Implementation of Texture Transfer Algorithm

Mradul Singhal*, Dhruv Goyal†, Jatin gupta‡, Abhishek Keshari§
*Enrollment No: 22116055, Email: mradul_s@ece.iitr.ac.in
†Enrollment No: 22112033, Email: dhruv_g1@ece.iitr.ac.in
‡Enrollment No: 22116038, Email: jatin_g@ece.iitr.ac.in
§Enrollment No: 22116003, Email: abhishek_k2@ece.iitr.ac.in

***Abstract*—This project was made as a part of *Digital Image Processing (ECN-316)* course under the guidance of *Dr. Vinod Pankajakshan*. It creates a way to transfer texture from one image onto the structure of another. It uses a method that combines picking small parts (patches) of texture and blending them smoothly to make the result look natural. The main steps include choosing texture patches, matching them to the target image, and blending the edges to avoid sharp lines. The method was tested on different images and gave good results. This can be used in photo editing and creating artistic effects. The project shows how texture manipulation can be made simple and useful for creative tasks.**

I. Motivation

The challenge in texture transfer lies in seamlessly applying the texture of a source image to the structure of a target image. Traditional methods, such as pixel-by-pixel synthesis, were computationally expensive and struggled with structured patterns, resulting in visible seams and poor alignment.

To overcome these shortcomings, we used patch based synthesis, which replaces pixels with larger patches to better capture texture patterns. To address the issue of seamless blending, we used customized cut to compute optimal boundaries for patch overlap, ensuring smooth transitions.

The Texture Transfer algorithm implemented in this project is based on the Research aricle published by Alexei A. Efros and William T. Freeman. [1] and also refers from the work by Emil Praun, Adam Finkelstein, and Hugues Hoppe. [2]

II. Introduction

Any image can be broken into two major parts: Texture and geomtric structure.

A. Texture

The texture of an image refers to the surface patterns, repetitions, or variations in intensity and color that give a sense of how the surface would ”feel” if touched (e.g., rough, smooth, grainy).

B. Geometric Structure

The geometric structure of an image refers to the overall shape, contours, and spatial arrangement of objects, defining their form and layout within the image.

The texture transfer algorithm generates a new image by combining these two properties of two different images. It combines the source texture of source image with the structure of the target image while ensuring seamless transitions between patches.

The algorithm's primary objective is to:

- Combine the texture from the source image with the structure from the target image.
- Ensure smooth transitions between patches to avoid visible seams.

III. Algorithm Workflow

The texture transfer algorithm involves three main phases: initialization, patch-based synthesis, and refinement. Fig. 2 provides an overview.

A. Initialization

The algorithm begins by loading the source and target images. Parameters such as patch size, overlap, tolerance, and the number of iterations are set. The output image is initialized as a blank image with the same size as the target image.

B. Patch-Based Synthesis

The target image is divided into overlapping patches. For each patch:

1) Candidate patches are selected from the source texture.
2) The similarity between patches is computed using the Sum of Squared Differences (SSD).
3) The best matching patch is selected, and a customized cut is computed to ensure seamless blending.

C. Refinement

The algorithm repeats the patch-based synthesis process for a predefined number of iterations. Patch size and overlap are reduced in each iteration, capturing finer details.

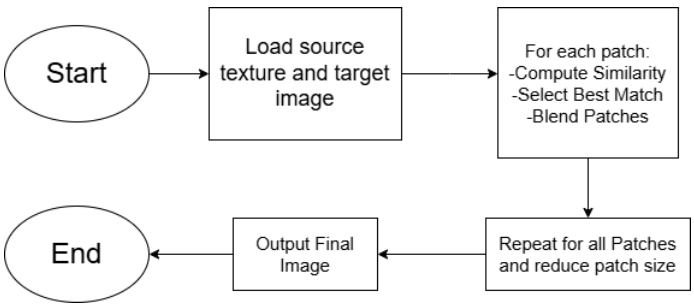

Fig. 1. Overview of the Texture Transfer Algorithm Workflow

IV. Key Components

A. Input Parameters

The algorithm relies on the following key parameters:

- **Patch Size:** Determines the size of texture patches.
- **Overlap:** Ensures smooth blending between neighboring patches.
- **Tolerance:** The tolerance that is acceptable.
- **Iterations:** Controls the number of refinement steps.
- **Reduction:** Determines the reduction in patch size after each iteration.

B. Sum of Squared Differences (SSD)

SSD measures the similarity between patches by computing the squared differences in pixel intensities. It gives information on how much the selected patch resembles from target structure and previously synthesized texture. Let there be two patches A and B, then their corresponding SSD is given by:

$$\text{SSD}(A, B) = \sum_{i=1}^{m} \sum_{j=1}^{n} (A(i,j) - B(i,j))^2$$

C. Customized Cut

We used dynamic programming to compute the optimal path through overlapping regions, to minimize the mismatch between neighboring patches.

D. Alpha Parameter

The alpha parameter balances the weight between preserving the target structure and maintaining texture consistency. In our code we used below relation for alpha parameter, for the given number of iterations, n, and iter_numth term.

$$\alpha = 0.1 + 0.8 \cdot \frac{n}{\text{iter\_num} - 1}$$

V. Applications

The texture transfer algorithm is widely used in:

- **Artistic Rendering:** Stylizing images using textures like brushstrokes or converting a coloured painting to black and white.
- **Material Synthesis:** Generating new materials by combining textures.
- **Visual Effects:** Applying realistic textures to 3D models.
- **Image Editing:** Enhancing photos by blending textures.

VI. Output Images

Fig.2 shows some sample outputs:

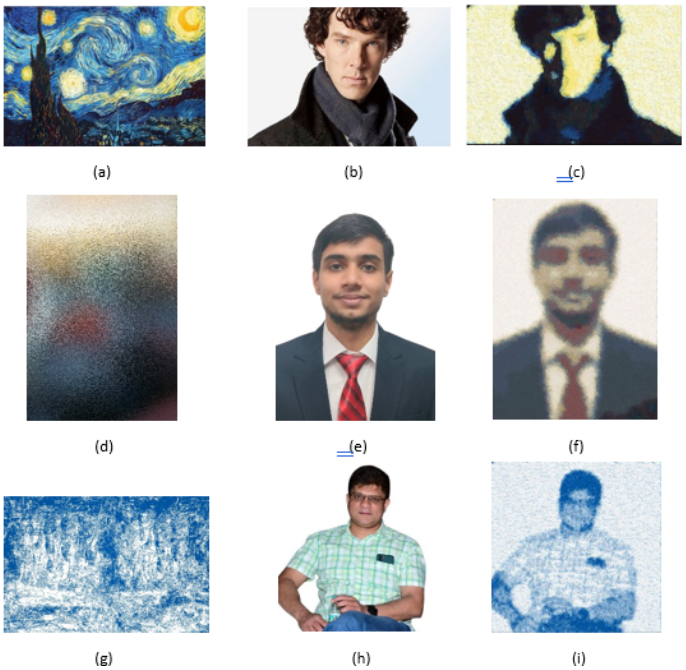

Fig. 2. (a), (d), (g) are source textures. (b), (e), (h) are target images. (c), (f), (i) are corresponding final outputs

VII. Team Members

1) **Mradul Singhal (22116055)** - Implemented the Texture Transfer algorithm along with the preparation of Report.
2) **Dhruv Goyal (22112033)** - Written and linked the frontend and backend, also wrote clear documentation
3) **Jatin Gupta (22116038)** - Implemented the Texture Transfer algorithm and helped in writing the documentation
4) **Abhishek Keshari (22116003)** - Research part of the project with the preparation of Report

References

[1] Alexei A. Efros and William T. Freeman. 2001. Image quilting for texture synthesis and transfer. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '01)*. Association for Computing Machinery, New York, NY, USA, 341–346. https://doi.org/10.1145/383259.383296

[2] Emil Praun, Adam Finkelstein, and Hugues Hoppe. 2000. Lapped textures. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '00)*. ACM Press/Addison-Wesley Publishing Co., USA, 465–470. https://doi.org/10.1145/344779.344987