

# HDR 画像におけるトーンマッピング関数の多段表現

## Multi-staging of Tone Mapping Functions for HDR Images

室伏辰耶<sup>†</sup>

岩橋政宏<sup>††</sup>

貴家仁志<sup>†</sup>

<sup>†</sup> 首都大学東京システムデザイン学部

<sup>††</sup> 長岡技術科学大学電機系

Tatsuya MUROFUSHI<sup>†</sup>

Masahiro IWAHASHI<sup>††</sup>

Hitoshi KIYA<sup>†</sup>

<sup>†</sup>Tokyo Metropolitan University

<sup>††</sup>Nagaoka University of Technology

### アブストラクト

本稿では、ハイダイナミックレンジ (HDR) 画像に対する階調変換の一つである、グローバルオペレータに基づくトーンマッピング関数処理を対象に、トーンマッピング関数の多段表現を提案する。提案法は、HDR 画像に対してトーンマッピング処理を施して得られたローダイナミックレンジ (LDR) 画像に、特定の処理を加えることにより、他のトーンマッピング関数を用いたのと同様の結果を直接得る方法である。その結果、提案法は、一度 HDR 画像を復元し再度 LDR 画像を得る方法に比べて、低メモリおよび低演算量の処理を可能とする。

### 1 はじめに

デジタルカメラや CG レンダリングに代表されるように、より広いダイナミックレンジの情報を必要とし、HDR 画像を取り扱うデバイスやアプリケーションが増えてきた。一方、ディスプレイなどの表示機器においては、技術的側面やコストなどの問題から、高いダイナミックレンジを持つ HDR 画像をそのまま表示できるデバイスは限られ、一般には普及していない。そのため、HDR 画像を従来の表示機器でも表示できるように、ダイナミックレンジを圧縮するトーンマッピング処理が広く研究されている [1-8]。

トーンマッピング処理において、最も重要な処理がトーンマッピング関数を用いたダイナミックレンジの圧縮である。トーンマッピング関数は、すべての画素に単一の関数を用いるグローバルオペレータ [1,2] と画素に応じて異なる関数を用いるローカルオペレータ [1,3] に大別される。グローバルオペレータは一般に計算量が少ないという利点がある。それに対して、ローカルオペレータは計算量が多いが細部の再現性に優れるという特徴を持つ。

トーンマッピング関数はそれぞれに特徴を持っており、使用するアプリケーションやデバイスに合わせて、適切

なトーンマッピング関数を選択する必要がある。しかし、トーンマッピング関数が多くのパラメータを持つことや、画質の評価が入力画像や主観に大きく依存することなどから、適切なトーンマッピング関数の選択は容易ではない。その結果、最適な結果を得るために異なるトーンマッピング処理を再度行うことも多い。

一般に、一度トーンマッピング処理をしてから別のトーンマッピング処理の結果を得ようとした場合、元の HDR 画像を用いてトーンマッピング処理を一からやり直すか、一度圧縮された輝度から HDR 画像の輝度を復元し、トーンマッピング処理をやり直す必要がある。このようなアプローチでは、メモリの消費量や演算量が大きくなってしまいう問題がある。

本稿では、元の HDR 画像の情報を必要とせず、一度トーンマッピング処理をして得られた LDR 画像の RGB 値のみを用いて、別のトーンマッピング処理と同様の結果を得るためのトーンマッピング関数の多段表現を提案する。これまでは、あるトーンマッピングされた信号値を別のトーンマッピングされた信号値に変換するような、トーンマッピング処理結果の変換はあまり研究されていなかった。提案する多段表現を用いることで、従来法に比べて低メモリ、低演算量で従来法と同様の処理結果を得ることができる。

### 2 準備

#### 2.1 トーンマッピング処理の概要

ここでは、図 1 に示す、トーンマッピング処理の基本的な流れを説明する [4,5]。

#### (A)HDR 画像の輝度計算

まず、トーンマッピング処理では、各画素  $p$  において HDR 画像の RGB 値から HDR 画像の輝度  $L_w(p)$  を求める。文献 [1] では、次式により HDR 画像の輝度  $L_w(p)$  を

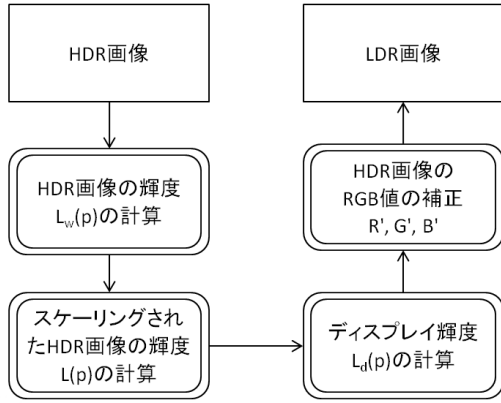


図 1: トーンマッピング処理の概要

計算している.

$$L_w(p) = 0.27R(p) + 0.67G(p) + 0.06B(p) \quad (1)$$

### (B)HDR 輝度のスケーリング

次に、次式を用いてスケーリングされた HDR 画像の輝度  $L(p)$  を計算する.

$$L(p) = a \cdot \frac{L_w(p)}{\bar{L}_w} \quad (2)$$

このとき、 $a$  は 0~1 の範囲のユーザー指定のパラメータであり、 $\bar{L}_w$  は HDR 画像の輝度  $L_w(p)$  の幾何平均である. 具体的に、 $\bar{L}_w$  は HDR 画像の総ピクセル数  $N$  を用いて次式で計算される.

$$\bar{L}_w = \exp\left(\frac{1}{N} \sum_p \log_e(\epsilon + L_w(p))\right) \quad (3)$$

ここで、 $\epsilon$  は対数の特異性を避けるための微小値である.

### (C) ディスプレイ輝度の計算

次に、トーンマッピング関数を用いて、スケーリングされた HDR 画像の輝度  $L(p)$  からディスプレイ輝度  $L_d(p)$  を計算する. このとき、スケーリングされた HDR 画像の輝度  $L(p)$  をトーンマッピング関数の入力として用いる.

後述するように、トーンマッピング関数には多くの種類がある [1-5]. 本稿は、このトーンマッピング関数の多段表現を考察するものである.

### (D)LDR 画像の生成

最後に、HDR 画像の RGB 値を次式を用いて補正し、LDR 画像の RGB 値を生成する.

$$\begin{aligned} R'(p) &= L_d \cdot \left(\frac{R(p)}{L_w(p)}\right)^s \\ G'(p) &= L_d \cdot \left(\frac{G(p)}{L_w(p)}\right)^s \\ B'(p) &= L_d \cdot \left(\frac{B(p)}{L_w(p)}\right)^s \end{aligned} \quad (4)$$

ここで、 $s$  は彩度を調整するためのユーザー指定のパラメータであり、0~1 の値をとる.  $s = 1$  と設定すると RGB の比率をコンスタントに保持し、 $s$  の値を小さくするほど彩度が低くなる. HDR 画像やトーンマッピング関数によっては、トーンマッピング後の LDR 画像の彩度が不自然に高くなることもあり、このような場合に  $s$  の値を調整することでより自然な仕上がりとなる.

## 2.2 トーンマッピング結果の相互変換

一般に、一度トーンマッピング処理をしてから別のトーンマッピング処理の結果を得ようとした場合、元の HDR 画像を用いて一からトーンマッピング処理をやり直すか、一度圧縮された輝度を HDR 画像の輝度に戻してからトーンマッピング処理をやり直す必要がある.

それに対して本稿では、図 2 に示すように、あるトーンマッピングされた信号値を別のトーンマッピングされた信号値に直接変換する方法を提案する.

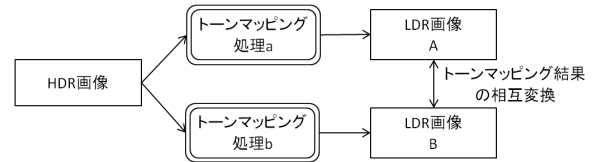


図 2: トーンマッピング結果の相互変換

## 2.3 トーンマッピング関数

ここでは、提案する多段表現で対象とするグローバルオペレータのトーンマッピング関数の例として、二つの一般化されたグローバルオペレータのトーンマッピング関数 [9] および代表的なトーンマッピング関数の一つである Reinhard のグローバルオペレータ [1] を示す.

### 2.3.1 Hill タイプのトーンマッピング関数

一般化されたトーンマッピング関数の一つとして、Hill タイプのトーンマッピング関数の一般形がある [9]. 0~1 の範囲に正規化された入力値  $x (= L(p))$  に対する Hill タイプのトーンマッピング関数の一般形は次式で示される.

$$y_H(x) = \frac{c}{(b/x)^a + 1}, \quad x \in (0, 1] \quad (5)$$

ここで、 $a$ 、 $b$  および  $c$  は実定数である. また、式 (5) の逆関数は、

$$x = y_H^{-1}(x) = \frac{b}{(c/y_H(x) - 1)^{1/a}} \quad (6)$$

で与えられる. さらに、式 (5) は、

$$c = b^a + 1$$

とすることで、出力値も 0~1 の範囲に正規化された Hill タイプのトーンマッピング関数の一般形として、

$$y_{Hn}(x) = \frac{x^a}{x^a + b^a}(1 + b^a), \quad x, y \in [0, 1] \quad (7)$$

と表現される。式 (7) の逆関数は、

$$x = y_{Hn}^{-1}(x) = \frac{b}{\{(b^a + 1)/y_{Hn}(x) - 1\}^{1/a}} \quad (8)$$

となる。

### 2.3.2 Reinhard のトーンマッピング関数

式 (5) は  $a = b = c = 1$  と置くことによって、代表的なトーンマッピング関数の一つである Reinhard のグローバルオペレータ [1]

$$y_R(x) = \frac{x}{1+x} \quad (9)$$

に帰着する。ここで、上式の逆関数は、

$$x = y_R^{-1}(x) = \frac{y_R(x)}{1 - y_R(x)} \quad (10)$$

と与えられる。

### 2.3.3 Log タイプのトーンマッピング関数

他の一般化されたトーンマッピング関数として、Log タイプのトーンマッピング関数がある [9]。0~1 の範囲に正規化された入力値  $x$  に対する Log タイプのトーンマッピング関数の一般形は、実定数  $\alpha$ ,  $\beta$  および  $\gamma$  を用いて、

$$y_L(x) = \frac{\log_e(\alpha x + 1) + \beta}{\gamma}, \quad x \in (0, 1] \quad (11)$$

と表現され、式 (11) の逆関数は、

$$x = y_L^{-1}(x) = \frac{\exp(y_L(x)\gamma - \beta) - 1}{\alpha} \quad (12)$$

として得られる。ここで、

$$\beta = 0, \quad \gamma = \log_e(\alpha + 1)$$

とすることで、出力値も 0~1 の範囲に正規化された Log タイプのトーンマッピング関数の一般形が、

$$y_{Ln}(x) = \frac{\log_e(\alpha x + 1)}{\log_e(\alpha + 1)}, \quad x, y \in [0, 1] \quad (13)$$

と与えられる。式 (13) の逆関数は、

$$x = y_{Ln}^{-1}(x) = \frac{(\alpha + 1)^{y_{Ln}(x)} - 1}{\alpha} \quad (14)$$

となる。

図 3 にトーンマッピング関数の一例を示す。

## 3 提案する多段表現

ここでは、図 2 に示したように、あるトーンマッピングされた信号値を別のトーンマッピングされた信号値に直接変換する方法を提案する。

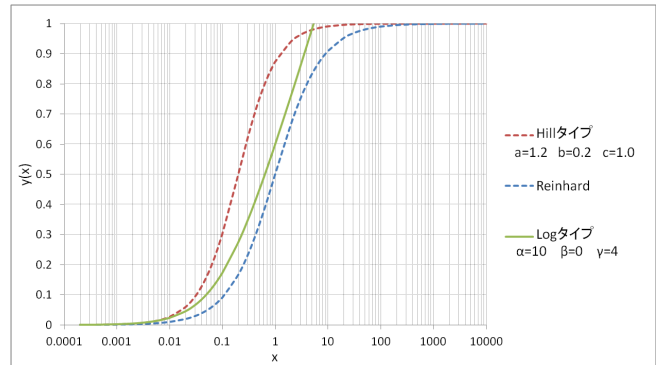


図 3: トーンマッピング関数例

### 3.1 トーンマッピング関数の多段表現

まず、提案する多段表現の準備として、ディスプレイ輝度の相互変換について説明する。トーンマッピング処理の過程で計算したディスプレイ輝度  $L_d(x)$  は、トーンマッピング関数の逆関数を用いてスケールされた HDR 画像の輝度  $L(x)$  に戻ることができる。ここで、あるトーンマッピング関数を用いて計算されたディスプレイ輝度  $L_d(x)$  に、そのトーンマッピング関数の逆関数と別のトーンマッピング関数との合成関数を用いることで、図 4 に示すようなディスプレイ輝度の相互変換が可能である。

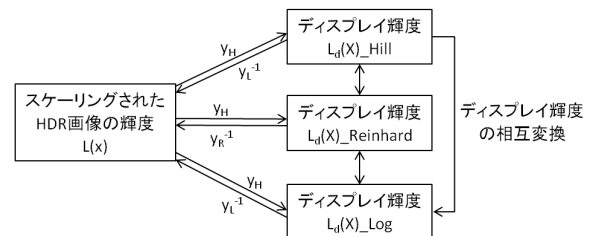


図 4: ディスプレイ輝度の相互変換

目標とするトーンマッピング関数の多段表現の概要を図 5 に示す。ここで、前段処理は一般的な手順のトーンマッピング処理である。後段処理では、前段処理で得られた LDR 画像を別のトーンマッピング処理によって得られる LDR 画像へ変換する。

図 6 には、LDR 画像の相互変換の詳細を示す。LDR 画像の相互変換は、ディスプレイ輝度の変換と RGB 値の変換から成り、前段処理で得られた LDR 画像の RGB 値  $R'_1$ ,  $G'_1$ ,  $B'_1$  を用いる。

### 3.2 ディスプレイ輝度の変換

今、前段処理で計算したディスプレイ輝度を  $L_{d1}(x)$ 、目標とする LDR 画像のディスプレイ輝度を  $L_{d2}(x)$  とおく。

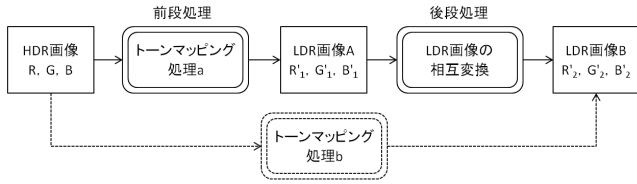


図 5: トーンマッピング関数の多段表現

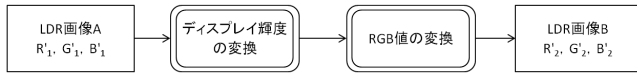


図 6: LDR 画像の相互変換

$L_{d1}(x)$  は、前段処理で得られる LDR 画像の RGB 値  $R'_1, G'_1, B'_1$  に式 (1) と同様の処理をすることで推定できる。先に述べたように、前段処理で用いたトーンマッピング関数の逆関数と目標とする LDR 画像に対応するトーンマッピング関数との合成関数を用いて、 $L_{d1}(x)$  から  $L_{d2}(x)$  を得ることができる。

例として、前段処理で式 (5) の Hill タイプのトーンマッピング関数を用いてトーンマッピングした LDR 画像を、式 (11) の Log タイプのトーンマッピング関数を用いてトーンマッピングした LDR 画像に変換する場合を説明する。このとき、入力  $x = y_H(x)$  であることに注意すると、式 (5) の逆関数である式 (6) と式 (11) との合成関数を用いて、

$$L_{d2}(x) = \frac{\log_e\left(\frac{\alpha b}{(c/y_H(x)-1)^{1/a}} + 1\right) + \beta}{\gamma} \quad (15)$$

と計算され、上式に  $1/y_H(x) = 1/L_{d1}(x)$  を乗じることで、

$$\frac{L_{d2}(x)}{L_{d1}(x)} = \frac{\log_e\left(\frac{\alpha b}{(c/y_H(x)-1)^{1/a}} + 1\right) + \beta}{y_H(x) \cdot \gamma}, \quad (16)$$

$(x \neq 0)$

を直接得ることができる。

### 3.3 RGB 値の変換

前段処理で得られた LDR 画像の RGB 値をそれぞれ  $R'_1, G'_1, B'_1$  とし、同様に目標とする LDR 画像の RGB 値をそれぞれ  $R'_2, G'_2, B'_2$  とすると、式 (4) より、

$$\begin{aligned} R'_2 &= R'_1 \cdot \frac{L_{d2}(x)}{L_{d1}(x)} \\ G'_2 &= G'_1 \cdot \frac{L_{d2}(x)}{L_{d1}(x)} \\ B'_2 &= B'_1 \cdot \frac{L_{d2}(x)}{L_{d1}(x)} \end{aligned} \quad (17)$$

という関係が得られる。ここで、ディスプレイ輝度の変換で求めた  $L_{d2}(x)/L_{d1}(x)$  を用いて上式を計算することにより、トーンマッピング結果の変換ができる。

## 4 シミュレーションと評価

### 4.1 LDR 画像の相互変換結果

ここでは、実際に HDR 画像に対してあるトーンマッピング処理をして得られた LDR 画像を、別のトーンマッピングした LDR 画像に変換した結果を示す。シミュレーションに用いた画像は、すべて RGBE フォーマットの HDR 画像である。

#### (A) 別タイプ間の変換

図 7 には、式 (5) の Hill タイプのトーンマッピング関数を与えて、一般的な手順でトーンマッピングした LDR 画像 ( $a = 1.2, b = 0.2, c = 1.0$ ) を示す。一方、図 8 には、式 (11) の Log タイプのトーンマッピング関数を与えて、一般的な手順でトーンマッピングした LDR 画像 ( $\alpha = 10, \beta = 0, \gamma = 4$ ) を示す。さらに、図 9 に、図 7 の LDR 画像を提案した多段表現を用いて図 8 の LDR 画像へ変換した結果を示す。

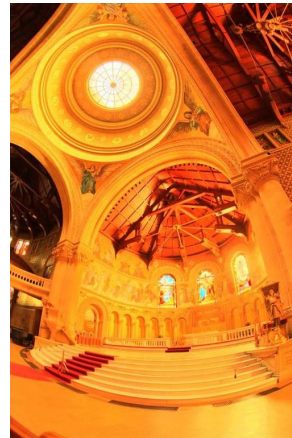


図 7: imageA\_Hill  
( $a=1.2 \ b=0.2 \ c=1.0$ )

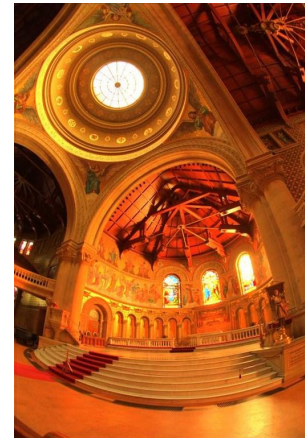


図 8: imageA\_Log  
( $\alpha=10 \ \beta=0 \ \gamma=4$ )

#### (B) 同タイプ間のパラメータ変換

提案した多段表現では、あるトーンマッピング処理をして得られた LDR 画像を、同じタイプだがパラメータの異なるトーンマッピング関数によって処理された LDR 画像に変換することも可能である。図 10 は、Hill タイプのトーンマッピング関数の一つである Reinhard のグローバルオペレータ (式 (9)) を与えて、一般的な手順でトーンマッピングした LDR 画像 ( $a = b = c = 1$ ) である。同様に、図 11 は、式 (5) の Hill タイプのトーンマッピング関数に別のパラメータを与えて、一般的な手順でトーンマッ

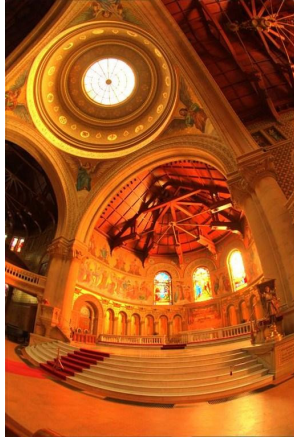


図 9: imageA\_conv

ピングした LDR 画像 ( $a = 1.2, b = 0.2, c = 1.0$ ) である。さらに、図 10 の LDR 画像から図 11 の LDR 画像へ変換した結果を図 12 に示す。



図 10: imageB\_Reinhard  
( $a=b=c=1$ )



図 11: imageB\_Hill  
( $a=1.2, b=0.2, c=1.0$ )



図 12: imageB\_conv

#### 4.2 変換結果の評価

提案する多段表現による LDR 画像の相互変換では、HDR 画像の輝度をトーンマッピングで圧縮した後に変換処理を行うため、変換後の LDR 画像は一般的な手順でトーンマッピングした LDR 画像と比べてわずかに誤差を含む。ここでは、PSNR と画像のコントラスト評価の観点から、生じる誤差が十分に小さいことを示す。

#### (A) PSNR

表 1 には、図 8 および図 9 に示した LDR 画像と、図 11 および図 12 に示した LDR 画像の二組の PSNR を示す。シミュレーション結果では、PSNR はいずれも高い

表 1: PSNR

	PSNR[dB]
imageA_Log	104.0771
imageA_conv	
imageB_Hill	inf
imageB_conv	

数値を示しており、一般的な手順でトーンマッピング処理をした LDR 画像と提案した多段表現による変換を行った LDR 画像間の誤差はわずかであった。

#### (B) コントラストの客観評価

ここでは、文献 [10]~[12] に示される方法を用いて、画像のコントラスト評価の観点から誤差の影響を考察する。表 2 に、画像のコントラスト評価の結果を示す。

表 2: コントラスト評価

	standard deviation[10]	relative entropy[11]
imageA_Log	30.4535	0.1535
imageA_conv	30.4536	0.1535
imageB_Hill	29.7575	0.1637
imageB_conv	29.7586	0.1637

	second order entropy[12]	Edge content based contrast[12]
imageA_Log	12.5918	12.1722
imageA_conv	12.5918	12.1722
imageB_Hill	11.8218	7.7292
imageB_conv	11.8218	7.7293

シミュレーションでは、一般的な手順でトーンマッピング処理をした LDR 画像と提案した多段表現による変換を行った LDR 画像間で、コントラスト評価の差はほぼなかった。

提案する多段表現によるトーンマッピング結果の変換結果は、一般的な手順によって得られる LDR 画像と比べて、PSNR およびコントラスト評価の観点から誤差はわ

ずかであり、二つの画像の違いを見分けることは困難である。提案する多段表現では、一度トーンマッピング処理をして得られた LDR 画像の RGB 値のみを用いるため、低メモリ、低演算量で別のトーンマッピング処理による結果を得ることができる。

## 5 まとめ

本稿では、グローバルオペレータのトーンマッピング関数を対象とした、HDR 画像におけるトーンマッピング関数の多段表現を提案した。提案した多段表現は、あるトーンマッピングされた LDR 画像を別のトーンマッピングされた LDR 画像に変換できる。ここで、提案した多段表現による変換は、一度トーンマッピング処理をして得られた LDR 画像の RGB 値のみを用いる。そのため、従来の方法と比べて低メモリ、低演算量で再度トーンマッピング処理を行うことができる。

## 参考文献

- [1] E. Reinhard, M. Stark, P. Shirley, and J. Ferwerda, "Photographic Tone Reproduction for Digital Images" *ACM Tran.Graphics*, Vol.21, No.3, p.267-276, 2002
- [2] F. Drago, K. Myszkowski, T. Annen, and N. Chiba, "Adaptive logarithmic mapping for displaying high contrast scenes" *EUROGRAPHICS*, 2003
- [3] R. Fattal, D. Lischinski, and M. Werman, "Gradient Domain High Dynamic Range Compression" *ACM Transaction on Graphics*, pp.249-256, 2002
- [4] E. Reinhard, G. Ward, S. Pattanaik, P. Debevec, W. Heidrich, and K. Myszkowski, "High Dynamic Range Imaging - Acquisition, Display and Image based Lighting" *Morgan Kaufmann*, 2010
- [5] 奥田正浩, "高ダイナミックレンジ画像処理技術と MATLAB シミュレーション~High Dynamic Range Image Processing~" *TRICEPS*, EX41, 2010
- [6] 奥田正浩, "HDR 画像~色空間から符号化まで~" *映像情報メディア学会誌*, Vol.64, No.3, p.299-305, 2010
- [7] 奥田正浩, "高ダイナミックレンジ画像処理" *電子情報通信学会誌*, Vol.93, No.2, p.139-144, 2010
- [8] R. Xu, S. N. Pattanaik, and C. E. Hughes, "High-Dynamic-Range Still Image Encoding in JPEG2000" *IEEE Computer Graphics and Applications*, Vol.25, No.6, pp.57-64, 2005
- [9] M. iwahashi and H. kiya, "Efficient Lossless Bit Depth Scalable Coding for HDR Images" *APSIPA, ASC2012*, to be publised Dec.2012
- [10] D. J. Jobson, Z. Rahman, and G. A. Woodell, "The Statistics of Visual Representation" *Proc. SPIE*, 4736, pp.25-35, 2002
- [11] G. Deng, "An Entropy Interpretation of the Logarithmic Image Processing Model With Application to Contrast Enhancement" *IEEE Transaction on Image Processing*, Vol.18, No.5, pp.1135-1140, MAY.2009
- [12] A. Saleem, A. Beghdadi, and B. Boashash, "Image fusion-based contrast enhancement" *EURASIP Journal on Image and Video Processing*, 2012  
<http://jivp.eurasipjournals.com/content/2012/1/10>