Artistic Stylization of Images and Video

Eurographics 2011

John Collomosse and Jan Eric Kyprianidis

Centre for Vision Speech and Signal Processing (CVSSP)
University of Surrey, United Kingdom

Hasso-Plattner-Institut, University of Potsdam, Germany

http://kahlan.eps.surrey.ac.uk/EG2011
Artistic Stylization Resources

- **Texts**
  - Strothotte & Schlechtweg
    - ISBN: 1558607870
  - Gooch & Gooch
    - ISBN: 1568811330
  - Romero & Machado
    - ISBN: 3540728767

- **Tutorials**
  - SIGGRAPH 99 (Green et al.) – 2D/3D NPR
  - SIGGRAPH 02 (Hertzmann) – 2D NPR
  - SIGGRAPH 03 (Sousa et al.) – 2D/3D NPR
  - Eurographics 05,06 and...
  - SIGGRAPH 06 (Sousa et al) – 3D NPR
  - SIGGRAPH 10 (McGuire) – 3D NPR for Games

- **Web Bibliographies**
  - [http://video3d.ims.tuwien.ac.at/~stathis/npr\lib/index.php](http://video3d.ims.tuwien.ac.at/~stathis/npr\lib/index.php)
  - [http://isgwww.cs.uni-magdeburg.de/~stefans/npr/nprpapers.html](http://isgwww.cs.uni-magdeburg.de/~stefans/npr/nprpapers.html)

- **Main Publication Forums**
  - **NPAR** (Symposium on Non-photorealistic Animation)
    - Held in Annecy even years, at SIGGRAPH odd years.
  - IEEE Trans Visualization and Comp. Graphics (**TVCG**)
  - IEEE Computer Graphics and Applications (**CG&A**)
  - Eurographics and **Computer Graphics Forum**
  - SIGGRAPH, SIGGRAPH Asia and **ACM ToG**
  - EG Symposium on Rendering (**EGSR**)
  - ACM/EG Symposium on Computer Animation (**EGSA**)
Non-Photorealistic Rendering (NPR)
Coined by Salesin et al., 1994

Stylized Rendering  Aesthetic Rendering  Artistic Stylization  Artistic Rendering

Anatomy of the Human Body
H. Gray, 1918
Motivation

Artistic Stylization

- Why?
  - Comprehension
  - Communication
  - Visualization
  - Aesthetics
  - Animation

Artistic Stylization can

- Simplify and structure the presentation of content
- Selectively guide attention to salient areas of content and influence perception
- Learn and emulate artistic styles
- Provide assistive tools to artists and animators (not replace the artist!)
- Help us to design effective visual interfaces
**Artistic Stylization**

- Rendering real images/video footage into pseudo-artistic styles
- Convergence of Computer Vision, Graphics (and HCI)

**Visual analysis enables new graphics.**  
**Graphical needs motivate new vision.**
Interactions with Vision

Rendering process is guided by...

- **User conscious interaction**
- Low-level image processing
- Higher level computer vision
- Direct Anisotropic filtering

**1990**
- Semi-automatic painting systems
  - P. Haeberli (SIGGRAPH 90)

**1997 1998 2000**
- Fully automatic painting
  - A. Hertzmann (SIGGRAPH 98)
  - J. Treveatt/Chen [EGUK 97]
  - P. Litwinowicz [SIGGRAPH 97]

**2002**
- Automatic perceptual
  - J. Collomosse [EvoMUSART 05]

**2005 2006**
- Anisotropy / filters
  - H. Winnemoeller [SIGGRAPH 06]
  - J. Kyprianidis [TPCG 08]

**2010**
- NPAR 2010 Grand challenges
  - T. Isenberg [NPAR 06]

**User evaluation**
  - T. Isenberg [NPAR 06]
Tutorial Structure

Part I: Classical algorithms (30 min)
- P. Haeberli (SIGGRAPH 90)
- A. Hertzmann (SIGGRAPH 98)
- J. Collomosse [EvoMUSART 05]

Part II: Vision for Stylisation (60 min)
- D. Decarlo [SIGGRAPH 02]
- J. Kyprianidis [TPCG 08]
- T. Isenberg [NPAR 06]

Part III: Anisotropy and Filtering (70 min)
- H. Winnemoeller [SIGGRAPH 06]
- J. Wang [SIGGRAPH 04]
- J. Collomosse [TVCG 05]

Part IV: Future Challenges (20 min)
- T. Isenberg [NPAR 06]

---

Rendering process is guided by...
- emulation
- User conscious interaction
- Low-level image processing
- Higher level computer vision
- Direct Anisotropic filtering


Late 1980s
- Advances in media emulation
  - D. Strassman (SIGGRAPH 86)
- Video painting
  - P. Litwinowicz (SIGGRAPH 97)
- Fully automatic painting
  - A. Hertzmann (SIGGRAPH 98)
  - Treveatt/Chen [EGUK 97]
  - P. Litwinowicz [SIGGRAPH 97]
- Space-time video
  - J. Wang [SIGGRAPH 04]
  - J. Collomosse [TVCG 05]

User evaluation
- T. Isenberg [NPAR 06]

User conscious interaction
- User subconscious interaction

BREAK!
Artistic Stylization of Images and Video
Part I – Classical Algorithms / Stroke Based Rendering
Eurographics 2011

John Collomosse

Centre for Vision Speech and Signal Processing (CVSSP),
University of Surrey, United Kingdom
- **Paint by numbers: Abstract image representations**  
  P. Haeberli, SIGGRAPH 1990

- **Almost Automatic Computer Painting**  
  P. Haggerty, IEEE CG & A 1991

- **Orientable Textures for Image based Pen-and-Ink Illustration**  
  D. Salisbury et al., SIGGRAPH 1997

- **Processing images and video for an impressionist effect**  
  P. Litwinowicz, SIGGRAPH 1997

- **Statistical techniques for the automated synthesis of non-photorealistic images**  

- **Automatic Painting based on Local Source Image Approximation**  
  Shiraishi and Yamaguchi, NPAR 2000.

- **Painterly Rendering with Curved Strokes of Multiple Sizes**  

- **Paint by Relaxation**  
  A. Hertzmann, CGI 2001

- **Fast Paint Texture**  
  A. Hertzmann, NPAR 2002
• Early painting systems lacked appropriate UI for rich digital painting

Xerox superpaint (1980s)

Windows Vista Paint 2007
Paint by numbers: Abstract Image Representations
Haeberli. (1990)

- Stroke based rendering (SBR)
- Painting is a manually ordered list of strokes, placed interactively.
- Stroke attributes sampled from the photo.
- Stroke colour and orientation are sampled from the source image
- Stroke order and scale are user-selected
- Addition of RGB noise generates an impressionist effect

Paintings with / without orientable strokes

\[ E(I) = \left[ \left( \frac{\partial I}{\partial x} \right)^2 + \left( \frac{\partial I}{\partial y} \right)^2 \right]^{\frac{1}{2}} \]

\[ \theta(I) = \arctan \left( \frac{\partial I}{\partial y} / \frac{\partial I}{\partial x} \right) \]

Sobel Edge detection

Edge Mag.

Edge orient.

Orientation

Photo credit: Haeberli ‘90.
More stylised orientation effects with a manually defined orientation field
Paint by numbers: Abstract Image Representations

Haeberli. (1990)

All tutorial code at http://kahlan.eps.surrey.ac.uk/EG2011
- Very similar system for pen-and-ink rendering of photos
- User defined orientation field.
  - Regions manually drawn and marked up with orientation
- Stroke (line) placement automatic. Strokes clipped to keep within regions.
- Stroke *colour* and *orientation* are sampled from the source image
- Stroke *order* and *scale* are user-selected
- Scale sampled from *Sobel edge magnitude*
- Regularly place strokes. Order of strokes *randomly generated*

Photo credit: Haeberli ’90.

Loss of detail in important regions

- Fully automated
Stroke grows from seed point bidirectionaly until edge pixels encountered.
Common recipe for SBR in the 1990s

- Sobel edge detection on blurred image
- Regular seeding of strokes on canvas
- Scale strokes inverse to edge magnitude
- Orient strokes along edge tangent
- Place strokes in a specific way using this data

An interesting alternative uses 2\textsuperscript{nd} order moments within local window to orient strokes.

- Extended to multi-scale strokes by Shiraishi and Yamaguchi (NPAR 2000)
Automatically Painting based on Local Source Image Approximation
Shiraishi and Yamaguchi (2000)

- 2D zero-moments for greyscale image \( I(x,y) \)

\[
M_{lm} = \sum \sum x^l y^m I(x,y).
\]
- The canvas is built up in layers from coarse to fine
  - Analysis window scale, and stroke scale are varied in proportion
- Artists do not paint with uniformly shaped short strokes (pointillism excepted!)
- Two key contributions (1998)
  - Multi-layer (coarse to fine) painting
  - Painting using $\beta$-spline strokes
- Spline strokes can be bump mapped for an improved painterly look (NPAR 2002)
- Greedy algorithm for stroke placement
- Regularly sample the canvas to seed strokes
- Build a list of control points for each stroke by “hopping” between pixels*

1) Pick a direction arbitrarily
   (some implementations explore both)

* In practice, best to use float coordinates and interpolate edge orientation
- Greedy algorithm for stroke placement
- Regularly sample the canvas to seed strokes
- Build a list of control points for each stroke by “hopping” between pixels*

2) Make another hop, resolving directional ambiguity by hopping in the direction of min $\theta$

* In practice, best to use float coordinates and interpolate edge orientation
- **Greedy algorithm** for stroke placement
- Regularly sample the canvas to seed strokes
- Build a list of control points for each stroke by “hopping” between pixels*

3) Keep hopping until end land on a pixel whose RGB colour differs (> threshold) from mean colour of stroke, or the stroke length is > a second threshold.

* In practice, best to use float coordinates and interpolate edge orientation

B-spline control points

\[ \theta_1 \]

\[ \theta_2 \]

Until termination criteria met
- Painting is laid down in multiple layers (coarse to fine)
- Band-pass pyramid (= differenced layers of low-pass)
- Strokes from early layers are visible in final layer

- Paint coarsest layer with large strokes
- Paint next layer with smaller strokes
  - Only paint regions that differ between the layers
  - Use RGB difference

Hertzmann (1998)
Tips and tricks

- Non-linear diffusion* instead of Gaussian blur sharpens the painting – preserves edges and accuracy of edge orientation.
- Build Gaussian pyramid at octave intervals, $\sigma=(1,2,4,8)$. 4 layers sufficient.
- Stroke thickness also at octave intervals
- Low-pass filter the hop direction $\theta$

Global Optimization to Iteratively Produce “Better” Paintings

- Hertzmann 1998 (Greedy stroke placement)
- Hertzmann 2001 (Global stroke optimization)
How to define the optimality of a painting ‘P’ derived from a photo ‘G’

\[ E(P) = E_{app}(P) + E_{area}(P) + E_{nstr}(P) + E_{cov}(P) \]

\[ E_{app}(P) = \sum_{(x,y) \in I} w_{app}(x,y)||P(x,y) - G(x,y)|| \]

\[ E_{area}(P) = w_{area} \sum_{S \in P} \text{Area}(S) \]

\[ E_{nstr}(P) = w_{nstr} \cdot (\text{number of strokes in } P) \]

\[ E_{cov}(P) = w_{cov} \cdot (\text{number of empty pixels in } P) \]

- Weighting \( w_{app} \) is derived from a Sobel edge magnitude (or user defined)
- The right strokes in the right place will minimize the energy function \( E(P) \)
Active Contours (Snakes)
Kass et al. (1987)

\[
E = \int_{s=0}^{s=1} \left( \alpha \frac{dP(s)}{ds} + \beta \frac{dP^2(s)}{ds^2} + \gamma I(P(s)) \right)
\]

2n-D Solution space

2-D Image

\(x_n, y_n\)
\((x_1, y_1)\)
\((x_2, y_2)\)
\((x_3, y_3)\)
\(\text{etc...}\)
Strokes selected at random and modified by local optimization to minimize $E(P)$

Strokes modelled as active contours (“snakes”)

- … but energy has no $1^{\text{st}}/2^{\text{nd}}$ order derivative terms
- $E(P)$ is approximated under control points

$$E(P) = E_{app}(P) + E_{area}(P) + E_{nstr}(P) + E_{cov}(P)$$

$$E_{app}(P) = \sum_{(x,y) \in I} w_{app}(x,y) ||P(x,y) - G(x,y)||$$

$$E_{area}(P) = w_{area} \sum_{S \in P} \text{Area}(S)$$

$$E_{nstr}(P) = w_{nstr} \cdot (\text{number of strokes in } P)$$

$$E_{cov}(P) = w_{cov} \cdot (\text{number of empty pixels in } P)$$
- Simplest solution (gradient descent)
  - Can be unstable for this weighted heuristic function

\[
\begin{align*}
\frac{\delta E(x_1)}{\delta x_1} \\
\frac{\delta E(y_1)}{\delta y_1} \\
\frac{\delta E(x_2)}{\delta x_2} \\
\frac{\delta E(y_2)}{\delta y_2} \\
\frac{\delta E(x_3)}{\delta x_3} \\
\frac{\delta E(y_3)}{\delta y_3} \\
\frac{\delta E(x_4)}{\delta x_4} \\
\frac{\delta E(y_4)}{\delta y_4} \\
\vdots \\
\frac{\delta E(x_n)}{\delta x_n} \\
\frac{\delta E(y_n)}{\delta y_n}
\end{align*}
\]
Dynamic programming solution (Amini et al. ‘90)

• Move each control point to obtain locally optimal position (5x5)
• $E(P)$ at control point dependent only on current $v_i$ and previous $v_{i-1}$

\[
\begin{align*}
    s_0(v_1) &= \min_{v_0} e_0(v_0) + e_0(v_1) + e_1(v_0, v_1) \\
    s_1(v_2) &= \min_{v_1} s_0(v_1) + e_0(v_2) + e_1(v_1, v_2) \\
    \vdots \\
    s_{i-1}(v_i) &= \min_{v_{i-1}} s_{i-2}(v_{i-1}) + e_0(v_i) + e_1(v_{i-1}, v_i)
\end{align*}
\]
- Sobel magnitude can be replaced with a manually sketched mask to alter emphasis.
Quick Start: OpenGL research code for bump-mapped paint strokes

- Strokes as Catmull-Rom (interpolating) splines
- Bump mapping via Multi-texturing (can be disabled)
- Dependency on OpenCV to load images (can substitute this trivially)
- Code used in “Empathic Painting” Collomosse et al. NPAR 2006

http://kahlan.eps.surrey.ac.uk/EG2011