일반논문 (Regular Paper)
방송공학회논문지 제27권 제1호, 2022년 1월 (JBE Vol.27, No.1, January 2022)
https://doi.org/10.5909/JBE.2022.27.1.104
ISSN 2287-9137 (Online) ISSN 1226-7953 (Print)

초-고해상도 영상 스타일 전이

김 용 구[a)‡]

Super High-Resolution Image Style Transfer

Yong-Goo Kim[a)‡]

요 약

신경망 기반 스타일 전이 기법은 영상의 고차원적 구조적 특징을 반영하여 높은 품질의 스타일 전이 결과를 제공함으로써 최근 크게 주목받고 있다. 본 논문은 이러한 신경망 기반 스타일 전이의 GPU 메모리 제한에 따른 해상도 한계에 대한 문제를 다룬다. 신경망 출력이 가진 제한적 수용장 특징을 바탕으로, 부분 영상 기반의 스타일 전이 손실함수 경사도 연산이 전체 영상을 대상으로 구한 경사도 연산과 동일한 결과를 생성할 수 있을 것으로 기대할 수 있다. 이러한 아이디어를 기반으로, 본 논문에서는, 스타일 전이 손실함수의 각 구성 요소에 대한 경사도 연산 구조를 분석하고, 이를 통해 부분 영상의 생성 및 패딩에 대한 필요조건을 구하고, 전체 영상의 신경망 출력에 좌우되는 경사도 연산 요구 데이터를 확인하여 구조화함으로써 재귀적 초고해상도 스타일 전이 알고리즘을 개발하였다. 제안된 기법은, 사용하는 GPU 메모리가 처리할 수 있는 크기로 초고해상도 입력을 분할하여 스타일 전이를 수행함으로써, GPU 메모리 한계에 따른 해상도 제한을 받지 않으며, 초고해상도 스타일 전이에서만 감상할 수 있는 독특한 세부 영역의 전이 스타일 특징을 제공할 수 있다.

Abstract

Style transfer based on neural network provides very high quality results by reflecting the high level structural characteristics of images, and thereby has recently attracted great attention. This paper deals with the problem of resolution limitation due to GPU memory in performing such neural style transfer. We can expect that the gradient operation for style transfer based on partial image, with the aid of the fixed size of receptive field, can produce the same result as the gradient operation using the entire image. Based on this idea, each component of the style transfer loss function is analyzed in this paper to obtain the necessary conditions for partitioning and padding, and to identify, among the information required for gradient calculation, the one that depends on the entire input. By structuring such information for using it as auxiliary constant input for partition-based gradient calculation, this paper develops a recursive algorithm for super high-resolution image style transfer. Since the proposed method performs style transfer by partitioning input image into the size that a GPU can handle, it can perform style transfer without the limit of the input image resolution accompanied by the GPU memory size. With the aid of such super high-resolution support, the proposed method can provide a unique style characteristics of detailed area which can only be appreciated in super high-resolution style transfer.

Keyword: Deep Learning, Style Transfer, Super High-Resolution, Receptive Field, Gradient Descent

Ⅰ. 서 론

예술적 스타일 전이(artistic style transfer)는 예술 작품이 가진 인지적 특징(색상, 질감, 스트로크 등)을 다른 영상의 한 부분 혹은 전체 영상에 적용하는 방법이다. 이러한 예술적 스타일 전이 기법은 비전 분야에서는 텍스처 합성 연구[1]의 일환으로, 그리고 그래픽스 분야에서는 비사실적 렌더링 기법으로[2] 오랜 기간 연구되어 왔다. 이러한 전통적 스타일 전이 연구들은 대부분 낮은 수준의 통계적 특징을 파라미터로 사용하거나[3] 혹은 MRF(Markov Random Field) 기반의 비-파라미터 샘플링 방식[4]을 이용해 텍스처 모델링을 수행함으로써 스타일 전이를 수행한다. 하지만, 사용된 모델 및 샘플링 방식의 한계 때문에 영상이 가진 고차원적 구조적 특징을 반영하는데 종종 그 성능의 제한을 보여 왔다. 2016년 소개된 Gatys 등의 기념비적 스타일 전이 연구[5]는 인공 신경망의 다중 레이어 출력을 기반으로 텍스처(스타일) 모델링을 수행함으로써 전통적인 초기 연구들과는 대별되는 새로운 스타일 전이 방법을 제시하였다. 이 방법은 영상이 가진 고차원적 스타일 구조를 적절히 반영한 매우 높은 수준의 차별적 전이 품질을 제공함으로써 스타일 전이에 대한 높은 새로운 관심을 불러일으키고 있다.

[5]의 연구 결과를 기반으로 다양한 스타일 전이 후속 연구들이 추진되고 있는데, 이들은 크게 스타일 전이 결과 영상을 최적화하는 방식들과 스타일 전이를 수행하는 신경망을 최적화하는 방식들로 구분할 수 있다. 우선, 영상 최적화를 기반으로 하는 방법들은 신경망을 최적화하는 기법들에 비해 느리지만 우수한 전이 품질을 제공할 수 있는 특징을 가지며, MMD (Minimum Mean Discrepancy) 관점에서 스타일 모델의 의미를 해석하고, 이를 통해 다양한 스타일 손실 함수를 제안함으로써 스타일 전이 기법을 확장한 연구[6], 그램 행렬(Gram matrix) 기반으로 정의된 [5]의 스타일 손실 최적화에 대한 학습 불안정성 해소를 수행한 연구[7], 스타일 전이 결과의 다양성 및 안정성 확보를 위한 연구[8], 그리고 비디오 및 가상현실 미디어에 대한 스타일 전이를 시도한 연구[9,10] 등이 대표적이다. 반면, 전이 품질은 다소 떨어지지만 영상 최적화를 기반으로 한 방식들의 느린 속도 문제를 해결하기 위한 다양한 시도들이 추진되고 있는데, 특정 스타일에 대한 능률적 스타일 전이 신경망 구축을 수행한 연구들[11,12]로부터, 다수의 특정 스타일에 대한 스타일 전이 신경망 학습을 수행하는 연구[13,14], 그리고 입력되는 임의 스타일에 대한 스타일 전이 신경망 구축을 수행하는 연구[15-17] 등으로 발전하고 있다. 이러한 신경망 기반 스타일 전이 최신 기법들에 대한 보다 자세한 내용 및 성능 비교 결과를 [18]에서 확인할 수 있다.

하지만 이러한 신경망 기반 스타일 전이에 대한 기존 연구들을 통해 얻을 수 있는 스타일 전이 결과의 해상도는 매우 제한적이며, 고해상도 스타일 전이 문제를 다룬 연구들은 거의 없다. 12GB 공유 메모리를 장착한 최신 GPU (Graphic Processing Unit, 예를 들어, NVIDIA Titan-V 모델)를 사용할 때, [11-12]와 같이 특정 스타일 혹은 특정 스타일 범주에서 동작하는 매우 가벼운 심층 신경망이 제공할 수 있는 최대 스타일 전이 해상도는 4K-UHD(Ultra High Definition: 3,840x2,160) 수준이고, 임의 스타일에 대한 높은 전이 품질을 제공하는 것으로 평가받는 [5]의 경우에는 1,024 해상도 수준[1]의 결과 제공이 고작이다. 임의 스타일을 지원하는 스타일 전이의 해상도 문제는 [21]에서 최초로 다루어졌는데, 지식증류(Knowledge Distillation) 기법을 기반으로 [15-16]의 스타일 전이 네트워크 압축을 통해, 최대 40M 화소(10,240x4,096)의 스타일 전이에 성공하였다. 이러한 결과는 기존의 스타일 전이 해상도 수준을 크게 개선하였을 뿐 아니라 임의 스타일 영상을 지원한다는 점에서 높은 활용성을 제공하고 있지만, 지원 가능한 해상도가 사용하는 GPU의 메모리 크기에 절대적으로 제한받을

a) 서울미디어대학원대학교 인공지능응용소프트웨어학과(Dept. of AI Software Eng., Seoul Media Institute of Technology)
‡ Corresponding Author : 김용구(Yong-Goo Kim)
E-mail: ygkim@smit.ac.kr
Tel: +82-2-6393-3222
ORCID:https://orcid.org/0000-0002-8905-1984
※ This research is supported by Ministry of Culture, Sports and Tourism and Korea Creative Content Agency (Project Number: R2020040238).
· Manuscript received October 25, 2021; Revised December 12, 2021; Accepted December 12, 2021.

1) [19]에서 Gatys 등은 고품질의 고해상도 스타일 전이 출력을 얻기 위해, 학습에 사용되는 VGG(Visual Geometry Group)-19 네트워크[20]의 고정된 수용장 크기를 고려하여, 재귀적으로 낮은 해상도에서 출력 해상도까지 반복적 학습이 필요하다는 사실을 밝혔고, 이러한 재귀적 학습 결과로 1,024 해상도의 고품질 스타일 전이 결과를 제시하였다.

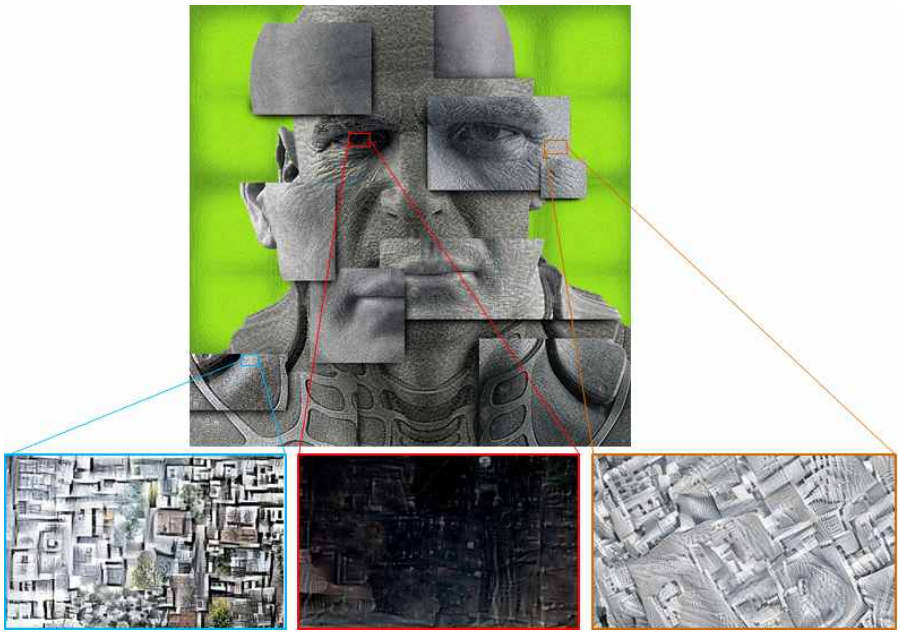

그림 1. 제안 방법을 이용한 30,000x30,000 해상도 스타일 전이 작품
Fig. 1. Style transfer artwork using the proposed method (30,000x30,000 resolution)

뿐 아니라, 기존 스타일 전이 심층 신경망을 압축하였기 때문에 기존 연구의 품질을 넘을 수 없다는 한계가 있다.

본 논문에서는, 이러한 신경망 기반 스타일 전이의 GPU 메모리에 따른 해상도 한계를 극복하고 고품질의 스타일 전이 수행이 가능할 수 있도록, 초고해상도 영상에 대한 [5]의 스타일 전이를 수행할 수 있는 구현 방법을 제안한다. 제안 방법은 장착된 GPU에서 처리 가능한 크기로 입력 영상을 파티션하고, 각 부분 입력 영상에 대한 경사도(gradient) 연산의 수행 결과가 전체 입력 영상에 대해 수행한 경사도 연산과 동일한 결과를 생성하기 위해 필요한 조건을 분석하여 이를 알고리즘으로 제시한 것이다. 또한, 이러한 조건을 실제로 구현할 수 있는 적절한 데이터 구조 제안하여, 이를 텐서플로우 모듈[24]을 통해 구현하고 그 결과를 제시하였다. 그림 1에 제안 방식을 구현한 텐서플로우 엔진을 사용하여, 아티스트와의 협업을 통해 생성한 초고해상도(30,000x30,000 해상도) 스타일 전이 결과를 도시하였다.

본 논문의 구성은 다음과 같다. 본 논문의 2장에서는 [5]의 스타일 전이 방식과 그 최적화 과정을 설명하고, 이를 통해 스타일 전이의 해상도에 따른 메모리 문제를 정의한다. 이 후, 본 논문의 3장에서는 이러한 문제를 해결하기 위한 분할 영상 기반 스타일 전이 최적화 방법과 그 구현에 대해서 상세히 설명한다. 본 논문의 4장에서는 고해상도 영상을 이용한 부분 영상 기반 스타일 전이 결과의 신뢰성 검증과, 다양한 초고해상도 실험 영상에 대한 스타일 전이 결과의 특징을 소개하고, 마지막으로, 5장에서 본 논문의 결론을 맺도록 한다.

Ⅱ. 최적화 기반 스타일 전이의 해상도 문제

Gatys 등의 스타일 전이 기법[5]은 영상 인식을 위해 개발된 VGG-19(Visual Geometry Group-19) 신경망[20]을 사용하여 스타일 영상이 가진 인지적 특징을 오브제 영상에 적용할 수 있는 최적화 방법을 개발한 것이다. 보다 구체적으로 설명하면, 그림 2에 도시된 구조를 가지는 기 학습된 VGG-19 신경망에 오브제 영상과 스타일 영상, 그리고 임의의 출력 영상을 통과시켜 각 레이어 출력을 얻고, 오브제 영상과 현재 결과 영상의 특정 레이어 출력들로부터 오브제 손실을 정의하고, 스타일 영상과 현재 결과 영상의 정해진 레이어 출력들로부터 스타일 손실을 정의한 후, 이 두

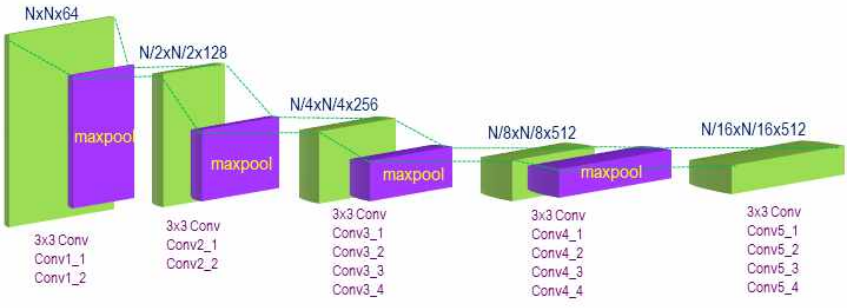

그림 2. VGG-19 네트워크의 [20] 레이어 구조
Fig. 2. Layer architecture of VGG-19 Network [20]

손실의 가중 합이 최소화 되도록 결과 영상을 생성하도록 한 것이다. 즉, $H_o \times W_o \times 3$ 크기의 RGB 오브제 영상과 $H_s \times W_s \times 3$ 크기의 스타일 영상, 그리고 스타일이 전이된 $H_o \times W_o \times 3$ 크기의 결과 영상을 각각 o, s, p라 하면, 식(1)에 보이는 바와 같이 o, p 사이에 정의되는 오브제 손실 L_o와 s, p 사이에 정의되는 스타일 손실 L_s의 가중 합을 최적화함으로써 Gatys 등이 제안한 스타일 전이 결과 p^*를 얻을 수 있다.

$$p^* = \min_p L(s,o,p),$$
$$L(s,o,p) = \lambda_{os} \cdot L_o(o,p) + L_s(s,p) \quad (1)$$

기 학습된 VGG-19 네트워크에 영상 x를 입력하여 얻을 수 있는 $H_j \times W_j \times C_j$ 크기의 j번째 레이어 출력을 $\phi_j(x)$라 하면, 식(1)의 오브제 손실과 스타일 손실은 다음과 같이 정의된다.

$$L_o(o,p) = \sum_{j \in O} \frac{1}{2} \| \phi_j(o) - \phi_j(p) \|_2^2, \quad (2)$$

$$L_s(s,p) = \sum_{j \in S} \lambda_j \left(\frac{1}{2H_j W_j C_j} \right)^2 \| G(\phi_j(s)) - G(\phi_j(p)) \|_F^2, \quad (3)$$

여기서 O와 S는 오브제 손실과 스타일 손실의 정의에 사용되는 ϕ의 레이어 집합을 의미하며([5]에서 각 레이어 집합은 $O = \{Conv4_2\}$, $S = \{Conv1_1, Conv2_1, Conv3_1, Conv4_1, Conv5_1\}$을 사용함), $\| \cdot \|_2$ 와 $\| \cdot \|_F$ 는 각기 2-노름(norm), 그리고 프로베니우스(Frobenius) 노름을 의미한다. 또한, 식(3)의 H_j, W_j, C_j는, [5]의 레이어 명칭에 따라 $j = Conv$X_Y(또는 $j = ReLu$X_Y)라 할 경우, $H_j = H_o/2^{X-1}$, $W_j = W_o/2^{X-1}$, $C_j = 64 \times 2^{X-1}$과 같고, $\phi_j(x)$를 $H_j W_j \times C_j$ 크기의 2차원 행렬로 재구성한 결과를 Φ_{jx}라 표시하면, 그램 행렬(Gram matrix)을 의미하는 $G(\phi_j(\cdot))$는 다음과 같이 표현할 수 있다.

$$G(\phi_j(x)) = \Phi_{jx}^T \Phi_{jx} \quad (4)$$

식(1)의 손실 최적화는 출력 영상에 대한 손실 함수 변화율을 기반으로 출력 영상을 반복 갱신함으로써 수행될 수 있는데, 현재 출력 영상을 $p^{(t)} = \{p_{i,j,k}^{(t)} | 1 \le i \le H_o, 1 \le j \le W_o, 1 \le k \le 3\}$라 하면, 경사 하강법(gradient descent)에 의해 갱신되는 다음 번 출력 영상은 다음과 같이 나타낼 수 있다.

$$p^{(t+1)} = p^{(t)} - \alpha^{(t)} \cdot g^{(t)}, \quad (5)$$

여기서 $\alpha^{(t)}$는 현재 출력 영상의 갱신을 위한 학습율을 의미하고, $g^{(t)}$는 현재 출력 영상 $p^{(t)}$에 대한 식(1)의 손실 함수 변화율을 나타내는데, 다음과 같이 출력 영상 $p^{(t)}$의 각 화소에 대한 편미분 집합으로 표현할 수 있다.

$$g(t) = \frac{\partial L(s,o,p^{(t)})}{\partial p_{i,j,k}^{(t)}} = \left\{ \frac{\partial L(s,o,p^{(t)})}{\partial p_{i,j,k}^{(t)}} \,\middle|\, i = 1,\ldots,H_o, j = 1,\ldots,W_o, k = 1,2,3 \right\} \quad (6)$$

식(6)에서 $\partial L(s,o,p^{(t)}) / \partial p_{i,j,k}^{(t)}$는 식(1)-식(3)의 정의에 따라 다음과 같이 나타낼 수 있다.

$$\frac{\partial L(s,o,p^{(t)})}{\partial p_{i,j,k}^{(t)}} = \lambda_{os} \cdot \frac{\partial L_o(o,p^{(t)})}{\partial p_{i,j,k}^{(t)}} + \frac{\partial L_s(s,p^{(t)})}{\partial p_{i,j,k}^{(t)}} \quad , \quad (7)$$

$$\frac{\partial L_o(o,p^{(t)})}{\partial p_{i,j,k}^{(t)}} = -\sum_{l \in O} \left[\phi_l(o) - \phi_l(p^{(t)})\right] \circ \frac{\partial \phi_l(p^{(t)})}{\partial p_{i,j,k}^{(t)}} \quad , \quad (8)$$

식(8)에서 ∘ 는 연산자 좌/우에 위치한 두 텐서의 항목별 곱 (element-wise multiplication)을 의미하고, 식(9)의 $G_{m,n}(\cdot)$은 그램 행렬 $G(\cdot)$의 m행 n열 원소를 말한다.

식(6)-식(9)로 주어지는 현재 결과 영상 $p^{(t)}$에 대한 경사도 연산은, 일반적으로 연산 및 메모리 효율이 높은 경사 역-전파(back-propagation) 방식[22]을 통해 이루어지는데, 이러한 경사 역-전파를 위해서는 각 레이어의 출력이 모두 저장되어 있어야 한다. 즉, [5]에서 정의한 식(6)의 연산을 위해서는 VGG-19 네트워크의 $Conv5\_1$ 레이어까지의 모든 출력이 저장되어야 하는데, 입력 영상의 화소 수가 N이라 하고 32-bit 부동소수점 연산을 사용한다고 가정했을 때, 이렇게 저장되어야 할 모든 출력의 메모리 요구량을 계산해보면 $2544N$ 바이트가 된다. 즉, $1{,}024 \times 1{,}024$ 해상도 영상을 입력으로 VGG-19 네트워크를 구동하여 $Conv5\_1$ 레이어 까지 모든 레이어의 출력을 저장하려면 대략 2.5GB의 메모리를 요구하게 되고, 오브제 영상과 스타일 영상에 대한 네트워크 구동, VGG-19 네트워크의 학습된 파라미터 및 최적화를 위한 중간 단계 연산 결과들과 파라미터들을 모두 고려하면, 12GB 공유 메모리를 가지는 최신 GPU를 사용하는 경우라도 1,024 해상도 정도의 영상에 대한 스타일 전이가 지원할 수 있는 최대 해상도가 된다.

Ⅲ. 분할 영상 기반 스타일 전이 손실 최적화 기법

본 장에서는, 앞서 설명한 최적화 기반 스타일 전이 방식의 해상도 한계를 극복하고 초고해상도 영상에 대한 스타일 전이를 가능하게 할 수 있는 방법에 대해 살펴본다. 이를 위해, 식(8)과 식(9)로 주어지는 오브제 손실과 스타일 손실에 대한 현재 출력 영상의 경사도 연산 구조의 특징을 자세히 살펴보고, 이를 통해 분할 영상의 입력을 통해 전체 영상의 손실 경사도를 얻을 수 있는 조건에 대해 알아본다. 그 후, 이러한 조건을 적용한 분할 영상 기반 초고해상도 영상의 스타일 전이 알고리즘에 대해 상세히 설명하기로 한다.

1. 오브제 손실 경사도의 연산 구조

식(8)의 맨 오른쪽에 있는 $\partial \phi_l(p^{(t)}) / \partial p_{i,j,k}^{(t)}$는 한 화소 $p_{i,j,k}^{(t)}$의 변화에 대한 스타일 전이 네트워크 출력 $\phi_l(p^{(t)})$의 변화율을 나타내는 $H_l \times W_l \times C_l$ 크기의 텐서를 의미한다. 즉, 현재 결과 영상 $p^{(t)}$에 있는 (i,j,k) 위치의 값 $p_{i,j,k}^{(t)}$가 1만큼 변할 때 $H_l \times W_l \times C_l$ 크기의 l-번째 VGG-19 네트워크 레이어 출력 $\phi_l(p^{(t)})$가 얼마나 변화하는지를 계산한 것인데, 이 변화율은 $p_{i,j,k}^{(t)}$에 영향 받는 $\phi_l(p^{(t)})$의 일정 영역[2]에서만 값을 가지게 되고 나머지 영역에 대해서는 모두 0이 되는 특징이 있다. 따라서 텐서 $\left[\phi_l(o) - \phi_l(p^{(t)})\right]$의 차원이 입력 영상 전체의 차원에 비례함에도 불구하고, 식(8)은 $p_{i,j,k}^{(t)}$의 변화에 영향 받는 $\phi_l(p^{(t)})$의 일부 영역만 보존될 수 있으면 전체 영상을 사용하는 경우와 동일한 변화율 연산을 수행할 수 있는 특징을 가진다.

이러한 특징을 기반으로, 이제부터 $H \times W \times 3$ 크기의 입력 영상 최 좌측 상단의 $N \times N \times 3$ 부분 영상을(여기서 N은 16의 배수이고, $N \ll H, W$ 로 가정함) VGG-19 네트워크에 통과시켰을 때, 그림 2의 $ConvX\_Y$ 레이어 출력 ($(N/Z) \times (N/Z) \times (64 \cdot Z)$ 크기, $Z = 2^{X-1}$)이 전체 영상을 입력으로 구한 값과 동일하게 되기 위한 조건을 생각해 보자. 우선, 각 레이어의 컨볼루션 연산은 패딩을 이용해

$$\frac{\partial L_s(s,p^{(t)})}{\partial p_{i,j,k}^{(t)}} = -\sum_{l \in S} \frac{\lambda_l}{2} \left(\frac{1}{H_l W_l C_l}\right)^2 \sum_{m,n} \left[G_{m,n}(\phi_l(s)) - G_{m,n}(\phi_l(p^{(t)}))\right] \cdot \left(\frac{\partial G_{m,n}(\phi_l(p^{(t)}))}{\partial p_{i,j,k}^{(t)}}\right) \quad (9)$$

2) 보다 자세히는, $\phi_l(p^{(t)})$의 각 특징 값 중에서 그 수용장(receptive field)이 $p_{i,j,k}^{(t)}$를 포함하게 되는 영역에서만 변화율 연산이 가능하게 된다.

입력 텐서의 가로, 세로 크기가 변화하지 않도록 수행되기 때문에 동일한 연산 결과를 얻기 위해서는 $N\times N\times 3$ 부분 영상의 오른쪽과 아래쪽에 원본 영상의 특정 영역까지를 포함시켜 전체 영상을 입력했을 때와 동일한 연산 결과가 생성될 수 있도록 해야 한다. VGG-19 네트워크의 모든 컨볼루션 연산이 동일한 크기의 3×3 커널을 사용한다는 점을 고려하면, 각 레이어 컨볼루션 연산의 입력 텐서는 맨 오른쪽과 아래쪽에 전체 영상을 입력했을 때 계산되는 입력 텐서와 동일한 1개씩의 값이 더 있어야 한다는 것을 알 수 있다. 즉, 입력 영상을 $(N+K_F)\times(N+K_F)\times 3$이 되도록 하고 각 컨볼루션 연산의 맨 오른쪽과 아래쪽에는 패딩을 적용하지 않도록 했을 때, K_F를 얼마로 하면 $N/Z\times N/Z\times 64Z$ 크기의 $ConvX\_\mathrm{Y}$ 레이어 출력을 얻을 수 있는지를 계산하면 된다. 이러한 계산을 통해 얻을 수 있는 K_F의 값은 다음의 식(10)과 같이 표현할 수 있다.

$$K_F=\sum_{L=1}^{X}2^{L-1}\cdot S,\quad S=\begin{cases}Y \text{ if } L=X\\ 2 \quad else \text{ if } L=\{1,2\}\\ 4 \quad else\end{cases} \tag{10}$$

다음으로, 입력 영상의 최 좌측 상단의 위치 대신, 임의 위치에 있는 $N\times N\times 3$ 부분 영상에 대한 VGG-19 네트워크의 순방향 패스를 생각해보자. 이 경우에는 기본적으로 해당 부분 영상의 상, 하, 좌, 우에 식(10) 크기의 원본 영상 패딩이 필요하게 되며, 동시에 한 가지 조건이 더 추가되어야한다. 추가되어야 하는 조건은 그림 2의 각 컨볼루션 그룹 사이에 위치한 maxpool 레이어 때문인데, 패딩된 부분 영상의 $ConvX\_\mathrm{Y}$ 레이어 출력이 전체 영상을 입력으로 구한 해당 레이어 출력의 일부 영역(부분 영상의 위치에 대응되는 영역)과 동일한 값을 가질 수 있도록 하려면, 부분 영상의 시작 위치와 그 크기 N 이 2^{X-1}의 배수가 되어야하는 것이다. 이는 X 번째 컨볼루션 그룹까지의 모든 레이어 연산이 거치게 되는 $X-1$번의 2×2 maxpooling 연산이 부분 영상과 전체 영상 입력에서 동일한 경계 위치를 가질 수 있도록 하는 조건이 되고, 따라서 이 조건은 다시 부분 영상에 적용할 패딩 크기 K_F에도 적용되어야 한다. 이러한 고려를 종합하여, $H\times W\times 3$ 크기를 가지는 입력 영상의 $(S_H, S_W, 0)$ 위치에서 시작하는 크기 $N\times N\times 3$ 부분 영상을 VGG-19 네트워크에 입력하여, 입력 영상 전체를 입력한 경우와 동일한 $ConvX\_\mathrm{Y}$ 레이어 출력을 얻기 위한 조건은 다음의 식(11) 및 식(12)로 나타낼 수 있다.

$$\begin{aligned}&S_H,\ S_W,\ N \text{ 은 } 2^{X-1}\text{의 배수,}\\ &K_F \text{ 크기의 원본 영상 패딩을 수행}\end{aligned} \tag{11}$$

$$\begin{aligned}&K_F=CEIL_{2^{X-1}}\left(\sum_{L=1}^{X}2^{L-1}\cdot S\right),\\ &S=\begin{cases}Y \text{ if } L=X\\ 2 \quad else \text{ if } L=\{1,2\}\\ 4 \quad else\end{cases}\end{aligned} \tag{12}$$

식(12)에서 $CEIL_{2^{X-1}}(x)$는 x보다 큰 2^{X-1}의 배수로 올림을 수행하는 연산을 의미한다.

이제부터, 이러한 조건을 만족하는 하나의 부분 영상에 대해 VGG-19 네트워크의 $ConvX\_\mathrm{Y}$ 레이어 출력을 얻고, 이 출력으로 정의되는 임의의 손실 함수(예를 들어, 식(8)의 $L_o(o,p^{(t)})$)에 대한 입력 부분 영상의 편미분을 구했을 때, 그 결과(패딩 영역을 제외한 크로핑 결과)가 입력 영상 전체를 네트워크에 통과시켜 구한 편미분 결과(부분 영상에 대응되는 영역의 결과에 국한)와 동일하게 되기 위한 조건을 살펴보자. 우선, 이러한 입력에 대한 손실의 경사도 연산 결과는 정의된 손실 함수에 대한 $ConvX\_\mathrm{Y}$ 레이어 출력의 편미분 값을 계산한 후 그 값을 각 레이어를 통해 역-전파함으로써 얻을 수 있는데, 여기서 각 레이어를 통한 편미분 역전파가 순방향 컨볼루션 커널을 이용한 역 합성곱(transposed convolution) 연산을 통해 이루어진다는 점에 주목하자. 그러면, 오차의 $ConvX\_\mathrm{Y}$ 출력에 대한 최종 계층 경사도가 해당 레이어 하위의 모든 컨볼루션 레이어에서 3×3 커널을 이용한 역-합성곱 연산을 통해 역-전파되기 때문에, 부분 영상으로부터 얻은 $ConvXY$ 레이어의 순방향 결과에 몇 개의 정상적인 특징 값이 더 필요한지를 구하는 문제가 전체 영상을 입력으로 했을 때와 동일한 부분 영상 기반 경사도 결과를 얻기 위한 역-전파의 조건이 된다. 이 조건은 식(10)의 유도에서 고려했던 방법을 그대로 답습하여, 총 $CEIL_1\left(\sum_{L=X}^{1}2^{L-X}\cdot S\right)$ 개의 $ConvXY$ 샘플이 식(11) 및 식(12)를 고려한 부분 영상 입력 결과의 상, 하, 좌, 우 경계에 추가되어야 함을 요구하며, $ConvX\_\mathrm{Y}$ 출

력에서 하나의 샘플 간격이 입력 해상도에서 2^{X-1} 간격에 해당함을 고려하면, 결과적으로 입력 영상에 식(13) 크기의 패딩을 추가함으로써 이 조건을 만족시킬 수 있다.

$$K_B = 2^{X-1} \cdot CEIL_1\left(\sum_{L=X}^{1} 2^{L-X} \cdot S\right) \tag{13}$$

따라서 GPU의 메모리 제한 때문에 전체 영상을 입력하여 식(8)의 경사도 연산이 불가능한 경우, GPU의 메모리가 수용할 수 있는 최대 크기로 식(11)-식(13)의 조건을 고려하여 입력 영상을 분할하고 분할된 부분 영상에 대한 경사도 연산의 수행 결과를 모아 전체 영상 입력에 대응하는 식(8)의 경사도 결과를 얻을 수 있다.

2. 스타일 손실 경사도의 연산 구조

다음으로, 식(9)에 표현된 현재 결과 영상 $p^{(t)}$에 있는 (i,j,k) 위치의 값 $p_{i,j,k}^{(t)}$에 대한 스타일 손실 $L_s(s,p^{(t)})$의 변화율 연산에 대해 알아보자. 이 변화율은 앞서 설명했던 식(8)의 오브제 손실 경사도 연산과는 달리 패딩된 부분 영상 입력만으로는 계산될 수 없는 특징을 보이는데, 그 이유는 식(9)의 맨 오른쪽에 있는 변화율 $\partial G_{m,n}\left(\phi_l(p^{(t)})\right)/\partial p_{i,j,k}^{(t)}$는 $p_{i,j,k}^{(t)}$가 $\phi_l()$에 영향을 미치는 수용장 영역의 변화에 의해 결정할 수 있는 스칼라 값이지만 여기에 곱해지는 오차 항 $\left[G_{m,n}(\phi_l(s)) - G_{m,n}(\phi_l(p^{(t)}))\right]$은 $\phi_l()$의 모든 출력에 영향을 받는 값이기 때문이다. 하지만, 식(4)로 주어지는 j번째 레이어 출력에 대한 그램 행렬 $G(\phi_j(x))$의 차원은 그 입력 x의 크기와는 상관없이 해당 레이어의 출력 채널 수 $C_j = 64 \times 2^{X-1}$에만 영향 받으며, VGG-19 네트워크에서는 그 최대 크기가 512로 제한되어 있어서 스타일 손실 정의에 사용되는 식(3)의 S에 속하는 모든 레이어에 대한 각 부분 영상 별 그램 행렬을 모두 저장하는데 요구되는 메모리의 크기가 크지 않다는 특징이 있다. 예를 들어, 입력 영상을 100개의 부분 영상으로 나누어 처리하는 경우, [5]에서 사용한 S에 속하는 모든 레이어의 그램 행렬을 저장하기 위해 필요한 메모리는 $100 \times (64 \times 64 + 128 \times 128 + 256 \times 256 + 2 \times 512 \times 512) \times 4 = 198.0M$ 바이트가 된다(4바이트 부동소수점 표현을 가정). 따라서 현재 출력 영상 $p^{(t)}$의 각 부분 영상에 대해 스타일 손실의 정의에 필요한 모든 레이어의 그램 행렬을 저장하고, 각 부분 영상을 다시 입력하면서 저장된 정보를 이용해 식(9)의 스타일 손실 경사도 연산을 수행함으로써 부분 영상을 기반으로 계산된 식(9)의 스타일 손실 경사도 연산 결과가 전체 입력 영상을 대상으로 수행된 스타일 손실 경사도와 동일한 값이 되도록 할 수 있다.

보다 구체적으로, 식(11)을 만족하도록 구성한 i번째 부분 영상을 $P_i(p^{(t)})$라 하고 여기에 식(12) 및 식(13)을 고려하여 패딩을 수행한 부분 영상을 $Pad_{K_F+K_B}\left[P_i(p^{(t)})\right]$라 했을 때, 각 부분 영상을 입력으로 다음을 생성한다.

$$G_{i,j}^{p^{(t)}} = G\left(\phi_j\left(Pad_{K_F+K_B}\left[P_i(p^{(t)})\right]\right)\right), \tag{14}$$

$$G_j^{p^{(t)}} = \sum_i G\left(Pad_{K_F+K_B}^{-1}\left\{\phi_j\left(Pad_{K_F+K_B}\left[P_i(p^{(t)})\right]\right)\right\}\right), \tag{15}$$

여기서 $G(\cdot)$는 식(4)의 그램 행렬 생성자를 의미하고, $j \in S$이며(식(3) 참고), 식(15)의 $Pad_{K_F+K_B}^{-1}\{\cdot\}$는 패딩 영역을 제외하는 슬라이싱 연산을 의미하는 것으로 $\phi_j\left(Pad_{K_F+K_B}\left[P_i(p^{(t)})\right]\right)$에서 $P_i(p^{(t)})$에 대응되는 특징 값만을 추려내는 역할을 한다. 전체 영상을 입력으로 생성한 $\phi_j(p^{(t)})$의 그램 행렬이 $\phi_j(p^{(t)})$를 임의의 크기로 파티션 한 각 부분 텐서들로부터 구한 부분 그램 행렬의 합과 같아진다는 점에 주목하면, 식(15)의 값은 각 부분 영상들로부터 구한 전체 영상 $p^{(t)}$의 j번째 레이어 그램 행렬이 된다. 이렇게 구한 식(14) 및 식(15)는 $Pad_{K_F+K_B}\left[P_i(p^{(t)})\right]$를 다시 VGG-19 네트워크에 입력할 때 $G_{i,j}^{T-p^{(t)}} = G_j^{p^{(t)}} - G_{i,j}^{p^{(t)}}$를 계산하여 상수 행렬로 함께 입력하는데 사용한다. 즉, 식(3)의 $G(\phi_j(p))$를 $G_{i,j}^{T-p^{(t)}} + G_{i,j}^{p^{(t)}}$로 대체하여 식(9)의 스타일 손실 경사도를 구함으로써 전체 영상을 입력하여 구한 식(9)의 경사도 중 부분 영상 $P_i(p^{(t)})$에 대응하는 스타일 손실 경사도를 얻을 수 있게 된다.

3. 분할 영상 기반 스타일 전이 손실 최적화 기법

앞 절에서 설명한 부분 영상 기반 오브제 손실과 스타일

손실에 대한 경사도 생성 방법을 이용하여, 입력된 전체 영상에 대한 스타일 전이 손실 최적화를 수행할 수 있는 방법에 대해 살펴보자. 본 절에서는 해상도 $H_o \times W_o \times 3$을 가지는 오브제 영상 o와 현재 출력 영상 $p^{(t)}$, 그리고 해상도 $H_s \times W_s \times 3$을 가지는 스타일 영상 s가 입력되었을 때, 현재 출력 영상 $p^{(t)}$에 대한 식(1)의 스타일 전이 손실 $L(s, o, p)$이 가지는 변화율 $g^{(t)} = \partial L(s, o, p)/\partial p^{(t)}$를 패딩된 부분 영상을 기반으로 생성하는 상세 알고리즘을 설명한다. 그 전체 과정은 1) 스타일 영상의 파티션 기반 스타일 손실 기준 값 생성, 2) 현재 출력 영상의 파티션 기반 부분 그램 행렬 생성, 그리고 3) 현재 출력 영상과 오브제 영상의 동일 파티션 입력을 대상으로 스타일 전이 손실 부분 경사도를 생성하는 것으로 이루어진다(그림 3 참조).

우선, 스타일 영상의 파티션 기반 스타일 손실 기준 값 생성은 스타일 영상 전체에 대한 그램 행렬, 즉 식(3)의 $G(\phi_j(s))$, $j \in S$를 생성하는 것인데, 식(5)로 표현된 현재 출력 영상 갱신이 반복적으로 수행되는 경우에도 한 번만 구하면 되는 값이기 때문에 따로 처리하도록 한다. 이 기준 값의 생성은 입력된 스타일 영상 s의 해상도와 식(11) 및 식(12)를 고려하여 패딩된 부분 영상 $Pad_{K_F}[P_i(s)]$를 얻는 것으로 시작된다(여기서, $i = 0,...,N_S-1$). 이러한 각 패딩된 부분 영상을 GPU에 입력하여 다음과 같이 스타일 손실 레이어 별 그램 행렬을 얻고,

$$G_{i,j}^{s} = G\left(Pad_{K_F}^{-1}\left\{\phi_j\left(Pad_{K_F}[P_i(s)]\right)\right\}\right), \quad i = 0,...,N_s-1,\ j \in S \tag{16}$$

CPU상에서, 이렇게 구해진 각 부분 영상에 대한 그램 행렬을 합하여 다음과 같이 스타일 손실 레이어 별 손실 기준 값을 생성한다.

$$G_j^s = \sum_{i=0}^{N_s-1} G_{i,j}^s \ , j \in S \tag{17}$$

그 후, 사용하는 GPU에서 처리할 수 있는 스타일 전이 최대 해상도를 고려하여 생성된 각 패딩된 현재 출력 영상 $Pad_{K_F+K_B}[P_i(p^{(t)})]\,(i = 0,...,N_o)$를 GPU에 입력하여 각 부분 영상에 대한 스타일 손실 레이어 별 그램 행렬을 얻고(식(14) 참고), CPU상에서 식(15)의 덧셈 연산을 수행하여 스타일 손실 레이어 별 현재 출력 영상에 대한 스타일 값을 생성한다. 즉, 이와 같은 현재 출력 영상의 파티션 기반 부분 그램 행렬 생성 과정을 통해서 CPU가 사용하는 메모리에는 식(14)의 각 부분 영상에 대한 스타일 손실 레이어 별 그램 행렬 $G_{i,j}^{p^{(t)}}\,(i = 0,...,N_o,\ j \in S)$와 스타일 손실 레이어 별 그램 행렬의 총 합 $G_j^{p^{(t)}}$를 기록하게 된다.

마지막으로, 동일한 파티션과 패딩을 적용한 부분 현재 출력 영상 $Pad_{K_F+K_B}[P_i(p^{(t)})]$와 부분 오브제 영상 $Pad_{K_F+K_B}[P_i(o)]$[3], 그리고 $G_{i,j}^{T-p^{(t)}} = G_j^{p^{(t)}} - G_{i,j}^{p^{(t)}}$, G_j^s $(j \in S)$를 GPU에 입력하여 다음과 같이 부분 영상에 대한 스타일 전이 손실 L을 구한다.

그 후, 입력된 현재 출력 부분 영상에 대한 식(18)의 스타일 전이 손실 변화율을 GPU상에서 계산하여 CPU에 출력하고, 이러한 부분 영상 기반 스타일 손실 경사도를 다음과 같이 역 패딩 및 역 파티션 하여 전체 입력 영상에 대한 스타일 손실 경사도를 얻은 후, 식(5)와 같은 갱신 방법을 통해 현재 출력 영상을 고도화한다.

$$L\left(Pad[P_i(o)],\, Pad\left[P_i(p^{(t)})\right]\right) = \lambda_{os} \cdot L_o\left(Pad[P_i(o)],\, Pad\left[P_i(p^{(t)})\right]\right) + L_s(p^{(t)}) \tag{18}$$

$$L_o\left(Pad[P_i(o)],\, Pad\left[P_i(p^{(t)})\right]\right) = \sum_{j \in O} \frac{1}{2} \left|\left|\phi_j(Pad[P_i(o)]) - \phi_j\left(Pad\left[P_i(p^{(t)})\right]\right)\right|\right|_2^2 \tag{19}$$

$$L_s\left(Pad\left[P_i(p^{(t)})\right]\right) = \sum_{j \in S} \lambda_j \left(\frac{1}{2H_jW_jC_j}\right)^2 \left|\left|G_j^s - G_{i,j}^{T-p(t)} - G\left(\phi_j\left(Pad\left[P_i(p^{(t)})\right]\right)\right)\right|\right|_F^2 \tag{20}$$

3) 앞으로, 순방향과 역방향 패딩을 모두 포함하는 연산자, $Pad_{K_F+K_B}[\,\cdot\,]$,를 간단히 $Pad[\,\cdot\,]$로 사용하기로 한다.

$$g^{(t)} = \sum_{i=0}^{N_o-1} P_i^{-1}\left(Pad^{-1}\left\{ \frac{\partial L\left(Pad\left[P_i(o)\right], Pad\left[P_i(p^{(t)})\right]\right)}{\partial\, Pad\left[P_i(p^{(t)})\right]} \right\}\right) \quad (21)$$

이상에서 설명한 분할 영상 기반 경사도 연산 알고리즘을 정리하면 그림 3과 같다. 그림에서, 각 변수 및 연산의 위치가 GPU상에 존재하는지 CPU상에 존재하는지를 표기한 점에 주목하자.

4. 초고해상도 영상 스타일 전이 알고리즘

앞 절에서 설명한 분할 영상 기반 스타일 전이 손실 최적화 기법을 고해상도 입력 영상에 직접 사용하는 것은 적절하지 않다. 이는 [19]에서 지적하였듯이, VGG-19 네트워크의 고정된 수용장 크기로 인해 특정 해상도 이상의 영상을 대상으로 식(1)의 최적화를 수행하게 되면 영상의 매우 작은 국부 영역으로 스타일 반영이 제한되어 스타일 전이가 잘 되지 않는 것처럼 보일 수 있기 때문이다. 그림 4에 이와 같은 현상을 도시하였는데, 그림 4.(c)의 스타일 전이 결과를 보면 그림 4.(a)의 오브제 영상에 그림 4.(b)의 스타일이 적절히 반영된 것으로 보이지 않는다. 하지만 그 일부 영역을 크게 확대해보면, 그림 4.(e)와 같이 매우 세밀하게 스타일 영상의 텍스처가 반영되어 있는 것을 볼 수 있다. 이러한

Algorithm: Gradient Calculation for Style Transfer based on Partitioned Input

Input: $o\ (H_o \times W_o \times 3)$, $p^{(t)}\ (H_o \times W_o \times 3)$, $s\ (H_s \times W_s \times 3)$

Output: $g^{(t)}\ (H_o \times W_o \times 3)$

1. Reference for Style Loss

Generate $P_i(s), i = 0, \dots, N_s$ [CPU: eqn.(11)-eqn.(12)]

$G_j^s = 0$ [CPU]

For $i = 0;\ i < N_s;\ i{+}{+}$

$G_{i,j}^s$ [CPU] $= G\left(Pad_{K_F}^{-1}\left\{\varphi_j\left(Pad_{K_F}[P_i(s)]\right)\right\}\right)$ [GPU: eqn.(16)]

G_j^s += $G_{i,j}^s$ [CPU: eqn.(17)]

2. Preprocessing for Style Loss Calculation

Generate $P_i\left(p^{(t)}\right), i = 0, \dots, N_0$ [CPU: eqn.(11)-eqn.(13)]

$G_j^{p^{(t)}} = 0$ [CPU]

For $i = 0;\ i < N_o;\ i{+}{+}$

$G_{i,j}^{p^{(t)}}$ [CPU] $= G\left(\phi_j\left(Pad\left[P_i\left(p^{(t)}\right)\right]\right)\right)$ [GPU: eqn.(14)]

$G_j^{p^{(t)}}$ [CPU] += $G\left(Pad^{-1}\left\{\phi_j\left(Pad\left[P_i\left(p^{(t)}\right)\right]\right)\right\}\right)$ [GPU: eqn.(15)]

3. Gradient Calculation

Generate $P_i(o), P_i\left(p^{(t)}\right), i = 0, \dots, N_0$ [CPU: eqn.(11)-eqn.(13)]

For $i = 0;\ i < N_o;\ i{+}{+}$

$L_o = \sum_{j \in O} \frac{1}{2}\left\|\phi_j(Pad[P_i(o)]) - \phi_j\left(Pad\left[P_i\left(p^{(t)}\right)\right]\right)\right\|_2^2$ [GPU: eqn.(19)]

$L_s = \sum_{j \in S} \lambda_j \left(\frac{1}{2H_j W_j C_j}\right)^2 \left\| G_j^s - G_j^{T-p^{(t)}} - G\left(\phi_j\left(Pad\left[P_i\left(p^{(t)}\right)\right]\right)\right)\right\|_F^2$ [GPU: eqn.(20)]

$L = \lambda_{os} \cdot L_o + L_s$ [GPU: eqn.(18)]

$g^{(t)}[i]$ [CPU] $= Pad^{-1}\left\{\frac{\partial L}{\partial\, Pad\left[P_i\left(p^{(t)}\right)\right]}\right\}$ [GPU: eqn.(21)]

그림 3. 부분 영상 기반 스타일 전이 손실 경사도 연산

Fig. 3. Gradient Calculation of Style Transfer Loss over the current output image based on partitioned inputs

그림 4. 오브제 영상 입력 해상도에 따른 스타일 전이 문제
Fig. 4. A Problem of Style Transfer from Object Image Resolution

특징이 VGG-19 네트워크의 고정 수용장 크기 때문에 발생하는 현상이다.

이러한 문제는 스타일 전이가 적절하게 수행될 수 있는 낮은 해상도로 오브제 영상의 해상도를 변경한 후 초기 스타일 전이를 수행하고, 점차 높은 해상도로 변경하면서 이전 해상도에서 수렴된 출력 영상을 다음 해상도의 초기 출력 영상으로 지정함으로써 해결할 수 있다. 또한, 입력 영상의 다양성에 따른 출력 영상 학습의 수렴 안정성을 확보하기 위해, 본 논문에서는 식(5)의 단순 경사 하강법 대신 CPU상에서 Adam 최적화기[23]를 구현하여 사용하였고, 보다 빠른 수렴을 위한 적응적 초기 학습률 지정 및 조기종료 방법을 사용하였다. 이러한 초고해상도 입력 영상의 스타일 전이를 위한 재귀적 학습 최적화 알고리즘을 그림 5에 도시하였다.

그림에서 적응적 초기 학습률 지정은 각 해상도의 첫 학습에서 최대 화소 갱신이 12.8만큼 이루어지도록 설정하였고, 최대 화소 갱신은 tpc(target pixel change) 파라미터를 통해 조절할 수 있도록 하였는데 이 초기 갱신 정도와 Adam 최적화기의 파라미터 $\beta_1 = 0.9$, $\beta_2 = 0.99$는 빠른 학습과 안정적인 수렴에 상호 영향을 주는 초-매개변수(hyper-parameter)이다. 또한, 한 번의 경사도 연산을 통한 평균 화소 갱신양이 0.1보다 적어지게 되면 더 이상 학습을 수행하지 않고 조기종료 하도록 설정하였는데, 여기서도 $stop\_condition$ 파라미터를 통해 조기종료 조건을 조절할 수 있도록 하였고 최대 학습 반복 횟수 N_{tr}은 100으로 설정하였다. 이러한 재귀적 알고리즘을 이용하여 그림 4의 오브제 영상과 스타일 영상에 대해 스타일 전이를 수행한 결과를 그림 6에 도시하였다. 적용된 초기 학습률 설정과 조기

Algorithm: Recursive Style Transfer for Super High Resolution Images

Input: $o\ (H_o \times W_o \times 3)$, $s\ (H_s \times W_s \times 3)$, r_o- initial target resolution for object image

Output: $p^{(*)}\ (H_o \times W_o \times 3)$

$N = \max[\text{ceil}(\log_2 \max(H_o, W_o)/r_o), \text{ceil}(\log_2 \max(H_s, W_s)/r_o)]$ (Iteration Number)

For $i = 0;\ i < N;\ i{+}{+}$

$o_{size} = \min[r_o \times 2^i, \max(H_o, W_o)]$, $s_{size} = \min[r_o \times 2^i, \max(H_s, W_s)]$ (Input Resolution)

$o_i = \text{resize}(o, o_{size})$, $s_i = \text{resize}(s, s_{size})$, $p_i^{(0)} = \text{resize}\left(p_{i-1}^{(*)}, o_{size}\right), p_0^{(0)} = o_i$ (Resize)

$\hat{g} = \hat{v} = 0$, $\alpha = None$

For $j = 0;\ j < N_{tr};\ j{+}{+}$

$g = \text{gradient\_calc}\left(o_i, s_i, p_i^{(j)}\right)$ (Gradient Calculation using fig.3)

Adam Optimizer

$\hat{g} = \beta_1 \cdot \hat{g} + (1-\beta_1) \cdot g$, $\hat{v} = \beta_2 \cdot \hat{v} + (1-\beta_1) \cdot g^2$ ($\beta_1 = 0.9$, $\beta_2 = 0.99$)

$update = \hat{g}/\sqrt{\hat{v} + \varepsilon}$ ($\varepsilon = 10^{-6}$), $bias\_cor = \sqrt{\left(1-\beta_2^{j+1}\right)}/\left(1-\beta_1^{j+1}\right)$

if α is $None$

$\alpha = tpc/(bias_{cor} \times max(|update|))$ (Initial Learning Rate: $tpc = 12.8$)

$p_i^{(j+1)} = p_i^{(j)} - \alpha \cdot bias\_cor \times update$

if $mean(|update|) < stop\_condition$ ($stop\_condition$: 0.1)

$break$

$p_i^{(*)} = p_i^{(j+1)}$

그림 5. 초고해상도 영상에 대한 재귀적 스타일 전이 알고리즘

Fig. 5. Recursive Style Transfer Algorithm for Super High Resolution Images

그림 6. 재귀적 스타일 전이 알고리즘의 수행 결과 예시

Fig. 6. A Style Transfer Result using the Recursive Algorithm in Fig.5

종료 방법에 힘입어, 512 해상도에서 시작한 4K 해상도 영상의 스타일 전이에는 총 5258.7 초의 시간이 소요되었고, 최종 해상도의 분할 블록 수(그림 3의 N_o)는 12가 사용되었다[4]. 그림의 결과를 살펴보면 전반적인 스타일 전이의 수준 및 품질은 그 초기 해상도인 512 해상도의 결과와 동일하고, 그 세부 영역에는 스타일 영상의 질감 및 붓의 터치 등 고해상도 스타일 특징이 적절하게 반영된 것을 확인할 수 있다.

5. 알고리즘 복잡도 분석

본 절에서는 그림 5의 재귀적 스타일 전이 알고리즘에 대한 수행시간 기준 복잡도를 살펴본다. 알고리즘의 수행시간 기준 복잡도는 알고리즘 수행을 위해 필요한 곱하기나 나누기 같은 단위 연산의 수에 대한 분석과, CPU의 메모리 사용에서 발생하는 캐시 미스(cache miss)나 페이지 폴트(page fault)에 대한 자세한 분석이 필요하다. 뿐만 아니라, 그림 5와 같이 알고리즘이 적응적으로 동작하는 경우에는, 입력에 따라 초기 학습률이나 조기종료 조건과 같이 속도에 큰 영향을 줄 수 있는 조건들이 달라지기 때문에 정확한 알고리즘 복잡도 분석이 매우 어렵다. 이와 같은 어려움을 고려하여, 본 절에서는 다음과 같이 알고리즘 수행 연산 및 메모리 사용에 대한 몇 가지 기본 가정을 두고 알고리즘 수행 속도에 큰 영향을 미칠 수 있는 중요한 요소들을 살펴보도록 하겠다.

1) 적응적 조기종료 조건에 의한 연산의 종료 시점은 입력 영상에 상관없이 모두 유사하고, 각 해상도 단계에서 평균적으로 $\alpha_i \times N_{tr}$에 해당한다. 여기서 α_i는 i번째 해상도 단계의 조기종료 시점을 나타내는 파라미터로 0에서 1사이의 값을 가진다.
2) 그림 3의 첫 번째 단계인 '스타일 손실 기준 값 생성'은 스타일 영상의 해상도가 변환될 때 한 번씩만 수행되기 때문에, 평균 $\alpha_i \times N_{tr}$번 반복 수행되는 두 번째 '스타일 손실 전처리' 및 세 번째 '경사도 계산' 단계에 비해 소요되는 연산량이 무시할 수 있을 만큼 적다.
3) 재귀적 해상도 변경에 따른 부분 영상의 생성(그림 3의 $P_i(o)$, $P_i(p^{(t)})$)은 GPU에서 한 번에 처리할 파티션 크기에 변화를 유발하지만, 그 변화의 폭이 크지 않고 GPU의 고도화된 병렬처리 방식에 의해 '스타일 손실 전처리' 및 '경사도 계산'의 부분 영상 별 처리 속도에 영향을 주지 않는다.

그림 5에서 각 해상도 단계 별 경사도 계산 및 Adam 최적화기 파라미터 갱신(출력 영상 갱신 포함)에 대한 수행시간 기준 복잡도를 각각 $C_G(i)$, $C_A(i)$라 하면, 첫 번째 가정으로부터 재귀적 스타일전이 알고리즘의 전체 복잡도 C는 다음과 같이 나타낼 수 있다.

$$C = \sum_{i=0}^{N-1} \alpha_i N_{tr} \times \left(C_G(i) + C_A(i) \right) \tag{22}$$

그림 3의 '스타일 손실 전처리' 및 '경사도 계산'에 대한 복잡도는 위 세 번째 가정으로부터 특정한 상수로 둘 수 있는데, 이를 각각 C_p, C_g라 하자. 그리고 i번째 해상도 단계에서 생성된 부분 영상의 총 수를 $N_o(i)$라 하면, 두 번째 그리고 세 번째 가정으로부터 식(22)의 경사도 연산 복잡도 $C_G(i)$는 다음과 같이 표현될 수 있다.

$$C_G(i) = N_o(i) \times \left(C_p + C_g \right) \tag{23}$$

또한, 식(22)의 $C_A(i)$는 i번째 해상도 단계에서 생성된 경사도를 기반으로 각 화소 위치 경사도에 대한 곱하기와 더하기, 그리고 제곱근 연산이 수행되는 복잡도를 나타내기 때문에 각 해상도 단계의 해상도에 비례하는 복잡도를 가진다. 따라서 초기 해상도에서 이러한 Adam 최적화기 파라미터 갱신을 수행하는 복잡도를 C_a라 하면, i번째 해상도 단계에서 요구되는 복잡도는 식(24)와 같이 나타낼 수 있다.

$$C_A(i) = 4^i \times C_a \tag{24}$$

하지만, 식(24)는 해상도에 따라 소요되는 연산의 수만

4) 사용된 GPU/CPU의 사양 및 상세 구현 사항은 Ⅳ. 실험 및 결과를 참고한다.

고려한 표현으로, 이러한 표현이 수행시간 기준 복잡도에서도 동일하게 적용될 수 있으려면 캐시 미스(cache miss)나 페이지 폴트(page fault) 비율이 각 해상도 단계에서 동일하게 유지되어야 한다는 가정이 필요하다. 일반적인 상황에서는 이러한 비율이 어느 정도 유지되는 것으로 가정하는 것이 타당하지만, 메모리 사용이 극단적으로 많아지거나 특히 스와핑이 발생하는 상황에서는 이러한 가정이 적절하지 않다. 그림 5의 알고리즘에서 CPU가 요구하는 주요 메모리는 현재 해상도 단계의 오브제 영상 o_i와 스타일 영상 s_i, 그리고 부동소수점 형식으로 존재해야 하는 현재 출력 영상 $p_i^{(j)}$, 계산된 경사도 값 g, Adam 최적화기 파라미터 $\hat{g}$, $\hat{v}$, 그리고 마지막으로 계산된 갱신량 $update$ 등이 있다. 현재 해상도 단계의 해상도를 $H_i \times W_i \times 3$이라 하면, 이러한 CPU 요구 메모리는 $H_i \times W_i \times 3 \times 22$ 바이트에 해당하고, 파이썬 프로그램 환경에서의 연산 특성을 고려하면 연산의 중간 단계에서 사용되는 특정 함수(예를 들어, 제곱근 계산 함수) 혹은 특정 연산의 결과를 담을 보조 메모리가 사용되기 때문에 실제로는 이 요구 메모리의 2배 이상이 사용된다. 따라서 10K 해상도($10{,}000 \times 10{,}000$)에서 그림 5의 스타일 전이 갱신을 수행하는 데에는 대략 12GB($10{,}000 \times 10{,}000 \times 66 \approx 6.15\text{GB}$의 두 배 이상) 이상의 메모리가 사용될 수 있다. 이 경우 메모리의 전체 사용에 있어서는 큰 무리가 없을 수 있지만, 부동소수점 형식의 각 파라미터의 크기가 $10{,}000 \times 10{,}000 \times 12 \approx 1.12\text{GB}$에 해당하기 때문에 가장 큰 크기의 1GB 페이지를 사용할 수 있는 경우에도 작은 크기의 여러 페이지에 데이터가 분리되어야하고, 따라서 연산에 빈번한 TLB(Translation Look-aside Buffer) 캐시 미스가 발생하여 연산 속도 저하가 발생한다. 이러한 상황은 30K 해상도($30{,}000 \times 30{,}000$)와 같이 극단적인 크기의 초고해상도 영상의 스타일 전이에서는 더욱 심각해지는데, 이 경우에는 사용되는 메모리가 110GB를 넘어 대규모 스와핑이 발생하게 되어 수행시간 기준 복잡도가 식(24)보다 매우 커질 수 있다. 이상에서 설명한 식(22)의 $C_G(i)$, $C_A(i)$에 대한 내용을 바탕으로, 제안된 재귀적 스타일 전이 알고리즘의 복잡도는 식(25)와 같이 표현될 수 있다.

$$C = \sum_{i=0}^{N-1} \underbrace{N_o(i) \cdot \left[\alpha_i \cdot N_{tr} \times \left(C_p + C_g\right)\right]}_{\text{GPU}} + \underbrace{4^i \left[\alpha_i \times C_a\right]}_{\text{CPU}} \quad (25)$$

식(25)에서 앞부분은 GPU에서 수행되는 수행시간 복잡도를 의미하고 뒷부분은 CPU에서 수행되는 수행시간 복잡도를 의미하는데, GPU 복잡도의 $N_o(i)$와 CPU 복잡도의 4^i의 총 합은 스타일 전이를 수행할 해상도의 비와 대체적으로 비례하기 때문에 알고리즘 복잡도는 대략 스타일 전이 해상도에 비례한다고 할 수 있다. 실례로 그림 6의 4K 해상도($4{,}096 \times 2{,}160$)와 그림 7의 8K 해상도($7{,}680 \times 4{,}320$) 스타일 전이에 각기 5258.7초, 그리고 23,504.1초가 소요되었는데, 그 비($23{,}504.1/5258.7 \approx 4.47$)가 대략 해상도의 비(4)와 유사하다. 이러한 해상도에 비례하는 알고리즘 복잡도는 메모리 사용에 문제가 발생하는 30K 해상도에서도 대체로 유지되는데, 이는 충분히 큰 해상도(예를 들어, 8K 해상도 이상) 이상에서 수행되는 재귀적 스타일 전이에서는 전반적으로 α_i 값이 감소하여 메모리 접근에 수반되는 부하를 상쇄하기 때문으로 보인다.

Ⅳ. 실험 및 결과

제안된 초고해상도 스타일 전이 기법의 동작과 특징을 살펴보기 위해 그림 5의 알고리즘을 텐서플로우[24] 버전 1.13을 이용해 구현하였다. 실제 구현에서는 입력 영상의 특징에 따른 초-매개변수 세부 조절이 필요 없는 에너지 정규화를 사용하는 스타일 전이 손실[8]을 최적화 하도록 하였고, 입력 영상의 해상도에 따른 파티션 구성(그림 3의 $P_{i(\,\cdot\,)}, i = 0,1,...,N$)은 GPU에서 처리 가능한 최대 화소 수(예를 들어, 100x100 해상도의 경우 10^4)를 입력 파라미터로 설정하고, 입력 영상의 장축과 단축을 $N \times N$ 또는 $(N+1) \times N$ 으로 나누어 각 파티션이 주어진 최대 화소수보다 작아지는 최소 N 값을 구하는 방식으로 처리하였다. 이와 같은 텐서플로우 기반 구현에서 한 가지 중요한 고려사항은 텐서플로우의 그래프 변수 누적에 따른 사항으로, 그림 3의 스타일 손실 기준 값 생성, 현재 출력 영상의 파티션 기반 부분 그램 행렬 생성 및 부분 영상에 대한 스타일 전이 손실 부분 경사도 연산에서 각 부분 영상에 대한 GPU 연산을 실행할 때마다 그래프 리셋과 세션의 초기화를 수행해야 한다는 점이다. 이러한 그래프 리셋과 세션 초기화 없이 새로운 부분 영상을 입력하여 연산을 수행하는 경우

에는 이전 부분 영상의 연산 결과가 그래프에 누적되어 GPU의 메모리 부족 오류가 발생하게 된다. 이러한 구현을 사용하여 실험을 수행한 시스템의 주요 사양은 Intel i7-6700K CPU (4Core/4GHz), 64GB DDR4 RAM (2666MHz), 그리고 12GB 공유 메모리를 가지는 Nvidia Titan-V GPU이고, Ubuntu18.04 운영체제가 사용되었다.

우선, 파티션 기반 스타일 전이의 결과가 전체 영상을 사용하여 수행하는 스타일 전이와 동일한 결과를 생성하는지에 대한 확인을 위해, 1024×768 해상도의 오브제 영상과 800×600 해상도의 스타일 영상을 대상으로 초기 목적 해상도(그림 5의 r_o) 파라미터를 1024로 고정하고 GPU에서 처리 가능한 최대 화소 수 파라미터를 조절하여 1개, 4개, 그리고 6개의 파티션에서 그림 5의 스타일 전이 알고리즘이 동작할 수 있도록 비교 환경을 구성하였다. 이 세 가지 서로 다른 파티션 구성으로 스타일 전이를 수행하면서 매 출력 영상의 갱신을 위해 계산된 $1024\times768\times3$ 차원의 경사도 값(그림 5의 $update$)을 피클(pickle) 모듈을 이용해 파일로 저장하고, 이렇게 구해진 파일들을 다시 읽어 1개의 파티션에서 구해진 매 갱신 단계에서의 전체 영상 기반 경사도 값이 각각 4개, 그리고 6개의 파티션에서 구해진 부분 영상 기반 경사도 값과 동일한지 확인하였다. 확인 결과, 출력 영상 갱신을 위한 매 반복에서 동일한 경사도 값이 생성되는 것을 검증하였고, 이를 통해 부분 영상 기반 경사도 생성 알고리즘이 정상적으로 동작함을 실험적으로 확인하였다.

다음은 초고해상도 스타일 전이의 특징을 살펴보기 위해 초기 목적 해상도의 설정에 따른 스타일 전이 품질의 변화와 스타일 영상의 해상도 변화에 따른 스타일 전이 품질의 변화 특징에 대한 실험을 수행하였다. 이러한 초고해상도 영상에 대한 스타일 전이 결과가 나타내는 특징은 본 논문에서 제안하는 부분 영상 기반 최적화 방식의 특징이 아니라 매우 높은 해상도의 영상을 사용하는 경우에 나타나는 신경망 기반 스타일 전이[5]가 가진 품질 특징에 해당하는 것으로, 본 논문이 실용적 환경에서 초고해상도 영상의 스타일 전이를 가능하게 하는 최초 보고이기 때문에 기존 연구에서는 이런 특징에 대한 고찰을 찾아볼 수 없었다. 우선, 초기 목적 해상도의 설정에 따른 스타일 전이 품질은 그림 4의 결과에서 어느 정도 유추할 수 있듯이, 낮은 목적 해상도에서 시작한 스타일 전이에서는 전반적인 오브제 영상의 큰 변화를 얻을 수 있고 목적 해상도가 높아질수록 오브제 영상의 전반적인 형태가 유지된 채 세부 영역에 스타일 전이가 이루어질 것을 기대할 수 있다. 그림 7에 이와 같은 특징을 보이는 실험 결과를 도시하였는데, 실험에 사용된 오브제 영상과 스타일 영상은 8K-UHD 해상도($7{,}680\times4320$)와 Full-HD 해상도($1{,}920\times1{,}080$)를 가지고 있으며(그림 7.(a)-(b) 참고), 스타일 전이에는 평균 23,504.1초(약 6시간 32분)가 소요되었다. 앞서 설명한 바와 같이 목적 해상도 384에서 재귀적 스타일 전이 반복을 수행한 결과(그림 7.(c))는 거의 오브제 영상이 원래 어떤 형태였는지를 알아보기 어려울 정도로 큰 변화가 관찰되는 반면, 목적 해상도 1,280에서 반복을 시작한 스타일 전이 결과(그림 7.(f))에서는 비교적 오브제 영상의 형태가 많이 남아있는 것을 확인할 수 있다. 이와 같은 특징은 오브제(또는 스타일) 영상이 복잡하고 많은 텍스처 영역을 가지는 경우(예를 들어, 그림 7의 오브제 영상)와 큰 구조체 영역을 가지는 경우(예를 들어, 아웃 포커스로 촬영된 인물 사진 등, 그림 8.(a) 참고)에서 서로 다른 큰 변화의 정도를 보인다. 즉, 예를 들어 1,024 해상도를 초기 목적 해상도로 설정한 초고해상도 스타일 전이 결과는 사용된 오브제 영상과 스타일 영상의 특징에 따라 그 결과의 전반적인 형태 변형 정도에 크게 차이날 수 있다는 것인데, 재귀적 반복 없이 초기 목적 해상도에서 스타일 전이를 수행한 결과만으로 그 정도의 차이를 확인할 수 있기 때문에 다양한 초기 목적 해상도에서 일차적인 스타일 전이실험을 선행함으로써 원하는 정도의 초고해상도 스타일 전이 수행 결과의 기준을 미리 결정할 수 있다.

두 번째로, 스타일 영상의 입력 해상도 변화에 따른 초고해상도 스타일 전이의 품질 변화 특징을 알아보기 위해 그림 5에서 재귀적 반복을 수행하는 입력 영상 해상도 (o_{size},s_{size})를 결정하는 방법에 대해 먼저 살펴보자. 입력 영상의 해상도는 처음엔 목적 해상도 크기로 설정되었다가, 매 재귀적 반복에서 그 크기가 점차 가로, 세로 2배씩 확대된 크기로 변환된다. 하지만, 설정된 해상도가 입력 해상도보다 크거나 같게 되면, 입력 영상을 더 크게 확대

그림 7. 초기 목적 해상도 차이에 의한 초고해상도 스타일 전이 품질 비교 실험 결과
Fig. 7. Comparison of super-high resolution image style transfer results according to the initial target resolution

하지는 않고 입력된 크기를 그대로 사용하도록 하고 있다.

따라서 입력된 스타일 영상의 해상도가 매우 작은 경우(예를 들어, 512 해상도)와 매우 큰 경우(예를 들어, 5K 해상도)를 가정해보면, 재귀적 반복의 수행 과정에서 높은 해상도로 확대된 오브제 영상에 대응되는 스타일 영상의 크기에 큰 차이가 발생할 수 있다. 예를 들어 그림 8.(a)와 같이 $6{,}720\times4{,}480$ 해상도를 가지는 오브제 영상을 512 해상도의 스타일 영상을 통해 재귀적 스타일 전이를 수행하는 경우($r_0=512$로 설정), 초기 목적 해상도로 크기가 바뀐 오브제 영상에 512 입력 해상도의 스타일 영상이 초기 스타일 전이 수행에 사용된 후 2배씩 오브제 영상이 확대되는 모든 재귀적 최적화 과정에서도 스타일 영상은 계속 512 해상도로 남아있게 된다. 이러한 스타일 영상의 해상도 차이로 인한 스타일 전이 결과의 품질 변화가 발생할 수 있는 이유는 VGG-19 네트워크의 각 레이어 특징 벡터가 고정된 수용장

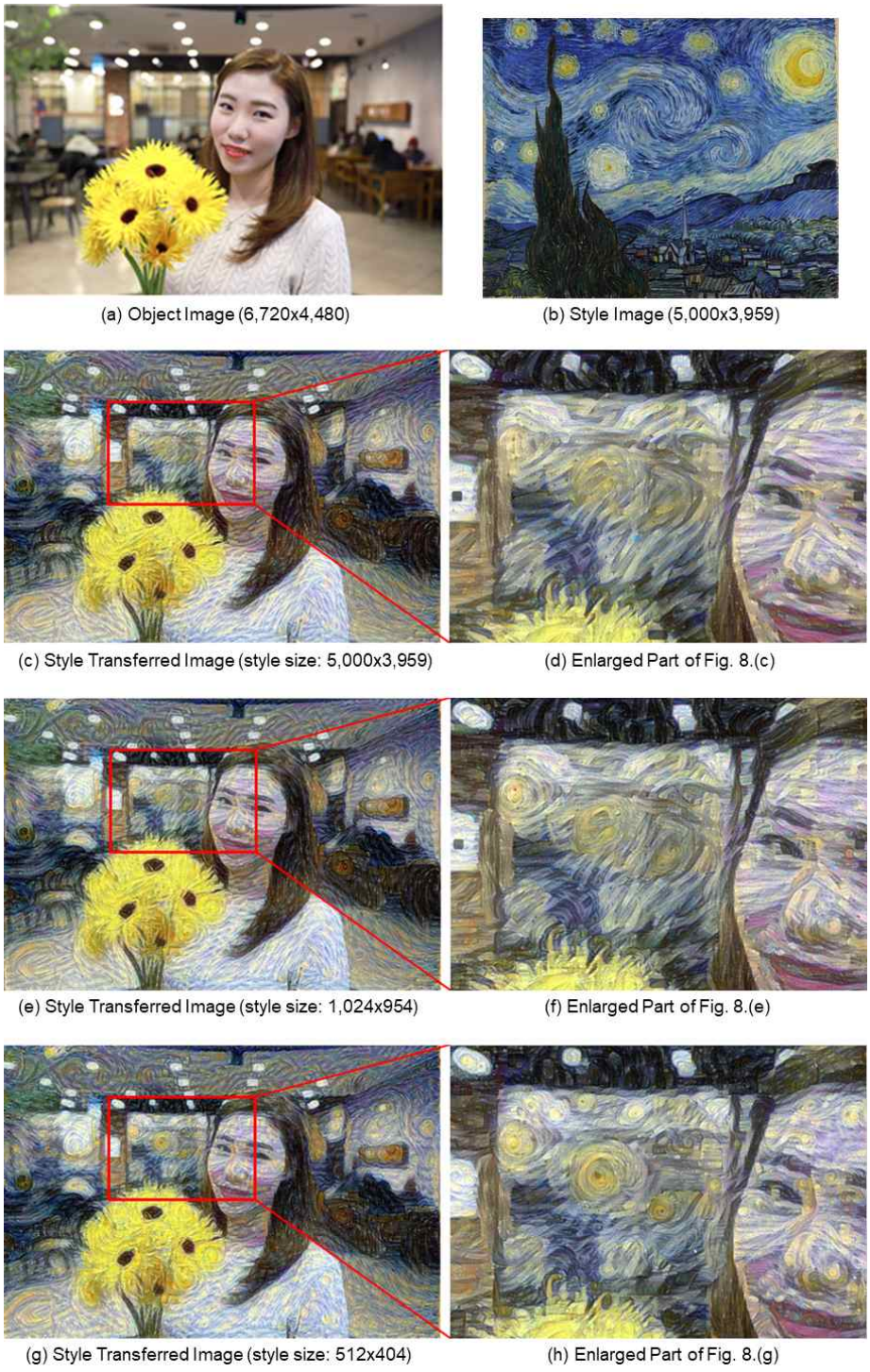

그림 8. 스타일 영상 입력 해상도 차이에 의한 초고해상도 스타일 전이 품질 비교 실험 결과
Fig. 8. Comparison of super-high resolution image style transfer results according to the resolution of style image

크기를 가지기 때문인데, 동일한 스타일 영상의 서로 다른 해상도 입력에 대한 신경망 출력 특징 벡터는 고정된 수용장 크기로 인해 그 공간적 분포에 차이가 발생하고, 이러한 공간적 분포의 차이가 스타일 손실 정의에 사용되는 그램행렬의 원소에 변화를 유발하기 때문이다.

따라서 동일한 스타일 영상이 서로 다른 해상도로 입력되는 경우의 스타일 전이는 초기 목적 해상도에서 낮은 입력 해상도 크기까지 확대되는 단계에서는 동일한 스타일 전이가 수행되지만, 그 이후에는 낮은 해상도를 입력한 경우와 더 높은 해상도를 입력한 경우의 스타일 손실이 달라져 세부 스타일 전이에서 차이가 발생할 수 있다. 이러한 세부 구조의 특징이 어떻게 달라지는지를 살펴보기 위해 그림 8에 다양한 스타일 영상 입력 해상도에 따른 초고해상도 오브제 영상 스타일 전이 품질을 도시하였다. 각 스타일 전이는 모두 동일한 초기 목적 해상도(512)를 사용하였기 때문에 그림 8.(c),(e),(f)에서 볼 수 있는 바와 같이 전반적인 스타일 전이는 모두 유사한 정도를 제공하고 있다. 하지만, 특정 영역을 확대한 그림 8.(d),(f),(h)를 보면 세부 영역에 반영된 스타일의 강도가 작은 해상도의 스타일 입력 영상을 사용할수록 더 커지는 것을 관찰할 수 있다. 또한, 그림 8.(d),(f),(h)의 각 단계 별 세부 스타일 전이 차이의 정도를 비교해보면, 그림 8.(d)와 그림 8.(f)에서의 세부 스타일 차이 정도에 비해 그림 8.(f)와 그림 8.(h)에서의 차이가 더 크고 확실하게 구분되는 것을 확인할 수 있다. 이와 같은 이유는 그림 7의 초기 목적 해상도 설정에 따른 스타일 전이 정도 특징에서 살펴본 바와 같이, 낮은 해상도에서 반영한 스타일의 재귀적 반복이 전반적인 변화 양상을 결정하기 때문이다. 즉, 그림 8.(h)는 512 해상도 스타일 전이에서 사용되었던 스타일 특징(그램 행렬)이 재귀적으로 1,024 해상도 이상에서 동일하게 반영되는 반면, 그림 8.(f)는 1,024 해상도까지 확대되어 반영된 스타일 영상의 스타일 특징이 재귀적으로 2,048 해상도 이상에서 반영되기 때문에 최종 스타일 전이 세부 영역에서 관찰되는 스타일 전이 정도의 차이는 1,024 해상도부터 혹은 2,048 해상도부터 고정된 스타일이 재귀적으로 반영되는 과정에 따른 전반적인 스타일 전이 결과의 특징이라 할 수 있다. 따라서 이러한 차이는 그림 4에서 관찰되는 바와 같이, 스타일 영상의 해상도 차이가 4K 이상인 경우(예를 들어, 30,000 해상도 오브제 영상의 스타일 전이에서 4K 해상도 스타일 영상 입력과 16,000 해상도 스타일 영상 입력이 제공하는 세부 스타일 변화 정도)에서는 거의 그 차이를 관찰할 수 없다는 특징이 있다. 그림 7과 그림 8에서 관찰할 수 있었던 이러한 스타일 전이의 특징은 재귀적 스타일 전이 반복이 수행되는 구조에서 오브제 영상과 스타일 영상의 해상도 차이가 크게 주어지는 경우에만 발생할 수 있는 독특한 현상이기 때문에 기존 연구들의 낮은 해상도 결과 고찰이나 발생망 출력에 의한 다소 높은 해상도 결과 고찰에서는 전혀 찾아볼 수 없는 독특한 특징으로, 이러한 특징에 대한 소개 및 고찰이 본 논문의 또 한 가지 중요한 기여임을 강조한다.

마지막으로, 본 논문에서 제안하는 알고리즘의 효용성을 검증하기 위해, 식(11)-식(13)의 패딩 조건과 식(20)의 $G_{i,j}^{T-p^{(t)}}$를 사용하지 않는 경우(현재 출력 영상의 전체 그램 행렬을 고려하지 않고, 파티션 별 그램 행렬을 사용하여 스타일 전이를 수행하는 경우)의 재귀적 스타일 전이에 대한 실험을 수행하였다. 실험을 위한 오브제 영상과 스타일 영상은 각기 $3{,}840\times2{,}160$ 해상도의 풍경 영상과 $1{,}300\times1{,}300$ 해상도의 패턴 영상이 사용되었고, 재귀적 스타일 전이를 위한 초기 목적 해상도는 모든 경우 $r_0=1{,}024$로 설정하였다. 그림 9에 이러한 실험 결과를 도시하였는데, 우선 그림 9.(c)에는 제안된 조건들이 모두 포함되지 않았을 때의 결과를 도시한 것으로 패딩 조건이 고려되지 않는 경우에는 부분 영상으로 처리된 각 영역들 사이에 줄이 생기는 왜곡을 확인할 수 있다. 두 번째로 그림 9.(d)에는 제안된 패딩 조건은 반영했지만 출력 영상 전체의 그램 행렬은 고려하지 않았을 때의 결과를 도시했는데, 부분 영상으로 처리된 각 영역들에 스타일 전이가 독립적으로 수행되어 영역 별 질감이 독립적이고 전체적인 색감의 통일성 및 균형적 세부 스타일 반영이 제한적임을 확인할 수 있다. 반면, 제안된 알고리즘을 사용한 그림 9.(e)의 결과에서는 이러한 왜곡이나 제한적 스타일 불균형 없이 전반적으로 우수한 스타일 반영이 수행된 것을 확인할 수 있고, 이러한 결과로부터 제안 방식의 효용성을 입증할 수 있다.

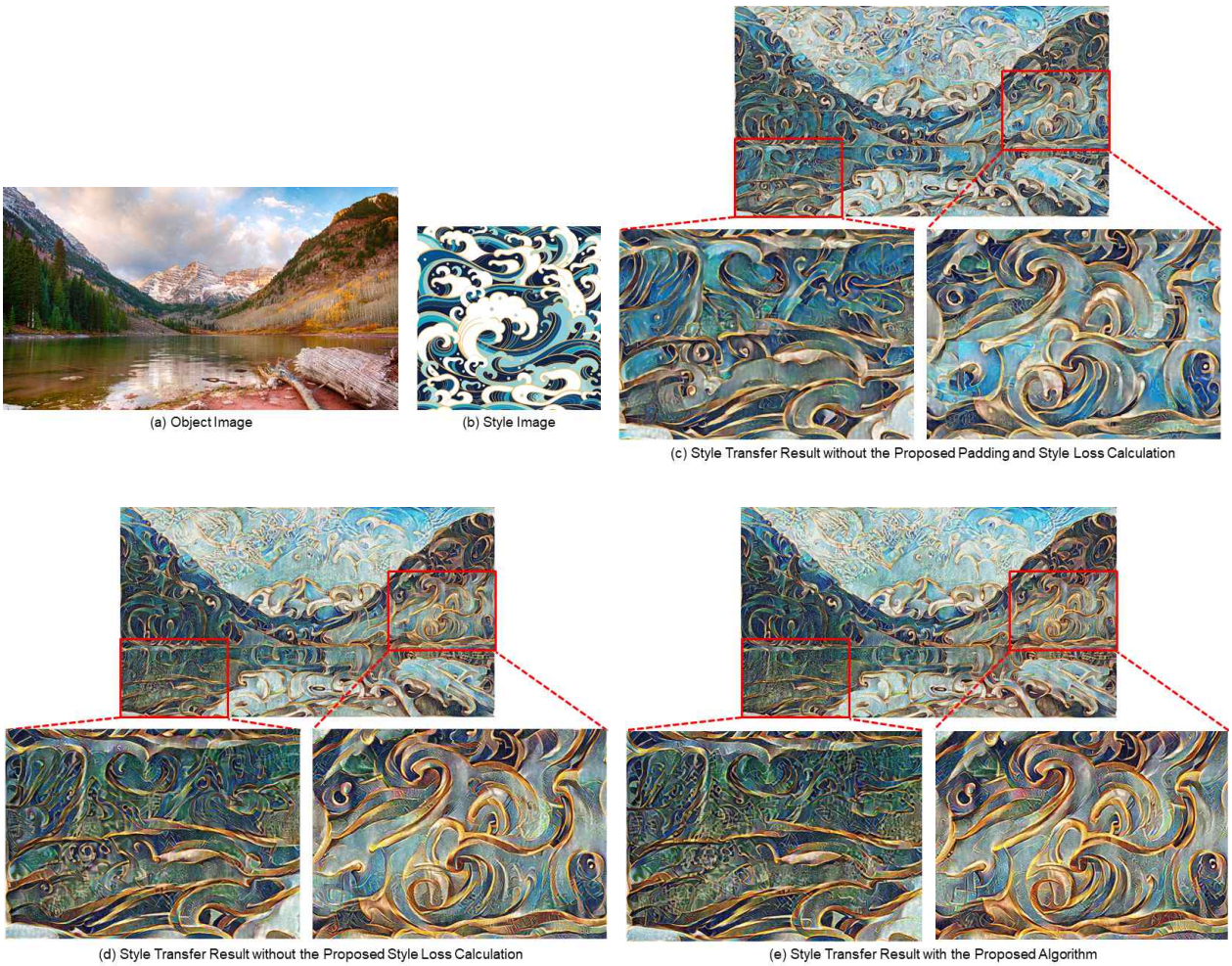

그림 9. 제안 알고리즘의 효용성 검증을 위한 초고해상도 스타일 전이 품질 비교 실험 결과
Fig. 9. Experimental results to verify the effectiveness of the proposed algorithm for super-high resolution style transfer

V. 결 론

영상의 고차원적 구조적 특징을 반영하여 높은 수준의 스타일 전이 품질을 제공할 수 있는 Gatys 등의 신경망 기반 스타일 전이 연구[5]가 소개된 이 후, 신경망을 이용한 다양한 스타일 전이 방법들이 활발하게 연구되고 있다. 특히 NVIDIA[25]나 Google[26]과 같은 회사들에 의해 최신 연구 결과들이 소프트웨어 형태로 공개됨으로써 스타일 전이에 대한 일반의 관심과 활용이 더욱 가속화되고 있다. 하지만, 이러한 신경망 기반 스타일 전이에 대한 기존 연구들을 통해 얻을 수 있는 스타일 전이 결과의 해상도는 매우 제한적이며, 사용하는 GPU의 메모리 크기에 절대적으로 제한받는 한계가 있었다.

본 논문에서는, 이러한 신경망 기반 스타일 전이의 GPU 메모리에 따른 해상도 한계를 극복하고 고품질의 스타일 전이를 수행할 수 있도록 초고해상도 영상에 대한 [5]의 스타일 전이 구현 방법을 제안하였다. 제안된 방법은 사용하는 GPU에서 처리 가능한 최대 크기로 초고해상도 영상을 파티션하고 각 부분 입력 영상을 이용한 스타일 전이 손실함수의 경사도 연산을 통해, 영상 전체를 입력하여 구한 경사도와 동일한 결과를 생성할 수 있도록 한 것이다. 이를 위해, 스타일 전이 손실함수의 각 구성 요소에 대한 경사도

연산 구조를 분석하여 부분 영상의 생성 및 패딩에 대한 조건을 구하였고 각 부분 영상의 입력으로 생성한 부분 그램 행렬을 부분 영상 기반 경사도 연산을 위한 보조 데이터로 구조화하였다. 또한, 손실함수 구성에 사용되는 VGG-19 네트워크의 고정 수용장 크기로 인한 스타일 전이 최적 상황(sweet spot)을 고려하여 초기 목적 해상도에서 입력 해상도까지 재귀적으로 스타일 전이를 수행하는 알고리즘을 설계하였다. 설계된 알고리즘을 텐서플로우 모듈을 이용해 구현하였고 구현된 소프트웨어를 통해 다양한 초고해상도 입력 영상에 대한 스타일 전이 결과를 제시하였을 뿐 아니라, 초기 목적 해상도와 스타일 영상의 입력 해상도에 따른 초고해상도 영상 스타일 전이 결과의 특징을 고찰하였다.

이러한 제안 방법을 통해 일반적인 스타일 전이 작품에서는 찾아볼 수 없는 매우 높은 해상도의 고품질 스타일 전이 출력을 생성하고, 이를 통해 매우 인상적인 국부 스타일 패턴의 형태를 제공할 수 있었다. 또한, 결과의 특징 고찰에서 보였던 바와 같이 스타일 전이의 전반적인 수준 설정과 국부적 스타일 패턴의 강도 조절이 가능하고, 이러한 기능을 이용해 낮은 초기 목적 해상도에서 생성한 고해상도 스타일 전이 결과에 새로운 스타일 영상을 입력으로 초고해상도 결과를 생성함으로써 전체적인 스타일 내 또 다른 세부 영역 별 스타일을 부가하는 것이 가능하고, 이러한 방식은 해상도 별 스타일의 종류와 대비, 그리고 강도 조절을 통해 다양한 작가의 의도를 삽입할 수 있는 새로운 도구로 활용할 수 있다. 이러한 특징에 힘입어, 본 논문에서 제안한 초고해상도 영상 스타일 전이 방법은 여러 아티스트들과의 협업을 통해 이미 수차례 전시에 사용된 바 있으며, 보다 다양한 형식의 미디어 창작활동에 많은 작가들을 대상으로 지속적인 활용이 기대된다.

참 고 문 헌 (References)

[1] P. Rosin and J. Collomosse, “Image and video-based artistic stylization”, Springer-Verlag, London, 2013.

[2] T. Strothotte and S. Schlechtweg, “Non-photorealistic computer graphics: modeling, rendering, and animation”, Elsevier Science, USA, 2002.

[3] D. Heeger and J. Bergen, “Pyramid-based texture analysis/synthesis,” Proc. of the 22nd annual conf. on computer graphics and interactive techniques, pp.229-238, 1995. doi:10.1145/218380.218446.

[4] A. Efros and T. Leung, “Texture synthesis by non-parametric sampling,” Proc. of the IEEE Int. Conf. on Computer Vision(ICCV), pp.1033-1038, 1999. doi:10.1109/ICCV.1999.790383.

[5] L. A. Gatys, A. S. Ecker, and M. Bethge, “Image style transfer using convolutional neural networks,” Proc. of the IEEE int. Conf. on Compute r Vis. and Patt. Recog.(CVPR), pp.2414 – 2423, 2016. doi:10.1109/CVPR.2016.265.

[6] Y. Li, N. Wang, J. Liu, and X. Hou, “Demystifying neural style transfer,” Proc. of the 26th Int. Joint Conf. on Artificial Intelligence(IJCAI), pp. 2230 – 2236, 2017. doi:10.5555/3172077.3172198.

[7] E. Risser, P. Wilmot, and C. Barnes, “Stable and controllable neural texture synthesis and style transfer using histogram losses,” arXiv preprint arXiv:1701.08893, 2018.

[8] Choi and Y.-G. Kim, “A normalized loss function of style transfer network for more diverse and more stable transfer results,” J. of Broadcast Engineering, vol.25, no.6, pp.980 – 993, 2020. doi:10.5909/JBE.2020.25.6.980.

[9] H. Huang, H. Wang, W. Luo, L. Ma, W. Jiang, X. Zhu, Z. Li, and W. Liu, “Real-time neural style transfer for videos,” Proc. of the IEEE Conf. on Computer Vis. and Patt. Recog.(CVPR), pp. 7044-7052, 2017. doi:10.1109/CVPR.2017.745.

[10] M. Ruder, A. Dosovitskiy, and T. Brox, “Artistic style transfer for videos and spherical images,” Int. J. of Computer Vision, vol.126, no.11, pp.1199-1219, 2018. doi:10.1007/s11263-018-1089-z.

[11] J. Johnson, A. Alahi, and L. Fei-Fei, “Perceptual losses for real-time style transfer and super-resolution,” Proc. of the European Conf. on Computer Vision(ECCV), pp.694-711, 2016. doi:10.1007/978-3-319-46475-6_43.

[12] A. Sanakoyeu, D. Kotovenko, S. Lang, and B. Ommer, “A style-aware content loss for real-time HD style transfer,” Proc. of the European Conf. on Computer Vision(ECCV), pp.698-714, 2018. doi:10.1007/978-3-030-01237-3_43.

[13] V. Dumoulin, J. Shlens, and M. Kudlur, “A learned representation for artistic style,” Proc. of the Int. Conf. on Learning Representations, arXiv preprint arXiv:1610.07629, 2016.

[14] H. Zhang and K. Dana, “Multi-style generative network for real-time transfer,” arXiv preprint arXiv:1703.06953, 2017.

[15] X. Huang and S. Belongie, “Arbitrary style transfer in real-time with adaptive instance normalization,” Proc. of the IEEE Int. Conf. on Computer Vision(ICCV), pp.1510-1519, 2017. doi:10.1109/ICCV.2017.167.

[16] Y. Li, C. Fang, J. Yang, Z. Wang, X. Lu, and M. Yang, “Universal style transfer via feature transforms,” arXiv preprint arXiv:1705.08086, 2017.

[17] L. Sheng, Z. Lin, J. Shao, and X. Wang, “Avartar-net: multi-scale zero-shot style transfer by feature decoration,” arXiv preprint arXiv:1805.03857, 2018.

[18] Y. Jing, Y. Yang, Z. Feng, J. Ye, Y. Yu, and M. Song, “Neural Style Transfer: A Review,” IEEE Trans. on Visualization and Computer Graphics, vol.26, no.11, pp.3365-3385, 2020. doi:10.1109/TVCG.2019.2921336.

[19] L. Gatys, A. Ecker, M. Bethge, A. Hertzman, and E. Shechtman, “Controlling perceptual factors in neural style transfer,” arXiv preprint arXiv:

1611.07865, 2016.
[20] K. Simonyan and A. Zissermann, “Very deep convolutional networks f or large-scale image recognition,” arXiv preprint arXiv:1409.1556v6, 2015.
[21] H. Wang, Y. Li, H. Ju, and M.-H. Yang, “Collaborative distillation for ultra-resolution universal style transfer,” arXiv preprint arXiv:2003.08 436, 2020.
[22] R. Hecht-Nielsen, “Theory of the backpropagation neural network,” in Proc. of the Int. Joint Conf. on Neural Networks(IJCNN), pp. 593-605, 1989. doi: 10.1109/IJCNN.1989.118638.
[23] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” a rXiv preprint arXiv:1412.6980v9, 2014.
[24] M. Abadi, et. al., “Tensorflow: A system for large-scale machine learni ng,” arXiv preprint arXiv:1605.08695, 2016.
[25] Fast photo style, https://github.com/NVIDIA/FastPhotoStyle (accesse d 13 Dec. 2021)
[26] Neural style transfer using tf.keras, https://www.tensorflow.org/tutoria ls/generative/style_transfer (accessed 13 Dec. 2021)

저 자 소 개

김 용 구

- 1993년 : 연세대학교 전기공학과 공학사
- 1995년 : 연세대학교 전기및컴퓨터공학과 공학석사
- 2001년 : 연세대학교 전기전자공학과 공학박사
- 2002년 ~ 2006년 : ㈜온타임텍 멀티미디어연구소 연구소장/이사
- 2009년 ~ 현재 : 서울미디어대학원대학교 뉴미디어학부 교수
- ORCID : http://orcid.org/0000-0002-8905-1984
- 주관심분야 : 초실감미디어, 컴퓨터비전, 딥-러닝, 비디오 압축, 멀티미디어 시스템