rainy scene from a scene taken on a clear day.

Comparing images taken on a clear day and a rainy day, the following differences are evident. In a rainy scene, naturally, the sky is cloudy and the road is wet. In addition, the contrast of an image in a rainy scene is lower than that of a clear day. Taking into account such differences, the proposed method segments an image into three kinds of area: sky, road, and the surrounding area, which includes buildings and trees. For each respective area, the following processes are executed.

1. Altering the sky area to a cloudy sky
2. Altering the road area to a wet road
3. Altering the intensity distribution of the surrounding area in HLS color space

Finally, these modified areas are combined to generate a rainy day image.

In this approach, the sky and road areas are extracted from an image. Manual extraction is both time and labor consuming. To address the problem, several approaches for image segmentation have been proposed. Among the most commonly used methods is expanding the area from a seed point. The seed point in an image is specified manually, and the area automatically expands based on similarity to the seed point. This method, however, always requires manual modification, especially in the boundary of the extracted area.

To determine the boundary of the target area, active contour models and Snakes [1, 2, 3, 4] are used. First, an initial contour is specified roughly around the boundary of the target area, and the contour converges to the exact boundary based on energy minimization. These methods can be used to extract relatively smooth contours. However, images taken from a video camera contain complex boundaries such as leaves of trees, and therefore an enormous amount of calculation time is needed to process a sequence of images because the methods are computationally expensive.

Methods called "Image Snapping" [5] and "Intelligent Scissors" [6] snap a point consisting of a contour to the boundary nearest a location indicated by a pointing device. These methods, however, always require user interaction.

Our approach of image segmentation is based on a threshold. The advantages of the approach are a short computation time and semi-automation of the process. Investigating the distributions of RGB and HLS values, the following features are detected in the areas of the sky and road. Intensities in the sky area in RGB and HLS spaces are higher than those of the other areas. The hue of the road area is distributed over a limited range, and the saturation of the road area is lower than the threshold. Using the features of the sky and road areas, the proposed method extracts each

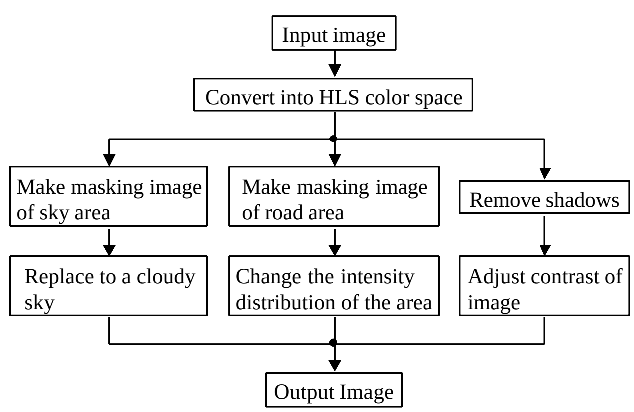


Figure 1 The overview of the proposed method

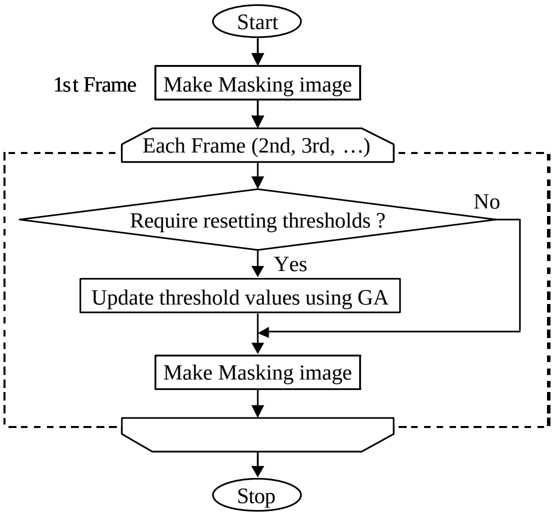


Figure 2 Process flow of making masking images of the sky area from a sequence of images

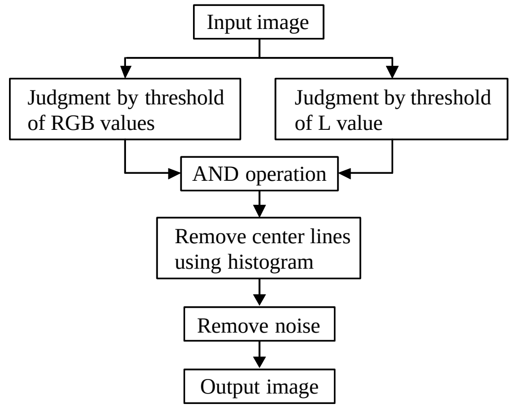


Figure 3 Process flow of making masking images of the first frame

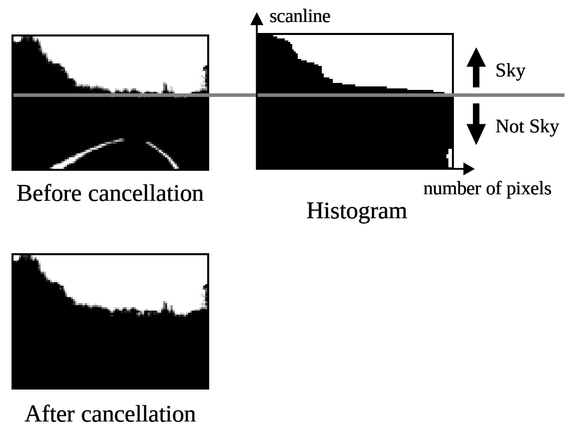


Figure 4 Cancel the sky area using a histogram

area from a sequence of images.

Figure 1 shows an overview of the proposed method. Each image is converted into HLS color space. A masking image of the sky area is generated, and the sky is replaced with a cloudy sky using the mask. In addition to the sky area, a masking image of the road area is generated, and the intensity distribution of the area is modified using a Color Co-occurrence Matrix (CCM) [7]. For the surrounding area, the contrast of the image is adjusted after shadows are removed. All the areas processed by the different methods are then combined into a single image.

In the next section, a method for extracting the sky area from a sequence of images is proposed. Methods for changing the road and the other areas are discussed in Sections 3 and 4, respectively. Conclusions and further studies are indicated in Section 5.

2. PROCESSING THE SKY AREA

2.1 MASKING IMAGES OF THE SKY AREA FOR A SEQUENCE OF IMAGES

A method for making masking images of the sky area from a sequence of images is shown in Fig. 2. First, based on a threshold, a masking image of the first frame is generated. To obtain accurate masking images for the next (second) frame, the necessity of resetting the threshold values is checked. If the result of the examination shows that resetting the thresholds is necessary, the optimum threshold values are calculated using a genetic algorithm [8]. That is, after the second frame, masking images are generated by using both the mask of the previous frame and the updated threshold values.

2.2 MASKING IMAGE OF THE FIRST FRAME

Figure 3 shows the process flow of creating a masking image of the first frame. First, two binary images are generated by threshold RGB and L values. That is, in RGB space, the pixels whose R, G, and B components are greater than thresholds are set to "1," and the other pixels are set to "0." In HLS space, L component is compared to a threshold, and the pixels having larger values than the threshold are set to "1." Then, the two binary images are processed using AND operation to extract the sky area.

The road's white center-line is also extracted, as the color is white and the intensity of the area high. To address the problem, we find the highest scanline having the maximum number of "0" pixels, and the sky area under the scanline is cancelled (see Fig. 4). That is, the scanline eliminates areas except the sky area from the binary image. A histogram plotting the number of "0" pixels for each scanline is generated to find the separator.

The number of pixels for each separated sky area is also counted, and the areas in which the number of pixels is smaller than a threshold are removed as noise. Using the method described above, most noise is removed from the binary images. The remaining noise should be removed with user interaction.

2.3 MASKING IMAGES OF THE FOLLOWING FRAMES

We can make use of the masking image of the first frame to create a masking image of the second frame because of the image coherence of the neighboring frames. As each frame is taken every 1/30th of a second, the boundary of the sky area moves only a few pixels in the next frame. The

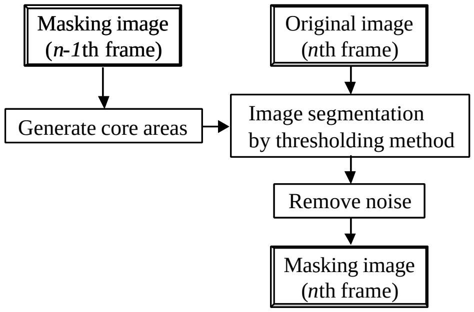


Figure 5 Process flow of the creation of masking images of the following frames

process of the proposed method for creating the following masks is shown in Fig. 5.

First, we generate core areas of both the sky area and the area excluding the sky area. Based on the image coherence, the core areas are extracted from the masking images of the previous frame. That is, culling several pixels around the boundary, the sky area of the previous frame is shrunk. The area except the sky area is also culled in the same way to determine the core area. Then, the areas except both core areas are examined to generate a masking image of the sky area for the next image. That is, the threshold method described in the previous section is applied to all areas except the core areas. The core area of the sky always remains as the sky area of the next frame. Finally, the extracted sky area in which the number of pixels is smaller than a threshold is removed as noise.

Using the core areas decreases the chance of erroneous extraction, and creates the accurate masking images.

2.4 ADJUSTING THRESHOLDS USING A GENETIC ALGORITHM

For the segmentation of the sky area, it is important to set the optimum thresholds, because the result greatly depends on the thresholds. The initial threshold values are not always optimum when the position and/or direction of a video camera are altered, however. It is therefore necessary to adjust the thresholds in scenes greatly changed from the scenes when the initial thresholds are set.

A genetic algorithm [8] is introduced to adjust the thresholds and reset the optimum values when the scene changes. The algorithm is a powerful method to optimize a set of the threshold values from a number of their combinations. This procedure is invoked when the extracted sky area undergoes significant change. Since the difference of the sky areas between the previous frame and the current frame is influenced by the brightness of images taken from a video camera, the optimum threshold values must be changed.

For a genetic algorithm, the question of how to design the genes and the fitness function is crucial. In the proposed method, a gene has a 24 bit-length consisting of 4 sets of threshold values for R, G, B, and L components. Six bits are assigned to each threshold, and are expressed as the offset values from the previous thresholds. That is, the most significant bit is assigned to a sign bit, and the other 5 bits shift the previous threshold at 32 levels in both positive and negative directions.

A fitness function is defined based on two evaluations: specificity and sensitivity. The specificity evaluates the accuracy of the extracted sky area, and the sensitivity literally expresses the sensitivity of the extraction of sky area. These functions are defined as follows.

$$Specificity = N_{Common} / N_{Judge} \tag{1}$$

$$Sensitivity = N_{Common} / N_{Sky} \tag{2}$$

where N_{Judge} is the number of pixels of the sky area determined by the thresholds coded in each gene, N_{Sky} is the number of "1" pixels of the masking image determined by the method described in the previous sections, and N_{Common} is the number of the pixels that belongs to both the core area and the sky area determined by the thresholds coded in each gene.

The fitness function is a weighted sum of the specificity and sensitivity, and is defined by the following equation.

$$Fitness\_func = m \cdot Specificity + Sensitivity \tag{3}$$

where m (m>0) is a weight.

In the genetic algorithm, several dozen genes are generated, and repeating the crossover and mutation processes, the genes are evolved. After few hundred iterations, the gene whose fitness function is highest gives the set of optimum threshold values for the next frame.

2.5 REPLACING THE SKY AREA WITH A CLOUDY SKY

Using the mask images of the sky area generated in the previous section, the clear sky is replaced with a cloudy sky image that has been taken in advance. The process is expressed by the following equation.

$$A_{r,g,b} = (1-t) \cdot O_{r,g,b} + t \cdot C_{r,g,b} \tag{4}$$

where $A_{r,g,b}$ expresses R, G, and B values of the resultant image, $O_{r,g,b}$ and $C_{r,g,b}$ are R, G, and B values of the original image and a cloudy sky, respectively, and t is a pixel value of the masking image.

To compose a cloudy sky smoothly in the original image, the boundary of the sky area is blurred using an averaging filter. That is, an averaging filter with 3×3 width is first applied to the pixels on the boundary of the sky area. Then, to make the boundary smoother, the same filter is applied again to pixels within a 2-pixel distance from the boundary.

Figure 6 shows an original image taken from a video camera on a clear day. A masking image of the sky area generated by the proposed method is illustrated in Fig. 7. Figure 8 shows the cloudy sky image that is used to replace the clear sky.

2.6 EXAMPLES

The proposed method is applied to an image sequence taken from a video camera set at the driver's seat of a vehicle.

The image sequence consists of 300 frames (10 seconds), and the resolution of each image is 720×480 pixels. Figure 9† shows 3 images every 100 frames, and the masking images of the sky area for each image are shown in Fig. 10†. Figure 11† shows the images where the sky areas are replaced with cloudy sky. The images composed of water droplets moving on the windscreen [9] are shown in Fig. 12†. Figure 12 uses the images shown in Fig. 11. To create a masking image of the first frame, initial threshold values shown in table 1 are used. The initial threshold values are determined by investigating the distributions of RGB and HLS values of several images. The proposed method using a genetic algorithm adjusts the threshold values 12 times. The threshold values for each image in Fig. 9 are also shown in table 1, and parameters of the genetic algorithm are shown in table 2. The brightness of the image except the sky area is adjusted to fit the cloudy sky. It took about 28 minutes to process 300 frames using an SGI O2 R12000 (270MHz).

(†: MPEG movies of these animations are available at http://www.eml.hiroshima-u.ac.jp/gallery/Animation/weather/index.html)

Figure 6 Original image

Figure 7 Masking image of the sky area

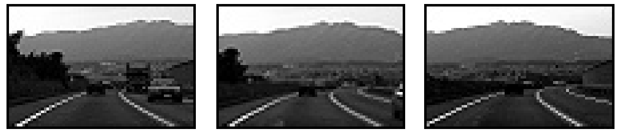
Figure 8 The sky area is replaced to cloudy sky

Table 1 Threshold values

	R	G	B	L
1st frame (initial values)	50	65	80	100
100th frame	57	0	114	162
200th frame	124	0	232	146
300th frame	154	63	69	170

Table 2 Parameters of genetic algorithm

number of generations	100
number of population	50
length of chromosome [bit]	28
crossover rate [%]	40
mutation rate [%]	5
number of elite	2



(a) (b) (c)

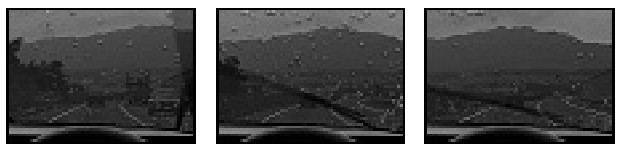
Figure 9 Original images, (a)100th frame, (b) 200th frame, (c) 300th frame

(a) (b) (c)

Figure 10 Masking images of the sky area, (a)100th frame, (b) 200th frame, (c) 300th frame

(a) (b) (c)

Figure 11 The images in which the sky areas are replaced with cloudy sky, (a)100th frame, (b) 200th frame, (c) 300th frame



(a) (b) (c)

Figure 12 The images composed of water droplets, (a)100th frame, (b) 200th frame, (c) 300th frame

3. CHANGING ROAD AREA TO A WET ROAD

The road area is also extracted from the original image based on a threshold in HLS color space. However, the road area is not simply replaced with a wet road, because every road has its own unique surface features. To address the problem, a Color Co-occurrence Matrix (CCM) [7] generated from wet road surfaces is used to change the intensity distribution of that particular road area.

3.1 MASKING ROAD AREA IMAGES

Investigating its distributions of hue, lightness, and saturation components, the road area is found to have the following features.

1. The range of the hue component is from green to blue.
2. The saturation component is lower than that of the other areas.

These features are utilized to generate masking images of the road area.

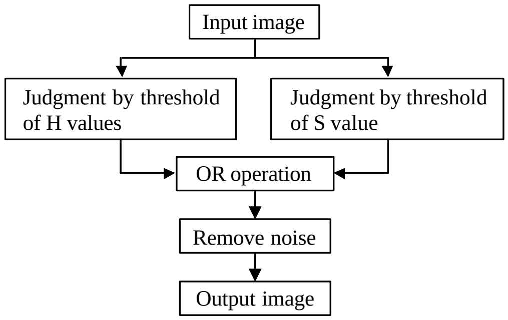


Figure 13 Process flow of creating a masking image of the road area

Figure 13 shows the process flow of creating a masking image of the road area. First, using the features of the road area described above, two binary images are generated. Then, the Boolean set operation OR is applied to the binary images to compensate the conditions of each other. The number of pixels for each separated road area is counted, and the areas in which the number of pixels is smaller than a threshold are removed as noise.

3.2 CHANGING THE INTENSITY DISTRIBUTION

The CCM represents the statistical relationship between neighboring pixels. That is, the elements of the CCM express probabilities that pixels with $\Delta = [\Delta x, \Delta y]$ offset have a pixel value L_r, and are defined by the following four-dimensional matrix.

$$CCM[x, y, L_h, L_r] = \frac{1}{K}\sum_{x=0}^{X}\sum_{y=0}^{Y} d(L[x, y] - L_h) \times d(L[x+\Delta x, y+\Delta y] - L_r) \tag{5}$$

$$K = (X - |\Delta x|)\cdot(Y - |\Delta y|) \tag{6}$$

where $L[x, y]$ is an intensity of pixel (x, y), and d is the Kronecker delta function. K is the normalization factor of the number of pixels within offset Δ. Images with similar CCMs appear similar to each other. The CCM requires a significant amount of memory because of the four-dimensional matrix. To address the problem, only the range containing the intensities of the road area is used to generate a CCM.

A CCM of a wet road is calculated in advance. Using the masking image of the road area, a CCM of the original image is calculated. Intensities of the original images are changed repeatedly to decrease the difference between the CCMs. The difference between two CCMs is defined by the following equation.

$$E = \sum_{\Delta x=-\Delta x_{\max}}^{\Delta x_{\max}} \sum_{\Delta y=-\Delta y_{\max}}^{\Delta y_{\max}} \sum_{L_h=1}^{N}\sum_{L_r=1}^{N} (CCM_d[\Delta x, \Delta y, L_h, L_r] - CCM_w[\Delta x, \Delta y, L_h, L_r])^2 \tag{7}$$

where CCM_d and CCM_w are calculated from the original image and a wet road, respectively. The larger the difference measurement of the pixel, the earlier the intensity of the pixel is changed. The iteration is repeated until the number of modified pixels reaches a threshold ratio exceeding the road area.

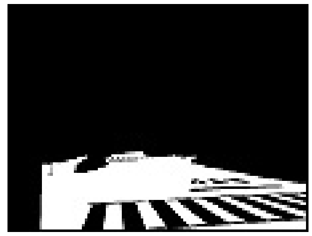


Figure 14 Masking image of the road area

Figure 15 The road area of fig. 6 is changed using a CCM of a wet road

Figure 14 shows a masking image generated by the proposed method from the original image shown in Fig. 6. Using a CCM calculated from a wet road, the original image is modified, and the resultant image is illustrated in Fig. 15. It took about one hour to process the image using an SGI O2 R12000 (270MHz).

4. CHANGING HLS COMPONENTS OF THE SURROUNDING AREA

Comparing images taken on a clear day and a rainy day, the remarkable differences of the surrounding area are shadows and the contrast between bright and dark areas. Therefore, changing the weather in the surrounding area should mainly take into account the following two points.

1. Removing shadows.
2. Decreasing the contrast.

4.1 REMOVING SHADOWS

The proposed method segments the surrounding area of the image into two parts: sunny and shadow areas. First, the boundaries of the shadow areas are detected by a differential filter after removing noise using an averaging filter, as a differential filter is sensitive to noise. An appropriate differential filter with 3×3 width is applied to R, G, and B components of an image, and if more than one output is greater than a threshold, the pixel is detected as a boundary. Any small sized region is then removed as a noise. Except the pixels on the boundary, each region is labeled considering 4 neighboring regions. The boundary is merged with the neighboring regions using the color information. Then, the regions are united if both the means and the variances of RGB and HLS values are very similar. Specifying the correspondence between the sunny and shadow areas, the means and variances of HLS values in both areas are calculated. Using these values, the distributions of HLS values are modified to remove shadow areas.

$$H' = a_h \cdot H + b_h$$

$$L' = a_l \cdot L + b_l \tag{8}$$

$$S' = a_s \cdot S + b_s$$

$$a_{h,l,s} = \sqrt{V_{sunny\_h,l,s} / V_{shadow\_h,l,s}} \quad (9)$$

$$b_{h,l,s} = E_{sunny\_h,l,s} - a_{h,l,s} \cdot E_{shadow\_h,l,s} \quad (10)$$

where (H, L, S) and (H', L', S') are HLS values of pixels in the original and modified images, respectively. $E_{sunny\_h,l,s}$, and $E_{shadow\_h,l,s}$ express the means of H, L, and S values in the sunny and the shadow areas, respectively. $V_{sunny\_h,l,s}$, and $V_{shadow\_h,l,s}$ express the variances of H, L, and S values in the sunny and the shadow areas, respectively. Figure 16 shows the image where a shadow on a building wall is removed by the proposed method.

4.2 DECREASING THE CONTRAST

Extracting shadow areas from images containing complex objects such as trees is difficult, and shadows cast on the complex object itself, such as leaves, do not impart a strong impression of a clear day because of their minute size. The area consisting of complex objects is therefore processed as a group. The average intensity of the area is lowered and the contrast diminished using the following equation.

$$L' = k \cdot (L - L_{ave}) + L'_{ave} \quad (11)$$

where L and L' are intensities of pixels in the original and modified images, respectively. L_{ave} and L'_{ave} are the average intensity of the area in the original and modified images, respectively, and L'_{ave} is specified by the user based on the average intensities of the cloudy sky and the road. Using the method, the average intensity and contrast of the image shown in fig. 16 is modified, where parameters k and L'_{ave} are set to 0.5 and 40, respectively. The resultant image is shown in fig. 17. Figure 18 shows an image in which water droplets are composed for a drive simulator scene [9].

Figure 16 Removing shadows from fig. 15

Figure 17 Decreasing the contrast of fig. 16

Figure 18 Composing water droplets for a drive simulator

5. CONCLUSION

We have developed a method for changing weather conditions of an image taken on a clear day. For image sequences taken from a video camera, we have developed a method for extracting the sky area automatically in cooperation with a genetic algorithm to adjust the threshold values.

There remains further research, however, to reach our goal. Even though the proposed method can reset the threshold values to extract the sky area, the extracted mask is not very precise when scenes change very rapidly due to a bumpy and winding road, for example, because the proposed method is based on the image coherence of neighboring frames.

The proposed method does not take into account movement of the scene when the sky area is replaced with a cloudy sky. However, if the viewing direction is changed, the cloudy sky must be mapped into a different position from the previous frame. To address the problem, the optical flow of the sky area needs to be calculated. It is important for applications dealing with a large number of images to develop a fast, accurate method for calculating the optical flow.

To generate more realistic images on a rainy day, puddles in the road and fog effects in the air caused by rainfall also need to be taken into consideration.

REFERENCES

[1] M. Kass, A. Witkin, and D. Terzopoulos, Snakes: Active Contour Models, *Proceedings of the First International Conference on Computer Vision*, 1987, 259-268.

[2] A. A. Amini, T. E. Weymouth, and R. C. Jain, Using Dynamic Programming for Solving Variational Problems in Vision, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(2), 1990, 855-866.

[3] D. J. Williams and M. Shah, A Fast Algorithm for Active Contours and Curvature Estimation, *CVGIP: Image Understanding*, 55(1), 1992, 14-26.

[4] D. Daneels, et al., Interactive Outlining: An Improved Approach Using Active Contours, *SPIE Proceedings of Storage and Retrieval for Image and Video Databases*, 1908, 1993, 226-233.

[5] Michael Gleicher, Image Snapping, *SIGGRAPH 95 Conference Proceedings*, 1995, 183-190.

[6] Eric N. Mortensen, William A. Barrett, Intelligent Scissors for Image Composition, *SIGGRAPH 95 Conference Proceedings*, 1995, 191-198.

[7] Tom Malzbender and Susan Spach, A Context Sensitive Texture Nib, *Proc. CG International '93*, 1993, 151-161.

[8] Lawrence Davis, *Handbook of genetic algorithms* (Wadsworth Inc., 1990).

[9] Kazufumi Kaneda, Shinya Ikeda, Hideo Yamashita, Animation of Water Droplets Moving Down a Surface, *The Journal of Visualization and Computer Animation*, 10(1), 1999, 15-26.