Dynamic Range Compression
Preserving Local Image Contrast
for Digital Video Camera


Yusuke Monobe, Haruo Yamashita,
Toshihary Kurosawa, and Hiroaki Kotera

**School of Electrical Engineering and Computer Science**
**Kyungpook National Univ.**
Abstract

- **Novel dynamic range compression method**
  - Conventional method
    - Using knee curve
  - Designed to preserve a local image contrast
  - Improved contrast of the highlight region

- **Proposed algorithm**
  - Automatically and adaptively enhances the local image contrast in the highlight regions
Introduction

- Dynamic range compression
  - From high dynamic range to low dynamic range
  - Adopted to digital video camera

Fig. 1. Conventional knee curve and auto knee curve
◆ Characteristics
  – Strongly compresses the highlight range
  – Single tone mapping curve to a whole image
  – Recently proposed method
    • Spatially variant tone mapping algorithm

◆ Proposed algorithm
  – Assume to preserve local contrast
    • If the local contrast at any pixel is not changed, the visual contrast is also preserved
Novel operator for preserving local image contrast

- Visual contrast
  - Depends on the local contrast
  - Propose a novel operator to preserve the local contrast
  - Call this operator LCRT (Local Contrast Range Transform)

Fig. 2. Concept of LCRT
– Total system

Fig. 3. Total system of our proposed algorithm
– Mathematical condition

\[
\frac{g(x, y)}{g_{ave}(x, y)} = \frac{f(x, y)}{f_{ave}(x, y)}
\]

(1)

- Rewritten as follows

\[
G(x, y) - G_{ave}(x, y) = F(x, y) - F_{ave}(x, y)
\]

(2)

where \( G(x, y) : \) logarithmic value of \( g(x, y) \)

- Approximately satisfied

\[
G_{ave}(x, y) \cong P(F_{ave}(x, y))
\]

(3)

- First-order Taylor expansion as follows

\[
P(F_{ave}(x, y)) \cong P(F(x, y)) + \left\{ \frac{dP(F(x, y))}{dF(x, y)} \right\} \cdot (F_{ave}(x, y) - F(x, y))
\]

(4)

\[
G(x, y) = P(F(x, y)) + \left\{ 1 - \frac{dP(F(x, y))}{dF(x, y)} \right\} \cdot (F(x, y) - F_{ave}(x, y))
\]

(5)
Eq. (5) is rewritten to the following equation (see appendix):

$$g(x, y) = p(f(x, y)) \times \left( \frac{f(x, y)}{f_{ave}(x, y)} \right)^\alpha \left[1- \frac{f(x, y)}{p(f(x, y))} \frac{dp(f(x, y))}{df(x, y)} \right]$$

(6)

This equation is known as the tone mapping curve, which automatically enhances local contrast at each luminance level.

The ratio of the input luminance level $f(x, y)$ to the local average $f_{ave}(x, y)$ is unchanged at each pixel.

$$G(x, y) = P(F(x, y)) + \alpha \cdot \left[1- \frac{dP(F(x, y))}{dF(x, y)} \right] \cdot (F(x, y) - F_{ave}(x, y))$$

(7)

Rewritten to the following in the luminance domain:

$$g(x, y) = p(f(x, y)) \times \left( \frac{f(x, y)}{f_{ave}(x, y)} \right)^\alpha \left[1- \frac{f(x, y)}{p(f(x, y))} \frac{dp(f(x, y))}{df(x, y)} \right]$$

(8)
– Gain parameter
  • Preserved local contrast when $\alpha$ is set to 1.0

– Local average $f_{\text{ave}}(x, y)$
  • Spatial averaging filter $A(x, y)$

\[
f_{\text{ave}}(x, y) = \langle A(x, y) \otimes f(x, y) \rangle \tag{9}
\]

  – Gaussian filter

\[
A(x, y) = K \exp\left\{-\frac{(x^2 + y^2)}{\sigma^2}\right\} \tag{10}
\]

\[
\iint A(x, y) \, dx \, dy = 1 \tag{11}
\]
Application of LCRT into digital video camera

- Fundamental tone mapping curve for digital video camera
  - Determined tone mapping curve

Fig. 4. Approximate knee curve
– Tone mapping curve must be continuous

• Using a cubic curve as follows

\[
if (f_c(x, y) < t_c) \\
p_c(f_c(x, y)) = f_c(x, y)
\]

\[
else \\
p_c(f_c(x, y)) = a \cdot f_c(x, y)^3 + b \cdot f_c(x, y)^2 + c \cdot f_c(x, y) + d
\]  \hspace{1cm} (12)

where suffix \( c \) : value in the camera gamma domain

\[
p_c(t_c) = at_c^3 + bt_c^2 + ct_c + d = t_c
\]  \hspace{1cm} (13)

\[
p_c'(t_c) = 3at_c^2 + 2bt_c + c = 1
\]  \hspace{1cm} (14)

\[
p_c(m_c) = am_c^3 + bm_c^2 + cm_c + d = 1
\]  \hspace{1cm} (15)

\[
p_c'(m_c) = 3am_c^2 + 2bm_c + c = s_c
\]  \hspace{1cm} (16)

where \( m_c \) : maximum input level

\( s_c \) : differential coefficient of the conventional knee curve

at the maximum input value.
\[ s_c = \frac{1 - k_c}{m_c - k_c} \]  \hspace{1cm} (17)

- **Coefficient \(a, b, c, d\)**

\[
a = \frac{(s_c - 1)t_c + (2 - m_c - m_c s_c)}{(t_c - m_c)^3} \hspace{1cm} (18)
\]

\[
b = \frac{2(1 - s_c)t_c^2 + (m_c s_c + 2m_c - 3)(t_c + m_c)}{(t_c - m_c)^3} \hspace{1cm} (19)
\]

\[
c = \frac{s_c t_c^3 + (s_c - 4)m_c t_c^2 + (6 - m_c - 2m_c s_c)m_c t_c - m_c^3}{(t_c - m_c)^3} \hspace{1cm} (20)
\]

\[
d = \frac{(1 - m_c s_c)t_c^3 + (m_c s_c + 2m_c - 3)m_c t_c^2}{(t_c - m_c)^3} \hspace{1cm} (21)
\]
• Tone mapping curve

Fig. 5. Conventional and approximate knee curve

$$m_c = 2.0$$

$$k_c = 0.9$$

$$t_c > 0.35 :$$ overshoots the conventional knee curve
Application of LCRT to approximate knee curve

- Luminance domain

\[
if (f(x, y)) < t \\
\quad p (f (x, y)) = f (x, y) \\
else \\
\quad p(f(x, y)) = \left\{a \cdot f (x, y)^3 + b \cdot f (x, y)^2 + c \cdot f (x, y) + d\right\}^r
\]

- Luminance of the threshold level \( t = t_c^r \)

\[
if (f(x, y)) < t \\
\quad \frac{dp(f(x, y))}{df(x, y)} = 1 \\
else \\
\quad \frac{dp(f(x, y))}{df(x, y)} = r \times \left\{a \cdot f (x, y)^3 + b \cdot f (x, y)^2 + c \cdot f (x, y) + d\right\}^{r-1} \\
\quad \times \left\{3a \cdot f (x, y)^2 + 2b \cdot f (x, y) + c\right\}
\]
• LCRT

Fig. 6. Application of LCRT to approximate knee curve (easy saturation)

Applied to it by substituting Eq.(23) for Eq.(8)

\[
g(x, y) = p(f(x, y)) \times \left( \frac{f(x, y)}{f_{ave}(x, y)} \right)^{\alpha \left( \frac{1 - f(x, y)}{p(f(x, y))} \right)\frac{dpt(f(x, y))}{df(x, y)}}
\]

\text{Eq.(23)}

Parameter \( r = 2.2, m = 2.0', k = 0.9', t = 0.35', \alpha = 1.0 \)
- LCRT for deciding $\alpha$

To compensate the degradation in visual contrast:
- The input luminance is smaller than the local average
- Enlarged by setting a gain parameter $\alpha$ higher than 1.0

**Fig. 7.** Application of LCRT to approximate knee curve using adjusted parameter $\alpha$ (hardly saturation)
◆ System
– Transformed into the luminance component $Y$
– Tone mapped image
  • Applying the approximate knee curve into the $Y$
– Spatial averaging filter
  • Local contrast gain
Experimental results

Calibration

– Digital still camera
  • Fujifilm Finepix F710
  • Two sensors
    – S (sensitive) and R (range) sensor
      » S-sensor is normal sensor
      » R-sensor is less sensitive and extends the ability to capture dark and light of photos

Fig. 8. Calibration image captured by S sensor
- S and R sensor

Synthesize the images with the mixing ratios

\[
\text{if } (S(x, y) < \text{th0}) \\
\quad f(x, y) = S(x, y) \\
\text{else if } (S(x, y) < \text{th1}) \\
\quad f(x, y) = \frac{\text{th1} - S(x, y)}{\text{th1} - \text{th0}} \times S(x, y) + \frac{S(x, y) - \text{th0}}{\text{th1} - \text{th0}} \times \{17.399 \times R(x, y)\} \\
\text{else} \\
\quad f(x, y) = 17.399 \times R(x, y)
\]

**Fig. 9.** Relationship between pixel levels captured by S and R sensors
- Mixing ratio

![Graph showing mixing ratio of S image and R image to synthesize HDR images.](image)

*Fig. 10.* Mixing ratio of S image and R image to synthesize HDR images

- Normal digital camera
  - Without multiple range sensors cannot capture
    - LDR image
Fig. 11. Experimental result

- Result image

(a) Linear range compression

(b) Knee curve

(c) Proposed method
Result image (another approach)

Fig. 12. CIC paper
Conclusion and future work

- **Dynamic range compression**
  - To preserved visual contrast
  - Guarantees the continuities in the output level
  - Preserving the local contrast in the highlight regions

- **Future work**
  - Currently takes high computation cost with a large memory
  - More efficient implementation
Appendix

\[ G(x, y) = P(F(x, y)) + \left\{ 1 - \frac{dP(F(x, y))}{dF(x, y)} \right\} \cdot (F(x, y) - F_{\text{ave}}(x, y)) \]

\[
\begin{align*}
F(x, y) &= \log\{f(x, y)\} \\
F_{\text{ave}}(x, y) &= \log\{f_{\text{ave}}(x, y)\} \\
G(x, y) &= \log\{g(x, y)\} \\
G_{\text{ave}}(x, y) &= \log\{g_{\text{ave}}(x, y)\} \\
P(F(x, y)) &= \log\{p(f(x, y))\}
\end{align*}
\]
\[
g(x, y) = p(f(x, y)) \cdot \left( \frac{f(x, y)}{f_{\text{ave}}(x, y)} \right)^{1 - \frac{dP(F(x, y))}{dF(x, y)}}
\]

\[
\frac{dP[F(x, y)]}{dF(x, y)} = \frac{d[\log\{p(f(x, y))\}]}{d[\log\{f(x, y)\}]} = \frac{\left\{ \frac{d[\log\{p(f(x, y))\}]}{df(x, y)} \right\}}{\left\{ \frac{d[\log\{f(x, y)\}]}{df(x, y)} \right\}} = \frac{\left\{ \frac{d[\log\{p(f(x, y))\}]}{dp(f(x, y))} \cdot \left\{ \frac{dp(f(x, y))}{df(x, y)} \right\} \right\}}{\left\{ \frac{d[\log\{f(x, y)\}]}{df(x, y)} \right\}} = \frac{1}{p(f(x, y))} \cdot \left\{ \frac{dp(f(x, y))}{df(x, y)} \right\} = \frac{f(x, y)}{p(f(x, y))} \cdot \frac{dp(f(x, y))}{df(x, y)}
\]