

종이 찢기 시뮬레이션을 이용한 색종이 모자이크 렌더링

지용재⁰ 박영섭 서상현 윤경현

중앙대학교 정보대학원 컴퓨터 그래픽스 연구실

{yj1023, cookie, shseo, khyoon}@cglab.cse.cau.ac.kr

Colored Paper Mosaic Rendering with Torn Simulation

Gi Yong Jea⁰, Park Young Sup, Seo Sang Hyun, Yoon Kyoung Hyun

Computer Graphics Laboratory, Chang-Ang University, Seoul, Korea

요 약

본 논문은 색종이 모자이크를 시뮬레이션하기 위해 색종이 타일의 생성과 배치, 두 가지 측면에 초점을 맞추어서 진행하였다. 먼저 색종이 타일을 생성하기 위해 색종이 데이터를 두 개의 폴리곤 계층으로 정의한다. 그 다음 색종이 데이터로부터 색종이 타일을 생성하고, 마지막으로 생성된 색종이 타일을 에너지 함수 값에 따라 캔버스 상의 최적의 위치에 배치한다. 본 논문에서 제안하는 기법은 다음과 같은 이점을 가진다. 첫 번째, 실제 사람이 손으로 종이를 찢는 과정과 유사하게 시뮬레이션 함으로써, 실제 찢어진 종이의 모양과 비슷한 결과를 얻을 수 있다. 두 번째, 찢는 대상을 색종이가 아닌 다른 종류의 종이, 예를 들어 신문지·잡지 등의 데이터로 교체 적용하면 색종이 모자이크뿐만 아니라 또 다른 형태의 종이 모자이크를 손쉽게 생성할 수 있다. 세 번째, 사용자가 에너지 함수의 변수를 조정함으로써 찢어진 색종이의 위치를 조절할 수 있다.

Abstract

This study is focused on following two issues for simulating colored paper mosaic. The first one is paper tile generation, and the other one is arrangement. At first we define the paper data as two layered polygon. Then, generate paper tile from paper data, and attach it on the best location of canvas according to energy value. The proposed method in this study has some benefits like follow. First, we can obtain more natural shaped paper tile because the phase is similar to real process of tearing paper. Second, if we redefine the structure of paper like newspaper, magazine and so on, the result can be generated with the variable type of paper. And lastly, by adjusting parameters of energy function, we can control the position of the torn paper.

키워드 : 색종이 모자이크, 타일 생성, 타일 배치

Keywords: colored paper mosaic, tile generation, tile arrangement

1. 서론

모자이크는 여러 가지 색상의 돌, 유리조각, 도판, 종이 등의 타일들을 평면에 늘어놓고 모르타르, 석회, 시멘트, 풀 등으로 접착시켜 무늬나 그림모양을 표현하는 기법이다. 모자이크 기법은 건축물, 공예품 등에서 회화적 장식을 표현하거나 화면의 구도·채색효과·

구체감을 강조하기 위한 수단으로 사용한다.

본 논문에서는 모자이크 예술 중에서 색종이를 찢어 붙이는 색종이 모자이크에 대한 표현 기법을 제안한다. 실제 사람이 색종이 모자이크를 할 때는 다음과 같은 두 가지 작업을 반복한다. 첫 번째는 색종이를 원하는 모양 및 크기에 맞춰서 찢는 작업이고, 두 번째는 찢어진 색종이 타일을 캔버스 상의 적절한 위치에 배치하는 작업이다. 그러므로 하나의 완성된 색종

이 모자이크 이미지를 컴퓨터로 표현하기 위해서는 색종이 타일 생성과 배치의 두 가지 문제를 고려해야 한다. 색종이 모자이크 시뮬레이션 하기 위해서 본 논문은 먼저 찢을 대상, 즉, 색종이 데이터를 색·정점·질감 정보를 가진 두 개의 폴리곤 구조로 정의한다. 그 다음, 색종이 데이터로부터 보로노이 다이어그램(Voronoi Diagram)[1]과 임의의 점 변형(Random Point Displacement) 알고리즘 등을 적용하여 찢어진 모양을 가지는 색종이 타일을 생성한다. 마지막으로 색종이 타일은 에너지 함수(Energy Function)의 결과값에 따라 캔버스 상의 적절한 위치로 회전 및 이동시킴으로써 배치시킨다. 위 작업은 더 이상 색종이 타일이 붙여질 공간이 없을 때까지 반복 수행함으로써, 최종적으로 완성된 색종이 모자이크 이미지를 얻는다.

본 논문에서 제안하는 기법은 다음과 같이 여러 이점을 가진다. 첫 번째, 실제 사람이 손으로 종이를 찢는 과정과 유사하게 시뮬레이션 함으로써, 실제 찢어진 종이의 모양과 비슷한 결과를 얻을 수 있다. 두 번째, 찢는 대상을 색종이가 아닌 다른 종류의 종이, 예를 들어 신문지·잡지 등의 데이터로 교체 적용하면 색종이 모자이크뿐만 아니라 또 다른 형태의 종이 모자이크를 손쉽게 생성할 수 있다. 세 번째, 사용자가 에너지 함수의 변수를 조정함으로써 찢어진 색종이의 위치를 조절하여 다양한 결과 영상을 만들 수 있다.

2. 관련 연구

색종이 모자이크를 생성하기 위해서는 색종이 타일의 생성과 배치, 두 가지 문제를 모두 고려해야 한다. 기존에 진행되었던 모자이크에 대한 연구들은 두 가지 문제점에 대해서 좋은 해결점을 제시한다.

기존의 색종이 모자이크 연구들은 색종이 타일을 생성하는 좋은 기법들을 보여준다. Park[2]는 색종이 모자이크를 표현하기 위해 수동적인 기법과 찢어진 효과를 표현하기 위한 임의의 중간점 교체(Random Midpoint Displacement) 알고리즘을 제안하였다. 그러나 이 기법은 사용자가 마우스로 색종이 타일의 모양과 위치를 모두 설정하도록 되어 있다. Seo[3]는 사용

자의 편리를 위해 보로노이 다이어그램을 적용하여 색종이 타일의 모양과 위치를 자동적으로 생성할 수 있는 방법을 제안하였다. 그러나 위 두 연구는 색종이 타일의 배치를 고려하지 않았기 때문에 입력 영상의 특징을 정확하게 유지하지 못하는 제한점을 가진다. Seo[4]는 이런 문제를 해결하기 위해 영역분할 방법과 Quad-Tree 분할 방법을 기반으로 한 색종이 모자이크를 제안하였다. 이 기법은 입력 영상의 특징을 유지할 수 있었지만 실제 모자이크 작품과는 느낌이 다른 영상을 생성한다.

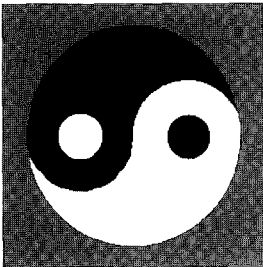
색종이 모자이크와 달리 기존에 진행되었던 타일 모자이크에 대한 연구는 색종이 타일을 배치하기 위한 많은 기법들을 보여준다. Hausner[5]는 CVD(Centroidal Voronoi Diagram) 기법을 적용하여 사각의 타일들을 고대 로마 시대의 벽화와 유사하게 배치할 수 있는 기법을 제안하였다. 그러나 이 기법은 타일들이 사각형으로 고정되는 제한을 가지고 있다. 이와 같은 제한 없이 임의의 형태를 가지는 모자이크 타일을 사용하기 위해 Kim[6]은 에너지 최소화 기법(Energy Minimization Framework)을 제안하였다. JIM(Jigsaw Image Mosaics)은 임의의 타일을 적절하게 에너지 함수에 따라 적절히 배치함으로써, 인상적인 결과를 생성한다. 그러나 이 기법은 데이터베이스로부터 이미지를 탐색하는 시간과 타일의 배치 과정에서 많은 시간이 소요된다. 이런 점을 극복하기 위한 기법으로는 Smith[7]가 제안한 CAVD (Centroidal Area Voronoi Diagram), Dalal[8]가 제안한 SSD, 그리고 FFT(Fast Fourier Transform)[9] 등이 제안되었다.

기존의 연구들은 모자이크 영상을 생성하는데 있어서 타일의 생성 또는 배치 중 한가지만을 고려하였다. 그러나 본 논문은 타일의 생성과 배치를 모두 고려하여, 실제 색종이 모자이크 제작 과정과 비슷하게 시뮬레이션 함으로써, 좀 더 자연스런 결과 영상을 생성한다.

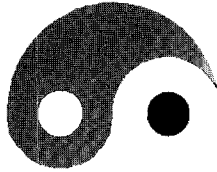
3. 이미지 처리

입력된 영상은 그 특징을 유지하기 위해 먼저 [그림

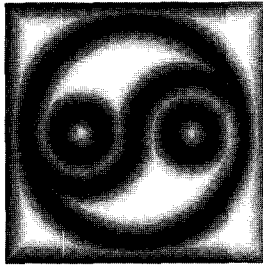
1(b)와 같이 여러 영역들로 분할한다. 영역 분할에 의해 생성된 이미지를 본 논문에서는 컨테이너 이미지라 한다. 각각의 컨테이너, 즉, 영역에 대해 영역의 경계선과 가장 가까운 거리 값을 구하여 [그림 1(d)]와 같이 거리 맵을 생성한다. 거리 값은 색종이 타일을 배치하기 위한 에너지 함수 값을 계산하는데 필요하다.



(a) 소스 이미지



(b) 컨테이너 이미지



(c) 이미지 경계선

(d) 거리 맵

[그림 1] 이미지 처리 과정 (a)→(b)→(c)→(d), (a) 입력된 소스 이미지, (b) 영역별로 분할된 컨테이너 이미지, (c) 소스이미지의 특징점(경계선), (d) 영역별 거리 맵

3.1. 이미지 영역 분할

입력된 이미지는 그 특징에 따라 색종이가 붙여질 영역을 분할한다. 본 논문에서는 이미지의 영역을 분할하기 위해 Meanshift 알고리즘[10]을 적용한다. Meanshift는 확률 밀도 경사도 예측 (density gradient estimation)을 위해 처음 제안되었고, 현재 일반화된 클러스터링 알고리즘으로 객체의 분할, 필터링, 최적화 문제의 해결에 유용하게 사용되고 있다. 그러나 Meanshift만으로는 사용자가 요구하는 명확한 이미지

의 특징을 추출하기는 어렵다. 그러므로 본 논문에서는 분할된 영역을 사용자가 임의로 병합할 수 있도록 인터페이스를 제공한다. [그림 1(b)]는 [그림 1(a)] 이미지에 대해 영역분할과 영역통합 과정을 거쳐 생성된 이미지를 보여준다.

3.2. 거리 맵 생성

거리 맵은 [그림 1(d)]와 같이 각각의 컨테이너 안의 픽셀들과 경계선 상의 픽셀들 사이의 가장 짧은 거리 값을 저장하여 표현한다. B 를 경계선 상의 픽셀이라 하면, 본 논문에서의 거리 값은 다음과 같이 정의된다.

$$D(x,y) = \min(\{s | s = \sqrt{(x-i)^2 + (y-j)^2}, \forall (i,j) \in B\}) \quad (1)$$

위 식(1)에 의해 생성된 거리 맵은 타일을 배치하기 위한 과정으로서의 에너지 값을 계산하는데 사용한다. 타일을 가능한 한 경계선부터 조밀하게 배치하기 위해 각 픽셀에 대한 경계선과의 최소 거리 값을 구하는 것이다.

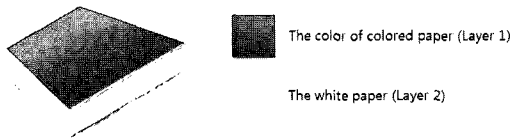
4. 색종이 데이터

본 논문은 실제 색종이 모자이크 작업과 유사하게 시뮬레이션 하기 위해서 2개의 풀리곤 계층으로 구성된 색종이 데이터를 정의한다. 이렇게 정의된 색종이 데이터들은 리스트로 관리되어 사용자가 필요할 때에 불러와서 사용한다. 색종이 데이터 정의는 다음과 같이 두 가지 이점을 가진다. 첫 번째, 실제 색종이를 찢는 것과 유사하게 시뮬레이션 하게 됨으로써, 자연스럽게 다양한 모양을 가지는 색종이 타일을 생성할 수 있다. 두 번째, 찢는 대상인 색종이 데이터를 신문지 또는 잡지 등으로 새롭게 정의함에 따라 손쉽게 또 다른 형태의 종이 모자이크를 생성할 수 있다.

4.1. 색종이 데이터 구조

본 논문은 찢을 대상인 색종이를 [그림 2]와 같이

색·점·질감 등의 정보를 포함하는 두 개의 폴리곤 계층구조로 정의한다. 첫 번째 계층은 색종이의 색을 가진 계층으로써, 일반적으로 색종이의 색이 표현되는 윗면을 의미한다. 두 번째 계층은 색종이가 찢어질 때 섬유유질이 분리되면서 나타나는 흰 종이 부분을 표현한다. 본 논문은 이 계층의 질감을 좀 더 사실적으로 표현하기 위해 텍스처 이미지 대신 펄린 노이즈(Perlin Noise)[11]함수를 사용한다. 이 기법은 Cassidy[12][13]에서도 종이의 질감을 표현하기 위해 사용되었다.



[그림 2] 색종이 데이터 구조

4.2. 색종이 데이터 관리

실제 존재하는 색종이의 색상은 제한되어 있으므로, 본 논문은 제한된 색상의 색종이 데이터들을 미리 파일로 구축하고, 파일로부터 색종이의 색상을 검색키로 가지는 데이터 리스트를 생성한다. 사용자는 리스트로부터 필요로 하는 색상을 요청하고, 얻은 색종이 데이터로 타일 생성작업을 수행한다. 이 기법은 제한된 색종이의 색상을 사용함으로써 더 사실과 같은 색종이 모자이크를 생성할 수 있는 장점을 가진다. 또한 리스트에 저장된 색종이들을 누적하여 찢음으로써, 실제 찢어진 색종이와 같이 자연스러운 모양의 색종이를 생성할 수 있다.

5. 색종이 타일의 위치, 크기, 색상

색종이 데이터가 정의되면, 찢어진 모양의 타일을 생성하기 위해서 찢을 타일의 색상과 크기를 결정한다. 이 작업을 수행하기 위해 각각의 컨테이너에 대해서 색종이 타일이 붙여질 임시 위치를 결정하고, 그 위치에 붙여질 색종이 타일의 색상과 크기를 결정한다.

다.

5.1. 색종이 타일의 위치 설정

색종이 타일의 크기와 색상을 결정하기 위해서 먼저 할 일은 각 컨테이너에 대해 타일이 붙여질 임시 위치를 결정하는 것이다. 일반적으로 사람들이 색종이를 붙일 때는 이미지의 경계선부터 타일의 크기가 큰 순서대로 붙여나간다. 이런 것을 고려하여 본 논문에서는 색종이 타일이 붙여질 임시 위치를 식(2)의 $P(x,y)$ 가 최대인 위치 (x,y) 로 정의한다.

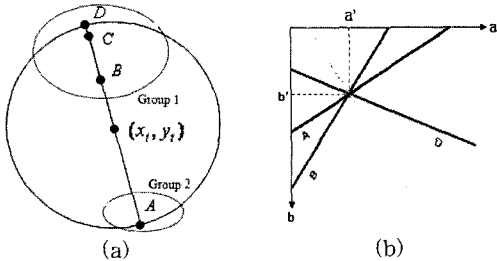
$$P(x,y) = \sum_{i=x-1}^{x+1} \sum_{j=y-1}^{y+1} D(x,y) - D(i,j) \cdot [x \neq i \text{ and } y \neq j] \quad (2)$$

식(2)에서 $[x \neq i \text{ and } y \neq j]$ 는 $x \neq i$ 그리고 $y \neq j$ 와 같으면 1, 그렇지 않으면 0이다. 그리고 각각 색종이 타일의 크기는 $D(x,y)$, 경계선과 가까운 정도는 $D(x,y) - D(i,j)$ 에 의해서 결정된다. 즉, 각각의 항목이 크면 클수록 경계선과 가까우면서 크기가 큰 색종이 타일을 붙일 수 있는 위치가 된다.

5.2. 색종이 타일의 크기

붙여질 색종이의 임시 위치 (x,y) 가 결정되면, 그 위치에서 붙일 수 있는 타일의 최대 크기를 대략적으로 결정한다. (x,y) 에서의 거리 값인 $D(x,y)$ 은 붙일 수 있는 색종이의 최소 크기로 정의할 수 있다. 그러나 가능한 한 큰 타일을 붙이기 위해서는 (x,y) 에서 경계선과의 거리가 먼 값인 $D_{\max}(x,y)$ 을 구한다. 본 논문에서는 허프 변환(Hough Transform)[9]을 적용하여 $D_{\max}(x,y)$ 을 구한다. 먼저 [그림 3(b)]와 같이 (x,y) 로부터 비슷한 기울기를 가지는 점들을 [그림 3(a)]와 같이 y 값을 중심으로 두 개의 그룹으로 묶는다. 그 다음 각각의 그룹에서 (x,y) 로부터 가장 작은 거리 값을 가지는 두 점을 선택하고 그 중에서 값이 큰 것을 $D_{\max}(x,y)$ 로 결정한다. 최종적으로 찢을 타일의 크기는 $D(x,y)$ 와 $D_{\max}(x,y)$ 로 결정한다. 단, 여기서 생성될 색종이 타일의 최소 크기(T_{\min})와 최대 크기(T_{\max})

는 유저에 의해서 제한된다.



[그림 3] 허프 변환을 이용한 타일 크기 계산

5.3. 색종이 타일의 색상

각 컨테이너에 대해서 색종이 색상은 컨테이너의 평균 색상 또는 입력 영상의 평균 색상에 따라 정한다. 사용자가 정의 한 일정 값을 T 라 하면, 타일의 색상은 다음과 같이 정의 된다. 컨테이너의 평균 색상을 $C1$, 입력 영상의 평균 색상을 $C2$ 라 정의할 경우, $C1$ 또는 $C2$ 로부터 RGB 채널의 유클리디언 거리 값이 T 보다 작은 색종이의 색상 중 하나를 타일의 색상으로 결정한다.

6. 색종이 타일 생성

찢을 색종이 타일의 크기(S)와 색상(C)이 결정되면, 그 다음으로 C 색상을 가지는 색종이 데이터를 리스트로부터 불러와서 S 만큼 찢어내는 작업을 수행한다.

6.1. 색종이 타일의 폴리곤 결정

색종이 타일 생성의 첫 단계는, 찢을 타일의 모양 즉, 폴리곤을 결정하는 것이다. 본 논문에서는 색종이 타일의 폴리곤을 결정하기 위해서 보로노이 다이어그램[1]을 적용한다. 먼저 색종이 객체를 3절에서 결정된 크기에 따라 격자로 분할한다. 만약 분할된 격자의 수가 n 개이면, 각각의 분할된 격자에 하나씩 총 n 개를 보로노이 사이트들을 배치하여 보로노이 다이어그램

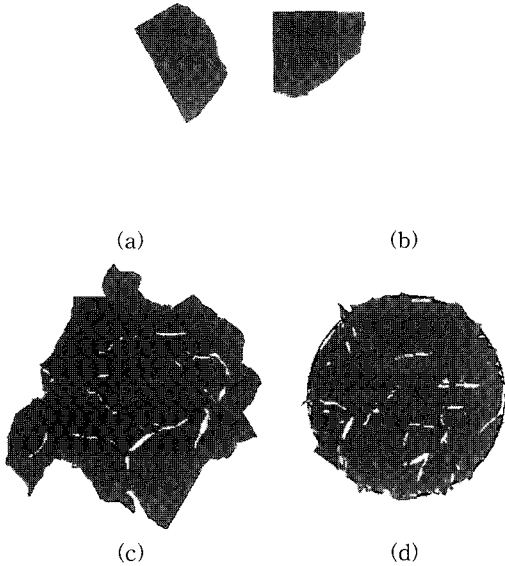
램을 생성한다. 마지막으로 작업은 생성된 n 개의 보로노이 폴리곤들 중 어떤 것을 타일의 폴리곤으로 사용할 것인지 결정하는 것이다. 일반적으로 실제 사람들이 색종이를 찢는다고 가정할 때, 색종이의 가장자리에서부터 찢어나가기 시작한다. 그러므로 본 논문에서도 가장자리 부분에서부터 찢어나가기 위해 n 개의 보로노이 폴리곤 중에서도 색종이 데이터의 정점을 포함한 폴리곤 중 하나를 임의로 선택하여 찢을 색종이 타일의 폴리곤으로 결정한다.

6.2. 폴리곤 경계선 완화

색종이 타일로 사용될 폴리곤은 보로노이 다이어그램의 특성상 [그림 4(b)]의 검은 선으로 표시된 것과 같이 거친 경계선을 가질 수 있다. 실제 사람이 손으로 색종이를 찢을 때는 힘의 방향을 갑자기 바꾸지 않기 때문에 거친 형태의 경계선은 색종이 타일로 바로 적용하기에는 부자연스럽다. 그러므로 본 논문에서는 좀 더 자연스러운 색종이 타일을 생성하기 위해서 베지어 곡선[14]을 적용하여 폴리곤의 경계선을 보간하여 부드럽게 만든다. 본 논문은 베지어 곡선을 생성하기 위해 [그림 4(b)]과 같이 정점 A 와 D 를 knot vector로, E 와 F 를 control point로 정의한다. 단, 부드러운 경계선을 생성하기 위해서, BC 와 같이 길이가 특정 값(T)보다 짧은 경계선은 베지어 곡선을 생성하는데 있어서 무시해 버린다. 베지어 곡선에 의해서 보간된 폴리곤의 최종 결과는 [그림 4(b)]의 붉은 선으로 표현된 폴리곤과 같다.

6.3. 종이의 찢어진 효과

색종이 타일 생성의 마지막 작업으로 폴리곤 경계선에 실제 종이가 찢어진 것과 같은 효과를 추가한다. 실제 종이의 찢어진 단면은 불규칙하며, 섬유질이 어긋나서 분리되기 때문에 흰 종이 부분이 나타난다. 본 논문에서는 이런 부분을 표현하기 위해 색종이 데이터의 각 계층에 대해 임의의 점 변형(이하, RPD) 알고리즘을 독립적으로 적용시킨다. RPD는 [그림 4(c)]와 같이 폴리곤 경계선 상의 임의의 점을 경계선의 수직 방향으로 변위시키는 알고리즘으로써, 경계선의 길이

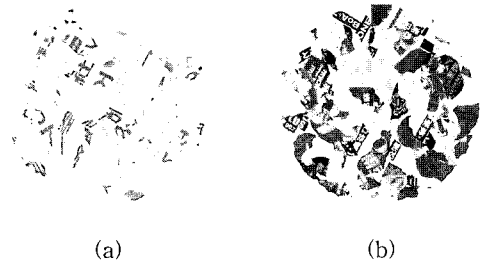


[그림 5] 색종이 타일 배치, (a) 최적의 배치 상태, (b) 잘못된 배치 상태, (c) 타일의 겹침을 중시한 배치 상태 ($W_e < W_o$), (d) 이미지 특징을 중시한 배치 상태 ($W_e > W_o$)

8. 결론 및 향후 연구

본 논문은 색종이 모자이크를 시뮬레이션 하기 위해 색종이 타일의 생성과 배치를 모두 고려하는 기법을 제안하였다. 본 논문의 기법은 실제로 색종이 모자이크를 수행하는 과정과 유사하게 진행되기 때문에 다음과 같은 이점을 가진다. 첫 번째, 실제와 비슷한 자연스런 모양의 색종이 타일을 생성할 수 있다. 두 번째로 찢는 대상, 즉 색종이의 구조를 신문지나 잡지와 같이 새롭게 정의하면, [그림 6]과 같이 또 다른 종이 모자이크를 손쉽게 표현할 수 있다. 세 번째, 사용자가 에너지 함수의 변수들을 조절함으로써 색종이 타일의 배치를 달리하여 또 다른 결과를 생성할 수 있다.

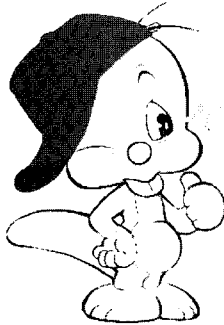
본 논문에서 제안한 알고리즘의 결과는 [그림 7, 8, 9]에서 보여준다. [그림 7]은 만화와 같이 비교적 단순한 영상에 적용한 결과이다. 본 논문의 알고리즘은 색



[그림 6] 다른 형태의 모자이크 표현, (a) 신문지 모자이크, (b) 잡지 모자이크

종이 타일의 모양을 임의로 생성하기 때문에 복잡한 영상에 적용할 경우, 이미지의 경계선을 넘어서 배치되는 경우가 생긴다. 이는 추후에 해결해야 할 문제이다. [그림 8]은 [그림 7]보다 크기가 큰 타일을 적용하여 색종이 모자이크 영상을 생성한 결과이다. 그러나 큰 타일을 생성하여 붙이기 때문에 몇몇 부분에서 영역의 크기보다 큰 타일이 생성되어 이미지의 경계가 무너지는 경우가 생긴다. 이를 해결하기 위해 영역별로 타일의 최소/최대 크기를 조절할 필요가 있다. [그림 9, 10]은 본 알고리즘에 의해서 생성된 또 다른 결과를 보여준다.

이 알고리즘에서 몇 가지 해결해야 할 문제들을 다음과 같다. 첫 번째, 색종이 타일의 모양이 임의로 생성되기 때문에, 매우 작은 컨테이너에 대해서 그 크기보다 큰 타일이 생성되어 배치될 수 있다. 이를 해결하기 위해서는 향후 사용자가 원하는 모양으로 타일을 생성할 수 있도록 보로노이 사이트들을 적절하게 배치할 수 있는 기법이 필요하다. 두 번째, 아직 텍스처 이미지 이외에 질감을 표현하기 위한 알고리즘이 없기 때문에 결과가 실제 색종이 모자이크와는 거리가 있다. 그러므로 텍스처 이미지 외에 타일의 그림자, 광원 등의 효과를 추가한다면, 실제 색종이 모자이크와 비슷한 영상을 얻을 수 있을 것이다. 세 번째, 본 논문의 타일 배치 알고리즘은 픽셀 단위로 수행되기 때문에, 처리 시간이 오래 걸린다. 이를 해결하기 위해 Fast Fourier Transform[9] 또는 GPU와 같이 가속화 알고리즘 또는 하드웨어를 적용함으로써, 처리 시간을 줄이기 위한 기법이 필요하다.



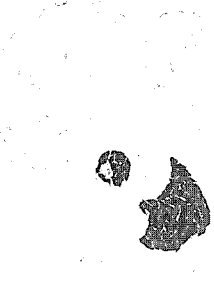
(a)



(b)



(c)



(d)

[그림 7] 결과, (a) 원본 이미지, (b) 이미지(a)에 색종이 모자이크 적용 결과 (이미지 크기: 273 x 343, $T_{\min} = 5$, $T_{\max} = 20$) (c) 원본 이미지, (d) 이미지(c)에 색종이 모자이크 적용 결과 (이미지 크기: 336 x 400, $T_{\min} = 5$, $T_{\max} = 20$)

참고문헌

[1] Berg M., Kerveld M., Overmars M., Schwarzkopf O, Computational Geometry Algorithms and Applications, Springer, 1997.
 [2] Park Y., Kim Y., Jho C., Yoon K., Mosaic techniques using color paper, Proc. KCGS'00 Conference, pp.42-47, 2000.

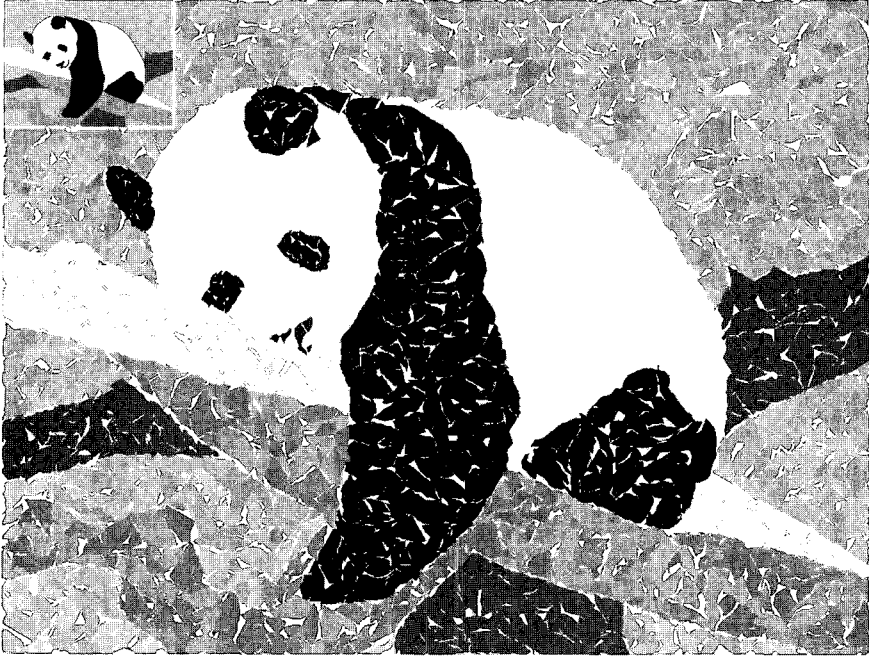
[3] Seo S., Park Y., Kim S., Yoon K., Colored paper mosaic rendering, Proc SIGGRAPH'01 Conference Sketches and Abstracts and Applications, p.157, 2001.
 [4] Seo S., Kang D., Park Y., Yoon K., Colored paper mosaic rendering based on image segmentation, Proc KCGS'01 Conference, pp.27-34, 2001.
 [5] Hausner A., Simulating decorative mosaics, Proc. SIGGRAPH'20, vol.20, pp.573-580, 2001.
 [6] Kim J., F. Pellacini, Jigsaw image mosaics, Proc. SIGGRAPH'02, vol.21, pp.657-664, 2002.
 [7] Smith K., Liu Y., Klein A., Animosaics, Computer Animation 2005 : SIGGRAPH/Eurographics Symposium on Computer Animation, ACM, pp.201- 208, 2005.
 [8] Dalal K., Klein A., Liu Y., Smith K., A spectral approach to npr packing. NPAR'06, Proceedings of the 4th International Symposium on Non-Photorealistic Animation and Rendering, ACM, 2006.
 [9] Gonzalez W., Digital Image Processing 2th Edition, Prentice Hall, 2002.
 [10] Comanucu D., Meer P., Mean shift: A robust approach toward feature space analysis. IEEE Transaction on Pattern Analysis and Machine Intelligence, pp.603-619, 2002.
 [11] Ken Perlin, An image synthesizer, SIGGRAPH'85 Proceedings, pp.287-296, 1985.
 [12] Cassidy J. Curtis, Sean E., Anderson, Joshua E. Seirns, Hurt W. Fleischer, David H. Salesin, Computer Generated Watercolor, SIGGRAPH'97 Proceedings, pp.421-430, 1997.
 [13] Cassidy J. Curtis, Loose and Sketchy Animation, SIGGRAPH'98 Abstracts and Applications, p.317, 1998
 [14] Mortenson M., Mathematics for Computer Graphics Applications 2nd Edition. Industrial Press, 1999.



[그림 8] 모자이크 렌더링 결과 이미지 (이미지 크기: 800 x 600, $T_{\min} = 5$, $T_{\max} = 20$)



[그림 9] 모자이크 렌더링 결과 이미지 (이미지 크기: 750 x 488, $T_{\min} = 10$, $T_{\max} = 50$)



[그림 10] 모자이크 렌더링 결과 이미지 (이미지 크기: 1,024 x 768, $T_{\min} = 10$, $T_{\max} = 30$)