

방향성을 고려한 텍스처 합성을 학습하기 위한 인공지능망

추연희^o, 김종현^{*}

^o강남대학교 소프트웨어융합부,

^{*}인하대학교 소프트웨어융합대학 디자인테크놀로지학과

e-mail: jonghyumkiun@inha.ac.kr

Artificial Neural Networks for Learning Directional Texture Synthesis

Yeon Hee Choo^o, Jong-Hyun Kim^{*}

^oSchool of Software Application, Kangnam University,

^{*}College of Software and Convergence (Dept. of Design Technology), Inha University

● 요약 ●

본 논문에서는 텍스처 합성을 할 때 CNN을 사용하여 효율성을 높이고 방향을 고려하여 동적인 결과로 품질을 개선시킬 수 있는 방법을 제안한다. 자유로운 회전 각도로 방향성을 고려하여 동적인 결과물을 생성할 수 있도록 하였으며, 기존 접근법인 사각형 형태의 마스크 블록이 아닌 다양한 회전 각도를 고려하여 학습을 했기 때문에 텍스처 합성 과정에서 방향성 특징을 좀 더 잘 표현할 수 있다.

키워드: 텍스처 합성(Texture synthesis), 회전된 텍스처 이미지(Rotated texture image), 방향성(Orientation), CNN(Convolutional neural networks)

I. Introduction

텍스처는 이미지를 구성하는데 있어 필수적인 요소라고 할 수 있다. 하지만 이미지를 생성하는데 있어, 텍스처를 일일이 그려서 만드는 것은 단순 작업이며, 반복되는 패턴에서 표현되는 부자연스러운 움을 수작업으로 자유 것은 어려운 일이다. 이 문제를 해결하기 위해 작은 텍스처 샘플 이미지만으로 더 큰 텍스처 이미지를 생성하기 위한 텍스처 합성에 대한 연구는 여전히 활발히 진행되고 있다. 본 논문에서는 CNN(Convolutional neural networks)을 이용하여 회전 패턴이 고려된 자연스러운 텍스처 합성 이미지를 추출하는 방법을 제안한다.

II. Preliminaries

Alexei와 Freeman에서는 인공지능을 이용하지 않고 주어진 샘플 텍스처 이미지에서 부분적으로 Patch block 단위로 합성을 진행한다 [1]. 이전 Patch block과 겹치게 이어 붙이되, 겹치는 부분의 오차가 최소가 되는 Patch block을 선정하여 겹치는 부분의 최소비용 경로를 합성을 진행한다. 결과적으로 더 큰 이미지 합성을 단순히 이어 붙이는 방법보다 자연스러운 결과를 보여주지만, Patch block의 다음 Patch block을 찾을 때 매번 샘플 텍스처 이미지의 모든 부분에서 Patch를 찾아야 한다.

또한 샘플 텍스처 이미지가 반복성이 적은 이미지라면 이전 Patch block의 overlap 부분의 오차가 최소가 되는 block 자체가 하나밖에 없는 경우가 생길 가능성이 높으며 같은 Patch block만 계속 합성된다. 비슷한 문제로 다음 Patch block으로 참고할 수 있는 patch의 개수가 적어서 오차가 최소가 되는 block이라 할지라도 겹치는 부분이 유사성이 적어서 아무리 최소비용 경로를 계산해도 자연스러운 경계선이 생성되지 않아 이질감이 생기는 문제가 있다.

이전 연구에서는 노이즈 이미지를 생성하여 샘플 텍스처 이미지와 각 레이어마다 Gram matrix를 계산하여 CNN을 이용한 Gradient descent로 결과 텍스처 이미지를 업데이트하는 방법으로 합성을 했다 [2]. 패치 기반의 합성보다는 빠른 계산 속도를 보여주지만 샘플 텍스처 이미지와 크기가 같은 결과 이미지만 생성되며, 반복성이 적은 샘플 텍스처 이미지일 경우 결과 텍스처 이미지가 항상 비슷하게 생성되는 문제가 있다.

텍스처 합성 패턴의 다양성을 증가시키기 위하여 Park과 Kim에서는 샘플 텍스처 이미지를 회전시키는 방법을 제시했다[3]. 본 논문에서는 계산 효율을 위하여 CNN을 사용하고 있으며 패턴의 다양성과 자연스러운 텍스처 합성을 위해 회전을 사용하고 있다는 점에서는 Park과 Kim의 연구와 유사하지만[3], 샘플 텍스처 이미지 회전 시에 존재하지 않는 널 공간(Null space)에 대한 처리나 텍스처 합성

방법에 있어 차이가 있다.

III. The Proposed Scheme

1. Preprocessing

동적인 텍스처 합성 결과를 위하여 샘플 텍스처 이미지 1개를 자유로운 각도로 회전시켜 여러 회전 샘플 텍스처 이미지들을 생성한다. 회전 샘플 텍스처 이미지 N 개를 준비하기 위하여 1) 샘플 텍스처 개수에 따른 회전 각도를 계산하고, 2) 이미지의 중심을 기준으로 해당 각도만큼 N 번 회전시켜 각각 회전 샘플 텍스처 이미지 N 개를 생성한다.

이때, 회전되면서 이미지가 존재하지 않는 부분인 널 공간이 생기는데 이 부분을 폐쇄주기 위하여 각 회전마다 이미지가 존재하지 않는 마스크(Mask) 영역을 계산한다. 본 논문에서는 원활한 마스크 영역을 계산하기 위해 회전 샘플 텍스처 이미지는 높이와 폭의 길이가 같은 정사각형 형태로 가정한다. 마스크는 다음과 같이 구한다: 회전각 θ 대하여 기존 이미지와 회전 이미지의 겹치는 부분을 $Line_{1, \dots, 3}$ 이라고 하고, 이미지의 왼쪽 상단을 원점으로 하는 평면상에 기존 이미지와 회전 이미지가 있다고 가정한다. 가정된 장면에서 직선 $L1$ 과 $L2$ 의 방정식을 계산할 수 있고, 회전된 샘플 텍스처 이미지의 각 모서리를 포함하는 직선의 방정식을 이용하여 이미지가 존재하지 않는 부분을 마스크 영역으로 만들 수 있다.

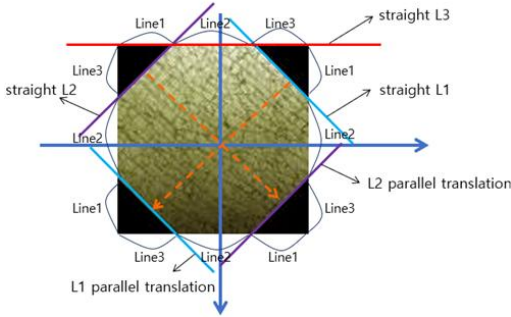


Fig. 1. The process of extracting null space regions from a rotated texture.

회전된 샘플 텍스처 이미지를 생성한 이유는 기존의 방법은 합성시 샘플 텍스처 이미지 하나만 참고하였기에 한 가지 방향에 국한되어 정적인 결과물이 생성됐기 때문에 샘플 텍스처 이미지를 회전시켜 다양한 방향성을 참고하여 여러 방향성을 지닌 패턴을 생성하기 위해서이다.

따라서 회전된 샘플 텍스처 이미지는 ‘샘플 텍스처 이미지를 회전시킨 이미지와 똑같진 않아도 유사하며, 회전된 방향성을 가지고 있으면 되기 때문에, 샘플 텍스처 이미지를 회전한 후 이미지가 존재하는 부분에 대해서만 patch block을 찾아서 참고하는 방식으로 회전 텍스처 이미지를 생성한다. 크기를 키운 결과물을 생성하기 위해 본 논문에서는 회전 샘플 텍스처 이미지를 생성하는 이미지 쿨링 합성 과정에서 크기를 미리 키운다.

2. Texture Synthesis

2.1. Artificial neural networks

본 논문은 Convolutional Neural Networks를 사용하는 부분은 이전 연구와 유사하다[2]. 이를 구현하기 위해 VGG-19 네트워크를 사용했으며, 16개의 컨볼루션 레이어와 5개의 풀링(Pooling) 레이어를 통해 얻은 Feature를 사용한다. 컨볼루션 필터의 크기는 $3 \times 3 \times k$ 이며, k 는 입력 특징 맵(Feature maps)의 개수이다. 스트라이드(Stride)와 패딩(Padding)은 1로 설정했고, 입력과 출력의 특징 맵은 크기가 같도록 설정했다. 필터 크기는 2×2 , 스트라이드는 2인 평균 풀링(Average pooling)을 사용하여 특징 맵을 $\frac{1}{2}$ 배로 만든다.

2.2 Texture synthesis

한 개의 회전 샘플 텍스처 이미지에 대하여 VGG-19 네트워크의 각 레이어마다 회전 샘플 텍스처 이미지와 노이즈 이미지의 Gram matrix를 각각 계산하고, 각 레이어마다 손실(Loss)를 계산한다. 모든 N 개의 회전 샘플 텍스처 이미지에 대하여 M 개의 레이어에서 각각 Gram loss들을 계산한다 (Gram loss는 $N \times M$ 개). 모든 Gram loss를 더한 전체 손실 L 을 픽셀 값에 대한 Gradient descent를 사용하여 노이즈 이미지를 업데이트한다. 최종적으로 모든 회전 이미지들의 방향 값들이 반영된 결과 텍스처 이미지를 얻을 수 있다.

텍스처 이미지 x 는 컨볼루션 필터를 거쳐 특징 맵들이 각각 레이어 l 에서 생성되며 이는 행렬 형태인 F^l 로 저장된다. F_{jk}^l 는 레이어 l 에서 j 번째 필터의 k 위치에 있는 특징 맵이다. G_{ij}^l 는 레이어 l 에서 특징 맵 j 와 j 에서의 Gram matrix 형태로 계산한다 (수식 1 참조).

$$G_{ij}^l = \sum_k F_{ik}^l F_{jk}^l \quad (1)$$

여기서 \vec{x} 와 \hat{x} 는 회전 텍스처 이미지와 노이즈 이미지이고, G^l 과 \hat{G}^l 은 레이어 l 에서의 Gram matrix이며, N_l 은 특징 맵의 개수이다. 또한 M_l 은 특징 맵의 크기이며, 손실함수(Loss function) E_l 과 전체 손실인 L 는 다음과 같이 계산한다 (수식 2 참조).

$$E_l = \frac{1}{4N_l^2 M_l^2} \sum_{i,j} (G_{ij}^l - \hat{G}_{ij}^l)^2 \quad (2)$$

$$L(\vec{x}, \hat{x}) = \sum_{l=0}^L w_l E_l \quad (3)$$

Fig. 2는 본 논문에서 제안하는 네트워크 개요이다. 회전 샘플 텍스처 이미지 하나를 기준으로 CNN을 거치면서 각 레이어에 L 대해 컨볼루션 필터를 거쳐 특징 맵이 만들어지고, 결과적으로 Gram matrix가 계산된다.

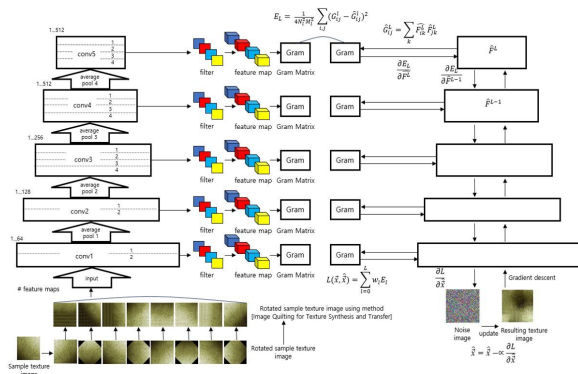


Fig. 2. Our framework.

모든 샘플 텍스처 이미지에 대해서 위 과정을 각각 실행하며 계산한 모든 E_L 를 더하여 전체 손실 함수 L 을 계산한다. 마지막으로 L 을 픽셀에 대해 Gradient descent를 적용하여 회전 샘플 텍스처 이미지의 Gram matrix와 같은 Gram matrix를 갖는 새로운 결과 텍스처 이미지가 노이즈 이미지로부터 업데이트돼 가면서 해를 찾이간다.

2.3 Hyperparameter

본 논문에서 제안한 방법을 실험할 때 사용한 하이퍼 파라미터는 다음과 같다.

Optimizer : L-BFGS
Loss Function : (MSE를 사용한) Gram Loss Function
Learning rate : 1 (pytorch의 optim 패키지의 LBFSG의 default 값)
Iteration : 1000
Epoch : 1
회전 샘플 텍스처 이미지 개수 : 8개

회전 샘플 텍스처 이미지 개수는 사용자가 설정할 수 있다. 하지만 8개보다 많으면 합성 시간이 오래 걸리고, 결과 품질에 대한 개선이 크게 향상되지 않았다. 여기서 8개라는 값은 실험적으로 찾은 값이다.

III. Results

Fig. 3과 4는 우리의 기법과 이전 기법을 비교한 결과이다. 회전을 통한 방향성이 고려되어 더 동적이고 자연스러운 결과를 얻을 수 있었다.

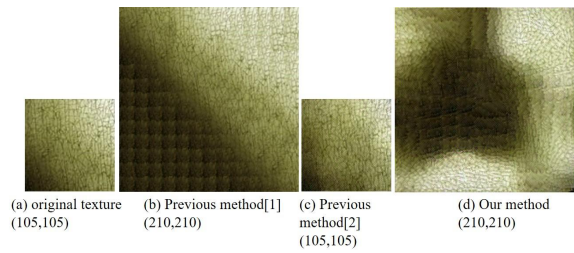


Fig. 3. Comparing the results of our method with previous method (scene1).

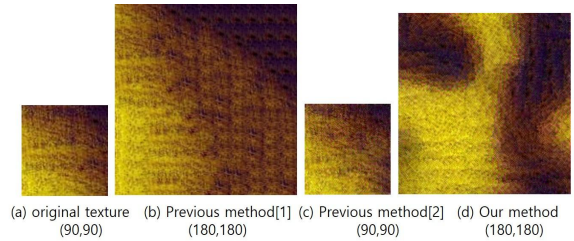


Fig. 4. Comparing the results of our method with previous method (scene2).

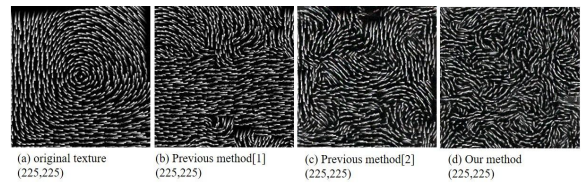


Fig. 5. Comparing the results of our method with previous method (scene3).

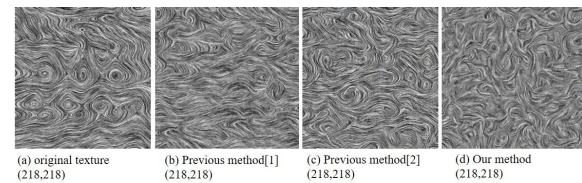


Fig. 6. Comparing the results of our method with previous method (scene4).

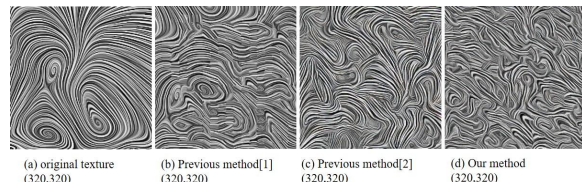


Fig. 7. Comparing the results of our method with previous method (scene5).

Fig. 5~7은 복잡한 패턴을 갖는 속도장에서 실험한 결과이다. 이전의 텍스처 합성보다 본 논문에서 제안하는 방법이 더 다양한 방향 패턴을 잘 표현해냈다. Table 1은 본 논문에서 제안한 방법과 이전 기법들 간의 이미지 합성 생성시간을 비교한 결과이다.

Table 1. Performance comparison of our method with previous method.

Method (hight,width)	Processing time (millisecond)
Our method – 8 rotation image generation taken times (210,210)	152420.34375
Our method – the time taken to run the CNN model (210,210)	40215.60546875
Previous method [2] – the time taken to run the CNN model (105,105)	21357.765625

IV. Conclusions

본 논문에서는 회전 텍스처 이미지를 CNN을 사용하여 방향성을 고려한 동적인 합성 결과를 효율적으로 생성하여 기존의 텍스처 합성 기법을 개선했다. 본 논문에서는 Gram loss function을 사용하여 기존 기법의 결과에서 보이는 노이즈가 완전히 없어지진 않는 문제가 있다[2]. 또한 회전 샘플 텍스처를 만들기 위하여 기존 이미지 쿼팅 기법으로 합성 시 이전 기법이 가지고 있는 문제가 회전 샘플 텍스처에 서도 나타날 수 있다[1]. 향후 이 문제를 개선하여 회전 텍스처 합성을 좀 더 안정적으로 수행될 수 있도록 알고리즘을 개선할 계획이다.

REFERENCES

- [1] Efros, Alexei A., and William T. Freeman. "Image quilting for texture synthesis and transfer." In Proceedings of the 28th annual conference on Computer graphics and interactive techniques, pp. 341-346. 2001.
- [2] Gatys, Leon, Alexander S. Ecker, and Matthias Bethge. "Texture synthesis using convolutional neural networks." Advances in neural information processing systems 28, 2015.
- [3] Han-Wook Park, Chang-Hun Kim, "Parallel Rotated Exemplar-based Texture Synthesis", Journal of the Korea Computer Graphics Society, Vol. 15, No. 1, pp. 17-23, 2009.