Restoration of Old Films Using Morphological Operators

Quôc Peyrot chojin@lrde.epita.fr

LRDE seminar, September 25, 2002
Table des matières

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>2</td>
</tr>
<tr>
<td>History background</td>
<td>3</td>
</tr>
<tr>
<td>Mathematical morphology</td>
<td>4</td>
</tr>
<tr>
<td>Connexity</td>
<td>6</td>
</tr>
<tr>
<td>Operators</td>
<td>8</td>
</tr>
<tr>
<td>Restoration of vertical scratches</td>
<td>18</td>
</tr>
<tr>
<td>Restoration of local random defects</td>
<td>31</td>
</tr>
<tr>
<td>Conclusion</td>
<td>47</td>
</tr>
</tbody>
</table>
Introduction

- Invention of “Lumière Cinematograph” in 1895 by Lumière brothers
- Deterioration due to short life span of films used
  - It is urgent to restore and preserve this cultural inheritance.
History background

• Classic way: physic and chemical restorations
  ▶ Time-consuming and money-consuming

European Commission created two projects:

• AURORA Project: AUtomated Restoration of ORiginal film and video Archives
  (Ended in 1999)

• Brava Project: BRoadcast Archives Restoration Through Video Analyses
  (Feb 2000 - Nov 2002)
Mathematical morphology

- Instantiated by G. Matheron and J. Serra in 1964

  ▶ Study geometric structures


  ▶ Bases of this study
Implementation

- Language: C++
- Image processing and Mathematical morphology provided by Olena: a generic image processing library in C++ from EPITA Research and Development Laboratory (LRDE\(^1\))

\(^1\text{http://www.lrde.epita.fr}\)
Connexity

Common connexity:

- 4-connexity
- 8-connexity
- 6-connexity (for hexagonal)

▷ we can have 3d-connexity and time connexity!
window2d win;
win.add(0, 0).add(1, 0).add(0, 1).add(-1, 0).add(0, -1);
Operators

- Primitives: Dilation and erosion
- Closing - Opening
- Reconstruction
- Area filter
- White Top-hat / Black Top-hat
Image sample

Figure 2: Image sample
Mathematical morphology

Operators

Dilation - Erosion

Figure 3: Dilation - Erosion

- **Dilation**: $\delta_S(A)$
- **Erode**: $\epsilon_S(A)$

```cpp
image2d<int_u8> result = morpho::dilation(lena, win_c8p());
image2d<int_u8> result = morpho::erosion(lena, win_c8p());
```
Mathematical morphology Operators

Closing - Opening

Figure 4: Closing - Opening

- Closing: \( \gamma_S(A) = (\epsilon_S \circ \delta_S)(A) \)
- Opening: \( \phi_S(A) = (\delta_S \circ \epsilon_S)(A) \)

```cpp
template<
    typename image_t,
    image_t::order order
>
image_t<order> morpho::closing(const image_t &input, const image_t::window &win);
```
```cpp
template<
    typename image_t,
    image_t::order order
>
image_t<order> morpho::opening(const image_t &input, const image_t::window &win);
```
Reconstruction

Figure 5: Reconstruction: mark and result

```cpp
morpho::sure_geodesic_reconstruction_dilation(marker,
                                          mask,
                                          neighb_c8());
```
Area filters

Figure 6: Minima killing with area=9

```
level::fast_minima_killer(image, area, neighb_c8());
```
White Tophat - Black Tophat

- **White Tophat**: \( wth_S(I) = I - \gamma_S(I) \)
- **Black Tophat**: \( bth_S(I) = \phi_S(I) - I \)

```
morpho::white_top_hat(lena, win_c8p());
morpho::black_top_hat(lena, win_c8p());
```
Gray images

\[ I = \sum_{g=1}^{G-1} \text{Threshold}_g(I) \]

\[ \text{op}(I) = \sum_{g=1}^{G-1} \text{op}(\text{Threshold}_g(I)) \]

▷ This is slow: there are other ways
Operators on Lena

Figure 8: Original Lena

Figure 9: Dilation and Erosion
Operators on Lena

Figure 10: Closing and Opening

Figure 11: Black and White Tophat
Restoration of vertical scratches

Characteristics:

- vertical lines
- long
- narrow
- brighter than neighbour
- do not move with time
Restoration of vertical scratches

Detection - Modelisation

- \(d_x < D_x\)
- \(d_y < D_y\)
- the artifact is a local maxima
- \(d_t < D_t\)
Restoration of vertical scratches

Detection - Algorithm 1

1. Horizontal white tophat with size $D_x$
2. Vertical opening with size $D_y$
3. Time opening with size $D_t$
4. Space-time reconstruction
5. Threshold
Restoration of vertical scratches

Detection - Algorithm 1 : result

Figure 12: Image source
Restoration of vertical scratches

Detection - Algorithm 1 : result

Figure 13: White tophat and 2D Opening
Detection - Algorithm 1 : result

Figure 14: 3D opening and Threshold
Detection - Algorithm 1: drawbacks

- Images are often noisy
  - Vertical scratches are discontinuated
- We can have near-vertical lines in the image
  - False detections
- Motion is often small
  - Time opening is not so useful
Restoration of vertical scratches

Detection - Algorithm 2

1. Horizontal white tophat with size $D_x$ (a)

2. Column accumulator

3. Horizontal white tophat with size $D_x$ on accumulator

4. Threshold on accumulator with $g_1$ (b)

5. Threshold on (a) with $g_2$, $g_2 < g_1$ (c)

6. Reconstruction of vertical lines using (b) as “column marker” on (c)
Restoration of vertical scratches

Detection - Algorithm 2 : result

Figure 15: Vertical scratches - Algorithm 2
Restoration of vertical scratches

Detection - Algorithm 2 : result

Figure 16: Vertical scratches - Detection result
Restoration of vertical scratches

Restoration

- We don’t have time information

▷ we have to work only in 2D
Restoration of vertical scratches

**Restoration - “algorithm”**

1. Neighbour pixels can be deteriorate by the artifact
   ▶ Dilation of mask using $C_8$ connexity

2. Horizontal opening on original image with size $D_x$

3. “cut and paste” using mask
Restoration of vertical scratches

Restoration - sample

Figure 17: Restoration sample
Restoration of local random defects

Figure 18: Image sample
Restoration of local random defects

Detection - first approach

1. Detect on each frame potential local random defects

2. Apply time criteria to reject false detections
Restoration of local random defects

Detection - Modelisation

Characteristics:

- Maximal area $S_{max}$
- Minimal area $S_{min}$
- Minimal contrast $g$
Restoration of local random defects

Detection - Algorithm

Input F:

1. Maxima killing with area \( S_{max} \Rightarrow F_1 \)

2. \( F_2 = F - F_1 \)
   \( \triangleright F_2 \) has only maxima with area \( < S_{max} \)

3. Threshold \( F_2 \) with \( g \Rightarrow F_3 \)

4. Reconstruction using \( F_2 \) as mask and \( F_3 \) as mark \( \Rightarrow F_4 \)

5. Binarize of \( F_4 \) using \( g_{low} \Rightarrow F_5 \)

6. Killing of area \( < S_{min} \)
Detection - Results

Figure 19: Original Image
Detection - Results

Figure 20: Maxima and Threshold
Detection - Results

Figure 21: Reconstruction and final frame detection
Detection - Results

Figure 22: Time analysis and final mask
Restoration of local random defects

Detection - Algorithm 1 drawback

- Time consuming
- False detections
Restoration of local random defects

Detection - Second approach

1. Space-Time analysis

2. Post-processing
Detection - Algorithm 2

1. Space-Time opening $\Rightarrow I_1$

2. $I_2 = I - I_1$

3. Threshold $I_2$ with $g_{low} \Rightarrow B_1$

4. Threshold $I_2$ with $g_{high} \Rightarrow B_2$

5. Reconstruction using $B_1$ as mask and $B_2$ as mark
Detection - Results

Figure 23: Space-time opening and difference
Detection - Results

Figure 24: Threshold $g_{low}$ and $g_{high}$
Restoration of local random defects

Detection - Results

Figure 25: Final detection mask
Restoration of local random defects

Restoration

- Tiny defects: gaussian for instance
- Medium defects: we can use Fourier domain
- Big defects: we have to use time informations
Restoration of local random defects

Restoration sample

Figure 26: Restoration sample

- Maybe we can adapt $g_1$ and $g_2$ for each frame
Conclusion

• Implementation of other time connexity criteria

• Implementation of Fourier domain restoration method

• Implementation of motion estimations

• Try to adapt $g_{low}$ and $g_{high}$ for each frame in local algorithm 2