Matting

Anat Levin, MIT CSAIL

With some slides from Alexei Efros & Fredo Durand

How does Superman fly?

Super-human powers?
OR
Image Matting?

Motivation: compositing

Combining multiple images.
Typically, paste a foreground object onto a new background

Page layout, magazine covers

From the Art & Nature of Digital Compositing
Photo editing
- Edit the background independently from foreground

Matting and compositing

Alpha
- $\alpha$: 1 means opaque, 0 means transparent
- 32-bit images: R, G, B, $\alpha$

The matting equations
$$I_i = \alpha_i F_i + (1 - \alpha_i) B_i$$

Replace background
$$I_i = \alpha_i F_i + (1 - \alpha_i) B_i$$
Review: alpha channel

Add one more channel:
- Image(R,G,B,alpha) → Sprite!

Encodes transparency (or pixel coverage):
- Alpha = 1: opaque object (complete coverage)
- Alpha = 0: transparent object (no coverage)
- 0<Alpha<1: semi-transparent (partial coverage)

Example: alpha = 0.7

Partial coverage or semi-transparency

Why is matting hard?

Matting is ill posed: 7 unknowns but 3 constraints per pixel

Photoshop layer masks

How many equations? How many unknowns?

\[
I_i = \alpha_i F_i^r + (1 - \alpha_i) B_i^r
\]
\[
I_i^b = \alpha_i F_i^b + (1 - \alpha_i) B_i^b
\]
\[
I_i^g = \alpha_i F_i^g + (1 - \alpha_i) B_i^g
\]
\[
I_i^a = \alpha_i F_i^a + (1 - \alpha_i) B_i^a
\]
“Pulling a Matte”

Problem Definition:
• The separation of an image I into
  – A foreground object image F,
  – a background image B,
  – and an alpha matte \( \alpha \)
• F and \( \alpha \) can then be used to composite the foreground object into a different image

Hard problem
• Even if alpha is binary, this is hard to do automatically (background subtraction problem)
• For movies/TV, manual segmentation of each frame is infeasible
• Need to make a simplifying assumption...

Blue Screen matting
Most common form of matting in TV studios & movies

Petros Vlahos invented blue screen matting in the 50s. His Ultimatte® is still the most popular equipment. He won an Oscar for lifetime achievement.

A form of background subtraction:
• Need a known background
• Compute alpha as SSD(C,B)_threshold
• Or use Vlahos' formula: \( \alpha = \frac{1}{1+\phi(C_g,C_b)} \)
• Hope that foreground object doesn’t look like background
  – no blue ties!
• Why blue?
• Why uniform?

Blue screen for superman?

The Ultimatte

Solution #1: No Blue!

The matting eq: \( I_i = \alpha F_i + (1 - \alpha) B_i \)
Background is known: \( B^g = 0, B^c = 0, B^a = 1 \)
Assumption: \( F^a = 0 \)
Now only 3 unknowns!
\[
I^a_i = \alpha F^a_i + (1 - \alpha) B^a_i \Rightarrow \text{get } \alpha \\
I^g_i = \alpha F^g_i + (1 - \alpha) B^g_i \Rightarrow \text{get } F^a, F^g
\]
Instead of reducing the number of unknowns, we could attempt to increase the number of equations. One way to do this is to photograph the object of interest in front of 2 known and distinct backgrounds.

How many equations? How many unknowns? Does the background need to be constant color?

The Algorithm

For every pixel: \((B^R, B^G, B^B), (B^R, B^G, B^B)\) – known

\[
\begin{align*}
1^k &= \alpha F^k + (1-\alpha)B^A \\
1^R &= \alpha F^R + (1-\alpha)B^G \\
1^B &= \alpha F^B + (1-\alpha)B^R \\
1^G &= \alpha F^G + (1-\alpha)B^R \\
1^B &= \alpha F^B + (1-\alpha)B^G \\
1^A &= \alpha F^A + (1-\alpha)B^R \\
\end{align*}
\]

Solve a system of 6 equations in 4 unknowns.

From Smith & Blinn’s SIGGRAPH 96 paper.
Natural image matting

The rules:
Only 1 input image is given (e.g. downloaded from the web), we have no control over the background
User can help, but want to minimize user work

User interfaces

The trimap interface:
- Bayesian Matting (Chuang et al., CVPR01)
- Poisson Matting (Sun et al. SIGGRAPH 04)
- Random Walk (Grady et al. 05)

Scribbles interface:
- Wang&Cohen ICCV05
- Levin et al. CVPR06

Trimap based algorithms
Assumptions: the trimap is narrow.
Thus we could guess F,B values in the mixed region by copying colors from neighboring F,B pixels
Given F,B solve for $\alpha$
Use $\alpha$ to refine F,B estimate
Use F,B estimate to refine $\alpha$ estimate
and so on

Problems with trimap based approaches
- Iterate between solving for F,B and solving for $\alpha$
- Accurate trimap required

A closed form solution to natural images matting

Anat Levin, Dani Lischinski and Yair Weiss
Presented at CVPR 2006

$$I_i \approx \alpha_i F_i + (1 - \alpha_i) B_i$$

- Analytically eliminate F,B. Obtain quadratic cost in $\alpha$
- Provable correctness result
- Quantitative evaluation of results

The matte as a linear function of intensity

Assume F,B are approximately constant in a window:

$$I_i \approx \alpha_i F_i + (1 - \alpha_i) B_i$$

$$\alpha_i \approx a I_i + b$$

$$a = \frac{1}{F - B}, b = -\frac{B}{F - B}$$

Profiles
Linear model from color lines

**Observation:**
If the F,B colors in a local window lie on a color line, then
\[ \alpha_i = a^R i R + a^G i G + a^B i B + b \quad \forall i \in w \]

**Linear relation- 1 channel case**

Assume F,B are approximately constant in a window:

\[ I_i \approx \alpha_i F + (1 - \alpha_i) B \quad i \in w \]
\[ \alpha_i \approx aI_i + b \quad a = \frac{1}{F - B}, b = \frac{-B}{F - B} \]

\[ \approx 2 \quad +1 \]
Examples for linear relations

\[
\begin{align*}
(0, 0, 0) & \Rightarrow (0, 0, 0) = 0 \cdot 0 + 0 \cdot 0 + 0 \cdot 0 + 1 \\
(2, 0, 0) & \Rightarrow (0, 0, 0) = -2 \cdot 0 + 0 \cdot 0 + 0 \cdot 0 + 1 \\
(-1, 2, 0) & \Rightarrow (0, 0, 0) = -1 \cdot 2 + 0 \cdot 0 + 0 \cdot 0 + 0
\end{align*}
\]

Linear model from color lines

Observation:
If the F, B colors in a local window lie on a color line, then
\[
\alpha_i = a^R R_i + a^G G_i + a^B B_i + b \quad \forall i \in w
\]

Result: F, B can be eliminated from the matting cost

Evaluating an \(\alpha\)-matte

\[
J(\alpha) = \sum_{w \in I} d(\alpha_w, \text{Span}\{R_w, G_w, B_w\}) + \varepsilon \cdot \text{smoothness}(\alpha)
\]

Evaluating an \(\alpha\)-matte, 1 channel case

Minimize:
\[
\sum_{j \in I} \sum_{w \in W_j} (\alpha_i^j + b_j - \alpha_j)^2
\]
Evaluating an $\alpha$-matte, 1 channel case

Minimize:

$$\sum_{j=1}^{n} \sum_{i \in W_j} \left( a_i I_i + b_j - \alpha_i \right)^2$$

Solving for $\alpha$ using linear algebra

Input:

Image + user scribbles

$$\alpha = \arg \min_{\alpha} \alpha^T L \alpha$$

s.t.

$$\alpha_i = 0, \quad i \in \mathbb{S}$$

$$\alpha_i = 1, \quad i \in \mathbb{G}$$

$$L(i,j) = \sum_{\mu \in W_{i,j}} \left[ 1 + (C_i - \mu)(C_j - \mu) \right]$$

Cost minimization and the true solution

Theorem:

Given: $I = F^T F + (1 - \alpha^T) B$

If:

- $F, B$ locally on color lines
- Constraints consistent with $\sigma$

Then:

$$\alpha^* = \arg \min_{\alpha} \alpha^T L \alpha$$

s.t.

$$\alpha_i = 0, \quad i \in \mathbb{S}$$

$$\alpha_i = 1, \quad i \in \mathbb{G}$$

Advantages:

- Quadratic cost - global optimum
- Solve efficiently using linear algebra
- Provable correctness
- Insight from eigenvectors

Matting and spectral segmentation

Spectral segmentation: Analyzing smallest eigenvectors of a graph Laplacian $L$ (E.g. Normalized Cuts, Shi&Malik 97)

$$L = D - W$$

$$W_{ij} = \exp^{-\|I_i - I_j\|^2}$$

$$W_{\text{cut}, ij} = \sum_{\mu \in W_{i,j}} \left[ 1 + (C_i - \mu)(C_j - \mu) \right]$$

Theorem

$F, B$ locally on color lines

$$J(\alpha) = \sum_{w \in \Gamma} d(\alpha_w, \text{Span} \{ R_w, G_w, B_w, 1 \})$$

$$= \alpha^T L \alpha$$

Where $L(i,j)$ local function of the image

$$L(i,j) = \sum_{\mu \in W_{i,j}} \left[ 1 + (C_i - \mu)^2 (C_j - \mu)^2 \right]$$

Input:

Image + user scribbles

$$\alpha = \arg \min_{\alpha} \alpha^T L \alpha$$

s.t.

$$\alpha_i = 0, \quad i \in \mathbb{S}$$

$$\alpha_i = 1, \quad i \in \mathbb{G}$$

$\alpha^*$

Spectral segmentation: Analyzing smallest eigenvectors of a graph Laplacian $L$. (E.g. Normalized Cuts, Shi&Malik 97)
Comparing eigenvectors

Eigenvectors as guides

Input Image  Eigenvectors  Scribbling  Matte

Matting results

Comparing results

Quantitative results

Experiment Setup:
- Randomize 1000 windows from a real image
- Create 2000 test images by compositing with a constant foreground using 2 different alpha mattes
- Use a trimap to estimate mattes from the 2000 test images, using the different algorithms
- Compare errors against ground truth
Quantitative Results

<table>
<thead>
<tr>
<th>Input</th>
<th>Trimap</th>
<th>Ground Truth</th>
<th>Poisson</th>
<th>Random Walk</th>
<th>Wang &amp; Cohen</th>
<th>Ours</th>
</tr>
</thead>
</table>

Quantitative results

Smoke Matte

Circle Matte

Conclusions

- Analytically eliminate F, B and obtain quadratic cost $\alpha^T L \alpha$.
- Solve efficiently using linear algebra.
- Provable correctness result.
- Connection to spectral segmentation.
- Quantitative evaluation.

Environment Matting and Compositing

Environment Matting Equation

$C = F + (1 - \alpha)B + \Phi$

- $C$ – Color
- $F$ – Foreground color
- $B$ – Background color
- $\alpha$ – Amount of light that passes through the foreground
- $\Phi$ – Contribution of light from Environment that travels through the object

Explanation of $\Phi$

$\Phi = \sum_{p \in S} \int R(x) T(x) dx$

- $R$ – reflectance image
- $T$ – Texture image

Code available:
http://www.cs.huji.ac.il/~alevin/matting.tar.gz

Slides by Jay Hetler

Douglas E. Zongker ~ Dawn M. Werner ~ Brian Curless ~ David H. Salesin
Performance
Calibration
Matting: 10-20 minutes extraction time for each texture map (Pentium II 400Mhz)
Compositing: 4-40 frames per second
Real-Time?

How much better is Environment Matting?

How much better is Environment Matting?

Movies!