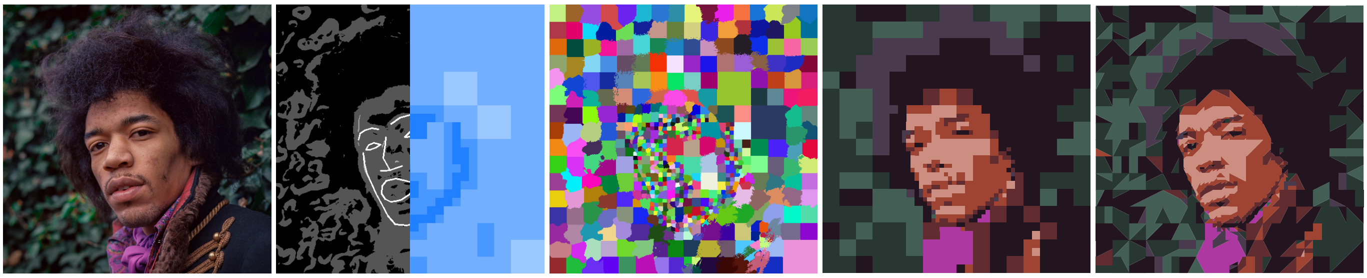
Vision, Modeling, and Visualization (2024)
L. Linsen and J. Thies (Eds.)

Geometric Portrait Stylization

D. R. Bukenberger

University of Tübingen & Technical University of Munich, Germany



Source Image | Detail & Quadtree Levels | Relaxed Superpixels | Pixelization | Stylized Result

Figure 1: *Visualizations of the individual stages and processes in our geometric abstraction pipeline, exemplified on a portrait photograph.*

Abstract

Our work extends common pixelization techniques, enabling novel geometric pop-art stylization. We employ dedicated feature analysis to autonomously extract facial features, ensuring the best recognizability of persons and facial expressions in portraits. Additionally, our method includes automated content-related detail level extraction for scenic image content. Based on these detail levels, a hierarchical structure sets the basis for non-uniform pixelization. A joint optimization routine computes a reduced color palette alongside the coarse superpixel segmentation. We propose an adapted modification to common superpixel methods to handle non-uniform sized cells, maintaining a comparable level of detail while allowing for a coarser, more pixelated look. Additionally, this intermediate result serves as the basis for our geometric abstraction by eventually clustering polygonal shapes based on the pixelization. Colors and shapes are derived from the source image to capture and reproduce the most essential details for recognizable characters and facial expressions. We document the theoretical details of our method, discuss and elaborate the possible extensions. Provided results of the intermediate pixelization are compared qualitatively to related methods. Compared to other stylization methods, our resulting geometric abstractions are generated automatically, preserving a high level of relevant details from the source image. Unlike simple filtering techniques or learning-based stylization methods, our approach allows for the incorporation of user input to highlight features. Furthermore, our method stays true to the original image and results in scale-independent vector graphics, rendering it a valuable tool for artists and graphic designers.

CCS Concepts* • *Computing methodologies* → *Image processing; *Non-photorealistic rendering; Computational photography;*

1. Introduction

Over the past decade, machine learning methodologies have substantially transformed the landscape of image generation and stylization [JYF*19]. Dedicated convolutional neural networks (CNNs) trained for specific tasks have proven instrumental in addressing numerous challenges encountered in computer vision tasks. However, the opaque nature of common image generation approaches presents challenges, as manual adjustments often involve iterating through various prompts in a trial-and-error manner, relying on chance for satisfactory outcomes. This inherent randomness complicates and obscures the artistic process. Particularly for artists with a well-defined creative vision, a controllable and deterministic abstraction process is a fundamental requirement.

High-resolution displays with very high pixel densities have become prevalent, surpassing the perceptual capabilities of the human visual system to discern individual pixels. In contrast, during the early stage of the home computer market, displays possessed limited capabilities, typically offering resolutions of less than 0.1 megapixels and a restricted color palette. The aesthetic appeal of graphics and games from this era, characterized by distinct constraints, inspired pixel-artists to incorporate these limitations into their artwork, requiring skillful abstraction techniques. Our modification of the utilized superpixel method introduces further coarsening to our intermediate pixelization result, allowing for a comparable level of detail as other methods while further emphasizing the block abstraction style.

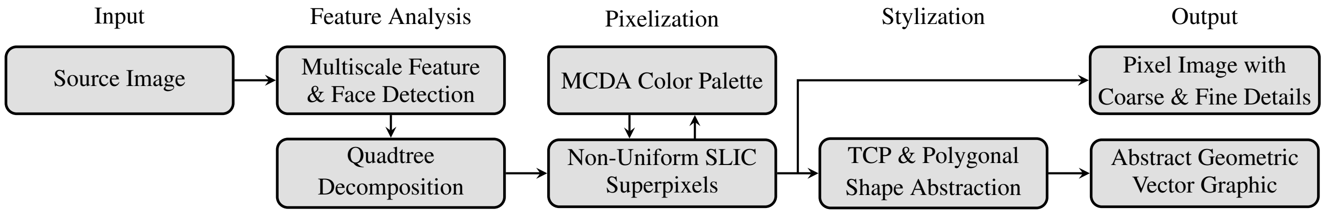


Figure 2: *Schematic steps of our pipeline, visualized in Figure 1. Automated feature analysis serves as input for the quadtree decomposition. This initializes the joint superpixel and color palette iteration. Abstract geometric results are based on the non-uniformly sized superpixels.*

While conventional pixelization methods typically focus on reducing the size of the input image to low resolution, we aim to relax this constraint in order to facilitate greater artistic flexibility. Our aim is to decrease resolution primarily in homogeneously colored regions, such as blurry backgrounds in portrait images or larger smooth areas in scenic photographs, while preserving details in crucial regions to ensure the recognizability of faces and objects. The objective of our paper is to advance existing techniques by combining them for producing refined artistic abstractions, surpassing mere pixelization. Therefore, we extend superpixels [ASS*10] to a non-uniform sized grid and allow both autonomous or user-guided methodologies for their generation. Furthermore, our generated pixelated outcomes serve as a input for approaching vectorized stylizations of *Wedha's Pop Art Portrait* (WPAP) style [Ras]. This particular art style adeptly captures details of varying sizes through geometric abstraction using shape-constrained polygons. Notably, these polygons can be colored with hues unrelated to the original image yet still yield recognizable and visually appealing abstractions. In our approach we utilized the non-uniform superpixel layout as basis for *Two-Colored-Pixel* (TCP) rendering. A TCP is characterized by a straight line dividing the pixel into two halves, each assigned a distinct color. This approach combines the constrained color palette and the superpixel clustering, enabling the creation of coarse abstract polygonal shapes faithfully recreating the input images appeal and fine details.

2. Related Work

Faithfully vectorizing low-resolution images is an ongoing challenge, with existing geometric- [KL11] and learning-based approaches [RGLM21]. Similarly, generating stylized pixel art characters or sprites with a distinctive retro video game aesthetic from given input presents an open non-trivial task. Works by Gerstner et al. [GDA*12, GDA*13] introduced techniques, to create (semi- or fully-automated) very low-resolution images with a restricted color palette based on a given input image, aiming to generate abstractions resembling the work of pixel-artists. This is achieved by combining two iterative converging procedures: *Simple Linear Iterative Clustering* (SLIC) Superpixels [ASS*10] group multiple pixels in a 5-dimensional feature space (CIELAB color and XY position), facilitating significant reduction in image resolution. A reduced color palette is determined using *Mass Constrained Deterministic Annealing* (MCDA) [Ros98], which implements a cool-down process starting with one mean-sample and high initialization-energy. In a subsequent publication, they expanded their approach to incorporate user constraints, allowing users to define weights for specific image regions through an importance map. These weights directly influence the MCDA, while superpixel clustering is indirectly affected by the available colors from the palette.

As extension of the SLIC concept, we introduce non-uniformly sized superpixels. These superpixels of varying sizes subdivide the initial image based on attention-guided level of detail maps, manually specified or automatically extracted from the input image. To determine the appropriate level of detail for general image content, we employ a method inspired by the work of Itti et al. [IKN98], utilizing multi-color channel feature analysis within a Gaussian-pyramid framework. For portraits, we employ a dedicated face detection algorithm proposed by Kazemi and Sullivan [KS14, SAT*16, Kin09] to infer a detail map from detected facial landmarks. A specialized quadtree algorithm utilizes the level of detail map to cluster the image and extract starting positions for the superpixel relaxation. This enables the creation of larger superpixels in homogeneous image regions to achieve greater abstraction, while preserving higher resolution in important regions such as sharp image features or facial details.

While recent learning-based methodologies concentrate on pixelizing game scenes and generating sprites [HWH*18, WCZ*22], deterministic methods for image stylization of this nature also rely on superpixels. For instance, the SLIC algorithm is commonly utilized for reducing color palettes to achieve a posterization effect [CSG21] or for portrait pixelization [SW21].

3. Method

The following elaborates details of our proposed pipeline, depicted in Figure 2. First, level-of-detail maps are extracted from the input or specified manually, followed by quadtree partitioning to facilitate the generation of non-uniformly sized superpixels. The combined annealing processes of converging superpixels and the restricted color palette forms the basis for geometric abstraction. TCPs further subdivide the superpixels and create coarse geometric abstractions using polygonal shapes. Visualizations of interim pipeline steps are presented in Figure 1, illustrating automatically generated detail maps, pixelization and resulting stylizations.

3.1. God is in the Details

A key feature in our abstraction process is the preservation of details from the original image, including sharp edges depicting objects or facial features in portrait photographs. To guide this abstraction with multiple levels of detail, we integrate straightforward grayscale importance maps. These maps, which can be either user-specified or generated automatically, serve as input for a quadtree decomposition of the input image. Centers of the quadtree cells provide the initial positions for the superpixels used in the next stage. For automatically generated detail maps, we provide a general method for capturing details in scenic images or a specialized method for portraits focused on faces. To offer more artistic control, detail maps can be incorporated from user-defined input as well.

General Feature Analysis Our method for extracting importance maps from scenic input images is derived from the bottom-up saliency-analysis approach of Itti et al. [IKN98]. Therein, a Gaussian pyramid of the input image is generated up to a suitable level. Subsequently, separated XY-Sobel filters are applied to the color channels of each level, and the combined Euclidean magnitude of all (rescaled) levels is accumulated in one map and normalized. A simple quantization process discretizes the resulting grayscale map into the desired number of discrete detail levels. Figure 3 shows the detail level analyses with the level-of-detail map in gray and the resulting quadtree level with shades of blue. Resulting superpixels of difference sizes (after the relaxation) are visualized on the right.

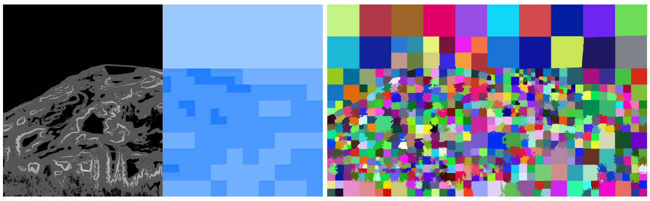


Figure 3: *A scenic mountain image, source and result in Figure 10. Left: saliency map extracted from the Gaussian pyramid quantized to 4 levels, then subdivided with a quadtree visualized in blue. Right: superpixels of different sizes after the relaxation process.*

Portrait Landmarks The importance maps provided by the multiscale feature extraction prove suitable for scenic images as it effectively highlights sharp features and focuses on high-frequency details throughout the image. However, for artistic portrait abstraction, only specific details are relevant, primarily the facial features such as the eyes, nose, mouth, and optionally the jawline. To address this, we employ the simple 68-landmark facial feature extraction technique proposed by Kazemi et al. [KS14] to generate dedicated importance maps tailored for portrait image input. These 68 feature points are categorized into groups representing the outline of the mouth and nose, the left and right eyes and eyebrows, and the estimated jawline. The importance map is then generated by simply tracing the points of the individual landmark groups.

For certain portrait inputs, facial details alone can sufficiently capture the essence of the image. This is facilitated by the quadtree subdivision approach, which gradually decreases cell sizes from coarser to finer details, ensuring intermediate regions are represented with reasonable detail. However, as illustrated in Figure 4, if the face region covers only a small region of the image, large portions may remain at very low quadtree levels, resulting in excessively coarse abstractions. To address this, we can incorporate the basic detail map that is quantized to only $k-1$ levels and portrait features are then assigned the highest level. This encourages more quadtree subdivisions in non-facial regions, enabling to capture more details in clothing or backgrounds as well.

Quadtree Decomposition The quadtree decomposition subdivides the image space based on the introduces level-of-detail maps, resulting in smaller cells in regions of higher detail and larger cells in coarser image regions. Centers of these quadtree cells serve as starting positions for the superpixels used in the upcoming step. The root level of the quadtree is initialized as a regular grid with equally sized cells. The initial grid resolution is set very coarse with 8×8 or 16×16 cells and specifies the maximum size of the coarsest superpixels. Each cell then examines all its pixels in the quantized detail-level map. If the cell contains integer values higher than its own level, it is a split into four smaller cells. This process continues recursively until neither cell requires further splits or until a maximum detail level is reached. For included results we used a maximum recursion depth of 3 to four levels. Figure 3 illustrates how smaller quadtree cells approach edges highlighted in the detail map, enabling the creation of detail-maintaining abstractions.

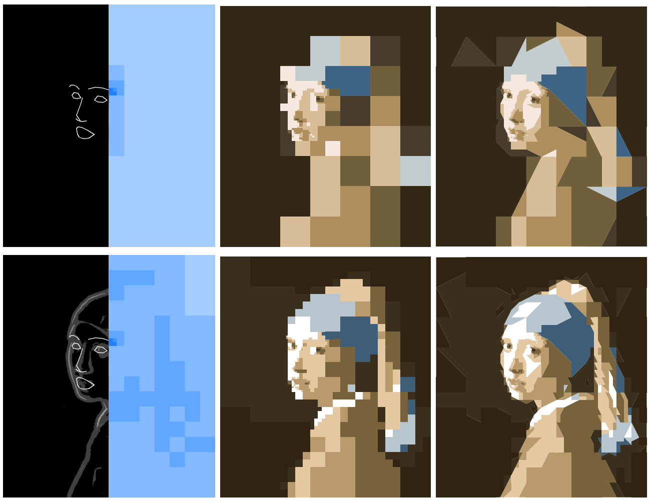


Figure 4: *Left: Extracted portrait landmarks (top) are combined (bottom) with the saliency map (gray) to guide the quadtree decomposition (blue). This ensures expressive facial features while still capturing essential details in clothing or background. Middle and right, respectively, show the pixelization and geometric stylization.*

3.2. Non-Uniform Pixelization

SLIC superpixels are favored for pixelated image stylization [SW21] over simple downsampling because they facilitate clustering based on both position and color in a 5-dimensional relaxation process. However, for our specific application, we need to adapt the formulation of the distance metrics used for the clustering to accommodate our non-uniformly sized superpixels.

Size Adaptive Clustering The principle of the SLIC clustering method proposed by Achanta et al. [ASS*10] closely resembles a simple k-Means iteration or a Lloyd relaxation on a Voronoi diagram. In this method, the image space is treated as a continuous space, sampled on the discrete pixel grid. For initialization, a fixed number of k superpixel centers is distributed over the input image either on a regular grid or, in our case, based on the quadtree cell centers. Subsequently, the following two steps are iterated until convergence, i.e., no further movement in an iteration:

- For each real image pixel, the *closest* superpixel center is located, and the pixel is labeled accordingly.
- The superpixel centers are repositioned to the centroid of all pixels labeled with their respective ID.

Proceedings published by Eurographics - The European Association for Computer Graphics.

A superpixel $\mathbf{C}_i = [L_i, a_i, b_i, x_i, y_i]^T$ is a vector of five dimensions, featuring three color values in the *Lab* space and two spatial *xy* dimensions. To fine-tune characteristics of the clustering and relaxation process and compensate for the different scaling of the color space and image dimensions, it's necessary to assign different weights to the distinct components. The metric specifying the distance between a superpixel i and image pixel j formulates as

$$D(i,j) = d_{Lab}(i,j) + \frac{m}{S} d_{xy}(i,j) \tag{1}$$

with separated Euclidean norms

$$d_{Lab}(i,j) = \sqrt{(L_j - L_i)^2 + (a_j - a_i)^2 + (b_j - b_i)^2}, \tag{2}$$

$$d_{xy}(i,j) = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}. \tag{3}$$

Achanta et al. propose the weighting in Equation (1) using $S = \sqrt{\frac{n}{k}}$, where n is the number of pixels in the input image, k the number of superpixels and m can be fine-tuned but was set to 1 for included results. As an example, Figure 5 illustrates the effects of different weightings: with a higher weight on color, superpixels become more fuzzy (middle), and with more weight on spatial distance, they are more compact (right).

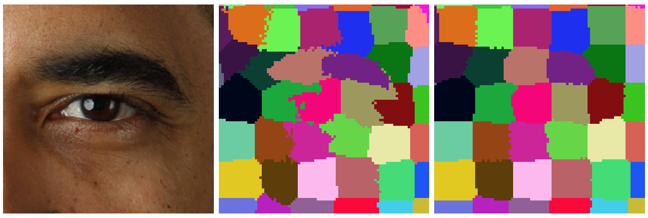


Figure 5: *Comparing the effect of weighting from Equation* (1) *with a detail of the source image on the left. Center: Color outweighs spatial distance and superpixels become more fuzzy. Right: Spatial distance outweighs color and superpixels are more compact.*

The original SLIC metric formulated in Equation (1) assumes equally sized superpixels. However, as illustrated in Figure 6, the standard Euclidean L_2 norm as a measure for spatial distance introduces a bias that deforms the superpixel shape, even in this perfect regular arrangement. While neighboring superpixels of the same size are unaffected, adjacent superpixels of different sizes are separated by curved cell boundaries. To circumvent this effect, we utilize the L_∞ norm and replace the spatial metric with

$$d_{xy}(i,j) = \max\left(|x_j - x_i|, |y_j - y_i|\right). \tag{4}$$

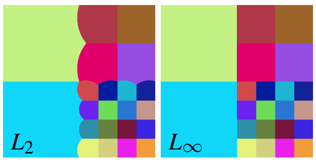


Figure 6: *With the L_2 norm, superpixels of difference size form rounded shapes, even in regular configurations. The L_∞ norm counteracts this behavior.*

3.3. Constrained Color Palette

To constrain the color palette during the abstraction process, we employ the *Mass Constrained Deterministic Annealing* (MCDA) algorithm [Ros98], which operates based in the three-dimensional color space of the input image. The algorithm aims to determine k clusters representing the final color palette. Unlike k-means, where all clusters are present from the start, MCDA initializes with only one cluster positioned at the mean position of all data points. Samples are subsequently assigned to their closest cluster center, which is repositioned in an iterative relaxation procedure. The system's global energy is initialized as twice the total variance over all samples, and the energy is decreased by a factor $\alpha < 1$ with each iteration. While Gerstner et al. use $\alpha = 0.7$ to decrease the system's energy, we observed that a slower convergence rate yields improved results, thus use $\alpha = 0.9$. If a cluster's largest eigenvalue exceeds the system's energy, it undergoes a split. In the original algorithm, this split is implemented as a *small directed random jitter* applied to each new cluster center. However, we found that using the clusters largest eigenvector as offsets in both positive and negative directions facilitates slightly faster convergence. The number of palette colors k can be user-specified, and by default, the annealing process continues until the color palette has converged. This stopping criterion can be implemented as the distance between palette colors, before and after each iteration, falling below a reasonably small threshold [GDA*12]. However, in rare edge cases, some images may cause palette oscillation. While this could be mitigated by introducing small amounts of noise to the palette, we chose not to introduce randomness to the process. Instead, we terminate an oscillating iteration after reaching a maximum number of steps.

3.4. Combined Iteration

A crucial aspect of the pixelation approach lies in the synergy resulting from the combined iterations of SLIC and MCDA. Therefore the initial SLIC iteration is extended by the two MCDA steps: Each iteration cycle involves the assignment of pixels and relaxation of superpixel clusters (SLIC), as well as the association, refinement, and expansion steps of the color palette (MCDA). The system iterates the following steps until palette convergence is achieved, i.e., when the specified number of colors is reached and the system energy falls below a certain threshold (we chose 1):

i. Label each pixel with the ID of the closest superpixel.
ii. Reposition superpixels to the geometric center of all pixels, labeled with their respective ID.
iii. Probabilistically associate color samples from the input image with superpixel colors and refine the matching.
iv. Lower the palette's energy, eventually leading to palette expansion if colors are split, and assign superpixels to new colors.

Extensions Additional improvements on the color palette can be achieved through bilateral smoothing, which aims to strengthen the color relation between directly adjacent superpixels. Therefore, the pixelization is assembled using the superpixels in their designated places, filtered bilaterally, and then resampled by the superpixels to store the filtered color. This operation takes place between steps ii. and iii. so that the palette refinement starts on smoothed colors.

Furthermore, in step iii., importance maps may be incorporated to weigh the influence of input pixels on the color palette. However, these importance maps are not be confused with the detail-level maps used in Section 3.1 to guide the quadtree decomposition.

Positional Constraints Our formulation, utilizing the L_∞ norm for superpixel shapes, resembles the idea of mosaic stylization by

Hausner [Hau01]. However, unlike mosaic tiles, superpixels should not float around uncontrollably as during a Lloyd relaxation but should be bound to specific destinations. Furthermore, to maintain a structured quad-like grid and ensure that superpixels remain somewhat close to their origin during the SLIC relaxation, additional constraints are necessary. Therefore, one can employ simple explicit Laplace smoothing [GDA*12], which formulates positional constraints relative to the mean position of the four direct superpixel neighbors. However, concerning the non-uniform superpixels introduced in Section 3.2, this quadrilateral grid neighborhood is not trivially given. Instead we use each superpixel's own initial position ${}_{xy}C^0$ as the destination of the smoothing force: After a relaxation step, each superpixels spacial xy component is updated as

$$ {}_{xy}C_i^{k+1} = (1-\alpha)\cdot {}_{xy}C_i^k + \alpha \cdot {}_{xy}C_i^0. \tag{5} $$

Extending this approach further also allows for individual α_i values for each superpixel, dependent on a normalized color distance to its origin. Thus, we formulate smoothing weights as $\alpha_i = d_{Lab}({}_{Lab}C_i^k, {}_{Lab}C_i^0)$. This has the effect that if a superpixel is close to a strong gradient and gets *pushed over*, the color distance and therefore the force pulling it back to its initial position will increase. This allows superpixels in homogeneously colored regions more roaming space during relaxation but keeps superpixels in high-frequency areas tighter bound to their designated position.

3.5. Geometric Stylization

The result of the modified pixelization may serve as a basis for further abstraction and stylization techniques. This obviously contrasts the initial purpose of classical pixelization, but provides a greater degree of artistic flexibility.

Two-Colored Pixels In general, a TCP is defined as a square pixel intersected by a line at two points, which is rendered with two different colors on both sides of the line [PK10]. We utilize this concept as an extended rendering extension for our superpixel results. Therefore, each superpixel is simply replaced by a TCP, thus covers multiple pixels of the input image. The separating line is determined based on the area of the source image, covered by the superpixel to be replaced by a TCP. However, for our application we limit the possible colors of the TCPs to the constrained color palette extracted in the prior pixelization.

With I being the input image containing all pixels p, a TCP searches for the optimal line L by minimizing the energy

$$ E(I,L) = \sum_{L(p)<0} \left| I_p - c^- \right| + \sum_{L(p)\geq 0} \left| I_p - c^+ \right| \tag{6} $$

where $L(p)$ states on which side of the line pixel p is. The averaged pixel colors from either side of the line are denoted as c^- and c^+, respectively, but will be eventually replaced by palette colors.

The number of possible line configurations depends on the number of real image pixels, covered by a superpixel, i.e., corresponding TCP. With our superpixels being derived from a quadtree, the number of pixels covered by a superpixel increases by powers of 2 per level. However, feasible computation can be maintained, using the hierarchical approach proposed by Pavic et al. [PK10].

The final geometric abstraction is constructed from TCPs of different sizes, constrained to the color palette. Consequently, same-colored portions of adjacent TCPs form coarse polygons that represent semantic details of the input image both in shape and color.

Output Details As the coarse abstraction ultimately comprises only polygonal shapes, the output is generated in vector graphic format and does not require further rasterization of the TCPs. The color palette is the result of clustering, thus may be subject to some averaging bias. To compensate for this and give the results more of a pop-art appeal, we slightly increase the lightness and saturation of the extracted palette for the export. For creating characteristic stylizations of black and white portraits, one could also apply user-specified palette colors or create a new one by randomizing the hue values and only maintaining the lightness of the colors. Figure 8 includes an example generated from a black and white input image with randomized color palette. As the polygons in the output are simple 2D coordinates, the method is suitable for stereo-consistent stylization methods [BSGL15] that remap the monoscopic points to a stereo image pair.

4. Results and Discussion

This section features comparisons of our intermediate pixelation results to other pure pixelization methods. Furthermore, we qualitatively evaluate the results of our geometric stylization approach, also in comparison to results of diffusion based image generation.

Input Other Method Our Pixelization Our Stylization

Figure 7: *A comparison to related pixelization methods by Gerstner et al. [GDA*13] (top two rows), Shang et al. [SW21] (bottom two rows) to our pixelization and final abstraction results.*

Proceedings published by Eurographics - The European Association for Computer Graphics.

4.1. Pixelization Results

The SLIC-based pixelization core of our method shares similarities with those proposed by Gerstner et al. [GDA*12, GDA*13] and Shang et al. [SW21]. A key distinction in our method, however, is the hierarchical multiscale approach with differently sized superpixels. Even when approaching similar resolutions as compared methods at fine details in our images, larger homogeneous regions are represented with larger superpixels, thus realizing a more blocky look and coarser abstraction. This allows us to maintain and represent a similar level of detail as related works but introduces a much coarser look of the pixel-style abstraction in general. Exemplary comparisons with results from related works are included in Figure 7, also featuring our intermediate pixelization results. This direct comparison highlights the coarser look of our results; for example, in the turban in the top row image or the hat in the bottom row, which are represented with a much coarser resolution, yet are still recognizable. Nonetheless, the quadtree subdivision, focused on extracted facial features, allows us to spend more and smaller pixels in facial regions like eyes, nose, and mouth, eventually matching the resolution of compared results. This dedication to detail also impacts performance, as we have to handle way fewer superpixels during the relaxation. For example, the bottom row result of Shang et al. includes $76 \times 76 = 5776$ superpixels, while ours only has 1621 superpixels, as the larger ones simply cover more space. Rendering the coarse pixelization using our proposed geometric abstraction with polygonal shapes allows to mitigate effects like staircasing and regaining more fine image details in general.

4.2. Geometric Abstraction

In addition to pixelization, which serves as an intermediate result, our focus lies on the stylization of images using abstract geometric polygons. The following presents results of this method applied to scenic image content as well as portrait images.

Portrait Image Stylization As stated with the title, the focus of our method lies in the stylization of portrait images. The dedicated use of facial feature extraction allows us to prioritize faces in the segmentation with the highest detail level. Figure 8 lists results of our portrait stylization alongside the source images.

Most noticeable here is how well facial features are preserved, expressions, emotions as well as characters are still recognizable. This is enabled by the smaller non-uniform superpixel sizes, as these allow more detail in highlighted feature regions. Furthermore, the TCP rendering recovers more shading details and sharp gradients from the pixelated base image. The reduced color palette is sampled from the source image, thus staying true to the input, which allows us to transport the actual look, feeling, or mood present in the source image to the abstraction.

Scenic Content The images in Figure 10 demonstrate the applicability of the abstraction method to images without faces. The abstraction is then guided solely by sharp gradients and prominent shape contours, extracted by the Gaussian pyramid. Nevertheless, the initial quadtree segmentation again facilitates the abstraction of larger homogeneous regions while still maintaining structure in fine and small details. For example, the blurry background in the

Figure 8: *Due to the dedicated facial feature extraction before the abstraction, our method applies well to portrait image content, regardless of whether it's photographed or painted. Enhanced colors give the geometric abstraction more pop-art appeal, while a randomized palette can create iconic stylizations from black and white images, as shown in the bottom example.*

upper image or the sky in the lower image are represented by large coherent polygons, while details of the flowers, butterfly wings, or shadings of the mountainside are reproduced with finer shapes.

Figure 10: *Examples of non-portrait content are solely based on detail-level-maps exacted from the input images, and capturing fine details while coarsening homogeneous regions.*

Comparison to Diffusion Models A key feature of diffusion models is their ability for large-scale generalization and applicability to reproduced a vast amount of different styles. However, this also makes it difficult to pin them down to an exact artistic vision and may entail a lengthy trial and error process with experimental configurations of prompts and pretrained models. Nevertheless, we let an experienced user try to recreate the geometric abstraction style for a comparison with our results, all featured in Figure 9. Three different base models were used: a simple basic model, one for stylized portraits (DynaVision), and one specialized on cinematic characters (Juggernaut), further a specially trained WPAP LoRA and enhancement styles *SAI Lowpoly* and *Minimalism*. The picked images are the results of around 30 different experimental model and prompt configurations to produce satisfactory results, roughly resembling the anticipated style. Eventually, all three results were produced using the prompt: *wpap style, geometric abstraction, polygon shapes, no color, grayscale, black and white*. The diffusion process is also initialized with image features from the source image, extracted with a multiresolution Canny edge filter. This allows to guide posture and image arrangement. Additionally, a specialized face-swapping feature was included, which should reproduce the face as true to character as possible.

Arguably, all three models produce visually striking images, clearly based on the given input image. However, the level of abstraction varies quite drastically: from lots of colorful polygons in the first model, to coarse shapes in the second one (strictly omitting the face), to only reduced gray tones and just a few shapes in the background. Further, dependent on the used diffusion model, faces are always biased by the most prominently featured style in their training data. So, none of the diffusion results actually generates a face recognizable as *Johnny Cash*. Neither did one of them pick up his sad facial expression nor roughly the correct age, and all of them even changed the direction he is looking at. Another essential feature for this particular image, depicting *the man in black*, is the grayscale color palette. Although explicitly stated in various formulations in the prompt, only one instance managed to actually produce a grayscale image while the others feature various bright colors, probably due to the strong influence in the trained models.

4.3. Implementation and Performance

Many of the employed algorithms are primed for parallelization with a proper GPU implementation to enhance overall performance. For instance, the SLIC algorithm is realized using a z-Buffer approach, which is trivially well-suited for hardware acceleration [HIKL*99]. However, even with our simple *Python* implementation, the iteration routines usually converge within seconds for megapixel-sized input images. Preceding steps (Section 3.1), such as hierarchical feature analysis, face landmark extraction, or quadtree segmentation, finish instantly. The generation of TCPs requires some additional computational effort due to the brute-force nature of finding the optimal separator line configurations. However, as TCPs are independent of each other, they can be computed in parallel on a multithreaded CPU in a matter of seconds as well. For example, our implementation generated the result in Figure 9 (right) in a total of 90 seconds on a 4 GHz CPU. For comparison, the shown diffusion results were produced on a 1080Ti GPU in 244, 514, and 524 seconds, respectively. The implementation of our method to reproduce the shown results is available at https://github.com/dbukenberger/GeometricPortraitStylization.

Figure 9: *Comparing results from image diffusion models and our geometric style (right). The diffusion process is also guided by essential features from the input, thus reproduces posture and image arrangement. Faces and colors are, however, heavily biased from predominant styles in the model training data. Diffusion results are pixel images of limited resolution whereas ours is exported as vector graphic.*

Proceedings published by Eurographics - The European Association for Computer Graphics.

4.4. Further Application

Results of our geometric abstraction are vector graphics in SVG format, easily renderable in all common web browsers. Thus, they may serve as a simple inverse steganography method: For instance, when our version of a person's portrait is included on a website, the portrayed person remains easily recognizable to humans, but the image itself is not trivially parseable by automated web crawlers, assembling large-scale face recognition databases. These crawlers would first need to rasterize the vector graphic to determine if it contains a person at all, adding a layer of complexity that could help protect the identity of individuals depicted in the images. Therefore, it might be useful to recreate the input image more accurately and with less abstraction. The image on the right of Figure 11 provides an example, offering a wider color palette and no hierarchical detail reduction, thus creating a more faithful representation of the input image. Due to the increased resolution, the unoptimized SVG file on the right has about 70 % of the filesize of the lossless input PNG. The result with coarser abstraction on the left has about 70 % of the filesize compared to a compressed JPG version.

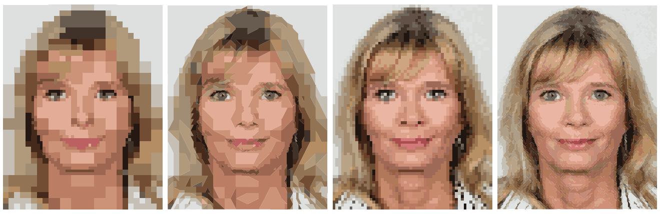


Figure 11: *The coarser abstraction on the left uses two hierarchy levels and 16 colors, the right one has no hierarchy and 32 colors. Both abstractions have the same base resolution of* 56×72 *pixels.*

5. Conclusion

In this work, we introduce a novel method for geometric stylization of portrait images that also generalizes well for scenic image content. As an intermediate step, we extend common pixelization methods and adapt them to feature non-uniform superpixels by exchanging the L_2 norm for the L_∞ norm as the spatial distance metric. Our method utilizes the coarse pixelated result and, in combination with a selectively relaxed color palette for posterization, uses TCPs as the rendering method. The hierarchical segmentation, based on image features, initializes the superpixels and allows for larger geometric abstraction in homogeneous regions while maintaining fine shapes in small details. Dedicated facial landmark detection enriches the level-of-detail map for a strong emphasis on faithful face abstraction, allowing for recognizable characters in portraits with convincing facial expressions and emotions. In the evaluation, we compared our pixelization results to related methods, demonstrating how our non-uniform pixelization can produce much coarser-looking abstractions while reproducing detail at the same level as competing methods. In a comparison with popular diffusion-based image generation methods, we highlighted the difficulties in such approaches to produce images in a given style while maintaining character details from a given input. As our method utilizes non-uniform and larger superpixels, much fewer of them are required in the computation, thus facilitate faster converging computations. Besides rasterized pixel graphics, our pipeline also naively supports output as vector data suitable for plots, large-scale prints, as basis for further manual stylization or inverse steganography method *hiding* portraits from web crawlers.

References

[ASS*10] ACHANTA R., SHAJI A., SMITH K., LUCCHI A., FUA P., SÜSSTRUNK S.: *SLIC Superpixels*. Tech. rep., Ecole Polytechnique Fédrale de Lausanne, 2010.

[BSGL15] BUKENBERGER D. R., SCHWARZ K., GROH F., LENSCH H. P. A.: Rotoscoping on Stereoscopic Images and Videos. In *Vision, Modeling & Visualization* (2015), Eurographics Association, pp. 111–118.

[CSG21] CHAO C.-K. T., SINGH K., GINGOLD Y.: Posterchild: Blend-aware artistic posterization. In *Computer Graphics Forum* (2021), vol. 40.4, Wiley Online Library, pp. 87–99.

[GDA*12] GERSTNER T., DECARLO D., ALEXA M., FINKELSTEIN A., GINGOLD Y., NEALEN A.: Pixelated image abstraction. In *Proceedings of the Symposium on Non-Photorealistic Animation and Rendering* (2012), Eurographics Association, pp. 29–36.

[GDA*13] GERSTNER T., DECARLO D., ALEXA M., FINKELSTEIN A., GINGOLD Y., NEALEN A.: Pixelated image abstraction with integrated user constraints. *Computers & Graphics 37*, 5 (2013), 333–347.

[Hau01] HAUSNER A.: Simulating decorative mosaics. In *Computer graphics and interactive techniques* (2001), ACM, pp. 573–580.

[HIKL*99] HOFF III K. E., KEYSER J., LIN M., MANOCHA D., CULVER T.: Fast computation of generalized Voronoi diagrams using graphics hardware. In *Proceedings of Computer graphics and interactive techniques* (1999), pp. 277–286.

[HWH*18] HAN C., WEN Q., HE S., ZHU Q., TAN Y., HAN G., WONG T.-T.: Deep unsupervised pixelization. *ACM Transactions on Graphics (TOG) 37.6* (2018), 1–11.

[IKN98] ITTI L., KOCH C., NIEBUR E.: A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence 20.11* (1998), 1254–1259.

[JYF*19] JING Y., YANG Y., FENG Z., YE J., YU Y., SONG M.: Neural style transfer: A review. *IEEE transactions on visualization and computer graphics 26.11* (2019), 3365–3385.

[Kin09] KING D. E.: Dlib-ml: A Machine Learning Toolkit. *Journal of Machine Learning Research 10* (2009), 1755–1758.

[KL11] KOPF J., LISCHINSKI D.: Depixelizing Pixel Art. *ACM Transactions on Graphics 30.4* (2011), 99:1 – 99:8.

[KS14] KAZEMI V., SULLIVAN J.: One millisecond face alignment with an ensemble of regression trees. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2014), pp. 1867–1874.

[PK10] PAVIĆ D., KOBBELT L.: Two-Colored Pixels. In *Computer Graphics Forum* (2010), vol. 29.2, Wiley Online Library, pp. 743–752.

[Ras] RASYID W.: Wedha's Pop Art Portrait (WPAP) Style. [Online], https://www.instagram.com/wpapcom/.

[RGLM21] REDDY P., GHARBI M., LUKAC M., MITRA N. J.: Im2vec: Synthesizing vector graphics without vector supervision. In *Proceedings of the IEEE/CVF* (2021), pp. 7342–7351.

[Ros98] ROSE K.: Deterministic annealing for clustering, compression, classification, regression, and related optimization problems. *Proceedings of the IEEE 86*, 11 (1998), 2210–2239.

[SAT*16] SAGONAS C., ANTONAKOS E., TZIMIROPOULOS G., ZAFEIRIOU S., PANTIC M.: 300 faces in-the-wild challenge: Database and results. *Image and vision computing 47* (2016), 3–18.

[SW21] SHANG Y., WONG H.-C.: Automatic portrait image pixelization. *Computers & Graphics 95* (2021), 47–59.

[WCZ*22] WU Z., CHAI L., ZHAO N., DENG B., LIU Y., WEN Q., WANG J., HE S.: Make your own sprites: Aliasing-aware and cell-controllable pixelization. *ACM Transactions on Graphics (TOG) 41.6* (2022), 1–16.

Proceedings published by Eurographics - The European Association for Computer Graphics.