A Randomized Algorithm for Natural Object Colorization

EUROGRAPHICS 2014
Sou-Young Jin  Ho-Jin Choi  Yu-Wing Tai

Kim, Wook
2014. 3. 28

Computer Graphics @ Korea University
Abstract

Grayscale image  Reference image

- Conventional results
  - Little color variation

- EG 2014 result
  - Wide variety of colors

Conventional results
Left: using reference, right: using strokes

EG 2014 result
Abstract

• Natural objects contain vivid color distribution
  ▪ Need to consider the **distribution** of target color

  ![How to?]

• Models the correlation between intensity and color
  ▪ H-S, H-V, S-V joint histogram

• Randomly swap and reassign color of a pixel
  ▪ For color consistency
Related Work

Scribble based methods

- Two consideration
  - The colors at marked pixels should be close to the user input
  - Similar pixels should have similar color

- Produce significant color blending
Related Work

Example based methods

- Two representative categories
  - Measuring texture similarity
  - Transferring mean color globally or locally

- The \textbf{same color values} will be assigned to pixels with the \textbf{same grayscale intensity}
Contribution

• Using scribble-based methods
  ▪ Time-consuming when the target object is too complicated

• Using example-based methods
  ▪ Not consider the randomness in natural object

  Insight is..

• Transfer the distribution of target colors instead of directly transferring color value
Algorithm

• Sampling Procedures in building the joint histogram
  ▪ To capture the intensity-to-color correlation

• Modeling objective function and randomized algorithm to minimize the objective function

• Describing implementation detail
  ▪ Using multilevel for speed up
Preprocessing

- Input images are segmented and each segment consists of only one semantic region.
Color sampling in an example image

- Build H-S, H-V and S-V joint histogram
  - Quantize the value of each channel into 60 bins
Objective Function

\[
\arg\min_{H,S} E_s(H,S|V) + \lambda E_d(H,S|V)
\]

- \(E_s(H,S|V)\) is the color consistency term to measure the color similarity between a pixel and its neighbors.

- \(E_d(H,S|V)\) is the negative log of probability to assign certain color to a pixel given the intensity value.

- Lambda is the relative weight.
  - Set to 1.
**Objective Function**

**Data Term**

\[
E_d(H,S|V) = -\log(P(H|V')P(S|V')P(H|S))
\]

\[
V' = \frac{\sigma_{\text{example}}}{\sigma_{\text{grayscale}}} (V - \mu_{\text{grayscale}}) + \mu_{\text{example}}
\]

- \(E_d(H,S|V)\) measures the fitness of assigning certain colors in H and S channels to a pixel given V.

- Correlation between the assigned values in H and S channels are captured in \(P(H|S)\).
Objective Function

Smoothness Term

\[ E_s(H, S|V) = \sum_p \sum_{q \in N(p)} w_{pq} (|S(p) - S(q)|) \]
\[ + \frac{1}{180} \min(|H(p) - H(q)|, 360 - |H(p) - H(q)|) \]

- q is the neighborhood of p
- \( W_{pq} \) is the weighting term
Objective Function

Smoothness Term (two different weight)

- Local weighting term (pink)
  - Ensures local color smoothness

- Nonlocal weighting term (Green)
  - Ensures colors to be propagated across nonlocal neighbors with similar structures
Objective Function

Smoothness Term (local weight)

\[ \tilde{W}_{pq} = \exp\left(-\frac{|V(p) - V(q)|^2}{2\sigma_p^2}\right) \]

- Sigma is the variance of intensities in 5X5 window

- Local weighting term is defined over four first-order neighboring pixels
  - **Minimizing the color difference** between local neighboring pixels if they have similar intensity
Objective Function

Smoothness Term (nonlocal weight)

\[ \tilde{W}_{pq} = \exp(-\sum_{r \in W} G(r) \| V(p - r) - V(q - r) \|^2) \]

- \( W \) is 7X7 local window
- \( G(r) \) is a Gaussian function
- The nonlocal neighborhoods are searched within 21X21 window around \( p \) and we keep the top 10 nonlocal neighbors
Minimizing Objective Function

Input images  | Original image  | Optimal color assignment  | EG 2014

Selecting only maximum probability minimizes objective function

Summation is 1

0  0.1  0.1  0.3  0.15  0.15  0.2  0.2  360  Hue

Intens-ity

0.1  0.1  0.3  0.15  0.15  0.2  0.2

Kim, Wook | 2014. 3. 28 | # 16
Randomized Algorithm

Initialization

- Given the intensity value
  - Randomly assign S value using S-V joint histogram
  - Randomly assign H value using H-S joint histogram
Randomized Algorithm

Swap and Reassign

- **Swap (90%)**
  - Randomly pick two pixels and swap their color
  - Compute the local energy
  - If the energy after swapping is lower, accept the swapping else undo the swapping

- **Reassign (10%)**
  - Randomly pick one pixel and change its color
  - Compute the local energy
  - If the energy after assignment is lower, accept the assignment else undo the assignment

- **Apply median and joint bilateral filter after each iteration**
  - To remove noise and isolated pixel colors
Multilevel Implementation

- Number of iteration in each step is 5 times total number of pixels
  - Due to the better initialization
# Running Time

<table>
<thead>
<tr>
<th>Examples</th>
<th>Resolution</th>
<th>Running Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Small Tree</td>
<td>1000×750</td>
<td>13.57 sec.</td>
</tr>
<tr>
<td>Needle Tree</td>
<td>1000×750</td>
<td>13.26 sec.</td>
</tr>
<tr>
<td>Hairs</td>
<td>1000×750</td>
<td>14.02 sec.</td>
</tr>
<tr>
<td>Wood</td>
<td>1000×750</td>
<td>12.84 sec.</td>
</tr>
<tr>
<td>Lizard</td>
<td>500×330</td>
<td>2.54 sec.</td>
</tr>
<tr>
<td>Toad</td>
<td>1000×816</td>
<td>14.81 sec.</td>
</tr>
</tbody>
</table>

**Table 1:** The running time of our algorithm implemented in Matlab in a machine with Intel Core\textsuperscript{TM} i5-3570 CPU@3.4GHz with 4.00 GB RAM.
Experimental Results

Transferring Color Richness

Input Grayscale image

Euro2014 Results

Original image
Experimental Results

Usage of a Randomized Algorithm

- Run algorithm twice on the same image
  - Produce different results
Experimental Results

Comparison to other techniques

Small tree

Welsh et al. [WAM02]

Levin et al. [LLW04]

Tai et al. [TJT05]

arg max color

Graph cut

Our results
Experimental Results

Failure Cases

• Left: Color of reference object is unlikely to appear on the target object

• Right: Colorization result without segmentation
### User Study

<table>
<thead>
<tr>
<th></th>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
<th>4th</th>
<th>5th</th>
<th>6th</th>
</tr>
</thead>
<tbody>
<tr>
<td>Welsh et al. [WAM02]</td>
<td>8.5 (11.0)</td>
<td>18.8 (23.6)</td>
<td>19.4 (21.7)</td>
<td>20.8 (18.4)</td>
<td>18.3 (14.1)</td>
<td>14.2 (11.1)</td>
</tr>
<tr>
<td>Levin et al. [LLW04]</td>
<td>18.8 (14.4)</td>
<td>16.0 (10.4)</td>
<td>11.7 (7.0)</td>
<td>8.6 (10.6)</td>
<td>7.1 (9.7)</td>
<td>37.9 (47.9)</td>
</tr>
<tr>
<td>Tai et al. [TJT05]</td>
<td>6.0 (11.9)</td>
<td>15.8 (16.7)</td>
<td>23.8 (20.4)</td>
<td>24.0 (20.0)</td>
<td>19.3 (21.7)</td>
<td>11.1 (9.3)</td>
</tr>
<tr>
<td>arg max color</td>
<td>8.9 (14.4)</td>
<td>17.8 (17.6)</td>
<td>20.4 (19.4)</td>
<td>20.1 (21.3)</td>
<td>22.2 (19.3)</td>
<td>10.6 (8.0)</td>
</tr>
<tr>
<td>Graph cut</td>
<td>4.6 (5.9)</td>
<td>13.3 (14.3)</td>
<td>16.1 (17.6)</td>
<td>20.0 (20.9)</td>
<td>25.7 (24.6)</td>
<td>20.3 (16.9)</td>
</tr>
<tr>
<td><strong>Our results</strong></td>
<td><strong>53.3 (42.7)</strong></td>
<td><strong>18.1 (17.3)</strong></td>
<td><strong>8.8 (13.9)</strong></td>
<td><strong>6.5 (8.6)</strong></td>
<td><strong>7.4 (10.4)</strong></td>
<td><strong>6.0 (7.1)</strong></td>
</tr>
</tbody>
</table>

- 53.3% of subjects rank our results the most natural
- 42.7% of subjects rank our results the most natural after knowing the reference images

More unnatural
Discussion and Conclusion

- Transfer the global color distribution
  - For color richness

- Future Work
  - Semantic colorization
  - Local user control