# Painterly Effects Rendering with Focus Based Level of Detail

Levente Kovács* and Tamás Szirányi*

*Distributed Events Analysis Research Group
Computer and Automation Research Institute, Hungarian Academy of Sciences
Kende u. 13-17, Budapest, H-1111, Hungary
{levente.kovacs,sziranyi}@sztaki.hu

**Abstract**

*We present a novel method for automatically creating scalable and portable 2D non-photorealistically rendered (NPR) images with painterly effects, with level of detail control. The novelty of the approach lies both in using optical image analysis for automatic focus region extraction, and in producing the painterly rendered images as vector graphics. The approach is also novel in that we automatically extract the regions of higher interest without the need of any tools, devices, or a priori image information. The method is a combination of automatic relative focus map estimation and simple digital strokes for the foreground, and color image segmentation for the background. The images are rendered into scalable vector graphics (SVG) for easy viewing and editing with a variety of existing applications and use in other designs/graphics.*

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation

## 1. Introduction

Non-photorealistic rendering means ways of artificially generating images or videos which will have a style resembling a real life style of painting, hand drawing, illustration, cartoon and so on. The topic we are concentrating on is a subclass of non-photorealistic rendering, namely 2D stroke-based painterly rendering. Such methods produce images or videos, either by using a model image as a starting point and producing a painterly rendered image based on the features of this model, or by providing aiding tools for the users packed in painting applications where they will paint the images by hand or in a semi-automatic environment. The goals of such renderings are various: visualizations, painting applications, effect plugins, assisting the drawing of cartoons, aiding medical illustrations, segmentations, indexing, etc.

The most well known classical painterly rendering methods are based on simulated stroke placement. The works of Haeberli [Hae90], Gooch [GCS02], Hertzmann [Her98], Hays [HE04], Shiraishi [SY00] and others are relevant works for different stroke-based painterly methods. Our method presented here differentiates itself from most others in three main areas: a) try to use the fewest possible layers (two in this case), b) automatically separate less and more significant areas (into foreground and background layers), c) produce a portable and easily scalable output. Santella et al. [SD02] have a closely related approach where they give higher detail to more important areas obtained by eye movement tracking. Our method also extracts relevant regions for generating a location dependent painterly rendering, but for this purpose we use an automatic blind deconvolution [Ric72] based relative focus map extraction approach. For extracting the relative focus maps we do not need to know anything about the images, the shooting conditions, or the camera, there is no need for the user to draw area weighting masks. First suggestions for using focus maps for controlling the rendering process were presented in [KS06]. Details on how this approach is better than traditional edge-based aproaches can be found in [KS07]. In this paper we provide means to create painterly effects automatically without user interaction, a priori image information or external devices. This technique should be taken more as an effect generator, which works best on images which have some level of difference in focusness over its areas, e.g. images taken with a relatively low depth of field, although this is not a prerequisite. On flat images, the whole image will get painted with multiscale edge-following strokes (a painting process guided entirely by this concept was presented in [KS04]).

The presented method is a fully automatic process, which needs a model image on the input whose features will be exploited. We introduce an area based level of detail in the rendering which performs better than classical edge based or correlation based approaches. To achieve this goal, we use a relative focus map extraction technique which can automatically obtain areas of the model image that are relatively more important. There are methods to define regions of interest (ROI) in images, but these ROIs are usually related to some predefined object shapes. Our relative focus map definition is more general, independent of shapes or structures. We reduce the painting process to two steps: first produce a background with low level of detail by a segmentation approach, then render a higher detailed foreground that contains the areas with higher relevance. The generated images are stored as vector graphics, for easy viewing, easy resolution change without recoding, and easy reuse in graphics applications.

## 2. Painting

The painting process we use to generate the painterly rendered images has the following main steps:

- Take an input image; this will be the model for the painting.
- Extract those image features that will guide the painting process, i.e. the weighted multiscale edge map of the model, the relative focus map of the model, and the segmentation map of the model.
- Generate the low-detail background layer.
- Render the higher detail foreground layer.

In the following we will elaborate on these steps.

### 2.1. Extracting regions of interest

For automatically extracting the relevant regions of the model images we use a recently introduced, blind deconvolution based approach. We exploit the properties of localized blurring functions obtained with a method based on blind deconvolution. The idea behind this approach is that we can use blind deconvolution to estimate the blurredness/sharpness of local image regions without a priori knowledge (hence the term "blind"), and we can use this information to classify these areas relative to each other based on their respective local blurriness. For classifying these areas we need to estimate to some extent the so called local point spread function – PSF – which is the analogy of the real life distortion function of an optical lens. For details of deconvolution based ROI extraction, results and numerical evaluation see [KS07].

The use of deconvolution for such porposes is a novel approach. Related focus estimation techniques usually use multiple shots with varying focus with the same camera, known a priori camera settings, etc. In our approach we do not need to know anything about the scene, the image or the camera, and there is only one image to work with. Figure 1 contains an example of an input image and its extracted relative focus map which we will use as the ROI region in the following.

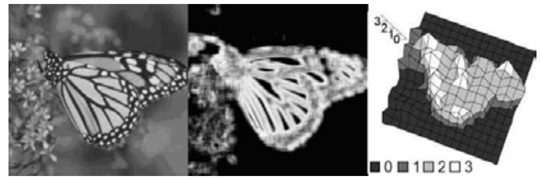


**Figure 1:** *Sample extracted relative focus map.*

### 2.2. Extracting the edge map

The painting process itself is guided by several image features. One of them is an edge map, which is used to determine the orientation of the placed strokes during the foreground painting. For this purpose we seek to obtain the most relevant edge curves from the model image, by using a multiscale weighted edge extraction. In this process the term multiscale refers to the process being a scale-space feature extraction method, based on a method of Lindeberg [Lin98, KS04], which produces weighted edges with higher weight corresponding to more accentuate edges. Edge point candidates will be searched through multiple scaled versions of the same input and edge points will be picked according to their gradient responses through the scales. The weights of the edges in our case will depend on two factors: the length of an edge curve and the weights of the individual points.

After extracting the weighted edges, we also apply focus weighting: on the areas where the focus map shows higher relevance the edge map will get higher weighting. The orientation of edges with high importance is determined by sampling a few neighboring pixels. A sample edge map is shown in Figure 2.

**Figure 2:** *Extracted edge map.*

### 2.3. Background filling

The following step of the presented painterly rendering method is to generate a low-detail backgropund on which the relevant regions will be generatedl. The low-detail background is produced by a reduced resolution color segmentation, for which we use a meanshift segmentation [CM02] based approach. Thus larger homogenous areas can be produced quickly, giving the sensation of a low-detail blurred background. This method proides an easy way for specifying the level of detail of the background. See Figure 3 for a sample generated background layer.

**Figure 3:** *Sample color segmented background layers with different color resolutions (left: model).*

### 2.4. Generating the painted image

Using the above described feature extraction methods, our painterly renderer has the following scheme. First an ordinary image is taken as a model. Then the relative focus map is extracted, which is followed by the edge extraction step, then, the color segmentation map is generated. This is followed by the painting process, which uses the segmentation map as its background. During the painting, small stroke templates are placed on the background on relevant areas, to achieve a higher detailed foreground. The strokes used can be any stroke template with a grayscale map with intensities corresponding to color weights, giving structural information. Here we use simple rectangular $10 \times 3$ pixel size shapes which are described by their position, orientation and color. The stroke placement has the following steps:

1. Pick a stroke position on the focus areas,
2. Get stroke color by majority sampling on the underlying model image area,
3. Take the orientation of the nearest edge curve,
4. Place the stroke.

These steps are iterated until the change caused by repeated stroke placement will reach a certain level. That is, after a number of placed strokes the mean square distance between the areas on the image and the model are obtained and if the difference between this value and the previously calculated one stays below a threshold (i.e. the process stalls), the process is stopped. Three effect generations become possible:

- painted foreground on the original model: the main areas will obtain a painterly look while the rest of the image stays the same.
- model image's main areas on the low-detail background: the image will achieve an accentuated in/out-of focus effect.
- painted foreground on the low-detail background: a full painterly generated image will be produced.

Figures 4 and 5 have sample images with extracted foreground and background layers and final results.

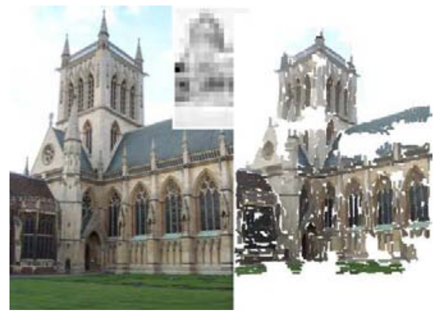

**Figure 4:** *Sample of input image, its focus map and the refined foreground layer.*

**Figure 5:** *Input image (top left), background layer (top right), foreground (bottom left), final (bottom right).*

## 3. Results and applications

The output of the process is a representation of the model with painterly effects. The paintings are converted into scalable vector graphics (SVG) for ease of handling, portability and easy scalability. First, we extract the perimeter points of the background areas and describe them by using the *path* SVG syntax, then we transcribe the stroke parameters of the higher detail layer into SVG commands (*rect* command). This way the rendered images can easily be viewed even in the majority of today's web browsers, edited in a multitude of applications and used in/for other graphics, designs or other artistic creations. Figures 6 and 7 show some generated images.

**Figure 6:** *Top left: SVG-transformed background. Top right: separated color areas. Bottom left: foreground layer. Bottom right: final SVG.*

## 4. Conclusions

We presented a 2D painterly rendering method with full SVG transformation, with novel uses of optical feature extraction and image analysis elements. The rendering method can be used to achieve painterly and cartoon-like effects on ordinary images. The resulting images can be easily used for graphical applications and designs, since the vectorial representation makes such produced images portable, for viewing and editing. The proposed technique and the background generation method can also be used for cartoon-like transformation of videos (by consistent segmentation based background generation and refinement over the motion areas).

**Figure 7:** *Sample result: input image (top), focus mask (middle), output SVG (bottom).*


## 5. Acknowledgements

This work has been supported by grant nr. 049001 of the Hungarian Scientific Research Fund (OTKA).


## References


[CM02] COMANICIU D., MEER P.: Mean shift: A robust approach toward feature space analysis. *IEEE Tr. on Pattern Analysis and Machine Intelligence 24*, 5 (2002), 603–619. 3

[GCS02] GOOCH B., COOMBE G., SHIRLEY P.: Artistic vision: Painterly rendering using computer vision techniques. In *Proc. NPAR* (2002), pp. 83–90. 1

[Hae90] HAEBERLI P.: Paint by numbers: Abstract image representation. *Computer Graphics 24*, 4 (1990), 207–214. 1

[HE04] HAYS J., ESSA I.: Image and video based painterly animation. In *Proc. NPAR* (2004), pp. 113–120. 1

[Her98] HERTZMANN A.: Painterly rendering with curved brush strokes of multiple sizes. In *Proc. SIGGRAPH'98* (1998), pp. 453–460. 1

[KS04] KOVÁCS L., SZIRÁNYI T.: Painterly rendering controlled by multiscale image features. In *Proc. SCCG 2004* (2004), pp. 183–190. 1, 2

[KS06] KOVÁCS L., SZIRÁNYI T.: 2d multilayer painterly rendering with automatic focus extraction. In *Proc. WSCG* (2006), pp. 141–145. 1

[KS07] KOVÁCS L., SZIRÁNYI T.: Focus area extraction by blind deconvolution for defining regions of interest. *IEEE Tr. on Pattern Analysis and Machine Intelligence 29*, 6 (2007), 1080–1085. 1, 2

[Lin98] LINDEBERG T.: Edge detection and ridge detection with automatic scale selection. *Intl. J. of Computer Vision 30*, 2 (1998), 117–154. 2

[Ric72] RICHARDSON W.: Bayesian-based iterative method of image restoration. *JOSA 62* (1972), 55–59. 1

[SD02] SANTELLA A., DECARLO D.: Abstracted painterly rendering using eye tracking data. In *Proc. NPAR* (2002), pp. 75–82. 1

[SY00] SHIRAISHI M., YAMAGUCHI Y.: An algorithm for automatic painterly rendering based on local source image approximation. In *Proc. NPAR* (2000), pp. 53–58. 1