Simplified Paintings-from-Polygons is NP-Hard

Daan van den Berg[0000−0001−5060−3342]

University of Amsterdam
d.vandenberg@uva.nl

Abstract. The simplified paintings-from-polygons problem (SPFP), in which paintings or other digital images are approximated by heuristically arranging overlapping semi-opaque colored polygons, is NP-hard. Every instance of the subset sum problem can be transformed to a SPFP instance, solved by some algorithm, and transformed back. Whichever algorithm one chooses, it cannot be more efficient than the most efficient algorithm for subset sum. Since subset sum is known to be NP-hard, and SPFP is at least equally hard, SPFP must also be NP-hard.

Keywords: Paintings-from-Polygons · PFP · Alpha Compositing · NP-hard · Evolutionary Algorithms · Plant Propagation Algorithm

1 Introduction

Being a blog topic of artistic nature for quite some time, approximating paintings from optimally arranging a limited set of semi-opaque colored polygons has been elevated to the realm of science since EvoMusArt 2019 [1] [3]. Having both a strong visual appeal and an untraversably large combinatorial state space, the problem proves an appealing testing ground for heuristic algorithms such as hill climbing, simulated annealing and the plant propagation algorithm [4][6][1][5]. Starting from an initial configuration of randomly scattered semi-opaque (partially) overlapping polygons, the various algorithms applied four mutation types: moving a polygon's vertex, changing a polygon's color and/or opacity, transferring a vertex from one polygon to another, and changing the 'drawing index' – assigning which polygon is to be drawn first, which second, and so forth during the rendering process. Numbers of vertices and polygons were constant throughout the experiments, and the change in pixel-by-pixel difference between the rendered polygon constellation and the target image, quantified in the mean squared error (MSE), monitored the progress for the different optimization algorithms (Fig. 2).

However visually appealing, no formal proof of the problem's hardness was given by the authors. To make some headway in this direction, I will show that a simplified version of this problem is NP-hard. The simplification is rather strict: given a number of 50% opaque greyscale polygons and a target greyscale image, the objective is to approximate any single pixel within the target image as closely

[1] Conference track of EvoStar 2019 in Leipzig, Germany

as possible. Apart from some practicalities and common sense requirements (the number of vertices per polygon should be ≥ 3, there should be more than 0 polygons etc.) the problem is as general as possible. A key role in the proof is played by the polygons' opaqueness and their drawing index – the order in which they are rendered to a 'flat' .png image, a process known as *alpha compositing*. For every pixel in a polygon constellation, the lowest level is given by the background color of the canvas, which is always black, solid, and fully opaque:

$$color_0 = 0 \tag{1}$$

Then, the color of the pixel is sequentially updated for each polygon covering that pixel. This can be recursively formulated as

$$color_i = \alpha_i \cdot color_i + (1 - \alpha_i) \cdot color_{i-1} \tag{2}$$

in which i indicates the i^{th} polygon covering the pixel, and α_i its opaqueness. Note that therefore, the influence of the lastly drawn polygon usually outweighs the influence of earlier drawn polygons on the rendered pixel's final color. For SPFP, all polygons have fixed 50% opaqueness, simplifying equation 3 to

$$color_i = \frac{1}{2} color_i + \frac{1}{2} \cdot color_{i-1} \tag{3}$$

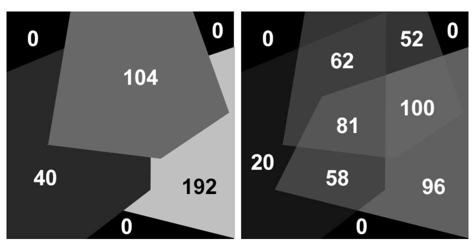

Fig. 1. Left: alpha-compositing of three fully opaque ($\alpha = 1$) greyscale polygons on a black canvas. **Right:** the same polygons, drawn in the same order, but in half opaqueness ($\alpha = 0.5$). Numbers inside the areas are greyscale values.

So if a polygon constellation holds seven polygons with greyscale colors (192, 40, 104, 16, 85, 17, 50) of which the first three are chosen to cover a pixel, the pixel's rendered color would be

$$\frac{1}{2} \cdot 104 + \frac{1}{4} \cdot 40 + \frac{1}{8} \cdot 192 = 81 \tag{4}$$

(Fig. 1 contains this exact numerical example in the central area). After sequentially rendering all the polygons in the constellation, the resulting 'flat' .png

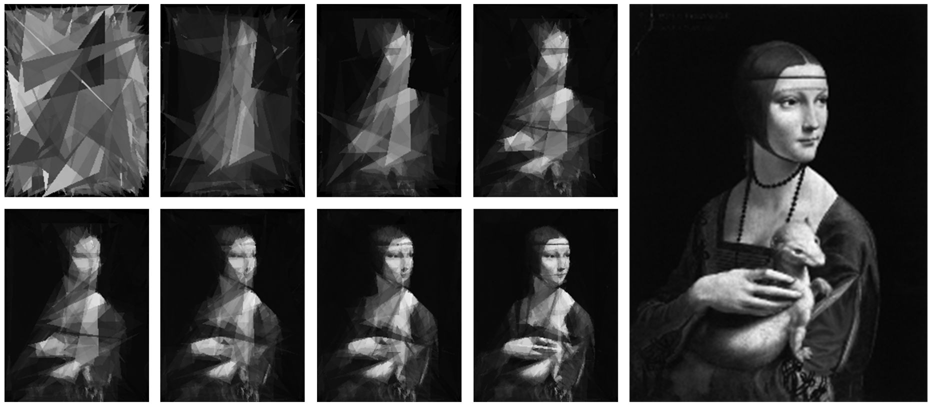

Fig. 2. The (simplified) paintings-from-polygons problem ((S)PFP) involves approximating a target image, usually a painting, from heuristically rearranging a set of semi-opaque polygons. The objective is to minimize the pixel-by-pixel error (MSE) between the rendered polygon constellation (smaller subfigures LRTB) and the target image (large subfigure). Numbers of vertices and polygons are fixed throughout the run.

image can then be MSE-wise compared to the target image. Better (heuristic) algorithms obtain ever lower MSE-values throughout the run, arranging polygon constellations ever more towards the target image (Fig. 2).

2 NP-Hardness: From Subset Sum to SPFP

In its optimization form, subset sum is the task of approximating a target value t as closely as possible by adding up a number of elements from a set S of m given integers $v_1...v_m$, for example $S = \{13, 17, 21, 23\}$ with $t = 41$. The problem is known to be NP-hard, which means there is no known exact algorithm that runs in subexponential time [2]. In its decision form ("which of the integers v sum up *exactly* to t ?"), it is NP-complete because of its polynomial verifiability.

The core of the proof to SPFP's NP-hardness is this: any subset sum-instance can be *transformed* to a SPFP-instance by creating a corresponding same-value greyscale polygon p_i for every integer v_i, and choosing a pixel within a target image such that $t_{pix} = t$. A polygon's restricted greyscale value mutations are multiplications by 2^k, in which k corresponds to the k^{th} chosen polygon. For our example, this leads to the following transformation

$$\{13, 17, 21, 23\} \rightarrow \begin{pmatrix} 26 \\ 52 \\ 104 \\ 208 \end{pmatrix} \begin{pmatrix} 34 \\ 68 \\ 136 \\ 272 \end{pmatrix} \begin{pmatrix} 42 \\ 84 \\ 168 \\ 336 \end{pmatrix} \begin{pmatrix} 46 \\ 92 \\ 184 \\ 368 \end{pmatrix}$$

in which the polygons' admissible greyscale values are vertically aligned, a transformation that by all means can be made in (quadratic) polynomial time. Now instead of selecting some values $v_1 ... v_m$ from set S, the task is selecting polygons $w_1 ... w_m$, and for each k^{th} selected polygon, mutating its color by a factor 2^k. Note that this transformation allows for the commutativity in subset sum, which is absent in SPFP. After selecting the polygons, the resulting value can be calculated as analogous to Equation 4 for alpha compositing,

$$\frac{1}{2} \cdot w_{1,1} + \frac{1}{4} \cdot w_{2,2} + ... + \frac{1}{2^m} \cdot w_{m,m}. \tag{5}$$

After choosing (the right) polygons and comparing the result from Eq.5 to target pixel t_{pix}, any solution can be polynomially transformed back to subset sum, by just taking the original greyscale values of the selected polygons, and dividing them by two. Thereby, any subset sum instance with a set of n integers can be transformed in polynomial time to a single-pixel approximation SPFP-instance with n polygons of $\alpha = 0.5$, and its solution can be transformed back, again in polynomial time. It follows that any algorithm that solves SPFP in time $O(f(x))$ can also solve subset sum in $O(f(x))$. Since subset sum is NP-hard, $f(x)$ cannot be subexponential, and SPFP must also be NP-hard.

3 Discussion & Acknowledgments

The original PFP-problem's mutable color and opaqueness, topological constraints, image dimensions and drawing indices are all likely of influence on the problem's hardness. How exactly, still remains to be quantified. Gratitude goes out to Tim Doolan (UvA) for constructive feedback and to Arne Meijs (UvA) for helping out with figure 2.

References

1. Geleijn, R., van der Meer, M., van der Post, Q., van den Berg, D.: The plant propagation algorithm on timetables: First results. EVO* 2019 p. 2 (2019)
2. Kleinberg, J., Tardos, E.: Algorithm design. Pearson Education India (2006)
3. Paauw, M., Van den Berg, D.: Paintings, polygons and plant propagation. Springer (2019)
4. Salhi, A., Fraga, E.S.: Nature-inspired optimisation approaches and the new plant propagation algorithm (2011)
5. Sulaiman, M., Salhi, A., Fraga, E.S., Mashwani, W.K., Rashidi, M.M.: A novel plant propagation algorithm: modifications and implementation. Science International **28**(1), 201–209 (2016)
6. Vrielink, W., van den Berg, D.: Fireworks algorithm versus plant propagation algorithm. In: IJCCI. pp. 101–112 (2019)