Vectorizing Hand-Drawn Vessel Profiles with Active Contours

Hörr, C.[1], Kienel, E. [1],Brunnett, G. [1]

[1] Faculty of Computer Science, Chemnitz University of Technology, Germany

{hoerr, kie, brunnett}@cs.tu-chemnitz.de

Vectorization of hand-drawn images is a very common problem in archaeology, but so far, there are almost no alternatives to precisely tracing the lines of interest manually with commercial vector graphics products. In this contribution, we use the well-known concept of active contours to convert digitized vessel profiles into a closed polyline, which is processable by previously developed feature extraction algorithms. While traditional implementations are either unacceptably slow or too restrictive in use due to predetermined parameters, we present several improvements for keeping the software reactive in real-time and intuitively operable even by inexperienced users. It turns out that the approach is extremely robust, so that up to 40 profile drawings could be vectorized per hour, what largely extended our database for automated classification issues. We were even able to quickly vectorize excavation maps drawn with pencil on millimetre paper, which poses massive problems to most other image segmentation methods.

Keywords: vectorization, active contours, hand drawings.

1. Introduction

For several years, non-contact surface scanners and computerized tomography have enriched, if not revolutionized the documentation of archaeological finds. Nowadays, although still quite expensive, their use has become increasingly fashionable, however it is sometimes forgotten, that the 3D models themselves are scientifically useful only to a limited degree. Of course, many impressive digitization projects have been conducted in the past, including applications of photorealistic and schematic visualization, damage mapping, and reconstruction, but most of them were restricted to prestigious sites and objects, and due to their exceptionality they often lack of a reusable methodology.

In contrast, comparably few attempts have been made to process masses of everyday finds such as pottery fragments or silices, even though great benefits could be expected in this field of archaeological work. (KARASIK and SMILANSKY, 2008) automatically estimate the rotational axis of range scanned rim sherds in order to produce stylized images that imitate the traditional hand drawings. A more general approach to automated documentation has been presented by (HÖRR *et al.,* 2010), where non-photorealistic rendering techniques are applied to artefacts of arbitrary kind.

A logical continuation of 3D scanning and documentation are the tasks of shape analysis, feature extraction, and finally automated classification. Concerning pottery typologies, some promising experiments have been carried out by research groups from Israel (GILBOA *et al.,* 2004; ADAN-BAYEWITZ *et al.,* 2009), TU Vienna (KAMPEL and SABLATNIG, 2007), and ourselves (HÖRR *et al.,* 2008a; 2008b). Good results have also been obtained on handaxe data (GROSMAN *et al.,* 2008; IOVIŢĂ, 2009).

In spite of these recent achievements in automated geometry processing, high acquisition costs for 3D scanning systems as well as the lack of skilled operation personnel are still insurmountable obstacles for many, especially minor archaeological institutions. Even more, already drawn and printed finds cannot be expected to be processed again. Hence, these assemblages will remain excluded from the chances of automated classification, and in turn, large amounts of data will remain unconsidered for the computerized analyses. This gives rise to the development of a cheap alternative to 3D scanners, which would allow for also converting manual drawings into a form that is processable by the feature extraction algorithms mentioned above. In other words, we are seeking for a tool or method that vectorizes the relevant parts of scanned images.

The main problem when processing digitized hand-drawn sketches is posed by the diversity of drawing conventions. Even the rather homogeneous domain of ceramic vessels comprises a large variety of styles, which is one of the major reasons for the difficulty of image-based comparison in archaeology. In this contribution, we focus just on the vessel profiles, since they contain almost all information necessary for morphological analysis. At least for profile sections there are three common drawing styles (fig. 1): outlined, filled with solid colour, and filled with a hatching or stippling pattern. While the first one is cheap to print and harmonizes well with possible auxiliary lines, the second one appears more plastic to the viewer. Hatched profiles are used rather seldom, because they are laborious in preparation, but in some sense they are a compromise between the former two.

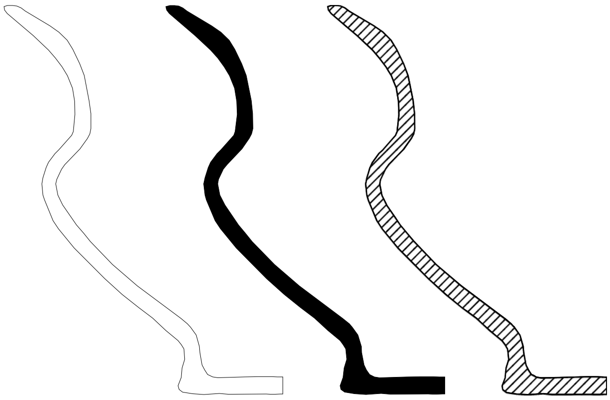


Figure 1: *Examples of different drawing styles.*

Our basic goal is the extraction of a closed vectorized curve from the rasterized image representing the profile and separating it from other image elements. During the past decades, a large number of image segmentation algorithms have been proposed, but in the view of the complexity and quality of human drawings it is impossible to develop a fully-automatic tool. Instead, we suggest using so-called **active contours**, which according to the pioneer work of (KASS *et al.*, 1987) are also referred to as "**snakes**" due to their geometric properties. Active contours follow a semi-automatic approach: deforming curves autonomously try to align with image edges, while they can be controlled during runtime via several parameters (see below) in order to obtain an optimal result. So far, they have been used predominantly in medical image segmentation (e.g. sonograms, computed tomography, magnetic resonance imaging, or histological sections (KIENEL *et al.*, 2007)), where the algorithmic requirements are even higher than on our comparably simple-structured profile drawings.

2. Mathematical Background

Mathematically spoken, a snake is defined as a parameterized curve S = {v(s) : v(x(s), y(s))} with a special potential energy. This energy is composed of three parts: the internal energy E_{int} that is given by the shape of the snake itself, the image energy E_{img} that is given by the structure of the underlying image along the path of the curve, and the constraint energy E_{con} that is caused by user interaction. In short, the total snake energy can be written as the accumulation of the energy contribution of every single snake point **v**.

$$E_{\text{snake}} = \int (E_{\text{int}}(\mathbf{v}) + E_{\text{img}}(\mathbf{v}) + E_{\text{con}}(\mathbf{v}))\,\mathrm{d}s$$

The basic idea of active contours is to strive for an optimal shape and position of the curve on the image by minimizing this energy functional. Hence, the partial energies have to be defined with respect to user-defined quality criteria and thus shall be introduced separately in the following sections.

Let us first consider the **internal energy**, which is generally used to keep the curve smooth and tense. Accordingly, it is most often defined as follows:

$$E_{\text{int}}(\mathbf{v}(s)) = \frac{1}{2}\cdot\left(\alpha(s)\cdot\left|\frac{\mathrm{d}\mathbf{v}}{\mathrm{d}s}\right|^2 + \beta(s)\cdot\left|\frac{\mathrm{d}^2\mathbf{v}}{\mathrm{d}s^2}\right|^2\right)$$

As the equation makes clear, the internal energy depends on the total length and the accumulated absolute curvature values. If no external influences would be present, the snake consequently would be apt to become shorter and smoother until it degenerates into a single point. The non-negative parameters α and β control the influence of tension (stretching) and rigidity (bending), respectively. Although they could be locally adjusted, in general they are kept constant along the snake for reasons of simplicity.

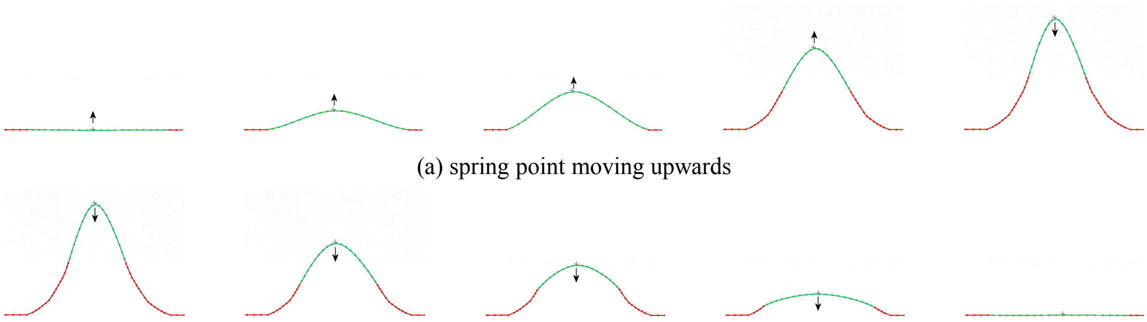
The **image energy** is closely related to the intensity differences in the image. It can be defined twofold (KASS et al. 1987), either as a line functional

$$E_{\text{img}}(\mathbf{v}(s)) = \kappa\cdot(G_\sigma * I(x,y)),$$

if dark lines on white background (or vice versa with negative values for κ) are the objects of interest or as an edge functional

$$E_{\text{img}}(\mathbf{v}(s)) = -\kappa\cdot\left\|\nabla(G_\sigma * I(x,y))\right\|^2,$$

which operates on the gradient image. Here, $I(x, y)$ is the grey value of the current pixel, ∇ is the gradient operator and κ is again a weighting parameter. The Gaussian filter G_σ with standard deviation σ is primarily used to smooth the images in order to reduce the influence of noise, but also to enlarge the attraction range of the image edges and thus to accelerate convergence speed. In more recent publications (e.g. KIENEL and BRUNNETT, 2009) it is replaced by the more powerful gradient vector flow approach (XU and PRINCE, 1997), but for our application the traditional image energy is absolutely sufficient.



(a) spring point moving upwards

(b) spring point moving downwards

Figure 2: *Snakes can be guided locally by applying an interactive spring force to a certain snake point. Due to the internal energy, the neighbourhood is also deformed, but the snake stays smooth.*

Quite often it is necessary to interactively manipulate the snake in order to guide it to certain image edges or to pull it away from unwanted local energy minima. When using the computer mouse, many interaction metaphors are possible, but the so-called spring tool has proven to be by far the most comfortable. In the context of our implementation, applying a spring means that the user introduces a pulling force with respect to a fixed snake point, while the current position of the mouse cursor acts as a movable attraction point. Under the influence of a spring, the behaviour of the snake reminds of a magnetic lasso or bungee rope, because the internal energy formulation always prefers short and smooth curves (fig. 2). In the end, the spring point p indirectly affects the immediate neighbourhood of the attached snake point vp as well. Finally, the **constraint energy** is defined as

$$E_{\mathrm{con}}(\mathbf{v}(s)) = \begin{cases} \omega \cdot \left\| \mathbf{v}_{\mathrm{p}} - \mathbf{p} \right\|^2 & \text{if } \mathbf{v} = \mathbf{v}_{\mathrm{p}} \\ 0 & \text{else} \end{cases}.$$

The parameter ω indicates how fast the snake point **vp** moves towards the spring point **p**.

In addition to the spring energy, we can also define other constraint energies. Especially a global balloon force (COHEN, 1991), which allows contour expansion similar to the inflation of a balloon, turned out to be very helpful. In a certain manner, it compensates the natural shrinking tendency of the snake caused by the internal energy. Conversely, a negative balloon force even accelerates shrinking.

In summary, five parameters influence the behaviour of the snake:

- the tension parameter α,, penalizing stretching,
- the rigidity parameter β, penalizing bending,
- the image force parameter κ,,
- the spring parameter ω,
- and the strength of the balloon force.

In the first view, one might get the impression that using the tool is overly complex. In practice, however, we could quickly find suitable parameter combinations for the most common scenarios and adjustments are often confined to a single parameter.

3. Implementation Details

Finding a local energy minimum is not trivial, because the solution space contains an infinite number of configurations. To this end, it is first of all necessary to discretize the snake to a time-dependent polyline which is deformed in an iterative process (KASS *et al.,* 1987). Since the complexity of the underlying computations is proportional to both the image size and the contour length, it is crucial to set up an efficient implementation, which ensures the semi-automatic extraction to be at least faster than a completely manual approach. We have previously shown how the energy equation can be efficiently solved in linear time using a proper matrix factorization, allowing for a curve deformation in real-time (KIENEL *et al.,* 2006). Moreover, a sophisticated tiling approach significantly reduces memory effort, while in combination with GPU assistance it can dramatically speed up the expensive image energy computation and, consequently, the whole contour extraction process (KIENEL and BRUNNETT, 2009).

Moreover, we have embedded the energy minimization framework in a multithreaded application that remains reactive, specially in the presence of user interaction, which is an essential condition for the acceptance among users. In contrast to implementations that require predefined constant parameters, our software supports various user interventions to control the extraction process. Further speedup of the whole image segmentation could be achieved by parallel execution of image energy computations. Basically, we used three threads being responsible for the following major tasks:

- the contour deformation computation,
- the image force computation,
- and the evaluation of user interaction.

Although multithreaded applications often significantly enhance the overall performance, it is well-known that additional implementation effort arises in order to synchronize the several threads appropriately. In our context, parameter changes are strictly prohibited within one iteration step, which can be considered as a critical section. Otherwise a consistent and intuitive snake deformation cannot be garantueed. Consequently, these changes have to be delayed until the iteration step is complete. Furthermore, without multiple threads, a user interaction with springs (or other interaction metaphors) runs into serious practical difficulties.

We designed our software with a special focus on a flexible and simple interface that does not require any deeper knowledge of the mathematical basics. We provide two convenient tools to support a fast workflow: Firstly, under certain conditions it is possible to automatically determine an adequate initial contour. And secondly, parts of a snake, which might be already partially aligned, can be quickly adjusted piecewise. Both tools are described in the following sections.

Obviously, the number of deformation iterations and the interactive effort can be minimized by a suitable initialization of the snake close to its final resting position. In the majority of cases, the inner of the profile is fairly homogeneous. This observation gives rise to a rather low-level but fully automatic segmentation method. We have implemented an optimized region growing approach similar to the "magic wand" tool known from Adobe's Photoshop that requires a single seed point to be picked by the user anywhere inside the target curve. Subsequently, the outer boundary of the final region is extracted. The resulting curve consists of many pixels which are converted to snake vertices with respect to a predefined point-to-point distance. The obtained initial contour most often provides a very good approximation of the final curve. Its jagged shape naturally originates from aliasing effects. In general, it can be smoothed by computing just a few snake iterations, while at the same time it snaps into the features of interest.

If the snakes are relatively long and complex-shaped, it might be hard to monitor all parts simultaneously. Concentrating on a particular area by interacting with the snake or by modifying parameters may lead to unwanted behaviour in other parts of the deforming curve in the meantime. Apart from that, we observed that sometimes the snake aligns well in most but not in all parts along the object boundary. In order to solve or circumvent these problems, we have designed a highly intuitive and user-friendly tool that helps to correct the position of a certain snake segment, while restricting the deformation locally. Similar to the above mentioned mouse interaction a snake can be grabbed at a certain point and dragged around at will. Two key ideas are fundamental for the superiority of the tool. Firstly, based on the idea of dividing the contour into segments (KIENEL *et al.,* 2006), we define a segment consisting of 10% of the total length of the contour adjacent to the closest vertex with respect to the cursor position. All the remaining vertices are fixed (marked red in the images), meaning that they are excluded from the computation of the energy minimization. Hence, deformation can be accelerated in terms of iterations per second as only a few snake vertices are deformed. Moreover, the user can focus on a particular part of the snake without having to take care about the rest of it. Secondly, the corresponding snake vertex is not rigidly linked with the position of the mouse cursor but only with a spring whose position is updated as the cursor is moved on the image. As a consequence, it suffices to just roughly trace the target outline and to enjoy watching how the snake smoothly locks into the object boundary.

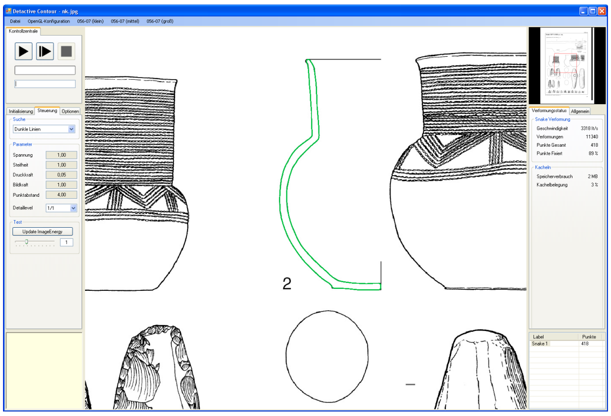


Figure 3: *Screenshot of our active contour implementation.*

4. Practical Aspects

4.1. Snake Initialization

As already mentioned, automatic snake initialization by region growing reduces vectorization time enormously. In case of hatched profiles things seem to be more complicated, because the expected convergence can then be achieved only when the snake is initialized around the profile. However, the idea of region growing can easily be modified to a rather uncommon region shrinking. Even if the image structure prevents this

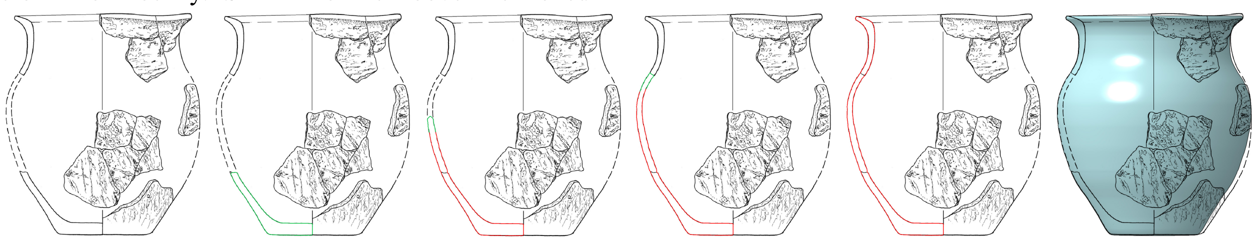


Figure 4: *Incomplete profiles do not pose problems as long as the missing parts are indicated by dashes or dots. While the image energy forces the snake to align with the dashed lines, the internal energy again ensures smoothness.*

approach, the snake will converge towards the actual profile curve sooner or later, simply because of its length minimization attempt.

4.2. Incomplete Profiles

The full potential of the active contour approach unfolds only, when the profiles have gaps, i.e. they consist of several non-overlapping sherds (fig. 4). Due to their striving for internal length and curvature minimization they also interpolate well in areas, where no image energy is present. Additionally, missing parts are often indicated by dashes, which induce further image energies that attract the snake properly. The remaining gaps are small enough to be closed adequately.

Although the interaction and computational effort are much higher than on complete profiles, in practice, connecting multiple parts of a profile is not that difficult as one might expect. In fact, there are even two promising approaches. The first one proceeds similar to that of the hatched profiles, where the snake is initialized around the profile in order to let it shrink and finally snap to its edges from the outside. The second one might be even faster, but requires more interaction and experience (fig. 4). At first, the snake is initialized in one of the sherds by region growing. Then, by using the spring tool and a moderate balloon force the nearby segment of the snake is pulled over the gap until it reaches the outside of the adjacent sherd. By pulling it further along the medial axis of the profile the balloon and image forces cause the snake to quickly snap to the profile edges – this time from the inside. This procedure is repeated until all profile parts are connected.

4.3. Reconstruction

As soon as the profile curve is represented by a closed polygonal chain and the position of the rotational axis is known, we are able to use it as a generatrix for the trivial construction of a virtual rotational body (fig. 5). Although the possibility of a 3D reconstruction is alluring, one should use it cautiously, because the suggestive power of images is often underestimated. On one hand, the rotational body is only an ideal that can deviate tremendously from the actual vessel. Moreover, the manual drawing could have been biased by many sources of error. On the other hand, decorations, and non rotationally symmetric attachments such as handles can be reconstructed not at all or only by an unjustifiably high effort. In the recently published work of (KOUTSOUDIS *et al.*, 2009) this has been tried anyway, but the scientific benefit of these 3D models beyond idealized typological templates is at least questionable. We merely reconstruct the vessels virtually in order to detect obvious drawing or digitization errors and to use the same feature extraction algorithms that have already been developed for range scanned vessels (HÖRR and BRUNNETT, 2008a). In this case, missing handles and decorations are even advantageous.

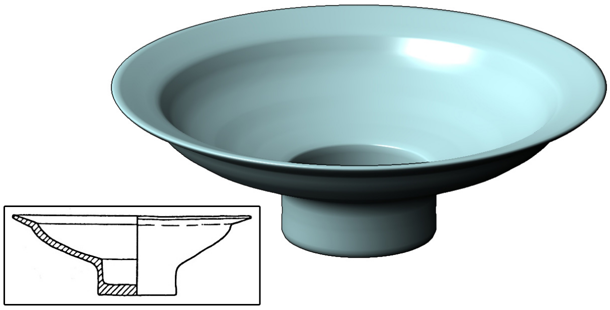


Figure 5: *A so-called footbowl reconstructed from a hatched profile.*

In order to get a high-precision vectorization, the vertex distance on the snake is usually set to less than five pixels. Thus, depending on the image size, snakes consisting of several thousand vertices are not uncommon. However, due to rasterization artefacts the snake may also include small bumps. When reconstructing the vessel as a rotational body, these bumps result in more or less visible horizontal rills that (maybe falsely) imply the use of a throwing lathe. Even though smoothing seems to be mandatory here, it has to be applied carefully, since the profiles often already represent idealized versions of the real issue and remaining features might be smoothed out as well. Hence, for the reconstruction it turned out to be more expedient to perform a feature-preserving thinning of the polyline, which implicitly smoothes the curve as well. To this end, we employ the well-known DOUGLAS-PEUCKER algorithm (1973). Empirical tests have shown, that it is sufficient to use 50 vertices for each the inner and outer wall.

4.4. Limitations

When it comes to exporting the extracted profiles, the scale and resolution of the scanned image have to be specified. While the resolution is usually encoded in the image header, the scale is known from the catalogue only. However, it is changing only very rarely throughout a publication, so that there is no additional work to be done by the editor. We should note that the final result of the curve extraction also depends on the correct alignment of the image. When scanning the tables, it might happen, that the template is slightly tilted, so that the drawn axes of rotation do not run vertically anymore. Such a biased profile may result in a reconstruction that conveys an impression that clearly deviates from the original. Thus, the editor should be attent to this issue and might need to realign the scan either manually or even simpler by marking two points on the axis. In fact, however, this topic interferes with daily work only marginally.

Things are more problematic when the drawing contains serious measurement errors. Occasionally we observed, that the base part of the profile has been drawn significantly too short, i.e. the axis of rotation has been shifted towards the profile (astonishingly, the opposite case almost never occurred). Even though the shape of

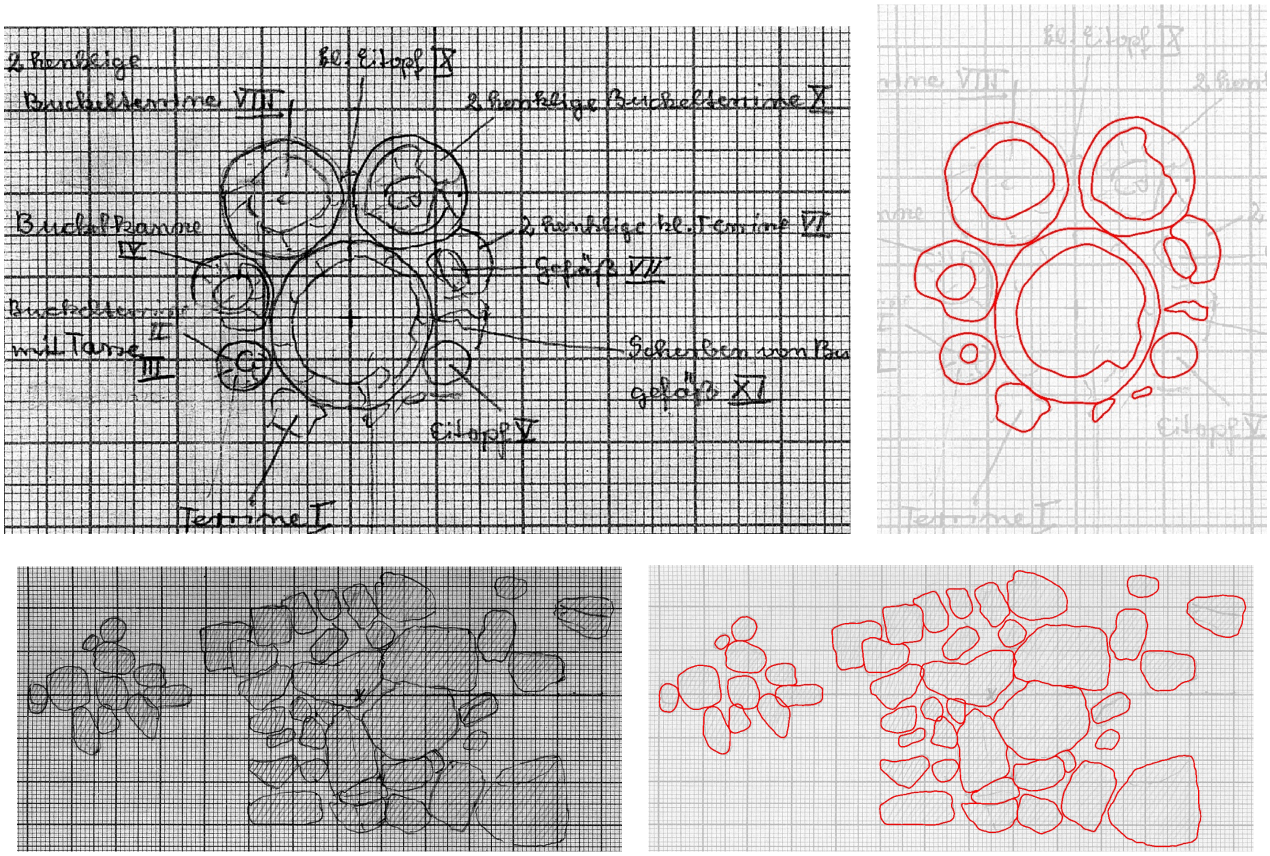


Figure 6: *Millimetre paper has an adverse effect on the image energy, so that automatic snake initialization is prevented and interactive intervention cannot be avoided. However, the spring tool is powerful enough to quickly pull the snake towards the actual regions of interest. The vectorizations on the right took four and twelve minutes, respectively, whereas a completely manual extraction would have taken up to five times as long.*

the profile is not affected by this carelessness, the virtual reconstruction and hence the relevant measures are almost useless. Correcting this kind of error is much more time-consuming, since it requires the editor to consult information in the finding catalogue.

Results and Conclusion

During daily practice it could soon be confirmed, that snakes are an optimal tool for the vectorization of vessel profiles. At Saxony's Archaeological Heritage Office in Dresden, we presented the software to archaeologists without any special mathematical or IT background by giving a short 15-minute tutorial. A little later, the editors were able to vectorize up to 40 digitized profiles per hour. As long as the tables are not available in terms of PDF files or single images, they still have to be scanned of course. However, this task can be done with justifiable effort and can be performed by unskilled workers as well.

We are convinced, that the presented method will enable many archaeological institutions which do not have the budget to purchase expensive 3D scanners to prepare their findings for automated digital shape analysis in any way, and to compare them with those published long time ago. Actually, vectorization of rasterized images is only the beginning of a chain of possible subsequent applications. Within a few weeks, databases containing several thousands of finds can be established and made available to colleagues. Then, by means of similarity estimation (HÖRR and BRUNNETT, 2008a) and automated classification (HÖRR *et al.*, 2008b) totally new fields of research open up.

It is worth to mention, that the presented tool is applicable for other vectorization purposes as well. Substantial improvements could be achieved for example for hand-drawn excavation maps or stratigraphic images. In figure 6, we have intentionally chosen two very difficult samples in order to illustrate the flexibility and robustness of the active contours approach. Although the grid structure of the millimetre paper and the bad contrast significantly influence the image energy, it is still possible to quickly guide the snakes towards the desired edges. Compared to a manual vectorization we achieved a speedup of approximately factor 5.

One thing left for future work is the support for open snakes, which differ from closed ones only in that the two endpoints are fixed. Then, breaking edges, ridges, valleys, and other curvilinear features could be captured as well.

References

ADAN-BAYEWITZ D., KARASIK A., SMILANSKY U., ASARO F., GIAUQUE R. D., LAVIDOR R., 2009. Differentiation of ceramic chemical element composition and vessel morphology at a pottery production center in Roman Galilee. *Journal of Archaeological Science 36* (11), pp. 2517–2530.

COHEN L. D., 1991. On active contour models and balloons. *CVGIP: Image Understanding 53* (2), pp. 211–218.

DOUGLAS D. H., PEUCKER T. K., 1973. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *The Canadian Cartographer 10* (2), pp. 112–122.

GILBOA A., KARASIK A., SHARON I., SMILANSKY U., 2004. Towards computerized typology and classification of ceramics. *Journal of Archaeological Science 31* (6), pp. 681–694.

GROSMAN L., SMIKT O., SMILANSKY U., 2008. On the application of 3-D scanning technology for the documentation and typology of lithic artifacts. *Journal of Archaeological Science 35* (12), pp. 3101–3110.

HÖRR C., BRUNNETT G., 2008a. Similarity estimation on ancient vessels. In *GraphiCon Proceedings 2008*, pp. 94–100.

HÖRR C., LINDINGER E., BRUNNETT G., 2008b. New paradigms for automated classification of pottery, In *Proc. Computer Applications and Quantitative Methods in Archaeology 2008*, in press.

HÖRR C., BRUNNETT G., VIX C., 2010. Line Drawings vs. Curvature Shading: Scientific Illustration of Range Scanned Artefacts, *In Proc. Computational Aesthetics 2010,* in press.

IOVIŢĂ R., 2009. Ontogenetic scaling and lithic systematics: method and application. *Journal of Archaeological Science 36* (7), pp. 1447–1457.

KAMPEL M., SABLATNIG R., 2007. Rule based system for archaeological pottery classification. *Pattern Recognition Letters 28* (6), pp.740–747.

KARASIK A., SMILANSKY U., 2008. 3D scanning technology as a standard archaeological tool for pottery analysis: practice and theory. *Journal of Archaeological Science* 35 (5), pp. 1148–1168.

KASS M., WITKIN A., TERZOPOULOS D., 1987. Snakes: Active contour models. *International Journal of Computer Vision 1* (4), pp. 321–331.

KIENEL E., BRUNNETT G., 2009. Tile-based image forces for active contours on GPU. In *Proc. Eurographics 2009,* pp. 89–92.

KIENEL E., VANČO M., BRUNNETT G., 2006. Speeding up snakes, In *Proc. VISAPP 2006*, pp. 323–330.

KIENEL E., VANČO M., KOWALSKI T., CLAUSS R., BRUNNETT G., 2007. A framework for the visualization of cross sectional data in biomedical research. In *Visualization in Medicine and Life Sciences*, pp. 77–97.

KOUTSOUDIS A., PAVLIDIS G., ARNAOUTOGLOU F., TSIAFAKIS D., CHAMZAS C., 2009. *Qp*: A tool for generating 3D models of ancient Greek pottery. *Journal of Cultural Heritage 10* (2), pp. 281–295.

XU C., PRINCE J. L., 1997. Gradient vector flow: A new external force for snakes. In *Proc. Computer Vision and Pattern Recognition 1997,* pp. 66–71.