

# IArtist : A self learning computer artist

Benjamin Raynal \*  
Institut Gaspard Monge

Xavier Gouchet  
French Algorists

Venceslas Biri  
Institut Gaspard Monge

Vincent Nozick  
Institut Gaspard Monge



Figure 1: Image before application, Image with style keyword "monet" and with style keyword "picasso"

**Keywords:** Style analysis, self learning, computer artist

## 1 Introduction

Designing a program that is not a tool of artistic creation, but a creator itself have been a real challenge for both digital artists and researchers. The most famous program of artistic creation is AARON [Cohen ], which is in continual development since 1973 by its creator, Harold Cohen. Unfortunately, AARON cannot learn new styles or imagery by its own, each new capability must be hand-coded by Harold Cohen. Roxame [Berger ], another artistic creation program created in 2001 by Pierre Berger, is based on artificial intelligence, and have its own style, emerging from both the artistic preferences of the user, and a stochastic process. This style can evolve and is refined at each work.

These programs can create paintings, but their styles need to be hand-coded by the author, or to be approved by the user. We propose a new method that performs iteratively random process on the image and submits the result to a judging function. If this function detects an amelioration according to the target style, the program continues to perform another process randomly chosen. Else, it removes the last process and try another one.

## 2 Program structure

We divide the creation process into three parts splitted as follows: define a painting subject and a style, represent the style from its Principal Component Analysis and then start the creation process. These three parts are detailed in the following sections.

### The subject and the style

The program requires two inputs : the subject of the painting and the reference style. These inputs can be an image, a set of images, or a set of words. In the latter case, a set of images can be automatically obtained from public database like Flickr or Google Image.

### Style analysis

For each image of the style image set, we build a vector containing valuations of image analysis criteria. We use an extensive list of different criteria such as:

- color histograms and gradient color histograms, from different color models (RGB, YCrCb, HSV...)
- connected components size and shape.
- significant Discrete Cosinus Transform coefficients.
- quantification of lines and circles, from Hough Transforms.
- normalized cross correlation between the input image and its flipped image (symmetry properties).

The program will perform a Principal Component Analysis of the vector set and then define the Reference Style Subspace (RSS) among the criteria that represents the best the style. The minimal zone containing the projection of these style vectors represents the target zone to reach to determine whether an image corresponds to that style or not.

### Creation process

The creation process transforms a subject image to reach the target style. Our program will iteratively perform a random elementary process on the subject image like color operations, smoothing, morphological operations, edge sharpening, image fusion, pixelization, etc. For each transformation, a valuation vector is build from the candidate image and is projected on the RSS. The transformed image is validated if the last projection is closer to the target zone compared to the previous result. If the projected vector reaches the target zone, the creation process is stopped and the current image is returned as the final result.

## 3 Results and discussion

Figure 1 depicts a sample result of our method. The left image is transformed to fit to "Monet" style on the center image and to "Picasso" style on the right image. These two styles have been both set up from a set of 30 images. We can notice that this method is subject to local minima convergence. This problem is easily solved with an "undo history stack". We also set up our program to accept up to two non-successful transformations.

## References

BERGER, P. Roxame. [www.roxame.com](http://www.roxame.com).

COHEN, H. Aaron. [www.kurzweilcyberart.com/aaron/history.html](http://www.kurzweilcyberart.com/aaron/history.html).

\*e-mail: raynal@univ-mlv.fr