

Color Restoration of Scanned Archaeological Artifacts with Repetitive Patterns

D. Gilad-Glickman & I. Shimshoni¹

¹University of Haifa, Department of Information Systems, Israel

Abstract

Our work addresses the problem of virtually restoring archaeological artifacts. Virtual restoration is the process of creating a noise-free model of a degraded object, to visualize its original appearance. Our work focuses on restoring the coloring of the object. We considered both 2D and 3D objects, including scans of ancient texts and 3D models of decorated pottery. Our denoising method exploits typical characteristics of archaeological artifacts, such as repetitive decoration motifs and a limited palette of colors. Our classification method is based on minimization of an energy function, which includes a correspondence term, to encourage consistent labeling of similar regions. The energy function is minimized using the Graph-Cuts algorithm.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Color, shading, shadowing, and texture I.4.6 [Image Processing and Computer Vision]: Segmentation—Pixel classification I.5.3 [Pattern Recognition]: Clustering—Similarity measures

1. Introduction

Historically significant artifacts tend to deteriorate over time. The objects may break, erode, get stained, and surface colors may fade or peel off. In our work, we address the problem of virtually restoring models of archaeological artifacts. We aim to restore only the color component, ignoring possible defects in the object's shape. We consider both 2D scans of ancient texts, as well as 3D scans of decorated pottery. Given a digital version of the object, we wish to alter its colors to simulate the object's original appearance. We restore the original appearance of a scanned object, by assigning each pixel a label representing its noise-free color.

Archeological artifacts are decorated, in many cases, by repetitive decoration motifs, painted with a small number of colors. Re-occurring letters in text may be considered a special case of such repetitions. Our method exploits those two characteristics. We assume that the number of colors is known, and that the degradation is of a gradual nature, and use Gaussian Mixture Models (GMM) to estimate the main colors. Repeating patterns are used to deduce the original appearance on damaged areas, assuming similar regions have deteriorated differently. We create a denoised version of the object by color classification; each point is assigned a label (i.e. a cluster index), representing the noise-free color it most likely originated from. The optimal labeling is detected by minimization of an energy function. Energy functions are usually comprised of a data term and a smoothness term. The data term measures the disagreement between a labeling f and the observed data I , and the smoothness term imposes spatial smoothness on labels of neigh-

boring points. We add a correspondence term to the energy function. The correspondence term acts on pairs of matching points, and encourages consistent labeling of similar regions. Our labeling method thus integrates both local and semantic cues, as well as a probabilistic prior, to infer the optimal labeling of the data. The correspondence term can be considered a generalization of the smoothness term; while the smoothness term imposes similarity on pixels that are geometrically close, the correspondence term enforces similarity on pixels that are semantically close. The energy function is minimized using Graph-Cuts, and each class is colored by the average color of all its members. The resulting model is considered a reconstructed version of the object. The method was tested on images of ancient texts and 3D models of archaeological artifacts. We compare the results of our algorithm to labeling according to GMM, and to the original Graph-Cuts. In both the 2D and 3D cases, our method achieved superior results; it managed to remove large portions of the noise, while keeping the repetitive pattern intact.

2. Algorithmic Flow

2.1. Method Overview

Our method aims to improve labeling results, by utilizing similarities within the data. We do so by minimizing an energy function with an additional *semantic similarity* element. The new energy term is represented in the graph by *correspondence edges*. Correspondence edges assemble corresponding vertices into cliques

in the graph, thus encouraging consistent labeling of matching regions. While the new similarity term enriches the graph with a new type of edges, it does not affect the minimization procedure. We thus minimize the extended graph using the same graph-cut algorithm that was defined on graphs with the original structure. For simplicity, the concept of the algorithm will first be exemplified on 2D images of ancient texts and then extended to 3D objects.

2.2. Modeling Color Distribution

Ancient texts are mostly bi-chromatic. Similarly, archeological artifacts are decorated with a limited pallet of colors [ALY08]. The colors of scanned texts and objects, however, often vary from the original colors. Limiting the number of main colors allows us to transform the de-noising problem into a clustering task. We model the color distribution and determine the main colors. We then compute the initial labeling by clustering each point to the most probable main colors. Modeling the colors' distribution is done using a Gaussian mixture model (GMM). The number of main colors is assumed to be known - given by a human observer or automatically inferred from the data. GMM clustering is done based on color information solely, ignoring any spatial information. Clustering is performed on points in a 3 dimensional color space (L^*, u^*, v^*) (CIELUV).

2.3. Energy Function

We search for the optimal labeling of the data by minimizing an energy function. Our energy function contains a data term and a smoothness term. But, in order to encourage consistent labeling of matching regions, we add an additional term, which we consider a *correspondence term*. The new term defines a cost for disconnecting corresponding points, i.e., assigning different labels to matching vertices. Adding the correspondence term to the energy equation yields:

$$E(f) = E_{data}(f) + E_{smooth}(f) + E_{corr}(f), \quad (1)$$

where the data term E_{data} represents the initial probability that a vertex has originated from each of the main colors. E_{data} is based on the prior probabilities computed by the GMM. The smoothness term E_{smooth} is based on the Potts model as an interaction penalty. The correspondence term E_{corr} imposes smoothness on labels of corresponding points. i.e. it encourages similar labeling of corresponding regions. Both the smoothness term and the correspondence term impose similarity between pixels. But while the smoothness term imposes similarity on pixels that are *geometrically* close, the correspondence term enforces similarity on pixels that are *semantically* close. The minimal cut of the graph is detected using the standard graph-cuts algorithm [BVZ01].

2.4. Similarity search

We consider two pixels p and q to be semantically close if their local neighborhoods N_p, N_q are similar. We detect matching regions by correlating local neighborhood (patches) with the full image (in the 2D case). Query patches are taken from all over the image, as overlapping tiles, with a vertical and horizontal shift of half window-size between one another. Since letters in texts tend

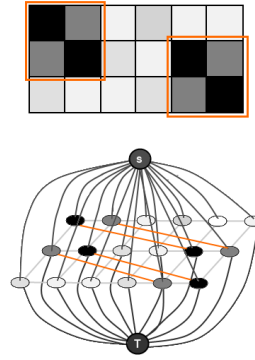


Figure 1: Matching patches on image (top), C-link edges connect corresponding vertices (bottom)

to have similar sizes and alignment, we did not use neither scaling nor rotation in the 2D case. We compute the normalized cross-correlation between each query patch and the full image, and apply non-maximal suppression [Kov00] to detect local maxima in the correlation map. Patches centered at correlation peaks are considered matching regions to the query patch.

2.5. Correspondence Term Weights

C-links (marked orange in Fig. 1), connect pairs of corresponding pixels, and thus define a semantic neighborhood system. We consider points with similar neighborhoods as matching points. The cost of a C-link corresponds to a penalty for discontinuity in labeling of corresponding pixels. Cutting a C-link represents the assignment of different labels to matching points.

In the 2D case, each query patch N_q is correlated with the full image. After detecting its matching regions $N_i^q, i = [1, \dots, m]$, the patches are aligned, and each pixel q in the query patch N_q is connected to its corresponding pixels $p_i \in N_i^q$. In the graph, vertex v_q is connected by C-links to its corresponding vertices v_{p_i} (see Fig. 1). We assume that corresponding pixels behave similarly to adjacent pixels.

3. 3D Adjustments

We extended the method for 3D objects, decorated by repetitive patterns. We wish to alter the appearance of the color data solely. Implementing the method on 3D objects required several adjustments. The major issues to consider were:

1. 3D object scans are not sampled on a regular grid
2. Color correlation between non-planar patches is not well defined
3. Object decorations may include similar regions in different orientations, as opposed to letters in texts, which are usually well aligned.

The initial stage - modeling the color distribution and initial labeling was done as before, see Section (2.2), based on vertices' colors.

3.1. Query Patches and Search Region

To reduce the complexity of the problem, we project the patches on the 2D tangent plane. To avoid major distortions in the projected image, we limit the size of the patches to an area with relatively small curvature. Similarly, the size of the search region is limited due to the object's geometry. While in the 2D case the whole object was included in the search region, here it may contain just a section of it. In the 2D case, query patches were taken sequentially, from the entire image. In the 3D case, however, we chose to mark the query patches manually, at points of interest. This reduces the complexity of the computation, by omitting the matching stage of unique or irrelevant patches. Figure 5 displays all query patches (marked in red. Marks appear partial when occluded by the object). The search region in this case is centered at the blue mark, and includes the entire facade. The user defines a radius for each patch, and all points within a geodesic distance smaller than that radius are considered part of the patch. Choosing query patches of different sizes enables the preservation of both fine details and more general structures.

3.2. Patch Projection and Rotation

Color correlation is not well defined between non-planar surfaces. To overcome this issue, we project both the search region and the query patches onto a plane tangent to the surface at the center of each patch. After projecting the vertices onto the tangent plane, we resample it to form an image with a regular square grid. We would like to allow detection of similar regions with different orientations. We thus project each query patch in multiple directions, with $d\theta = 5^\circ$, for $\theta \in [0, 360^\circ)$. Projecting in multiple directions also allows patches with self-symmetries to contribute multiple instances.

3.3. Matching and Alignment

After rotating the patches, projecting and resampling, we can now search for matching regions using 2D correlation, as explained in Section (2.4). We compute the correlation between the search region and all rotated versions of the query patches, and transform the pixels of maximal correlation into vertices in the mesh. Each query patch is now related to a list of maximum correlating vertices. We define a patch around each correlation peak, and align them with the non-rotated query patch.

3.4. 3D Neighboring

3D meshes are not regularly sampled, and require a more fuzzy definition of neighborhood. Practically, we detect the K-nearest neighbors of each vertex by efficiently searching a KD-tree [FBF77]. We add a distance component to the weight functions, to compensate for the variability in neighbor distances.

3.5. Data Term and Smoothness Term Weights

The data term for the 3D case is the same as for the 2D case. Smoothness edges connect vertices to their local neighbors. We also use the Potts model as an interaction penalty, but add a distance element to the weight function, to reflect the non-regularity of the grid.

3.6. Correspondence Term Weights

C-links connect corresponding vertices. After aligning the corresponding regions with the query patch, we connect each vertex in the query patch to its K-nearest neighbors (with $K=100$) from each *corr_patch* (j). In the graph, vertex v_q is connected by C-links to its corresponding vertices v_p , with weights similar to the smoothness edges.

4. Results

This section presents the results of our method, when applied to 2D images and 3D models of painted object. We compare our method (GCC) to the initial labeling by maximizing GMM probabilities and to the original Graph-Cuts (GC) method.

4.1. 2D images

We first tested our method on 2D images of ancient texts. The text are handwritten, which allows for greater variability even within different instances of the same letter. We consider the whole image as the search region. Correspondences are defined between reoccurring letters, so we thus set the size of the query patches ($N_Q \times N_Q$) according to the size of the letters. Query patches are taken from all over the image, as overlapping tiles, with a vertical and horizontal shift of half window-size between one another. No attempt was made to fit the tiles exactly around the letters.



Figure 2: Left: Papyrus 66, Dated 200 AD. Part from the Bodmer Papyri collection, Right: GMM labeling

Papyrus 66, displayed in Fig. 2, is a Greek manuscript of the Gospel of John. The Papyrus is part of the Bodmer Papyri collection and is dated 200 AD ([Ala74], [Hun61], [PB56], [MBB62]). The result of GMM clustering are presented in Fig. 2. Graph Cut labeling is shown in Fig. 3 (left). The result of our method is presented in Fig. 3 (right). The following parameters were used: Number of classes = 2, $N_Q = 30$ pixels, Correlation threshold = 0.6. As can be seen, the result of our algorithm is much better than the ordinary graph-cuts result. Much of the noise that is present in the GC result, was successfully removed by the GCC, while the letters in the text stayed intact.

4.2. 3D objects

The method was tested on 3D models of archaeological artifacts, scanned in the Computerized Archaeology Laboratory, at the Institute of Archaeology in the Hebrew University of Jerusalem. Search region and query patches were manually marked at points of interest on the model. The query patch radius (r_Q), was set according

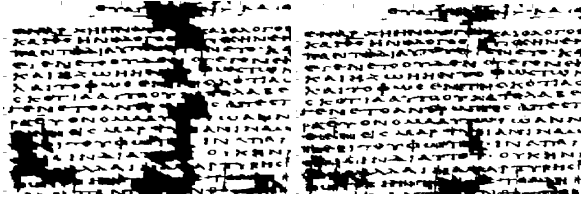


Figure 3: Papyrus 66: binary image. Left: GC labeling, Right: GCC labeling

to element size, and the size of the search region is determined according to the curvature of the surface.

Fig. 4 presents the upper part of a decorated Phoenician bichrome jug from the Tel Dor excavation, dated to late Iron Age I [Gil99]. To avoid large distortions in the projected image we restored only the decorated facade.



Figure 4: Phoenician Bi-chromatic jug

In Fig 5(left), the query patches are marked by red circles, and the center of the search region is marked blue. The entire (cropped) model is considered as the search region. Fig. 5(right) presents the results of clustering according to GMM probabilities. The following parameters were selected: 6 classes, $r_Q = 10$ mm, Correlation threshold = 0.6. Fig. 6(left) and Fig. 6(right) presents the results of GC and GCC labeling, respectively. Both GC methods manage to smooth out much of the noise, but while GC merges the red lines, GCC separates them.

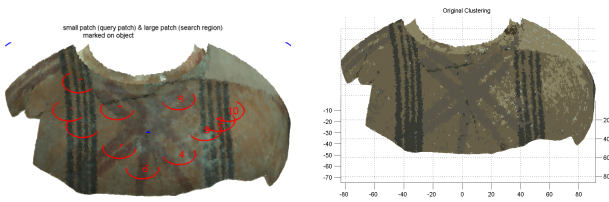


Figure 5: Left: Decorated facade. Query patches marked red on search region (the entire facade). Right: Labeling according to GMM, colored by average color

5. Conclusions

We have presented a method for virtually restoring the colors of archaeological artifacts. Our method utilizes repetitive patterns, such

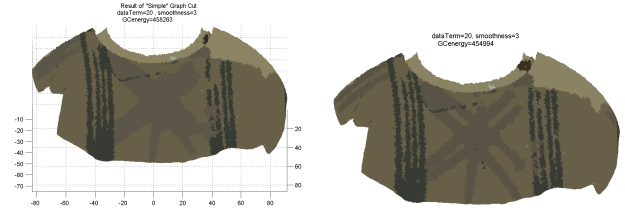


Figure 6: Left: GC labeling, colored by average color. Right: GCC labeling, colored by average color

as decoration motifs and reoccurring letters, to infer the original appearance of degraded regions. In the restoration process, each point is assigned a label, representing the color it most likely originated from. The optimal labeling is detected by minimizing an energy function using the Graph-Cuts algorithm. Our energy function includes the standard data and smoothness terms. To encourage consistent labeling of matching regions, we added a third (correspondence) term, to the energy function. The correspondence term acts upon pairs of matching points, assigning a cost for inconsistent labeling of corresponding points. Our method thus incorporates both local and semantic information, along with a probabilistic prior, to infer the optimal labeling. We also detailed the adjustments needed for applying the method to 3D models. The method was tested on 2D images of ancient texts and 3D models of decorated pottery, yielding better results relative to those of GMM and the original Graph-Cuts method. The algorithm successfully removed noise, while preserving and enhancing the repetitive patterns.

Acknowledgements: This research was funded in part by GRAVITATE under EU2020-REFLECTIVE-7-2014.

References

- [Ala74] ALAND K.: Neue neutestamentliche papyri iii. *New Testament Studies* 20, 04 (1974), 357–381. 3
- [ALY08] ALIAGA D. G., LAW A. J., YEUNG Y. H.: A virtual restoration stage for real-world objects. *ACM Transactions on Graphics (TOG)* 27, 5 (2008), 149. 2
- [BVZ01] BOYKOV Y., VEKSLER O., ZABIH R.: Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23, 11 (2001), 1222–1239. 2
- [FBF77] FRIEDMAN J. H., BENTLEY J. L., FINKEL R. A.: An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software (TOMS)* 3, 3 (1977), 209–226. 3
- [Gil99] GILBOA A.: The dynamics of phoenician bichrome pottery: a view from tel dor. *Bulletin of the American Schools of Oriental Research* (1999), 1–22. 4
- [Hun61] HUNGER H.: Zur datierung des papyrus bodmer ii (p66). *Anzeiger der Österreichischen Akademie der Wissenschaften philosophisch-historische Klasse* 97 (1961), 12–23. 3
- [Kov00] KOVESI P. D.: Matlab and octave functions for computer vision and image processing. Online: http://www.csse.uwa.edu.au/pk/Research/MatlabFns/#_match (2000). 2
- [MBB62] MARTIN V., BARNES J. W. B., BODMERIANA B.: *Papyrus Bodmer II, supplément: Évangile de Jean, chap. 14-21*. Bibliothèque Bodmer, 1962. 3
- [PB56] POPYRUS BODMER I.: *Evangile de jean chap. 1-14*. Publié par Victor MARTIN (1956). 3