Surface Registration Markers from Range Scan Data

John Rugis[1,2] and Reinhard Klette[1]

[1] CITR, Dep. of Computer Science, The University of Auckland
Auckland, New Zealand
[2] Dep. of Electrical & Computer Engineering, Manukau Institute of Technology
Manukau City, New Zealand
john.rugis@manukau.ac.nz

Abstract. We introduce a data processing pipeline designed to generate registration markers from range scan data. This approach uses *curvature maps* and *histogram-templates* to identify local surface features. The noise associated with real-world scans is addressed using a (common) *Gauss filter* and *expansion-segmentation.* Experimental results are presented for data from The Digital Michelangelo Project.

1 Introduction

There are a number of challenges associated with 3D range scan digitizations (or 3D surface reconstructions in general) of real-world objects. At first the captured data should be accurate, and second the (in general) various data sets need to be unified into one consistent surface model. In general, only a partial section of an object surface is acquired with each scan, and a number of scans need to be aligned and subsequently merged together. When each scan is taken from a different uncalibrated viewpoint, aligning the scans can be time consuming [11, 15]. There exists a number of algorithms [2, 3, 14], including the widely used Iterative Closest Point algorithm (ICP), that can refine a given rough alignment of scan pairs, two at a time. Initial matching and alignment of multiple (possible many) 3D scans is still largely an open problem. Since range scans normally already each have the same scaling, alignment to some fixed reference points involves a linear transformation consisting of a 3D translation and a 3D rotation. Exact alignment is in practice not possible due to the noise in real-world data. Input data uncertainty is also present when surface patches have been generated via less reliable computer vision 3D surface recovery techniques such as photometric stereo or structure from motion (see, e.g., [10] for 3D surface recovery techniques).

Early work used 2D features, such as contour based grouping using relaxation methods [13]. For the visualization of implicit surfaces we cite [1]. More recent work in [5] uses (what they refer to) a “curvature map method” to characterize a local signature for every point in a scan. This can be seen as a continuation of surface characterizations by Gauss maps in differential geometry, or by extended Gaussian images (see, e.g., [17]) in computer vision.

In this paper we propose a matching and initial alignment approach based on identifying a small number of registration markers. The registration markers that we generate are based on local surface curvature features. An advantage to using surface curvature is that it is rotation and translation invariant. However, one problem that needs to be overcome is that curvature, being a second derivative property [4, 9], is very sensitive to local noise.

The sequence of steps in the processing pipeline that we use for surface marker generation is to (i) calculate (noisy) curvatures from the point data, (ii) filter and segment the curvature data, and then (iii) identify and mark local features using the curvature data. We illustrate the application of this sequence of steps to multiple scans by an example. For each step in this processing pipeline, we select and define appropriate methods and algorithms.

As a particular difficulty, in practice, scan overlap regions can be small relative to the scan size, and thus our registration markers need to be smaller than the scan overlap. We keep, from the outset, the demands of extensive real-world datasets in mind. Finally we demonstrate our approach on data from The Digital Michelangelo Project [11, 12].

2 Curvature and Curvature Estimators

Studies on surface curvature can be traced back to original work by Gauss [6]; see books on differential geometry (e.g., [4]) or a discussion of surface curvature in the context of 3D image analysis in [9]. The surface curvature of continuous smooth surfaces is a well-defined property, however, when working with point set data, triangulated surfaces or 3D digital images, the surface curvature can only be estimated. There are a number of different existing curvature estimators for such situations [9]. In this paper, we use an uncompensated orthogonal cut

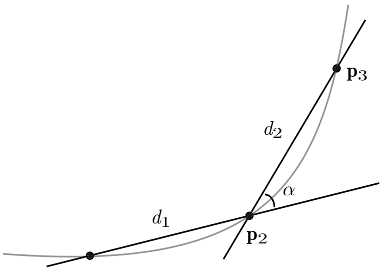


Fig. 1. Planar line curvature estimation.

method to calculate a mean curvature as described in [16].

The curvature estimation approach that we use, for each scan point, is to firstly identify the four nearest neighbor points associated with two orthogonal planar cuts. Next, for each of the two cuts, calculate an estimated signed planar line curvature. Finally, take the mean of these two curvatures as an estimate of the mean surface curvature.

With reference to Figure 1, the planar line curvature (in each planar cut) is estimated as the incremental angular advance divided by the incremental change in length

$$\kappa = \frac{\alpha}{(d_1 + d_2)/2} \tag{1}$$

where d_1 is the length of the line segment from $\mathbf{p}_1$ to $\mathbf{p}_2$ and d_2 is the length of the line segment from $\mathbf{p}_2$ to $\mathbf{p}_3$. Consequently, the curvature can be calculated as

$$\kappa = \left(\frac{2}{||\mathbf{v}_1|| + ||\mathbf{v}_2||} \right) \cos^{-1}\!\left(\frac{\mathbf{v}_1 \cdot \mathbf{v}_2}{||\mathbf{v}_1|| \, ||\mathbf{v}_2||} \right) \tag{2}$$

where $\mathbf{v}_1 = \mathbf{p}_2 - \mathbf{p}_1$ and $\mathbf{v}_2 = \mathbf{p}_3 - \mathbf{p}_2$.

In practice this results in a (noisy) mean curvature value associated with each 3D scan point. We also tested further curvature measures (as described in [9]); however, the mean curvature estimate gives results that correspond to actual image appearance of the object and thus is the most relevant feature for our purposes.

3 Curvature Maps, Filtering and Segmentation

In this section, we convert the mean curvature data at surface scan points into a (2D) *curvature map*, which is an array of the same dimension (ignoring squashing) as the given 2D scan array, where values are mean curvatures at scan points [16]. There are a number of advantages when using such curvature maps, including the possibility of visualization and processing using 2D image processing techniques. Pixels in 2D images are, by standard convention, stored in 2D arrays and we apply adjacency definitions based on the related orthogonal grid. If the 3D point data has been acquired in a 3D orthogonal grid, then the curvature

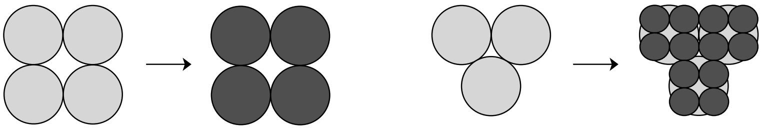


Fig. 2. Mappings of orthogonal (on the left) or hexagonal (on the right) grids into an orthogonal grid.

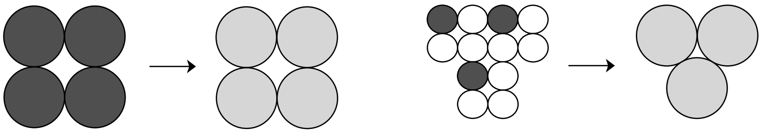


Fig. 3. Reverse mappings of orthogonal (on the left) or squashed hexagonal (on the right) grids into the original grid.

mapping is straightforward (defined by orthogonal cuts parallel to coordinate planes, see [7]).

Data acquired in the Michelangelo project uses a hexagonal adjacency (i.e., 6-adjacency in the image plane) in a virtual projection plane, defined by the order and geometry of scan acquisition. In such a case we can use a special squashed dot mapping [16].

Mappings for both cases, either orthogonal or hexagonal, are shown in Figure 2, with the reverse mapping as shown in Figure 3. The second mapping has the expense of quadrupling the number of pixels. (We do not discuss further scan acquisition geometries in this paper.)

We then apply a 2D Gaussian filter to the curvature map to reduce noise. As an illustration of this effectiveness of this technique, consider noisy data from a sampled planar surface. The noisy data points will (incorrectly) exhibit a symmetrical distribution of positive and negative curvatures centered around zero, with the average value being (correctly) zero. The Gaussian filter performs exactly the desired spatial averaging. Note that we are addressing the noisy data problem in 2D rather than in the original 3D domain. The terms in an $n_1 \times n_2$ Gaussian convolution kernel centered at $(0,0)$ are determined (as usual) using the formula

$$h(n_1,n_2) = \frac{h_g(n_1,n_2)}{\sum\limits_{n_1}\sum\limits_{n_2} h_g} \quad \text{with} \quad h_g(n_1,n_2) = e^{-(n_1^2+n_2^2)/2\sigma^2}$$

where σ is effectively (in our case) a smoothing area factor.

The filtering process correctly reduces the extremes in the distribution of mean curvature values. The resulting mean curvatures are more representative of the actual surface curvature, which is concentrated in a much smaller range around zero. Now, in order to reveal previously obscured detail, we perform a linear expansion centered around this zero-curvature region of interest.

Multi-threshold segmentation of the expanded data is then performed to partition the data into a number of *curvature bins*. This data reduction enables the approach presented in the next section. An example of how we perform a combined expansion-segmentation operation is given in the experiment section of this paper.

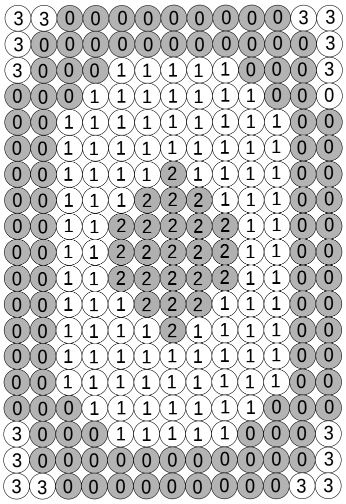
Also, in preparation for local feature identification, we reverse the 3D to a 2D curvature mapping process, by selecting pixels, depending on the original adjacency as illustrated in Figure 3, and update the mean curvature associated with each of the 3D scan points with its filtered and expansion-segmented value.

4 Local Feature Identification

We use *histogram-templates* to search through the filtered and segmented curvature data for local surface features. A histogram-template consists of (i) a sliding window that is segmented into subsets and (ii) a set of histograms, one for each subset. Each histogram accumulates curvature counts into a number of curvature bins. The number of histogram curvature bins is assigned to be exactly the same as the number of segmentation curvature bins in the previous section.

Note that, even though we are now working within the 3D dataset, we do the histogram-template search in a 2D fashion using the 2D orthogonal grid.

The histogram-templates are carefully designed (at the given scale of scan data) to characterize local surface features; a multi-scale approach would be relevant if uncertainty increases for the given scale of scan data. An example



3	3	0	0	0	0	0	0	0	0	0	3	3
3	0	0	0	0	0	0	0	0	0	0	0	3
3	0	0	0	1	1	1	1	1	0	0	0	3
0	0	0	1	1	1	1	1	1	1	0	0	0
0	0	1	1	1	1	1	1	1	1	1	0	0
0	0	1	1	1	1	1	1	1	1	1	0	0
0	0	1	1	1	1	2	1	1	1	1	0	0
0	0	1	1	1	2	2	2	1	1	1	0	0
0	0	1	1	2	2	2	2	2	1	1	0	0
0	0	1	1	2	2	2	2	2	1	1	0	0
0	0	1	1	2	2	2	2	2	1	1	0	0
0	0	1	1	1	2	2	2	1	1	1	0	0
0	0	1	1	1	1	2	1	1	1	1	0	0
0	0	1	1	1	1	1	1	1	1	1	0	0
0	0	1	1	1	1	1	1	1	1	1	0	0
0	0	0	1	1	1	1	1	1	1	0	0	0
3	0	0	0	1	1	1	1	1	0	0	0	3
3	0	0	0	0	0	0	0	0	0	0	0	3
3	3	0	0	0	0	0	0	0	0	0	3	3

Fig. 4. Subsets for the definition of a histogram template (here: four histograms for indicated positions) for a surface pit; the individual histograms are not shown in this figure.

histogram template is shown in Figure 4.

In the example, the histogram-template is designed to characterize pit-like surface defects which have radial symmetry. The template indicates that four histograms, indexed zero through three, are tabulated for each selected 3D data point [3]. Histogram two is the center region which should contain negative curvature values, histogram one is the pit perimeter which should contain positive curvature values, histogram zero is a guard ring which should contain low curvature magnitude values, and histogram three is a "don't care" region. Individual histogram bins are assigned such that first bin accumulates the count of largest negative magnitude curvatures and the last bin accumulates the count of largest positive magnitude curvatures.

The histogram-template is used to search through the curvature values looking for the desired feature match.

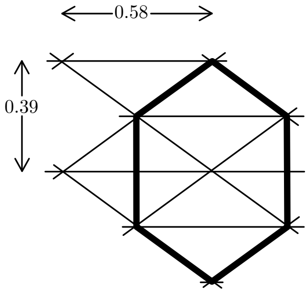


Fig. 5. Scanning grid.

5 Performance Evaluation

The value of algorithm performance evaluation in the field of computer vision has been discussed in [8]. A possible starting point for performance evaluation is to use a data set having known indisputable characteristics. These indisputable characteristics can be used to establish what is referred to as a *ground truth*.

5.1 Synthetic Data

One way to insure the existence of ground truth in a data set is to model and generate synthetic data having known properties. In our case we have chosen to synthesize data using a hexagonal adjacency orthogonal projection grid. Because curvature is scaling dependent, we needed to establish numerical distance

[3] Excluding those close to the array's border.

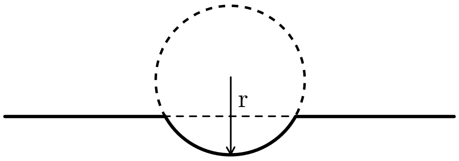


Fig. 6. The indentation model.

values for our grid. Figure 5 shows the values that we used[4]. We will refer to the illustrated vertical dimension as the *resolution* of the scan.

The synthetic objects that we "scanned" consisted of a planar surfaces having spherical indentations and spherical bumps. The indentation model is shown in Figure 6. Note that the sphere cuts into the plane by a distance of one half the radius. Spherical bumps are modeled similarly.

5.2 Performance Measures

We evaluated our curvature estimator in the presence of varying levels of Gaussian noise. We defined noise level by a scale factor relative to the scan resolution. A noise scale factor of one means that the noise sigma value is equal to the resolution. A factor of two means that the noise sigma value is equal to the resolution dived by two, and so on. Of course, the actual curvature of a planar surface is zero and the curvature at any point on a sphere is a constant equal to the reciprocal of its radius.

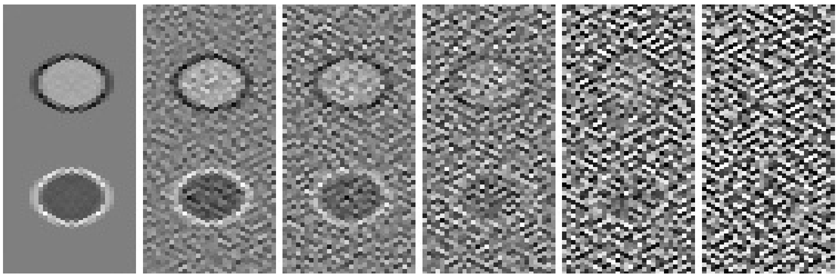


Fig. 7. Curvature estimation: no noise, sf:16, sf:8, sf:4, sf:2, sf:1.

Figure 7 illustrates the performance of the 2-cut curvature estimator in the presence of varying levels of noise. Positive curvature is shading coded as white. Zero curvature is shading coded as medium gray and maximum negative curvature is encoded as black. Thus, bumps are shown in the top half of the figure

[4] These values closely match those used in the real-world data that we explore later in this paper.

and indentations in the bottom half of the figure.

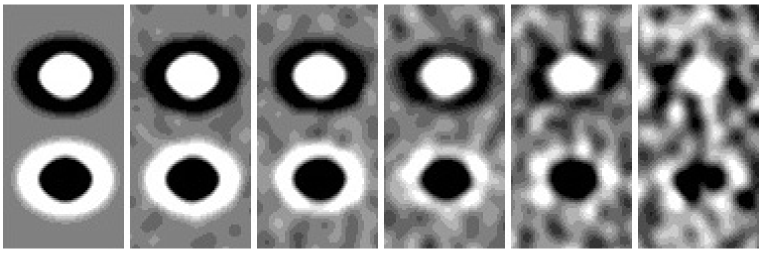


Fig. 8. Filtered and segmented: no noise, sf:16, sf:8, sf:4, sf:2, sf:1..

Figure 8 illustrates the performance of the filtering and segmentation process, again in the presence of varying noise levels.

5.3 Feature Identification Evaluation

We applied the complete feature identification process to a scan of a more comprehensive synthetic object. The object has sixteen indentations associated with spheres having radii which vary from 0.1 to 2.0. Figure 9 illustrates the results pictorially. Identified features are each marked with five red pixels. Note that because the feature identification process searches through (nearly) every image pixel, the same feature is usually hit and marked more than once.

The results from Figure 9 are summarized in Table 1. We can make a number of observations. Increasing noise levels tend to cause the process to associate a

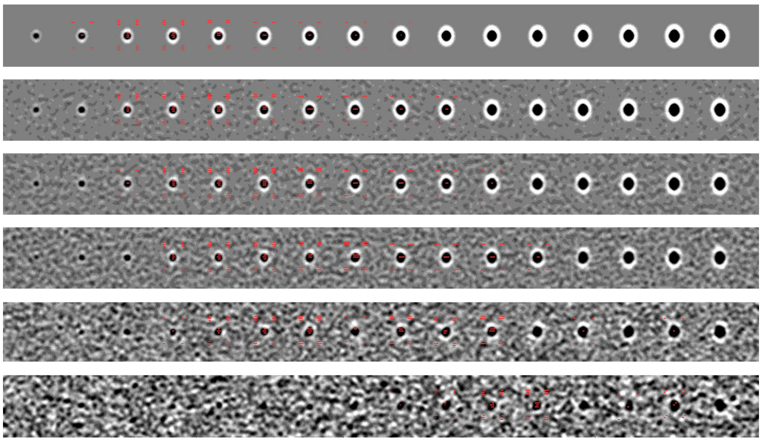


Fig. 9. Feature identification results: synthetic data and six noise levels.

noise factor	marker hits: small radius ——————— large radius indentation																features
none		3	9	10	8	4	3	2	1								8
16			9	10	12	8	6	5	3	3							8
8			3	9	10	12	10	5	5	3	2						9
4				7	10	10	8	12	6	6	5	3					9
2				2	8	10	13	3	9	7	11		2		2		10
1									1	3	9	12		2	4		6

Table 1. Summary of feature marker hits from Figure 9.

smaller size to the feature. The fixed size matching template does in fact identify a range of indentation feature sizes. There are no false positives, even in the presence of rather high noise.

6 Experiments and the David Dataset

We have also tested our approach using the extensive David dataset from the University of Stanford Digital Michelangelo Project [11, 12]. The David data set consists of 1.93 giga-bytes of data which has been made available in nine compressed files. The uncompressed data set represents approximately 1.1 billion 3D space points. There are a total of 6,540 raw scan files collected into 515 groupings. Each scan was acquired over a fixed width of approximately 140mm and a height of generally no larger than 600mm. The David statue is over five meters tall.

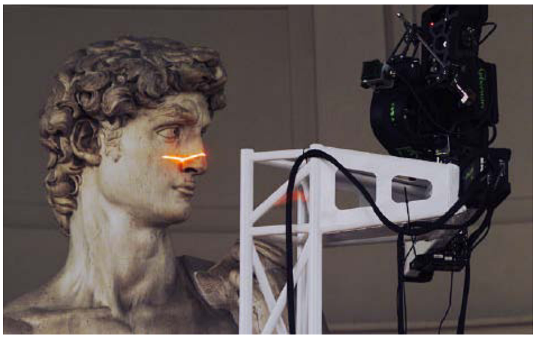


Fig. 10. The scanner acquiring 3D data [11] using structured lighting (known as light plane projection [10]).

The scanner is shown acquiring data in Figure 10, and a rendered image of the points from two adjacent overlapping scans is shown in Figure 11. It is not uncommon for a scan to contain upwards of 800,000 points. The scanning system's physical geometry is illustrated in Figure 12. The cyan colored boxes represent the volume regions associated with possible individual scans.

The University of Stanford group responsible for the project undertook a rather time consuming initial manual alignment of the individual scans. They reported both on this process and the need for an automated global matching method[12]. The surface identification markers described in this paper are candidates for a feature based global matching technique.

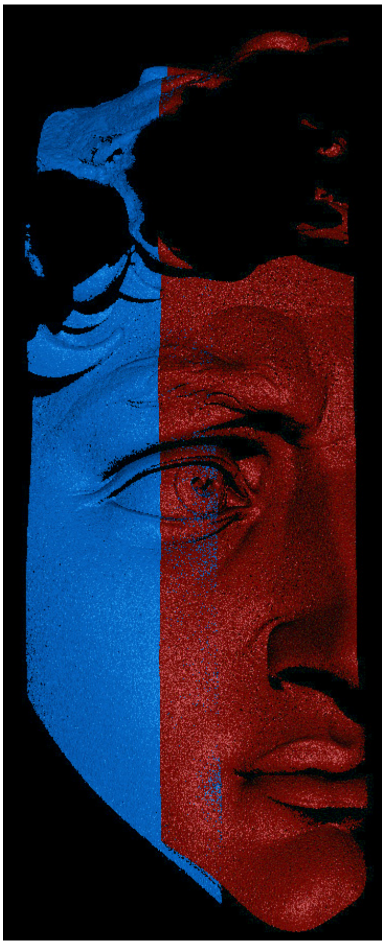


Fig. 11. Points from two scans ("blue" and "red scan") rendered with lighting.

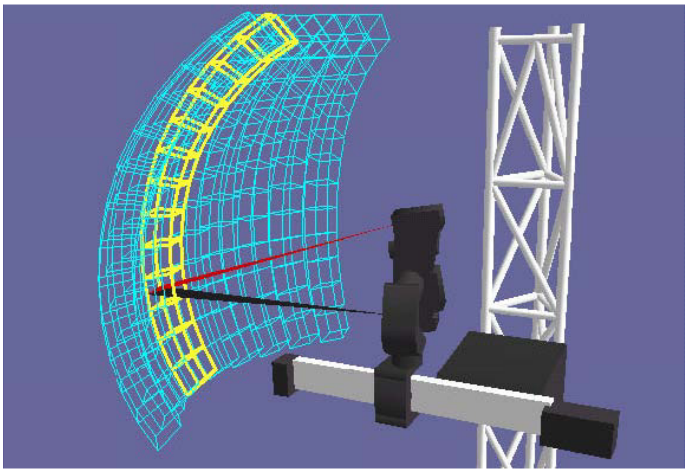


Fig. 12. The scanner's imaging volume [11].

6.1 Data Structure

Each scan is acquired in a regular sweeping pattern of scan lines, left-to-right, and top-to-bottom. Each scan line consists of a zig-zag pattern of 486 points as shown in Figure 13. The scanning pattern results in a hexagonal grid. Of course, the scanner does not find the reflective surface of an object at all points, and this is illustrated, for example, in Figure 13 where the white dots represent the surface of an object and the darker dots mean that nothing was found in that direction.

The regular scanning pattern suggests 6-adjacency (based on the hexagonal grid) shown in Figure 14. A '-1' entry in the array means that no surface point was found. A non-negative integer entry is an index value into another array of surface point data including its 3D location and a not yet calculated mean curvature value.

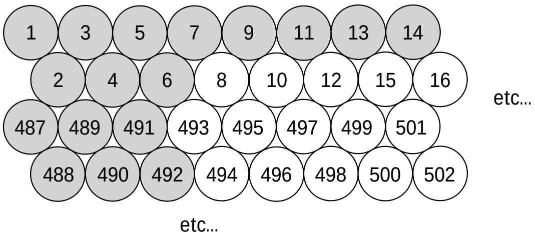


Fig. 13. Top left corner of the scanning sequence.

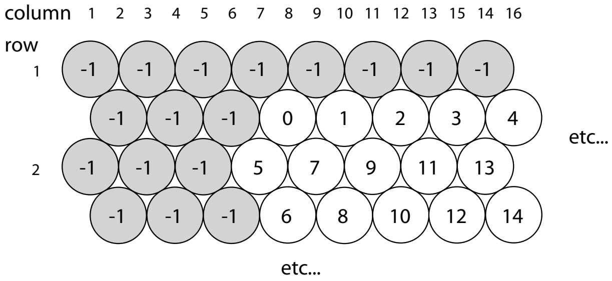


Fig. 14. The hexagonal grid of a scan.

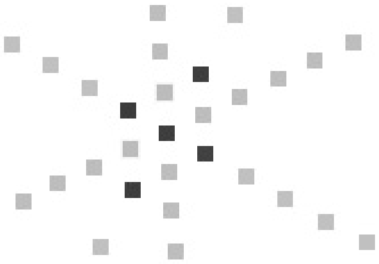


Fig. 15. 6-adjacency and orthogonal cut points.

6.2 Curvature Estimators, Filtering and Segmentation

Since the 3D data points have 6-adjacency within each scan, the nearest neighbor orthogonal planar cut points [5] are selected as shown in Figure 15 (dark squares), and the squashed dot mapping is used as discussed above.

An eight-bit precision mean curvature map of a section of a scan of David's face is shown in Figure 16 along with a histogram of that entire scan. Positive surface curvature is defined as that which bends away from the scanning source or, equivalently in this case, as that curvature associated with viewing a convex hull from the outside. Maximum positive curvature is shading coded as white. Zero curvature is shading coded as medium gray and maximum negative curvature is encoded as black. Surface detail in this image is obscured by noise.

For Gaussian filtering of the David data, a sigma of four is large enough to smooth noise but not so large as to remove surface detail. A sample Gaussian convolution kernel matrix having dimensions eight and sigma four is:

[5] Note that, particularly in the case of data having 6-adjacency, a three-cut mean calculation is also possible. Experiments using this more expensive method gave similar results to those presented in this paper.

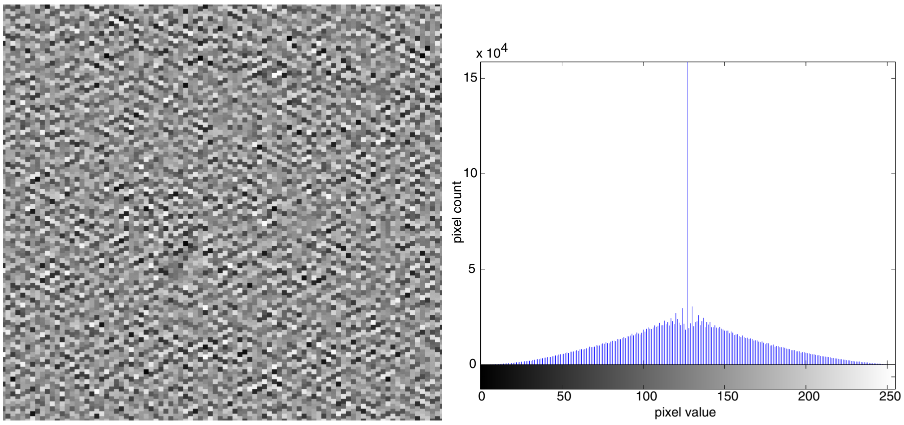


Fig. 16. Noisy mean curvature map (on the left) and its histogram.

```
0.0099 0.0119 0.0135 0.0144 0.0144 0.0135 0.0119 0.0099
0.0119 0.0144 0.0163 0.0174 0.0174 0.0163 0.0144 0.0119
0.0135 0.0163 0.0185 0.0197 0.0197 0.0185 0.0163 0.0135
0.0144 0.0174 0.0197 0.0209 0.0209 0.0197 0.0174 0.0144
0.0144 0.0174 0.0197 0.0209 0.0209 0.0197 0.0174 0.0144
0.0135 0.0163 0.0185 0.0197 0.0197 0.0185 0.0163 0.0135
0.0119 0.0144 0.0163 0.0174 0.0174 0.0163 0.0144 0.0119
0.0099 0.0119 0.0135 0.0144 0.0144 0.0135 0.0119 0.0099
```

A Gaussian filtered version of the previous curvature map is shown in Figure 17. Note that there is now not much visible detail and that the levels are now mostly concentrated around the middle medium gray as shown in the associated

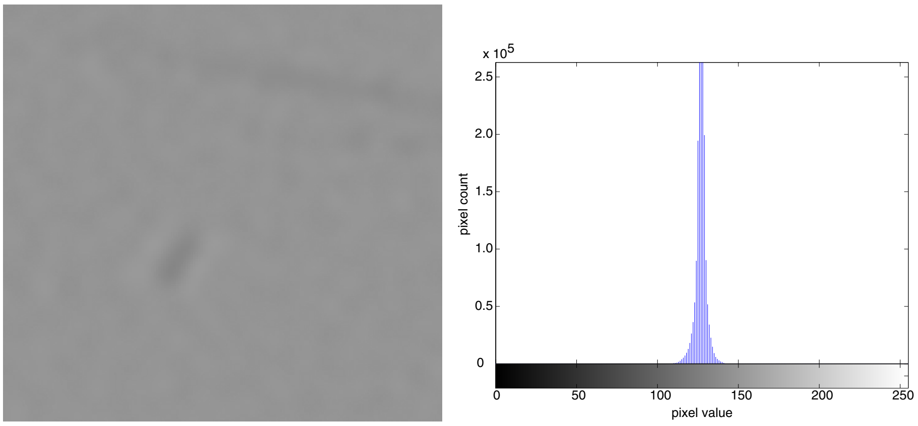


Fig. 17. Processed map (on the left) and used Gaussian filter.

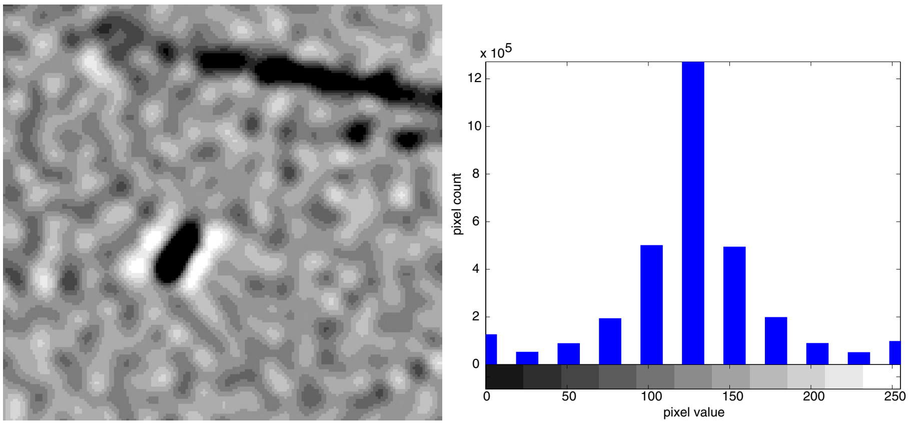


Fig. 18. Processed map: multi-level segmentation.

histogram.

As mentioned earlier in this paper, we need to expand the middle histogram region to recover detail. Because of the eight-bit precision used, the middle region contains only a small discrete range of integer values. We perform a linear expansion around the center (medium gray) and clip the results (to full-black and full-white). The result, in this example, is an image containing pixels having only eleven different integer intensity values.

In general, this limited precision expansion will always also simultaneously multi-level segment the data into a limited number of segmentation bins.

An *expansion segmented* version of the previous mean curvature map is shown in Figure 18. Curvature detail is now readily apparent. The histogram shows the range of eleven segmentation values.

We now reverse the squashed dot mapping process, by selecting pixels as illustrated in Figure 3, and update the mean curvature associated with each 3D point with its filtered and expansion-segmented value.

6.3 Feature Identification

There are numerous small (approximately 2-4mm diameter) surface 'pits', due to weathering and abuse, found scattered on the surface of the David statue[6] that are ideal candidates for surface markers. The pit-like features certainly meet the requirement of being smaller than scan overlap regions. One of these pits is

[6] This can be verified by rendering images of triangle mesh versions of the scan data with specular reflections and appropriately placed lighting.

clearly visible in Figure 18. A pit can be characterized by its center of negative curvature and a rim of positive curvature.

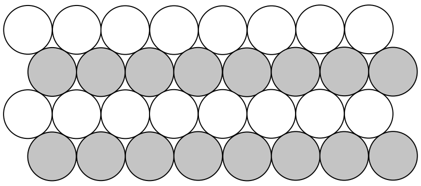


Fig. 19. Every-other-row pixel subset for feature searching (shown as white).

Experimental comparison showed that it was possible to reduce the histogram template search time by employing data thinning, as illustrated in Figure 19, where only every other row is examined[7]. This thinning also results in pixels defined in an orthogonal grid, which, rather conveniently, means that we can use a rectangular search template.

We used precisely the histogram template given earlier as an example in Figure 4 to search through the curvature data for pit surface features. In this particular experiment, the non-square dimensions of the template compensate for the unequal aspect ratio of the rectangular data pixel subset. In fact, this template maps back to a (nearly) square region in the 3D dataset.

The identification test for pit features that we used with the David dataset is a minimum of twelve black (fully negative curvature) values in the center region, a minimum of four white, or nearly white (very positive curvature) values in the rim, a maximum of four black or nearly black (very negative curvature) values in the guard ring, and a maximum of two white (fully positive curvature) values in the guard ring. This identification test was able to identify pit surface features as reported in the next section.

6.4 Some Feature Mapping Results

We illustrate the application of the example histogram template to four different, but overlapping, scans of David. The results are shown in Figure 20. The first and third scans where acquired from the same view-point. The second and forth scans where acquired from different view-points. Note that there is at least one feature that links each scan to at least two other scans. For example, there is

[7] The sampling frequency appears to be at least double that of the Nyquist frequency associated with the smallest underlying surface features, and thus, there is no aliasing associated with data thinning by a factor of two.

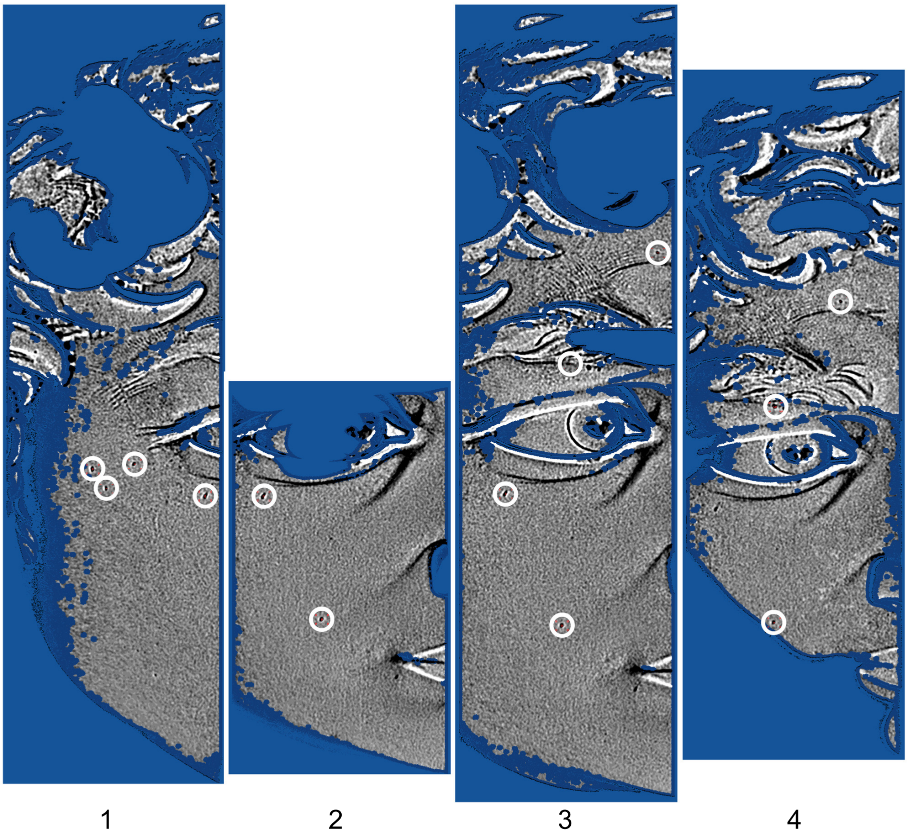


Fig. 20. Marker results.

a feature just below the left eye that links scan one to each of scans two and three.

Comparison, by the authors, with visual observations of the 3D triangle mesh models mentioned previously, confirms that common features in different scans have been successfully identified and marked using our processing pipeline. Because these registration markers are associated with the 3D dataset, additional 3D marker characteristics, such as the size of the pits, can be calculated.

7 Conclusions

We have demonstrated the possibility and effectiveness of data smoothing *after* a shading encoded mean curvature calculation on noisy data. In general, this approach reduces complexity in that it moves the smoothing process from the 3D domain into the 2D domain. We have also demonstrated the effectiveness of using 2D (rotation invariant) radially symmetric matching templates to identify 3D surface features.

Further work is anticipated to include additional marker characterizations and using curvature based registration markers to automatically generate an initial alignment of diverse overlapping scans of the same object. Results using different curvature measures (besides the mean curvature) will be published in a forthcoming report.

Acknowledgements: The first author acknowledges the financial support from The Department of Electrical and Computer Engineering at Manukau Institute of Technology. The authors thank Stanford University for access to the Digital Michelangelo Project archive.

References

1. Balsys, R.J. and K.G. Suffern: Visualisation of implicit surfaces. *Computers & Graphics*, **25**:89–107, 2001.
2. Brown, B.J. and S. Rusinkiewicz: Non-rigid range-scan alignment using thin-plate splines. In Proc. *3D Data Processing Visualization Transmission*, pages 759–765, 2004.
3. Chang, M.C., F.F. Leymarie, and B.B. Kimia: 3D shape registration using regularized medial scaffolds. In Proc. *3D Data Processing Visualization Transmission*, pages 987–994, 2004.
4. Davies, A., and P. Samuels: *An Introduction to Computational Geometry for Curves and Surfaces.* Oxford University Press, Oxford, 1996.
5. Gatzke, T., C. Grimm, M. Garland, and S. Zelinka: Curvature maps for local shape comparison. In Proc. *Shape Modeling and Applications*, pages 244–253, 2005.
6. Gauss, C.F.: *General Investigations of Curved Surfaces* (Reprint of publications from 1825 and 1827). Dover Publications, New York, 2005.
7. Hermann, S. and R. Klette: Multigrid analysis of curvature estimators. In Proc, *Image and Vision Computing New Zealand*, pages 108–112, 2003.
8. Klette, R., Stiehl, H.S., Viergever. M.A, and Vincken, K.L. (eds.): *Performance Characterization in Computer Vision.* Kluwer, Dordrecht, 2000.
9. Klette, R. and A. Rosenfeld: *Digital Geometry.* Morgan Kaufmann, San Francisco, 2004.
10. Klette, R., K. Schlüns, and A. Koschan. *Computer Vision - Three-Dimensional Data from Images.* Springer, Singapore, 1998.
11. Levoy, M.: The digital Michelangelo project. In Proc. *3D Digital Imaging Modeling*, pages 34–43, 1999.
12. Levoy, M., K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, J. Shade, D. Fulk: The digital Michelangelo project: 3D scanning of large statues. In Proc. *SIGGRAPH*, pages 131–144, 2000.
13. Parent, P. and S. Zucker: Trace inference, curvature consistency, and curve detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, **11**:823–839, 1989.
14. Pulli, K.: Multiview registration for large data sets. In Proc. Int. Conf. *3D Digital Imaging and Modeling*, pages 160–168, 1999

15. Pulli, K., B. Curless, M. Ginzton, S. Rusinkiewicz, L. Pereira, and D. Wood: Scanalyze v1.0.3: A computer program for aligning and merging range data. Stanford Computer Graphics Laboratory, Stanford, 2002.
16. Rugis, J.: Surface curvature maps and Michelangelo's David. In Proc. *Image and Vision Computing New Zealand*, pages 218–222, 2005.
17. Sun, C. and J. Sherrah. 3-D symmetry detection using the extended Gaussian image. *IEEE Trans. Pattern Analysis and Machine Intelligence*, **19**:164–168, 1997.