

레터논문 (Letter Paper)

방송공학회논문지 제18권 제6호, 2013년 11월 (JBE Vol. 18, No. 6, November 2013)

<http://dx.doi.org/10.5909/JBE.2013.18.6.935>

ISSN 2287-9137 (Online) ISSN 1226-7953 (Print)

평균이동 분할계산기법을 사용한 역 컬러라이제이션 기반의 컬러영상압축

유태경^{a)}, 이석호^{b)†}

Color Image Compression based on Inverse Colorization with Meanshift Subdivision Calculation

Taekyung Ryu^{a)} and Suk-Ho Lee^{b)†}

요 약

본 레터논문에서는 컬러라이제이션(Colorization)기반 영상압축방법을 위해 평균이동기반의 영상분할법을 사용하여 컬러라이제이션 행렬을 분할하는 방법을 제안한다. 실험을 통해 제안한 방법을 사용할 경우 계산 속도는 대략 30배 이상 빨라지고, 기존의 역컬러라이제이션 기반의 압축방법에서 생기는 번짐(smearing) 현상도 많이 제거가 되는 것을 볼 수 있었다.

Abstract

In this letter, we propose a method for colorization based coding, which divides the colorization matrix into smaller sub-matrices using the meanshift segmentation. Using the proposed method the computation speed becomes more than 30 times faster. Furthermore, the smearing artifact, which appears in conventional colorization based compression method, is greatly reduced.

Keyword : Color image compression, colorization, meanshift segmentation, optimization.

1. 서 론

최근에 컬러라이제이션(Colorization)^{[1][2]}을 이용한 새로운 컬러영상 압축기법이 제안되었다^{[3]-[5]}. 이것은 컬러영상의 색도(色度)성분에 대한 압축방법이다. 즉, 휘도성분에 대

해서는 기존의 JPEG 또는 JPEG2000의 1-채널 압축방법 등으로 압축을 수행하고, 색도성분에 대해서는 역(逆)컬러라이제이션 (inverse colorization)기법을 사용하여 압축을 수행한다. 역컬러라이제이션 기법이란 컬러영상으로부터 나중에 전체 컬러를 재구성할 수 있는 대표픽셀(Represent-

a) 엔티엠 신사업본부(New Business Division, MTM Co.,LTD)

b) 동서대학교 컴퓨터정보공학부(Division of Computer and Information Engineering, Dongseo Univ.)

† Corresponding Author : 이석호(Suk-Ho Lee)

E-mail: petrasuk@gmail.com

Tel: +82-51-320-1744

※ 본 연구는 2012년도 동서대학교 교내 특별연구비 지원 및 정부(교육과학기술부)의 재원으로 한국연구재단의 지원 (No.2013R1A1A4A01007868)하에 수행되었음.

· Manuscript received August 29, 2013 Revised October 29, 2013 Accepted October 29, 2013

tative Pixels : RP)들을 추출하는 방법을 말한다. 인코더(Encoder)에서 RP를 추출한 후 이를 디코더(Decoder)로 보내면 디코더에서는 휘도성분과 RP를 사용하여 전체 컬러를 재구성하게 된다. 본 레터논문에서는 컬러리제이션 계산시의 계산량을 줄이기 위해 평균이동분할(Meanshift Segmentation)방법^[6]을 이용하는 알고리즘을 제안한다. 평균이동분할 방법을 사용하면 컬러라이제이션을 수행하는 행렬을 여러 개의 부(部)행렬들로 분할할 수 있게 되어 인코더에서 RP를 추출하는 계산 과정과 디코더에서 전체 컬러를 재구성하는 계산 과정을 여러 개의 작은 부행렬들에 대한 병렬계산으로 수행할 수 있어 전체적으로 인코딩과 디코딩 계산이 빨라지게 된다.

II. 선행연구

1. 컬러라이제이션(Colorization)

컬러라이제이션이란 흑백영상에 색상을 입히는 기술로 수동으로 몇 개의 픽셀들에 대한 색상정보를 주면 자동으로 모든 픽셀들의 색상정보가 결정이 되는 기술을 말한다^{[1][2]}. Levin은 최적화를 이용한 컬러라이제이션 방법을 제안하였다^[1]. 이는 다음과 같은 에너지항을 최소화함으로써 얻어진다.

$$J(u) = \sum_{r \in \Omega} \left(u(r) - \sum_{s \in N(r)} w_{rs} u(s) \right)^2 + \sum_{r \in \Omega} (u(r) - x(r))^2 \quad (1)$$

위의 에너지항은 Cb와 Cr성분에 대해서 각각 따로 존재한다. 여기서 Ω 은 색상정보가 주어진 영역을 말하며, r 은 현재 고려중인 픽셀을 의미하며, $N(r)$ 은 r 의 주변에 있는 픽셀들을 말하며 s 은 $N(r)$ 에 속한 픽셀이다. x 은 r 픽셀에 대한 함수이며 r 픽셀이 RP이면 $x(r)$ 은 그 픽셀에서의 색상성분값(Cb 또는 Cr값)을 갖고, RP가 아니면 0을 값으로 갖는다. u 는 구하고자 하는 해가 된다.

이때 가중치인 w_{rs} 은 휘도영상으로부터 계산되는데, 보통 $w_{rs} \propto e^{-\frac{(y(r) - y(s))^2}{\sigma^2}}$ 으로 놓는다. 여기서 y 는 휘도

영상이고, σ^2 은 스케일링을 위해 들어간 양수값이다. 식 (1)은 행렬식으로 표현이 된다. 즉,

$$J(u) = \|x - Au\|^2 \quad (2)$$

로 표현할 수 있다. 여기서 $A = I - W$ 인데, 여기서 A, I, W 는 모두 $N \times N$ 행렬이며, N 은 전체영상의 픽셀의 갯수를, I 는 단위행렬을 말하며, W 는 (r, s) 위치에 다음과 같이 정의되는 w'_{rs} 값을 가진 행렬이다:

$$w'_{rs} = \begin{cases} 0 & r \in \Omega \text{일 때} \\ w_{rs} & \text{그외의 경우} \end{cases} \quad (3)$$

식 (2)의 A 행렬은 역행렬이 존재한다는 것이 증명되었다^[1]. 그러므로 $J(u)$ 를 최소화하는 u 는 $u = A^{-1}x$ 에 의해 구할 수 있다.

2. 역컬러라이제이션을 이용한 컬러 영상압축

역컬러라이제이션을 이용한 컬러영상압축은 인코더에서 컬러라이제이션의 반대 과정인 역컬러라이제이션을 수행하는 압축방법이다^{[3][5]}. 인코더는 휘도성분을 기존의 압축 방법, 즉, JPEG이나 JPEG2000등으로 압축하고, 색상성분은 역컬러라이제이션을 수행하여 RP를 추출한다. 보통 RP의 갯수는 매우 적기 때문에 압축의 효과가 나타난다. 그러나 RP추출법은 최종의 RP집합이 나올 때까지 반복해서 행렬 A 의 역행렬을 구해야 하는데 A 가 크기 때문에 계산량이 많이 요구된다는데 문제가 있다. 또 행렬 A 의 구성을 잘못하게 되면 디코더에서 색상값을 재구성할 때 색이 섞이는 번짐현상이 발생한다. 그러므로 본 레터논문에서는 평균이동기반의 영상분할을 사용하여 A 행렬을 여러 개의 작은 부행렬들로 분할하여 계산하는 방법을 제안한다.

III. 제안하는 방법

평균이동기반 영상분할을 사용하여 전체 영상의 영역 S

를 $S = S_1 \cup S_2 \cup \dots \cup S_m \dots \cup S_M$ 으로 분할할 수 있게 된다. 제안한 방법에서는 이렇게 영상의 영역을 분할시킨 후에 각 분할된 영역에 대한 컬러라이제이션 부행렬을 구성하여 색상정보를 얻는 방법을 취한다. 즉, 분할된 영역 S_m 내에 속한 RP집합을 Ω_m 이라 할 때, 즉, $\Omega_m \subset S_m$ 일 때, 각 분할된 영역 S_m 에 대해 다음의 에너지항들, 즉 $J_1(u), J_2(u), \dots, J_M(u)$ 들을 최소화시킨다:

$$J_m(u) = \sum_{r \in R_m} \left(u(r) - \sum_{s \in N(r)} w_{rs} u(s) \right)^2 + \sum_{r \in U_m} (u(r) - x(r))^2 \quad (m = 1, 2, \dots, M) \quad (4)$$

$R_m = \{r \mid r \notin \Omega_m, r \in S_m\}$, $U_m = \{r \mid r \in \Omega_m, r \in S_m\}$ 이다. 위의 에너지항들을 최소화하여 얻게 되는 색상정보 $u(r)$ ($r \in \Omega_m, m = 1, 2, \dots, M$)가 원래영상의 색상정보를 잘 재현한 색상정보가 되게 하려면 Ω_m 의 RP들이 S_m 내에 있는 픽셀들에 대해서만 영향을 미치고, 그외의 다른 영역내의 픽셀들에 대해서는 영향을 미치지 않아야 한다. 그런데 평균이동기반의 분할에 의해 얻어지는 S_m 영역들은 색상기반의 분할법이면서 색상간의 경계선에서 확실하게 구분하면서 영역을 분할하기 때문에 이런 조건을 잘 만족한다.

식 (4)도 행렬식들로 표현할 수 있다. 즉, $u_m \in S_m$, $x_m \in R_m$ 일때 $x_m = A_m u_m$ ($m = 1..M$)이 성립한다. 즉, 평균이동기반의 영역분할에 의해 분할된 영역의 갯수만큼의 행렬식들로 표현이 된다. 여기서 부행렬 A_m 들은 식 (4)의 행렬 A 와 같은 방법으로 구성이 되는데, S_m 내에 있는 픽셀들과 Ω_m 의 RP들만 가지고 구성된다는 점이 다르다. 행렬 A 과 부행렬 A_m 들의 관계는 그림 2에 나타나 있다. 행렬 A_m 는 A 의 원소들 중 S_m 영역내에 있는 픽셀들의 색상값과 RP들간의 관계를 나타내는 원소들로만 구성이 된다.

A 행렬의 원소들 중 다른 영역에 있는 RP가 끼치는 영향을 나타내는 원소(그림 2의 A 행렬의 원소들 중 회색으로 표시된 부분)들은 분할되는 과정에서 제외가 되어 번짐 현상이 제거되어 색상의 재구성 성능이 좋아진다.

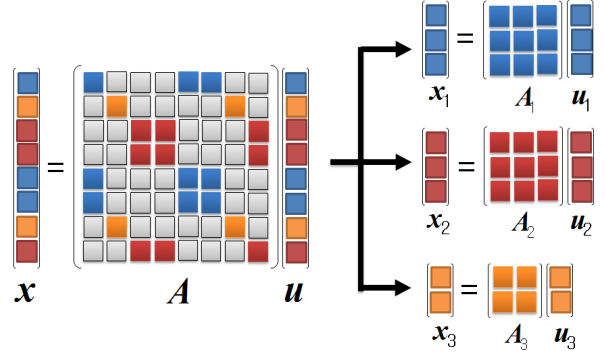


그림 2. 컬러라이제이션 행렬의 분할
 Fig. 2. Dividing the colorization matrix

그림 3은 제안한 방법을 컬러라이제이션에 사용한 전체 시스템도를 나타내고 있다.

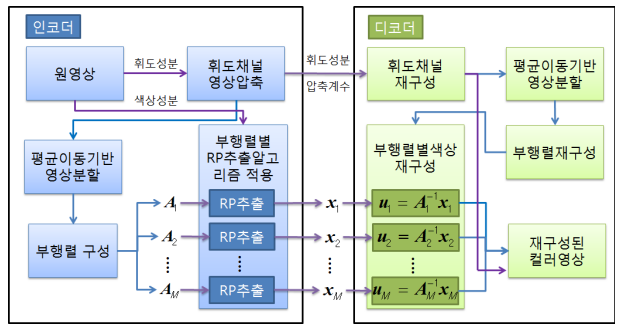


그림 3. 전체 시스템도
 Fig. 3. Overall system diagram

IV. 실험 결과

본 실험에서는 RP추출방법으로는 단순 랜덤 RP추출, 논문 [3]-[5]의 방법을 각각 사용하고, 컬러라이제이션 방법으로는 Levin의 방법과 제안한 분할계산법을 사용한 결과를 비교하였다. 컬러라이제이션의 방법으로 제안한 분할계산법을 사용할 경우 RP추출방법으로 어떤 방법을 사용하더라도 속도와 성능이 향상이 된다. 논문 [5]도 평균이동방법을 사용하고 있으나 RP를 추출하는 방법으로 사용하고 있다는 점에서 본 레터논문과 내용이 다르다. 그림 4는 각각의 방법에 대한 영상결과를 보여주고 있고, 표 1은 SSIM과

표 1. 제안한 방법과 Levin의 방법의 실험결과 정리
Table 1. Comparison results between the various methods

방법	RP추출법	컬러라이제이션	SSIM (Cb)	SSIM (Cr)	PSNR (dB)	색상압축률	계산속도 (sec)
방법1	Random	Levin's	0.7632	0.7722	21.5329	3.9%	320
방법2		Proposed	0.8101	0.8304	25.3252	3.9%	12
방법3	논문 [3]	Levin's	0.7834	0.7945	23.4942	3.2%	341
방법4	논문 [4]	Levin's	0.7755	0.7812	22.0401	3.2%	334
방법5	논문 [5]	Proposed	0.8232	0.8432	32.4352	2.8%	17
방법6	JPEG		0.8143	0.8323	33.2112	4.1%	6

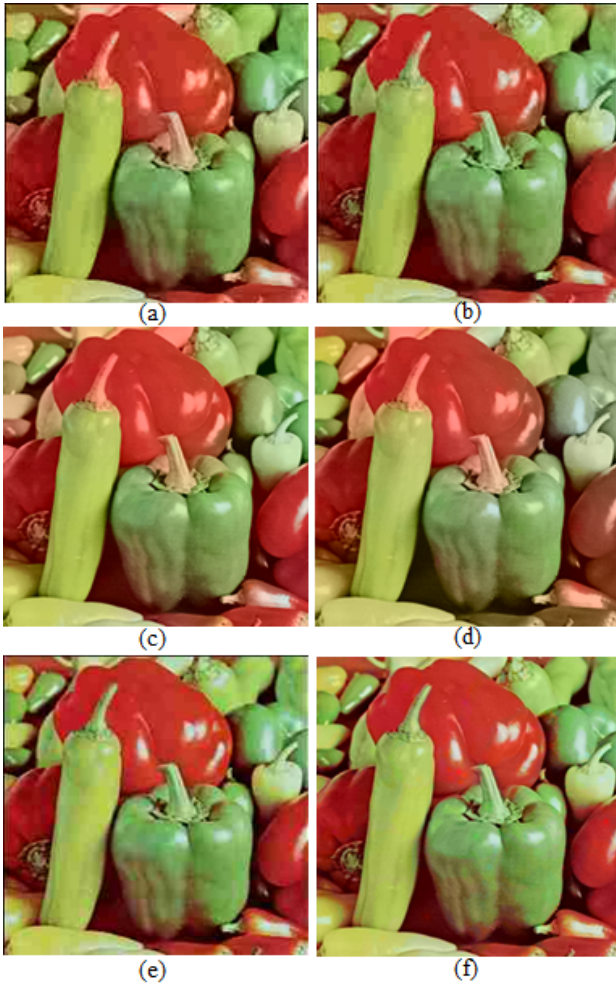


그림 4. 결과비교 : 그림 (a)~(f)은 각각 표 1의 방법 1~6의 결과
Fig. 4. Experimental Comparison

PSNR의 값을 비교하고 있다. 4:1:1포맷을 적용할 때 원색 상 정보량은 $128 \times 128 \times 2 = 32768$ byte (128×128 에 Cb,Cr을 1 byte씩 할당)가 된다. 방법1의 경우 RP를 319개 이므로 $319 \times 4 = 1276$ byte(319개의 RP에 대해 Cb,Cr값 1byte씩, x,y좌표 1byte씩 할당)되어 색상 압축률은 $1276/32768=3.9\%$ 가 된다. JPEG은 QP=19일 때량 (색상정보 1362byte) 비교하였다. 표 1의 압축률은 전체컬러영상의 압축률이 아니라 색상정보만의 압축률을 나타내고 있다.

제안한 방법을 사용할 경우 번짐 현상이 적어지고 속도가 개선이 되는 것을 확인할 수 있다. 계산량의 감소는 다음과 같다. 행의 갯수가 N 인 행렬의 역행렬의 계산량(덧셈과 곱셈의 수)은 가우스 소거법을 사용할 경우 대략 $\frac{1}{3}N^3$ 이 된다. 그러므로 만약 N 개의 픽셀 갯수를 가진 영상을 같은 수의 픽셀을 가진 M 개의 영역을 분할하였다고 가정하면 계산량은 $\frac{M}{3}(\frac{N}{M})^3 = \frac{1}{3}N^3(\frac{1}{M})^2$ 이 된다. 실제로 실험을 통해 알고리즘을 돌려본 결과 계산 속도는 약 30배 빨라졌다.

참 고 문 헌 (References)

- [1] A. Levin, D. Lischinski, and Y. Weiss, "Colorization using Optimization," ACM Transactions on Graphics, vol. 23, pp. 689-694, Aug. 2004.
- [2] L. Yatziv and G. Sapiro, "Fast image and video colorization using chrominance blending," IEEE Trans. Image Processing, vol. 15, no. 5, pp. 1120 - 1129, 2006.
- [3] L. Cheng and S.V.N. Vishwanathan, "Learning to Compress Images and Videos," Proc. ICML, vol. 227, pp. 161-168, 2007.
- [4] X. He, M. Ji, and H. Bao, "A Unified Active and Semisupervised Learning Framework for Image Compression ," IEEE CVPR, pp. 65-72, 2009.
- [5] Taekyung Ryu, Byung Gook Lee, and Suk-Ho Lee, "Image Compression System using Colorization and Meanshift Clustering Methods," to appear in Proc. CUTE 2013, Dec., Danang.
- [6] D. Comaniciu, and P. Meer, "Mean Shift : A Robust Approach toward Feature Space Analysis," IEEE Trans. PAMI, vol. 24, pp. 603-619, May 2002.