

DigitLAB, an Environment and Language for Manipulation of 3D Digitizations*

Oscar E. Ruiz, Associate Professor
Interdisciplinary Research Center (CII) on CAD/CAM/CG.
EAFIT University. Medellin, COLOMBIA

Abstract: In Computer Aided Geometric Design the fitting of surfaces to massive series of data points has many applications, ranging from medicine to aerophotogrametry. However, even the mathematical meaning of fitting a surface to a set of points is dependent on functional considerations, and not only on the geometric properties of the point set. Also, characteristics of some parts of the data set must be interpreted as stochastic in nature, while others must be taken as literal and therefore they become constraints of the surface. For these reasons, among others, automated surface fitting alone does not produce results usable at industrial level. At the same time, it does not take advantage of sampling patterns, particular shapes of the cross sections, functionally different regions within the object, etc. The latest literature reviews show the need for utilities to process point data sets that must be asynchronous, (applicable at any time and upon any region of the point set). Addressing this need, this article reports new tools developed within DigitLAB, a language that allows topological traversal, retrieval and statistical modifications to the data, and surface fitting. They can handle arbitrary topology, as case studies in medicine, mathematics, landscaping, etc. discussed here demonstrate.

Résumé: Dans la Conception Aidée par Ordinateur (CAO) l'ajustement de surfaces a séries massives de points en 3D a beaucoup d'applications, dès la médecine à l'aerophotogramétrie. Toutefois, la seule

definition mathématique de cet ajustement dépend de considérations fonctionnelles et ne pas seulement de la géométrie de la série de points. En plus, quelques caractéristiques des données doivent être interprétées comme aléatoires, en tant que d'autres doivent être pris littéralement et donc, elles contraignent la surface. Par ces raisons l'ajustement automatique des surfaces n'est pas prêt pour l'emploi industriel. Au même temps, l'ajustement automatique ne profite pas des schèmes de capture des points, des formes particulières des sections traverses, des regions fonctionnellement différents sur l'object, etc. Les dernières revisions de litterature montrent un besoin d'utilitaires qui permettent de transformer les données d'une manière asincronique (applicable en n'importe quel temps ou sur quelle portion de la série de points). Cet article rapporte les nouvelles utilitaires développés en DigitLAB, qui permettent la recuperation, le traitement statistique des données et l'ajustement de la surface. Ces utilitaires sont capables de récupérer des topologies arbitraires, comme le montrent les cas étudiés en médecine, mathématiques ou modelage du terrain.

1. Introduction. The general problem of surface reconstruction from point clouds is not one of computational geometry only (to infer shape from data), but one of statistics and data processing, to ensure diagnostics and a minimum quality of the data input to the geometric algorithms. Collateral tools help to take advantage from special characteristics or patterns of the

1. * For Publication in *Proceedings of the Joint Conference : IDMME'2000 & CSME FORUM 2000.* Montreal, Canada. May 16-19, 2000

digitization. Patterned (planar or grid) 3D digitizations are common in industrial, medical, and artistic applications. With the introduction of stereolithography and rapid prototyping, the treatment of patterned planar digitizations has drawn renewed interest from the engineering community. However, the attempt for fully automated topology recovery from planar data have encountered problems, mainly related to the determination of the cross sections of the object on one level and the relation between sections in neighboring levels. This publication reports a computer environment, DigitLAB, that improves the results of [12] in processing the data and inferring shape. It exploits the advantages of planar 3D digitizations for surface reconstruction, being able to deal with non trivial topologies. As side benefit, it allows application of the tools developed on mesh integration from range images. Therefore, it covers planar and grid (from optical sampling) patterned 3D point data.

This article is organized as follows: Section 2 surveys the relevant literature. Section 3 discusses the geometric tools devised along with examples. Section 4 presents additional case studies with objects of diverse topology and geometry. Section 5 concludes the paper.

2. Literature Review and Background.

Fig. 1 shows a conceptual classification of steps for surface reconstruction from point data. The main tasks are data capture, topology recovery, and continuity enforcement (smoothing). This paper refers to *Topology Recovery* as the process to identify and formally represent neighborhood information in the data set.

Data Acquisition: Data acquisition implies the steps of object fixturing, equipment calibration and data collection and correction (following the calibrated

parameters), to produce the cloud of (x,y,z) coordinates. A short summary rather than a complete survey of sampling methods is included here. The interested audience is invited to read [17] for deeper insight. Acquisition hardware may work by contact or remotely. Contact measurement is based on the position of the kinematic joints that hold a probe touching the object. In metrology centers (Coord. Measurement Systems - CMS) the computer commanded position of these joints produces predefined trajectories of the probe tip. Trajectories have mostly 3 degrees of freedom (X-Y-Z table), and therefore recondite features are not reachable in this case. When more *dofs* are present, the probe is able to reach creases and holes, at the penalty of manual measurement. In CMS the digitization is generally realized in planar patterns, while articulated arms hardly achieve this systematic pattern. It is claimed and illustrated here that approximate planar samples can be obtained with pre-processing tools ([14, 12]) if other characteristics of the digitization (density, homogeneity, etc) are present.

Remote methods may be optical, acoustic or magnetic, depending on which physical phenomena they rely upon. For example, range imaging records a depth field in grid patterns corresponding to pixel arrays. Computer axial tomography builds planar slices of raster data (x rays) based on absorption properties of the scanned object ([16, 10]). Interferometer methods record dark regions on the object, which are ones of iso-distances (levels) measured on the ray direction.

Independently of the physical hardware used, this investigation concerns methods producing patterned samplings (planar, grid, or quasi-planar). The reason for this focus is the availability of neighborhood

information implicitly present on pattern sampled data.

Topology Recovery: Notice that B-Rep models in half wing edge format ([8]) cannot directly apply for surface

report the difficulty in completing or inferring lost or hidden regions of the surface. The present article assumes that in topology recovery such regions should not be inferred. Rather, algorithms should only recover the portion of the object actually witnessed, leaving to other tools with different reasoning the artificial completion of surface portions. Hollow, partial objects with holes (for example a carnival mask) are cases in which no completion should be made. Therefore, inferring portions of the object not sampled is beyond the scope of this article.

For cases where shell closure is a goal, Alpha Shapes ([4]) and Marching Cubes ([7]) are used as part of larger procedures [6, 5, 9]. As a coarse summary, Alpha Shapes establish sets of points closer to each other than a parameter α . When $\alpha = 0$, the alpha shapes are exactly the original set of points. When $\alpha = \infty$, the alpha shapes are the convex hull of the given point set. User-selected ranges of intermediate α values recover the connectivity of the point set in the form of a simplicial complex formed by 0, 1, 2 and 3 simplexes. In [6] it is discussed that although the output of this stage is a topologically correct mesh, geometrical degeneracies (for example dangling edges) may be present, and the collection of simplexes is post - processed in order to leave only 2-simplexes with an acceptable geometry. Marching Cubes ([7]) algorithm builds a closed faceted surface that approaches an implicit surface in $\mathbf{R}^3(f(\mathbf{p}) = 0)$, where the function $f()$ may be inferred from the digitization samples (as in [9]).

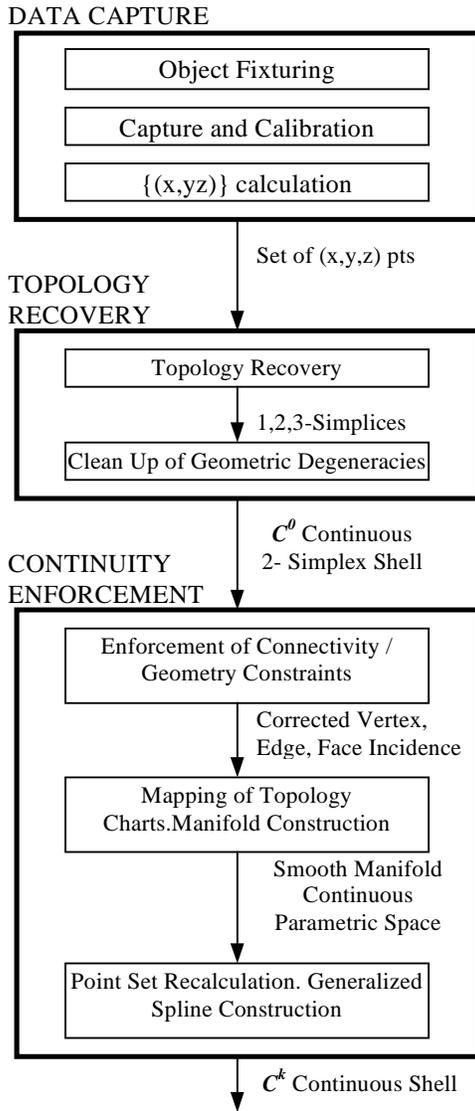


Fig. 1: Surface Reconstruction Steps

reconstruction since they are watertight closed. In general, an extended B-Rep structure must be devised to record *absence* of surface on some parts, or existence of borders of the recognized surface or partial mask (possibly with holes). Authors [17,9]

Regarding the carrier geometries of the shells, this investigation uses very simple geometries such as 3 and 4 - vertex facets.. The last ones are of course not flat in general, but are easily subdivided into

triangles. These primitives have been found sufficient to recover a correct topology.

Surface Smoothing: Once a topologically correct shell is attained, applications may require a level of continuity (typically C^1 or C^2) on the surface built. For this purpose, several attempts on parametric surfacing are present in literature. One of the best fundamented is the one by Grimm ([5]). It starts with a topologically correct C^0 continuous shell, and covers it with vertex, edge and face *charts* in order to obtain a complete mapping between the C^0 shell and a manifold M . This mapping enables to define a chart-depending parameterization that produces a C^2 continuous surface. In Fig 1, *Enforcement of Connectivity / Geometry Constraints* refers to the re-calculation of vertices whose incidence degree is too high or faces with too many sides. Additionally, vertex re-calculation is used to restore planarity to faces with the correct incidence degree ([1, 3]). The result is a topological atlas that offers correct conditions on the vertices, edges and faces for chart construction. Chart construction results in the coverage of the manifold surface with overlapping patches in the parametric space, which help to ensure C^2 continuity of generalized B-spline surfaces. In [6] Alpha Shapes are used as input to Grimm’s method, along with a corrected set

of points, calculated as the least deviated from the sampling of the physical object.

Mentioned in the literature [6, 17], and from our own experience, it appears that a considerable effort may be spent in ensuring C^1 or C^2 continuity in selected regions of the object where only C^0 continuity exists. Sharp “character” edges are present in objects (for example car bodies), and only their detection already presents formidable difficulties. On the other hand, in the surveyed literature the smoothing by parametric quadrilateral patches in which two of their vertices coincide (which means triangular regions) is not explicitly informed. This aspect is important, since geometrical coincidence leads to degeneracies in tangent vectors at these positions. The treatment of these degeneracies is obviously of capital importance for continuity enforcement.

From the survey presented, it is clear that industrial usage of digitization tools requires foremost topological correctness. We show that it can be achieved with simpler approaches, accompanied by collateral tools that allow to see, evaluate, subdivide, selectively retrieve and prepare the data for the application of the *surfacing* algorithms. The later are only one step (vital, but not even the last one) of a large

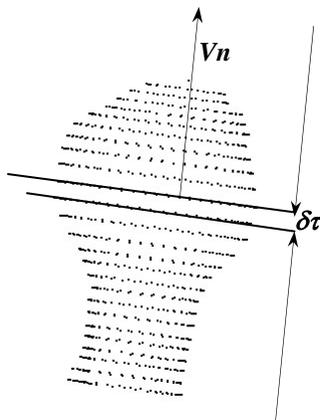


Fig. 2: Quasi-planar digitization of femur (head portion).

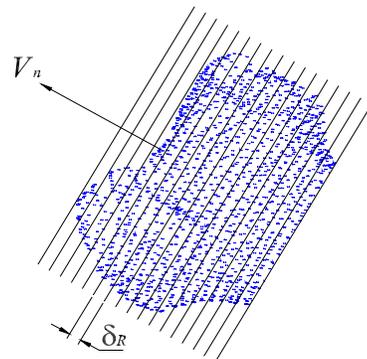


Fig. 3: Random, sparse digitization of femur.

set of operations required to produce an industrially useful CAGD surface. The rationale behind DigitLAB is to provide the tools to treat the data, along with the ones to surface it. As such, we conform ourselves with very simple geometries (triangles and 4 vertex facets), with a strong focus on topology correctness, and data treatment, within a comfortable environment.

3. Geometric Reasoning Tools. In this section each tool devised will be presented along with its application example.

3.1 Point Set Partition: A first step in treating a digitization that supposedly follows a planar pattern is to identify it and / or correct it to ensure such a characteristic. Among others implemented, the following instruction addresses such a need:

```
pocket = pocketing (
    point_list,          /*IN: point cloud */
    normal_vec,         /*IN: common normal */
    plane_separation,  /*IN: distance between
                        sampling planes */
    dist_tolerance,   /*IN: sampling tolerance*/
):
```

this instruction is issued after the user identifies a normal vector (nearly) common to all digitization planes, via statistics tools. *pocketing()* classifies points from a unordered set into pockets. If the digitization has good quality one may trust each pocket to define its own normal vector (via least squares fit). Otherwise one may impose a common normal vector. Next, each point has to be projected against several close planes. In some cases a point can be projected onto two or more consecutive planes (each level “borrows” points from neighboring ones). This option is useful when a digitization is too sparse, and so, information from neighboring sections must be consulted to “complete” a section. Obviously in this case the quality of the original digitization is very dubious, and the modified one will have fuzzy

characteristics that must be addressed (see *filter* tools ahead). The result of *pocketing()* is a list, sorted by level, of point subsets. Each subset corresponds to the points recorded (or projected) on a sampling plane and it is still disordered.

Figs. 2 and 3 show digitizations of a femur. They are quasi-planar and random respectively. The first one presents an inherent normal vector, while in the random one the user defines what the virtual digitization planes would be. Depending on the call parameters, points *nearby* each plane are replaced by their projections on such plane. *dist_tolerance* of 50% would produce no points shared by levels. Less than 50% would leave points claimed by no plane. Above 50% points are borrowed among planes.

3.2 Recovery of cross sections: The next step after the partition into planar sections is to recover each cross section of the object, cut by a sampling plane. In previous work, the authors recovered convex or star (non-null kernel) sections by angular sort about a pivot point [11]. At this time, there exist no restrictions on the cross section. DigitLAB recovers disconnected sets of general polygons with holes. Mathematically, one starts with a disordered set of coplanar points S_i belonging to level i . Conceptually, S_i is first partitioned into the different subsets S_{ij} (still disordered) that make each polygon j in the section:

$$\Sigma_i = \{ S_{i1}, S_{i2}, S_{i3}, \dots, S_{iN} \},$$

$$\Phi = S_{ij} \cap S_{ik} \text{ for all } j \neq k, S_i = \bigcup_{j=1}^{j=N} S_{ij} \quad (1)$$

where each S_{ij} contains the points that form a chain of close neighbors. That is, the ones belonging to a closed contour.

Each subset S_{ij} is then ordered into a list L_{ij} which is the closed contour representing such a cut of the object. Notice that the L_{ij} 's

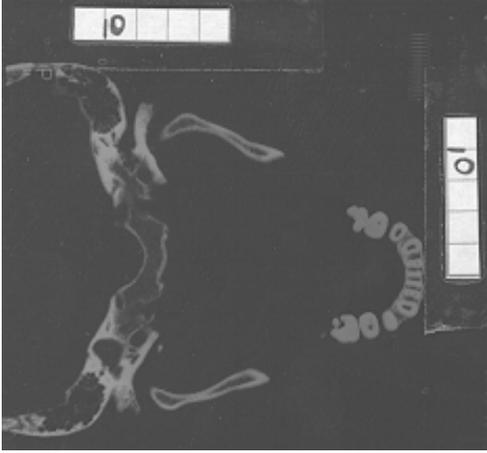


Fig. 4: CAT section of human yaw.

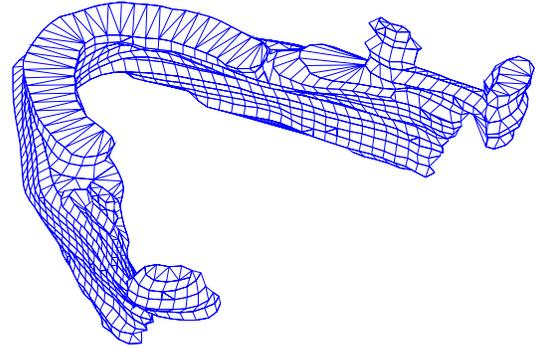


Fig. 5: Final DigitLAB model from a CAT of human yaw.

satisfy also equation (1). Although the conceptual steps (partition + ordering) are different they are implemented at once in the following algorithm.

```

contour_set  rebuild_section  (
  set_of_points  S, // IN. Point set in the level
  real           $\delta$  // IN. max. neighbor dist.
)

```

```

{
1  section = [];
2  while ( S )
3  {
4    contour = closest_chain( S,  $\delta$  );
5    S = S - contour;
6  }
7  section = section + { contour };
7  return ( section );
}

```

```

contour closest_chain(
  set_of_points  S, // IN / OUT. Point set
  in the level
  real           $\delta$  // IN. Characteristic
  distance of Digitization
)

```

```

{
1  seed = first( S );
2  seed_neighbors = [];
3  while ( seed )
4  {
5    seed_neighbors = seed_neighbors + { seed };
6    next = closest( seed, ( S - { seed, pred( seed ) } ) );
7    if ( next )
8    {
9      d = distance( seed, next );
10     if ( d <  $\delta$  )
11       seed_neighbors = [ seed_neighbors

```

```

12     next ];
13     else
14       next = NULL;
15     }
16     seed = next;
17   }
18   return( seed_neighbors );
}

```

The algorithm displayed is intended only for illustration purposes, since many details are not addressed. However, it shows that a fundamental assumption is the compliance of the digitization with the condition that $\delta < \frac{1}{2} \delta_{detail}$ (Nyquist criteria), where δ is the effective sampling interval and δ_{detail} is the dimension of the smallest object feature that the designer wants to model. Unless such a condition is met, from principles of digital sampling no algorithm will be able to ensure adherence to the object geometry or topology. Fig. 5 shows the results of applying DigitLAB algorithms for shape recovery for a human yaw from a Computer Axial Tomography (Fig. 4). The yaw presents sections that not only have no kernel, but also are disconnected. The algorithm succeeds in recovering the disjoint contours. For the sake of space savings no intermediate steps are shown.

3.3 Section Filtering and Resampling: Filtering and resampling performed on each

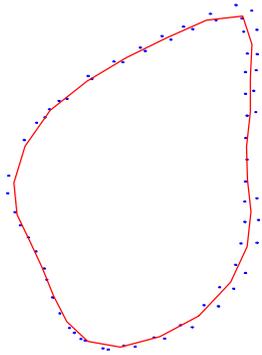


Fig. 6: Effect on contour filtering on a rough point sequence.

section must be done after ordering the point set (contour recovery). The main reasons to apply these tools to the contour loops are: (i) later stages require similar number of vertices in all loops (ii) the vertex interval must be stable within each loop, (iii) rough contours, caused by deficient digitization conditions and/or treatment tools (“borrowing” points between levels) must be smoothed.

Fig 6 shows the effect produced by point projection and point borrowing between contours. The section shown was initially sparsely populated, leading to problems in boundary recovery. With point borrowing and projection the level certainly becomes

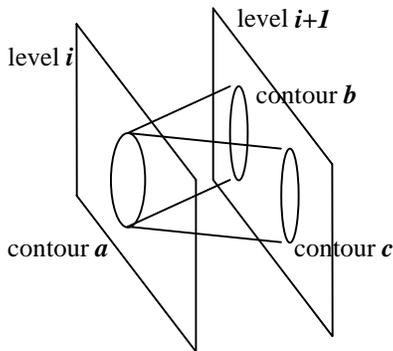


Fig. 7: Pairs (a,b) and (a,c) of the lofting between contours in levels i and $i+1$

more populated, at the obvious price of “fuzziness” in the definition of the boundary. Filtering helps to lower the “teeth” effect, although it must be applied with care, since it rounds sharp edges that may be actually part of the object, and shrinks the sections when applied repeatedly or when the filtering window is too large. A typical instruction performing this task looks like:

```
smooth_polys = build_resample( polys, N, filter )
```

where N specifies the number of points in the filter window, and *filter* specifies that the resample is of filter type. Other types are: by distance, by number of points, etc.

3.4 Interlevel Surfacing: After the sections of the object (cut by the sampling planes) are recovered, a sequential automatic surfacing operation follows, linking consecutive cutting planes (Fig. 7). For two sets of M and N contours in two consecutive levels the questions that must be solved are: (i) find the set of pairs in the relation $l(a,b)$, meaning “ a,b are contours, with $a \in level_i$, $b \in level_{i+1}$, and a and b are consecutive cuts of a topological cylinder or tree“. (ii) given that $l(a,b)$ holds, contours a and b must be lofted by a bijective function $f(): a \rightarrow b$. $f()$ which is the skin of the loft

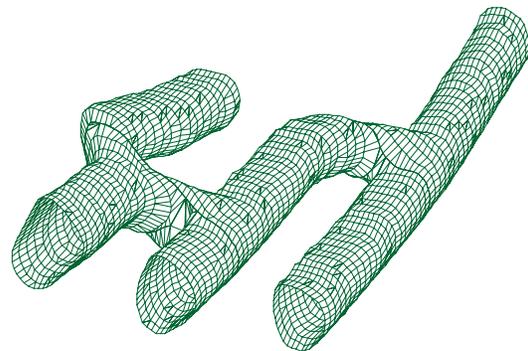


Fig. 8: DigitLAB lofting for “tunnel” data set.

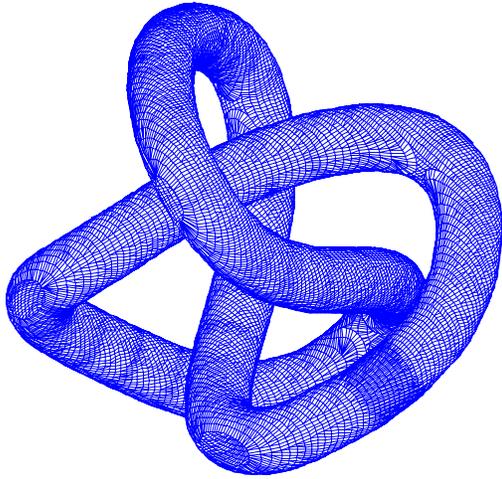


Fig. 9: DigitLAB result for the topological tri-foil.

operation. (iii) if there are pairs (a,b) y (a,c) satisfying $l()$, find a strategy for b and c to share contour a . (iv) given a contour a that appears in no pair (a,b) , generate a triangulation that covers it. a is a dead end and must be closed.

Although these are (relaxed) mathematical definitions, it is clear that these problems are under-specified. Aspect (i) requires geometric reasoning algorithms that determine whether, for the sampling performed, there are reasons to believe that contour a of level i participates in the pairs (a,b) , (a,c) , ...etc, with b, c in level $i+1$. To satisfy (ii) one counts with infinitely many $f()$ bijections from a to b . This is indeed the cause for the lofting operation to be ill-defined in CAD packages. In DigitLAB one is interested in one map that renders 4-vertex facets (3-vertex facets render $f()$ non-bijective) with shape factors acceptable for Finite Element Analysis. For aspect (iii) Fig. 7 suggests the need for allowing pairs (a,b) and (a,c) and shows that heuristics are needed so the two loftings share the contour a . Aspect (iv) is a standard problem with many solutions found in Computational Geometry literature.

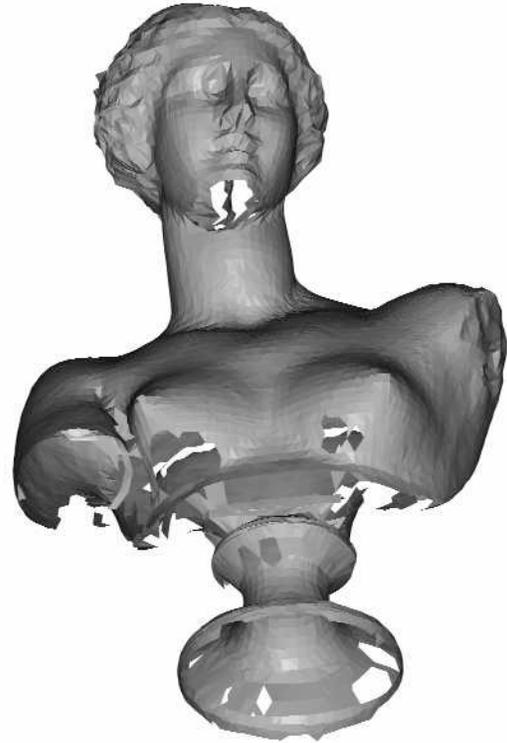


Fig. 10: Integration of Range Masks from several range pictures using DigitLAB.

The figures presented display results in solving these questions for levels in which all contours in both levels are external. Fig. 8 shows a data set (called “tunnels”) presenting bifurcated loftings.

4. Additional Topologies and Study Cases.

Topological Trifoil: Fig. 9 displays the results of DigitLAB applied to a data set called “tri-foil”. The algorithms discussed are able to recover the topology of self-trespassing donuts with no difficulty. Notice that this object is not homeomorphic to either spheres or torii.

Range Images: In processing range images a first step is the synthesis of incomplete and non-connected masks from each image. As mentioned before, a data structure has been devised by the author in collaboration with Fraunhofer IGD (Darmstadt, Germany) to accommodate such relaxation from the traditional half wing edge. However, a

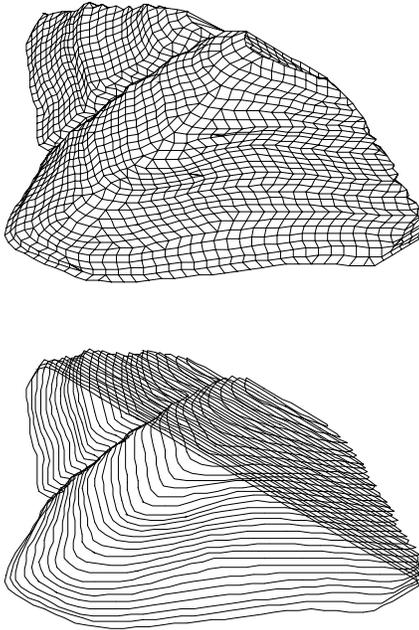


Fig. 11: Mountain topography recovery from aerophotogrametric data. Below: recovered contours. Above: Inter-level surfacing.

fundamental obstacle remains in the integration of different images to complete the object shell as much as possible with the images recorded [9, 15, 2]. DigitLAB has provided a contribution to the solution of this problem by applying the following sequence of operations to the reunion (un-processed) of individual masks: (a) planar resampling, (b) contour recovery, and (c) filtering. Operation (a) produces a virtual digitization of the whole object with fuzzy regions where masks overlap. Operation (b) recovers the contours as if these fuzzy regions were produced by “point borrowing”. Lastly, operation (c) smooths the jagged contours, therefore averaging the overlapping portions. Thereafter, normal inter-level lofting follows. This strategy is far from perfection because many statistical issues remain. However, it is presented here as an unexpected bonus when using DigitLAB. Results are displayed in Fig. 10, which shows the masks integration. The

empty spaces are places where the range pictures have dark regions; no presence of the object is detected by the camera. Therefore, the integration algorithm has no data for those neighborhoods.

Aerophotogrametric Landscaping: The geometric kernel from DigitLAB has been used to recover the topography of mountains sampled by pictures from a flying plane, and whose data is patterned in iso-altitude levels. Observe in Fig. 11 that this topology represents a simple case compared to the ones just discussed, but it shows of the applicability of the algorithms assembled. The processing was performed for AeroEstudios, a company specialized in aerophotogrametry and landscaping operations (see [13] and acknowledgment section).

4. Conclusions. The material presented shows that there is a particular advantage in devising algorithms for pre - processing of the point set, as well as for specific surface recovery. Since all known algorithms fail when the data set is not properly sampled, these tools become mandatory in working in industrial environments. Moreover, they are justified by the next stages of this activity, such as the fitting of selectively smooth patches. In those cases a suite of geometric diagnostic and reasoning tools would be as important as in topological recovery. However, before arriving to the issue of surface smoothing, topological correction needs to be perfected. Future work by the author focus on the completeness of section recovery, in order to deal with empty inner spaces in the solid.

Acknowledgments: This investigation was supported by Colciencias (Colombian National Council for Science and Technology) through projects AIS-96 and FALCON-99 and EAFIT University. The algorithms for synthesizing masks from

range pictures were developed by the author and Peter Neugebauer at Fraunhofer IGD, Medical Imaging Division (Darmstadt, Germany) with the sponsorship of Deutscher Akademischer Austauschdienst (DAAD), and will be subject of a focussed specific publication. The modelers MicroStation and AutoCAD and their APIs and programming technical support are part of the ADN and SYNNERGY agreements of the CII-CAD/CAM/CG with Autodesk and Bentley Systems respectively.

References

- [1] Catmull, E., Clark, J., "Recursively Generated B-Spline Surfaces on Arbitrary Topological meshes". *Computer Aided Design*, Vol 10, No 6, Nov, pp 350-355, 1978.
- [2] Curless, B., Levoy, M., "A volumetric Method for Building Complex Models from Range Images". *Proceedings SIGGRAPH-96*. pp 303-312, 1996.
- [3] Doo, D., Sabin, M., "Behavior of Recursive Division of Surfaces near Extraordinary Points". *Computer Aided Design*, Vol 10, No 6, Nov, pp 356-360, 1978.
- [4] Edelsbrunner, H., "Three Dimensional Alpha Shapes". *ACM Transactions on Graphics*, Vol 13, No 1, Jan, pp 43-72, 1994.
- [5] Grimm, C., Hughes, J., "Modeling Surfaces of Arbitrary Topology using Manifolds". *Proceedings, SIGGRAPH - 95 Annual Conference Series*, Los Angeles, August 6-11, n. 29, pp. 359-368, 1995.
- [6] Guo, B., "Surface Reconstruction: From Points to Splines". *Computer Aided Design*, Vol 29, No 4, pp 269-277, 1997.
- [7] Lorensen, W., Cline, H., "Marching Cubes: A High Resolution 3D Surface Construction Algorithm", *ACM Computer Graphics*, Vol. 21, No. 24, July, pp. 163-169, 1987.
- [8] Mantyla, M., "An Introduction to Solid Modeling", *Computer Science Press*. Rockville, USA, 1988.
- [9] Neugebauer P., "Reconstruction of Real World Objects via Simultaneous Registration and Robust Combination of Multiple Range Images". *International Journal of Shape Modeling*, Vol. 3, No. 1&2, pp 71-90, 1997.
- [10] Ramos, A., Basso, K., Partichelli, P., Scharcanski, J., Longhi, M., "Um Sistema para Visualizaçao Tridimensional de Tomografia Computarizada". *Prodeedings, XIII Congreso Latinoamericano de Informática*, pp. 545, 1987.
- [11] Ruiz, O., Henao, C., "Modelaje Geométrico de Estructura Ósea". *X Congreso Internacional de Ingeniería Grafica*, June 3-5, Málaga, Spain, 1998.
- [12] Ruiz, O., Posada, J., "Computational Geometry In The Preprocessing Of Point Clouds For Surface Modeling". *Proceedings, IDMM-98 (Integrated Design and Manufacturing in Mechanical Engineering)*, May 27-29, Compiègne, France. pp. 613-620, 1998.
- [13] Ruiz, O., Trujillo, A., Toro, C., "Falcon: Software Tools for Aerophotogrametric Processing". *Proceedings, XII International Conference on Graphics Engineering- Ingegraf*, June 1-3, Valladolid, Spain, 2000.
- [14] Schrader, S., "DigitLAB I. Surface Reconstruction from Planar Digitizations", Thesis. EAFIT University, Medellín, COLOMBIA, 1999.
- [15] Turk, G., Levoy, M., "Zippered Polygon Meshed from Range Images". *Proceedings SIGGRAPH-94*. pp 311-318, 1994.
- [16] Vanzandt W., "Scientific Visualization: One Step in lab Analysis Workflow", *Advanced Imaging*, Feb, pp. 20, 1992.
- [17] Varady, T., "Reverse Engineering of Geometric Models – An Introduction". *Computer Aided Design*, Vol 29, No 4, pp 255-268, 1997.