



Style and Content, Texture Synthesis

Jun-Yan Zhu

16-726, Spring 2025

Many slides are borrowed from Alyosha Efros, Lvmín Zhang,
Maneesh Agrawala , Bill Freeman

photo © [Gatys et al.¹, 2016]

Collection Style Transfer



Photograph ©Alexei Efros



Monet



Van Gogh



Cezanne



Ukiyo-e

Style and Content Separation

A

Classification

A	B	C	D	E
<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>
A	B	C	D	E
<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>
A	B	C	D	E
B	C	A	E	D

Domain Adaptation

B

Extrapolation

A	B	C	D	E
<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>
A	B	C	D	E
<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>
A	B	C	D	E
?	?	C	D	E

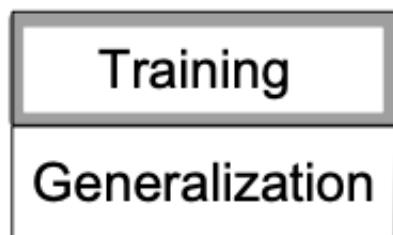
Paired Image-to-Image Translation

C

Translation

A	B	C	D	E	?	?	?
<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>			
A	B	C	D	E			
<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>			
A	B	C	D	E	?	?	?
?				?	F	G	H

Unpaired Image-to-Image Translation



Separating Style and Content
[Tenenbaum and Freeman 1996]

$$y_k^{sc} = \sum_{i=1}^I \sum_{j=1}^J w_{ijk} a_i^s b_j^c.$$

Style and Content

Adversarial loss

$$\mathbb{E}_x \log(1 - D_Y(G(x))) + \mathbb{E}_y \log D_Y(y)$$



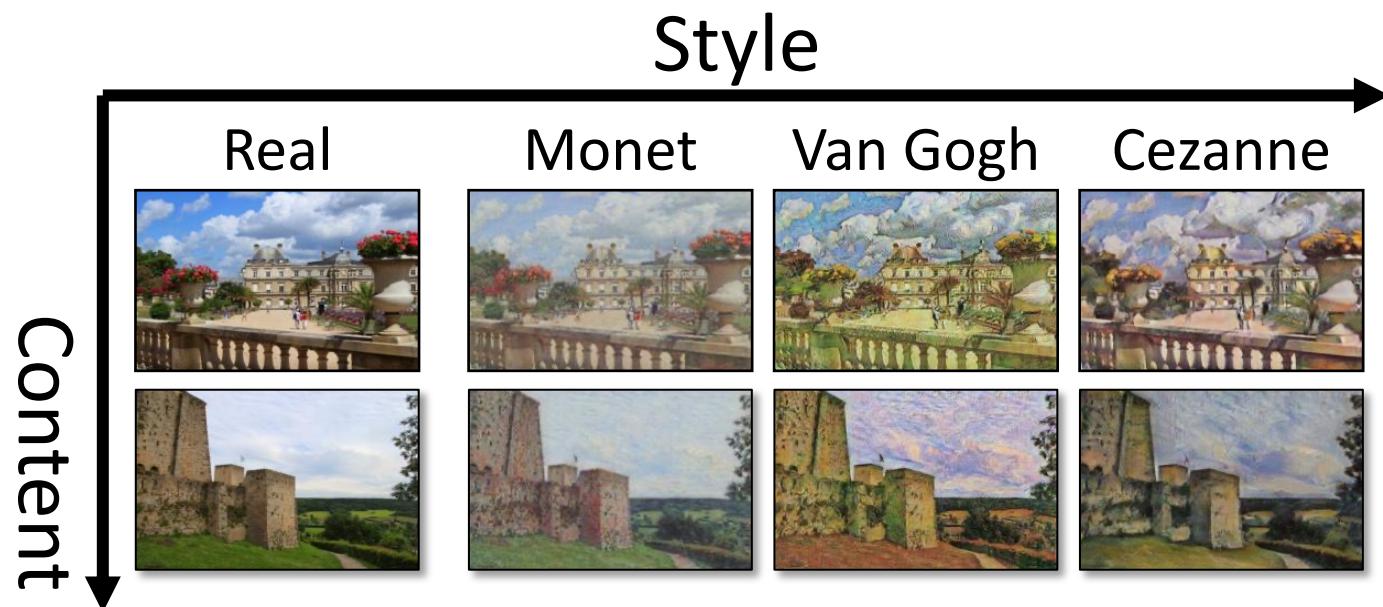
$p(x) \rightarrow p(y)$ change style

Cycle-consistency loss

$$\mathbb{E}_x \|F(G(x)) - x\|_1$$

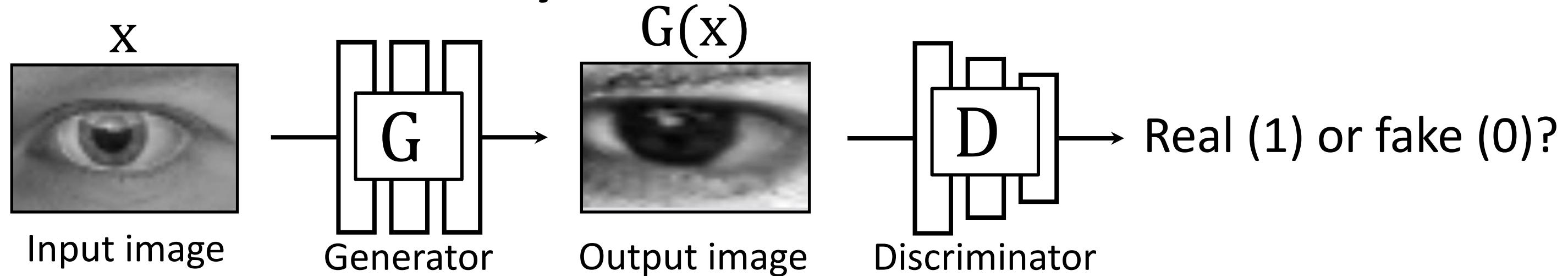


Bidirectional: preserve content



Separating Style and Content
[Tenenbaum and Freeman 1996]

Style and Content

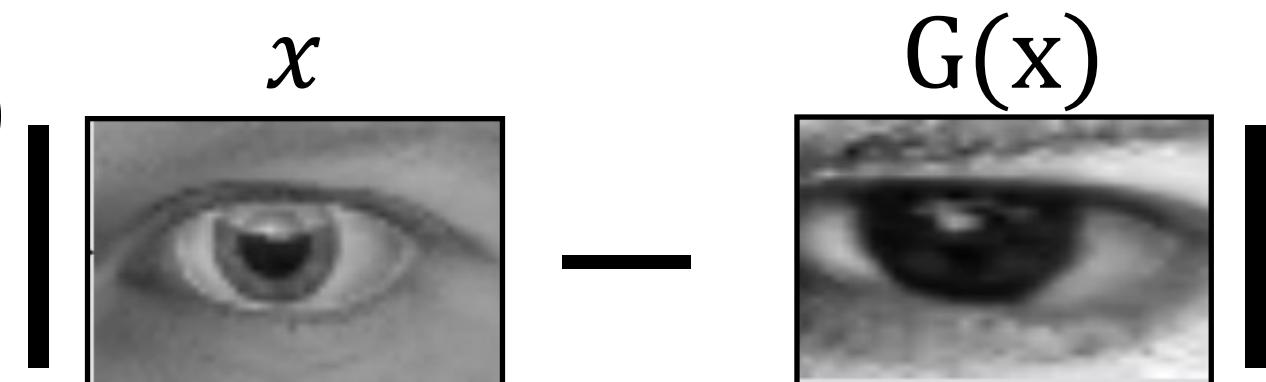


Adversarial loss (change style)

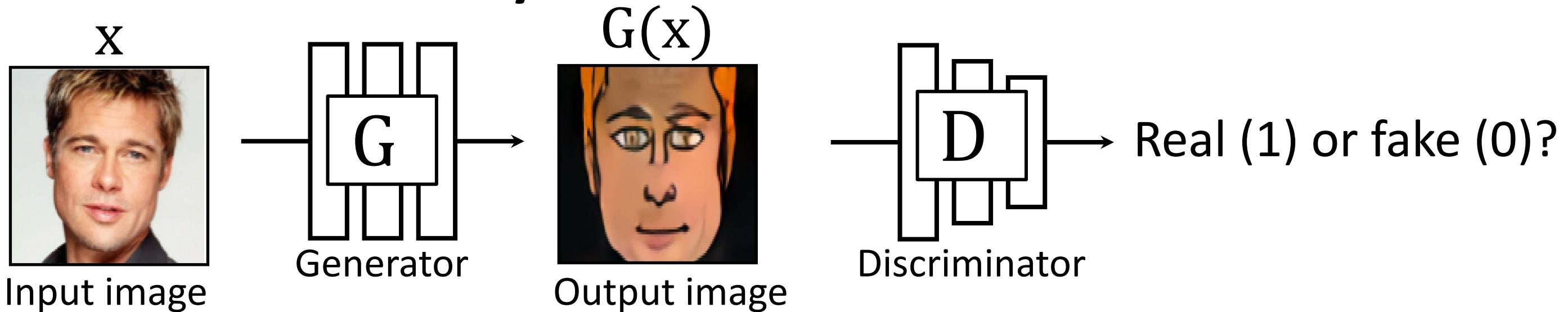
$$\mathbb{E}_x \log(1 - D_Y(G(x))) + \mathbb{E}_y \log D_Y(y)$$

L1 loss (preserve content in pixel space)

$$\mathbb{E}_x \|G(x) - x\|_1$$



Style and Content



Adversarial loss (change style)

$$\mathbb{E}_x \log(1 - D_Y(G(x))) + \mathbb{E}_y \log D_Y(y)$$

Feature loss (Preserve content in feature space)

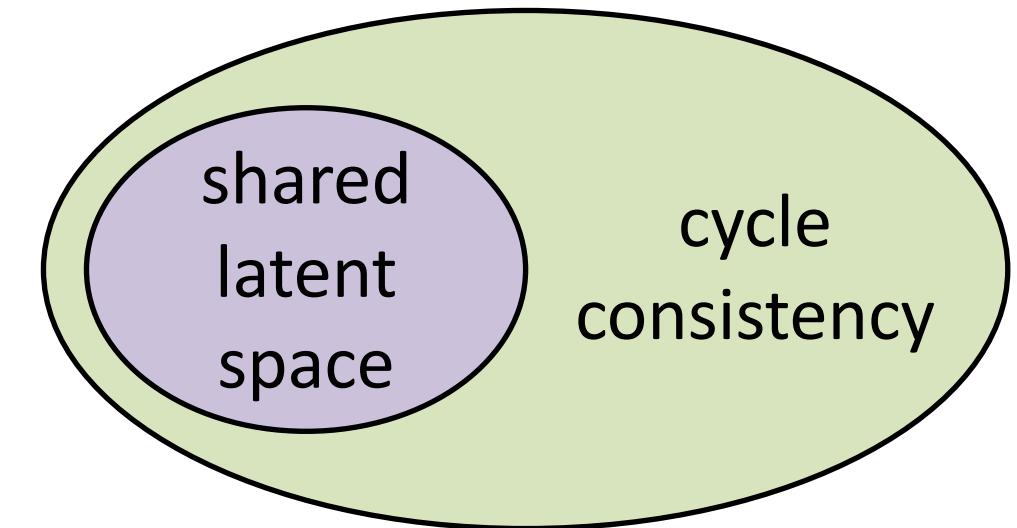
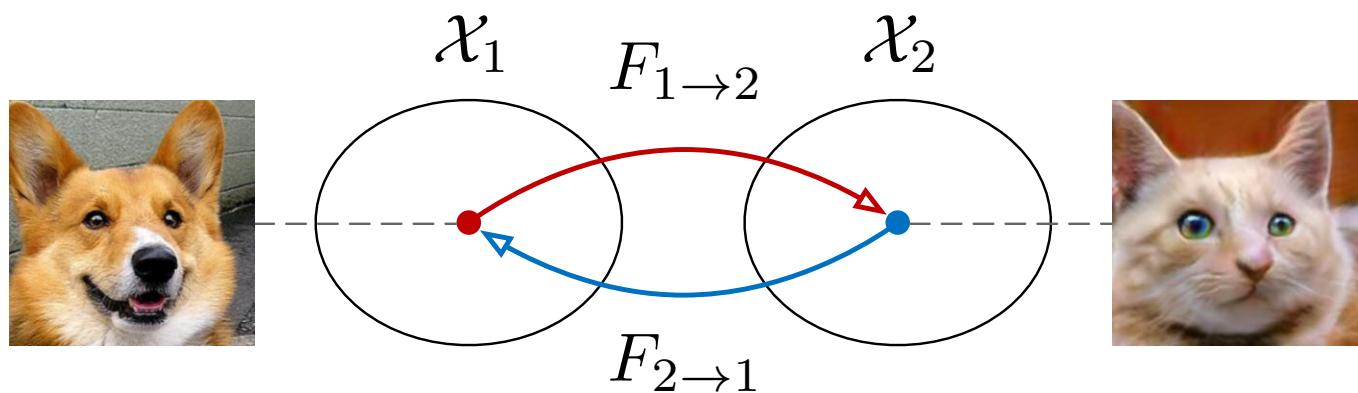
$$\mathbb{E}_x \|F(G(x)) - F(x)\|$$

$$|F(\text{Input}) - F(\text{Output})|$$

DTN [Taigman et al., 2017]

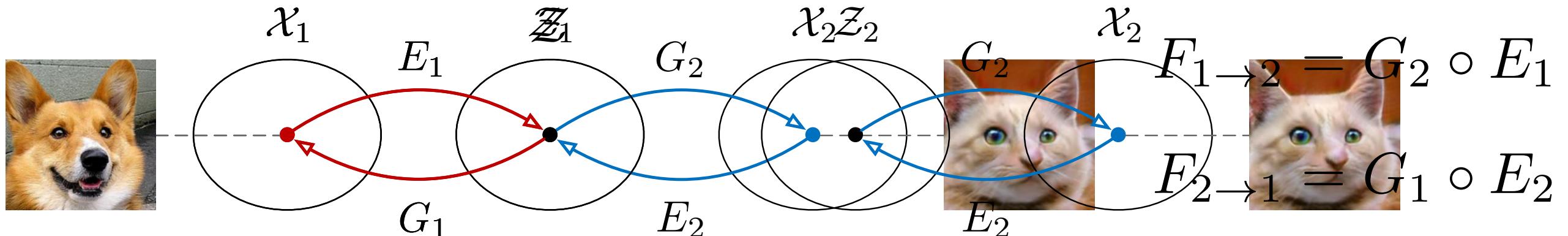
CycleGAN and UNIT

- CycleGAN (**cycle consistency**)



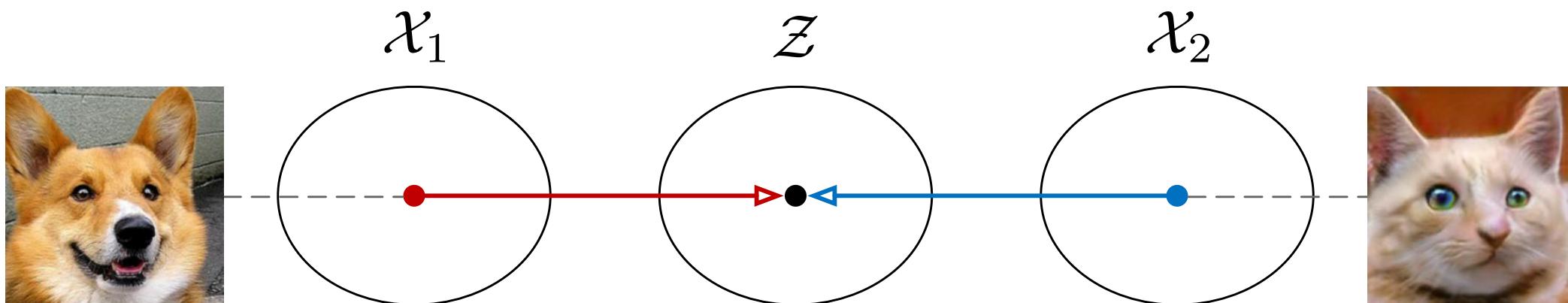
- UNIT (**shared latent space**) [Liu et al. 2017]

shared latent space \Rightarrow cycle consistency



Disentangling the Latent Space

- UNIT
 - A single **shared, domain-invariant** latent space \mathcal{Z}



Disentangling the Latent Space

- Multimodal UNIT (MUNIT)
 - A **content** space \mathcal{C} that is **shared, domain-invariant**
 - Two **style** spaces $\mathcal{S}_1, \mathcal{S}_2$ that are **unshared, domain-specific**

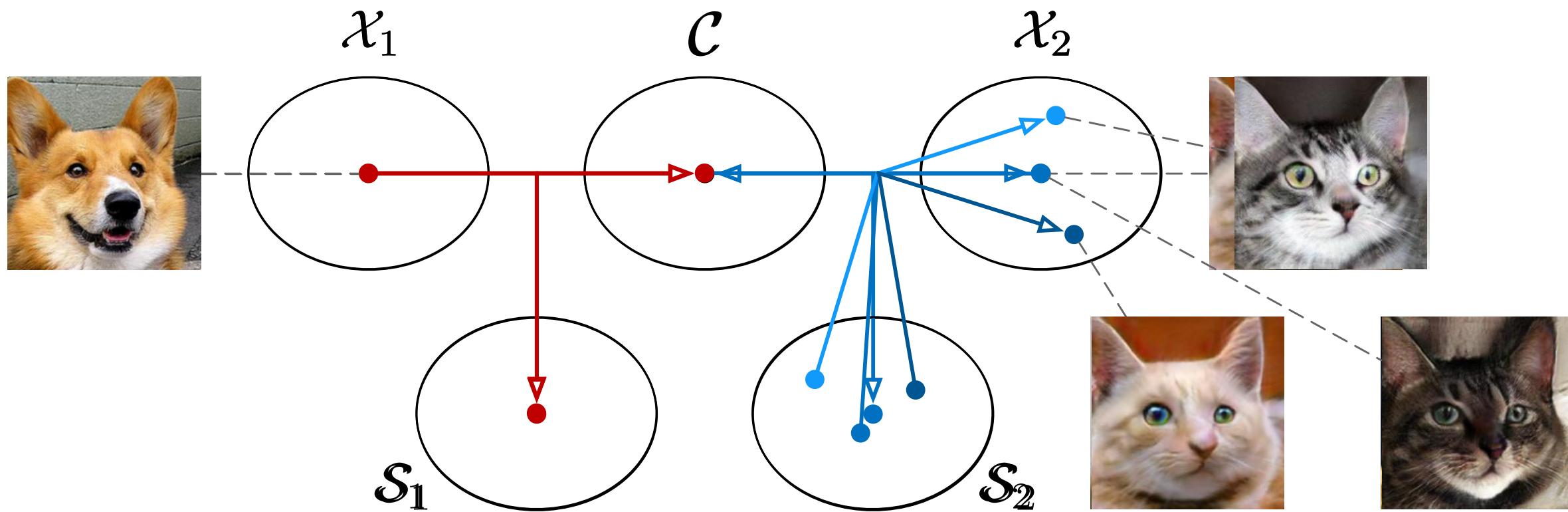
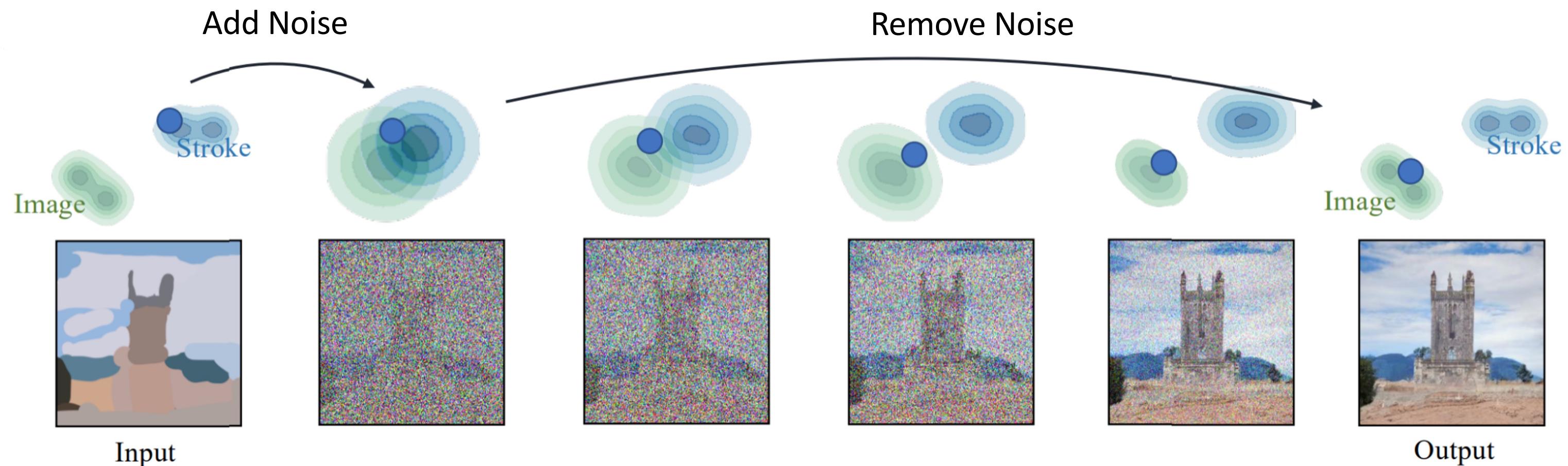


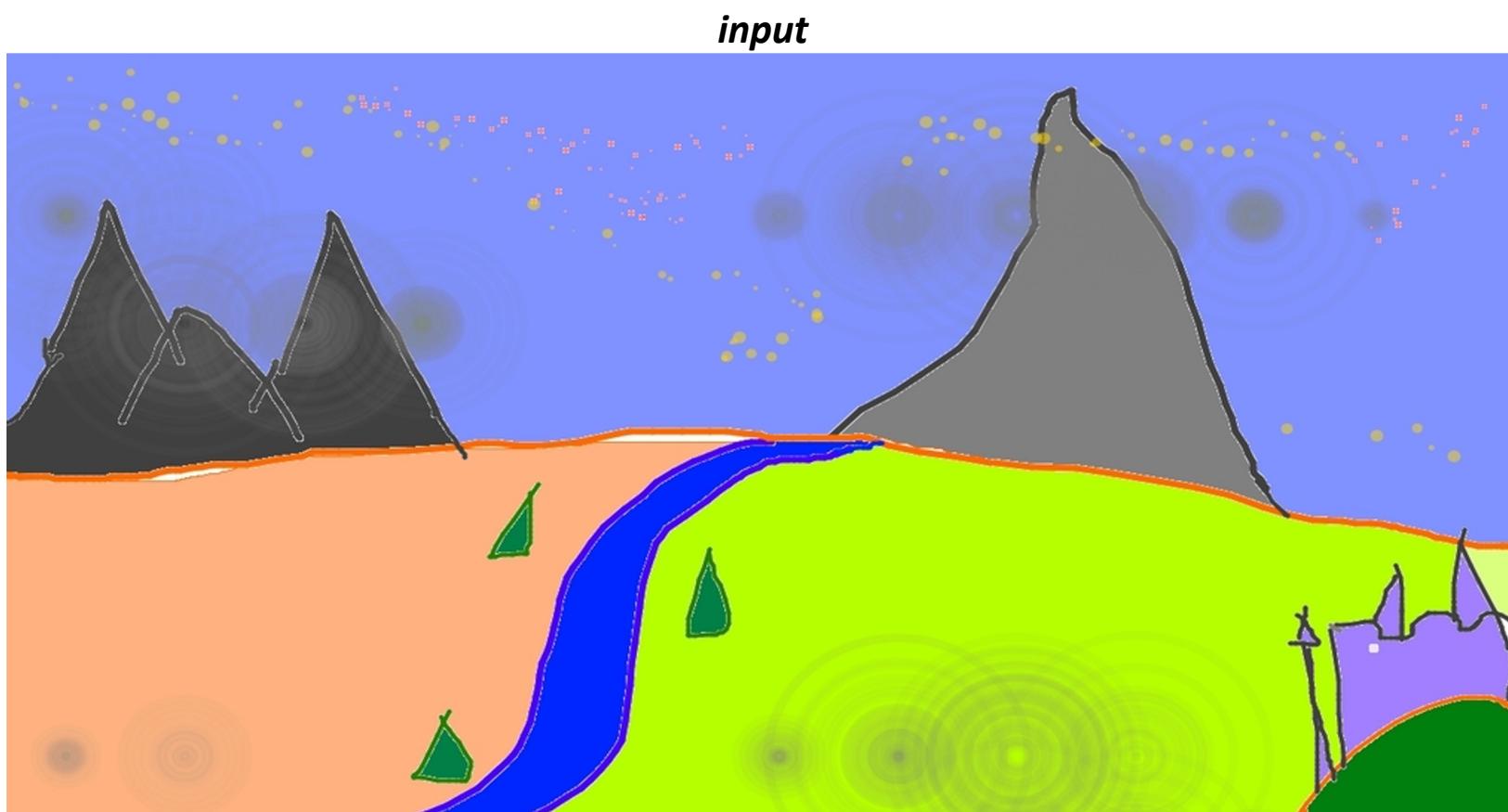
Image-to-Image Translation with Diffusion Models

Guided Image Synthesis

SDEdit (<https://arxiv.org/abs/2108.01073>) recipe: diffuse → denoise

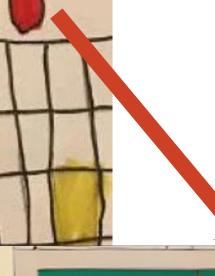
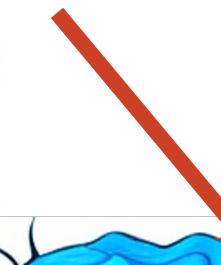


Guided Image Synthesis

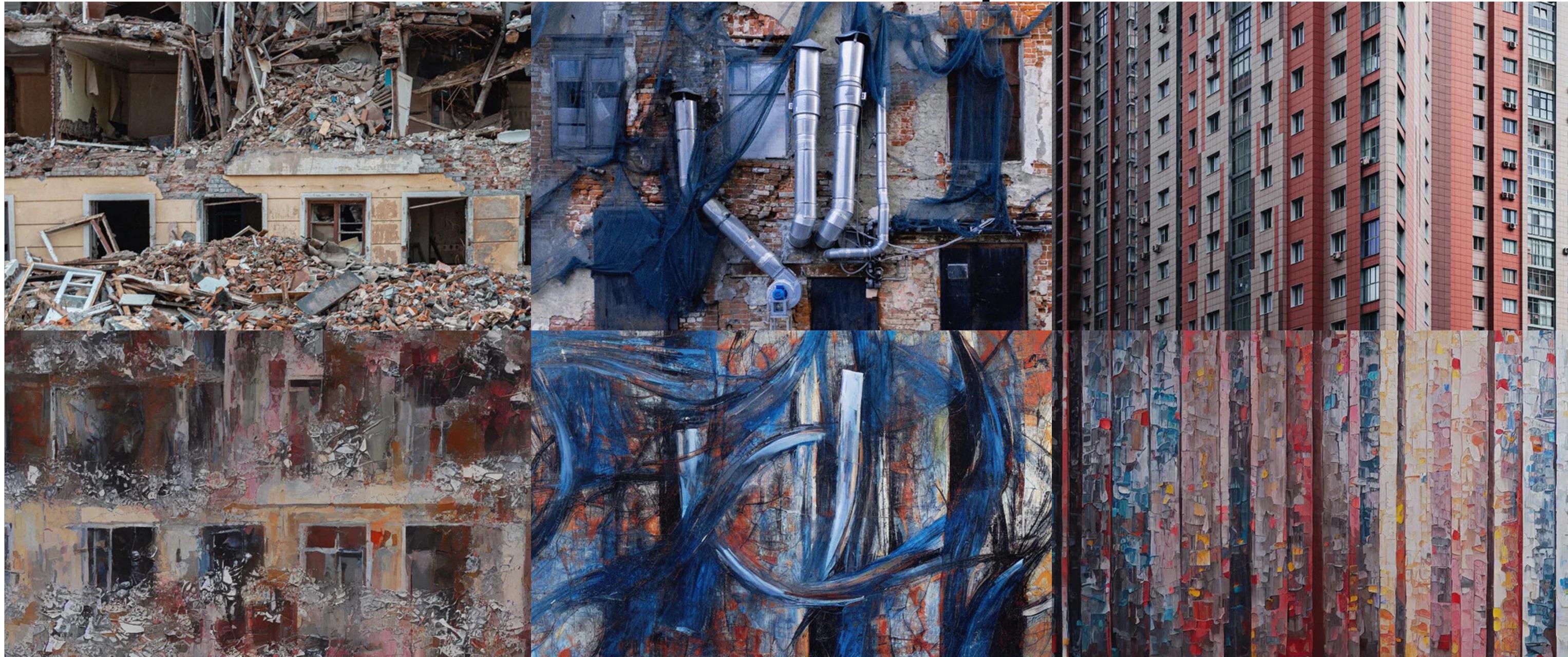




“Upgrade” your child’s artwork



abstract art from photos

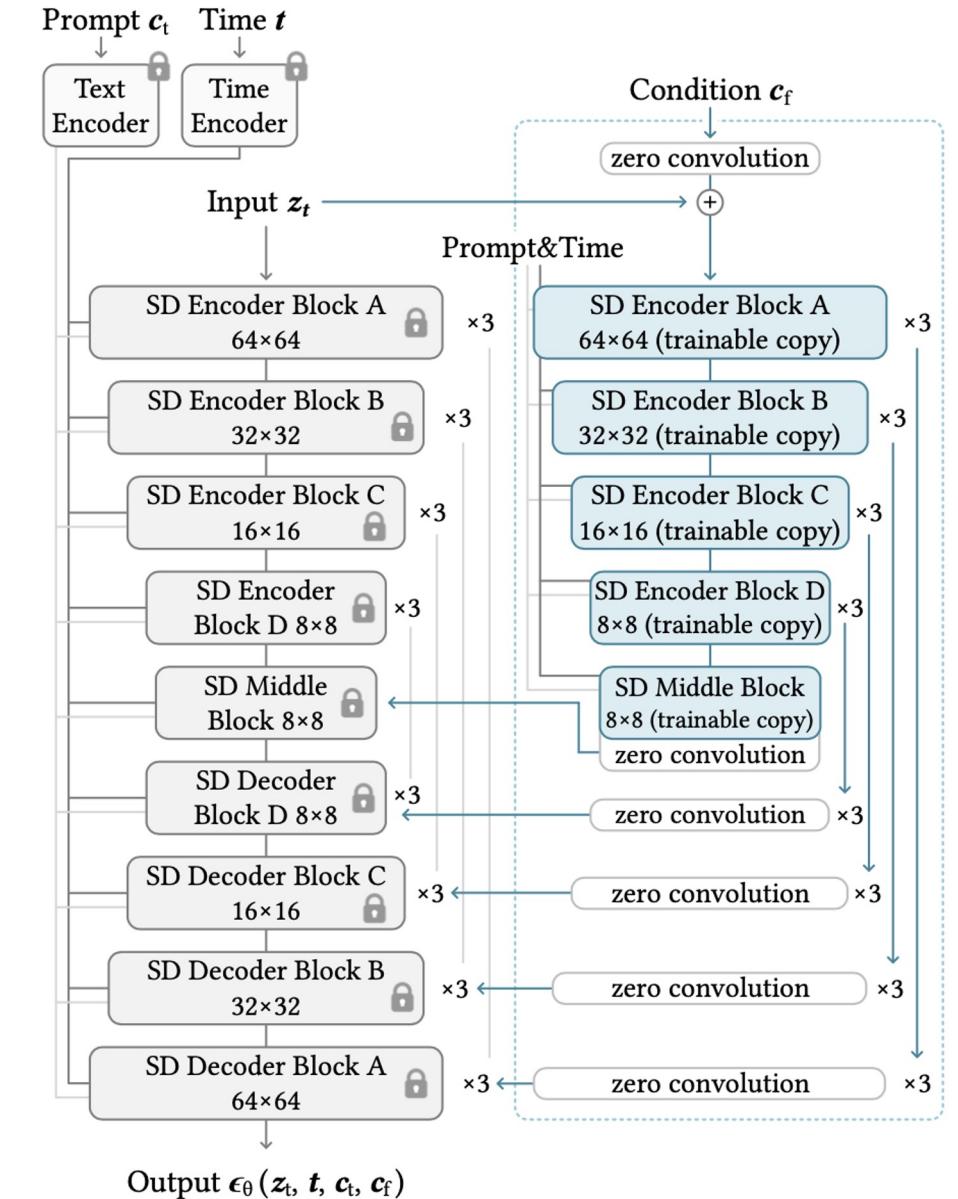
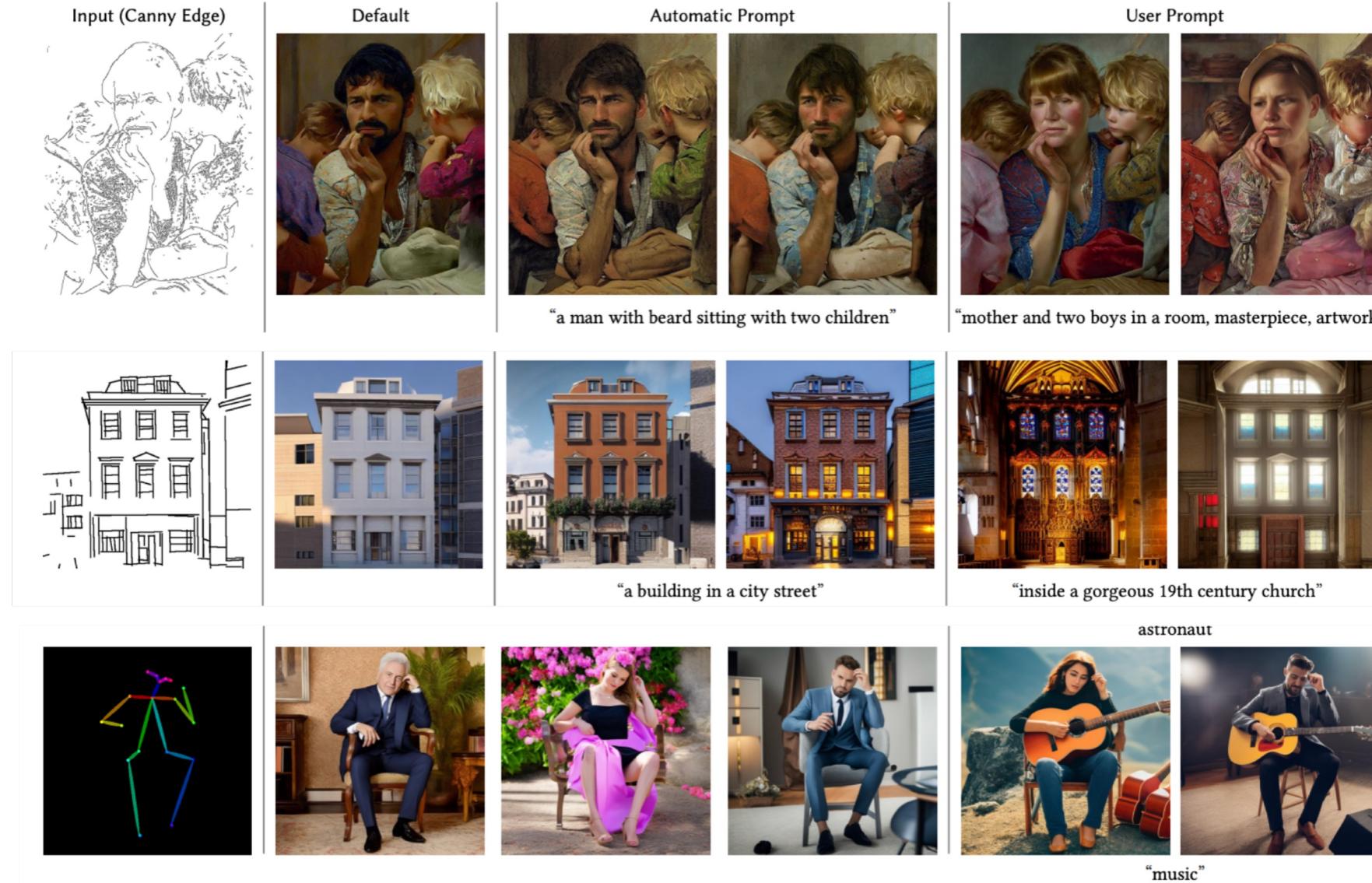


original post by [u/Pereulkov](#)

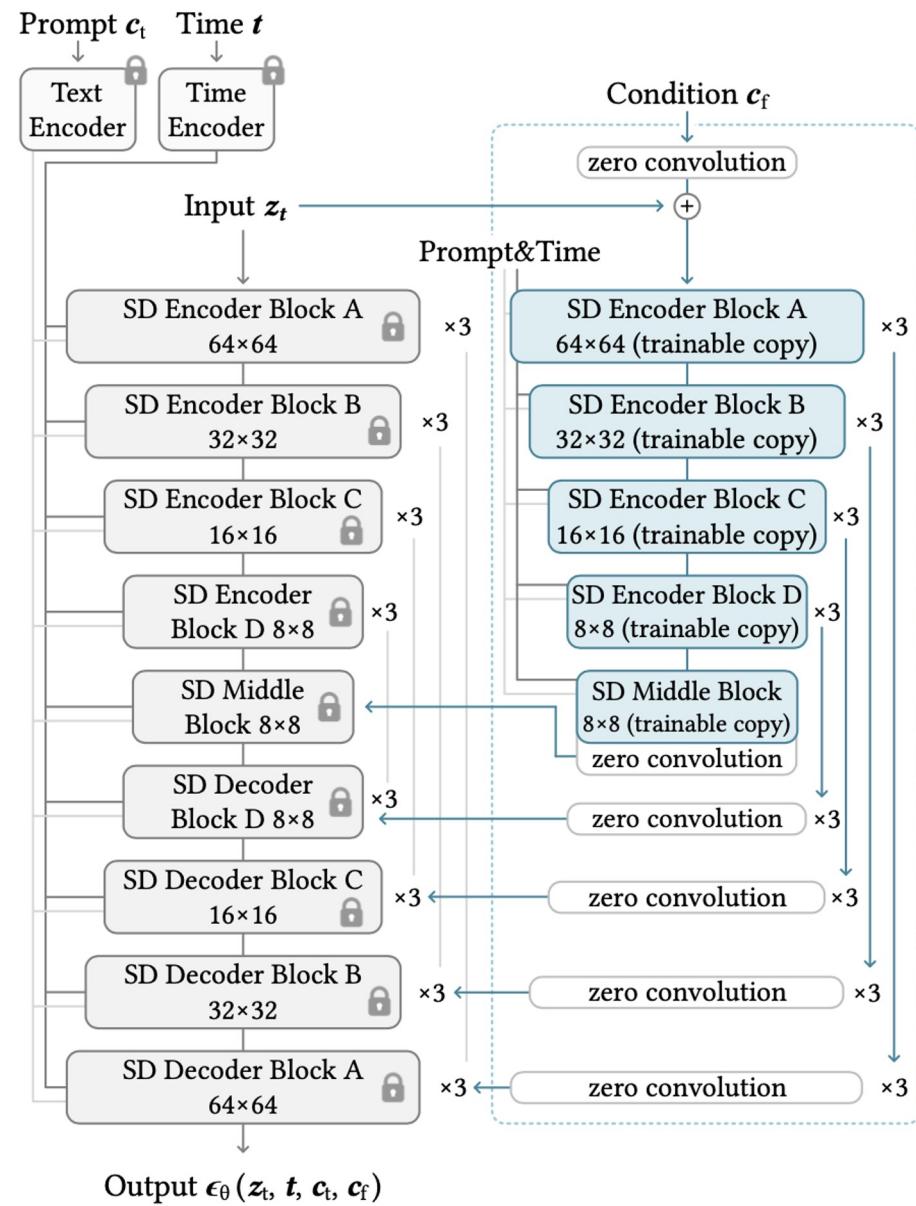
https://www.reddit.com/r/StableDiffusion/comments/xhyad/i_made_abstract_art_from_my_photos/

ControlNet: Adding Conditional Control to Text-to-Image Diffusion Models

ControlNet

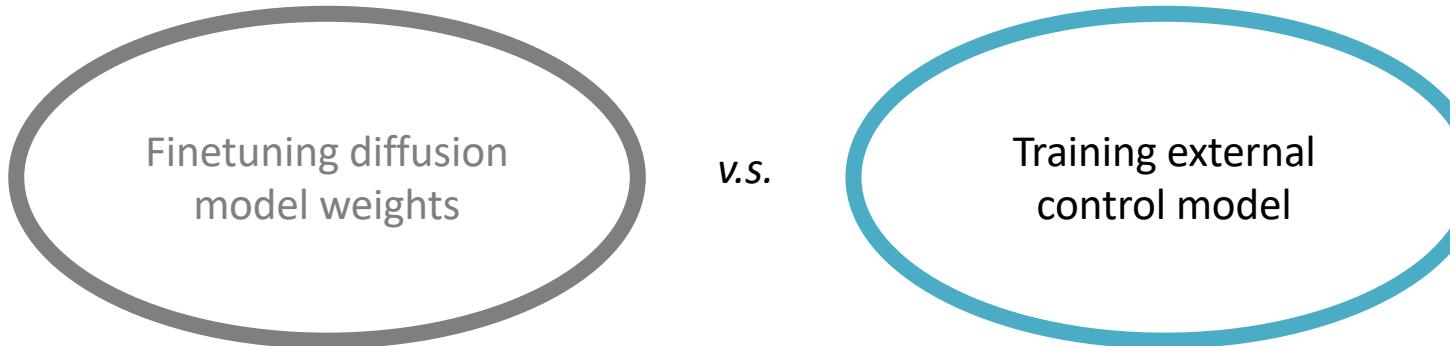


Architecture of ControlNet



- Using external model to process control signals.
- Re-using pretrained weights as the backbone of control model.
- Connecting with zero-initialized layers to reduce initial noise.

Using external model to process control signals



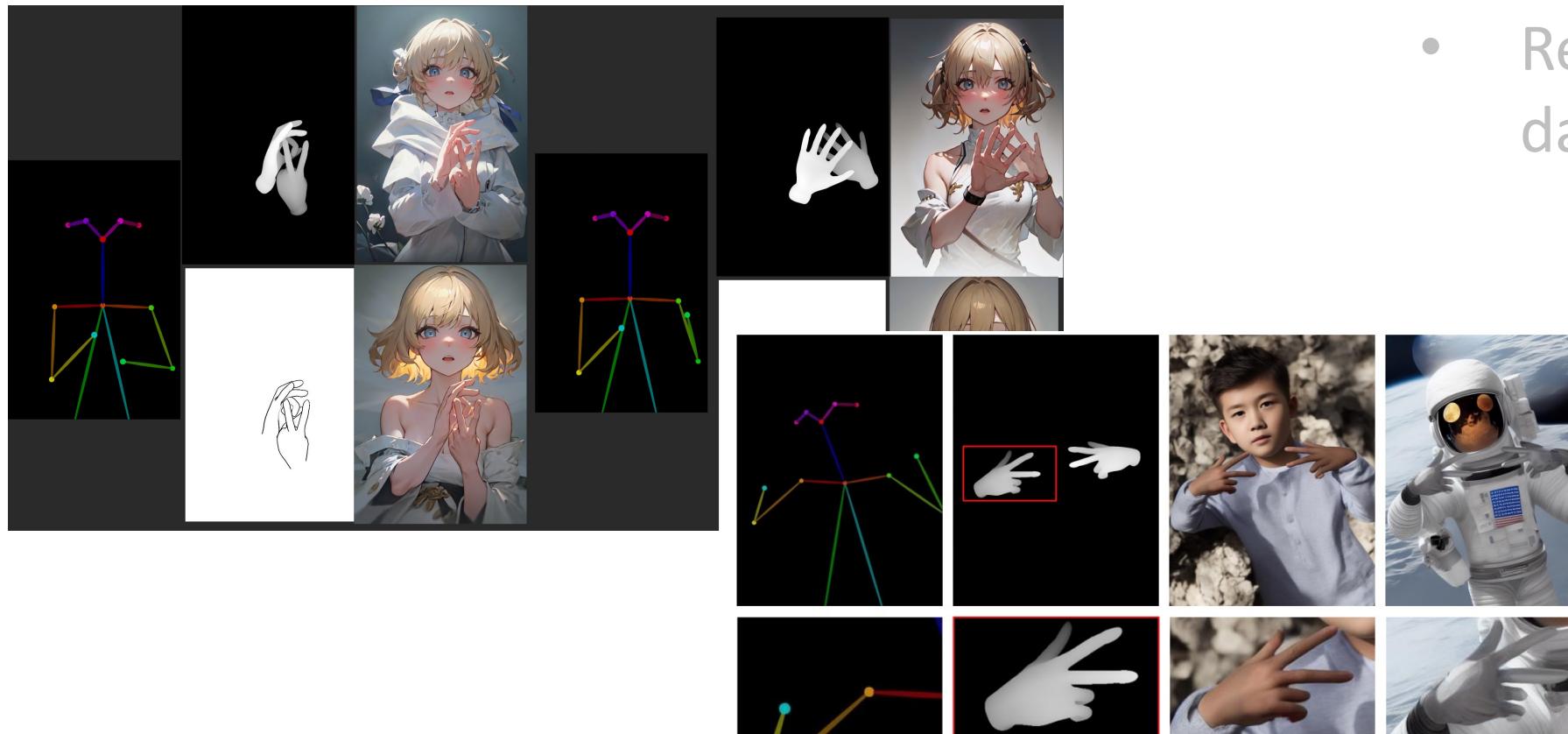
- Composable control (multiple conditions)
- Minimal influence to the base model (the base model can be changed)
- Reduced overfitting risk (training with small dataset becomes easier)

Using external model to process control signals

Finetuning diffusion
model weights

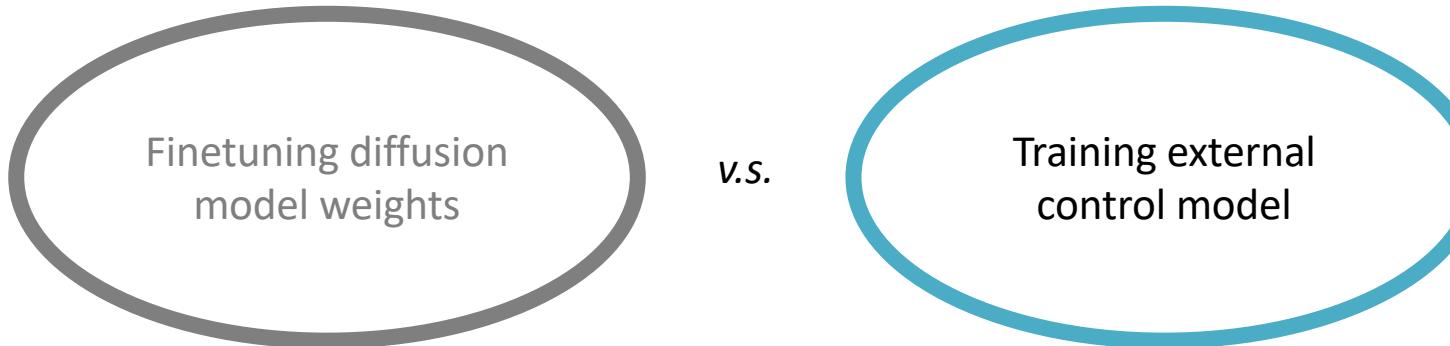
v.s.

Training external
control model



- Composable control (multiple conditions)
- Minimal influence to the base model (the base model can be changed)
- Reduce overfitting (training with small dataset becomes easier)

Using external model to process control signals



- Composable control (multiple conditions)
- Minimal influence to the base model (the base model can be changed)
- Reduce overfitting (training with small dataset becomes easier)



“house”



SD 1.5



Comic Diffusion



Progen 3.4

Using external model to process control signals

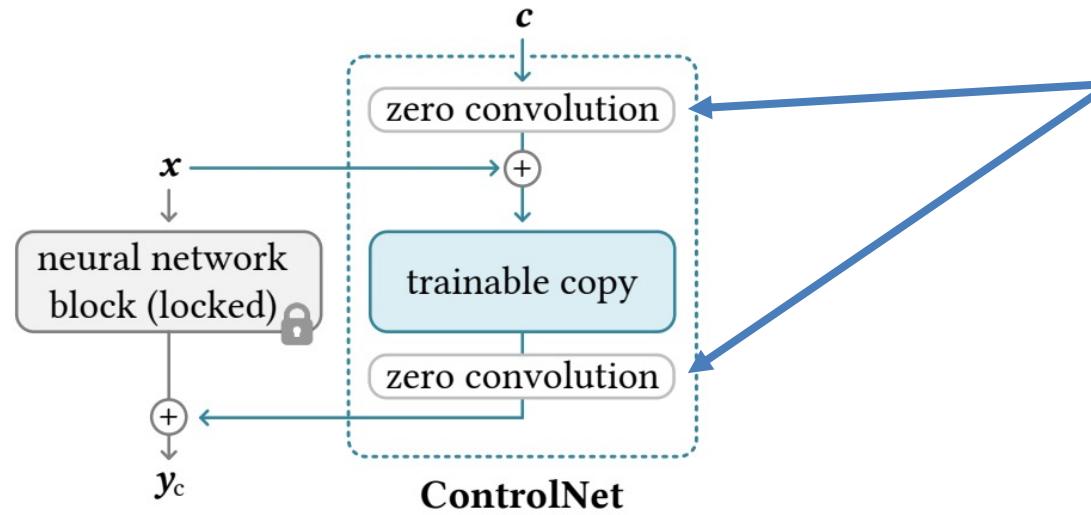


without ControlNet
(using Stability's "official" method to add
the channels to input layer, same as their
depth-to-image structure)

SD + ControlNet

- Composable control (multiple conditions)
- Minimal influence to the base model (the base model can be changed)
- Reduce overfitting (training with small dataset becomes easier)

Using zero-initialized layers to reduce initial noise



Zero-initialized connection layers

- Reduce initial harmful noise
- Protect the trainable copy

Applications



Input Canny edge

Default

"masterpiece of fairy tale, giant deer, golden antlers"

"..., quaint city Galic"



Input human pose

Default

"chef in kitchen"

"Lincoln statue"

Applications

Control Stable Diffusion with Lineart

Image

Prompt
bag

Run

Images
5

Seed
12345

Preprocessor
 Lineart Lineart_Coarse None

Advanced options

Control Stable Diffusion with Lineart

Image

Prompt
wolf

Run

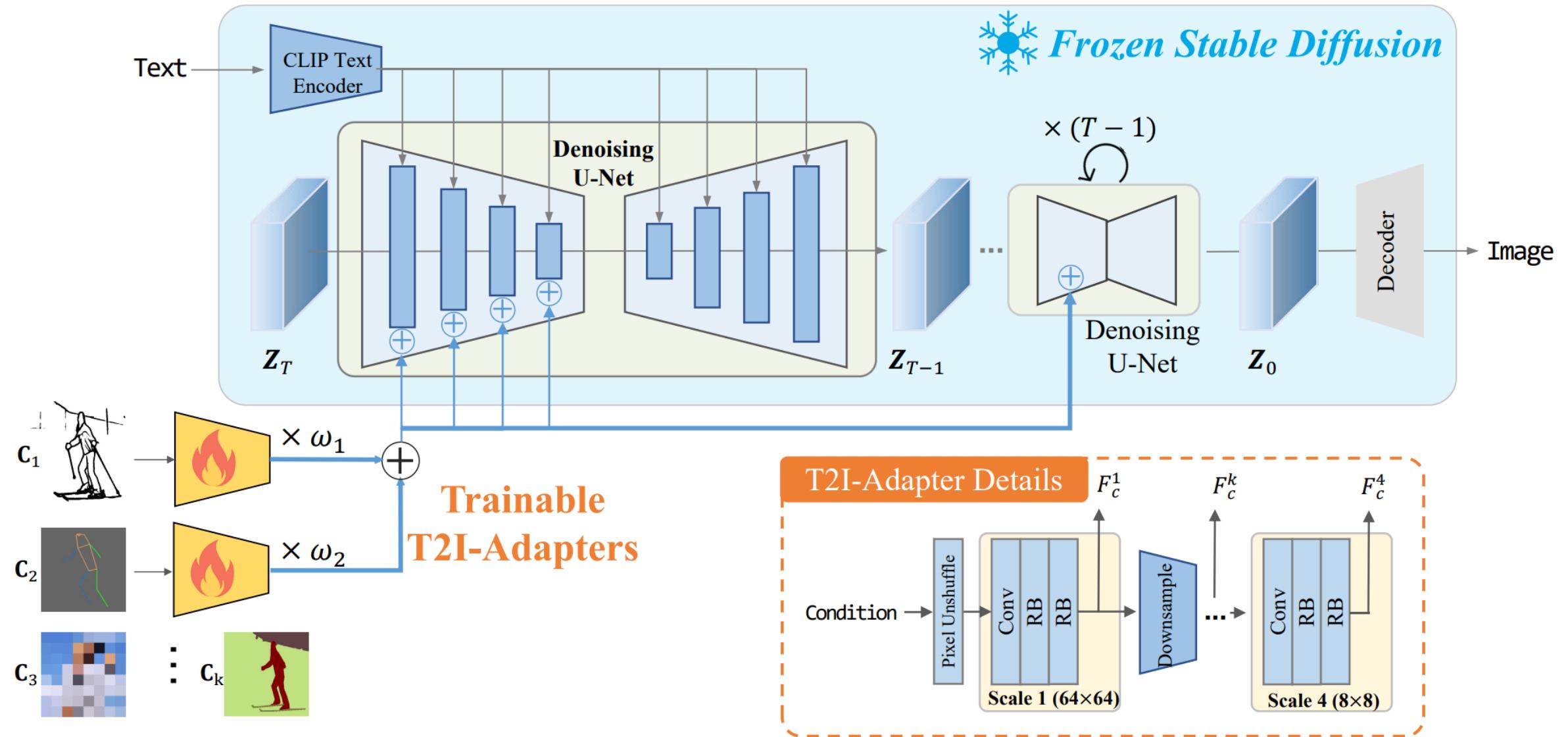
Images
5

Seed
12345

Preprocessor
 Lineart Lineart_Coarse None

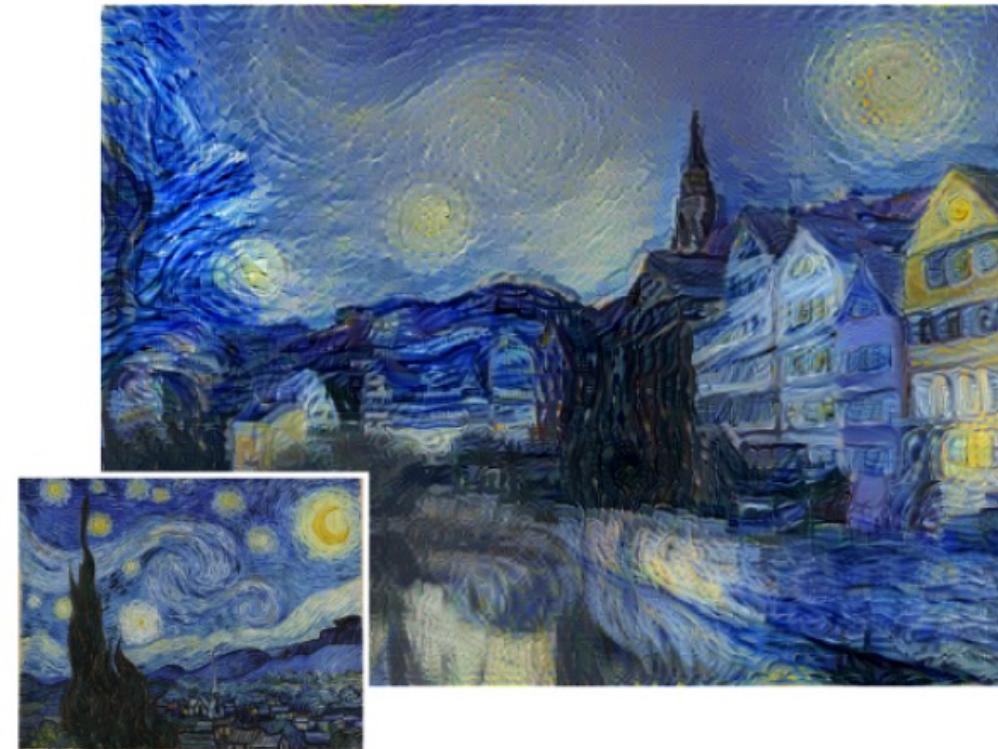
Advanced options

Conditional Diffusion Models



Conditional Diffusion Models

- Lightweight
 - The backbone model weight is frozen.
- Composable
 - use multiple controls (adapters) together (e.g., depth+pose)
- Generalizable and reusable
 - Can work with other backbone models



Style and Content, Texture Synthesis

Jun-Yan Zhu

16-726, Spring 2025

Texture

- Texture depicts spatially repeating patterns
- Many natural phenomena are textures



radishes



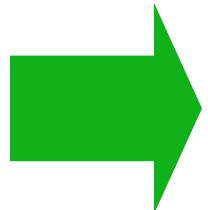
rocks



yogurt

Texture Synthesis

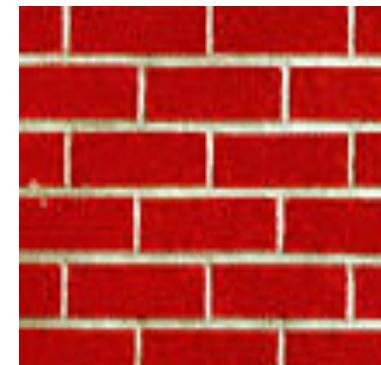
- Goal: create new samples of a given texture
- Applications: virtual environments, inpainting, texturing surfaces



Non-parametric Texture Synthesis

The Challenge

- Need to model the whole spectrum: from repeated to stochastic texture



repeated

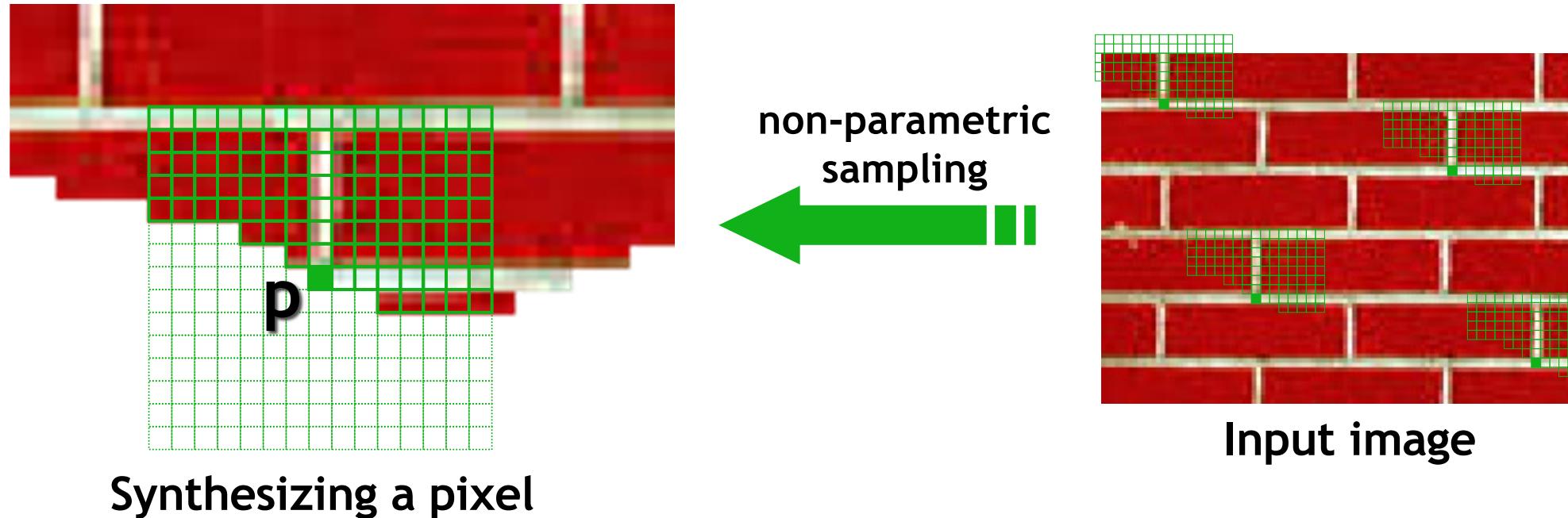


stochastic



Both?

Efros & Leung Algorithm

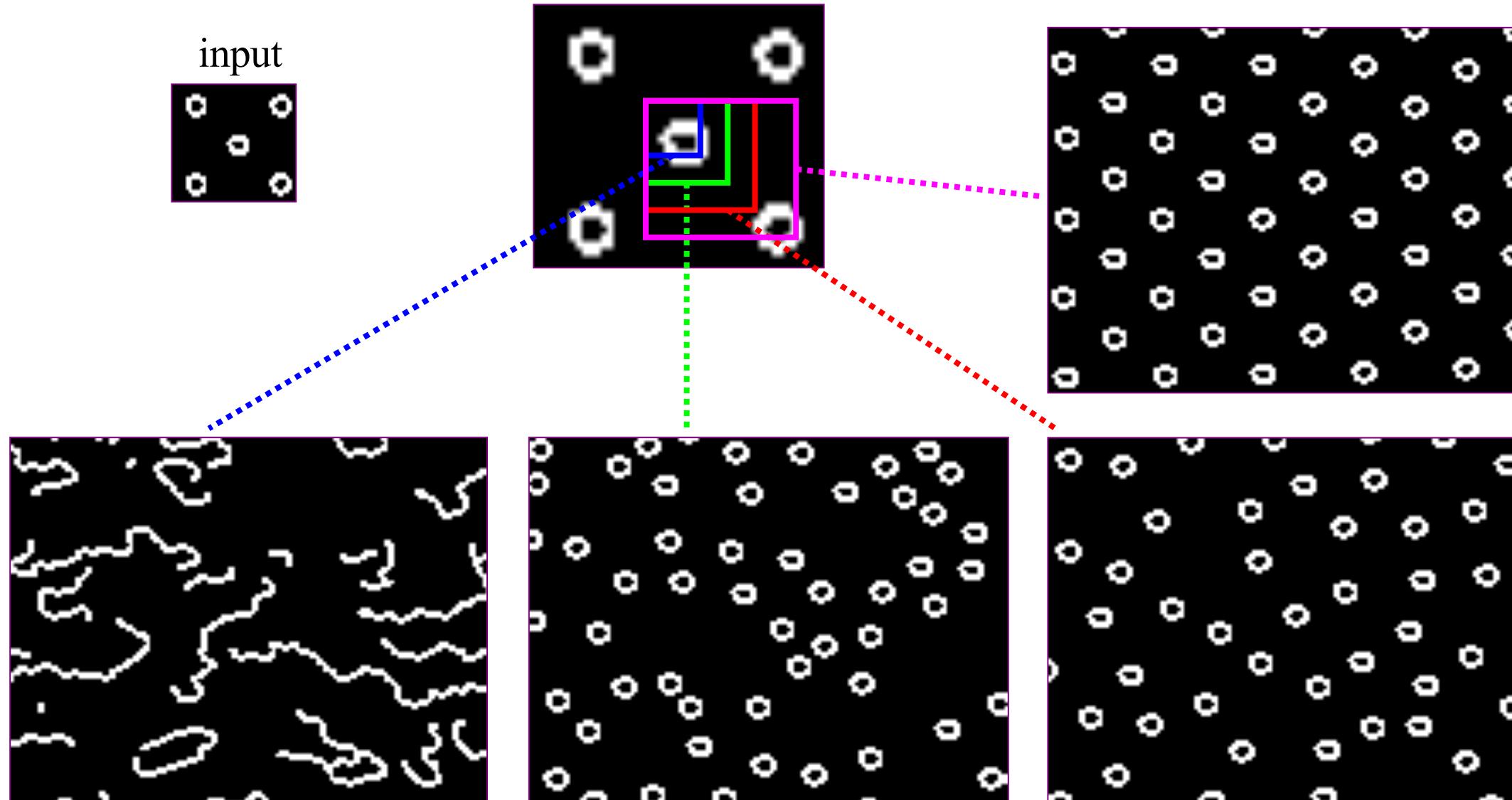


- Assuming Markov property, compute $P(p | N(p))$
 - Building explicit probability tables infeasible
 - Instead, we *search the input image* for all similar neighbourhoods — that's our pdf for p
 - To sample from this pdf, just pick one match at random

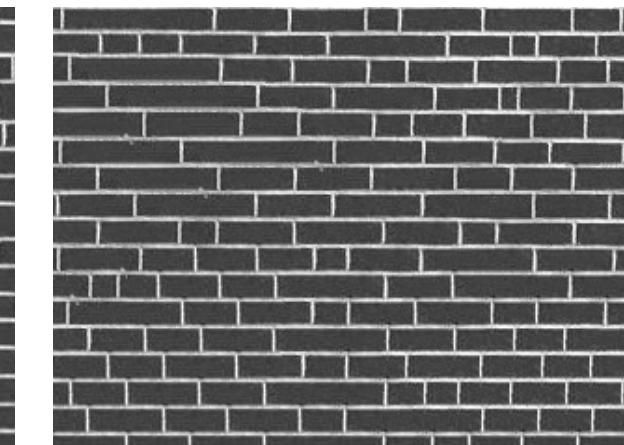
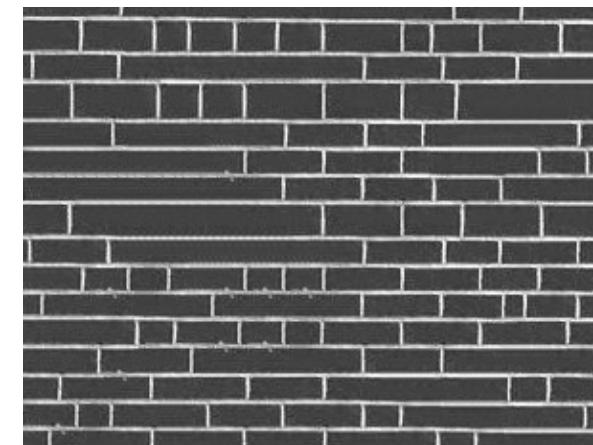
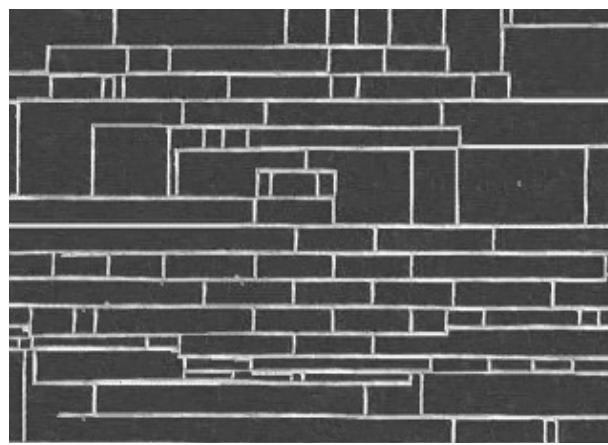
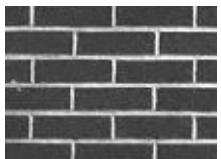
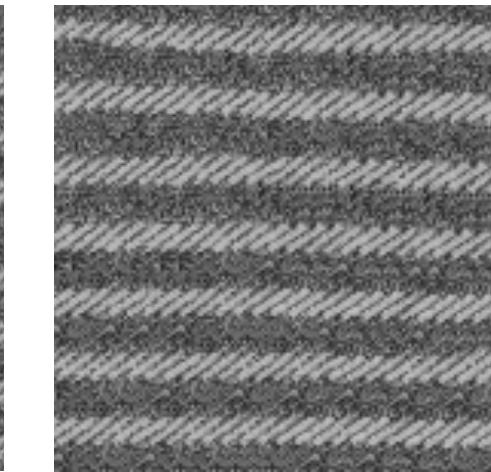
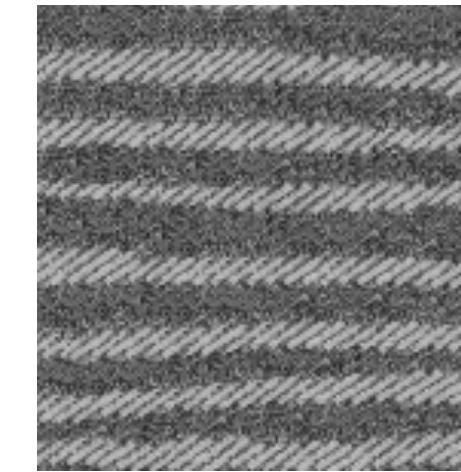
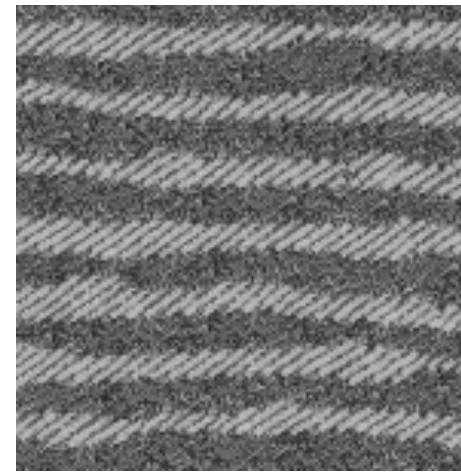
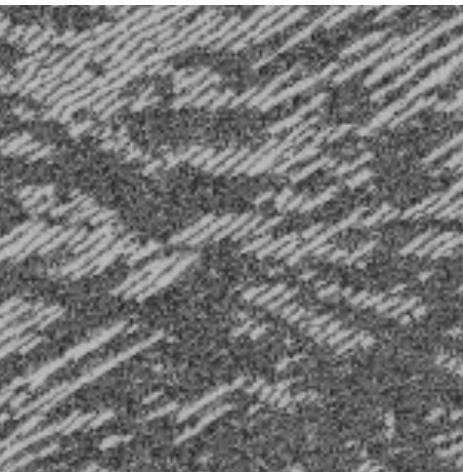
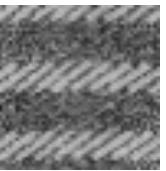
Some Details

- Growing is in “onion skin” order
 - Within each “layer”, pixels with most neighbors are synthesized first
 - If no close match can be found, the pixel is not synthesized until the end
- Using *Gaussian-weighted* SSD is very important
 - to make sure the new pixel agrees with its closest neighbors
 - Approximates reduction to a smaller neighborhood window if data is too sparse

Neighborhood Window



Varying Window Size

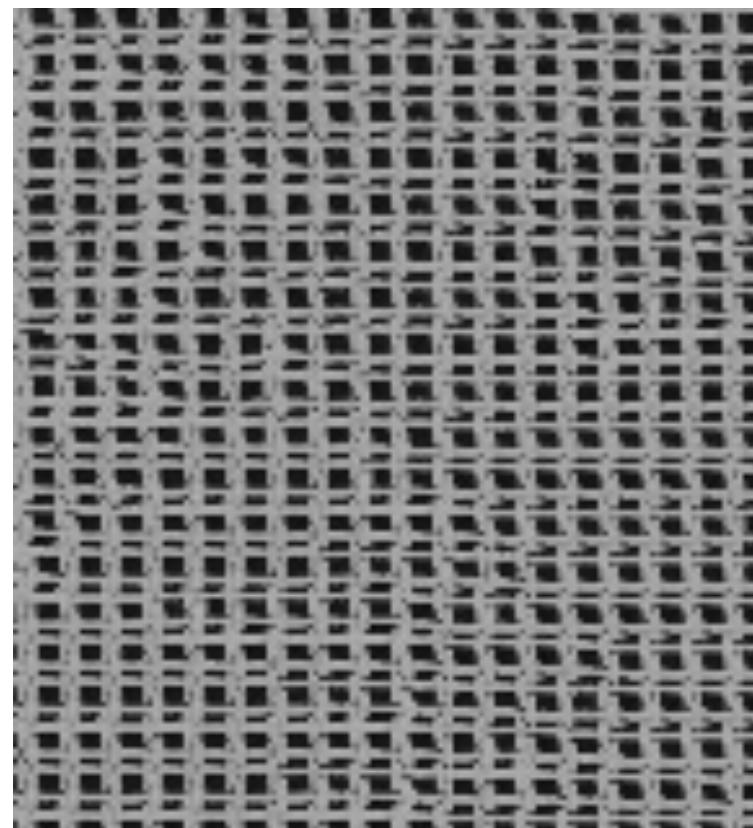
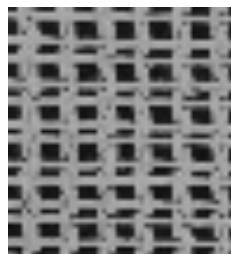


Increasing window size

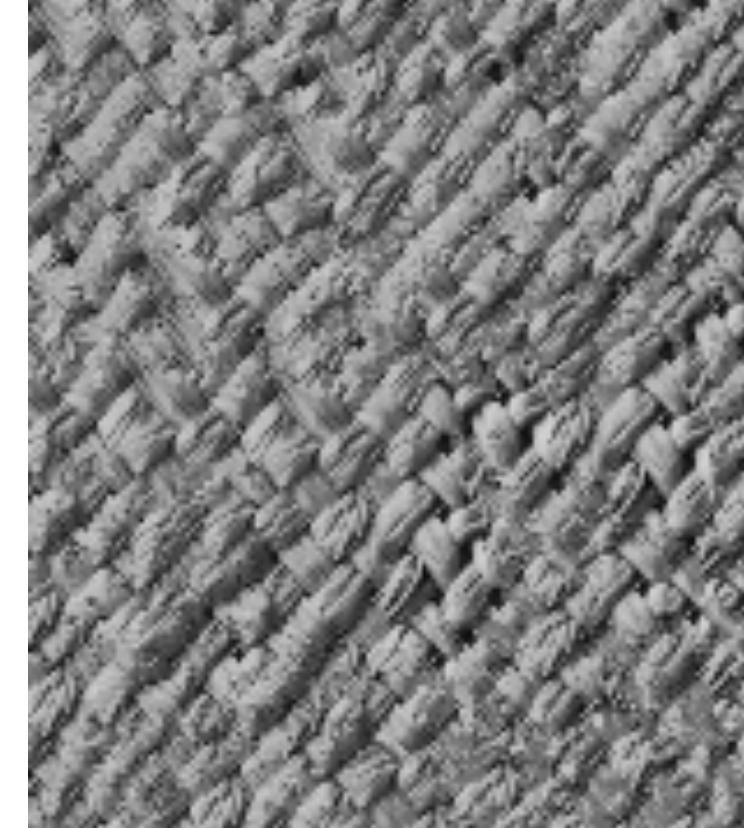
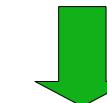
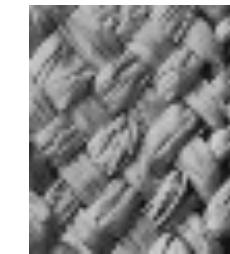


Synthesis Results

french canvas

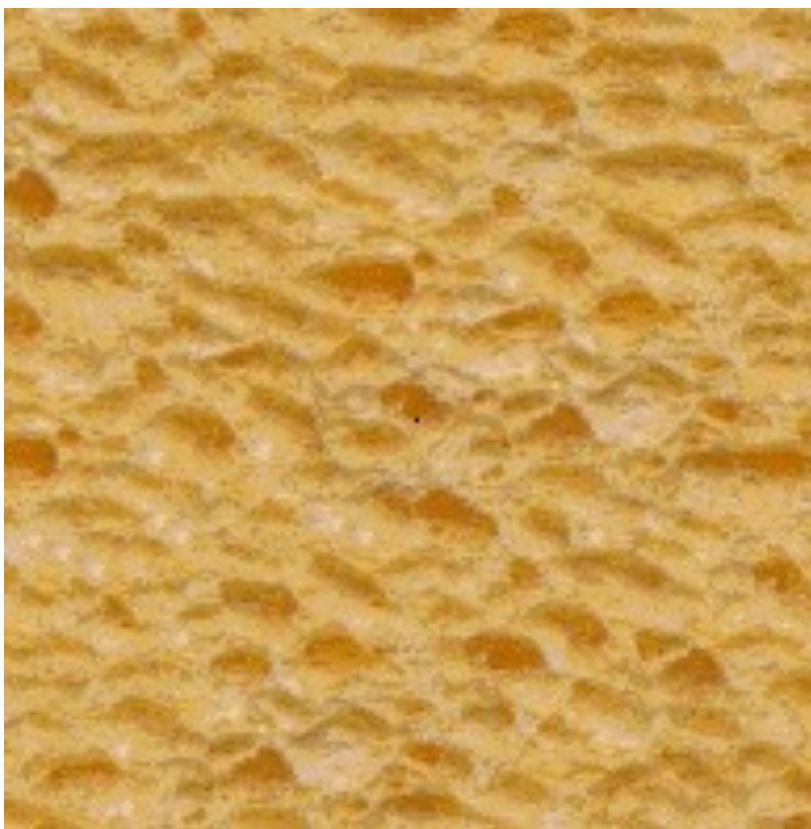


rafia weave

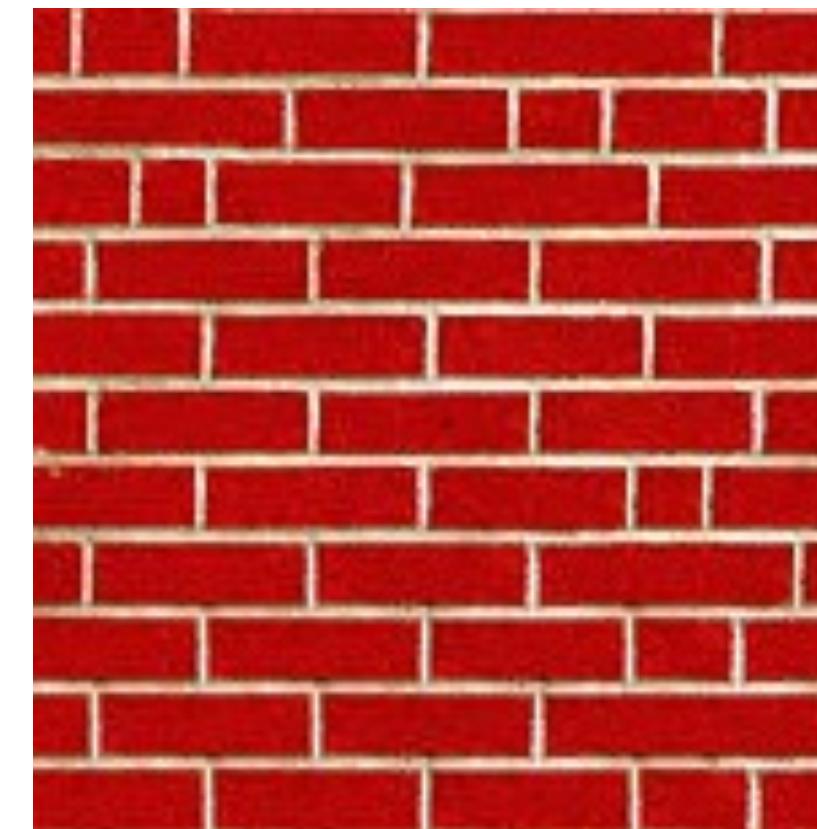
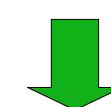
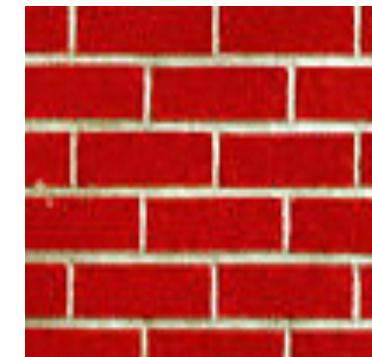


More Results

white bread



brick wall



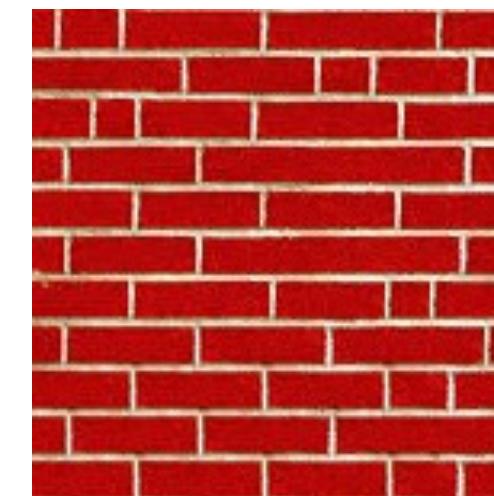
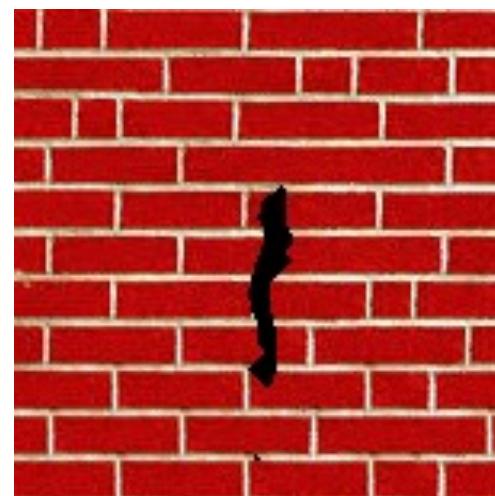
Homage to Shannon

uring in the unsensatior
r Dick Gephardt was fai
rful riff on the looming
nly asked, "What's your
tions?" A heartfelt sigh
story about the emergenc
es against Clinton. "Boy
g people about continuin
ardt began, patiently obs
s, that the legal system h
e with this latest tanger

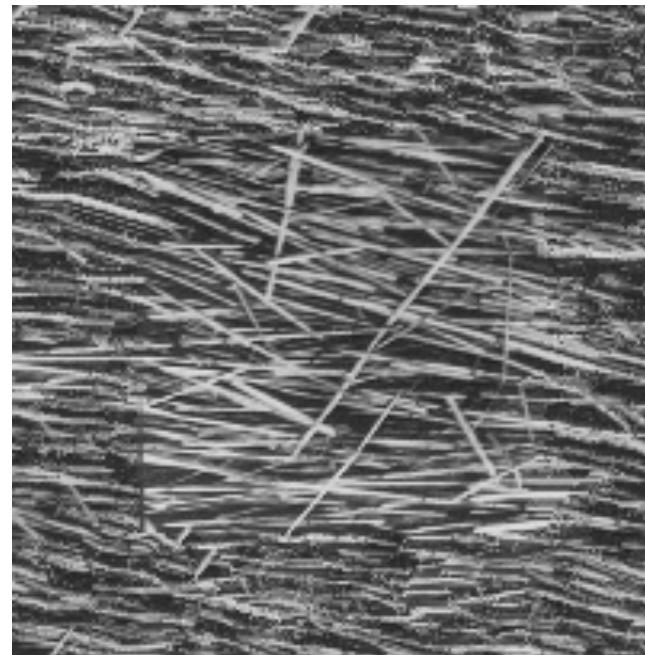
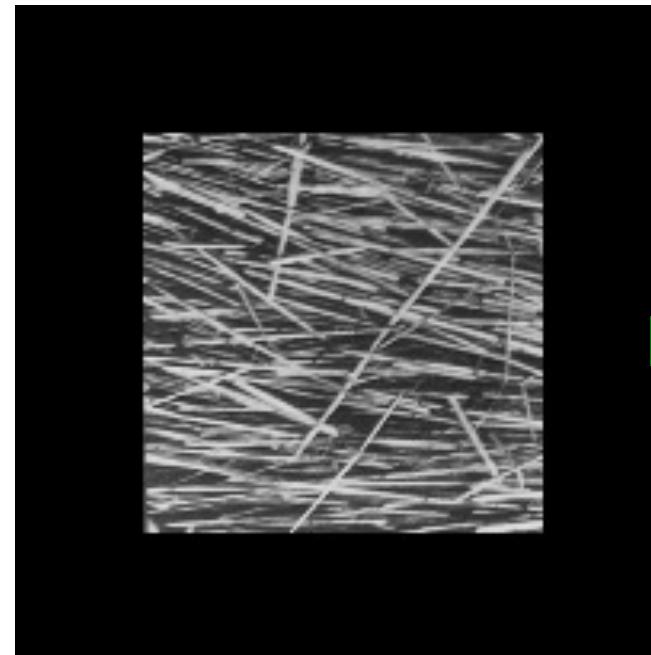
I b' s'no p'ob'p'ieh' h'v' h'ht' l'mr ai thanne' t
e g'le per' s' otot' " la' t'tin' n'irth' feori' P'unt' a
v'as "he' d'c'l' e'or' t' a' o' A' l' b'
h'c' 'y'f' te' t' fit' a'us? t'ri'ooear' J'cat' h'c'
t'iae' " t'dab' h'bb'i' e' h'j'h'mr' te' opm' t'P' v'ur' d'
e' t'f'it' t' s' l' t' h'c' h'c' h'c' h'c' h'c'
n' t' se' ' w'is' b'nt' u'rn' h'c' h'c' h'c' h'c' h'c'
n' id' r'f' p'f' e' p'j'dt' g'lag' h'c' h'c' h'c' h'c' h'c'
u'ton'c' f' u'hes' i'f' n' A'nf' er' h'c' h'c' h'c' h'c' h'c'
" j'henly' n' A'nf' er' h'c' h'c' h'c' h'c' h'c'
d'thf' p'f' d' t' l' o' n' h'c' h'c' h'c' h'c' h'c'
r' f' fa' if' h'c' h'c' h'c' h'c' h'c'
t' l' b' s' h'c' hsk' as' h'c' k' y's' h'c' h'c' h'c' h'c' h'c'
e' en' s' t' h'c' h'c' h'c' h'c' h'c' h'c' h'c' h'c' h'c'
l' t' in' t' r' l' t' n' t' " t' f'ff' a'fe' c' d' t' e' l' r' h'c' h'c' h'c' h'c' h'c'
s' C'z' h'c'
s' t' h'c'
t' h'c'
d' v'ite' h'c'
t' o' p'f' h'c'
t' o' p'f' h'c'
t' o' p'f' h'c'
t' o' p'f' h'c' h'c' h'c' h'c' h'c' h'c' h'c' h'c' h'c' h'c'

thaim, them . "Whnephartfe lartifelintomimen
fel ck Clirtioout omaim thartfelins,f aut' s anent
the ry onst wartfe lck Gephntoomimeationl sigak
Chiooufit Clinut Cll riff on, hat's yordn, parut tly:
ons yoontonstehtwasked, paim t sahe loo' riff on l
nskoneploourtfeas leil A nst Clit, "Wleontongal s
k Cirtioouirtfepe óng pme abegal fartfenstemem
tienstenetorydt telemephmin'sverdt was agemer
ff ons artientont Cling peme asurtfe atish, "Boui s
nal s fartfelt sig pedr tl'dt ske abounutie aboutoo
tfeone newwas you abownhardt thatins fain, ped,
ains, them, pabout wasy arfiut cöuityl d, l n A h
ole einthringbooreme agas fa bontinsyst Clinut
ory about continst Clipeoininst Cloke agatiff out C
stome zinemen fly ardt beoraboul n, thenly as t C
cons faimeme Diontont wat coutlyohgans as fan
ien, phrtfaul, "Wbaut cout congagal comininga
mifmst Cliiy abon al coountha.emungairt tfoun
Whe looorystan loontieph, lntly on, theoplegatick
mul tatiezontly atie Diontiomt wal s f tbegae ener
nthahsat's enenhhmas fan, "intchhthnrw ahons v

Hole Filling



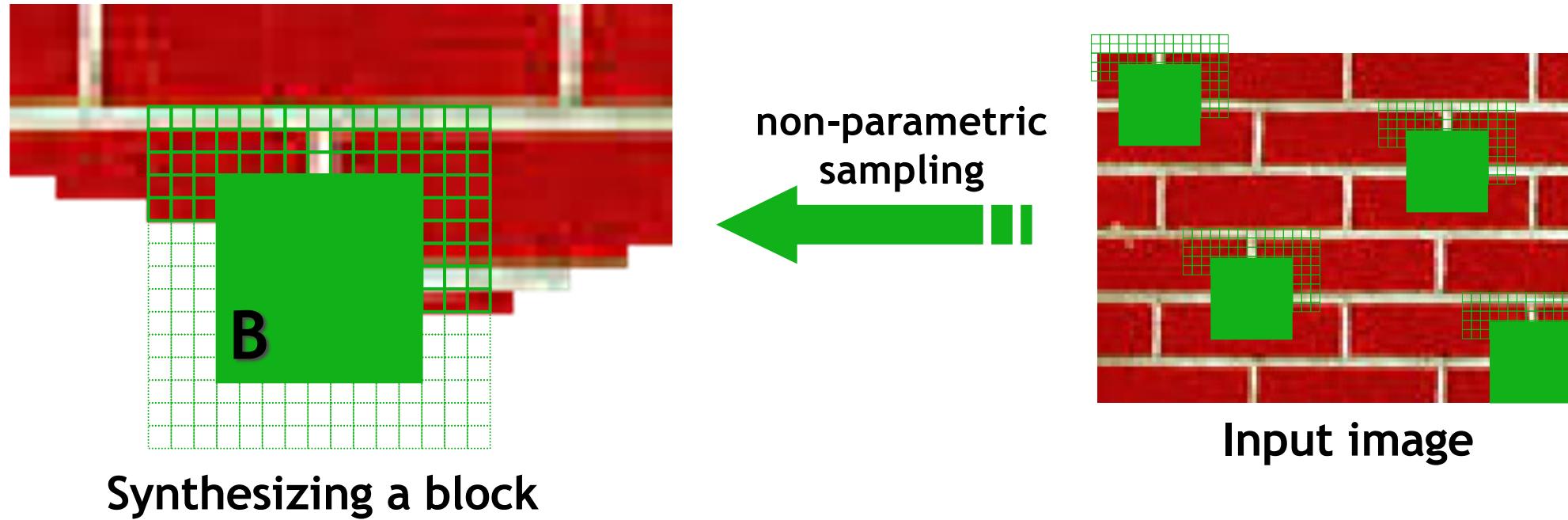
Extrapolation



Summary

- The Efros & Leung algorithm
 - + Very simple
 - + Surprisingly good results
 - + Synthesis is easier than analysis!
 - ...but very slow

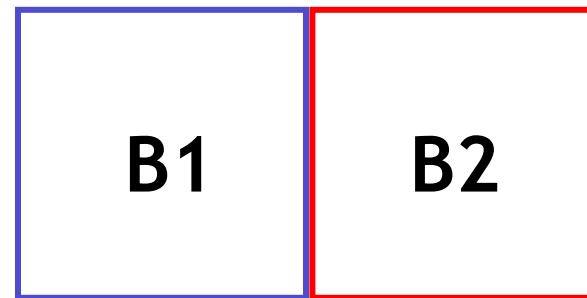
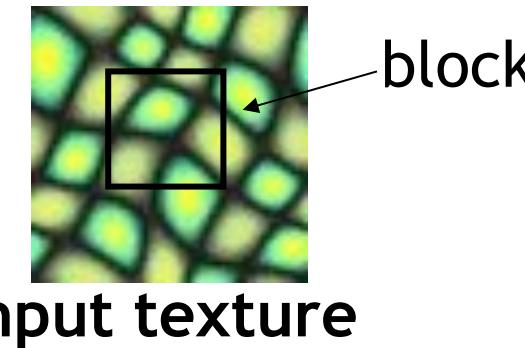
Image Quilting [Efros & Freeman]



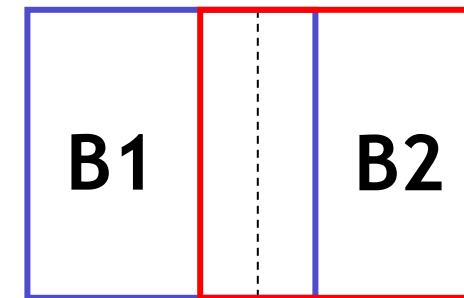
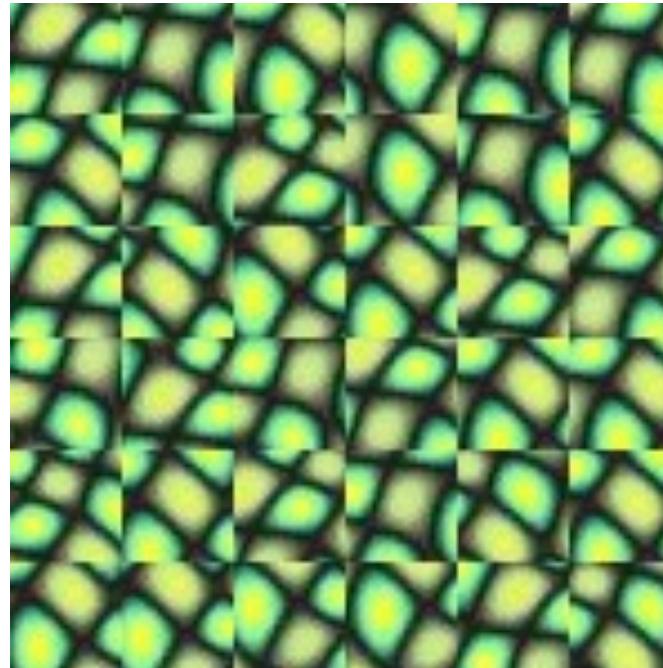
- Observation: neighbor pixels are highly correlated

Idea: unit of synthesis = block

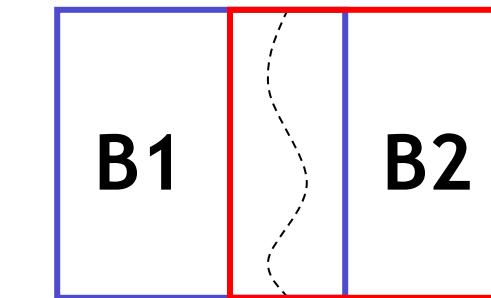
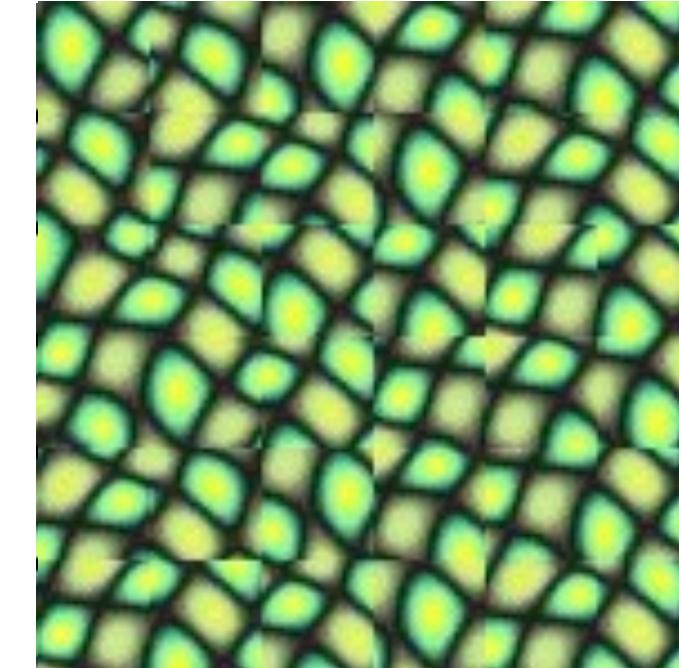
- Exactly the same but now we want $P(B | N(B))$
- Much faster: synthesize all pixels in a block at once
- Not the same as multi-scale!



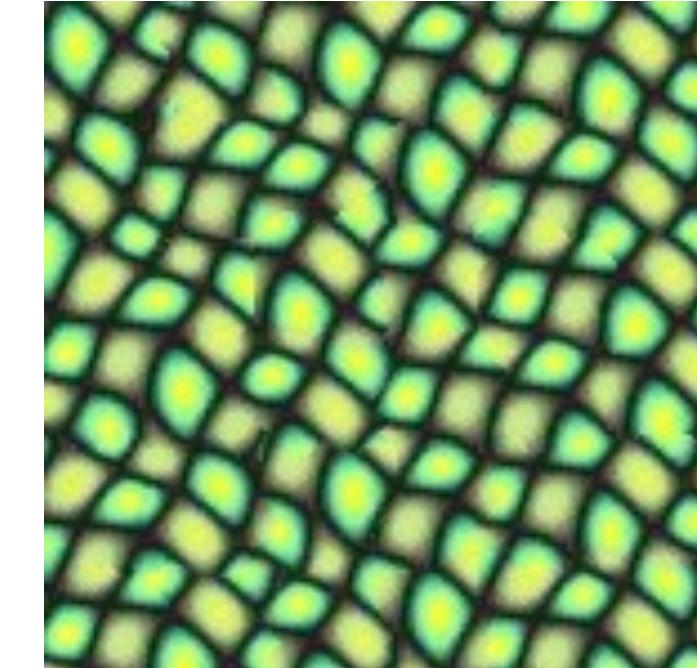
**Random placement
of blocks**



**Neighboring blocks
constrained by overlap**

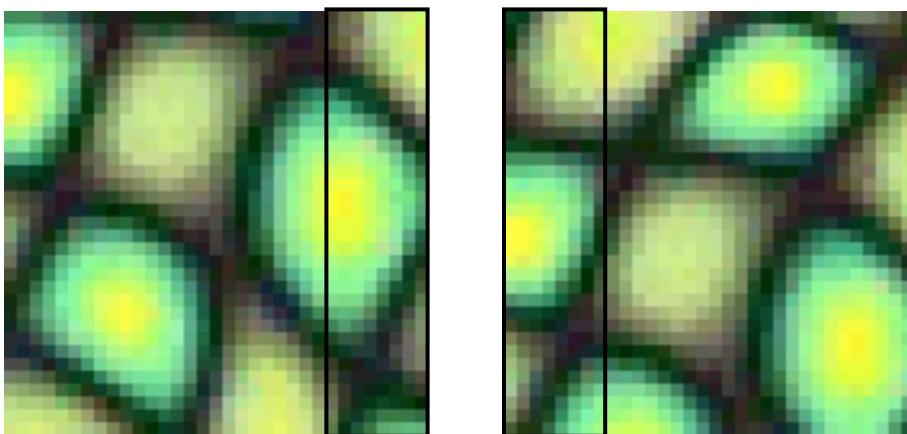


**Minimal error
boundary cut**

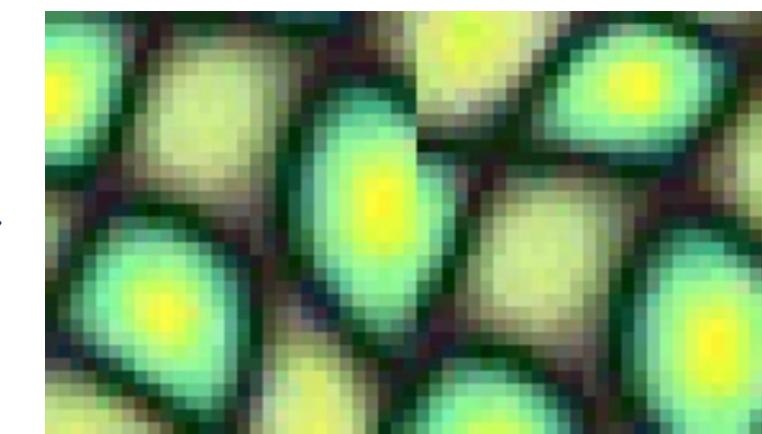


Minimal error boundary

overlapping blocks

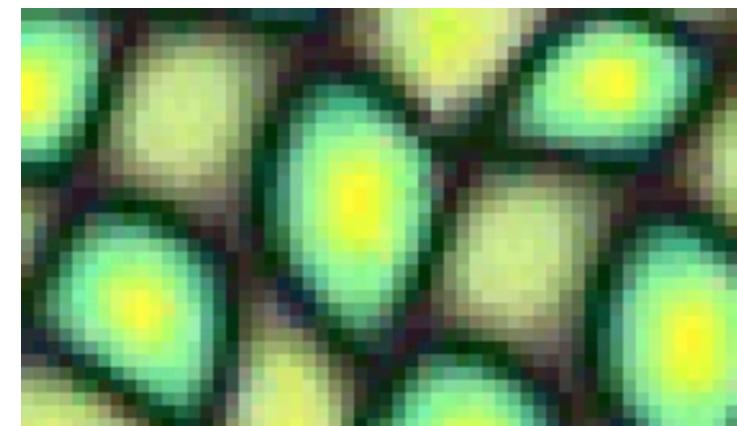


vertical boundary



$$\left(\begin{array}{c} \text{[Heatmap block]} \\ - \\ \text{[Heatmap block]} \end{array} \right)^2 = \text{[Binary mask with red border]}$$

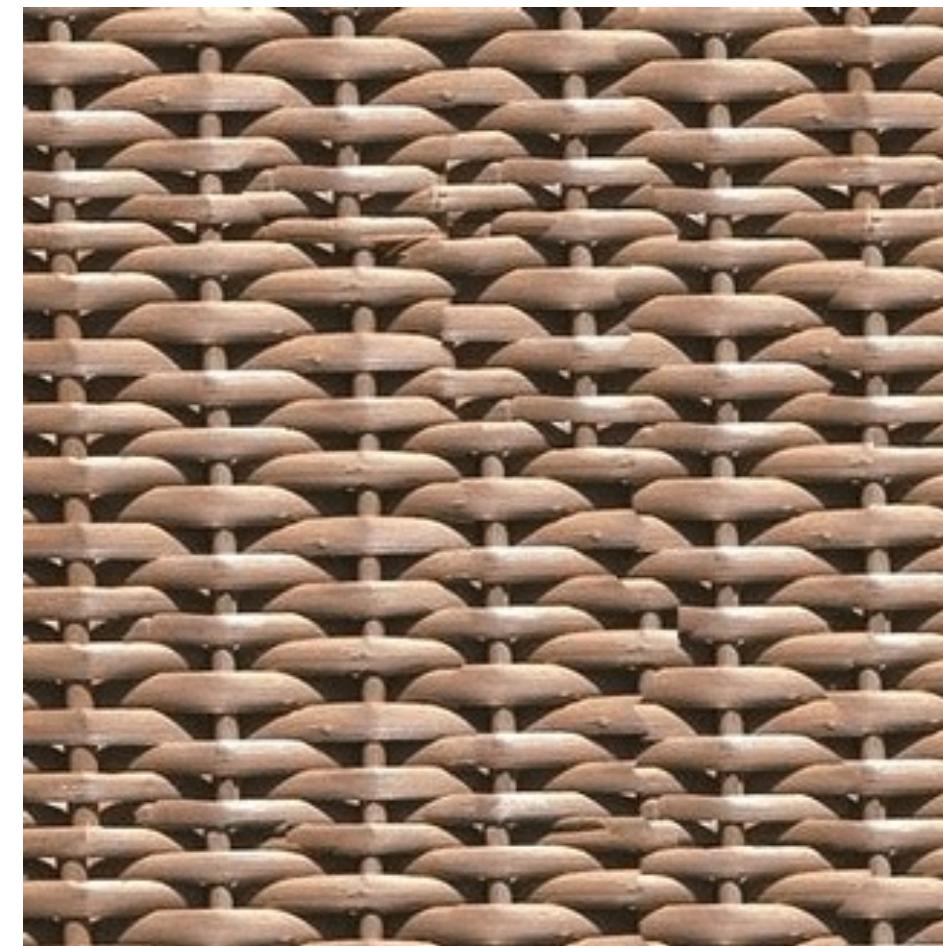
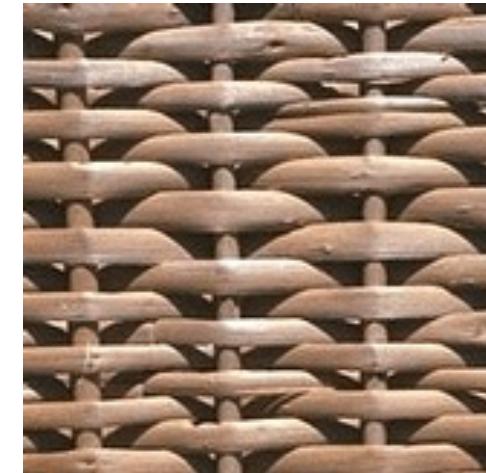
overlap error

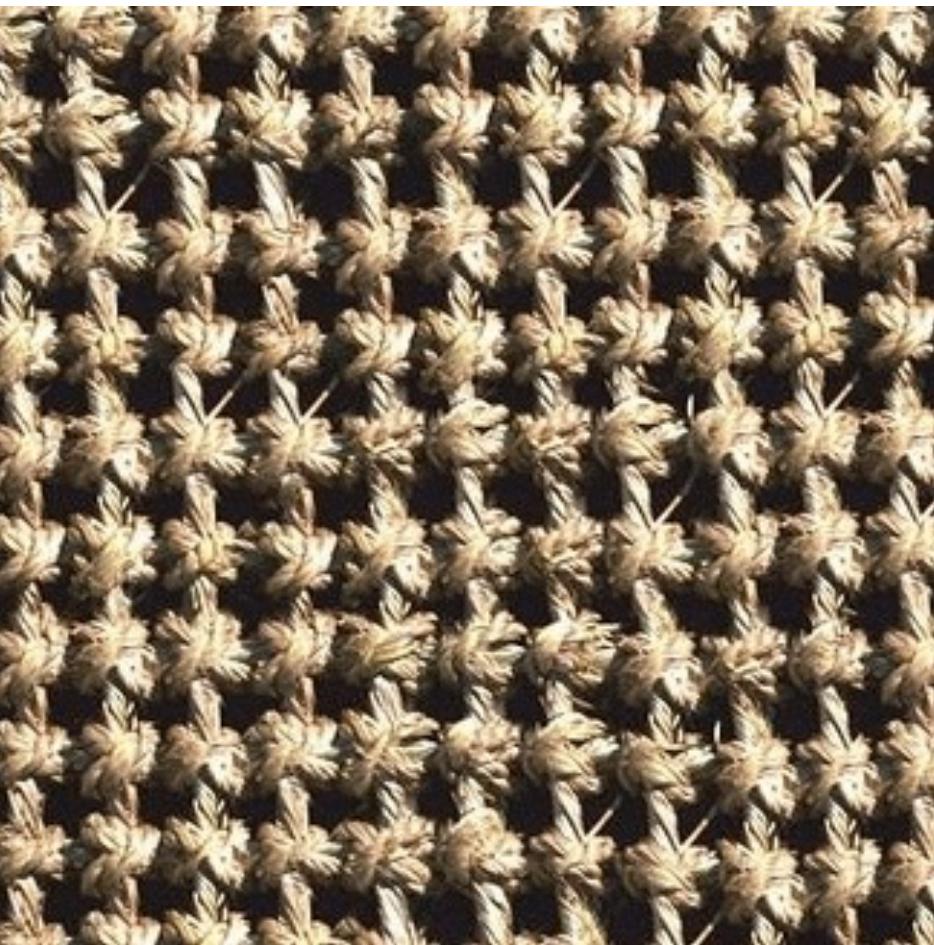


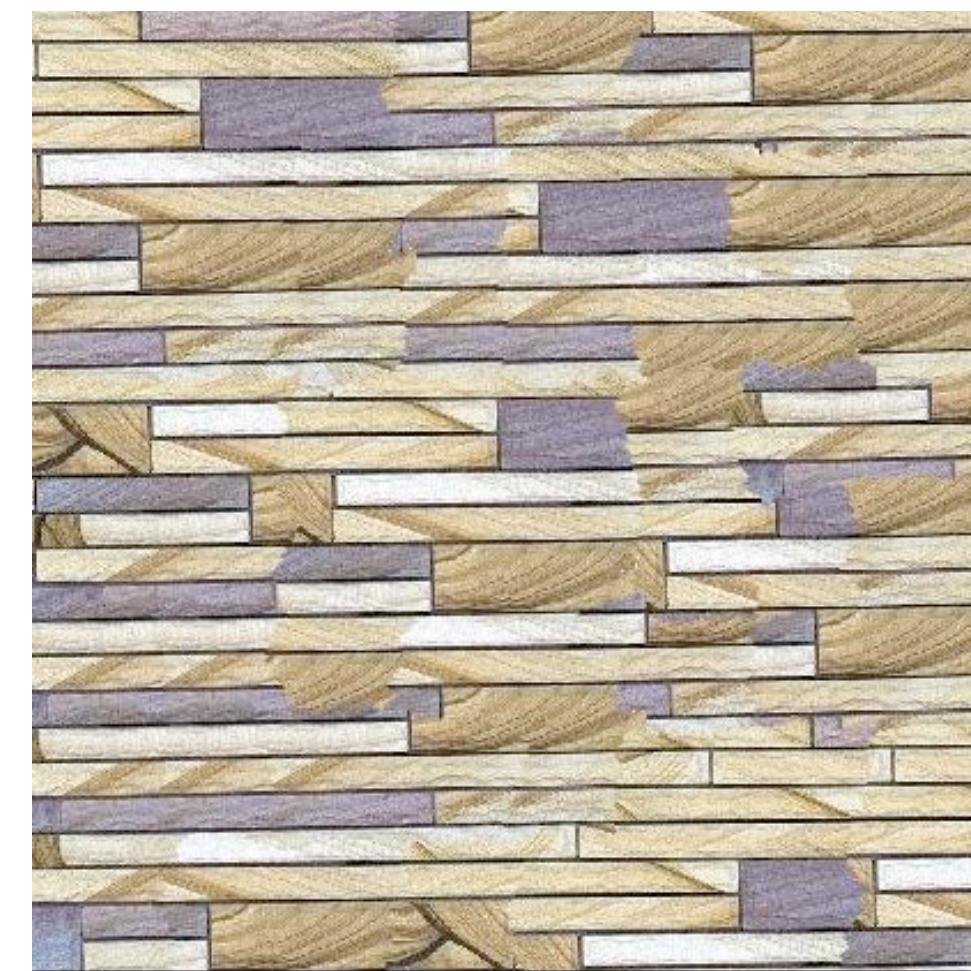
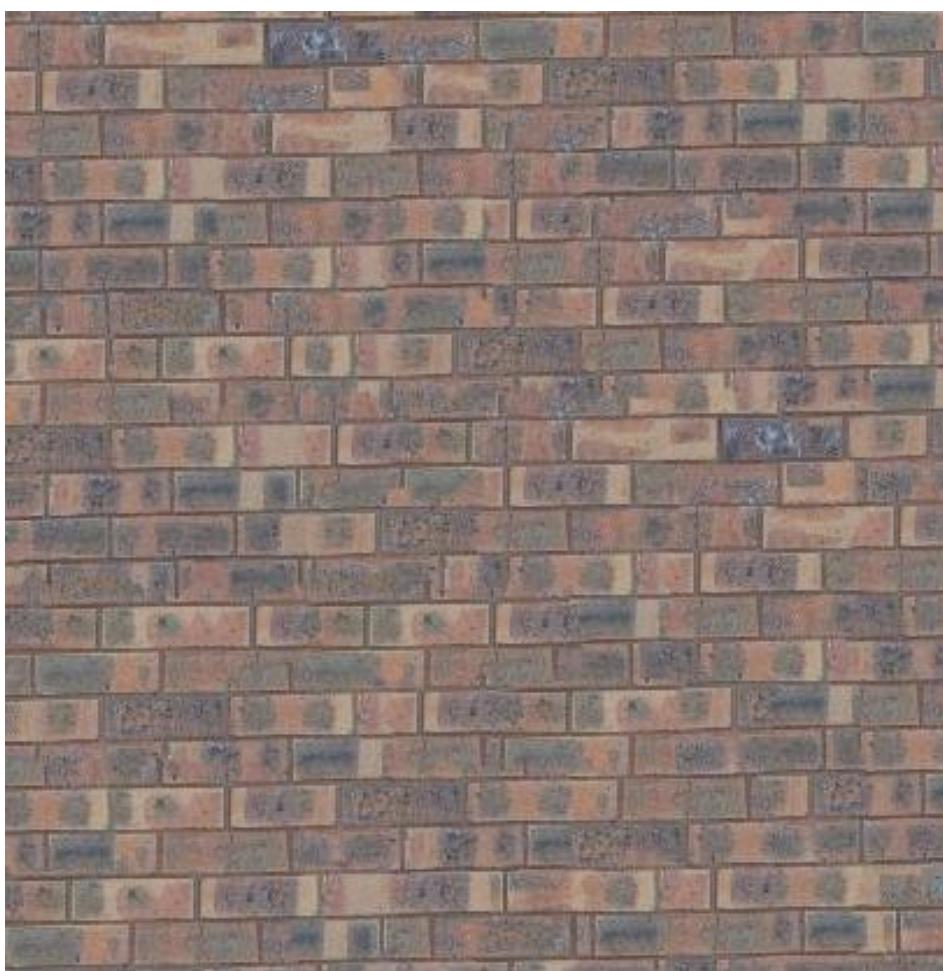
min. error boundary

Our Philosophy

- The “Corrupt Professor’s Algorithm”:
 - Plagiarize as much of the source image as you can
 - Then try to cover up the evidence
- Rationale:
 - Texture blocks are by definition correct samples of texture so problem only connecting them together

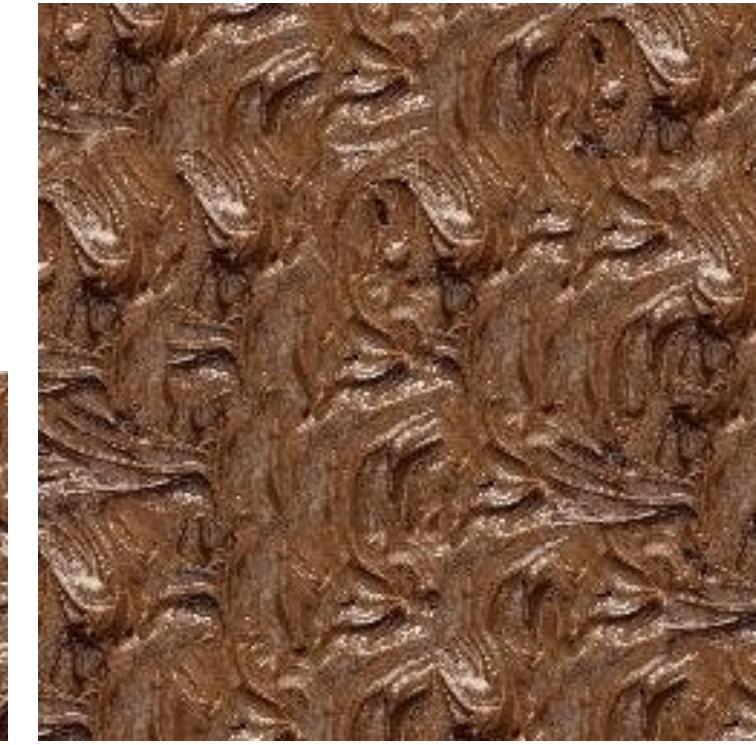
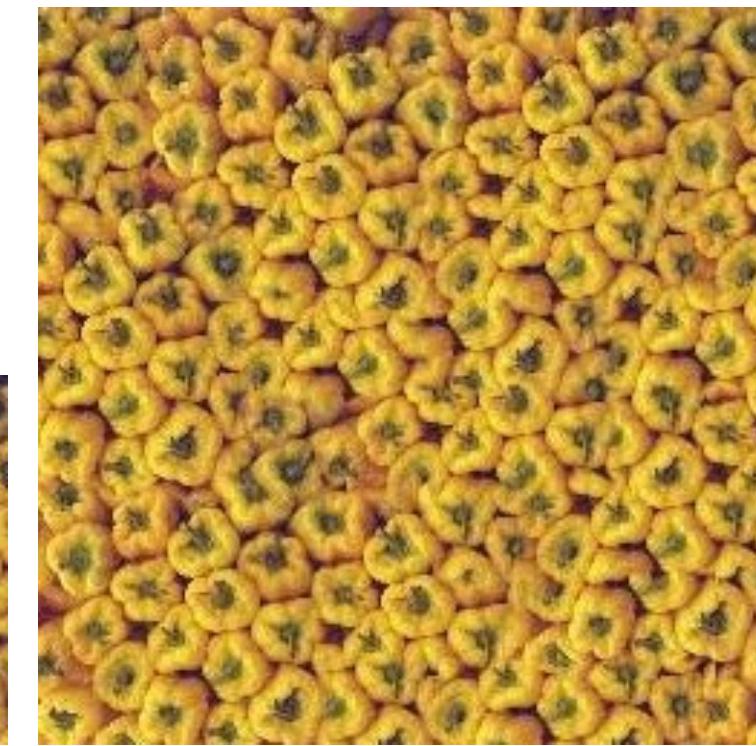
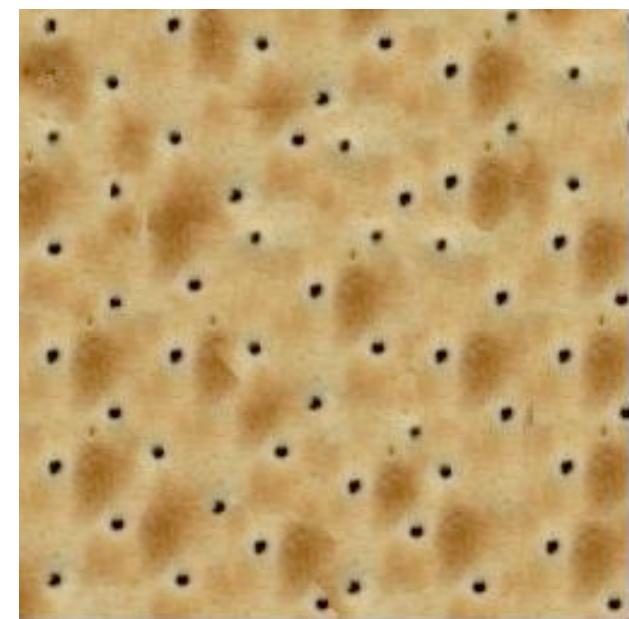
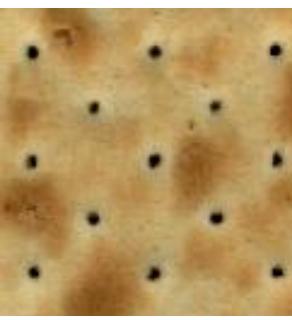






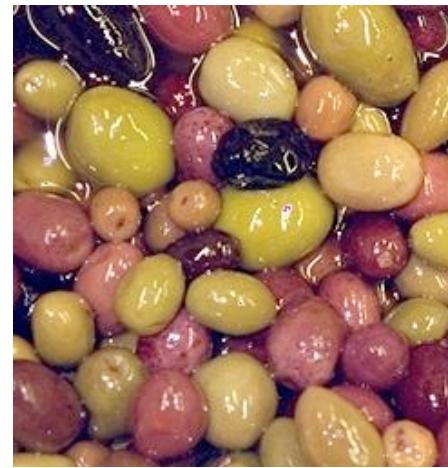


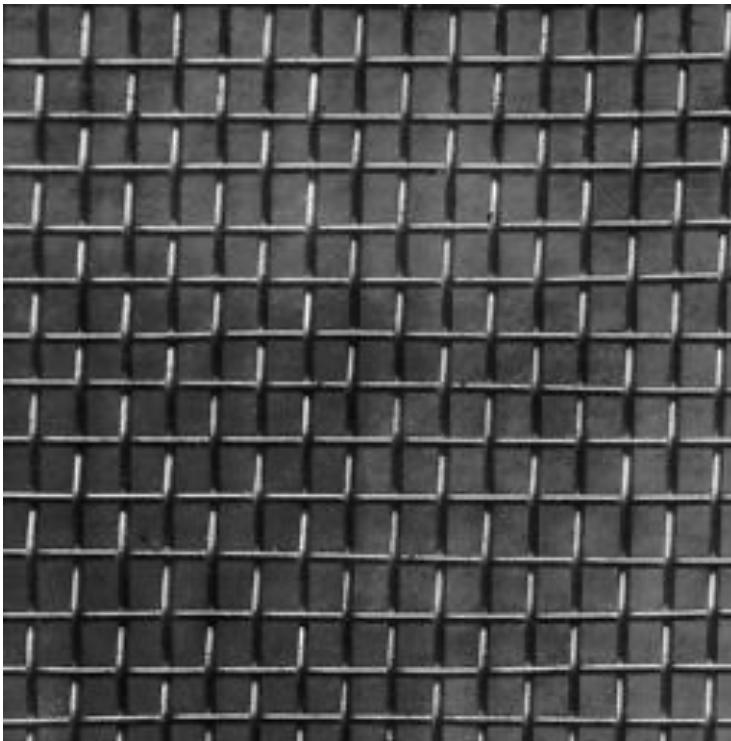




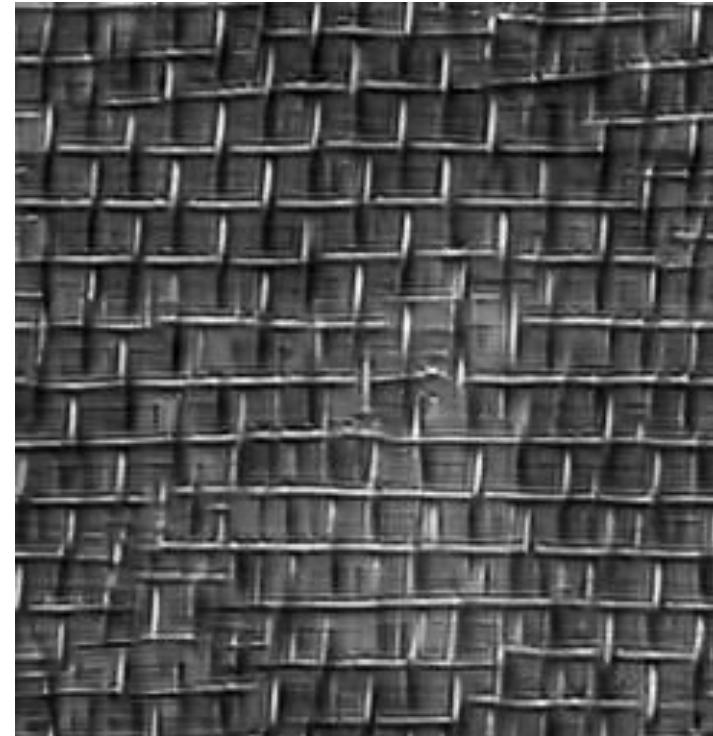


Failures (Chernobyl Harvest)

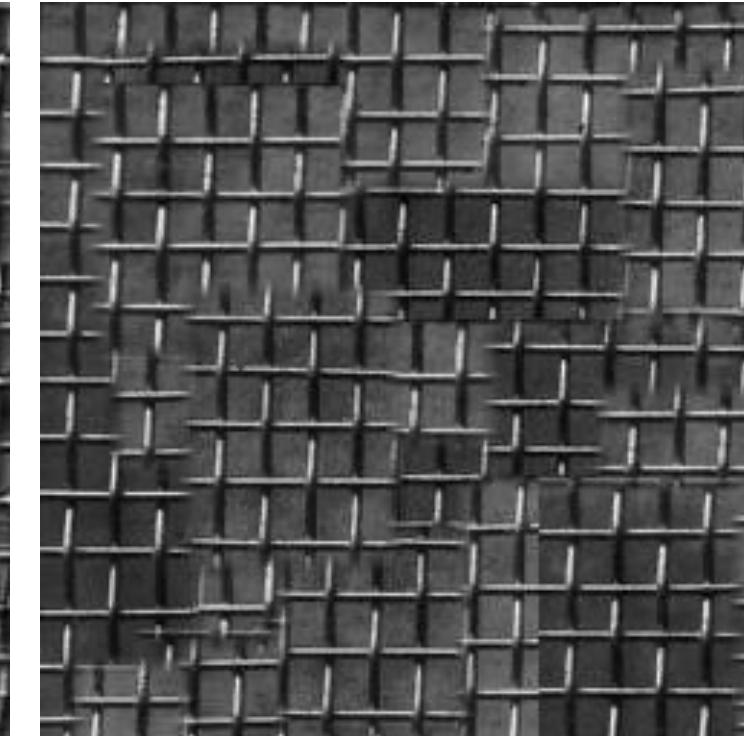




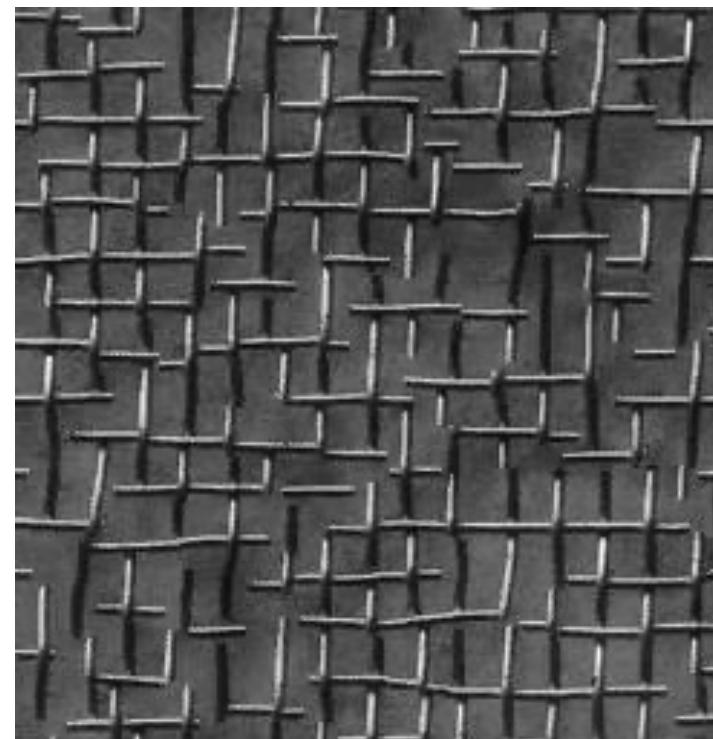
input image



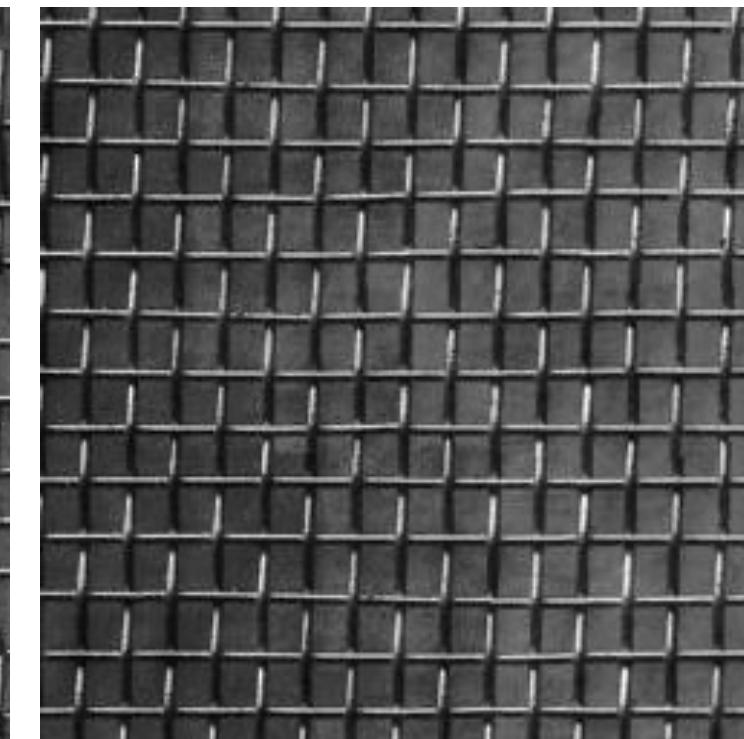
Portilla & Simoncelli



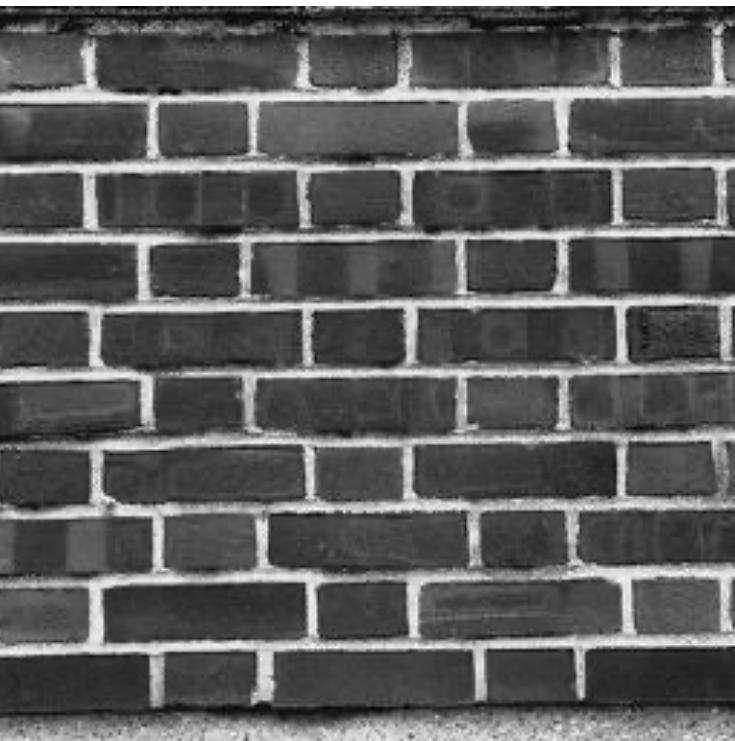
Xu, Guo & Shum



Wei & Levoy



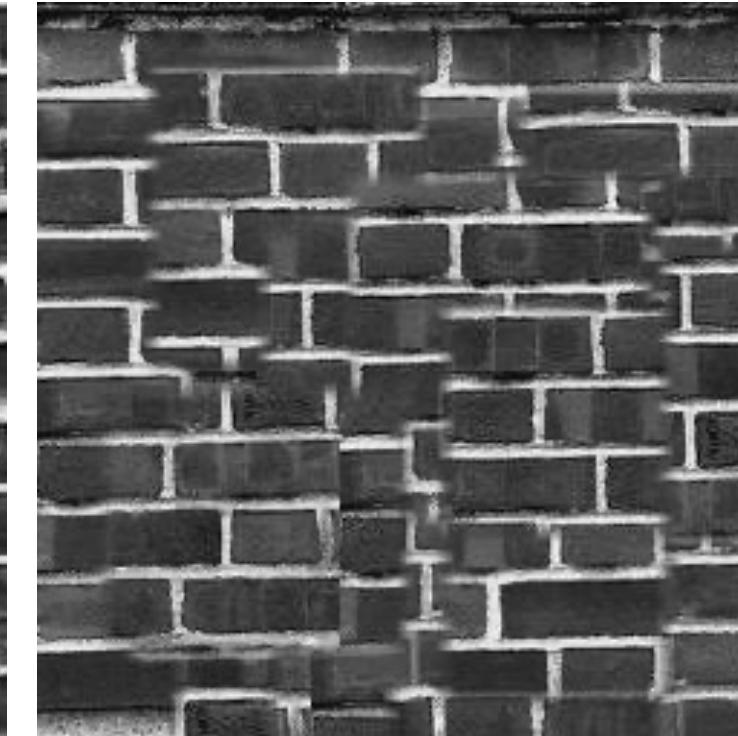
Efros and Freeman



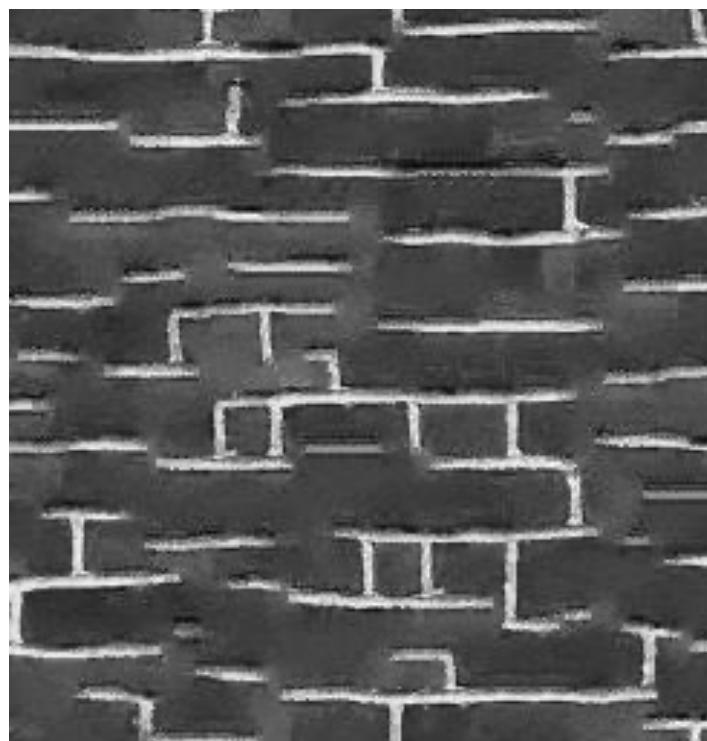
input image



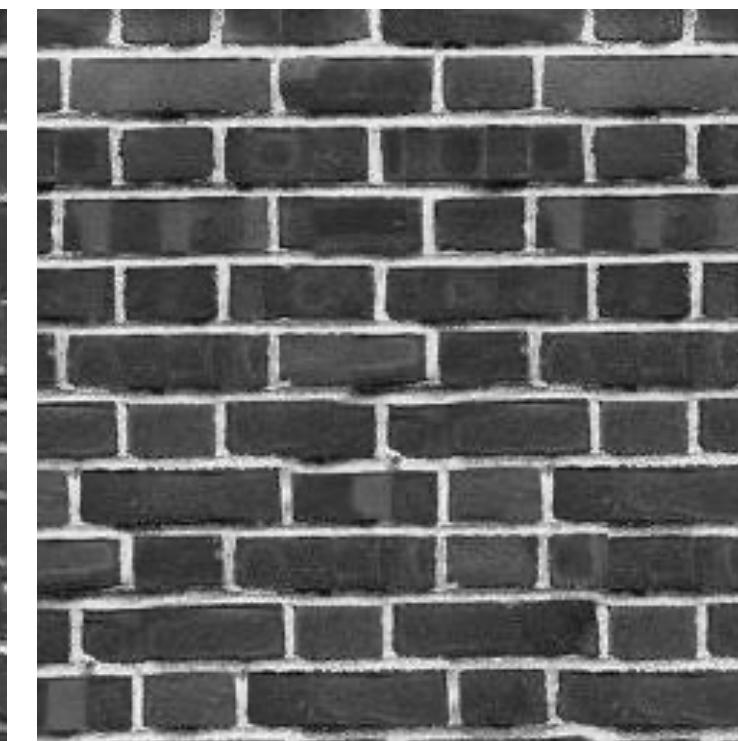
Portilla & Simoncelli



Xu, Guo & Shum



Wei & Levoy



Efros and Freeman

describing the response of that neuron as a function of position—is perhaps the most functional description of that neuron. We seek a single conceptual and mathematical framework to describe the wealth of simple-cell receptive fields neurophysiologically¹⁻³ and inferred especially if such a framework has the virtue of helping us to understand the function in a deeper way. Whereas no generic model can account for all simple-cell receptive fields (DOG), difference of offset Gabor, derivative of a Gaussian, higher derivative of a Gaussian, higher derivative function, and so on—can be expected to account for the simple-cell receptive field, we nonetheless

input image

Portilla & Simoncelli

... sen ons
ols nnuunncce tiarpn, nelolc ewiomsir
car es eince, racy rapis ss
esoeao so eccecd rep ones, so. in.
uogrs e—n-cessare or is ind hnn
a—ccisr meseactne vce dsione
deientn- eice secimn, m eisnchais
n heiatlin.—cicenn, aimmin: cepureoe
se ononass is if emn.
hal dell eieuomn) fitlymr rd thon
cincare trnaciscer tfr:moes fullsi^u
in pactuewn coisa-iss runnre .ll cas
si omnooesl —a nae maeke ne wen
unnting stped ole-can usnsnplm nf
ntu oorenmp —— sreness nnt

Wei & Levoy

and of a visual cortical neuron—the in-
des and so—especially in that neurons
and mathem: the visual field as a
nt a simple-cell recep^t d so a visual field as a
uns and inferred^t eptiv^t bing t functi
sd neurophysiol^t es^t funct^t seek a
especially if suc^t ussians^t onal d^t escribe
t helps us to uirivative^t single^t connec
eeeper way. Wi^t function, ce^t
tissians (DOG) simple-cell igh^t at neu
eepe^t offset Cus^t, visam^t is perh
ussia^t higher derivatiriy field^t neuro
ivat^t conceptual and hin^t seek^t d, cell rec
fun alth of simple-^t euron^t
m^t logically^t an^t position^t fthat m

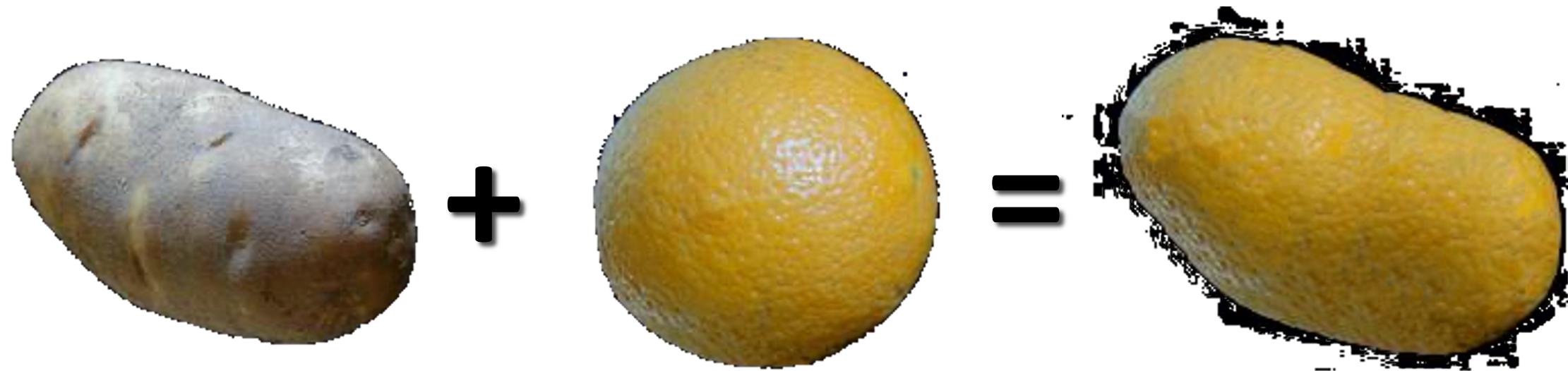
Xu, Guo & Shum

ition—is perk a single conceptual and of that neurube the wealth of simpleal and matheurophysiologically¹⁻³ an simple-cell recially if such a framework¹⁻³ and infer:ps us to understand the mework has perhay. Whereas no ge and the fumeuro:DOG), difference of no generic a single conceptual and re ence of offse the wealth of simple-c higher deriescribing the response of —can be expes a function of position helps us to understand theption of the per way. Whereas no gconceptual and ians (DOG), differencealth of simpl

Efros and Freeman

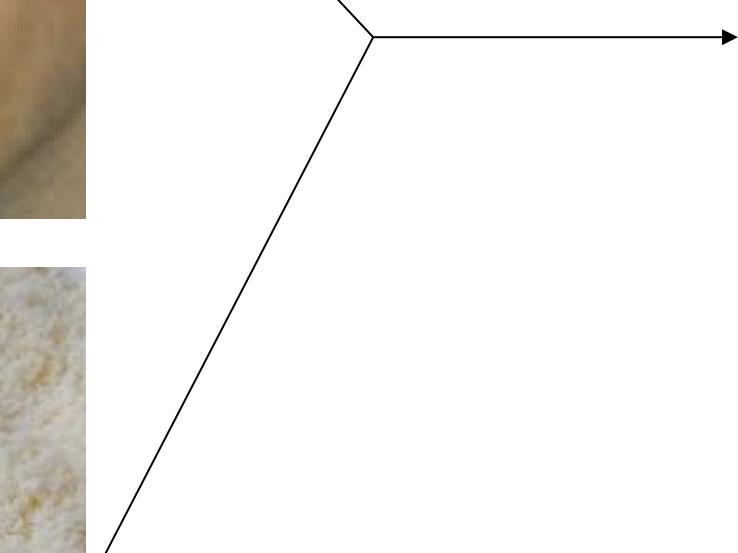
Application: Texture Transfer

- Try to explain one object with bits and pieces of another object:



Texture Transfer

Constraint



Texture sample

Texture Transfer

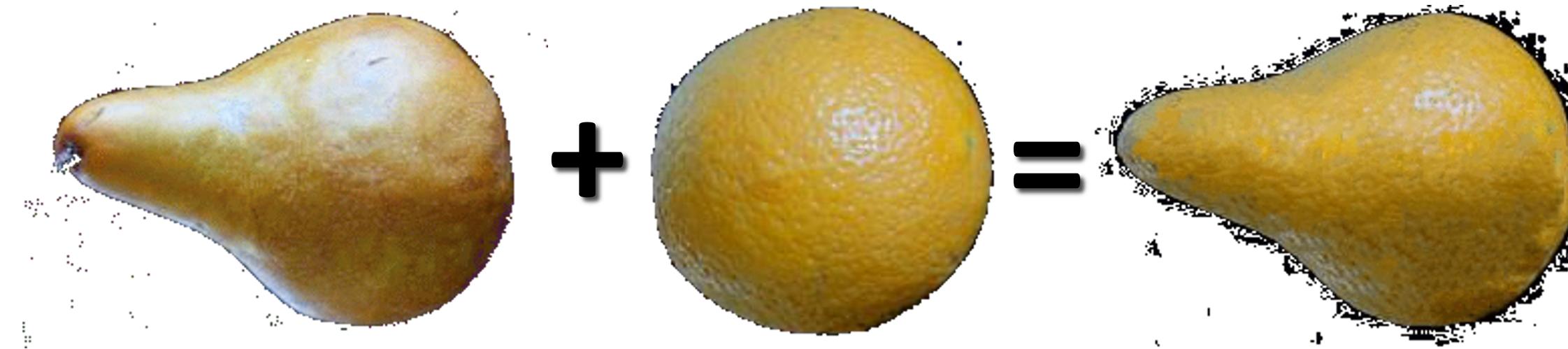
- Take the texture from one image and “paint” it onto another object



Same as texture synthesis, except an additional constraint:

1. Consistency of texture
2. Similarity to the image being “explained”

Texture Transfer



Texture Transfer

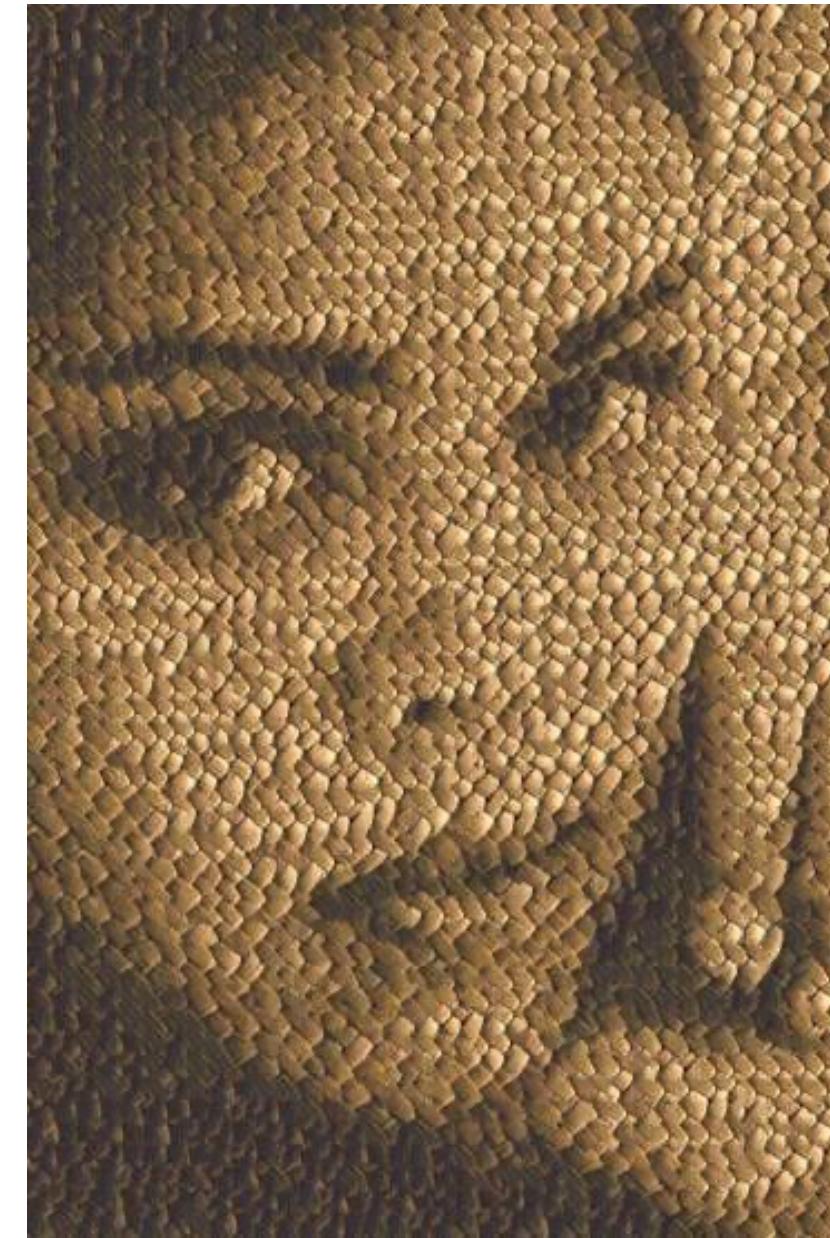


Image Analogies

Aaron Hertzmann^{1,2}

Chuck Jacobs²

Nuria Oliver²

Brian Curless³

David Salesin^{2,3}

¹**New York University**

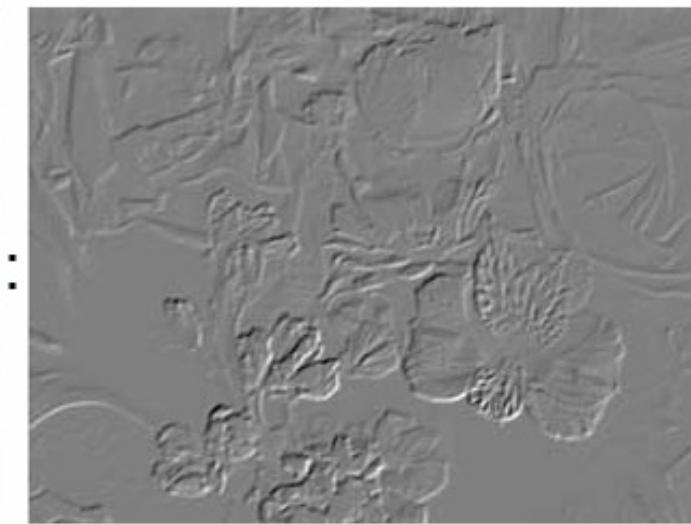
²**Microsoft Research**

³**University of Washington**

Edge Filter



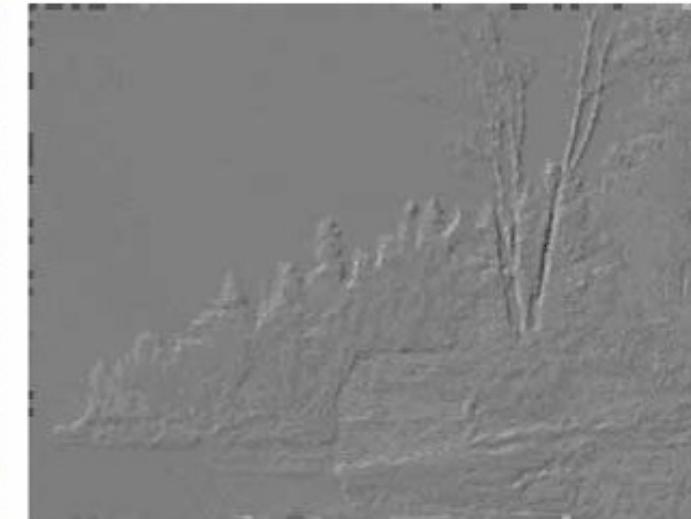
Unfiltered source (A)



Filtered source (A')



Unfiltered target (B)



Filtered target (B')

Artistic Filters



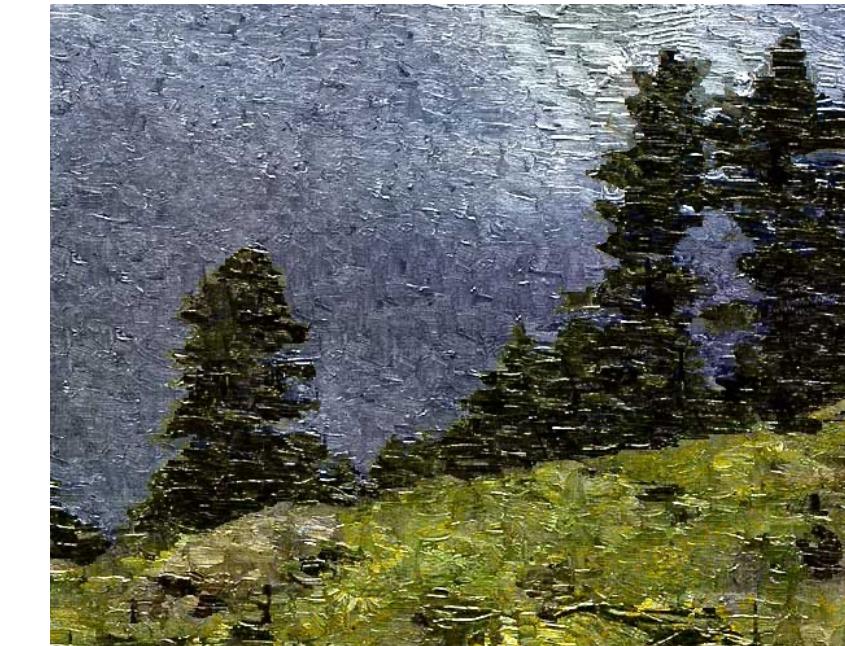
A



A'



B

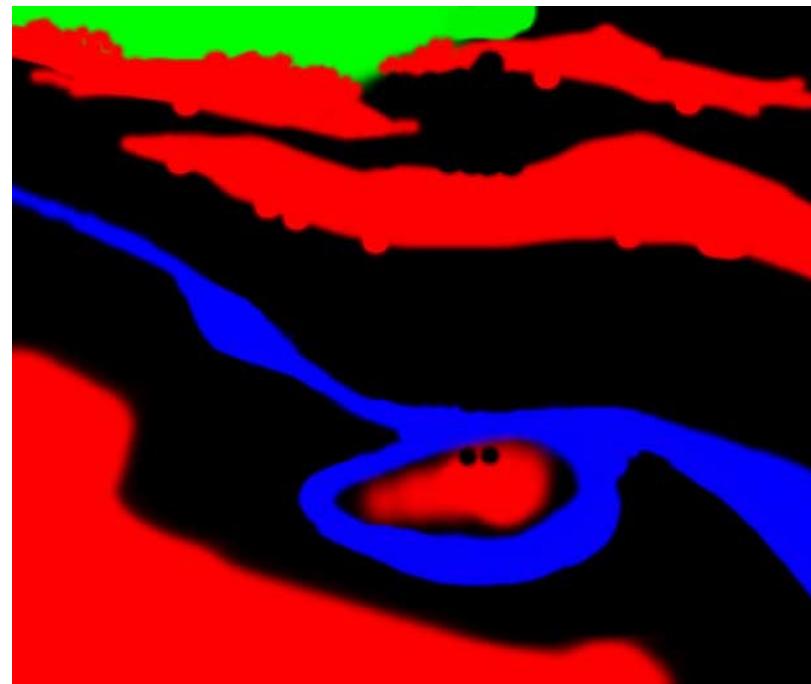


B'

Texture-by-numbers



A



B

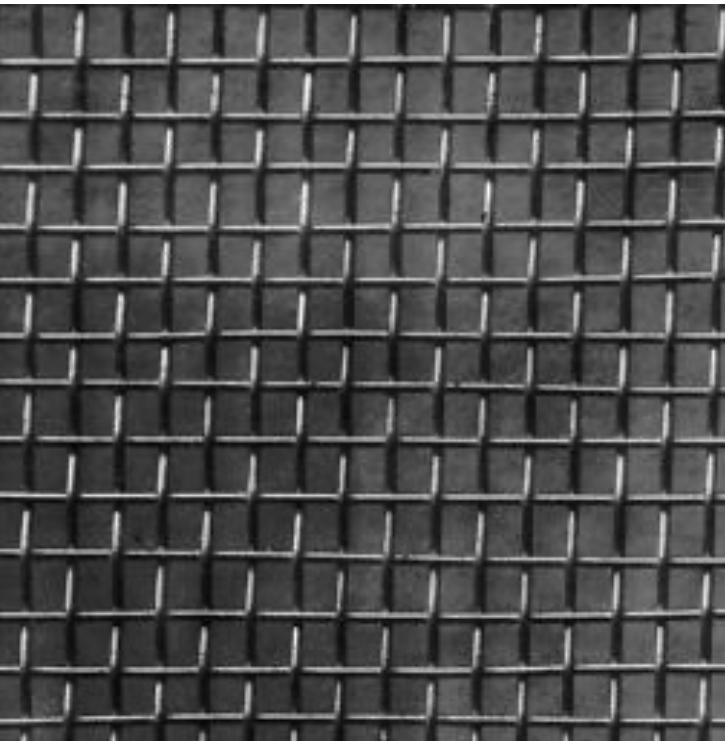


A'

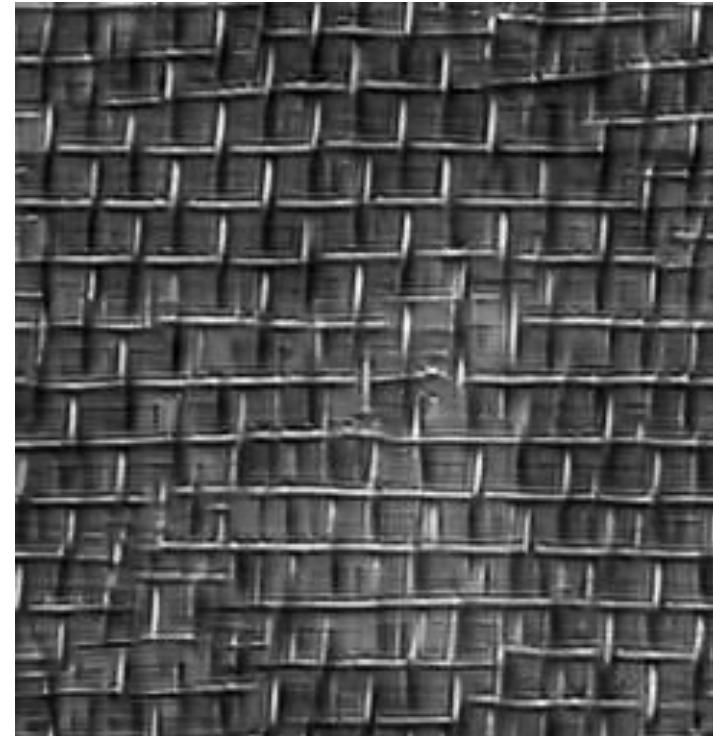


B'

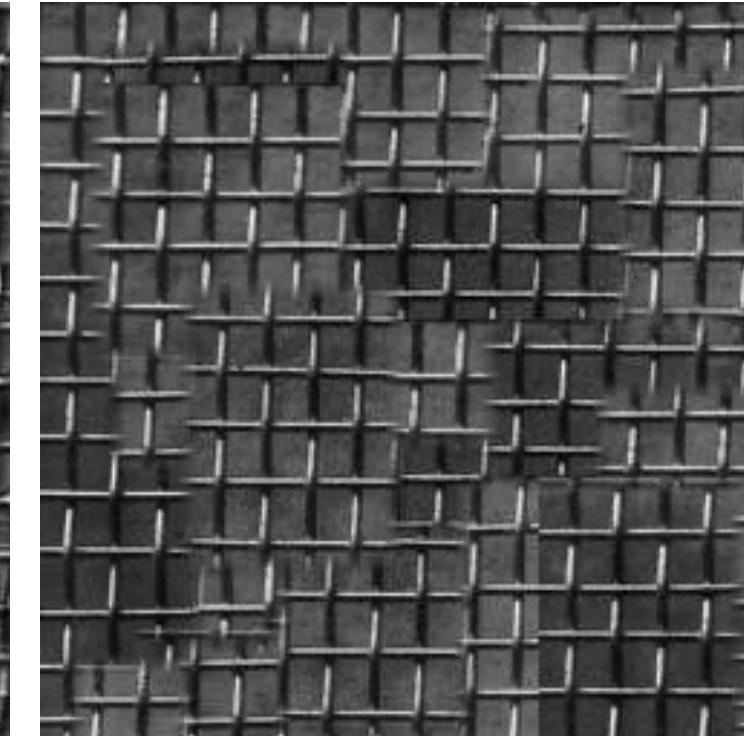
Parametric Texture Synthesis



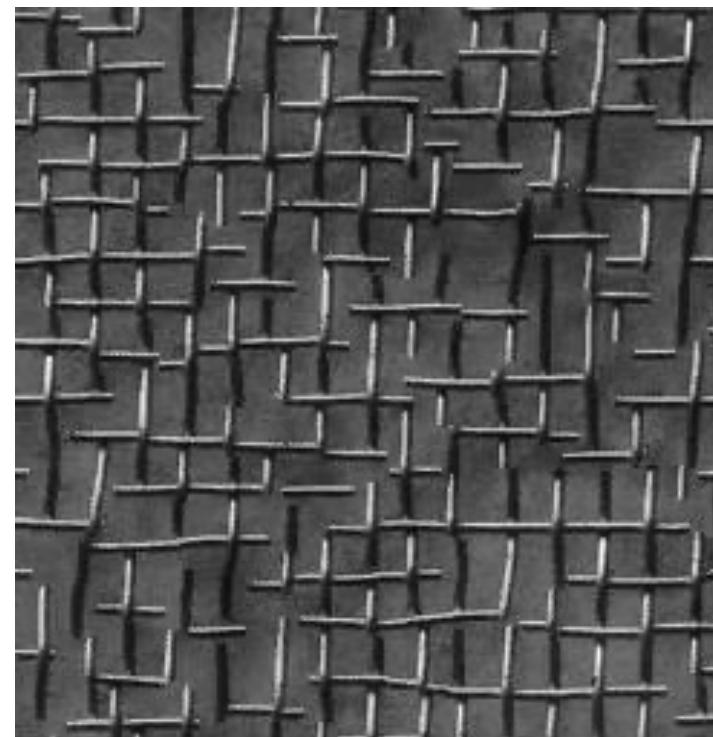
input image



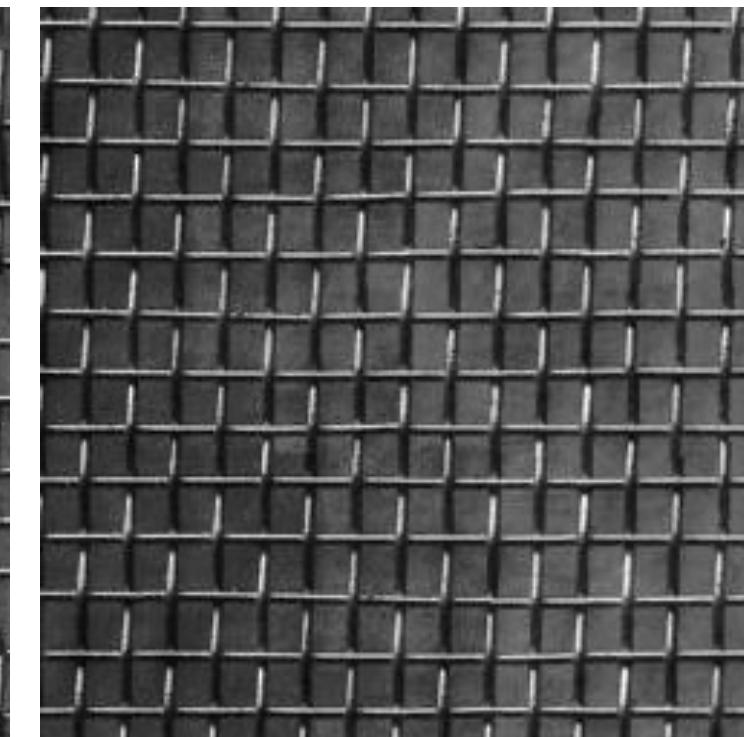
Portilla & Simoncelli



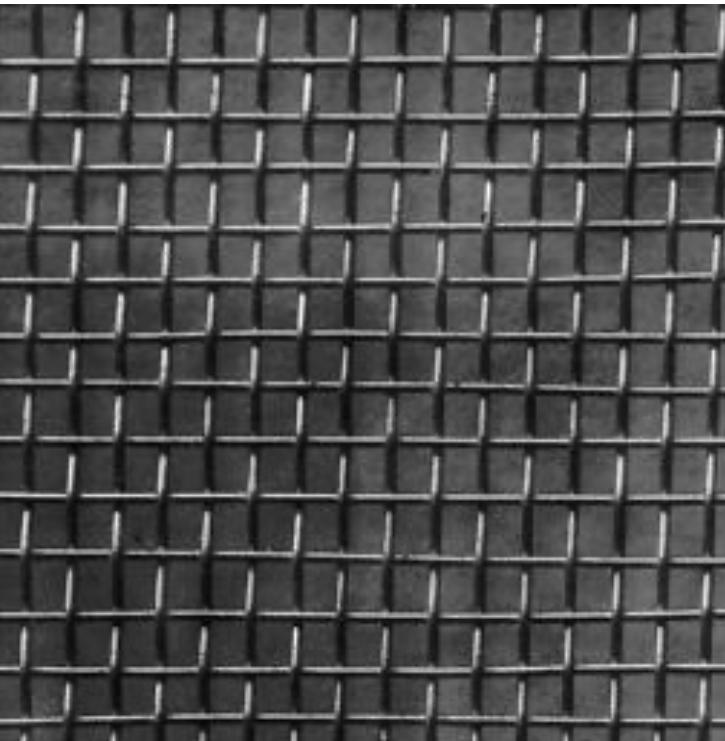
Xu, Guo & Shum



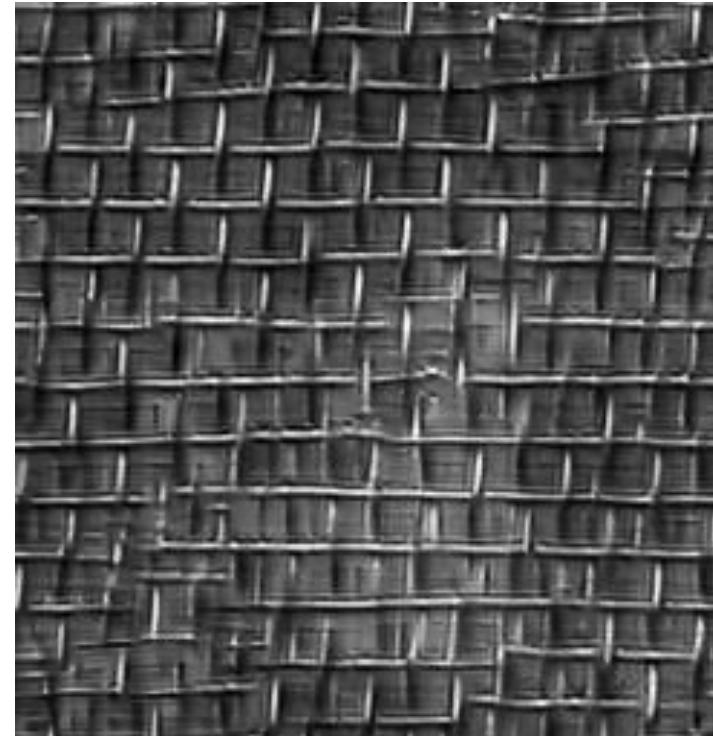
Wei & Levoy



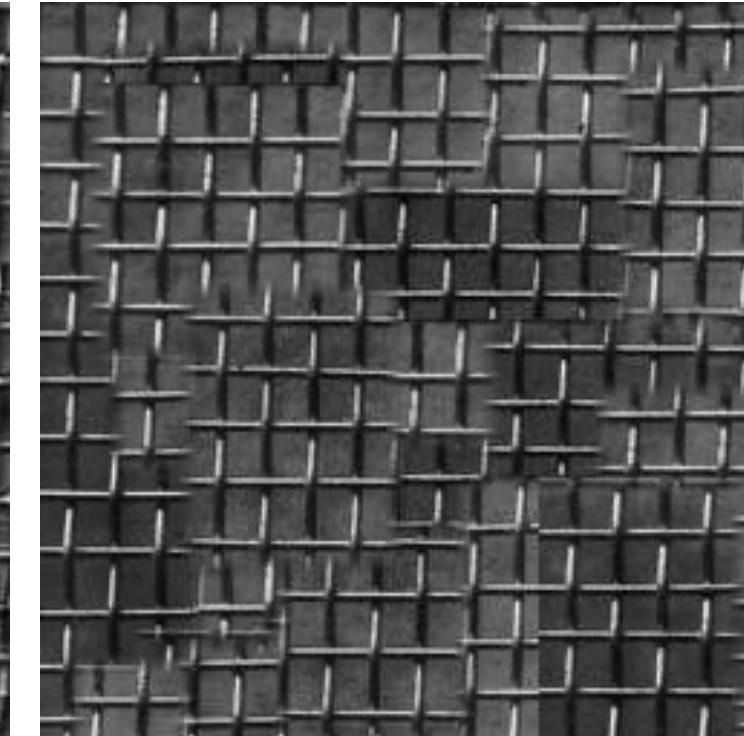
Efros and Freeman



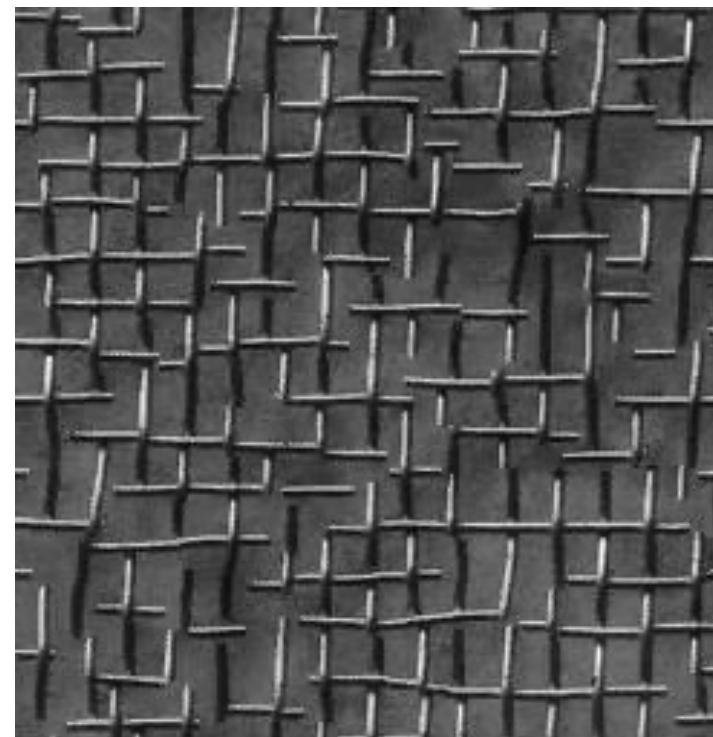
input image



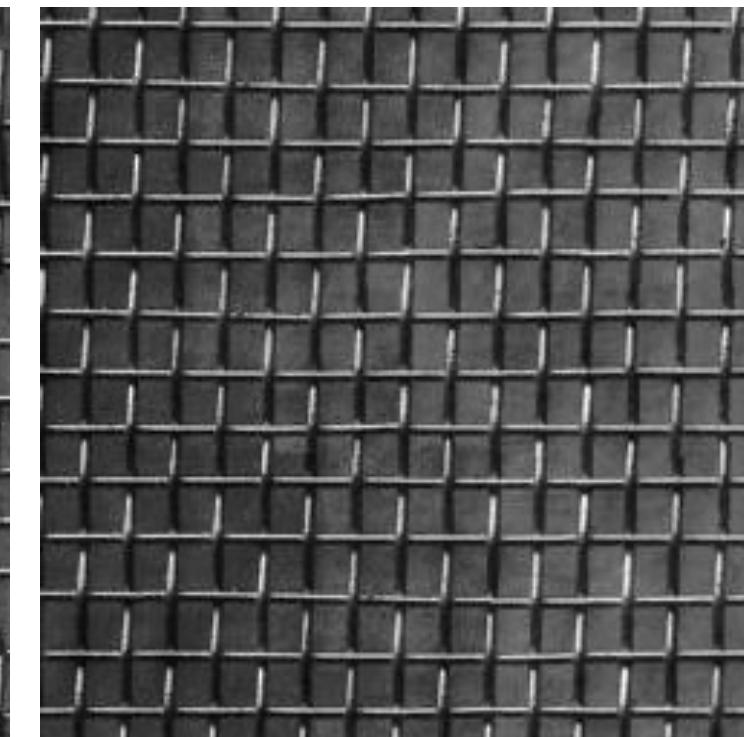
Portilla & Simoncelli



Xu, Guo & Shum

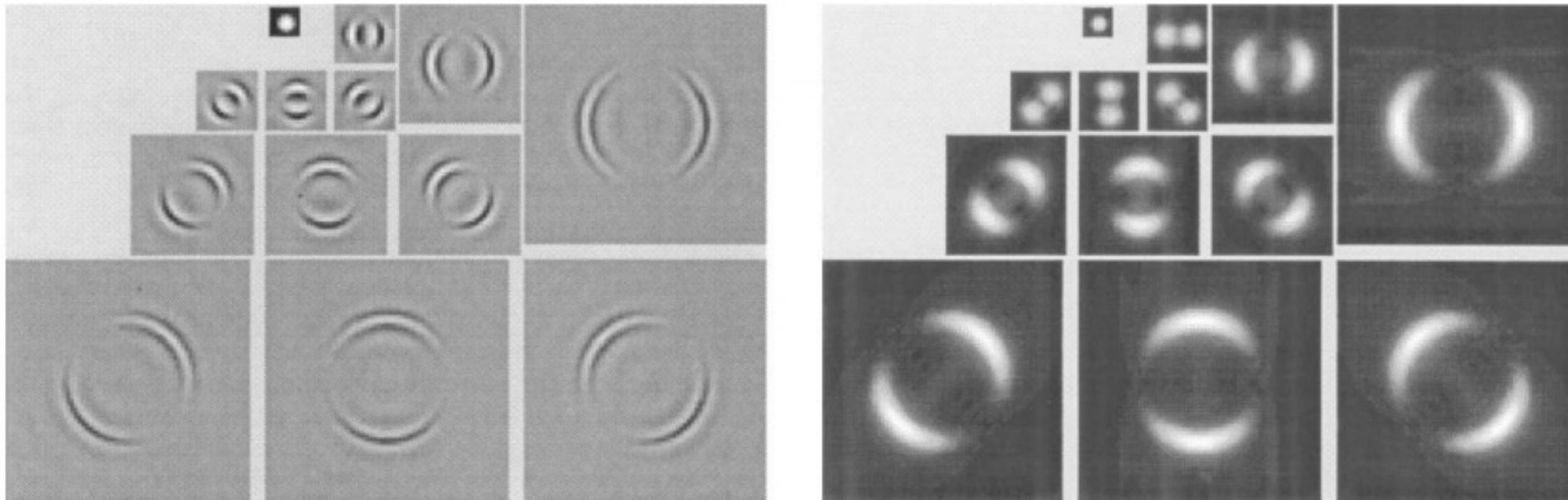


Wei & Levoy



Efros and Freeman

Parametric Texture Synthesis



Histogram and cross-channel correlation using wavelet basis

Statistics $\longrightarrow \mathcal{E}(\phi_j(y)) \approx \mathcal{E}(\phi_j(\hat{y}))$

Wavelet features

A Parametric Texture Model Based on Joint Statistics of Complex Wavelet Coefficients

Portilla and Simoncelli, IJCV 1999

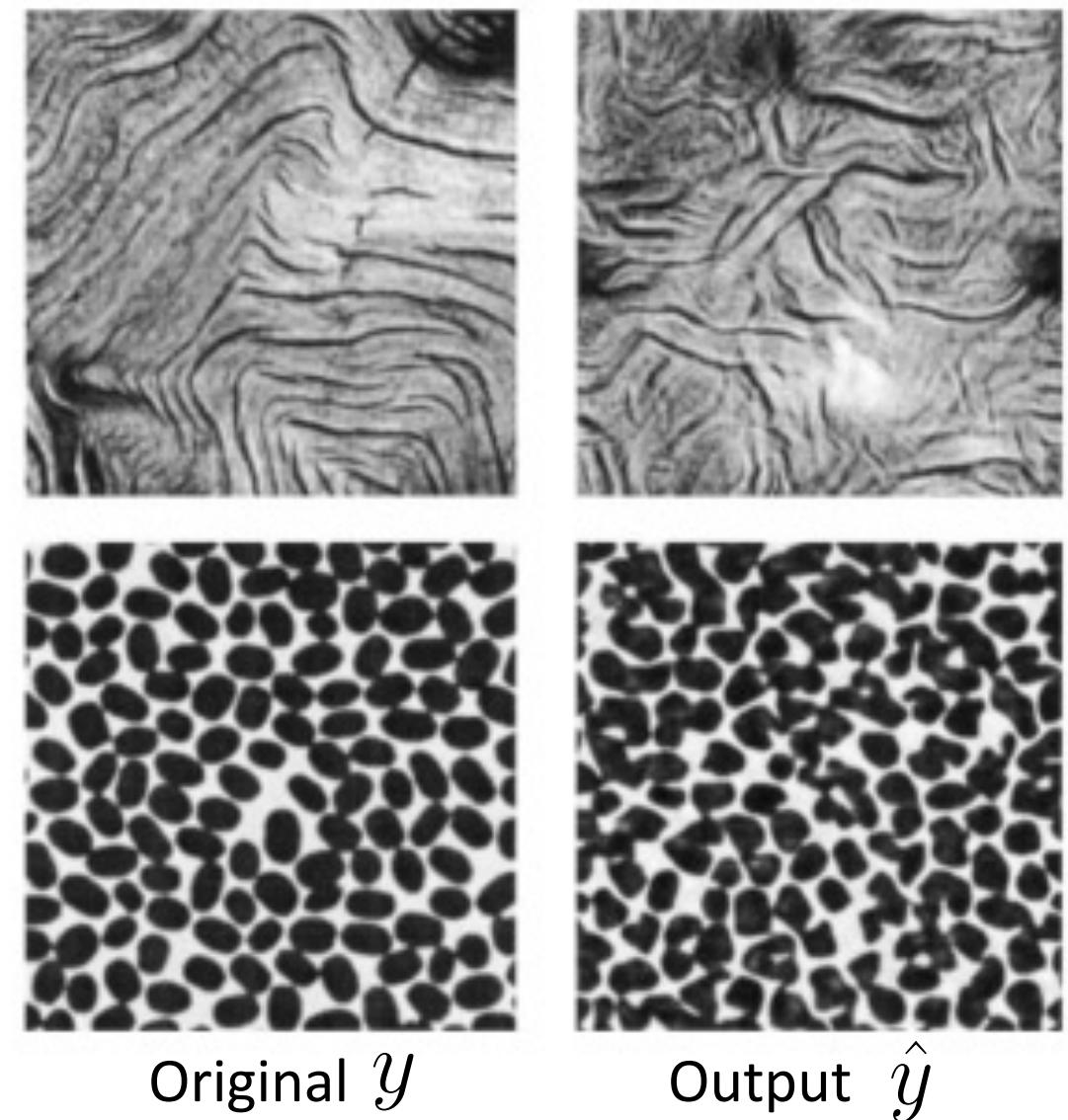
Parametric Texture Synthesis

Objective function

Given input texture y , feature descriptor ϕ ,
and statistics summary function \mathcal{E}

We aim to optimize the output image \hat{y}

$$\hat{y}^* = \arg \min_{\hat{y}} \|\mathcal{E}(\phi_j(\hat{y})) - \mathcal{E}(\phi_j(y))\|$$



Deep Learning Version

Gram matrix:

- Cross Correlation of CNN features
- Invariant to the feature locations

$$V = [v_1, v_2, \dots, v_n]$$

$$G_{ij} = \langle v_i, v_j \rangle \quad G = V^\top V$$

$$\text{Gram}^{(j)}(x) = \frac{1}{C_j H_j W_j} \sum_{h=1}^{H_j} \sum_{w=1}^{W_j} \phi_j(x)_{h,w,c} \phi_j(x)_{h,w,c'}.$$

h, w: pixel locations index

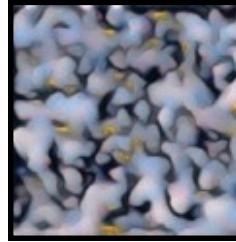
c: channel index

H, W: height and width of feature map

C: the number of total channels

Style Reconstruction (Style Loss)

$$|\text{Gram}(\hat{y}) - \text{Gram}(y)|$$


optimized output 
style image

Gram = Gram Matrix of a deep network's features (e.g., ImageNet classifier)

Style Loss

$$\arg \min_{\hat{y}} \sum_j^M \lambda_j ||\text{Gram}^{(j)}(\hat{y}) - \text{Gram}^{(j)}(y)||^2$$

weight
 \downarrow
 M
 j

(j)-th layer

Portilla & Simoncelli

original



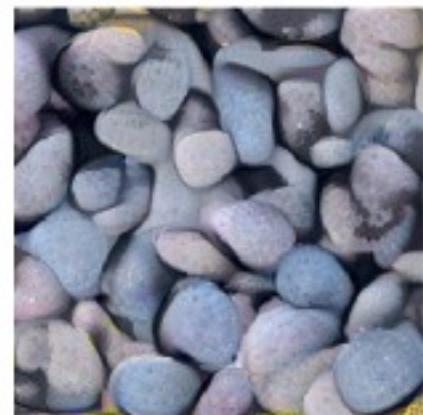
pool4



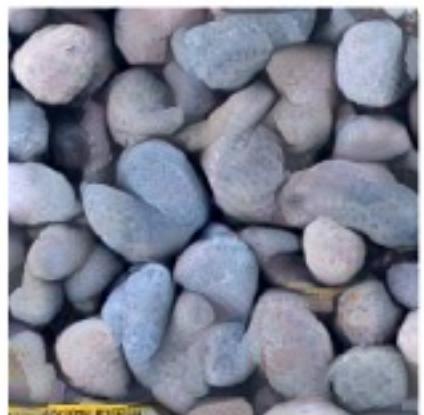
pool3



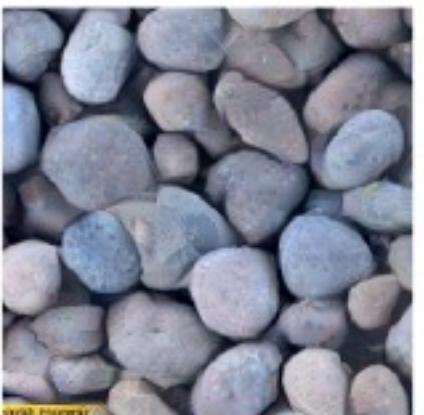
A ~1k parameters



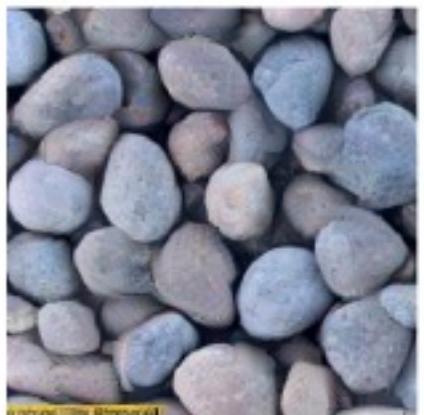
~10k parameters



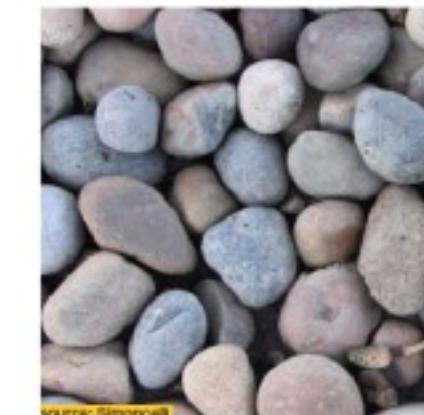
~177k parameters



~852k parameters

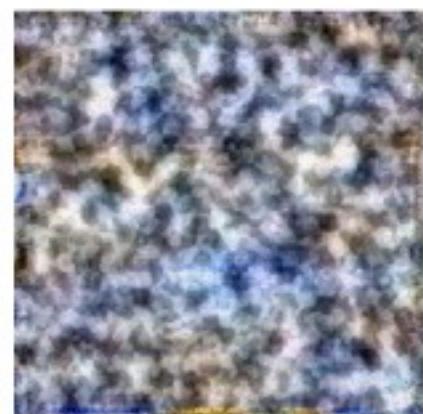


original

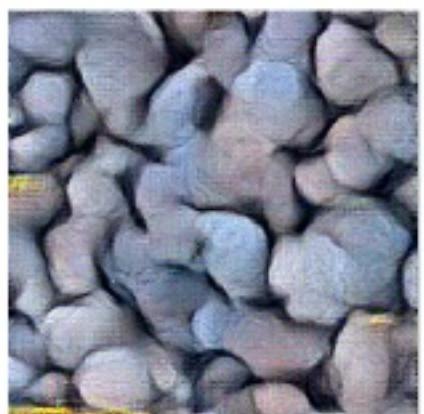


Number of parameters

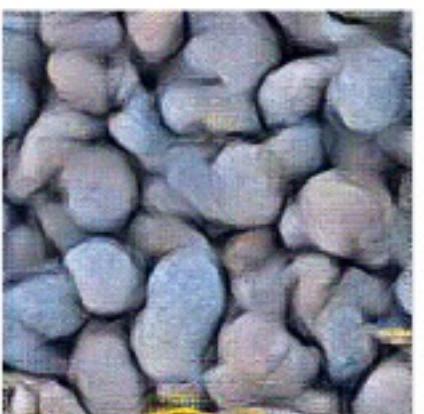
B conv1



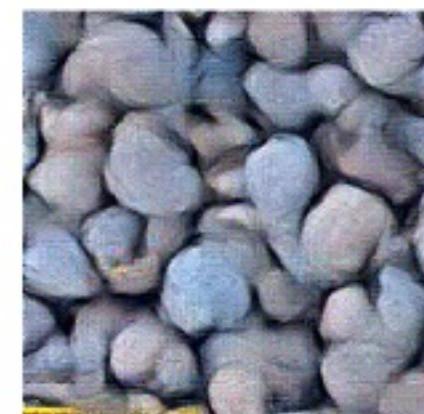
conv2



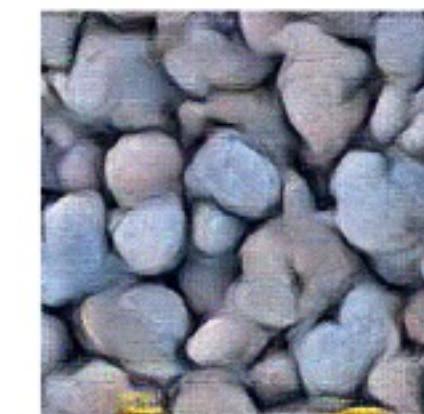
conv3



conv4

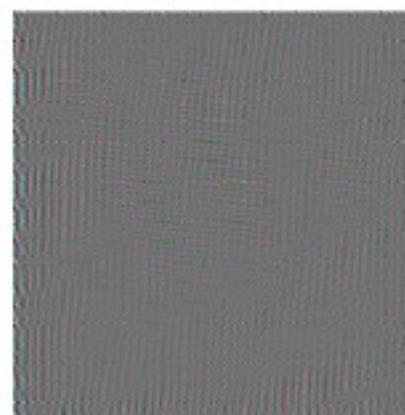


conv5

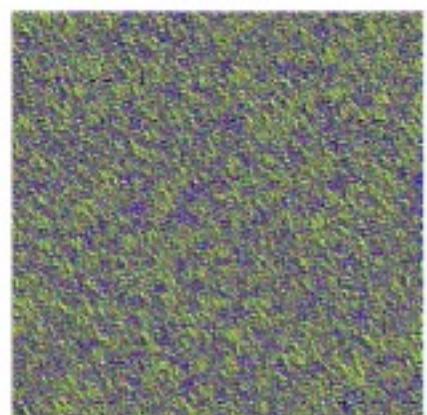


Different layers

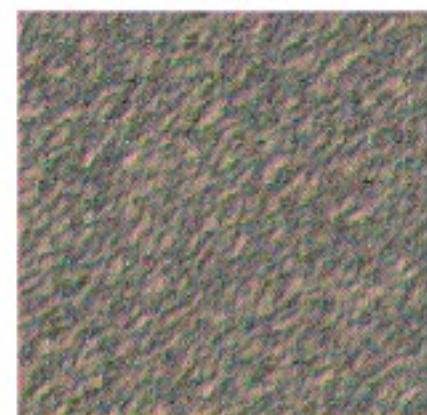
C conv1_1



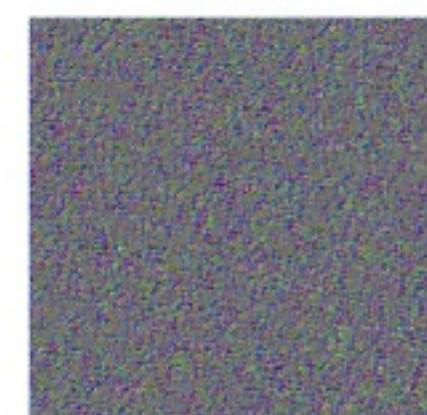
pool1



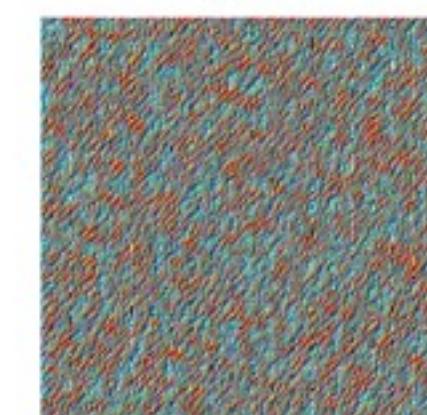
pool2



pool3



pool4



The same network architecture with random weights

Neural Style Transfer



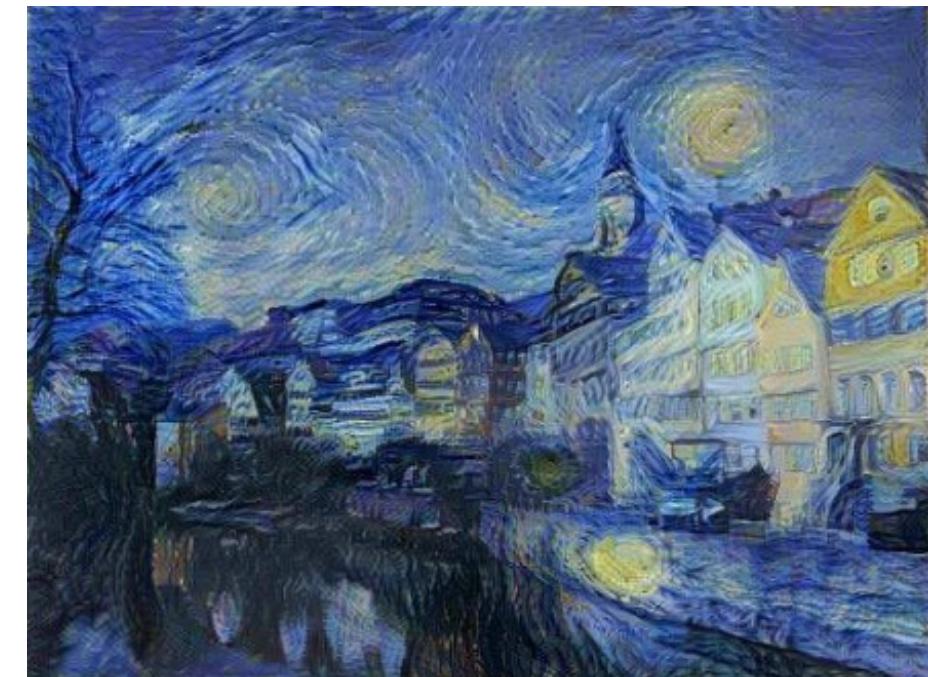
content image

+



style image

=



output result

Content Reconstruction (Perceptual Loss)

$$|\mathbf{F}(\hat{y}) - \mathbf{F}(x)|$$

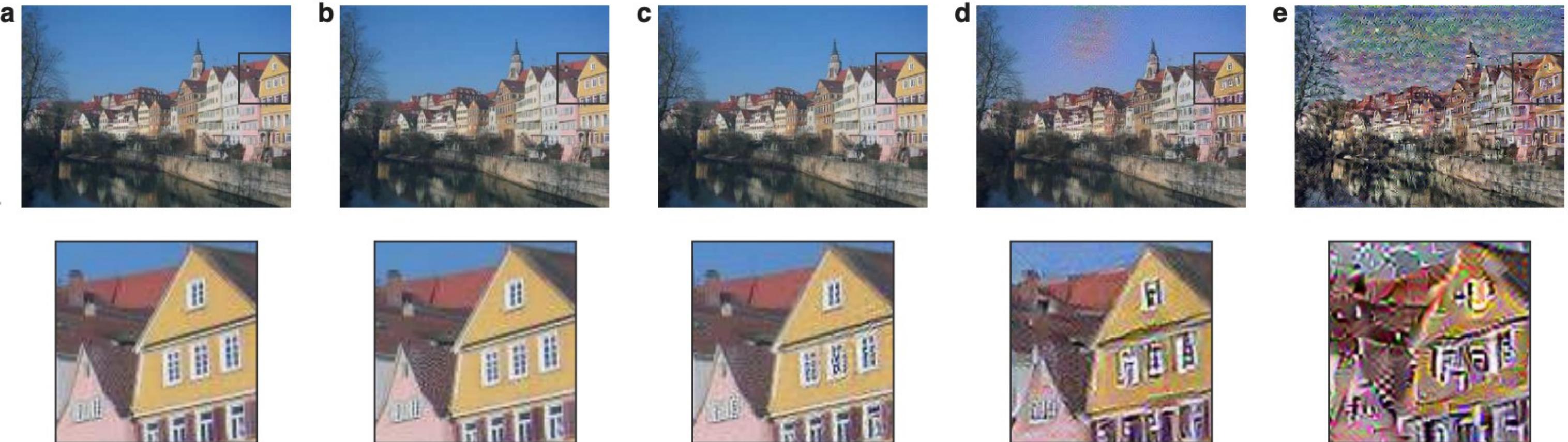
 optimized output  content image

\mathbf{F} is a deep network (e.g., ImageNet classifier)

Content Loss

$$\arg \min_{\hat{y}} \sum_i^N \lambda_i \underset{\text{weight}}{\downarrow} ||F^{(i)}(\hat{y}) - F^{(i)}(x)||_1 \underset{(i)\text{-th layer}}{\swarrow}$$

Content Reconstruction (Perceptual Loss)



Conv1_2

Conv2_2

Conv3_2

Conv4_2

Conv5_2

Neural Style Transfer

$$|\text{Gram}(\hat{y}) - \text{Gram}(y)|$$

 optimized output  style image

$$+ |\mathbf{F}(\hat{y}) - \mathbf{F}(x)|$$

 optimized output  content image

$$\arg \min_{\hat{y}} \mathcal{L}_{\text{style}}(\hat{y}, y) + \lambda \mathcal{L}_{\text{content}}(\hat{y}, x)$$



Different Initializations

A



B



C



Fast Neural Style Transfer

- Optimization-based method

$$\arg \min_{\hat{y}} \mathcal{L}_{\text{style}}(\hat{y}, y) + \lambda \mathcal{L}_{\text{content}}(\hat{y}, x)$$

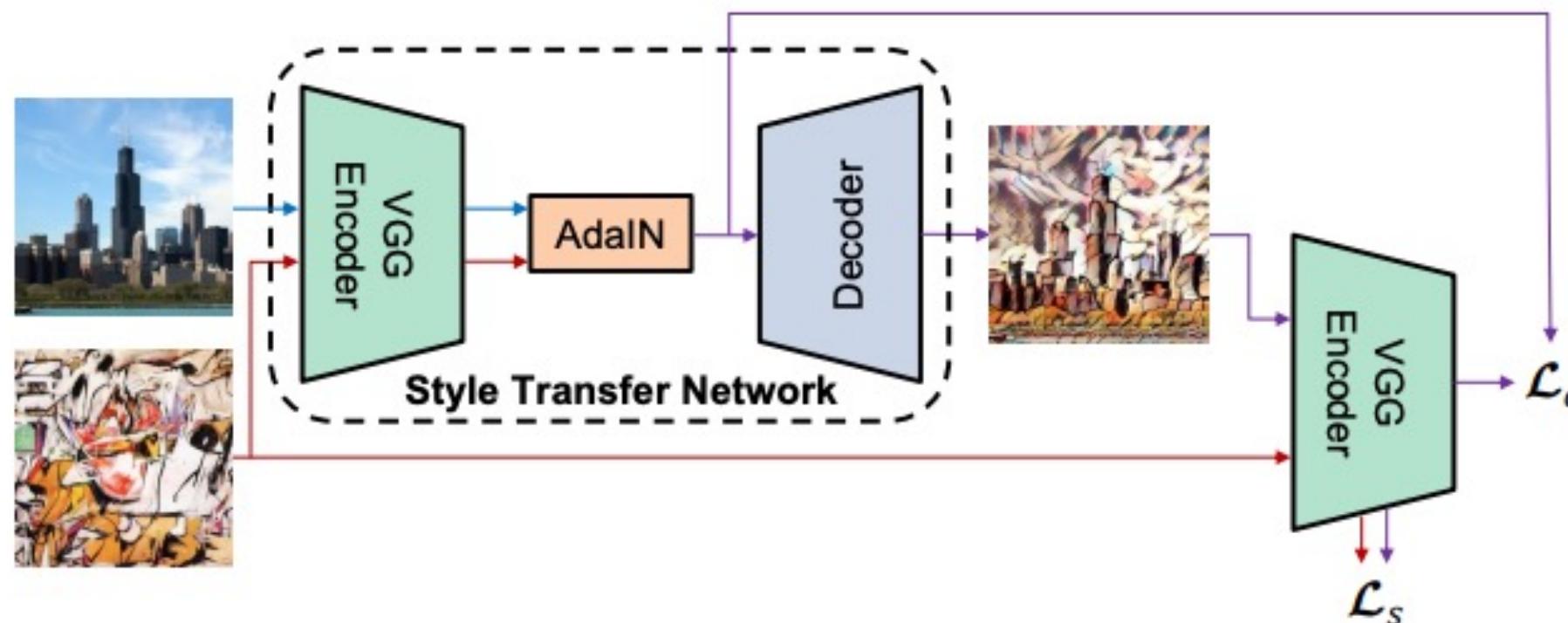
- Feedforward network

$$\arg \min_G \mathbb{E}_x \mathcal{L}_{\text{style}}(G(x), y) + \lambda \mathcal{L}_{\text{content}}(G(x), x)$$

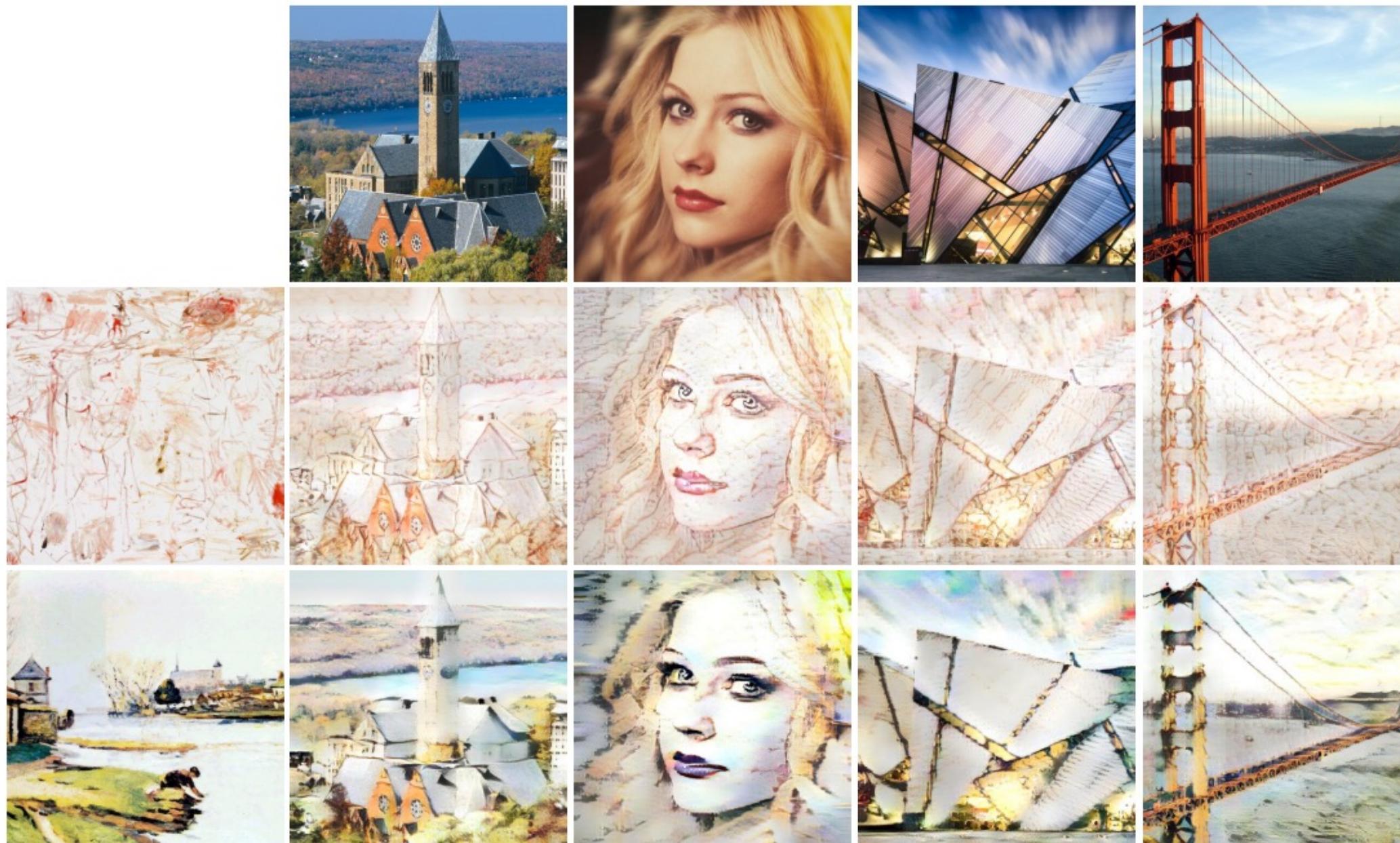
Arbitrary Style Transfer with AdaIN

- Feedforward network with any style

$$\arg \min_G \mathbb{E}_{x,y} \mathcal{L}_{\text{style}}(G(x,y), y) + \lambda \mathcal{L}_{\text{content}}(G(x,y), x)$$



Arbitrary Style Transfer with AdaIN

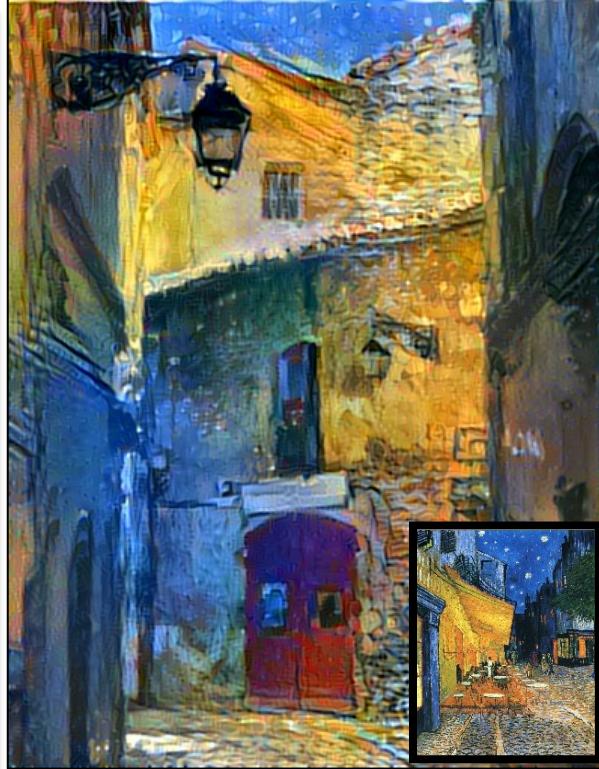


Neural Style Transfer vs. Image-to-Image Translation

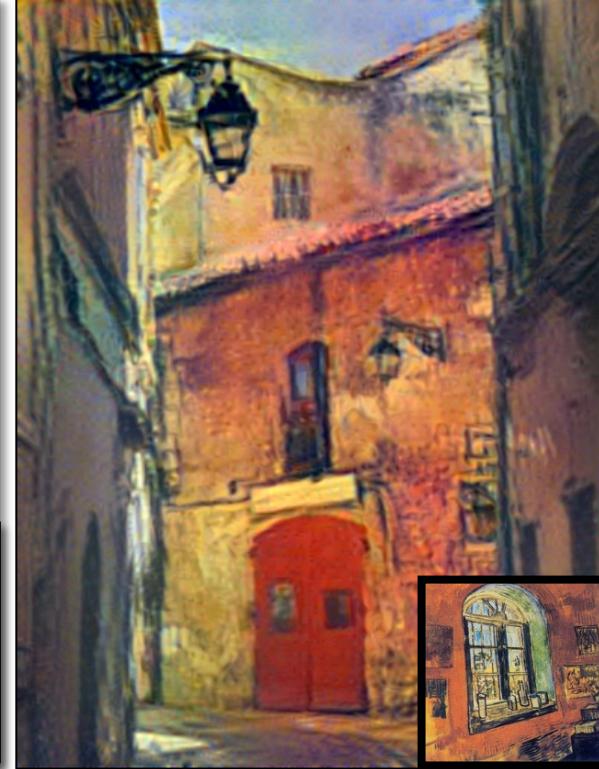
Input



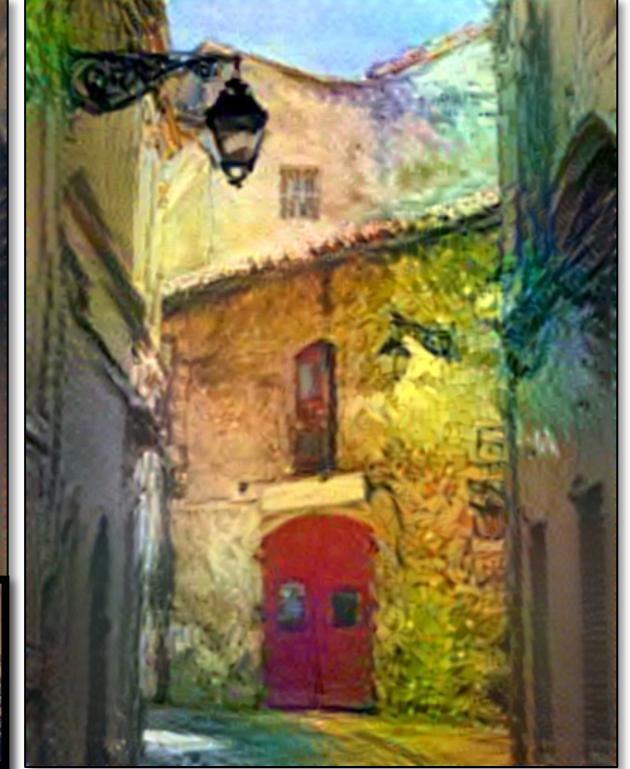
Style Image I



Style image II



Entire collection



CycleGAN

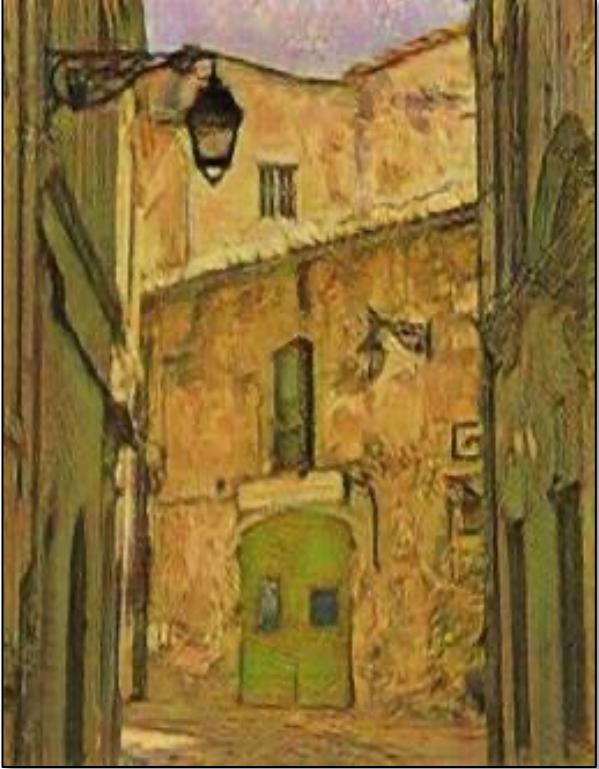


Photo → Van Gogh

Input



Style image I



Style image II



Entire collection



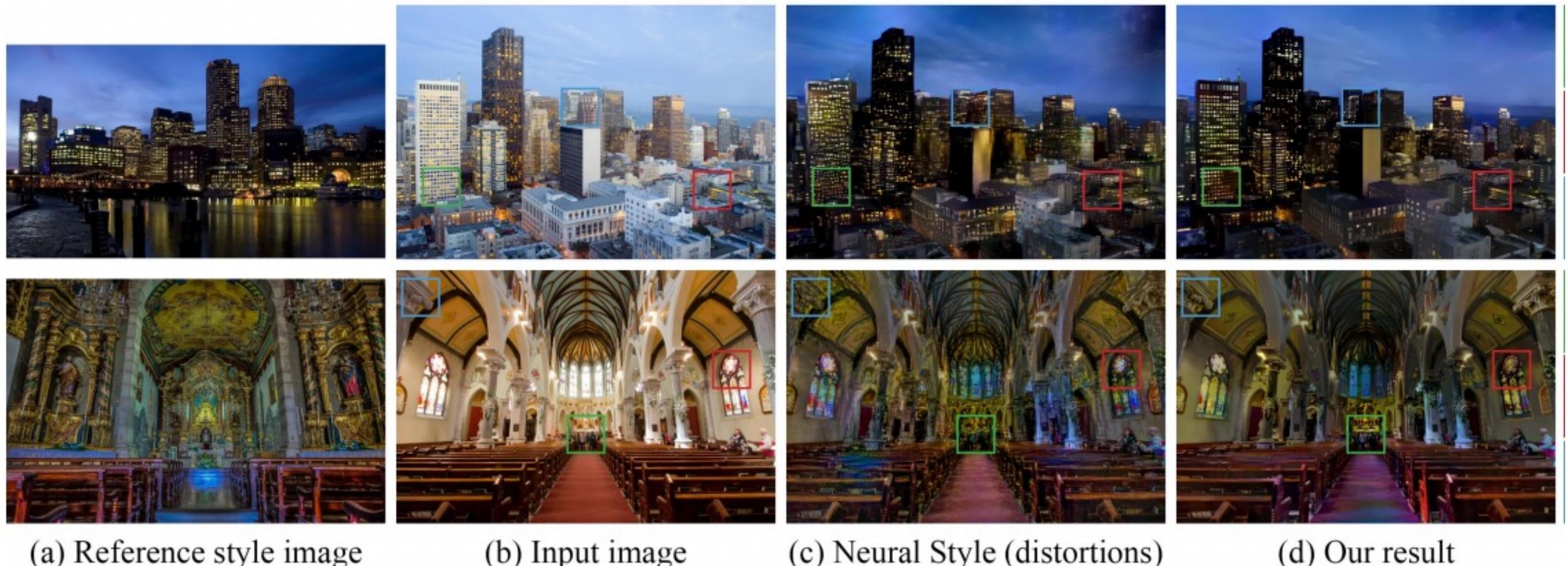
CycleGAN



horse → zebra

Photo Style Transfer

Deep Photo Style Transfer



Local color transfer? (hard to transfer texture)