

3중첩 구간적 베지어 3차 곡선을 이용한 실사 영상의 컬러 보정에 관한 연구

권희용* · 이지영**

요 약

원 영상에 근접한 색채 재현을 위한 선형 변환을 이용한 영상의 컬러 보정은 컬러 공간의 비 선형성으로 인해 색의 왜곡 현상이라는 문제가 발생하게 된다. 이러한 문제를 극복하기 위해 선형 이론인 임의의 평면상에 주어진 자료점들로 구성되는 베지어 곡선이 사용되어 왔다. 그러나, 이 베지어 곡선은 자료점의 개수에 따라 차수가 증가하게 되므로 수치적 계산상의 많은 제약을 받게 된다.

본 논문에서는 각 구간에서나 전체 구간에서의 차수가 3차이면서 베지어 곡선의 특성을 갖는 "3중첩 구간적 베지어 3차 곡선"(TPBC Curve; Triplicated Piecewise Bezier Cubic Curve)를 이용하였다.

이에 따른, TPBC-곡선과 20차 베지어 곡선을 이용하였을 때와 비교하여 컬러 보정 시 발생하는 왜곡 현상, 그리고 좁은 영역의 컬러 보정으로 인한 작업량의 증가를 감소시킨 결과를 보여주고자 한다.

1. 서론

반도체 기술이 발전하면서 컴퓨터의 성능이 급속도로 향상되었고, 컴퓨터의 가격이 저렴하여 사회 전반에 컴퓨터 보급이 빠른 속도로 확산됨으로써 우리 사회의 많은 업무가 컴퓨터를 이용하여 처리될 수 있도록 사회구조가 개선되어 가고 있다. 따라서 화상처리에 대한 관심이 매우 높아졌으며 이 분야의 연구도 활기를 띠기 시작하여 부분적으로는 우리의 실생활에서 실용화되고 있다. 우리 실생활에서 가장 보편적으로 쉽게 접할 수 있는 컬러 모니터와 프린터와 같은 컬러 화상 주변장치들은 원 화상에 가까운 컬러를 재현하기 위해 컬러 변환 과정을 수행하

게 된다. 이러한 변환과정에서 각각의 컬러 화상 주변장치들의 기계적인 특성으로 인하여 원 화상에 가까운 컬러를 재현하지 못하는 경우가 대부분이다. 이러한 컬러 왜곡 현상은 컬러 화상 주변장치의 구성요소들의 비 선형적 특성에 기인한 것으로써 컬러 프린터의 경우 종지와 염료, 기계적 장치들의 특성에 비 선형성의 문제가 존재하고 컬러 스캐너의 CCD 소자들은 여러 파장들에 대한 감도에서 비 선형성의 문제를 나타내면서 원 화상과는 다른 컬러로 표현이 되어진다. 이러한 비 선형적 특성문제를 원 화상과 가까운 컬러 재현을 위하여 컬러 보정을 수행하게 되는데, 컬러 보정의 목적은 주어진 주변 장치들의 비 선형성에 기인하는 컬러 왜곡을 최소화하여 컬러 화상 주변장치 사이에서 컬러 재현의 일관성 실현을 위한 것으로써 하나의 화소가 색 좌표계에서 원하는 색 좌표값을 갖기 위해

* 세명대학교 전산정보 대학원 석사과정

** 세명대학교 소프트웨어학과 부교수

주어진 색 좌표계의 색 좌표값을 변환하는 일련의 과정인 것이다. 컬러 보정의 방법에는 ICC (International Color Consortium)에서 제안한 Profile Format을 비롯하여 Apple사의 ColorSync, HP의 ColorSmart, Tektronix의 TekColor 방식 등을 들 수 있다[1].

이러한 기존의 방법으로는 선형 변환 방식과 LUT(Look Up Table)을 이용한 보간 방식, 컬러 모델을 이용한 방식, 신경 회로망을 이용한 방식 등이 있는데, 선형 변환 방식은 계산상으로 간단하다는 장점 때문에 중 저급 영상 기기에 주로 쓰이는 컬러 보정 방식으로써 컬러 왜곡의 모델링과 컬러 보정의 두 단계를 수행하는 방식이다. 먼저 단계인 컬러 왜곡 모델링은 컬러 좌표와 출력 컬러 좌표간의 관계를 선형 변환으로 모델링 하게 되고, 컬러 보정은 주어진 컬러 좌표 값을 모델링 하는 과정에서 계산된 행렬을 이용하여 변환한 뒤 출력함으로써 작업이 이루어지게 된다. 선형 변환에서 이용할 수 있는 컬러 좌표계로는 (RGB), (CMYK), (L*a*b*), (XYZ), (L*u*v*)등 여러 좌표계를 사용할 수 있다. LUT(Look Up Table)을 이용한 방식은 대부분 고급 기종에서 사용되는데, 이 방식은 LUT의 구성과 보간에 의한 컬러 보정 과정의 두 단계로 이루어진다. 즉, 선형 변환에서와 같이 컬러 샘플들의 입력 컬러 좌표값과 출력 컬러 좌표값의 순서쌍으로 LUT를 구성한다. 이때 컬러 좌표값이 LUT상에 존재하는지 여부에 따라 LUT상의 값들을 보간 함으로써 보정된 컬러 좌표 값을 얻게 된다. LUT을 이용한 방법의 특징은 상대적으로 선형 변환 방식에 비하여 보간 과정에서 많은 시간이 걸리는 단점이 있으나 선형 변환 방식보다는 보정 후 감소된 색차를 얻을 수 있다. 컬러 모델을 이용한 방식은 컬러 왜곡 과정을 주어진 영상 기기를 구성하는 요소

들의 광학적 특성을 고려한 수식으로 모델링 함으로써 컬러 보정을 수행하는 방식이다. Adobe사의 경우 베지어 블렌딩 함수를 이용한 선형 변환 방식을 이용한 컬러 보정을 사용한다. 베지어 곡선을 이용한 컬러 보정 시 낮은 차수를 사용하면 불필요한 컬러가 보정되는 왜곡 현상이 발생하고, 반대로 너무 높은 차수를 사용하면 로컬한 영역의 컬러를 보정하여 다단계의 보정이 필요하다는 문제가 뒤따른다[2]. 따라서 본 논문에서는 대부분 응용분야에서 일반화되어 있는 3차 베지어 곡선을 적용한 3중첩 구간적 베지어 3차 곡선(TPBC-곡선 : Triplicated Piecewise Bezier Cubic Curve)을 이용하여 컬러 왜곡 현상을 최소화하고 원하는 영역만을 보정하여 기존 방법보다 감소된 평균 색차를 얻기 위한 실험의 결과들을 통해 비교하여 보여주고자 한다.

II. 컬러 모델 이론

색을 지정하는 가장 정확한 방법은 모든 색에 대해 주파수 대역 그래프를 사용하는 것이겠지만, 실제로 이런 방법을 사용하는 것은 사실상 불가능하다. 그래서 특정한 상황에서 색상을 설명하는 다양한 방법이 개발되었는데, 이를 컬러 모델(Color Model)이라 한다.

컬러 모델은 컬러들과 다른 컬러들과의 관계를 표현하는 방법으로써 컬러 화상 주변 장치는 서로 다른 이유 때문에 서로 다른 컬러 모델을 사용한다. 예를 들면 TV와 컴퓨터 모니터를 포함해 거의 모든 디스플레이 장치는 픽셀을 이용하여 빛을 가산하는 RGB 컬러 모델에 근거해 만들어지고, 종이와 같은 매개물을 이용하는 프린터는 감색 색체계인 CMY 컬러 모델을 사용

하게 된다. 이 경우 RGB 컬러 모델의 입력으로부터 CMY 컬러 모델의 출력으로 변환을 수행하게 되는데 이때 컬러 왜곡 현상이 발생하게 된다. 이러한 주변 장치들의 서로 다른 컬러 영역 사용으로 인해 발생하는 컬러 왜곡 현상은 CMS(Color Management System)으로 최소화시키게 된다. 인간의 컬러 지각 능력은 컬러의 3속성인 색상, 명도, 채도의 3가지 속성에 의해 반응한다. 이것을 수치적으로 표현한 것을 삼중 자극 값이라 부르며 삼중 자극 값을 기반으로 하여 컬러 공간을 구성하게 되는 것이다[4]. 이렇게 컬러 재현을 위해 사용되는 매체와 방법에는 여러 방법이 있으나, 컬러를 재현하는 방식은 가색 색체계(RGB)와 감색 색체계(CMYK)만이 존재한다[3].

2.1 RGB Color Model

RGB 모델은 빛의 삼원색인 Red, Green, Blue이 기본색이 되는 컬러 모델로서 모니터에 기초하며, 여러 색의 빛을 더하면 흰색이 되는 빛의 성질을 이용한 것으로 기본 색을 더하여 새로운 컬러를 만들어내므로 가색 색체계라 불리운다[4].

이 모델에서는 RGB 각 값의 범위를 0에서 1까지의 소수를 사용하기도 하지만 실제 대부분의 그래픽 소프트웨어에서는 0에서 255까지의 값을 사용한다.

모니터에서는 픽셀의 형태로 빛을 만들게 되는데 픽셀의 색상이 바로 RGB 컬러 모델을 이용하여 빛을 만들게 된다. 그러나 실제 Drawing 프로그램에서는 RGB 컬러 공간보다는 출력에 기초한 공간인 CMY 영역을 사용하게 되는데 이것은 Drawing 프로그램에서 실제 작업한 영상의 색상과 출력 영상의 색상이 차이가 나기 때문이다.

2.2 CMY(CMYK) Color Model

CMY 컬러 공간은 Cyan, Magenta, Yellow로 구성된다. 이것은 RGB 컬러 공간과 반대의 공간이며, Cyan, Magenta, Yellow는 Red, Green, Blue와 각각 보색이다. CMY 컬러 모델은 흰 종이(255)로부터 잉크의 색을 감산하여 색을 표현하기 때문에 감산 색체계라고 한다. 재현 방식은 흰색으로부터 Cyan, Magenta, Yellow를 감산하여 서로 다른 색을 재현하는 것이다. 즉 Cyan은 Red를 흡수하고, Magenta는 Green, Yellow는 Blue를 흡수하여 색을 재현하게 된다. RGB와 CMY는 보색 관계에 있기 때문에 두 공간 사이의 변환은 다음 식(1)과 같이 흰색에 대한 보수를 취한다.

$$\begin{aligned} C &= 1.0 - R \\ M &= 1.0 - G \\ Y &= 1.0 - B \end{aligned} \quad (1)$$

이 수식과 컬러 공간들은 평균화되어 있으며, 모든 값들은 0.0과 1.0 사이의 값이다. 프린팅 업계에서 이 CMY 모형에 네 번째 컬러를 추가하였는데, Cyan, Magenta, Yellow에 검정색을 더한 색깔이 처리될 수 있는 컬러들로 알려져 있고, 이 컬러 모형을 CMYK라고 부른다. 검정(Black, K)은 다른 세 가지 컬러들의 조합에 의해 만들어지는 것보다 순수 검정색이 더욱 좋기 때문에 프린팅 처리를 할 때 사용된다. 순수한 검정색은 뛰어난 대비를 제공한다. 또한 검정 잉크가 컬러 잉크보다 더 비용이 적게 드는 추가적인 요소이다[5].

CMY에서 CMYK로 변환은 식(2)와 같다.

$$K = \min(C, M, Y)$$

$$\begin{aligned} C &= 1.0 - R \\ M &= 1.0 - G \\ Y &= 1.0 - B \end{aligned} \quad (2)$$

위의 변환 공식을 사용할 경우에도 색의 왜곡 현상이 발생하게 되는데, 포토샵의 경우 예를 들면 CMYK모드에서 C : 100! 은 Cyan 영역으로 재현 할 수 없는 색상을 뜻한다[6].

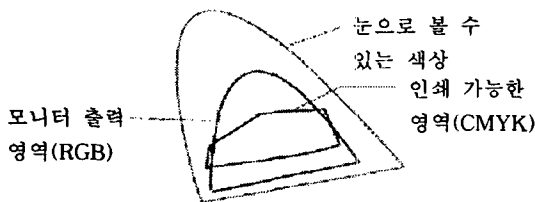


그림 1. 색 공간 모델

그림 1을 보면 모니터 출력 영역인 RGB에 비해 인쇄 가능한 영역인 CMYK가 좁은 것을 알 수 있다. 따라서 CMYK 영역 외부의 색상에 대해 CMYK 영역으로의 매핑이 필요하다는 것을 알 수 있다.

III. 3중첩 구간적 베지어 3차 곡선

2차원 평면상에 있는 임의의 점들을 정해진 순서에 따라 부드럽게 연결하면 하나의 곡선을 얻을 수 있는데 이렇게 임의의 완전히 자유로운 형태의 곡선을 그리기 위한 여러 방법 중 가장 널리 사용하는 도구로는, 수학적 특성이 강한 스플라인 곡선과 수학적이라기보다는 공학적 의

미에 기반을 두고 있는 베지어 곡선이 대표적이다. 주어진 n개의 자료점들로부터 스플라인 곡선을 구성하기 위해서는 적어도 n개의 선형 연립방정식을 풀어야 하는데, 이를 컴퓨터에 실현시키는 데에는 많은 연산과 기억용량이 요구되고, 대량의 누적 계산으로 인한 오차가 크게 작용하게 된다. 그리고 베지어 곡선 구성에서, 자료점이 많을 경우 그에 비례하여 베지어 다항식의 차수가 높아지게 되는데, 차수가 높은 다항식은 매끄러운 곡선을 표현할 수는 있지만, 컴퓨터 수 체계의 유한성 때문에 계산상의 제약을 받기 때문에 대부분의 응용분야에서는 사실상 3차 곡선을 사용하는 방법이 일반화되어 있다[7]. TPBC-곡선은 이러한 스플라인 곡선과 베지어 곡선들의 계산상의 어려움을 극소화하면서 부드러운 곡선을 얻을 수 있다.

3.1 Bezier Curve

J. C. Ferguson은 1963년 항공기 설계에서 곡선과 곡면의 정의를 위해 매개변수 3차 방정식을 처음으로 사용하였고, 1970년 Bezier는 더 명확하게 벡터 계수들이 실제 의미를 갖는 방법으로 Ferguson 곡선형태를 재구성하였다. 즉, 많은 선형 근사 이론의 응용분야에서 평면이나 공간상에 있어 단일 가중치 함수인 스플라인 곡선으로는 임의의 곡선을 일반적으로 구성할 수 없으므로 선형 근사 이론의 결과들을 임의의 곡선 구성에 적용시키기 위해서는 x, y축 각각을 독립적으로 취급하는 것이다. 이 방법이 Bezier에 의해 처음으로 제안되었다.

베지어 곡선은 n+1개의 제어점이 입력되고 이것을 벡터 $P_k=(x_k, y_k)$, $k=0,1, \dots, n$ 으로 표시한다.

Bezier 좌표 함수는

$$F(u) = \sum_{k=0}^n P_k B_{k,n}(u) \quad (3)$$

$$B_{k,n}(u) = C(n, k) u^k (1-u)^{n-k} \quad (4)$$

$$C(n, k) = \frac{n!}{k!(n-k)!} \quad (5)$$

위와 같이 식 (3),(4),(5)와 같이 표현된다.

(3),(4),(5)의 식을 통합하면 다음 식 (6)과 같다.

$$B(u) = \sum_{k=0}^n P_k \frac{n!}{k!(n-k)!} u^k (1-u)^{n-k} \quad (0 \leq u \leq 1) \quad (6)$$

$B(u) = n$ 차 제어점 P_k 로 정의된 이항 함수이다. 계수 u 는 첫 제어점 $k=0$ 일 때 $u=0$ 이고 마지막 제어점 $k=n$ 일 때 $u=1$ 이 된다.

식 (6)은 x , y 축 각각의 매개 변수 방정식들의 집합으로 양 함수로 재 정의되어 식 (7),(8)과 같다.

$$x(u) = \sum_{k=0}^n x_k B_{k,n}(u) \quad (7)$$

$$y(u) = \sum_{k=0}^n y_k B_{k,n}(u) \quad (8)$$

(7),(8) 식에 의해 Bezier 곡선선분의 픽셀 좌표를 얻을 수 있게 된다. 베지어 곡선의 특징은 첫 제어점과 마지막 제어점을 제외하고 어떠한 제어점도 통과하지 않는다. 이러한 베지어 곡선의 특징 때문에 색상 보정에서는 출력 지향 벡터가 생성 되게 된다. 곡선은 항상 제어점 내부에 포함된다. 이것은 색상을 보정하였을 경우 인접 색상의 보정도 보장하게 된다. (4)번식의 $B_{k,n}(u)$ 를 베지어 곡선의 제어점을 연결하는

블렌딩 함수라 부르는데 이것은 각 제어점 사이의 이항계수를 이용하여 곡선을 생성시키게 된다. 베지어 곡선의 제어점을 한 위치에 여러 개를 두면 곡선을 그 위치로 지향시킬 수 있다. 이것은 컬러 보정에서 비 선형 공간에 존재하는 로컬 영역의 색상 보정을 가능하게 해준다. 적은 계산량으로 곡선의 국부적인 변형을 이루기가 쉽다. 스플라인 곡선의 계산량이 비해 적은 계산량으로 국부적인 변형을 이룰 수 있기 때문에 색상 보정을 위한 곡선의 차 수를 높일 수 있게 된다[8].

3.2 Triplicated Piecewise Bezier Cubic-Curve

TPBC-곡선은 주어진 자료 점들로부터 임의의 두 자료점 사이에서 3중복이 되는 3차 베지어 곡선들의 선형결합으로 구성하는 곡선이다.

일반적으로 매개변수 함수는 각 점간의 직선 거리를 가상적인 곡선 거리로 매개 변수화 함으로써 구성된다. 따라서 주어진 점들의 집합에 각 점들의 순서가 정해져 있다고 가정할 때, 매개변수는 다음과 같이 구성할 수 있다. 먼저, 각 점간의 직선거리를 구한 다음, 각 점간의 직선 거리를 그들의 총합으로 나누어 정규화시켜, 0과 1 사이의 값으로 매개 변수화 한다[9,10].

매개변수 t 와 그에 대응되는 점의 x , y 좌표와의 관계식을 구성한 매개변수 다항식을 각각 $x(t)$, $y(t)$ 라 하면 얻고자 하는 함수는 $(x(t), y(t))$ 로 표현되며, 0과 1 사이의 모든 t 값에 대응되는 점을 구하여 연결함으로써 자유로운 형태의 부드러운 곡선을 그릴 수 있다. 따라서 이러한 매개변수화 과정을 거쳐 구성된 다항식은 매개변수의 값이 두 점간의 직선거리

의하여 결정되기 때문에 다양한 형태의 곡선을 구성할 수 있다.

TPBC-곡선을 구성하기 위해 전체 구간 $I = [t_1, t_n]$ 을 $n-3$ 개의 3중침 부분 구간

$I_j = [t_j, t_{j+3}]$, ($j=1,2,\dots, n-3$)를 정의하자.

4개의 점 $[P_1, P_2, P_3, P_4]$ 이 주어졌을 때 3차 Bezier 곡선은

$$Q(t) = (1-t)^3 P_1 + 3t(1-t)^2 P_2 + 3t^2(1-t) P_3 + t^3 P_4 \quad (t \in [0, 1]) \quad (9)$$

식 (9)와 같이 된다. 따라서 구간 I 의 부분 구간 I_j 에서의 3차 베지어 곡선은 식 (10)와 같이 나타내어진다.

$$Q_j(t) = \begin{cases} \frac{1}{(t_{j+3}-t_j)^3} ((t_{j+3}-t)^3 P_j + 3(t-t_j)(t_{j+3}-t)^2 P_{j+1} \\ + 3(t-t_j)^2(t_{j+3}-t) P_{j+2} + (t-t_j)^3 P_{j+3}) \\ t \in I_j = [t_j, t_{j+3}], j=1, 2, \dots, n-3 \\ 0, t \in I_j \end{cases} \quad (10)$$

여기서 P 는 주어진 자료점이고, P 는 조정된 점이다. 조정된 점이란 t_j 에 대해서 t_{j-1}, t_j, t_{j+1} 이 동일선상에 있도록 만든 점이다.

또한, 전체구간 I 에서의 TPBC-곡선 함수 $T(t)$ 는 위에서 설명한 $Q_{j-2}(t), Q_{j-1}(t)$ 와 $Q_j(t)$ 에 의하여 다음과 같은 일반적인 형태의 함수식으로 다음과 같이 식 (11)로 나타내질 수 있다.

$$T_j(t) = \frac{1}{3} (Q_{j-2}(t) + Q_{j-1}(t) + Q_j(t)) \quad (11)$$

(단, j 는 곡선의 형태에 따라 다르다.)

IV. TPBC 곡선을 이용한 컬러 보정

컬러 보정을 위한 TPBC 함수는 3중침된 각각의 구간을 32의 컬러 공간으로 표현하여 계수를 1/32인 0.03125로 설정하였다. 그리고 베지어 블렌딩 함수를 사용하여 x, y 좌표를 얻게 되며, x, y 좌표는 각각 256개의 공간이 설정되고 그림 2의 10라인에 의해 중침된 구간에서의 평균값으로서 좌표값을 보여주게 된다. 이 좌표값이 컬러 인덱스 값으로 변환되어 보정 시에는 x 의 인덱스 값을 이용하여 y 값의 인덱스를 보정 컬러로 출력하게 된다.

```

1 for i = 0 to 구간 I0 count step 3 // 첫 번째 구간
2 for u = 0 to 1 step 0.03125 // 1/32개의 컬러공간
3 x[0]좌표 = x[0]좌표 + B(u)
4 for i = 0 to 구간 I1 count step 3 // 두 번째 구간
5 for u = 0 to 1 step 0.03125
6 x[1]좌표 = x[1]좌표 + B(u)
7 for i = 0 to 구간 I2 count step 3 // 세 번째 구간
8 for u = 0 to 1 step 0.03125
9 x[2]좌표 = x[2]좌표 + B(u)
10 new[x좌표] = 1/3 (x[0] + x[1] + x[2]) // 3중침 구간
11 RGB(x) = RGB(y) // 보정 색상
    
```

그림 2. TPBC 함수 처리

4.1 컬러 변환 등화기

컬러 변환 등화기는 컬러 보정 시스템의 컬러 보정의 한 단계로써 TPBC 함수를 이용하여 원 영상을 보정하고 목적 영상을 얻게 되는 일련의 과정으로 컬러 왜곡 모델링과 컬러 보정처리를 수행하게 된다.

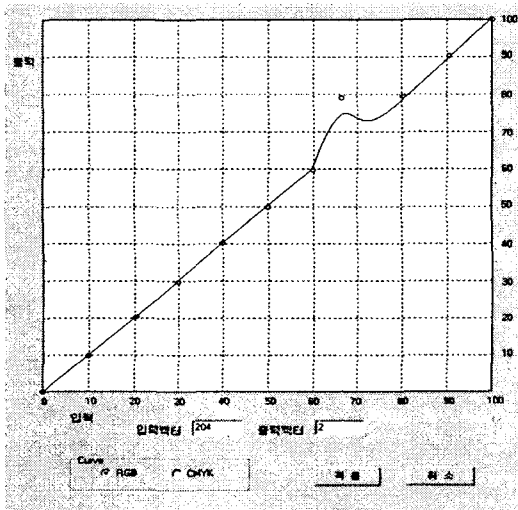


그림 3. 컬러 왜곡 모델링

4.1.1 컬러 왜곡 모델링

입력 컬러를 목적 컬러로 동일시하여 모델링 하는 초기화 과정을 컬러 왜곡 모델링이라 하는데, 그림 3과 같이 선분공간은 컬러 매핑을 위한 선형 공간으로 제어점인 원에 의해 컬러의 보정이 가능하게 되며, 선분공간에서 TPBC 곡선의 제어점들을 선형공간에 위치시키고 정렬함으로써 제어점 추가 시 발생하는 곡선의 변화를 없애고 입력 컬러와 출력 컬러를 1:1로 매핑하게 된다. 즉 X축은 입력 벡터로 RGB 컬러의 색상 공간인 0-255사이의 공간을, Y축은 출력 벡터인 왜곡 RGB 컬러로 입력 RGB칼라와 동일한 Value를 가지게 된다.

RGB 컬러 공간은 0-255사이인 256개의 공간이 존재한다. 이에 따라서 9개의 제어점을 기준으로 TPBC 곡선의 3중첩되는 8개의 각각의 한 구간에 32개의 컬러 공간이 존재하여 증가치 계수 u 를 0.03125(1/32)로 하고 첫 제어점의 곡선 계수인 u_0 와 마지막 제어점의 곡선 계수인 u_1 사이의 곡선 계수 좌표를 각각 직선으로 연결함으로써 컬러 왜곡 모델링의 선분을 생성시킬 수 있게 된다.

4.1.2 컬러 보정 모델링

컬러 보정 모델링은 실제 컬러를 보정하는 단계로, 본 논문에서 제시한 TPBC 함수를 사용하게 된다. 컬러 보정 시 사용하는 Bezier 함수의 차수가 낮게 되면 상대적으로 속도는 빠르지만 주변의 곡선에 영향을 미치게 되어 곡선의 모든 선분이 이동하게 된다. 이것은 컬러의 매핑이 곡선의 선분에서 이루어지는 것을 감안하였을 경우 모든 벡터 즉 모든 컬러에 대해 왜곡 현상이 발생하게 된다. 반면에 차수가 상대적으로 높게 되면 곡선의 선분이 로컬 영역에서만 이동이 되어 편집된 제어점 주변의 일부 벡터에 대해서만 컬러의 보정이 이루어지게 된다. 컬러의 보정은 특정 컬러를 보정하였을 경우 그 컬러의 주변에 있는 컬러도 보정이 되어야 한다. 그러나 차수가 너무 높게 되면 이것이 로컬 영역으로만 한정되기 때문에 컬러 보정 시 원하고자 하는 컬러를 보정할 수 있지만 영역의 보정은 보장할 수 없게 되므로 컬러의 보정을 위해서는 적절한 차수가 중요한 영향을 끼치게 된다. 이에 본 논문에서는 컬러의 왜곡 현상을 최소화시키고 적절한 영역의 컬러 보정을 위해 낮은 차수인 3차의 베지어 함수이지만 한 구간에 대해 3중첩을 시킴으로써 보다 높은 차수의 효과를 낼 수 있는 TPBC 함수를 사용하여 컬러

보정을 수행하였다.

제안된 컬러 보정 모델링은 9개의 제어점을 이용하여 컬러 변환 공간을 재정의 하게 되고 입력 색상의 벡터값을 x 축으로 하여 선분 픽셀 좌표의 y 축을 보정 컬러로 출력하게 된다.

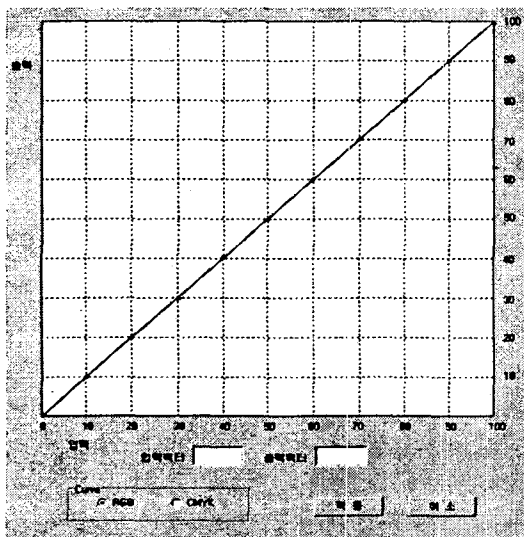


그림 4. 컬러 보정 모델링

그림 4의 컬러 보정 모델링은 11개의 제어점 중 7번째 제어점을 입력 좌표 70%에서 출력 좌표 80%로 변경한 것으로 주변의 곡선에 영향을 최소한으로 줄인 것을 보여준다.

거의 모든 영상은 고주파 특징보다는 저주파 특징이 주로 나타나게 된다. 즉 영상은 윤곽선과 같은 색상의 갑작스런 변화보다는 얼굴과 같이 명암이 부드럽게 변하는 색상의 분포가 많게 된다. 따라서 그림 4와 같이 벡터의 변화가 생겼을 경우 인접 벡터에 영향을 어느 정도 가함으로써 색의 부드러운 변화를 보장할 수 있게 된다. 영상은 고주파 성분의 픽셀을 지나면서 또한 다른 명암도의 영상을 가지게 된다. 즉 거의 모든 영상은 윤곽선을 지나면서 다른 명암도

를 가지게 된다. 이러한 경우 베지어 블렌딩 함수의 차수가 작게되면 매핑 함수인 곡선 선분이 거의 모두 이동하게 되어 컬러의 보정이 넓게 이루어지게 된다.

따라서 벡터의 변화가 생겼을 경우 곡선의 변화가 주변의 곡선에 영향을 최소한으로 줄일 경우 컬러의 왜곡 현상을 줄일 수 있게 된다. 이에 본 논문에서는 저 차수인 3차 베지어 함수이지만 한 구간에서 3중복을 시킨 TPBC-곡선을 이용하여 보정 편차 영상과 히스토그램의 분석을 통하여 20차 베지어 함수를 사용하였을 때와 비교하여 보다 나은 효과를 보여주고자 한다.

V. 실험 및 결과

5.1 시스템 개발 환경

〈표 1〉 시스템 개발 환경 사양

구분	종류	사양
H/W	Processor	Pentium-III 600
	Memory	256M
	VGA Card	NVidia Vanta
	VGA Memory	32M
	Monitor	SyncMaster 17GLSi
S/W	Language	Borland Delphi 6.0

본 연구는 멀티 미디어 관련 분야로 멀티 미디어 개발에 용이한 Delphi 6.0을 이용하여 개발하였다. 전체적인 하드웨어와 소프트웨어의 시스템 환경은 위의 표 1과 같다.

5.2 컬러 보정 처리 흐름도

영상의 출력은 LUT을 통과하여 LUT에 저장되어 있는 Value를 컬러의 영상의 값으로 보정

영상을 출력 하게된다. LUT은 색상의 정보를 저장하고 있는 버퍼로써 컬러 왜곡 모델링을 통해 영상을 출력 할 경우 Value 값은 원 영상 그대로인 0-255사이의 값을 가지게 된다. 즉 0의 인덱스에 대해 0의 값을 갖고 128의 인덱스에 대해 128의 값을 갖게 된다. 따라서 영상에서 읽어온 색상의 정보를 LUT의 인덱스로 취급하여 LUT의 값을 컬러 값으로 System Palette에 출력하게된다. 반면 컬러 보정 모델링을 통과하였을 경우 LUT의 내용은 보정된 컬러를 Value 값으로 갖게되는데 예를 들어 128의 색상 정보를 130으로 보정하였을 경우 LUT은 0의 인덱스에 0의 값을, 128의 인덱스에 130의 값을 갖게 된다. 따라서 영상의 색상은 128의 색상을 명암도 130으로 변화하여 출력하게된다.

5.3 컬러 보정 처리 결과

그림 6은 256 by 256의 24비트 True 컬러의 Lenna 영상으로 Red, Green, Blue 각각을 보정 처리한 결과를 보여주게 된다.



그림 6. Lenna 원 영상

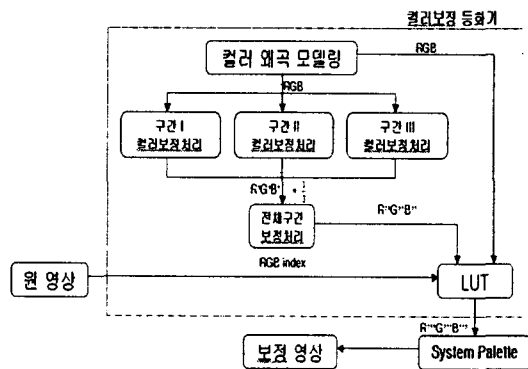


그림 5. 컬러 보정 흐름도

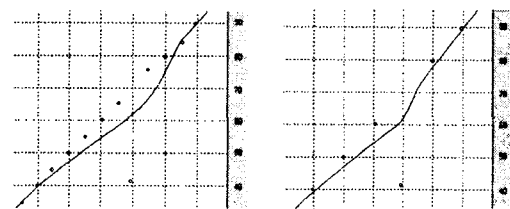
따라서 보정되지 않은 영상은 컬러 왜곡 모델링의 RGB값은 LUT에 Value값으로 저장되게 되고 보정 영상은 매핑된 R''G''B'''를 Value로 하여 LUT에 저장되게 된다. 원 영상의 RGB 값은 LUT의 인덱스로 저장되어 있는 Value를 실 RGB로 취급하여 시스템 팔레트에 출력하게 되고 이것이 보정 영상으로 출력되게 된다.

5.3.1 Red 보정 결과

Red 영상은 색상 명암도 179을 160으로 보정 하였을 경우의 결과는 다음과 같다.

그림 7의 곡선으로 컬러의 보정 효과를 보면 (a)는 5블럭, (b)는 4블럭 정도의 보정 효과를 보여주고 있다.

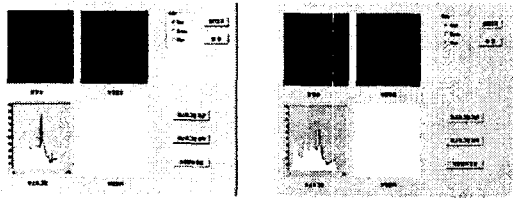
(a)는 출력 지향 벡터의 경우 40% 구간에서 설정을 하게 되는데, 이것은 컬러의 극심한 변화를 보정할 경우 보정의 어려움을 초래하게된다.



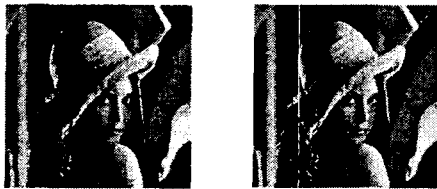
(a) 20차 Bezier 곡선 (b) TPBC-곡선

그림 7. Red 컬러 보정 모델

그림 8의 히스토그램을 보면 그림 (a) 와 (b)의 경우 컬러 보정의 구간은 약 30% 구간으로 거의 차이가 없다. 그러나 (a)의 경우에는 특정 색상에 매우 많은 분포를 가지고 있는데, 이것은 많은 제어점의 영향으로 곡선의 기울기가 커지기 때문이다. 따라서 특정 출력 백터에 대해 색상이 지나치게 집중되는 현상으로 색상의 부드러운 변화를 가지지 못하는 현상을 가져온다.



(a) 20차 보정 (b) TPBC 보정
그림 8. Red의 히스토그램과 보정 편차 영상



(a) 20차 보정 (b) TPBC 보정
그림 9. Red 보정 영상 결과

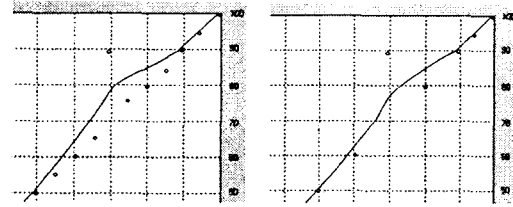
그림 9의 Red 보정 영상 결과를 보면 (a)의 경우 특정 화소에 대해서만 명암도가 낮아졌고, (b)의 경우 영상이 전체적으로 명암도가 약간 낮아진 것을 볼 수가 있다.

5.3.2 Green 보정 결과

Green 영상은 색상 명암도 170을 180으로 보정하였을 경우의 결과는 그림 10과 같다. 그림 10의 곡선으로 컬러의 보정효과를 보면 (a)는 4블럭, (b)는 3블럭 반정도의 보정 효과를 보여주

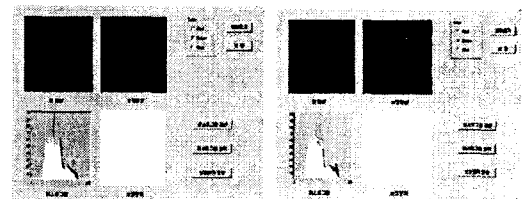
고 있다.

색상의 미세한 보정의 경우 색상의 차이가 거의 존재하지 않기 때문에 제어점의 개수가 많을수록 즉 차수가 높을수록 보정 효과가 뛰어난 것을 보여주고 있으며, TPBC 곡선을 이용한 보정 역시 차수는 낮지만 비슷한 효과를 나타내고 있다.



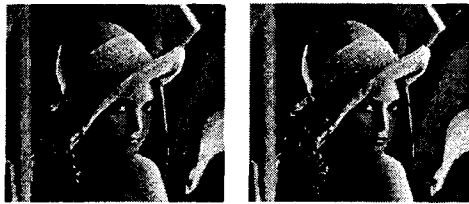
(a) 20차 Bezier 곡선 (b) TPBC-곡선
그림 10. Green 컬러 보정 모델링

그림 11의 (a) 번은 색상 편차가 크지 않음에도 불구하고 베지어 곡선의 기울기가 완만하기 때문에 근접한 색상 정보를 모두 보정하여 보정 편차 영상의 명암도가 높게 나타난다. (b)의 경우에는 색상의 왜곡 현상이 발생하게 되지만 명암도는 낮은 편으로 나타나게 된다. 히스토그램에서 볼 수 있듯이 특정 색상만을 보정하였기 때문에 특정 색상의 분포가 높게 나타나게 되고 근접한 색상 정보에는 거의 영향을 주지 않았다.



(a) 20차 보정 (b) TPBC 보정
그림 11. Green의 히스토그램과 보정 편차 영상

그림 12의 보정 영상 결과를 보면 (a)의 경우는 높은 차수로 인한 로컬 영역의 보정이 이루어져 원 영상과 차이가 없음을 알 수 있다. (b)의 경우는 육안으로는 구별하기 힘든 정도의 미세한 왜곡현상이 발생하였다.



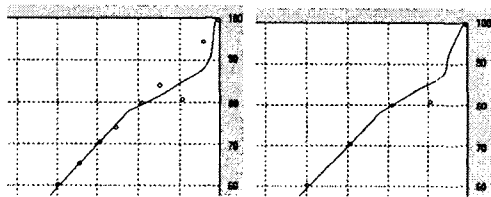
(a) 20차 보정 (b) TPBC 보정

그림 12. Green 보정 영상 결과

5.3.3 Blue 보정 결과

Blue 영상은 색상 명암도 247을 230으로 보정하였을 경우의 결과는 그림 13과 같다.

그림 13의 곡선으로 각 곡선별 보정 효과를 보면 (a)는 2 블록 (b)는 한 블록 반에 대해 보정이 이루어지고 있다.

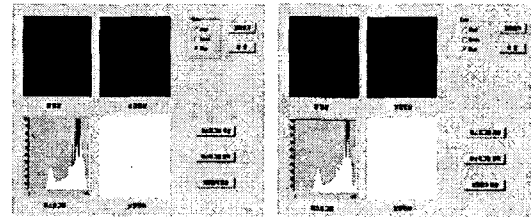


(a) 20차 보정 (b) TPBC 보정

그림 13. Blue 컬러 보정 모델링

그림 14의 (a)번은 베지어의 차수가 너무 높게 되어 원하는 컬러 정보만을 보정하는 효과를 얻기는 하지만 좁은 지역에서의 컬러 보정으로 인해 주변의 색상 정보는 보정을 수행하지 않기 때문에 고주파 성분, 즉 윤곽선만을 보정하는 효과를 보여주고 있다. 반면 (b)번은 히스토그램

의 Red 분포가 전체의 약 30% 구간을 보정하는 효과를 얻게 된다. 이것은 보정되는 컬러의 이웃한 색상도 보정되는 효과를 가지게 된다.



(a) 20차 보정 (b) TPBC 보정

그림 14. Blue의 히스토그램과 보정 편차 영상

그림 15는 실제의 영상 결과로 (a)번의 경우는 특정 화소에 대해서만 색의 보정이 이루어졌기 때문에 거의 변화가 생기지 않는다. (b)번의 경우 색의 보정이 어느 정도 이루어 졌다고 볼 수 있다.



(a) 20차 보정 (b) TPBC 보정

그림 15. Blue 보정 영상 결과

VI. 결론 및 추후 연구

컬러 모니터와 프린터와 같은 컬러 화상 주변 장치들은 원 화상에 가까운 색채를 재현하기 위해 컬러 변환 과정을 수행하게 되는데, 컬러의 변화 없이 원 화상을 얻어내기 위해서는 컬러

보정 작업은 필수적이며 매우 중요한 요소이다. 그러나 이 경우 컬러 보정 작업은 컬러 화상 주변장치의 색상공간의 배치형태가 비 선형적으로 상이하기 때문에 컬러 정보의 완벽한 전달이 이루어지지 못하는 문제점이 있다. 즉, 입력장치의 색상형태는 대부분 RGB 모드로 구현되고 있고, 출력장치의 색상 형태는 대부분 CMYK모드로 구현되어 있어서 이들간의 만족할 만한 매핑이 이루어지지 않고 있다. 이 경우 색 보정을 위해서는 각종 다양한 방법의 선형 변환방법이 이용되고 있지만 본 논문에서는 3중첩 구간적 베지어 3차 곡선을 이용하여 불필요한 변환요소의 결과를 최소화하고 특정영역의 색상정보를 효과적으로 제어할 수 있는 방법으로 20차 베지어 곡선을 이용했을 때와 비교하여 제안하였다. 한 구간에 대해 TPBC-곡선은 주어진 자료점들로부터 임의의 두 자료점 사이에서 3중복을 시킴으로써 중복된 구간에 대해서는 20차 베지어 합수보다 제어점 이동 폭이 줄어드는, 즉 보정이 더욱 쉽다는 결과가 나타났고, 차수가 낮음으로 인해 실수 연산이 줄어드는 반면 가산 연산이 증가하면서 향상된 연산 속도의 효과를 가져왔다. 또한 제어점을 먼저 설정하여 왜곡 모델링을 하였기 때문에 국부적인 변화를 줄 경우 제어점의 이동만으로 국부적인 변화가 가능함을 보였다. 따라서 사용자는 적은 작업량으로 색상의 보정을 이룰 수가 있다. 개선되어야 할 사항으로는 명암이 아주 낮은 때인 10~15% 일 때의 공간에서와 반대로 아주 높은 시인 85~100% 일 때의 공간에서는, 20차 베지어 곡선에 비해 떨어지는 성능을 올릴 수 있는 문제를 보임에 따라 향후에는 원 화상에 가까운 색채를 재현하기 위해 컬러 보정을 위한 최적의 3중첩 곡선의 차수에 대한 지속적인 연구가 필요하리라 사료된다.

참고문헌

- [1] 한규서, 김춘우, "기준색 고정 색 보정 방법과 칼라프린터-스캐너에의 응용", 한국화상학회지, 2권 1호, pp. 1-10, 1996, 12.
- [2] Randy Crane, "A Simplified approach to Image Processing" Prentice-Hall, 1997.
- [3] 최형일, 이근수, 이양원 "영상처리 이론과 실제", 홍릉과학출판사, 1997.
- [4] 오영환, 신성용, "컴퓨터 그래픽스", 홍릉과학출판사, 1993.
- [5] 노용덕, "Computer Graphics" 대림출판사, 1992.
- [6] 김하진, "PC 그래픽스 도구로서의 3중첩 구간적 Bezier 3차 곡선의 구성", 정보과학회 논문지, vol.20, no.2, pp.225-232, 1993. 2.
- [7] Kim, Ha-Jin; Kim, Min-Koo and Yoon, Kyung-Hyun, "On the Construction of Surfaces by Some Piecewise Cubic Interpolating Polynomials", J. OF KISS, Vol.10, No.3, pp.182-189, August 1983.
- [8] Barsky, B. A., "Computer Graphics and Geometric Modeling Using Beta-splines", pp.3-4, pp.59-68, Springer-Verlag, New York, 1988.
- [9] Bartels, R. H. and Barsky, B. A., "An Introduction to Splines for Use in Computer Graphics and Geometric Modeling," pp.36-46 and pp.145-246, Morgan Kaufmann Publishers, Inc., 1987.

A Study on the Color Correction of Real Image Using the Triplicated Piecewise Bezier Cubic-Curve

Hee-Yong Kwon* · Jie-Young Lee**

Abstract

Due to non-linear characteristics of color spaces, color corrections using linear conversions for real image near color reappearance causes color distortions. In order to overcome this problem, the Bezier Curve, constructed with a set of arbitrary plane in the linear theory, has been used.

However, the Bezier Curve increases in proportion to the number of data points, resulting in higher computational complexities.

This paper attempts to use a Triplicated Piecewise Bezier Cubic-Curve (TPBC-Curve) of which the degree is cubic on the whole interval while keeping the characteristics of Bezier Curves.

By Comparing the TPBC-Curve with Bezier Curve of 20 degree, the paper not only reduces the distortion during color correction but also lessens the relative increase of workload that is caused by the color correction in a small zone.

* Dept. of computer & information science graduate school of semyung university

** Dept. of software semyung university