

# 영상 디스타일화

이현준<sup>o</sup>                      이승용

포항공과대학교

{crowlove, leesy}@postech.ac.kr

## Image Destylization

Hyunjun Lee<sup>o</sup>                      Seungyong Lee

POSTECH

### 요약

본 논문에서는 주어진 영상을 필터링하여 영상의 스타일을 제거하는 방법을 제시한다. 스타일이 제거된 영상은 영상 분류, 특징점 인식, 영상 분할 등의 다양한 용도에 쓰일 수 있다. 또한 원래 영상과 스타일이 제거된 영상을 비교하여 영상의 스타일을 유추할 수 있다. 본 논문에서는 이를 위해 주어진 영상에서 스타일 벡터를 계산한 후 계산된 스타일 벡터를 이용하여 영상에 양방향 필터링을 적용한다. 이때 영상의 경계 부분에서 스타일을 효과적으로 분리하는 방법과 다중 해상 처리 방법을 적용하여 다양한 크기와 방향의 스타일을 찾아낸다. 그 결과 주어진 영상에서 다양한 크기와 방향의 스타일을 제거하고 영상의 중요한 내용만을 효과적으로 나타낸다.

### Abstract

We propose an image filtering technique that removes various image styles. To destylize a given image, we define image styles as repeated patterns existing in the image. For all pixels of the image, we compute image styles as style vectors. We remove image styles by using bilateral filtering based on these style vectors. Destylization results show well smoothed images while preserving feature boundaries. Our method effectively removes image styles and reveals image structures clearly, and results can be applied to several applications such as texture transfer.

**키워드:** 비사실적 렌더링, 텍스처 분석, 양방향 필터링, 영상 평탄화

**Keywords:** non-photorealistic rendering, texture analysis, bilateral filtering, image smoothing

## 1. 서론

영상의 필터링은 영상 처리에서 가장 기본이 되는 작업 중의 하나이다. 필터링의 가장 기본적인 개념은 영상의 각 화소에 어떤 함수를 적용시켜 영상의 정보를 더 효과적으로 드러내는 것이다. 그러므로 필터링은 컴퓨터 비전 및 컴퓨터 그래픽스에서 영상을 처리하는 데 가장 중요한 작업 중 하나라고 할 수 있다.

필터링 기법 중에서도 영상의 평탄화 (image smoothing)는 영상의 노이즈나 지나치게 자세한 상세 정보 (detail information)를 제거한다. 가우시안 평탄화 (Gaussian smoothing)는 평탄화 기법 중에서 가장 기본이 되는 기법이다. 그러나 가우시안 평탄화는 영상의 큰 구조마저도 평탄화하여 영상의 내용을 파악하기 힘들게 한다는 단점이 있다.

가우시안 평탄화의 단점을 해결하기 위하여 다양한 평탄화 기법이 연구되었다 [1, 2]. 그 중 양방향 필터링 (bilateral filtering) 기법은 영상을 평탄화할 때 영상의 색차에 따른 경계 부분을 보존하면서 영상을 효과적으로 평탄화한다.

그러나 실제 사람들이 그린 예술적 작품은 다양한 도구와 기법에 의해 만들어진다. 그렇기 때문에 색차에만 의지하여

경계 부분을 나누게 되면 실제 사람이 인지하는 경계와는 다른 결과를 얻는다. 그러므로 본 논문에서는 양방향 필터링을 기반으로 하여 영상을 평탄화하되 영상의 색차가 아닌 스타일의 차이를 계산한 후 이를 이용하여 영상을 사람이 인지하는 것과 비슷하게 평탄화하는 방법을 제시한다.

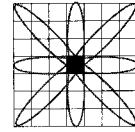
이와 별개로, 영상의 스타일을 찾아내고 분석하는 텍스처 분석 (texture analysis)은 컴퓨터 비전 및 영상 처리에서 오랫동안 연구된 분야이다. 텍스처 분석은 영상에 존재하는 텍스처의 형태, 크기, 방향 등을 분석하여 이를 이용해 영상 분할 (image segmentation), 영상 분류 (image classification) 등을 수행한다.

그러나 텍스처 분석에서는 물체 인지, 영상의 분류 등에 초점을 맞추었기 때문에 텍스처 분석 결과를 이용한 영상의 필터링이나 평탄화에 관한 연구는 많이 이루어지지 않았다.

본 논문에서는 영상의 스타일을 분석한 후 이를 이용해 영상의 디스타일화 작업을 수행하고, 이 과정에서 발생할 수 있는 문제들을 해결하기 위한 방법들을 제시한다. 본 논문에서 제시하는 평탄화 기법은 기존의 연구에 비해서 다음과 같은 장점을 지닌다.

- 스타일 정보 기반의 평탄화: 영상의 평탄화할 때 기존의

색이나 위치 정보만을 사용하던 것에서 벗어나 영상의 스타일 정보를 이용하여 평탄화 하는 방법을 제시한다. 스타일 정보를 이용한 평탄화로 영상의 다양한 스타일을 제거하여 영상의 구조적 정보를 보다 더 찾아내기 쉽게 한다.



- **경계 부분 처리:** 화소 사이의 스타일 벡터를 비교할 때 스타일 벡터의 일부분만 비교하여 영상의 경계 부분에서 서로 다른 스타일이 섞이지 않고 경계가 뚜렷하게 남아 있게 한다.
- **다중 해상 처리:** 필터링 시에는 다중 해상 기법을 적용하여 기존의 양방향 필터링보다 더 빠른 처리가 가능하면서 매끄럽게 필터링 된 결과를 만들어 낸다.

**그림 1:** 영상의 한 화소와 그에 해당하는 8방향 스타일 벡터. 영상의 각 화소에 대하여, 여덟 방향의 이웃 화소들을 이용하여 스타일 벡터를 계산한다.

화한다. 이는 수식 (2)로 표현된다.

$$\mathbf{H}(p) = \frac{1}{w} \sum_{n \in \mathbf{N}} \mathbf{I}(n) c_s(p, n) c_f(p, n) \quad (1)$$

$$w = \sum_{n \in \mathbf{N}} c_s(p, n) c_f(p, n)$$

$\mathbf{N}$  : 화소  $p$ 의 주위 화소들의 집합

$\mathbf{H}(p)$ ,  $\mathbf{I}(p)$ 는 각각 필터링 된 결과 영상과 원본 영상을 의미하며,  $w$ 는 정규화 항목이다.  $c_s$ 와  $c_f$ 는 각각 두 화소 사이의 거리 차이와 스타일 차이로부터 계산된 유사도를 의미한다. 유사도는 거리에 대한 가우시안 함수로, 수식 (2)와 같이 계산되었다.

$$c(p, n) = \exp \left\{ -\frac{1}{2} \left( \frac{d(p, n)}{\sigma_d} \right)^2 \right\} \quad (2)$$

여기서  $d$ 는 유클리드 거리를 의미하며,  $d_s$ 와  $d_f$ 는 각각 두 화소 사이의 위치와 스타일 차이에 따른 거리를 의미한다.

## 2. 알고리즘

영상을 디스타일화 하기 위해서는 우선 영상의 스타일을 정의해야 한다. 영상에서 스타일은 특정 패턴의 반복으로 표현될 수 있다. 예를 들어, 펜하나 연필화의 경우에는 펜 스트로크의 반복되는 패턴으로 스타일을 표현할 수 있으며 유화와 같은 경우에는 나이프나 붓으로 그리는 스트로크의 반복되는 패턴으로 표현할 수 있다.

스타일이 정의되었으면 스타일을 제거하기 위해 두 가지의 과정을 거친다. 먼저 영상에 존재하는 다양한 반복 패턴을 검출한 후 이들 패턴의 크기, 방향 등과 같은 특징 요소들을 스타일 벡터로 계산한다. 그 다음 계산된 스타일 벡터를 이용하여 영상을 디스타일화 한다.

### 2.1 스타일 벡터 계산

영상의 스타일을 계산하는 연구는 컴퓨터 비전에서 오랫동안 이루어져 왔다. 여기에는 DFT (discrete Fourier transform) 나 DCT (discrete cosine transform)를 이용한 방법, 필터 뱅크를 이용한 방법 [3, 4], 웨이블릿 변환 (wavelet transform)을 이용한 방법 등이 존재한다 [5, 6]. 이들 중 본 논문에서는 DFT를 이용하여 스타일 벡터를 계산하였다.

스타일 벡터 계산 과정에서는 먼저 주어진 영상을 흑백 영상으로 바꾼다. 그 다음 영상의 각 화소에 대해서 여덟 방향의 주위 화소들을 이용하여 스타일 벡터를 계산한다 (그림 1). 한 화소에서 한 방향의 스타일 벡터를 계산할 때는, 먼저 해당 화소를 중심으로 해당 방향의 주위 화소들을 이용하여 1D DFT를 수행한 후 계산된 결과 벡터를 가져 온다.

결과 벡터에서 첫 번째 원소는 중심 화소와 주위 화소들의 색의 평균값이고, 이는 영상의 스타일에 크게 영향을 미치지 않으므로 결과 벡터에서 제거한다. 그 다음 나머지 결과 벡터의 파워 스펙트럼을 구하여 그 벡터의 절반을 스타일 벡터로 저장한다. 절반만을 스타일 벡터로 저장하는 이유는 파워 스펙트럼이 벡터의 중심을 기준으로 대칭을 이루기 때문이다.

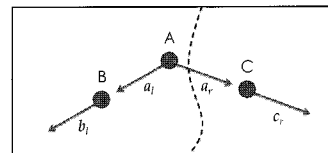
이 과정을 여덟 방향에 대해 수행하여 주위 여덟 방향의 스타일 벡터를 계산한 후 계산된 스타일 벡터를 모두 합쳐 영상의 스타일 벡터로 저장한다. 결과적으로 각 화소에는  $\{8 \times (\text{필터 반지름}/2)\}$  크기의 스타일 벡터가 저장된다.

### 2.3 경계 부분 처리

양방향 필터링에서 화소 사이의 유사도를 측정할 때 각 화소의 모든 스타일 벡터를 이용하면 경계 부분에서 문제가 생길 수 있다. 그림 2에서, 화소 A와 B는 같은 스타일 안에 있지만 A와 B의 모든 스타일 벡터를 이용하여 유사도를 측정하면 A의 스타일 벡터는 다른 스타일의 정보를 포함하고 있기 때문에 유사도가 작게 계산된다.

이 문제를 해결하기 위해 다시 그림 2를 보면  $a_i$ 과  $b_i$ 은 서로 유사한 값을 가졌으나  $a_r$ 과  $c_r$ 은 서로 다른 값을 지녔다는 것을 유추할 수 있다. 즉, 중심 화소에서 이웃 화소로 향하는 벡터 방향의 스타일 벡터를 이용하여 두 화소 사이에 영상의 경계가 존재하는지의 여부를 판단할 수 있다.

그러므로 스타일 벡터 사이의 거리를 측정할 때 중심 화소와 이웃 화소를 각각  $p_c$ ,  $p_n$ 이라고 하면, 여덟 방향의 스타일 벡터 중  $p_c$ 에서  $p_n$ 으로 가는 벡터인  $p_n - p_c$ 의 방향에 가장 가까운 한 개 혹은 두 개의 스타일 벡터 사이의 거리를 계산한 후 이를 가중 평균하여 두 화소의 스타일 벡터의 차이로 삼는다.



**그림 2:** 영상의 구조적 경계 부분에서의 예. 화소 A와 B는 화소 C와 다른 스타일을 가지고 있다. 이를 찾아내기 위해 각 화소의 스타일 벡터 중 일부만을 이용하여 유사도를 측정한다.

### 2.2 필터링

스타일 벡터를 계산한 다음에는 계산된 스타일 벡터를 이용하여 영상을 양방향 필터링 (bilateral filtering) 기법으로 평탄

## 2.4 다중 해상 필터링

스타일 벡터를 이용한 양방향 필터링은 양방향 필터링의 필터링 범위에 비례하여 필터링 시간이 증가한다. 그러므로 영상의 큰 스타일을 제거하기 위해 필터링 범위를 키우면 매우 많은 시간이 소요된다. 또한 필터링 범위가 작으면 영상의 큰 스타일을 효과적으로 제거할 수 없다.

그러므로 이 문제를 해결하기 위해 주어진 영상을 가우시안 피라미드 (Gaussian pyramid)로 만든 다음 피라미드의 각 영상에 대해 필터링을 수행한다 (그림 3).

각각의 해상도에서 필터링을 수행하고 난 후에는 필터링 결과들을 합쳐야 한다. 이를 위해 먼저 필터링을 수행할 때 영상의 각 화소에 대해 지역적 평탄도 값을 정의한다.

$$\text{지역적 평탄도 } (I_f) = \frac{\sum_{n \in N} c_s(p, n) c_f(p, n)}{\|N\|} \quad (3)$$

$\|N\|$  : 주위 화소들의 개수

이 값은 필터링 시에 주위 화소들과의 유사도의 평균을 의미하며 0과 1 사이의 값을 가진다 (그림 4).

가우시안 영상 피라미드의 각 단계의 영상을 필터링 하여 얻어낸 필터링 결과 영상과 지역적 평탄도 배열을 각각  $I(p)$ 와  $L(p)$ 라고 정의한다. 그 다음 가장 작은 해상도의 영상에서부터 필터링 결과 영상과 지역적 평탄도 배열을 다음 해상도의 필터링 결과 영상과 지역적 평탄도 배열과 합친다. 이를 위해 먼저 이전 단계의 필터링 결과 영상과 지역적 평탄도 배열을 결삼차 보간법 (bicubic interpolation)을 이용하여 두 배로 확대시킨다. 그 다음 확대된 필터링 결과 영상과 다음 해상도의 필터링 결과 영상을 확대된 평탄도 배열을 이용하여 선형 보간하여 합친다. 이 과정을 가장 작은 해상도의 영상에서 원래 해상도의 영상에까지 반복하여 결과 영상을 얻어낸다 (수식 (4)).

$$\begin{aligned} R_k(p) &= W_{k-1}^*(p) R_{k-1}^*(p) + (1 - W_{k-1}^*(p)) I_k(p) \\ W_k(p) &= \max(W_{k-1}^*(p), L_k(p)) \end{aligned} \quad (4)$$

$$R_1(p) = I_1(p), \quad W_1(p) = L_1(p), \quad k = 1, \dots, n$$

여기서  $W_{k-1}^*(p)$ 와  $R_{k-1}^*(p)$ 는 각각 확대된 이전 단계의 평탄도 배열과 확대된 필터링 결과 영상이며,  $L_k(p)$ 와  $I_k(p)$ 는 다음 해상도의 평탄도 배열과 필터링 결과 영상이다.

결과적으로 주위에 비슷한 스타일 벡터가 많은 화소는 더 넓은 범위의 픽셀과 혼합되고 그렇지 않은 화소는 더 적은 범위의 픽셀과 혼합되어 영상의 경계 부분이 유지되면서 부드럽게 평탄화된 결과를 얻을 수 있다.

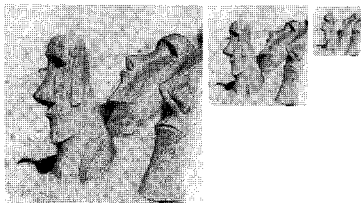


그림 3: 가우시안 영상 피라미드. 왼쪽에서부터 원본 영상, 1/2, 1/4로 축소된 영상.

## 3. 실험 결과

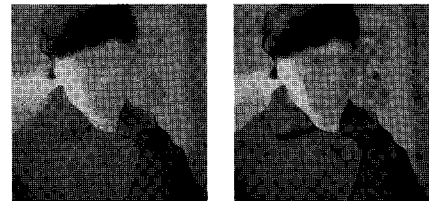
본 논문에서는 모든 영상에 대해 필터 반지름을 다섯 픽셀로, 가우시안 피라미드 영상의 개수를 네 개로 고정하였다.



그림 4: 그림 3에 해당하는 지역적 평탄도 배열의 시각화 결과; 흰색 = 1, 검은색 = 0. 지역적 평탄도 값이 클수록 한 화소가 주위 화소들과 유사한 스타일 벡터를 가졌다는 것을 의미한다. 지역적 평탄도는 영상의 한 구조 내에서는 큰 값을 지니며, 구조의 경계 부분에서는 0에 가까운 값을 지닌다.

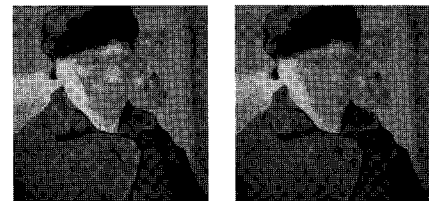
기존의 양방향 필터링은 유화 스타일의 그림을 효과적으로 디스타일화 하지 못하지만, 본 논문에서 제시한 방법은 스타일을 효과적으로 제거하는 것을 볼 수 있다 (그림 5). 그림 6(a)는 [6]에서 사용한 가보 변환 (Gabor transform)을 이용하여 각 화소의 스타일 벡터를 계산한 후 양방향 필터링을 수행한 결과이다. 그림 6(b)와 비교해보면 스타일이 덜 제거되었고 영상의 경계 부분에서 무리 (halo)가 진 것을 확인할 수 있다. 그림 7(c)는 연필화의 필터링 결과이다. 그림의 윤곽선이 다른 색과 섞이지 않고 잘 표현되었음을 알 수 있다. 그림 7(b)를 보면 윤곽선 부분에서는 평탄도가 0에 가깝고 그렇지 않은 부분에서는 1에 가깝다는 사실을 알 수 있다.

원본 영상과 디스타일 된 영상을 가지고 있으면 이들을 이용하여 영상 스타일 전이와 같은 응용 분야에도 사용할 수 있다. 그림 9(b)는 이미지 아날로지 (image analogy) 기법 [7]을 이용하여 그림 8의 스타일을 그림 9(a)에 전이시킨 결과이다.



(a) 양방향 필터링 (b) 제시된 방법

그림 5: 양방향 필터링과의 비교 결과. 제시된 방법은 영상의 스타일을 제거하면서 영상의 구조를 효과적으로 드러낸다.

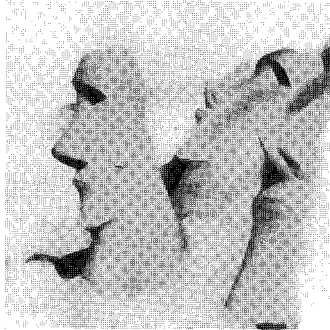


(a) 가보 변환 (b) 제시된 방법

그림 6: 가보 변환과의 비교 결과. 제시된 방법은 스타일 벡터의 일부만을 이용하여 영상의 구조적 경계를 효과적으로 처리한다.

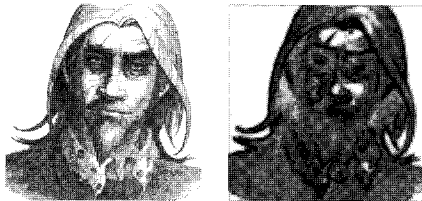


(a) 원본 영상 (b) 평탄도 맵



(c) 필터링 결과

그림 7: 이스터(Easter)석상 필터링 결과.



(a) 원본 영상 (b) 평탄도 맵

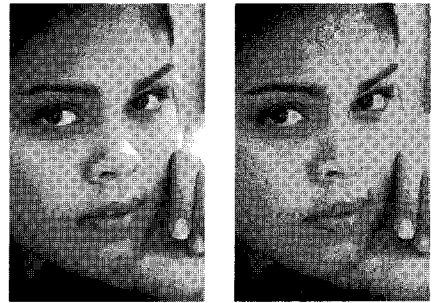


(c) 필터링 결과

그림 8: 펜화 필터링 결과.

#### 4. 결론 및 향후 연구

본 논문에서는 영상의 스타일 벡터를 계산하여 영상을 평탄화하는 방법을 제시하였다. 이를 위해 영상의 스타일을 정의한 후, 양방향 필터링으로 이를 제거하여 영상의 스타일이 제거된 결과를 얻어내었다. 이 때 영상의 경계 부분을 효과적으로 처리하기 위해 필터링 시에 스타일 벡터의 일부만을 이용하는 방법을 제시하였다. 또한 다중 해상 기법을 이용하여 영



(a) 원본 영상 (b) 스타일 전이 영상

그림 9: 스타일 전이 결과. 그림 8(a)의 펜 선이 전이되었다.

상의 큰 스타일에서 작은 스타일까지 효과적으로 제거할 수 있었다.

본 논문에서는 주어진 영상을 흑백 영상으로 변환시켜 스타일 벡터를 계산하였다. 그러나 색이 있는 영상에서 색 정보를 효과적으로 이용할 수 있다면 더 좋은 결과를 기대할 수 있을 것이다. 또한 스타일을 제거하는 과정에서 스타일의 구조를 파악하여 이를 다룰 수 있다면 텍스처 전이와 같은 응용 분야에서도 보다 더 효과적으로 사용할 수 있을 것이다.

#### 감사의 글

본 연구는 정보통신부 지정 게임 애니메이션 센터 (ITRC)의 재정지원을 통해 수행되었습니다.

#### 참고문헌

- [1] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 7, pp. 629–639, 1990.
- [2] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *ICCV '98: Proceedings of the Sixth International Conference on Computer Vision*, 1998, p. 839.
- [3] J. S. D. Bonet, "Multiresolution sampling procedure for analysis and synthesis of texture images," in *Proc. ACM SIGGRAPH 1997*, pp. 361–368, 1997.
- [4] O. G. Cula and K. J. Dana, "Compact representation of bidirectional texture functions," in *Proc. CVPR*, 2001, pp. 1041–1047.
- [5] B. S. Manjunath and W.-Y. Ma, "Texture features for browsing and retrieval of image data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 8, pp. 837–842, 1996.
- [6] Y. Qu, T.-T. Wong, and P.-A. Heng, "Manga colorization," *ACM Transactions on Graphics*, pp. 1214–1220, 2006.
- [7] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin, "Image analogies," in *Proc. ACM SIGGRAPH 2001*, pp. 327–340, 2001.