

Photorealistic Style Transfer via Wavelet Transforms

<input checked="" type="checkbox"/> Prof.	<input checked="" type="checkbox"/>
☰ 코드 깃허브 링크	
<input checked="" type="checkbox"/> Reviewed	<input checked="" type="checkbox"/>
⌚ 작성일시	@February 4, 2022 3:13 PM
✉️ Paper (해당시)	https://arxiv.org/abs/1903.09760
⌚ 최종 편집	@March 21, 2022 9:32 PM
☰ Note (추가info)	
👤 Reviewer (담당자)	김 김하연
⌚ 발표영상 (해당시)	http://gofile.me/6UrAI/O8CVKgQ6C
☰ Keywords	Generative model study Paper review / Study Style Transfer
⌚ 추가 자료	
⌚ PPT (해당시)	
📅 발표일자	@February 22, 2022

▼ Question

(상업) LL filter가 어떻게 texture를 capture할 수 있는가?

A : LL filter는 average pooling과 동일하다.

Summary

1. wavelet pooling and unpooling : preserve information of the content to the transfer network

2. progressive stylization : stylize the image faithfully

Abstract

기존 연구 한계 : Recent style transfer models have provided promising artistic results. However, given a photograph as a reference style, existing methods are limited by spatial distortions or unrealistic artifacts, which should not happen in real photographs.

연구 제안 : We introduce a theoretically sound correction to the network architecture that remarkably enhances photorealism and faithfully transfers the style. The key ingredient of our method is wavelet transforms that naturally fits in deep networks. We propose a **wavelet corrected transfer based on whitening and coloring transforms (WCT2)** that allows features to preserve their structural information and statistical properties of VGG feature space during stylization.

연구 의의 : This is the first and the only **end-to-end model** that can stylize a 1024×1024 resolution image in 4.7 seconds, giving a pleasing and photorealistic quality **without any post-processing**. Last but not least, our model provides a stable video stylization **without temporal constraints**.

wavelet transforms based on WCT(whitening and coloring) : stylization 동안 구조적 정보와 VGG feature space의 통계적 특성을 유지

Introduction

- 기존 연구

- WCT : severe distortions due to their strong abstraction ability(artistic style transfer)
- DPST(Luan) : regularizer term, optimization problem = heavy computational cost(photorealistic style transfer)
- PhotoWCT(Li) : upsampling 대신 unsampling 사용 ⇒ but, 여전히 maxpooling으로부터 발생하는 information loss 해결 X, 사후처리(post processing)에서 artifact 해결
- 연구 제안
 - **wavelet pooling**(minimal information loss)을 이용하여 기존의 max-pooling과 unpooling을 대체
 - motivation : 정확한 신호복원을 위해서 반드시 inverse operation이 필요
 - **progressive stylization** : multi-level strategy 대신 사용, single pass를 통하여 progressively transform features! ⇒ simple and efficient(single decoder), 기존 multi-level strategy에서는 반복적으로 encoding과 decoding과정에서 artifact가 amplify되는 문제 발생하였는데, 이를 해결
- Summarize
 - spatial distortion은 reconstruction condition(inverse)을 만족하지 못한 network로부터 발생
 - 이 논문에서는 wavelet corrected transfer & progressive stylization을 employ
 - 결과 : first end-to-end photorealistic style transfer model, 고해상도 이미지 처리 가능, faster, better visual quality, fewer parameter, temporally stable stylization

Method(WCT2)

1. Wavelet corrected transfer
 - a. Haar wavelet pooling and unpooling
4가지 커널 : $\{LL^T LH^T HL^T HH^T\}$

$$L^\top = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \end{bmatrix}, \quad H^\top = \frac{1}{\sqrt{2}} \begin{bmatrix} -1 & 1 \end{bmatrix}$$

L : capture smooth surface and texture

H : extract vertical, horizontal, and diagonal edge-like information

- wavelet pooling의 특성 : 원래 신호가 연산에 mirroring(unpooling)을 함으로써, 정확히 복원가능 \Rightarrow component-wise transposed-convolution & summation
- 결과적으로, maxpooling은 정확한 inverse를 가지지 않기 때문에, 정확히 신호를 복구하지 못하는 반면, wavelet-pooling은 minimal information loss & noise amplification으로 이미지를 stylize가능

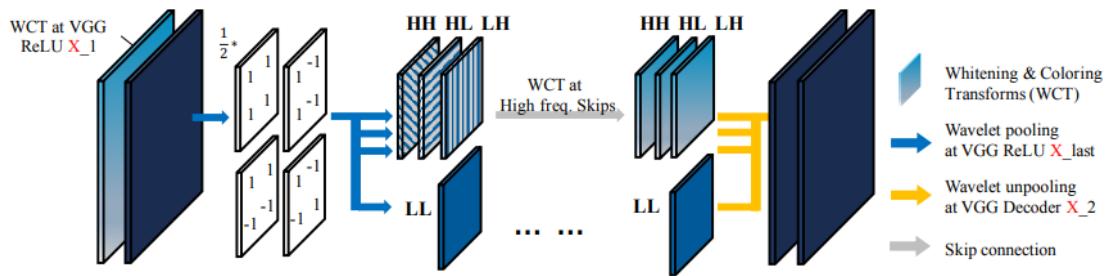
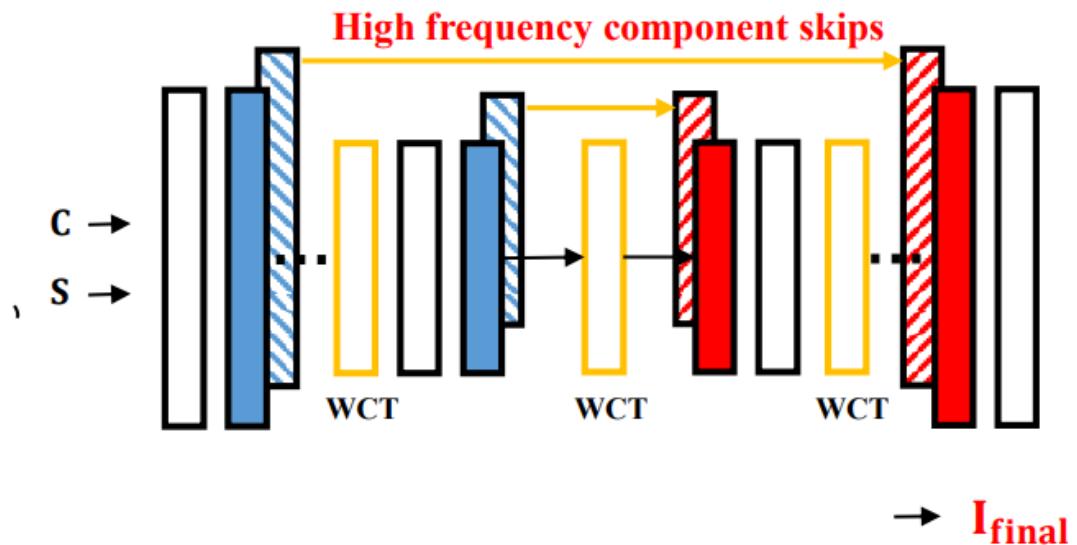


Figure 10: The proposed module using Haar wavelet pooling and unpooling.

b. Model architecture

- max-pooling and unpooling \rightarrow wavelet pooling and unpooling
- Encoder : use ImageNet pre-trained VGG-19 network from conv1_1 layers to conv4_1
- High frequency components (LH,HH,HL)는 decoder로 skip, LL은 다음 encoding layer로
- Decoder : encoder의 mirror 구조(wavelet unpooling aggregate the component)

Progressive stylization



2. Stylization

a. WCT(whitening and coloring)

Universal Style Transfer via Feature Transforms [WCT논문, 참고]

- WCT : VGG feature domain에서 content와 style의 correlation을 바로 match(기존 WCT논문에서는 multi-level stylization 사용)
- content feature를 SVD(singular value decomposition)을 통해 eigenspace of style feature로 사영시킴

b. Progressive stylization

- progressively transform features withing a single forward-pass
- 장점 : simple(single pair of encoder and decoder), suffers less from amplifying errors than the multi-level

Analysis

1. Wavelet pooling

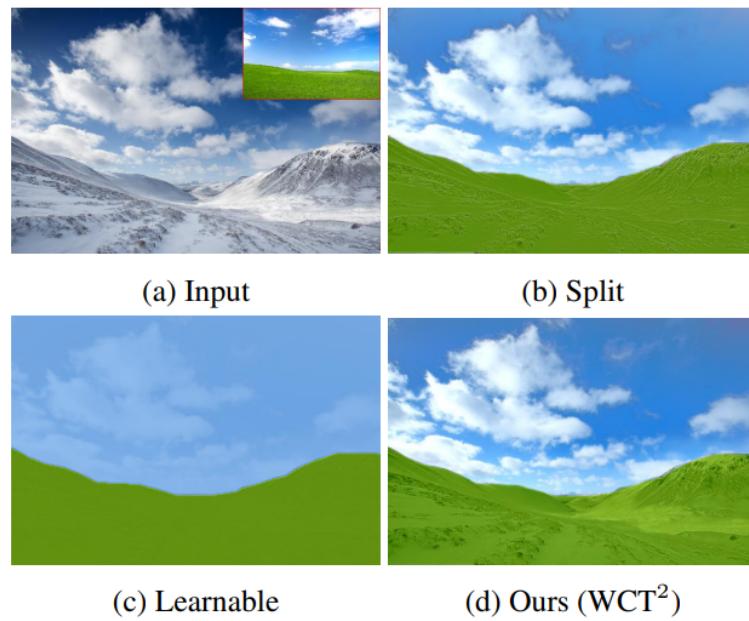
- low frequency component captures smooth surface and texture while the high frequency components detect edges. → separately control the stylization
- 즉, LL은 texture or space에 영향, {HH, LH, HL}은 edge에 영향
using only LL = average pooling 사용(average pooling은 많은 연구에서 max-pooling보다 나은 결과를 보임)



(d) Ours (LL only)

LL component만 stylize(edge성분은 unstylized)

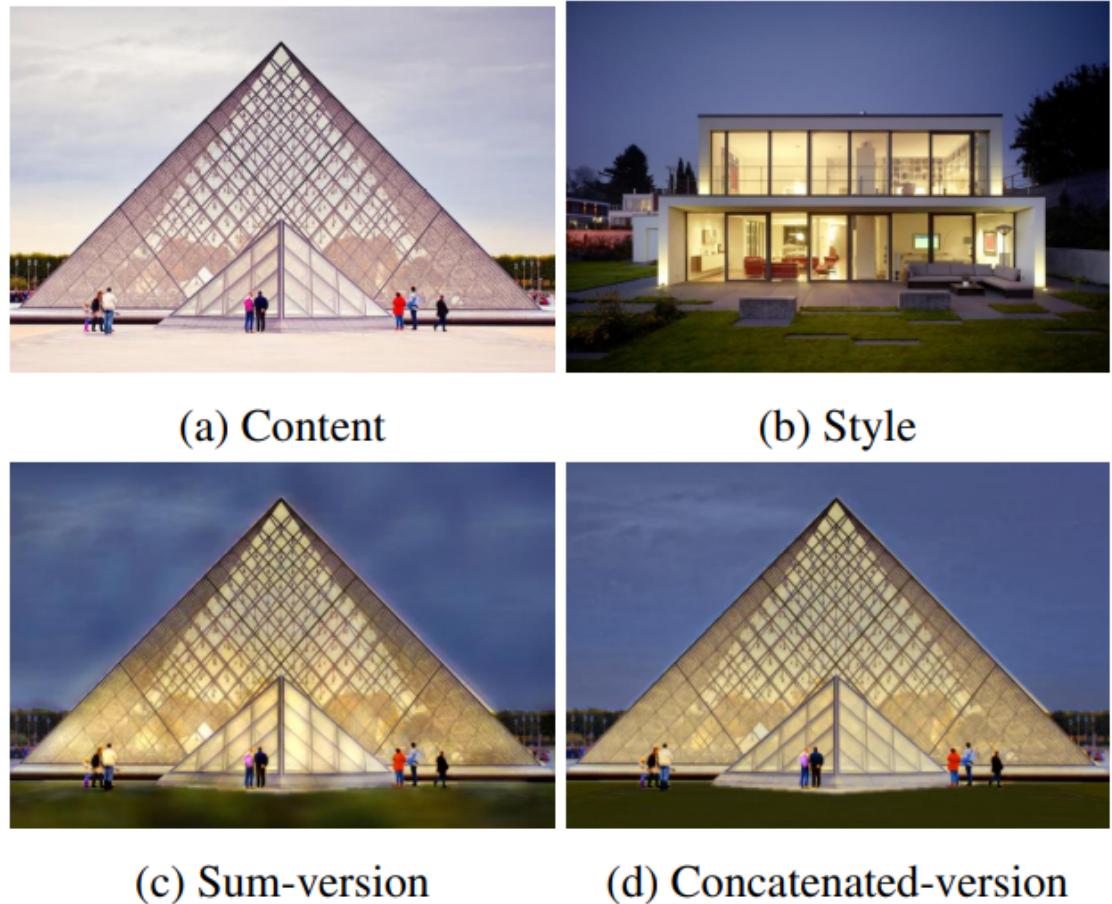
2. Ablation study : wavelet pooling을 다른 pooling 방법과 비교



(b), (c)는 lack of representation power of altered feature statistics

3. Unpooling options

- sum 대신 concatenation을 사용한 경우 : conv layer에서 parameter 수 증가
 - sum은 more stylized output, concat은 clearer image
- ▼ Fig. 5



4. Progressive VS multi-level strategy

- wavelet pooling의 정확한 reconstruction property 때문에, multi-level stylization을 사용하는 경우, contrast 증가 → more vivid result
- 따라서, single pass를 기반으로 progressive stylization을 사용(conv3_2, conv2_2, and conv1_2는 stylization effect가 further increase)

▼ Fig. 6

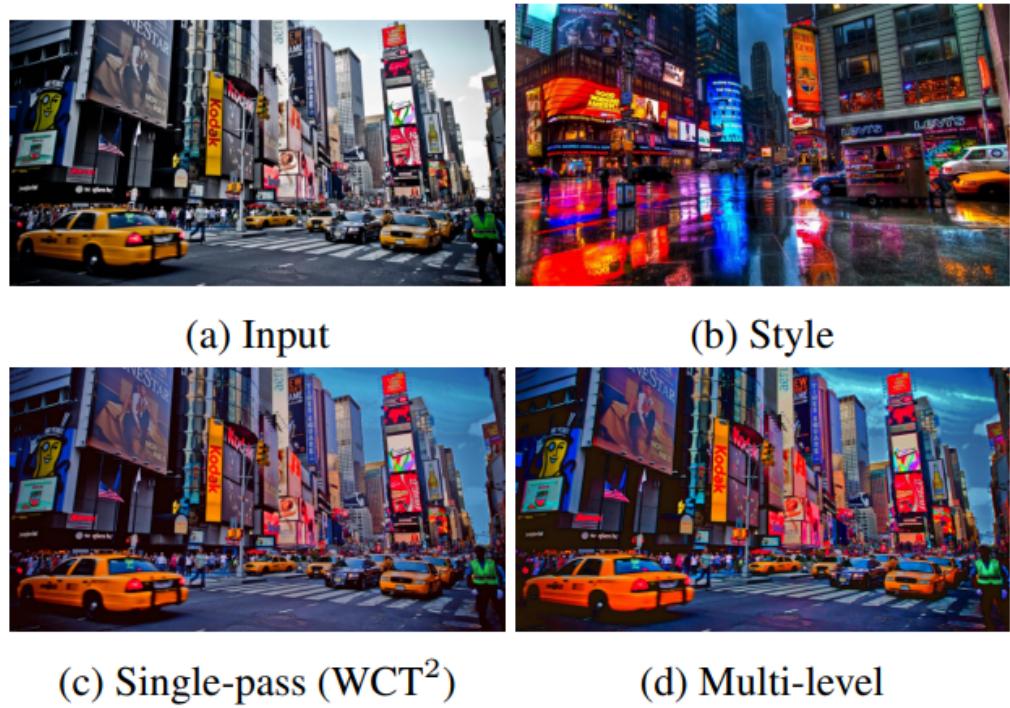
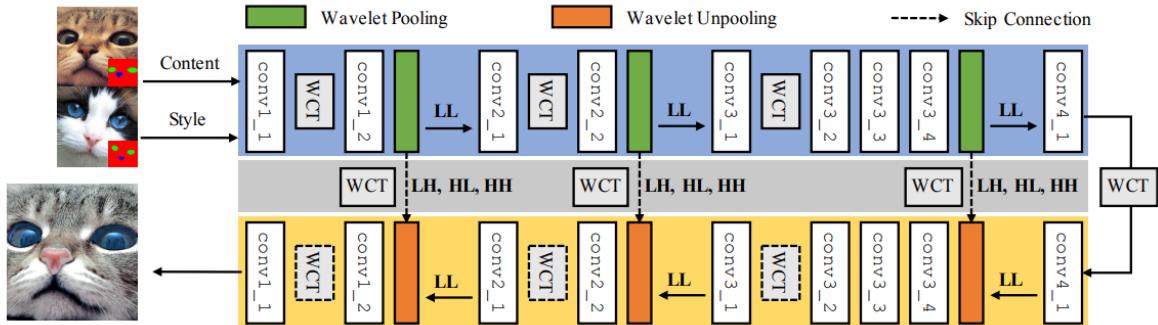


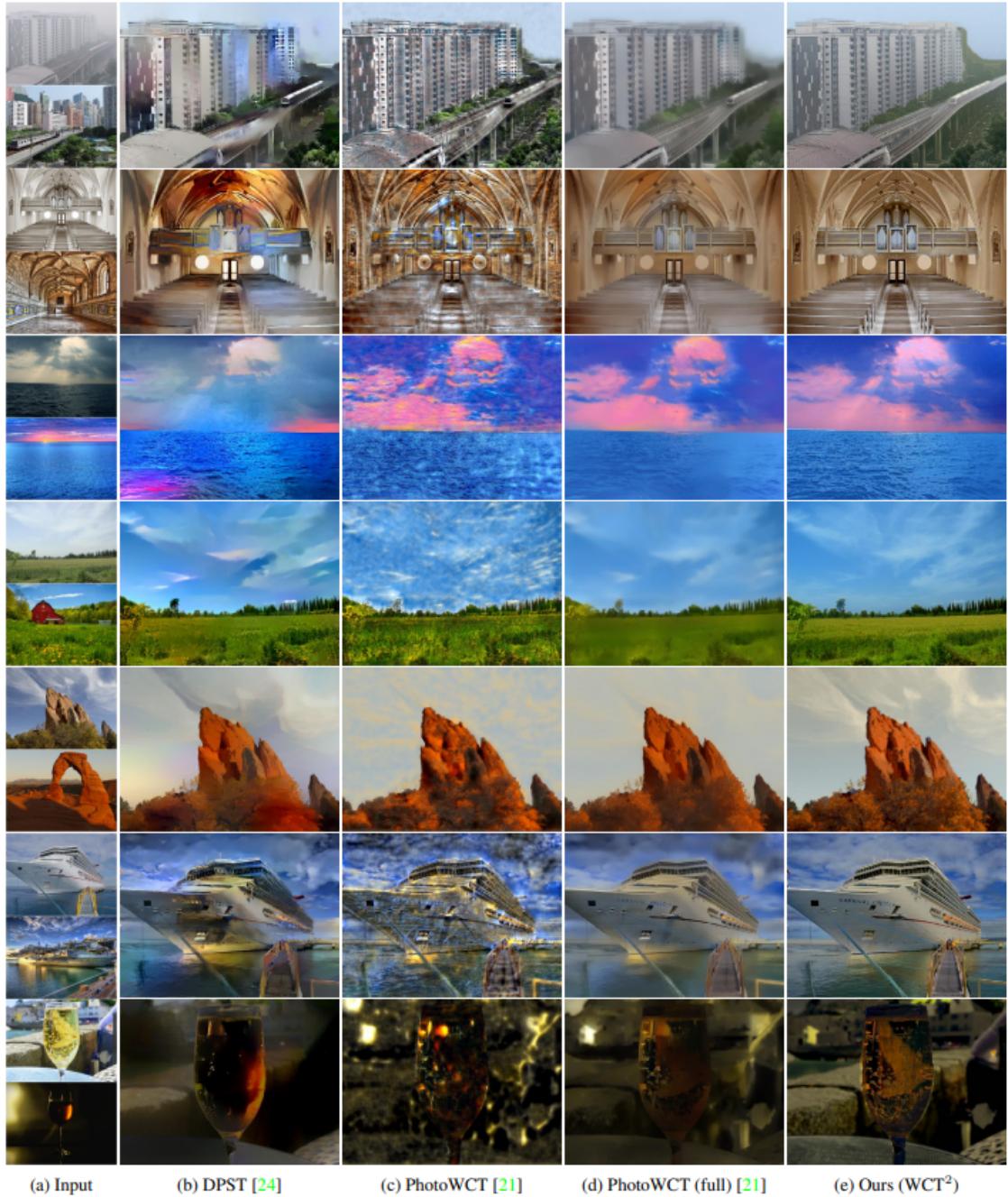
Figure 6: Stylization strength with more whitening and coloring transforms. Single-pass is our baseline (Section 4.4).



Result & Conclusion

1. Result : based on concatenated version, WCT(whitening and coloring)은 LL에만 사용

▼ Fig.7



a. implementation detail

- encoder-decoder 구조 with fixed VGG encoder weight
- decoder : Microsoft COCO dataset으로 훈련, L2-reconstruction loss + Gram matching loss 를 minimize
- NAVER Smart Machine Learning platform을 사용하여 훈련

- ADAM optimizer로 learning rate 10^{-3} 사용
- semantic map 사용(한계)

b. Result

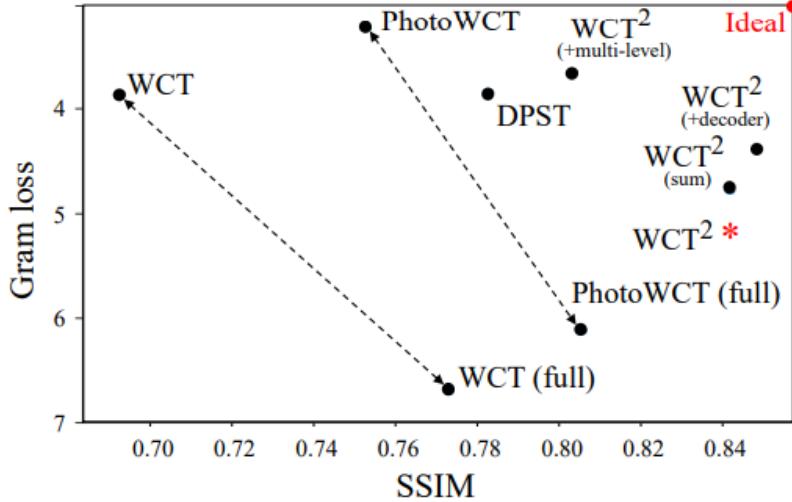


Figure 9: SSIM index (higher is better) versus Style loss (lower is better). Ideal case is the top-right corner (red dot). Dashed lines depict the gap before and after the post-processing steps, *i.e.*, smoothing. The baseline WCT² with concatenation is denoted by the red asterisk.

Image Size	PhotoWCT (full)		
	DPST	(WCT + post)	Ours
256 × 256	306.9	3.2 + 9.2	3.2
512 × 512	1020.7	3.6 + 40.2	3.8
896 × 896	2988.6	3.8 + OOM	4.4
1024 × 1024	3887.8	3.9 + OOM	4.7

Table 1: Runtime comparision of DPST, PhotoWCT (full) and ours in seconds. OOM denotes out-of-memory error.

SSIM : Structural Similarity Index

- Conclusion : first end-to-end photorealistic style trasfer method(WCT2), wavelet transform의 완벽한 recovery특성으로 구조적 정보 유지, progressive stylization사용 (better result with less noise amplification), WCT2 = scalable, lighter, faster, better

References

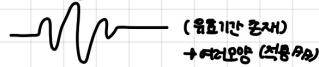
1. WCT2 presentation : [https://deview.kr/data/deview/2019/presentation/\[142_2\]신호처리+이론으로+실용적인+스타일+변환+모델+만들기+\(Better+Faster+Stronger+Transfer\).pdf](https://deview.kr/data/deview/2019/presentation/[142_2]신호처리+이론으로+실용적인+스타일+변환+모델+만들기+(Better+Faster+Stronger+Transfer).pdf)[발표 ppt], <https://tv.naver.com/v/11207601>[교수님 발표영상]
▼ Wavelet(개인 정리 - pedia 참고)

Wavelet 정의

1. Fourier 변환: 일자소진 변화는 반영X → 주, 영정화 전동

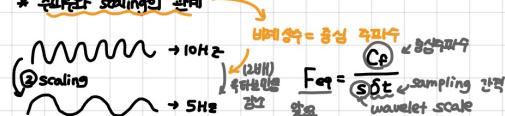
(시간 / 공간에 국한되지 않은 시장파의 흥미로 대체를 나타내기 때문)

→ Wavelet! (시간과 공간에 국한됨!)
평균이 0인 짜르게 크기가 줄어드는 전동파형



① scaling : 신호를 확장이나 축소

$$\Psi(s) = \frac{1}{\sqrt{s}} \text{ scaling} \Psi(t) \text{ or } \frac{1}{\sqrt{s}} \text{ 주파수}$$



② Shifting: 시작을 시프팅하는..

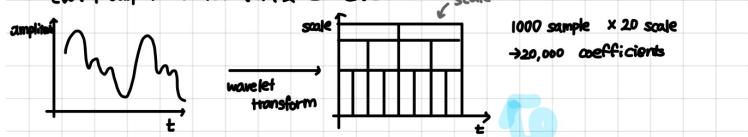
$\Psi(t-k)$ 신호에서 찾는 특징에 맞춰 원래의 wavelet 이용하여 찾는다

→ 중심이 k

2. Types of wavelet (연속/이산)

① continuous : 시간주파수 분석에 good

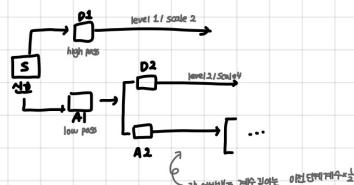
cont output = scale / 주파수와 시간의 합의인 대로



② discrete : 더 적은 계산으로/이상적으로 표현에 good

- scaling : 2^j (일정 scaling은 2)

- translation : 2^j m (m=1, 2, 3, ..)



3. Wavelet transformation 이론:

① Fourier 변환: 시간에 대한 흥미/선흐를 주파수 성분으로 분해



⇒ 신호내에 '어떤 주파수'를 분석하는가, but '언제' 그 주파수가 존재하는지 알 수 X

→ 단시간 푸리에 변환: 단시간을 짧은 시간으로 분할하여

각 시간 구간마다 어떤 주파수가 존재하는지 분석!!

(시간↓ : 어떤 '시간에 어떤 주파수'에 대한 정보)

(시간↑ : 어떤 '주파수'가 그 시간내에 존재하는지 파악하는 것)
(상대 주파수 분석 등↑)

) 즉, 시간 분석능력

주파수 분석 능력을 trade off

② wavelet transform: 시간주파수 분석 능력을 확장 (시간축은 한정) → 균형 (주파수: 길이) + 세밀(고정)

[고정파 : 시간축은 ↑ (별로 좋지)

[자유파 : 주파수 대역도 ↑ (시간은 같아)

*변환 : scale & shifting 하여 분석

① 전尺度 wavelet을 이용해서

단위 주파수 대역으로 변환 (frequency)

② 절점 scale을 가로면서 변환

(선택한 대역을 C로 구함)

비교한 대역은 만족하면 C 부여

Scale↓ :

파워를 더해가면서 분석

→ 시간축 능력이 좋다

Scale↑ :

선흐를 더해가면서 분석

→ 주파수 분석 능력이 좋다

*Haar wavelet

wavelet 7가지형

① 쪼개는법

② 전통화면에서 진폭이 0으로 수렴



$$\text{Ex)} [1, 3, 5, 7, 7, 1, 5, 3] \xrightarrow{\frac{5-1}{2}} \\ [2, 6, 4, 4, -1, -1, 3, 1]$$

$$[(4, 4, -2, 0), -1, -1, 3, 1] \\ \text{or} \text{不尽단계} \\ [(4, 0)-2, 0, -1, -1, 3, 1]$$

$$[(4, 0)-2, 0, -1, -1, 3, 1]$$

