Sparse Pixel Sampling for Appearance Edit Propagation

Tatsuya Yatagawa  (The University of Tokyo)
Yasushi Yamaguchi  (The University of Tokyo / JST CREST)
Stroke-based image editing

Appearance-space edit propagation (AppProp)
[An and Pellacini 2008]

This is simple and intuitive to use.
Energy function of AppProp

\[ f_{\text{Energy}} = \sum_{i=1}^{n} \sum_{j=1}^{n} w_j z_{ij} (e_i - g_j)^2 + \sum_{i=1}^{n} \sum_{j=1}^{n} z_{ij} (e_i - e_j)^2 \]

- Stroke-compatibility
- Edit-conformity

© ACM, An and Pellacini 2008
Acceleration of AppProp

Original AppProp

Original & Strokes  →  Quadratic energy minimization  →  Result

Acceleration by sampling & interpolation

[Xu et al., 2009, Bie et al., 2011]

Original & Strokes  →  Samples (k-means)  →  Quadratic energy minimization  +  Interpolation  →  Result
Problems on sampling

**Stroke-dependent**
[Xu et al. 2009]
😊 Need resampling

**Stroke-independent**
[Bie et al. 2011]
😊 Sample only once

We use independent one

(a) Stroke-dependent sampling

user interaction → sampling → interpolation → sampling → interpolation → ...

(b) Stroke-independent sampling

user interaction → interpolation → interpolation → interpolation → interpolation → ...

Time
Problems on interpolation

K-means & Convex combination
[Bie et al. 2011]

Sparse pixel sampling & Affine combination
[Ours]

K = 4

Samples lie inside of the set
Convexity is inappropriate

Use extremum samples
Use affine combination
Comparison of approaches

<table>
<thead>
<tr>
<th>Approach</th>
<th>Resampling</th>
<th>Interpolation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Xu et al. 2009</td>
<td>K-d tree</td>
<td>Necessary</td>
</tr>
<tr>
<td>Bie et al. 2011</td>
<td>K-means</td>
<td>Unnecessary</td>
</tr>
<tr>
<td>Ours</td>
<td>Sparse pixel sampling</td>
<td>Unnecessary</td>
</tr>
</tbody>
</table>

Features of our approach

- sample pixels **only once** as a precomputation
- use **affine** to improve interpolation flexibility
Demo (system overview)
Problem statement

Affine color reconstruction

\[ I_i \approx \frac{\sum_{c \in C} w_{ic} a_c I_c}{\sum_{c \in C} w_{ic} a_c} \]

- \( I_i \): pixel color
- \( C \): sample pixels
- \( w_{ic} \): pixel similarity
- \( a_c \): weight coefficient

Problem we solve for sampling

\[ \min_{\{a_c\}} \frac{1}{2} \sum_{i \in I} \left( I_i - \frac{\sum_{c \in C} w_{ic} a_c I_c}{\sum_{c \in C} w_{ic} a_c} \right)^2 \]
Sparse pixel sampling

- Samples are unknown
- Neither convex nor linear.

→ Solve the problem iteratively

Initialization

Take $C^0$, the candidate for final samples

↓

Sample discarding (decrease samples) ↔ Greedy resampling (increase samples)
Sample discarding

\[
\min \left\{ a_c \right\} \frac{1}{2} \sum_{i \in Q} \left( I_i - \frac{\sum_{c \in C^0} w_{ic} a_c I_c}{\sum_{c \in C^0} w_{ic} a_c} \right)^2
\]

\[
\downarrow \text{Relaxation} \quad \text{(refer to the paper for detail)}
\]

Sample discarding

\[
\min \frac{1}{2} \| I - WDa \|^2 + \frac{1}{2} \| Da - 1 \|^2 + \lambda \| a \|_1
\]

\[
\rightarrow \text{Pixels with zero coefficients are discarded}
\]
Greedy resampling / Update rule

Resampling criteria

\[
Error(p) = \left| \frac{\sum_{c \in C} w_{ic} a_c I_c}{\sum_{c \in C} w_{ic} a_c} - I_p \right| > \tau
\]

Final update rule

\[
C^{k+1} = \{c \in C^k : a_c \neq 0\} \cup \{p \in Q : Error(p) > \tau\}
\]
Sampling results

Please look at our paper for discussions.
Online image editing

Original / Strokes

Samples

Samples w/ edits

Result

Quick response ( < 1sec)

Resolution: 1200 x 800
Demo

Resolution: 1200 x 800
Results

(a) (b) (c) (d)

(a) (b) (c) (d)

Please visit our project website for other results 😊
Comparison: edit results

(a) Input with strokes  
(b) An and Pellacini 2008  
(c) Li et al. 2010

(d) Bie et al. 2011  
(e) Xu et al. 2013  
(f) Ours

Preserve high frequency details & Almost interactive feedback
Comparison: approximation errors

- **approximated image**
- **difference**

(a) k-means

(b) Ours

😊 Better approximation for detailed regions
Comparison: amount of strokes

Tradeoff between task amount and accuracy
Comparison with state-of-the-art [Xu et al., 2013]

(a) Xu et al. 2013
6 strokes

(b) Ours
13 strokes

Amount of user tasks

Interactive feedback

Tradeoff!!
Conclusion and future work

Contributions

• Sparse pixel sampling can be performed only once
• Better interpolation with affine combination

Sparse pixel sampling

• Sample discarding with compressive sensing
• Greedily resampling with approximation errors

Future work

• Explore a better sampling approach for videos
• Find new applications of the sparse pixel sampling
Thank you for your kind attention!!

**MATLAB demo** is available in our project website!!

http://www.graco.c.u-tokyo.ac.jp/~tatsuya/projects/sps/

(This link also appears in the head of our paper)