



## Présentation du sujet

Pour le projet du module de traitement d'image, j'ai choisit d'implémenter une méthode de rendu non-photoréaliste.

Aaron Hertzmann propose dans son article, « Painterly Rendering with Curved Brush Strokes of Multiples Sizes », une méthode pour créer des images ayant l'apparence de peintures à partir de photographies. Cette méthode permet de simuler les coups de pinceaux d'un peintre en fonction de différents paramètres qui seront exposés plus loin. Ces paramètres permettent de donner un style de peinture précis à une image.

## I - Résumé de l'article

### 1 – Méthode principale

```
function paint(sourceImage,  $R_1$  ...  $R_n$ )
{
    canvas := a new constant color image

    // paint the canvas
    for each brush radius  $R_i$ ,
        from largest to smallest do
    {
        // apply Gaussian blur
        referenceImage = sourceImage *  $G_{(fg R_i)}$ 
        // paint a layer
        paintLayer(canvas, referenceImage,  $R_i$ )
    }

    return canvas
}
```

Cet algorithme va simuler les méthodes utiliser par les peintres pour générer une image ayant l'aspect d'une peinture. L'image sera organisé en un ensemble de couche qui seront peintes successivement. Chaque couches sera associées à une taille de pinceau constante. La couche ayant la plus grande taille de pinceau est peinte en première, puis ensuite nous affinons les détails de l'image en peignant les couches ayant des tailles de pinceau plus petites.



Chaque couches sera peinte à partir d'une version lissée de l'image originale. Nous avons gardés le choix du filtre de Gauss comme méthode de lissage de l'image. A.Hertzmann précise que la diffusion non linéaire peut être utilisé à la place du filtre de Gauss pour obtenir de meilleur résultat.

*Nous utilisons ces paramètres pour le lissage de l' image avec le filtre de Gauss :*

*Déviation standard :  $\sigma = f_{\sigma} \cdot R_i$ , avec  $R_i$  le rayon du pinceau et  $f_{\sigma}$  un facteur constant*

*Taille du noyau :  $(6 \cdot \sigma + 1) \times (6 \cdot \sigma + 1)$*

Une couche est formée par un ensemble de coups de pinceaux et chaque coups de pinceaux est représentés par un ensemble de points de contrôles formant une B-Spline. Le calcul des points de contrôles se fait à partir du gradient de la luminance de l'image lissée.

*La luminance d' un pixel est calculée avec la fonction suivante :*

$$L(r, g, b) = 0.30 * r + 0.59 * g + 0.11 * b$$

*Les orientations x et y du gradient sont calculées en effectuant la convolution de l' image I avec les noyaux de Sobel :*

$$\begin{array}{ccc} -1 & 0 & 1 \\ G_x = -2 & 0 & 2 * A \end{array} \text{ et } \begin{array}{ccc} -1 & -2 & -1 \\ G_y = 0 & 0 & 0 * A \\ -1 & 0 & 1 \end{array}$$

*On approxime l' amplitude G du gradient en calculant :*

$$G = 0.5 * |G_x| + 0.5 * |G_y|$$









