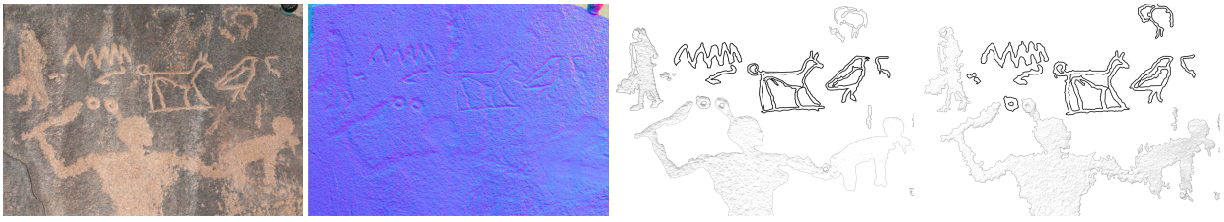
Non-Photorealistic Animation and Rendering
David Mould and Pierre Bénard (Editors)

EXPRESSIVE 2015

Semi-Automatic Digital Epigraphy from Images with Normals

Sema Berkiten[1], Xinyi Fan[1], and Szymon Rusinkiewicz[1]

[1]Princeton University, USA



(a) Color Image (b) Surface Normals (c) Manual Digital Epigraphy (d) Our Result

Figure 1: ***(a)*** *Albedo and* ***(b)*** *Normal maps are estimated with a photometric stereo algorithm.* ***(c)*** *Digital epigraphy done manually on Adobe Photoshop and Illustrator in high precision by archaeologists which took 7-8 hours in total.* ***(d)*** *Result from our user-guided system which took 6-7 minutes.*

Abstract

We present a semi-automated system for converting photometric datasets (RGB images with normals) into geometry-aware non-photorealistic illustrations that obey the common conventions of epigraphy (black-and-white archaeological drawings of inscriptions). We focus on rock inscriptions formed by carving into or pecking out the rock surface: these are characteristically rough with shallow relief, making the problem very challenging for previous line drawing methods. Our system allows the user to easily outline the inscriptions on the rock surface, then segment out the inscriptions and create line drawings and shaded renderings in a variety of styles. We explore both constant-width and tilt-indicating lines, as well as locally shape-revealing shading. Our system produces more understandable illustrations than previous NPR techniques, successfully converting epigraphy from a manual and painstaking process into a user-guided semi-automatic process.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Line and curve generation

1. Introduction

Scientific illustrations are used in various fields to represent objects in a more effective and succinct way than is possible with photography. In archaeology, it is paramount for publications of record to use illustrations in agreed-upon styles, to document findings and communicate them to others for further study. For example, the conventions for epigraphic documentation, used for writing, drawings, and other inscriptions in stone, are based on traditional pen-and-ink drawings. While it offers great clarity, making epigraphic illustrations by hand can be time-consuming and imprecise. The quality and accuracy of the drawing usually depends on the archaeologist's talent, the size of the drawing paper and the object, and the time spent on the drawing. Recently, archaeologists have begun to adapt this painstaking process into the digital realm, by using tools such as Adobe Illustrator to trace over photographs. While still time-consuming (typically taking many hours per illustration), digital epigraphy removes the need to be on-site to execute the drawing, although it still may be necessary to go back to the site to correct perspective and add details that are hard to see in the photograph. Figure 2 shows an example of digital epigraphy of rock inscriptions from the Sinai done manually by archaeologists [Tal12].

In traditional epigraphy, experts only use line drawings, sometimes with varying thickness and color. However, it is tedious and difficult to illustrate small surface details without creating distractions with line drawings. Recently, experts started looking for new ways to illustrate details. Museums and archaeologists have begun using Reflectance Transformation Imaging (RTI) for both visualization and digital preservation purposes [MVSL05]. RTI capture consists of photographing the object under study from a fixed viewpoint, but under varying lighting. Using this collection of photometric data, techniques such as albedo/normal estimation, re-lighting, and material modification may be used to produce enhanced visualizations. On the other hand, RTI is not as effective at conveying large-scale structures at a glance, and hence has not been adopted in publications of record as a replacement for epigraphy.

In this work, we propose a user-guided system to create digital epigraphy with both line drawings and detail illustrations from photometric datasets. We focus on ancient rock inscriptions that are created by either carving into the rock or roughening ("pecking out") the surface. We make use of a dataset of 4000-year-old inscriptions located in a desert, constantly under strong sunlight, making capture challenging. Furthermore, the rock surface is usually very rough and eroded because of weathering, which makes any type of captured data very noisy. The main challenges are to capture these inscriptions in high precision, find the relevant information, and attenuate the effects of noise caused by surface roughness and weathering. We propose a semi-automatic pipeline to solve this problem as follows:

- Capturing photometric datasets in challenging terrain, and estimating surface albedo and normals;
- Rectification of the surface normals to correct perspective;
- Segmentation of the inscriptions from the rock surface and classification based on carving technique (either slightly deeper grooves, or shallow pecked-out regions); and
- Stylization of the inscriptions in various styles.

For grooves, we produce illustrations in a traditional epigraphic line-drawing style, with line thickness optionally modulated by the grooves' orientation to give a sense of relief. For pecked-out regions, we explore shaded and stippled styles, with control over whether the shading depicts global shape or only local detail. Our pipeline offers the user the control needed to produce clear illustrations, and deals with poor signal-to-noise more effectively than existing NPR methods.

2. Related Work

Segmentation. Image segmentation has been one of the oldest and most widely studied problems in the field [Sze10]. Popular approaches include clustering techniques which aggregate pixels into regions with similar contents [JMF99], graph cuts and energy based methods [BK04] [KZ04]. However, the problem can become trickier in some situations, including segmenting the rock inscriptions which we focus on. Less work has been addressed in the context of segmenting on nearly flat surfaces. [ZTS09] has designed a method to extract reliefs and details from relief surfaces based on depth, but it is hard to apply their method to our dataset where some pecked-out inscriptions do not even have noticeable depth change. For robust segmentation on these datasets, we adopt the idea of combining color and normals for segmentation [TFFR07].

(a) Photography (b) Epigraphy

Figure 2: *Photograph of a rock inscription and digital epigraphy done manually.*

2D Line Drawings. Line extraction and drawing on 2D images has been extensively studied. For instance, [KLC07] proposed using anisotropic Difference of Gaussians (DoG) guided by edge tangent flow to detect coherent lines while suppressing noise. [SKLL07] extracted lines by imitating human line drawing process using estimation of a likelihood function with consideration of feature scales and line blurriness. Shape and shading on the image were represented by using tone and strokes by [LXJ12]. [Win11] reviewed the variations of the DoG operator, which was shown to give the most aesthetic results. [XK08] depicted images using only black and white colors by formulating the problem of thresholding as an optimization over a graph of segmentation connectivity. All of those methods solely depend on the color variation in the image, which can be a drawback on rock inscriptions, where the color difference between the inscription and the background is usually small, and the surface roughness results in noise with traditional edge detection algorithms. On the other hand, we propose using both color and surface normal statistics at the same time to detect inscriptions.

3D Line Drawings. Several works have been published on how to represent 3D models with curves. [DFRS03] proposed a new type of view-dependent curves — suggestive contours — which are the loci of points at which occluding contours appear with minimal change in viewpoint. Later, they proposed a new family of lines — suggestive highlights — which are complementary to the suggestive contours [DR07]. [JDA07] introduced apparent ridges, which were defined as the loci of points that maximize a view-dependent curvature. [KST08] proposed another type of

view independent curve: demarcating curves, which are the zeros of the normal curvature in the curvature gradient direction. [KST09] defined relief edges as the zero crossings of the normal curvature in the direction perpendicular to the relief edge. [KST13] extended multi-scale edge detection on images to 3D surfaces by determining the optimal scale for each point on the surface. These methods only use the geometry to find the extrema points, which can fail if the inscriptions are shallow and the surface is rough.

Shape Illustration. There are various non-photorealistic techniques in the literature to depict a shape from 2/2.5/3D images. For instance, [KMI*09] generated a stippling texture on an image from input samples by measuring a texture similarity metric. [TFFR07] investigated various non-photorealistic illustrations on 2.5D images (images with normals). [BMS*10] used Hermite Radial Basis Function Implicits to generate a robust distribution of points on a given 3D surface to position the drawing primitives, which are then rendered to depict shape and tone using silhouettes with hidden-line attenuation, drawing directions, and stippling. [VBGS08] proposed a view-dependent shape descriptor called apparent relief, which combines the convexity of the 3D object and the normal variations in image-space. Several works focused on using varying line/stroke thickness to depict a 3D shape. [SPCP03] extracted and rendered feature edges by varying the thickness based on perceived curvature of the 3D model and [SFWS03] used pen strokes to depict the 3D shape by making them thicker at certain curvatures, junctions and creases. [HS07] mimicked the human drawing process by following the connectivity of the feature edges and rendering them using a ribbon metaphor, where the thickness is determined by the twist of the ribbon. [GVH07] computed the stroke thickness of contours and suggestive contours from depth, radial curvature, and light direction.

The previous works have been tailored to work with only either color or geometry information. However, we believe that the archaeological illustrations such as epigraphy represent not only the geometry of the artifact, but also variations in texture. We can use the photometric data for both geometry and texture information (normal maps and albedo extracted from the photometric data, respectively). Examples of why we need both sources of information are shown in Figure 3. [BSMG05] and [TFFR07] showed how to use photometric data for stylized non-photo realistic renderings. The former produces geometry and tone-aware hatching-like renderings; while the latter explores how to reformulate some previous work, such as suggestive contours and ridges and valley lines, for photometric data. Unfortunately, neither of these produces good results for archaeological illustrations. They are either incomplete or noisy, as we will show in Section 7.

3. Overview

Our main challenge is to produce meaningful (shape-indicating), relevant (including the most representative features) drawings of archaeological inscriptions. Unfortunately, the signal-to-noise ratio is typically low: the inscriptions are shallow, while there is considerable noise caused by surface roughness. We briefly review how our datasets are acquired and rectified (Section 4), then divide the problem into two major components: finding and classifying the inscriptions with user-guided segmentation (Section 5), and then rendering them in different geometry-aware non-photorealistic styles (Section 6). We present results on a number of datasets, and argue that this type of stylization is difficult to achieve with previous methods (Section 7).

4. Data Acquisition and Preprocessing

We begin with datasets captured at an ancient amethyst mining site in an area of Upper Egypt called Wadi el Hudi, which was documented in 1952 by [Fak52]. The terrain consists of several hills covered with granite rocks. The inscriptions are mostly located on the ridges and their position and surroundings make them hard to capture. The ridges are steep and the rocks on the ridges are usually loose, which makes it challenging to set up an acquisition system on the ground. Because of the environmental conditions, the recording must be done during the day under strong sunlight. Also, carrying heavy equipment or power sources to the site is impractical, since it takes a short hike to reach some of the inscriptions. However, the resolution of the acquisition system should still be high, because the inscriptions are mostly on flat surfaces with shallow depth. For these reasons, the most practical setup is one based on a camera and flash, in preference to 3D acquisition systems such as structured-light scanners, multi-view stereo, or a laser scanner.

The datasets are captured using a photometric acquisition setup similar to the one used by Toler-Franklin et al. [TFFR07], as shown at left. Specifically, a series of digital-SLR images of the object are captured with a fixed camera position, and a hand-held flash is moved around the object to obtain different light directions. A black cloth is used to shade the object and the camera. Several mirror and white diffuse spheres, which will be used to estimate the light directions and intensities in each image later, are placed next to the object. Also, one image with no flash is taken at the beginning of each capture session, which will be subtracted from all other images with flash to remove ambient lighting as much as possible. After the photometric datasets are captured, we use a variant of the photometric stereo algorithm [Woo80] to estimate surface normals and the true albedo from those images: example re-

Figure 3: ***Left column:*** *Color images of some rock inscriptions;* ***Right column:*** *Normal maps.* ***Top row:*** *While the inscriptions are visible on the color image, some of them are not recognizable on the normal map.* ***Bottom row:*** *Normal map reveals more information than the color image for some other cases.*

sults are shown in Figure 3. In this paper, we show results from around 10 representative datasets.

4.1. Rectification

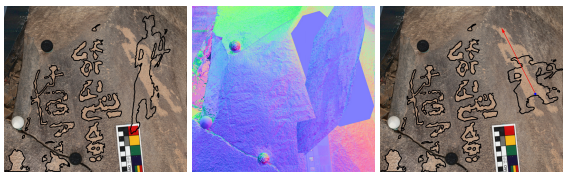
One of the important conventions of documenting rock drawings is that they need to be recorded from a perpendicular view to the surface. However, it is not always possible to place the camera perpendicularly to the surface because of the position and height of the object or the steepness of the terrain around it. Another problem is that the inscriptions sometimes start from one face of the rock and continue on the other. Recording them from an angle will make them look skewed, and this is not desirable for documentation. Archaeologists usually warp the image to straighten the parts of the image by either eyeballing to flatten these regions, or if they recorded some 3D points via some terrain mapping systems such as total station, they manually select the control points on the image corresponding to those 3D points and rectify the image. However, it is time-consuming to both capture those 3D points and rectify the image based on those points.

In our system, we allow the user to rectify the image without any extra data. The user can select a nearly planar region R, by selecting a few points around the region to create a closed polygon. Given this selection, we automatically rectify the region as follows:

$$\vec{n}_{mean} = \texttt{normalize}\Big(\sum_{(x,y)\in R} N(x,y)\Big) \quad (1)$$

$$\vec{r} = \vec{n}_{mean} \times < 0,0,1 > \quad (2)$$

$$\alpha = \texttt{acos}(\vec{n}_{mean} \cdot < 0,0,1 >) \quad (3)$$



(a) (b) (c)

Figure 4: ***(a), (b)*** *Rectified inscription and normal map where the planar region to be rectified is defined by user.* ***(c)*** *User defined rotation axis (marked with red arrow) and flattened inscription.*

where N is the normal map, $\vec{r}$ is the rotation axis, and α is the rotation angle. Once the rotation axis and angle are computed, the region is rotated around $\vec{r}$ and all the normal values are rotated accordingly as shown in Figure 4.a and b. This approach assumes that the region is nearly planar which is the case for most of the datasets. We also allow the user to draw a rotation axis manually by drawing a line on the image and use this axis instead to rotate the image around, which requires two rotation operations: first we rotate the image so that the average normal along the user-defined rotation axis would be equal to the z-direction and then we rotate the image around this fixed rotation axis to make the region flat as shown in Figure 4.c. The user also can control scaling of the preselected region after flattening, as done in Figure 4.

5. Segmentation

We focus on rock inscriptions, which can be classified into two types: carved grooves, and "pecked out" areas on the rock surface.[†] Their characteristic roughness and incompleteness caused by erosion make the creation of epigraphy very difficult. Existing NPR techniques are insufficient to produce nice and understandable illustrations for such rock inscriptions. We approach the problem by first performing a robust segmentation to create masks for the two types of inscriptions, and then stylize them separately. An implementation pipeline for this segmentation part is described in the following subsection, with details described below.

5.1. Implementation Pipeline

Preprocessing includes downsampling, histogram equalization, and bilateral filtering to reduce noise and increase contrast between the foreground inscriptions and the background rock surface. Figures 5.a and 5.b show a color image before and after the preprocessing, respectively.

[†] Pecking-out is a term used to describe areas where the rock surface has been roughened to form a design or figure.

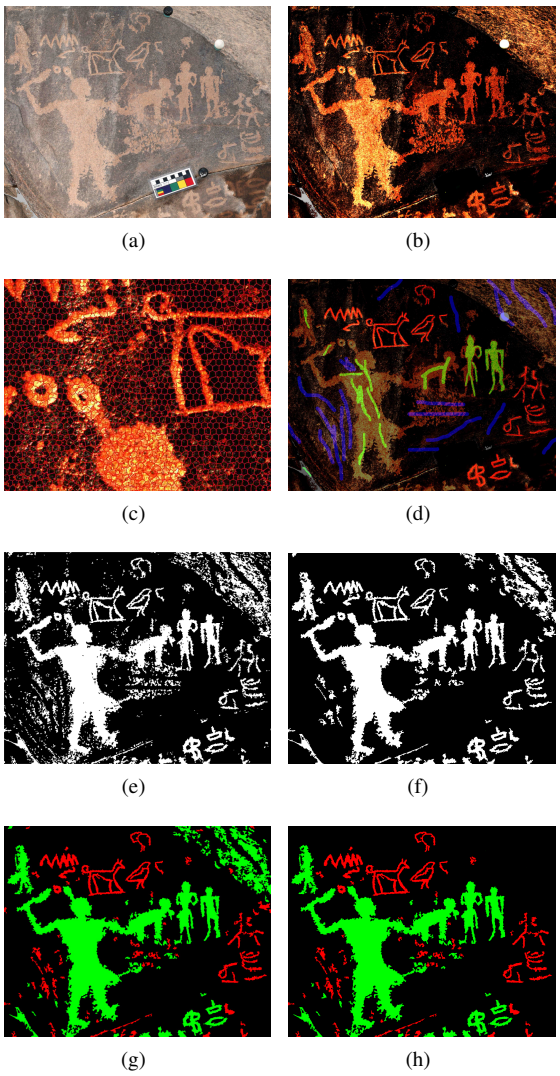


Figure 5: *(a) Original color image. (b) Color image after preprocessing. (c) A close up look at the superpixel over-segmentation. (d) User input strokes for labeling. (e) Graph-cut foreground segmentation result. (f) Foreground segmentation result after post-refinement. (g) Foregrounds classification result. (h) Foreground mask for stylization after user cleanup.*

Over-segmenting the color image can help to enforce local consistency, thus better maintaining the boundaries of the foreground inscriptions. We apply a clustering-based algorithm [ASS*12] to aggregate pixels into superpixels and create an over-segmentation of the image: Figure 5.c shows an example.

Feature descriptors are extracted from both colors and normals. Color-based features are based on statistics over each superpixel. These are augmented by information from normal maps that helps improve descriptiveness, especially when some inscriptions are difficult to see in the color image. The normals also help with distinguishing between carved grooves and pecked-out areas. Details are explained in section 5.3.

In the **labeling** step we ask the user to draw sparse strokes to indicate pixels belonging to 3 different classes. As shown in Figure 5.d, red strokes refer to foreground grooves, green strokes refer to foreground pecked-out areas, and blue strokes mark the background.

A **graph-cut** approach [BVZ01, KZ04, BK04] is used to group the pixels into foreground and background. Energy functions are designed based on feature descriptors and user labeling as explained in section 5.3.

Post-refinement is applied to the graph-cut segmentation output to enhance foreground connectivity and remove noise on the background. Figure 5.e and Figure 5.f give an example of the segmentation result before and after the post-refinement processing.

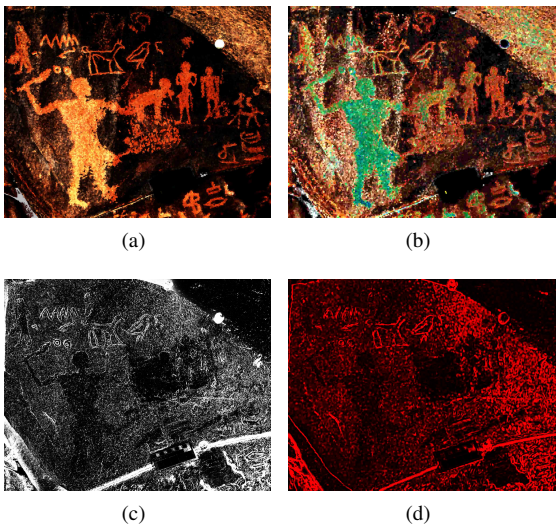
Foreground classification is done after foreground/background segmentation to produce masks for the two types of rock inscriptions. A Naive Bayes classifier is trained based on the user input labels. Figure 5.g shows an example classification result. We also allow the user to edit the label of each connected component, so that they can fix mislabeled components or remove irrelevant parts from the foreground, as illustrated in Figure 5.h.

5.2. Feature Design

We compute a feature descriptor for each pixel using cues from both color and normal images.

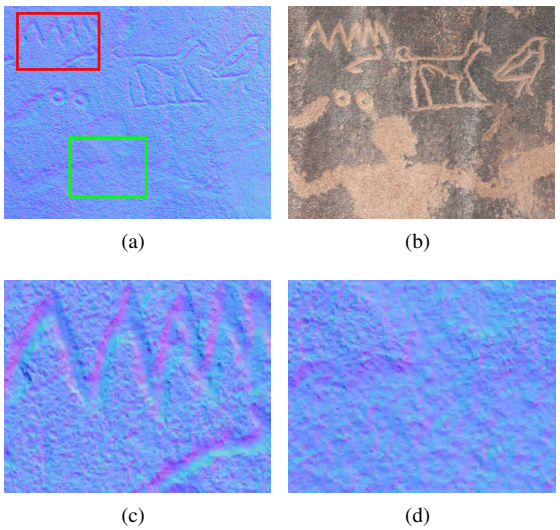
Color-based features. Color images of rock carvings are noisy, and just using the color of each pixel as a feature yields results that are not robust. Instead, we first over-segment the color image into near-equal-sized superpixels. As shown in Figure 5.c, these superpixels preserve the boundaries in the original image while capturing redundancy in the data, and they also help to reduce computational cost by allowing features to be computed only once per superpixel (instead of per pixel). Means and standard deviations are computed for each RGB channel within the neighborhood defined by each superpixel. Pixels from the same superpixel are assigned the same color feature. Figure 6.a and Figure 6.b visualize the color feature for the image shown in Figure 5.a.

Normal based features. By only looking at the color image, it is hard to distinguish the carved grooves from the pecked-out regions. However, these two types of inscriptions have recognizably different normal statistics, as illustrated in Figure 7. Therefore, it is reasonable to compute features from normals in addition to color. We take



(a) (b) (c) (d)

Figure 6: ***(a)*** *Visualization of color mean within each superpixel.* ***(b)*** *Visualization of color standard deviations within each superpixel.* ***(c)*** *Visualization of the approximated surface gradient magnitude.* ***(d)*** *Visualization of the bigger eigenvalues of the structured tensors. (The contrast of the images are increased for better visualization.)*



(a) (b) (c) (d)

Figure 7: ***(a)*** *Normals of a region with 2 types of inscriptions.* ***(b)*** *Color image of the same region with 2 types of inscriptions.* ***(c)*** *A close up look at the carved in region (in red rectangle).* ***(d)*** *A close up look at the pecked out region (in green rectangle).*

the x and y components from the normal image to approximate surface gradient, and then compute the gradient magnitude. We also compute the covariance matrix in a Gaussian weighted square neighborhood for each pixel, and add the bigger eigenvalue as a new entry to the feature descriptor: this helps to distinguish grooves (which will have one big eigenvalue) from roughened regions (which will have two small eigenvalues). As visualized in Figure 6, normal based features help to capture edges on the surface.

5.3. Inscription Segmentation

In order to produce labeled segmentation for both grooves and pecked areas, we first run a graph-cut based binary segmentation to obtain foreground and background, and then train a classifier on the foreground segment to distinguish between the two types of inscriptions. Such a two-fold process is more efficient and more reliable, compared to multi-label graph-cut segmentation.

Training. We ask the user to label the two types of foreground inscriptions and the background by drawing strokes in different colors. Let $\mathcal{L} = \mathcal{L}_0 \cup \mathcal{L}_1$ represent the set of labeled pixels, where $\mathcal{L}_0$ denotes the background labeled pixels and $\mathcal{L}_1 = \mathcal{L}_{10} \cup \mathcal{L}_{11}$ denotes the two types of foreground labeled pixels. For labeled pixels from $\mathcal{L}_0$ and $\mathcal{L}_1$, we run hierarchical clustering [ML12] to aggregate them into clusters in feature space. Let $f^s_{\alpha j}$ denote the centroid of the j-th cluster with label $\alpha \in \{0,1\}$. We call f^s_{0j} background seeds and f^s_{1j} foreground seeds.

Energy function. We define a graph on the image in which each pixel is a node and each node has 4 edges connecting its 4-neighbors. A binary graph-cut segmentation algorithm is applied to find the least-cost boundaries that smoothly partition the graph into foreground and background by minimizing an energy function $E(\alpha, f)$, where $\alpha = (\alpha_1, \alpha_2, ..., \alpha_n)$ is the label assignment, $f = (f_1, f_2, ..., f_n)$, is the vector of feature descriptors, and n denotes the number of pixels. We choose to apply the graph-cut segmentation on pixels instead of superpixels because the normal features are extracted from per-pixel neighborhoods. The energy function consists of two terms:

$$E(\alpha, f) = U(\alpha, f) + V(\alpha, f), \tag{4}$$

where $U(\alpha, f)$ denotes the data term and $V(\alpha, f)$ denotes the smoothness term.

We define the data term as

$$U(\alpha, f) = \sum_{p \notin \mathcal{L}} -\log \frac{\min_j \left\| f_p - f^s_{\alpha_p j} \right\|}{\sum_\alpha \min_j \left\| f_p - f^s_{\alpha j} \right\|} + \sum_{p \in \mathcal{L}} -\log \varepsilon, \tag{5}$$

where $\|\cdot\|$ denotes Euclidean distance and $\varepsilon > 0$ is a user defined small constant.

The smoothness term is defined as

$$V(\alpha, f) = \gamma \sum_{(p,q)\in\mathcal{N}} I(\alpha_p \neq \alpha_q) e^{-\beta(f_p - f_q)^2}, \qquad (6)$$

where γ and β are user-defined positive weighting parameters. $\mathcal{N}$ is the set of neighboring pixel pairs in the grid graph. $I(\cdot)$ is an indicator function that is 1 if $\alpha_p \neq \alpha_q$, 0 otherwise.

Cleanup. After the graph-cut segmentation, we perform a post-refinement process to enhance foreground connectivity and remove speckles. For each superpixel, we assign the same label to all pixels within it by taking a majority vote on pixel labels from the graph-cut output. We then remove noise in the background and fill small holes in the foreground by removing small connected components.

Label classification. Finally, we train a Naive Bayes classifier [Fuk90] based on feature descriptors of labeled foreground pixels from $\mathcal{L}_1 = \mathcal{L}_{10} \cup \mathcal{L}_{11}$, and predict the type to be either a carved-in or pecked-out inscription. We then assign pixels within each connected component with the same label by conducting a majority vote, and create a labeled segmentation mask for stylization.

Iterative refinement. In order to give the user fine control over segmentation, we implement an iterative scheme that allows the user to add strokes to refine the segmentation. It takes $3-4$ iterations on average for an experienced user to create a decent segmentation result. An example is shown in Figure 8.

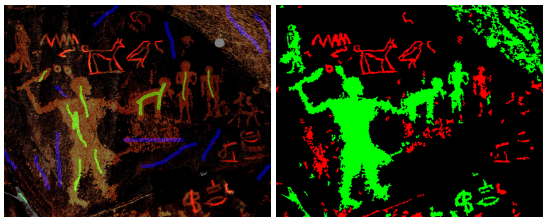
5.4. Evaluation and Comparisons

Graph-cut on color. We show the results of applying graph-cut segmentation on only the color image. Figure 9 shows that color alone is not enough to capture the real foreground inscriptions, especially the grooves, and therefore it is necessary to have more robust features based on normals as well.

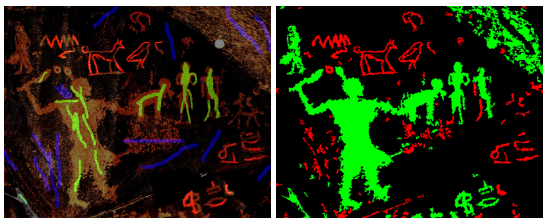
K-means clustering. We compare our foreground/background segmentation result to that of performing k-means clustering on pixels in feature space. Figure 10 shows side-by-side comparisons between our results and the k-means results. We notice that simple clustering yields more noise and incomplete foreground.

We also compare our final labeled segmentation result to aggregating the pixels into 3 clusters by k-means, as shown in Figure 11. On the basis of this, we conclude that a supervised approach yields more reliable results for segmenting rock inscriptions.

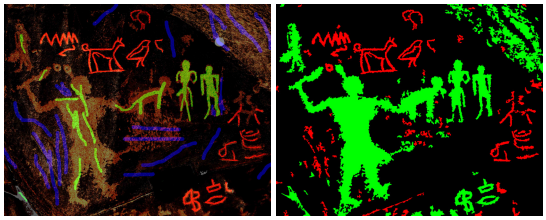
Multi-label graph-cuts. We show result of directly performing multi-label graph-cut segmentation and compare it to foreground/background segmentation followed by foreround classification. The computation time of the multi-label segmentation is roughly 50% higher than our method. As shown in Figure 12, our two-stage process produces more reliable results.



(a) Iteration 1

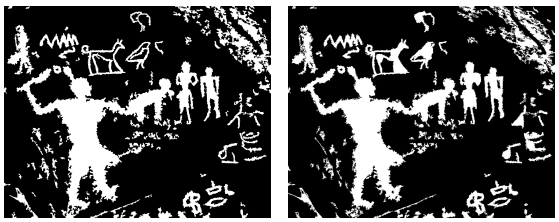


(b) Iteration 2



(c) Iteration 3

Figure 8: ***Left column:*** *Labels drawn by the user.* ***Right column:*** *Segmentation result.*

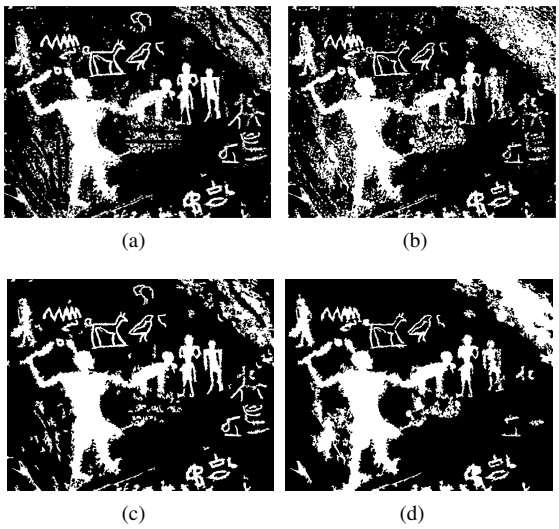


(a) (b)

Figure 9: ***(a)*** *Our foreground/background segmentation results.* ***(b)*** *Foreground/background segmentation results of graph-cut on color.*

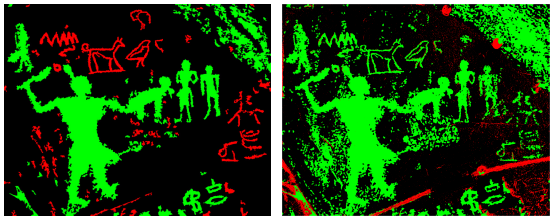
6. Stylization

Our system uses the labeled segmentations to drive stylized rendering. We describe two categories of stylization: *shape illustration* that depicts the depth of grooves or the general shape of the whole rock surface, and *detail illustration* to depict surface details in the pecked-out regions.



(a) (b) (c) (d)

Figure 10: ***(a)*** *Our segmentation result before cleanup.* ***(b)*** *K-means clustering result before cleanup.* ***(c)*** *Our segmentation result after cleanup.* ***(d)*** *K-means clustering result after cleanup.*



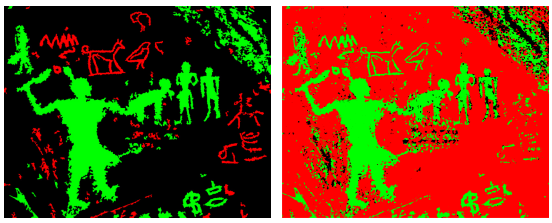
(a) (b)

Figure 11: ***(a)*** *Our final labeled segmentation result.* ***(b)*** *K-means 3 clustering result.*

6.1. Shape Illustration

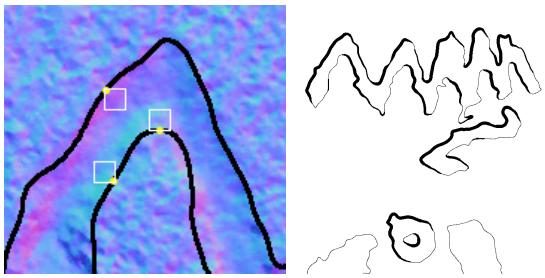
We first extract contours from the segmentation masks and convert each into a discrete curve by sub-pixel sampling the contour. Then, the user can manipulate the contours by controlling parameters for Gaussian smoothing, bilateral filtering, and unsharpening along the contour. Each filter can be applied to change either the contours' shape or their properties such as normals, curvatures, and colors. The user can also add a sense of relief by varying stroke thickness, or provide an impression of overall shape using stippling.

Relief. Some of the inscriptions are carved into the rock (grooves) which makes them either sunk or bas-relief depending on the carving technique. One way of representing reliefs with line drawings is to draw them with different line thicknesses. One of most common conventions used for reliefs in digital epigraphy is to assume a light source at a



(a) (b)

Figure 12: ***(a)*** *Our final labeled segmentation result.* ***(b)*** *3-label graph-cut result.*



(a) (b)

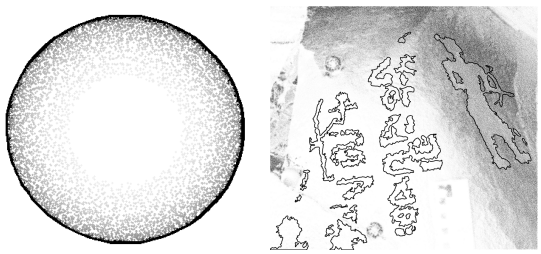
Figure 13: ***(a)*** *Thickness calculation.* ***(b)*** *Example of a relief illustration.*

45°angle from the top-left side of the image. The goal is to use line thickness to indicate the shading of the surface under this hypothetical lighting. In digital epigraphy, this is done manually by offsetting the mask for the inscriptions. However, this method only gives us general information about whether it is sunk relief or bas relief. For a more precise depiction, we use the depth of the inscription to determine the line thicknesses.

To determine the depth of the inscription without actually integrating the normal map, we take a small window, W, from each curve point in the opposite direction of the curve normal as shown in Figure 13.a and take the average normal within that window. Later, we could just use the dot product of the light direction with the average normal to determine the line thickness. However since the surface is not necessarily perpendicular to the camera and we want to get rid of the noise caused by the rough surface, we first apply a band-pass filter (Difference of Gaussians) on the normal map, N:

$$N_{DoG} = N * G(\sigma_1) - N * G(\sigma_2) \tag{7}$$

where $G(.)$ are 2D Gaussian functions with σ_1 removing the noise caused by the surface roughness and σ_2 removing the low frequency component of the normal map, which represents the rough shape of the surface. We later use the DoG to compute the average direction for each point on the curve,



(a) (b)

Figure 14: ***(a)*** *Stippling on a sphere.* ***(b)*** *Stippling to depict the rough shape of a curved boulder.*

and the thickness is set proportional to the dot product of the average DoG and the imaginary light direction.

$$t(i) =< 1,-1,0 > \cdot \Big(\frac{1}{|W_i|} \sum_{(x,y)\in W_i} N_{DoG}(x,y) \Big) \tag{8}$$

where $t(i)$ is the computed line thickness at point i. Later, the user can further smooth the thickness along the curve to have a smoother transition along the contour as described below:

$$t(i)_{smooth} = \frac{1}{g_{total}} \sum_{j=-r}^{r} g(j,r)t(i+j) \tag{9}$$

where $g(j,r)$ is a Gaussian function in which the variance of the function is calculated from the maximum range r. In Figure 13.b, an example is shown where the relief illustration is applied to the grooves while the pecked out regions are rendered with a constant line thickness.

Stippling. To depict the rough shape of the surface, we use stippling derived from the normal map. We compute the stippling density in small grids by taking the dot product of the z direction with the average normal within the grid, W, as follows:

$$d = 1 - \Bigg(< 0,0,1 > \cdot \Big(\frac{1}{|W|} \sum_{(x,y)\in W} N(x,y) \Big) \Bigg) \tag{10}$$

A number of stippling points proportional to the density are randomly rendered within the small grid with a color inversely proportionally to the density. When the surface is tilted away from the camera, the stippling density will be close to one and the color will be darker, and vice versa, as shown for a sphere in Figure 14.a. This method can be used over all the image when the rock surface is curved to indicate the general shape of the surface as shown in Figure 14.b.

6.2. Detail Illustration

To illustrate the surface details, we compute the dot product of the surface normals with either the z-direction or the local average normal as follows:

$$I(x,y) = \begin{cases} d = N(x,y) \cdot B(x,y) & \text{if } d < \tau \\ 1 & \text{otherwise.} \end{cases} \tag{11}$$

where τ is a user defined threshold, which controls the elimination of the flatter features and $B(x,y)$ is either $< 0,0,1 >$ or the low frequency component of the normal map, $N_g(x,y)$, computed by Gaussian filtering the normal map. When the z-vector is used, it will depict the general shape of the surface such as curviness; on the other hand, if $N_g(x,y)$ is used, it will reveal the local surface details only. If the surface is flat and perpendicular to the camera, both methods will give similar results. In our results, extracted details are applied to the pecked out regions as a texture.

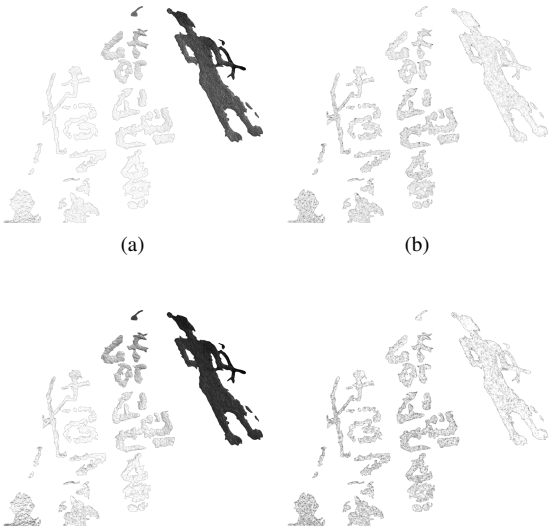
6.3. User Control

The user can control all the sigma and threshold values, and can use either the albedo, gray tones, or black and white for details. The stippling method described in the previous section can also be used for detail illustration by using $N_g(x,y)$ instead of the z-direction. Different methods for detail illustration are shown in Figure 15.

7. Results and Discussion

In this section, we show results on several inscription and non-inscription datasets, as well as comparisons to some previous work. Figure 16 demonstrates several styles of our epigraphy pipeline on one dataset. We demonstrate generating outlines of the inscriptions, adding surface details, giving relief effect to the grooves, stippling for shape depiction, and stippling to illustrate the surface details respectively. For comparison, Figure 17 shows a digital epigraphy done manually by archaeologists in two styles. As they reported, it took 5-6 hours to trace over the image and another 1-2 hours to add texture details to the pecked out regions. The main reason of the manual epigraphy taking such a long time is the desire for great precision rather than lack of experience.

In Figures 18 and 19, several previous works are compared. Results from image-based line drawings are shown in Figure 18, where we compare our result to Coherent Line Drawings (CLD) by [KLC07] and extended Difference of Gaussians (xDoG) by [Win11]. Even though xDoG handles the noise caused by surface roughness better than CLD, it fails when the variation in color is small, such as in the right column of the figure. To test 3D line drawing methods, we project the surface normals onto a 3D plane and use those projected normals to compute 3D contours. As seen in Figure 19, previous works cannot handle the noise well,



(a) (b) (c) (d)

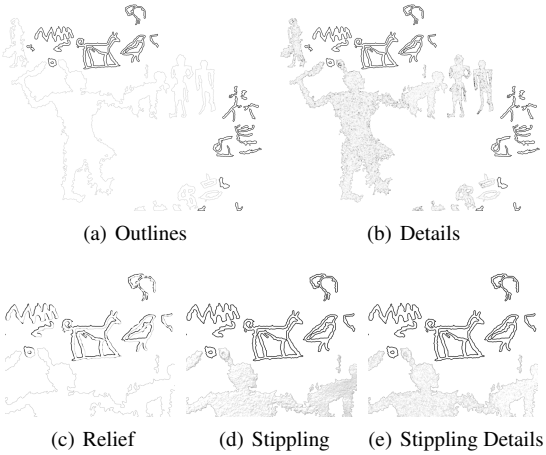
Figure 15: *Detail illustrations for pecked out regions.* ***Top row:*** *details computed using Equation 11;* ***Bottom row:*** *random stippling;* ***Left column:*** $B =< 0,0,1 >$*;* ***Right column:*** $B = N_g(x,y)$.

and even when the surface is smoother it does not produce contours enclosing the inscriptions as needed for epigraphy. Given that the previous works use only 2D color or 3D geometry information, the comparison to our approach could be potentially unfair. However, it also shows the necessity of using both color and geometry information.

Results on several other rock inscriptions are shown in Figure 20, where the last two datasets are deeply engraved stones different than other rock inscriptions. We also demonstrate our system's usability on non-inscription datasets in Figure 21.

8. Conclusion

In this paper, we presented a user-guided system to produce digital epigraphy from photometric datasets. The pipeline consists of segmenting the inscriptions from the rock surface and labeling them based on their carving techniques and rendering them in various non-photorealistic styles. We stated that our system efficiently and quickly (in less than 10 minutes) produces results comparable to the manual epigraphy done manually by archaeologists in hours. We compared our results to several previous works on both 2D and 3D line drawing algorithms. Finally, we showed results for several rock inscriptions, rock engravings, and non-inscription objects.



(a) Outlines (b) Details (c) Relief (d) Stippling (e) Stippling Details

Figure 16: ***(a)*** *Outlines of the inscriptions computed by the segmentation algorithm, grooves are drawn with thicker lines than pecked out regions.* ***(b)*** *Pecked out regions are textured by adding surface details using Equation 11.* ***(c), (d), and (e)*** *are closeup images showing relief, random stippling for shape depiction and stippling for detail illustration, respectively.*

Figure 17: *Digital epigraphy done manually by archaeologists using Adobe Photoshop and Illustrator.*

Acknowledgments

We would like to thank our archeologist collaborators Kate Liszka and Bryan Kraemer for their valuable ideas and providing manual digital epigraphy. This work is partially supported by NSF grants IIS-1012147 and IIS-1421435.

References

[ASS*12] ACHANTA R., SHAJI A., SMITH K., LUCCHI A., FUA P., SÜSSTRUNK S.: SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Trans. PAMI 34*, 11 (Nov 2012), 2274–2282. 5

[BK04] BOYKOV Y., KOLMOGOROV V.: An experimental comparison of min-cut/max- flow algorithms for energy minimization in vision. *IEEE Trans. PAMI 26*, 9 (Sept 2004), 1124–1137. 2, 5

[BMS*10] BRAZIL E. V., MACÊDO I., SOUSA M. C., VELHO L., DE FIGUEIREDO L. H.: A few good samples: Shape &

Figure 20: *Results on several other datasets.*

tone depiction for hermite RBF implicits. In *Proc. NPAR* (2010), pp. 7–15. 3

[BSMG05] BARTESAGHI A., SAPIRO G., MALZBENDER T., GELB D.: Three-dimensional shape rendering from multiple images. *Graph. Models 67*, 4 (July 2005), 332–346. 3

[BVZ01] BOYKOV Y., VEKSLER O., ZABIH R.: Fast approximate energy minimization via graph cuts. *IEEE Trans. PAMI 23*, 11 (Nov 2001), 1222–1239. 5

[DFRS03] DECARLO D., FINKELSTEIN A., RUSINKIEWICZ S., SANTELLA A.: Suggestive contours for conveying shape. In *Proc. ACM SIGGRAPH* (2003), pp. 848–855. 2

[DR07] DECARLO D., RUSINKIEWICZ S.: Highlight lines for conveying shape. In *Proc. NPAR* (Aug. 2007). 2, 13

[Fak52] FAKHRY A.: *The inscriptions of the amethyst quarries at Wadi el Hudi*. Service des antiquités de l'Égypte. Government Press, 1952. 3

[Fuk90] FUKUNAGA K.: *Introduction to Statistical Pattern Recognition (2nd Ed.)*. Academic Press, 1990. 7

[GVH07] GOODWIN T., VOLLICK I., HERTZMANN A.: Isophote distance: A shading approach to artistic stroke thickness. In *Proc. NPAR* (2007), pp. 53–62. 3

[HS07] HOUSE D. H., SINGH M.: Line drawing as a dynamic process. In *Proc. Pacific Graphics* (2007), pp. 351–360. 3

[IFP95] INTERRANTE V., FUCHS H., PIZER S.: Enhancing transparent skin surfaces with ridge and valley lines. In *Proc. IEEE Visualization* (Oct 1995), pp. 52–59, 438. 13

[JDA07] JUDD T., DURAND F., ADELSON E.: Apparent ridges for line drawing. *ACM Trans. Graph. 26*, 3 (July 2007). 2, 13

[JMF99] JAIN A. K., MURTY M. N., FLYNN P. J.: Data clustering: a review. *ACM computing surveys (CSUR) 31*, 3 (1999), 264–323. 2

[KLC07] KANG H., LEE S., CHUI C. K.: Coherent line drawing. In *Proc. NPAR* (2007), pp. 43–50. 2, 9, 13

[KMI*09] KIM S. Y., MACIEJEWSKI R., ISENBERG T., ANDREWS W. M., CHEN W., SOUSA M. C., EBERT D. S.: Stippling by example. In *Proc. NPAR* (2009), pp. 41–50. 3

[KST08] KOLOMENKIN M., SHIMSHONI I., TAL A.: Demarcating curves for shape illustration. *ACM Trans. Graph. 27*, 5 (Dec. 2008), 157:1–157:9. 2

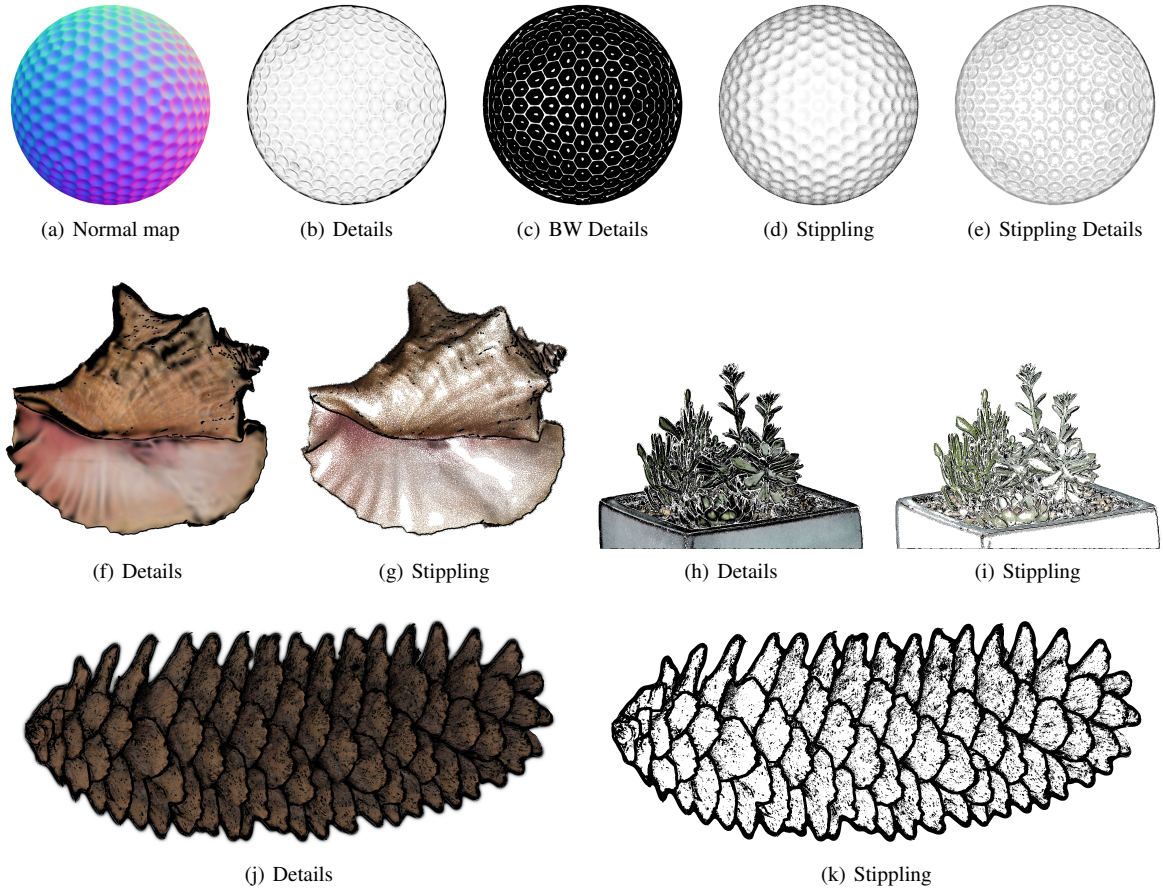
[KST09] KOLOMENKIN M., SHIMSHONI I., TAL A.: On edge detection on surfaces. In *Proc. CVPR* (June 2009), pp. 2767–2774. 3, 13

[KST13] KOLOMENKIN M., SHIMSHONI I., TAL A.: Multi-scale curve detection on surfaces. In *Proc. CVPR* (2013), pp. 225–232. 3

[KZ04] KOLMOGOROV V., ZABIN R.: What energy functions can be minimized via graph cuts? *IEEE Trans. PAMI 26*, 2 (Feb 2004), 147–159. 2, 5

[LXJ12] LU C., XU L., JIA J.: Combining sketch and tone for pencil drawing production. In *Proc. NPAR* (2012), pp. 65–73. 2

[ML12] MUJA M., LOWE D. G.: Fast matching of binary features. In *Computer and Robot Vision (CRV)* (2012), pp. 404–410. 6



(a) Normal map (b) Details (c) BW Details (d) Stippling (e) Stippling Details

(f) Details (g) Stippling (h) Details (i) Stippling

(j) Details (k) Stippling

Figure 21: *Several non-inscription objects are used with our system where we use the depth discontinuity to find the contours.*

[MVSL05] MUDGE M., VOUTAZ J.-P., SCHROER C., LUM M.: Reflection transformation imaging and virtual representations of coins from the Hospice of the Grand St. Bernard. In *Proc. VAST* (2005), pp. 29–39. 2

[SFWS03] SOUSA M. C., FOSTER K., WYVILL B., SAMAVATI F.: Precise ink drawing of 3D models. *Computer Graphics Forum 22*, 3 (2003), 369–379. 3

[SKLL07] SON M., KANG H., LEE Y., LEE S.: Abstract line drawings from 2D images. In *Proc. Pacific Graphics* (2007), pp. 333–342. 2

[SPCP03] SOUSA M. C., PRUSINKIEWICZ P., COSTA M., PRUSINKIEWICZ S. P.: A few good lines: Suggestive drawing of 3D models. *Computer Graphics Forum (Proc. Eurographics 22* (2003), 2003. 3

[Sze10] SZELISKI R.: *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010. 2

[Tal12] TALLET P.: *La zone minière pharaonique du Sud-Sinaï*. Institut français d'archèologie orientale du Caire - IFAO, 2012. 1

[TFFR07] TOLER-FRANKLIN C., FINKELSTEIN A., RUSINKIEWICZ S.: Illustration of complex real-world objects using images with normals. In *Proc. NPAR* (2007), pp. 111–119. 2, 3

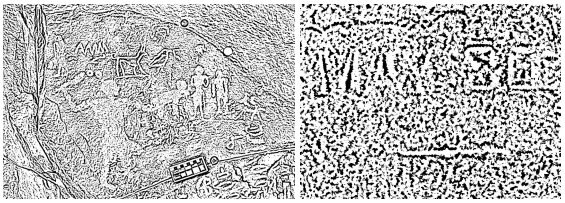
[VBGS08] VERGNE R., BARLA P., GRANIER X., SCHLICK C.: Apparent relief: A shape descriptor for stylized shading. In *Proc. NPAR* (2008), pp. 23–29. 3

[Win11] WINNEMÖLLER H.: Xdog: Advanced image stylization with extended difference-of-gaussians. In *Proc. NPAR* (2011), pp. 147–156. 2, 9, 13

[Woo80] WOODHAM R. J.: Photometric method for determining surface orientation from multiple images. *Optical Engineering 19*, 1 (1980), 139–144. 3

[XK08] XU J., KAPLAN C. S.: Artistic thresholding. In *Proc. NPAR* (2008), pp. 39–47. 2

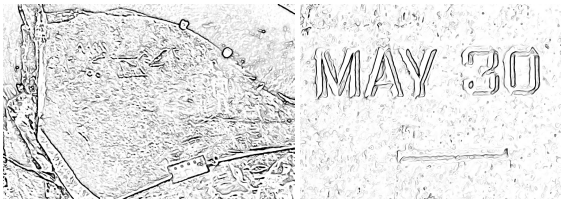
[ZTS09] ZATZARINNI R., TAL A., SHAMIR A.: Relief analysis and extraction. In *ACM SIGGRAPH Asia* (2009), pp. 136:1–136:9. 2



(a) CLD on Albedo [KLC07]

(b) CLD on Normals [KLC07]

(c) xDoG on Albedo [Win11]

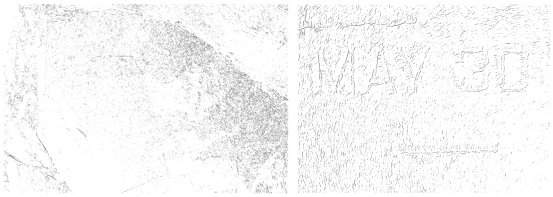


(d) xDoG on Normals [Win11]

(e) Our Results

Figure 18: *Comparisons to previous image-based line drawing techniques.*

(a) Ridges & Valleys [IFP95]



(b) Suggestive Contours & Highlights [DR07]

(c) Apparent Ridges [JDA07]

(d) Relief Edges [KST09]

(e) Isophotes

Figure 19: *Comparisons to previous 3D model-based line drawing techniques.*