Color Preserving Image Transformation with Scissor Collage

Ziv Epstein[1], Robin Pollak[1], and Dmitriy Smirnov[1]

1 **Computer Science Department, Pomona College**
Claremont,CA
`{ziv.epstein, robin.pollak, dmitriy.smirnov}@pomona.edu`

Abstract

1998 ACM Subject Classification I.3.5 Computational Geometry and Object Modelling – Geometric Algorithms, languages and systems, K.3.1 Computer Uses in Education – Computer-assisted instruction

Keywords and phrases polygonal congruence, color matching, image transformation, geometry, rigid transformations

Digital Object Identifier 10.4230/LIPIcs.SoCG.2016.66

1 Introduction

2 Algorithm

We begin by receiving an uploaded target image, I^t, and a query term used to retrieve a second source image,I^s . Then, we triangulate each image into a set of colored triangles $T^t = \{\Delta_1^t, \cdots, \Delta_n^t\}$ and $T^s = \{\Delta_1^s, \cdots, \Delta_m^s\}$. Next we use k-means to cluster the colors in I^t and I^s to get the top k colors of each [**?**, k-means]y extracting the color centroids from each clustering. Denote these sets $\{C_1^t, \cdots, C_k^t\}$ and $\{C_1^s, \cdots, C_k^s\}$. We then pair up these colors by implementing weighted bipartite matching on the graph with the edge weight of edge $[C_i^t, C^s, j] = |H(C_i^t) - H(C_j^s)|$ where $H(C)$ is the hue of color C.

Now, for both triangulations T^t and T^s, we recolor the triangles using the color redistrubtion algorithm and get T_1^s and T_1^s to ensure the total areas of each color is the same for all k colors across all triangles. Given these color balanced triangulations T_1^t and T_1^s we now recolor each triangle in T_1^s according to the matching described above and get T_2^s. Finally, we use scissors congruence to recreate P_{target} from P_{base}.

2.1 Color Redistribution Algorithm

Given a colored triangulation T and color centroids $C = \{C_1, \cdots, C_k\}$, first for each triangle Δ in T we compute the average hue of the pixels in its interior, $\bar{H}(\Delta)$ and then color it with $C_j = \arg\min_C |\bar{H}(\Delta) - H(C_i)|$, the closest of the k colors centroids. Then we balance the total area of each color, by creating k bins B_j with an area threshold of $\sigma = \frac{1}{k}\sum_{\Delta\in T} \text{area}(\Delta)$ and an area

$$\text{area}(B_j) = \sum_{\Delta\in T} (\text{area}(\Delta)|\Delta \text{ is colored } C_j).$$

Now, for a given bin B_j, if $\text{area}(B_j) > \delta$, then we consider the smallest triangle $\Delta_{min} \in B_j$. If $\text{area}(B_j) - \text{area}(\delta_{min}) \geq \delta$ then we recolor Δ_{min} to the bin with the closest hue B_ℓ with $\text{area}(B_\ell) < \delta$. If $\text{area}(B_j) - \text{area}(\delta_{min}) < \delta$, that is Δ_{min} cannot be removed with diminishing the area of B_j below δ, we instead consider the largest triangle $\Delta_{max} \in B_j$. We divide t_max into two smaller triangles by drawing a ray from a vertex such that one of the new triangles Δ'_{max} has area area$(B_j) - \delta$ and recolor Δ'_{max} as described above. This process is repeated until there is are no bins to place new triangles in.

3 Implementation

References

1 William Wallace and John Lowry. ‘Question 269’. *New Series of the Mathematical Repository 3* (1814). Ed. by Thomas Leybourn, pp. 44–46.