Split and Match: Example-based Adaptive Patch Sampling for Unsupervised Style Transfer

1. Abstract

Transferring the style from one image onto another can be considered a problem of style transfer. In texture transfer synthesize a texture from a source image while constraining the texture synthesis in order to preserve the semantic content of a target image. The major limiting factor for previous approaches has been the lack of image representations that explicitly represent semantic information and, thus, allow to separate image content from style. Here we presents a unsupervised method for style transfer from an example image to input image. To preserve the structure and capture the style of an example image, propose a “Split and Match” based adaptive patch partition approach.

2. Introduction

Style transfer is the task of transforming an image in such a way that it mimics the style of a given example. This type of applications are of special interest in film post-production, graphics and photography. The difficulty of this task is bound to the complexity of defining the style as a composition of different visual attributes such as color, shading, texture, lines, strokes and regions. Most of previous style and texture transfer algorithms rely on non-parametric methods like sampling pixels directly from an example texture for texture synthesis while using different ways to preserve the structure of the target image but they are not effective.

In this propose a new method to overcome the structure preserving, we are given as an example image containing a mixture of style and content. Based on this example image, will predict how well a source patch matches an example match. Patch dimensions should be large enough to represent the patterns that characterize the example style, while small enough to forbid the synthesis of content structures present in the example image.

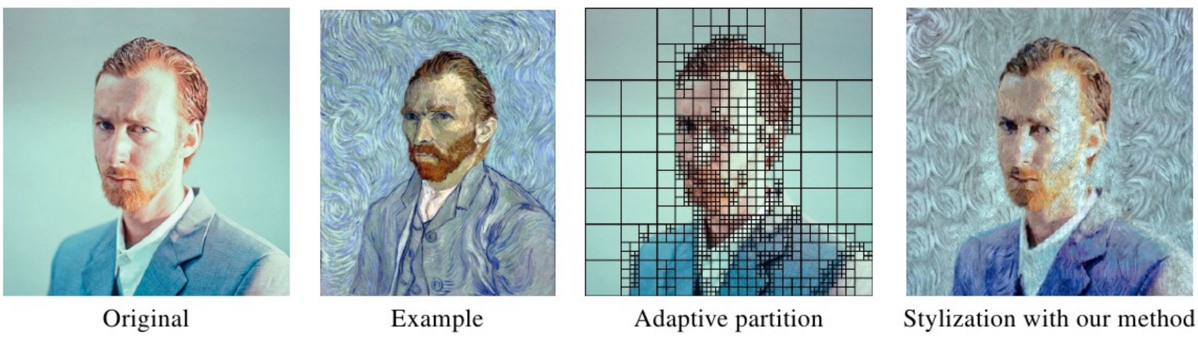


Figure 1. Illustration of the proposed unsupervised style transfer

3.Split and Match Style Transfer

Problem Definition

Let u : $\mathbf{\Omega_u} \rightarrow \mathbf{R^3}$ be an input image and $\mathbf{v : \Omega_v} \rightarrow \mathbf{R^3}$ an example image. The correspondence map between input image and example image is $\boldsymbol{\varphi} : \mathbf{\Omega_u} \rightarrow \mathbf{\Omega_v}$. It maps the each point $\mathbf{x} \in \mathbf{\Omega_u}$ in an input image to correspondence point $\boldsymbol{\varphi}(\mathbf{x}) \in \mathbf{\Omega_v}$ in an example image. And the target image can be defined as $\mathbf{\hat{u} = v(\varphi)}$.

The input image would be divided into partitions $\mathbf{R_n} \in \mathbf{\Omega\ u}$ based on correspondence map $\boldsymbol{\varphi}$, while capturing the style of example image **u** and capturing the semantic structure of input image **v**. To achieve a convincing style transfer, regularity is also required at the boundary between neighboring correspondent regions.

This split and match based unsupervised style transfer divided into sub-problems independently, following are the four sub-problems below:

- **Split and match:** Compute the adaptive partition R for source image **u**. This adaptive partition will depends on the depends local variance of a partition and the patch similarity between the input and example images.

fig 2. Source Image

fig 2. Example Image

- **Optimization:** By considering this as a graph labeling problem, search of the optimal map $\boldsymbol{\varphi}$ between source patch and example patch, where the nodes of the graph are the regions $\mathbf{R_i}$. Set of K candidate labels for the region $\mathbf{R_i}$, the label belongs to $\mathbf{\Omega_v}$ being a patch coordinate in image v. This probabilistic labeling problem is solved by belief propagation.

Fig 4. After texture transfer on input image

- **Bilinear blending** between neighboring regions and reconstruction of **û**. This blending strategy ensures smooth transitions between neighbor patches at a very low computational cost.

- **Global color and contrast matching:** Applying a global color transfer method in chrominance channel to capture the color style, and a contrast transformation to match the global contrast of the example image.

Fig 5. After texture + color transfer on input image

3.1 Split and Match adaptive partition

Decomposing an input image into a suitable partition based on local variance of quadtree cell and the similarity between input and example images.

- **Pseudo Code :**

  Input Images: u,v**;** parameters**:Y0,Y1,ω**
  Output: Set of regions R={Ri}ni=1, set of candidate labels L={Li}ni=1

```
Initialization:R1←{Ωu}
for every region Ri∈R do
    xi←center of Ri
    σi←√Var(puxi)
    Compute yi= arg min y d[puxi,pvy]
    ifζ(puxi,pvyi)is true then
        Split Ri into four:
        m←]R−1
        R←{R\Ri}∪{Rm+1,...,Rm+4}
    else
        Compute spatially constrained k-NN:
        Li←{lik}Kk=1 with|lik−lik+1|> χ
    end if
end for
```

$\zeta(p_{uxi},p_{vyi}) = (\sigma_i+d[p_{uxi},p_{vyi}] > \omega \text{ and } \tau_i>Y0) \text{ or } \tau_i>Y1$
$d[p_{uxi},p_{vy}] = ||p_{uxi}-p_{vy}||^2 / \tau^2_i$

Explanation :

'u' is our input image.
'v' is our output image.

puxi – Patch in image 'u' with center at 'xi'
pvy – Patch in image 'v' with center at 'y'

Parameters:

Y0 – min size of a patch, beyond which splitting stops. (8^2)
Y1 – max size of a patch in which case a split is definite. (256^2)
ω - is a similarity threshold (fixed to ω:= 15 in practice)
R – set of patches in the image 'u' at an instant.
(Starts with one patch, which is the whole image)
Li – The label which is the closest match to a patch Ri.

Algorithm :

1. Initialize set R with image 'u'.
2. Find a patch in 'v' which is the closest (using distance metric on luminance channel of YUV), to Ri. For finding best patch in 'v' we traverse the entire image with all possibilities of patches having same dimension (Ti). An optimization here would be to consider only the patches in 'v' using a stride. Using stride 2 would result in 4 times lesser candidate patches.

3. Check if the condition to split is met or not -
- If size is greater than Y1 split.
- If size is lesser than Y0 don't split.
- If within these limits and the distance between the Luminance channel of the 'u' and 'v' images is greater than (15 (omega) – standard deviation of patch 'i').

4. If split is done, replace Ri in the set R, with the resulting 4 square patches.
5. If split condition is not met, we find k Nearest Neighbors which are candidates labels (Lik) for the patch Ri.
'i' indicates the patch number.
'k' indicates the kth candidate label for patch 'i'.
The centers of these 'k' neighbors are constrained to not be closer than Ti/2.
$|l_{ik}-l_{ik+1}| > \chi$
This is done so as to induce variety of choice within the labels.
6. We loop back to step 2 and pick a new element from the set R, Ri.

3.2 Markov Random Fields modeling

Markov Random Fields (MRF) are a well known inference model for computer vision problems, widely used to model texture synthesis and texture transfer. Within this framework, the problem of example-based style transfer can be solved by computing the Maximum a Posteriori from a well chosen joint probability distribution on all quadtree patch labels.

For a quadtree patch $\mathbf{p}^{u}_{xi}$, we first compute a set of K candidate labels $\mathbf{L}_i = \{\mathbf{l}_{ik}\}^{K}_{k=1}$ as a strategy to reduce the dimensionality of the labeling problem. We consider now an inference model to compute the most likely set of label assignments for all the patches in R, where labels represent patch correspondences between **u** and **v**.

More precisely, we search for the set of label assignments $\hat{L} = \{ \hat{l}_i \}^{n}_{i=1}$ maximizing the probability density

$$\mathbf{P(L)=1/Z\ \pi_i\, \varphi(l_i)\ \ \pi_{(i,j)\in N}\, \psi(l_i , l_j)}$$

where Z is normalization constant, φ is the data fidelity term

$$\boldsymbol{\varphi(l_i) = \exp(-d[p^{u}_{xi} , p^{v}_{li}]\lambda_d)}$$

and $\psi(l_i , l_j)$ is a pairwise compatibility term between neighboring nodes i and j ((i, j) $\in$ N means that R_i and R_j are neighbors in Ω_u)

$$\boldsymbol{\psi(l_i , l_j)= \exp(-d[\tilde{p}^{v}_{li} , \tilde{p}^{v}_{lj}]\lambda_s + |l_i - l_j |\, 2\, \lambda_r)}$$

This function ψ is composed of a smoothness term and a term penalizing label repetitions.

In patch-based MRFs, the compatibility term ensures that neighbor candidate patches are similar in their overlapping region. To define this properly, we first extend each region R_i of the partition R by $\tau_i\theta$ in each direction (θ = 0.5). This permits to define an overlap between two neighboring extended regions $\tilde{\mathbf{R}}_i$ and $\tilde{\mathbf{R}}_j$. The term $d[\tilde{p}^{v}_{li} , \tilde{p}^{v}_{lj}]$ in $\psi(l_i , l_j)$ is the distance between the corresponding extended patches in v over this $\tilde{\mathbf{R}}_i \cap \tilde{\mathbf{R}}_j$.

While we search for smooth intensity transitions in the overlapping part of neighbor candidate patches, we also aim to penalize two neighbor nodes to have exactly the same label, thus we encourage $|\mathbf{l_i} - \mathbf{l_j}|^2$ to be large as a strategy to boost local synthesis variety.

Note that computing an exact MAP inference to solve directly is an intractable combinatorial problem due to the high dimensionality of image based graphical models, but approximate solutions can be found by iterative algorithms.

We adopted the Loopy Belief Propagation method, Basically, neighboring variables update their likelihoods by message passing and usually after a small number of iterations, the approximate marginal probabilities (beliefs) of all the variables in a MRF are computed.

Finally, a patch in the reconstructed image û with estimated label $l_i^{\hat{}}$ is given by $p^{\hat{u}}_{xi} = p^{v}_{l^{\hat{}}}$.

3.3. Bilinear blending

Although we compute label correspondences that are likely to be coherent across overlapping regions, seams can still be noted in the reconstructed image û across the quad-tree patch boundaries.
Bilinear blending between neighboring regions and reconstruction of û. In order to remove visible seams we have to apply an effective method inspired on linear alpha blending.

$$\tilde{u}(x) = \sum_{s=1}^{S} \alpha_s(x)\, \tilde{p}^{\hat{u}}_{x_s}(x)\,, \text{ where } \alpha_s(x) = \frac{\delta(x, \partial\tilde{p}^{\hat{u}}_{x_s})}{\sum_{s=1}^{S} \delta(x, \partial\tilde{p}^{\hat{u}}_{x_s})} \tag{7}$$

is the weighting factor and $\delta(x, \partial\tilde{p}^{\hat{u}}_{x_s})$ is the normalized closest distance between pixel x and the patch border $\partial\tilde{p}^{\hat{u}}_{x_s}$:

$$\delta(x, \partial\tilde{p}^{\hat{u}}_{x_s}) = \frac{|x - \partial\tilde{p}^{\hat{u}}_{x_s}|^2}{\tau_s^2}. \tag{8}$$

This blending strategy ensures smooth transitions between neighbor patches at a very low computational cost

3.4. Global color and contrast transfer

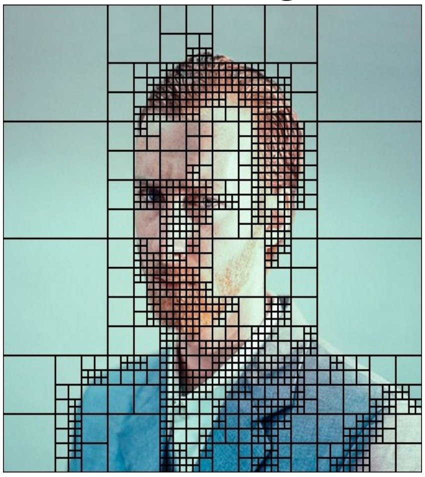
Now, we consider that color and contrast are two features in style that may be consistently modeled as global transformations. Applying a global color transfer method in chrominance channel to capture the color style, and a contrast transformation to match the global contrast of the example image.

The color transfer is formulated in two steps: first, an example-based Chromatic Adaptation Transform (CAT) has been designed to obtain an illuminant matching between input and example images. Second, the dominant colors of the input and example images are optimally mapped. The main strength of the method comes from using optimal transportation to map a pair of meaningful color palettes,and regularizing this mapping through thin plate splines.

Results:

Source Image

Example Image

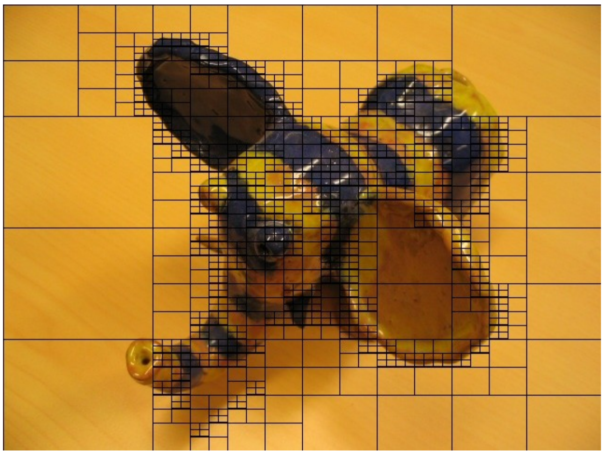


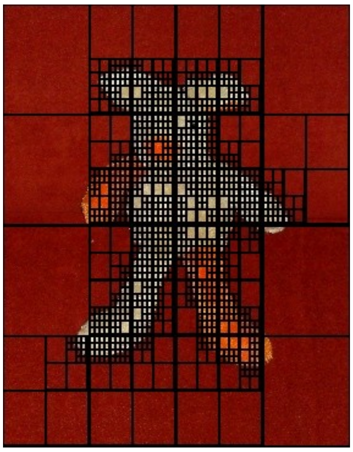
Adaptive Partition

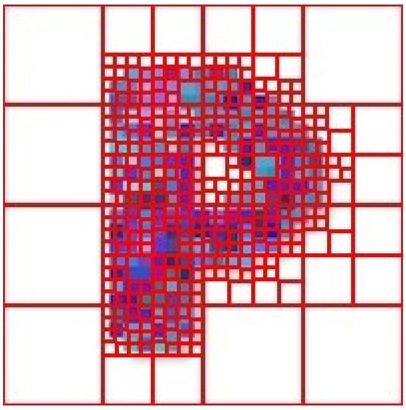
Only texture transfer

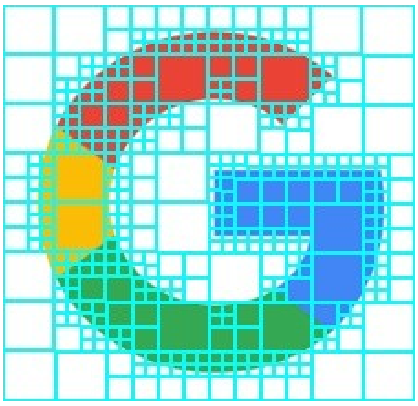
Final output

Some more results for Adaptive Partition









Conclusion

Proposed a new style transfer method is able to synthesize textures independently of their scale. This method is naturally not guaranteed to transfer textures that belong to the same semantic category in the input and example images.