Image Vectorization
What is Vectorization?
Image Vectorization

• Goal
  – Convert a raster image into a vector graphics
  – Vector graphics include
    • points
    • lines
    • curves
    • paths
    • polygons
    • regions
    • …
Why Vector Graphics

• Compact
• Scalable
• Editable
• Easy to animate
Compact

input raster image
37.5KB

optimized gradient mesh
7.7KB
Why Vector Graphics

- Compact
- **Scalable**
- Editable
- Easy to animate
Scalable

original image  vector form  bicubic interpolation
Why Vector Graphics

• Compact
• Scalable
• **Editable**
• Easy to animate
Editable
Why Vector Graphics

- Compact
- Scalable
- Editable
- Easy to animate
Easy to Animate
Related Work

• Cartoon drawing vectorization
  – Skeletonization, tracing, and approximation
• Triangulation-based method
• Object-based vectorization
  – Bezier patch
  – Subdivision
Image Vectorization using Optimized Gradient Meshes

Jian Sun, Lin Liang, Fang Wen, Heung-Yeung Shum
Siggraph 2007
Surface Representation

• A tensor-product patch is defined as

\[ m(u, v) = F(u)QF^T(v) \]

• Bezier bicubic, rational biquadratic, B-splines…
  – Control points lying outside the surface
Ferguson Patch

\[ m(u, v) = U C Q C^T V^T \]

\[
Q = \begin{bmatrix}
      m^0 & m^2 & m^0_v & m^2_v \\
      m^1 & m^3 & m^1_v & m^3_v \\
      m^0_u & m^2_u & m^0_{uv} & m^2_{uv} \\
      m^1_u & m^3_u & m^1_{uv} & m^3_{uv}
\end{bmatrix}
\]

\[
C = \begin{bmatrix}
      1 & 0 & 0 & 0 \\
      0 & 0 & 1 & 0 \\
      -3 & 3 & -2 & -1 \\
      2 & -2 & 1 & 1
\end{bmatrix}
\]

\[
U = \begin{bmatrix}
      1 & u & u^2 & u^3
\end{bmatrix} \quad V = \begin{bmatrix}
      1 & v & v^2 & v^3
\end{bmatrix}
\]
Gradient Mesh

- Control point attributes:
  - 2D position
  - Geometry derivatives
  - RGB color
  - Color derivatives

\[ f(u, v) = UCQ^f C^T V^T \]
Flow Chart

Original  Initial Mesh  Optimized Mesh  Final Rendering
Mesh Initialization
Mesh Initialization

- Decompose image into sub-objects
- Divide the boundary into four segments
- Fitting segments by cubic Bezier splines
- Refine the mesh-lines
  - Evenly distributed
  - Interactive placement
Mesh Optimization

Input image

Initial rendering

Final rendering
Mesh Optimization

- To minimize the energy function

\[ E(M) = \sum_{p=1}^{P} \sum_{u,v} \left\| I_p(m(u,v)) - f_p(u,v) \right\|^2 \]

P: number of patches
Levenberg-Marquardt Algorithm

• Most widely used algorithm for Nonlinear Least Squares Minimization.
• First proposed by Levenberg, then improved by Marquardt
• A blend of Gradient descent and Gauss-Newton iteration
Smoothness
Smoothness Constraint

• Add a smoothness term into the energy

\[
E'(M) = E(M) + \lambda \sum_{p=1}^{P} \sum_{s,t} \{ \| m(s - \Delta s, t) - 2m(s, t) + m(s + \Delta s, t) \|^2 + \| m(s, t - \Delta t) - 2m(s, t) + m(s, t + \Delta t) \|^2 \}
\]

which minimizes the second-order finite difference.
Optimized Gradient Mesh guided by Vector Line/Fields
Optimized Gradient Meshes guided by Vector Line/Field

- Given vector fields $V_u$ and $V_v$, we optimize

$$E''(M) = E'(M) + \beta \sum_{p=1}^{P} \sum_{u,v} \{ w_u(m(u,v)) \left( \frac{\partial m(u,v)}{\partial u} , \perp V_u(m(u,v)) \right) \}^2$$

$$+ w_v(m(u,v)) \left( \frac{\partial m(u,v)}{\partial v} , \perp V_v(m(u,v)) \right) \}^2$$
More Results