Acknowledgements

The writing of a doctoral dissertation is no simple undertaking and my own work in this respect has proven to be no exception. Quite naturally, I am indebted to a number of people, both in my professional and personal life, without whom this manuscript could have never been brought about.

First of all, I would like to express my gratitude to my doctoral advisor Prof. dr. Frank Van Reeth for introducing me to the wonderful field of Computer Graphics, and for his enthusiasm and flexibility throughout the realization of this dissertation. I would also like to thank the rest of the academic staff of the *Expertise Center for Digital Media*: our director Prof. dr. Eddy Flerackers, Prof. dr. Philippe Bekaert, Prof dr. Wim Lamotte and Prof. dr. Karen Coninx. I must express my appreciation for their unceasing efforts at providing an academically inspiring and prolific environment.

I owe a special note of gratitude to my fellow workers, who all contributed in their very own way in making my years at the EDM a very pleasurable experience. Therefore, I would like to thank my close colleagues Cosmin Ancuti, Koen Beets, Bert De Decker, Fabian Di Fiore, Maarten Dumont, Yannick Francken, Mark Gerrits, Tom Haber, Chris Hermans, Erik Hubo, Tom Jehaes, Pieter Jorissen, Kris Luyten, Tom Mertens, Peter Quax, Cedric Vanaken, William Van Haevre and Tom Van Laerhoven for their academic input as well as the much appreciated diversions. Ingrid Konings and Roger Claes of the EDM support staff also deserve a special mention as they made the bureaucratic entanglements considerably easier. I must also thank Heidi for the regular - and usually much needed - caffeine breaks.

My appreciation also goes out to Kathleen Denis and Ilona Stouten from the *XIOS Hogeschool Limburg*, for providing me with some real world datasets and helping me to gain insight in the subject of bent sheet metals.

Nietzsche once said that without music, life would be a mistake. I could not agree more, and my fellow members of the wind orchestra *Koninklijke Harmonie van Peer* probably could not either. The wonderful music played there and equally wonderful people were always a very welcome break from my academic contemplations. The same goes for my band members of the aspiring rock group *The Grams*: a tad more earsplitting yet equally uplifting. Also a big thank you to my non musical friends for their comradery in general and for offering me a listening ear when I needed one. Thanks to Sofie in particular for taking the proverb *mens sana in corpore sano* to new heights.

Last, but in no way least, I am extremely grateful to my family. My parents have been a constant source of encouragement, unwavering faith and support throughout the many days working on this thesis. I also want to thank my brothers Rik and Bert and their respective partners Anja and Yi-Jen , for their insights and support. Finally, I would be remiss if I did not thank my little niece Lien, just for being there and appreciating my general silliness when around her.

Abstract

Data segmentation involves the analysis and grouping of conceptually meaningful sections from the input, so that the individual segments contribute to an higher order understanding of the data as a whole. Usually, segmentation is a crucial first step towards further data processing. This dissertation discusses two segmentation techniques, both based on a hierarchical statistical analysis.

In the first part a technique is developed to segment motion vectors generated from moving images into independently moving entities. This segmentation is then used to automatically regenerate missing frames. This is especially useful for historical movies, as these often have been the subject of severe degradations, due to the susceptibility of celluloid to several degradations.

The second part deals with the planar segmentation of three dimensional point clouds. To this end, a statistical method is developed, based on principal components analysis and graph theory. The planar segmentation is then used in a number of industrial inspection problems, motivated by examples from the sheet metal bending industry. The dissertation is concluded with a hybrid rendering paradigm, combining both points and planar geometry into a single rendering framework.

Contents

List of Figures

List of tables

CHAPTER 1

Introduction

Contents

1.1 Motivation

The term *segmentation* in the general context refers to the partitioning of data into semantically distinct, non-overlapping groups or clusters, after which these meaningful components can be subjected to further manipulation. In most cases, data segmentation signifies a crucial first step towards a higher order understanding of the data as a whole.

Virtually every branch involving scientific data reasoning employs segmentation to gain insight in the structure of the underlying observations. For instance, successful speech recognition depends largely on an accurate initial

identification of the word boundaries in a spoken sentence. In market analysis, segmentation involves the identification of groups of people sharing consistent consumer profiles.

Due to the complexity of the segmentation process itself, it remains a topic of intense and ongoing research. This complexity arises partly because of the dependence of the analysis on the specific context within which the segmentation is performed. Furthermore, the quest for generality is complicated by the fact that the segmentation problem is rather ill-defined. For example, in image processing, segmentation may denote the grouping of regions which fulfill a certain color homogeneity criterium, but may also signify the separation of foreground objects from background regions, even though colors across boundaries might be reasonably similar.

1.2 Scope of the Research

This dissertation is titled *Statistical Segmentation for Computer Graphics* as we will discuss in detail two topics in segmentation involving the synthesis and analysis of visual data. First, we discuss the segmentation of motion data into distinct layers. This analysis forms the basis of a novel frame interpolation technique for historical films, from which it is possible to regenerate a number of missing frames for a damaged movie sequence. Next, we turn our attention to the segmentation of unstructured three dimensional point sets - or *point clouds* - into planar structures. This analysis is subsequently used for applications in industrial inspection and an efficient visualization paradigm.

Although every segmentation task takes some statistical properties into account distinguishing the individual components, the rigorous statistical concepts underpinning the techniques we developed motivated us to adapt the term *Statistical Segmentation*.

1.3 Contributions

This dissertation will introduce a number of novel ideas and techniques for statistical data segmentation in computer graphics. In this section we confine the presentation of our contributions to a brief summary. The introductory chapters for the two respective parts of this thesis will restate our original work in more detail.

- Chapter 2 introduces a novel motion and feature adaptive mesh warping algorithm, which is tightly coupled with a user interactive layer segmentation system.

- In chapter 3, we will present a new point cloud segmentation algorithm, which identifies planar clusters from 3D point sets. Our approach involves a combination of techniques from statistical multivariate analysis and graph theory.
- A novel 2D triangulation procedure is proposed in chapter 4. Our approach filters Delaunay triangles for a topologically correct reconstruction, and is specifically designed to handle variations in sampling density.
- Chapter 4 also establishes a new planar surface reconstruction algorithm. To this end, the density adaptive triangulation is supplemented with a new surface stitching algorithm.
- Hybrid rendering is introduced in chapter 5, replacing parts of point clouds with planar geometry to obtain an efficient representation and visualization paradigm.

1.4 Organization

This dissertation comprises two main parts. The first part deals with the segmentation of motion data in the context of historical films. In chapter 2 we present a complete system for the resynthesis of entire missing movie sequences. In the setting of movie heritage this addresses the destructive ageing affects archived celluloid film is often subject to.

In the second part, we discuss the planar segmentation of point clouds, as well as some applications for which the planarity assessment serves as an initial analysis. Chapter 3 presents a hierarchical PCA based method which identifies planar regions from an unstructured point set. This segmentation proves to be ultimately useful for point data recorded from manufactured objects and environments.

In chapter 4 the planar segmentation enables a structural analysis for bent sheet metals, for which the accuracy of bending and cutting operations, the two main forming processes, are estimated. We also introduce a surface reconstruction algorithm for piecewise planar point clouds, enabling the reverse engineering of planar parts.

Chapter 5 concludes the part on point cloud processing. In this chapter, a hybrid rendering algorithm is introduced which combines both point and polygonal primitives into a single framework. The motivation is that although pure point rendering offers a promising approach to visualize complex three dimensional data, a hybrid approach is in some cases more beneficial in terms of storage and rendering efficiency.

Each of these parts is preluded with a general introduction in which we lay out the problem setting, motivate the segmentation and restate our original work in that particular setting. At the end of each part, in combination with the concluding remarks we also give some directions for future research.

Part I

Frame Interpolation

Movie Restoration

Motivation

Classical film typically exhibits some forms of degradation, caused by the long time storage process. The celluloid medium on which movies have been stored, causes archive material to be subject to several types of chemical and mechanical aberrations. Improper storage environments can contribute to such artifacts as dust and dirt, locally polluting the film material. Other degradations, such as mechanical abrasions can affect an entire sequence of frames. Mold, which is also a common phenomenon in historic films, has an even more destructive impact on the film, and can mutilate entire sequences of frames.

All the material contained in various film archives constitutes a wealth of historic information, recording artistic and cultural developments spanning over a century, which is obviously essential to preserve. One effort towards historical film perpetuation is the transfer of archival material to a digital medium, which is made possible by the advances in the last decades in automated scanning technology. However, in addition to digitization, the affected material should also be restored to its original quality, preferably with an automatic approach.

There exists a large body of work on the restoration of degraded film material, most of which deals with the repair of individual frames, rather than the regeneration of lost frames. It is therefore not our intention, nor is it within the scope of this work, that a full treatise is given of these restoration

Figure 1.1: Typical frame defects in historical films. Top row: scratch defects. Bottom row: frame deterioration caused by mold. Images from [Schallauer 99].

techniques. We refer the interested reader to Schallauer's [Schallauer 99] or Kokaram's work [Kokaram 98], in which more general overviews on the topic of frame restoration are given.

Arguably the most common defect in historical film is the pollution of the frame content by dust and dirt which can be caused by an improper storage environment. Dust and dirt are manifested as the appearance of light or dark spots or *blotches*, which can be relatively easily removed with automated algorithms [Biemond 99] [Tenze 00]. Another frequently addressed problem is the automatic detection and removal of *line scratches* [Bretschneider 00] [Joyeux 99] [Besserer 04], which are caused by the contact of the film with mechanical parts of the projector. These artifacts tend to be somewhat more problematic to handle, since rather than having a local effect, scratches can extend over a number of frames.

Overview

Previous work on movie restoration mainly focusses on the repair of individual frame defects. In contrast, we address the most severe form of degradation, in which an entire sequence of frames is lost permanently, or degraded beyond local repair. To this end we propose a semi-automatic system, in which the user manually segments motion layers from input motion on both sides of the missing film. From this segmentation, lost frames are automatically regener-

ated by interpolating the motion layers and compositing the layers with an automatically generated and motion-compensated background layer. We will show how this hybrid approach results in a robust and general system for the reconstruction of missing film frames. The results of our approach will be demonstrated on a number of sequences from digitized archive film material.

Contributions

The next chapter introduces a complete user assisted system, targeted towards the recuperation of permanently lost sequences of historical film. The manual task of layer specification is based on the specification of two dimensional subdivision meshes with local interpolating characteristics. We will also introduce a novel motion and feature adaptive mesh warping algorithm which automatically generates the motion compensated background layer. The warping mesh is determined from a dense motion estimation, and takes the precision and uniformity of the underlying motion vectors into account.

CHAPTER 2

Frame Reconstruction in Historical Films

Contents

2.1 Introduction

Since the very beginning of cinematography, a historic wealth of information has been recorded and stored on film. Unfortunately, the very celluloid medium on which motion pictures are captured proves to be ultimately susceptible to severe degradation. These ageing affects should be reversed as much as possible. In addition, since manual restoration is a tedious and costly process,

an automatic or semi automatic system is highly desirable. Our reconstruction system addresses the most severe form of film degradation, in which complete sequences of frames are beyond repair by traditional restoration techniques, or are simply lost permanently. To this end, we propose a layered approach to the frame reconstruction problem, in which a reasonable amount of user interaction is allowed to completely resynthesize a given number of frames. We will demonstrate the results of our technique on a number of sequences from actual archive footage.

To the best of our knowledge, we are not aware of automatic frame regeneration techniques that specifically address the restoration of historical films. On the other hand, missing frame interpolation is a relatively well studied problem in the setting of video coding, and is used in video compression, where the high temporal redundancy of natural video sequences is exploited, as well as in other processing applications.

Frame rate conversion algorithms are concerned with retargeting a given sequence to a different frame rate. Applications include conversion to the 50Hz HDTV standard, or rendering a sequence in slow motion. Besides these applications, frame rate conversion is also commonly used in low bandwidth video coding, where a number of frames can be skipped [Han 97] [Liu 00b] which are then interpolated during the decoding stage.

The simplest techniques employ schemes such as frame repetition and frame averaging [Tekalp 95], which produce jerky motion and ghosting effects respectively, and are thus unsuitable for high end purposes.

To overcome these problems, more sophisticated frame interpolation approaches were developed, taking into account the motion trajectories between subsequent frames to arrive at a smoother transition. These motion trajectories are generated during the video encoding stage, and are derived from the motion estimated from individual pixel or entire block displacements. In either case, these motion vectors are likely to contain a certain degree of mismatched motion vectors. Outliers are filtered out to generate an approximation to the true motion vectors [Dane 04] [Blume 02], after which intermediate frames are generated by proportionally warping the reference frame by the filtered motion vectors. Although there exist techniques that take occlusion effects into account [Liu 00a], the underlying assumption is that two subsequent frames are relatively close to each other with respect to their moving features. Also, the difficult case of ambiguous motion near object boundaries, is insufficiently handled.

A more unified way of dealing with occlusions and motion boundaries is described by the layer based motion description, in which the notion of independent motion layers within an image is introduced. This representation

captures more of the salient features of real world motion, since an image is segmented into layers with depth ordering. Wang and Adelson [Wang 94a] pioneered the layered motion representation by segmenting a dense optical flow field into regions of coherent motion, which defines the individual layers. With this layered description, a moving sequence can be described with a relatively small number of independently moving layers, each with their own velocity map. The major limitation of the approach lies in the simplicity of the motion segmentation, which assumes an affine motion model. Also, the segmentation into layers is imperfect as pixels with inaccurate motion vectors might be assigned to the wrong layer.

Subsequent work on layering addresses the segmentation accuracy with various methods such as graph cut segmentation [Xiao 04] [Zhang 05], likelihood estimation [Jepson 02], or by imposing smoothness constraints [Nicolescu 03]. Next to addressing the layering problem within an EM framework, Smith et al. [Smith 04] also consider the relative depth ordering of the respective layers by tracking strong edges in a sequence.

2.2 Overview

From the previous section, we conclude that the layered motion representation presents the most promising framework for the regeneration of an entire missing sequence. However, in the context of frame resynthesis, the correct ordering of the layers depends to a great extent on the information that is available from the images that are used as a reference for the intermediate frame generation. Consider for example a sequence depicting an object traveling through a static scene, in which a scene object occludes the motion trajectory at some point of the sequence. Suppose that the missing frames are situated where normally the occlusions between the static and moving object are shown. In other words, the frames from which the synthesis is performed contain the moving object on either side of the occluder. In this situation, no single automatic algorithm will be able to correctly estimate the depth relationship between the moving object and the static scene.

In order to deal with complex layer interactions, we propose a semi-automatic reconstruction framework, in which the notion of independent motion layers is adopted. Allowing a reasonable amount of user input, our system is more robust in terms of complex layer occlusions. In addition, our technique attains a greater flexibility in terms of the motion model employed for the resynthesis of the intermediate frames, in contrast with the simplified affine motion model often assumed in the various automatic motion segmentation algorithms.

Figure 2.1 shows an overview of the proposed system, which takes as input

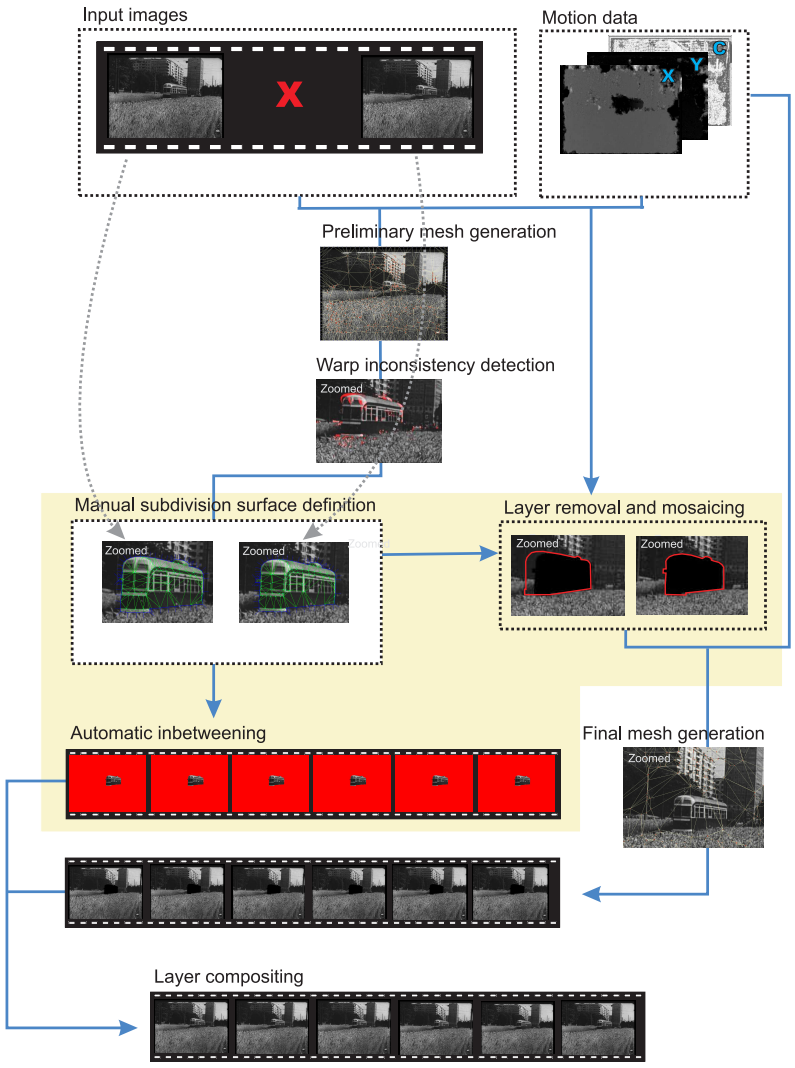


Figure 2.1: Schematic of the proposed frame interpolation framework.

Figure 2.2: Independent motion layer paradigm. Given a leading and trailing frame, independent motion layers are extracted by manual segmentation.

the two frames enclosing the missing frames together with a dense motion map relating these reference frames. This motion map can be calculated with any dense motion estimator, like for instance the well known Lucas Kanade motion estimation routine [Lucas 81]. The core of the automatic algorithm consists of a mesh warping routine. A triangular mesh is superimposed on the reference frame, after which the nodal points are displaced according to their associated motion vectors which are derived from the dense motion map, thereby deforming the underlying content to generate intermediate frames. The topology of the mesh is a crucial factor in the quality of the reconstructed frames, and the nodal points should preferably be concentrated on strong image features. The automatic mesh design is discussed in more detail in the next section.

User intervention involves the layer specification, in terms of both content and relative ordering. A specialized subdivision surface is defined over each object of interest in both reference frames, so that the vertices reside on corresponding object features. These pairs of related subdivision surfaces each define a motion layer, for which the user can indicate the relative ordering. Once the motion of independently moving objects is segmented using this procedure, inbetweens are generated for each separate layer, guided by the two interrelated meshes. During this process, the user can easily correct the nodal positions of each intermediate mesh, to ensure a physically plausible deformation.

The final stage of the frame reconstruction involves the compositing of each of the layered inbetweens, in accordance with the imposed order, with the automatically generated background layer.

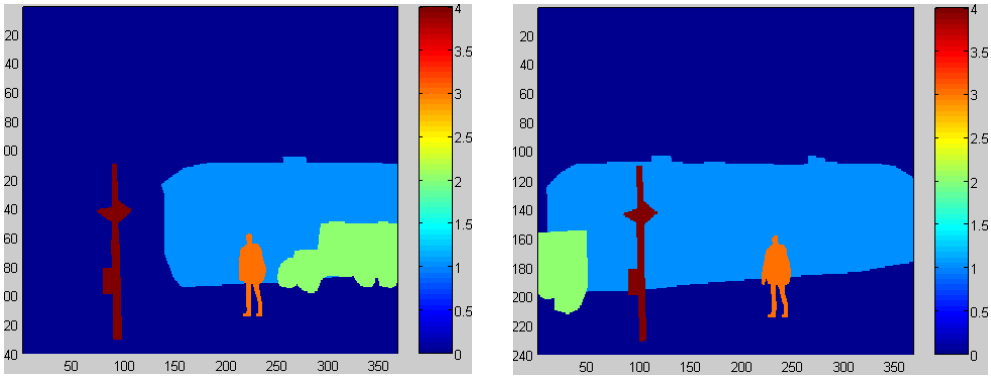


Figure 2.3: During layer specification, the user indicates the relative ordering of the layers.

2.3 Background Mesh Design

The background layer is basically treated as an elastic deformable sheet, which is realized by means of a two dimensional mesh. Intermediate frames are generated by displacing the vertices with a fraction of the associated motion vectors. More formally, if frames J and $J+N$ are the leading and trailing frames respectively, the intermediate frame $J+n$ is generated by displacing the nodal points (x_i, y_i) with associated motion (u_i, v_i) according to the following linear relationship:

$$\begin{aligned} x_i' &= x_i + \frac{n}{N} u_i \\ y_i' &= y_i + \frac{n}{N} v_i \end{aligned}$$

Content based mesh structures are commonly used for low bitrate video compression. Altunbasak's approach [Altunbasak 97] involves nodes selection, upon which a mesh is built for motion compensated warping. In contrast, Wang et al. [Wang 94b] propose a mesh design algorithm in which the motion guiding mesh is realized deforming an initially regular mesh, thereby minimizing an objective function. This mesh can eventually be split to allow for more detailed motion compensation [Wang 96]. This mesh based approach is also adopted in video object tracking [Valette 04].

The mesh topology is essential for the quality of the generated intermediate frames, and should be adaptive to both the image features and the underlying motion field. To this end, we impose three criteria for the design of the mesh:

Figure 2.4: Motion adaptive warping mesh, superimposed on a reference frame from a panning sequence. By weighing the candidate vertices with the image gradient, nodal points are placed on strong image features.

- The nodal points should preferably be placed on strong image features by using a measure such as the image gradient.
- The mesh should be adaptive to the underlying motion, that is, in regions with a large variation in motion direction and size, more mesh vertices should be placed with respect to those regions exhibiting a more uniform motion distribution.
- The motion vectors associated with the nodal points should have a high accuracy, to ensure the correct warped position.

These three constraints are addressed with an adaptive quadtree approach in combination with a motion confidence map. This confidence map is introduced, because the dense motion estimation is very likely to be affected by the noise contained in historical film material, even after manual or automatic cleanup. The motion confidence C gives per motion vector an accuracy likelihood, and is constructed by the normalized cross correlation of every pixel in one reference frame with its motion compensated counterpart in the other frame.

The quadtree is constructed using the distribution of the underlying motion field as a splitting criterion. At every level of the recursion, σ_v and $\sigma_{|v|}$

are determined for the quadrant under consideration, denoting the standard deviations of the motion vector direction and magnitude respectively. If either of the standard deviations do not satisfy the threshold τ, the quadrant is uniformly split. In the other case, thereby ensuring a uniform motion field is underlying the quadrant, a point within this quadrant is chosen as a nodal point. For each pixel (x_i, y_i) in the quadrant, a weight w_i is calculated by multiplying the motion confidence of the pixel's associated motion vector with the magnitude of the image gradient G:

$$w_i = b(x_i, y_i) \cdot C(x_i, y_i) \cdot \|G(x_i, y_i)\|_2$$

where b is a binary valued function, returning 0 if (x_i, y_i) lies within a distance r of a previously generated nodal point. The point (x_i, y_i) that maximizes the weight w_i is chosen as a mesh vertex.

Preceding the actual mesh generation, a number of uniformly spaced nodal points along the image borders are added to the points generated by the quadtree approach, to ensure complete image coverage of the mesh. The final mesh is constructed using a Delaunay triangulation.

The inaccuracy of motion estimation along image borders is a typical problem with dense motion estimation. Moreover, digitized historic films often show a black border surrounding the actual film frame as a result of the scanning process (for instance, see figure 2.4). Since we have appended nodal points along these image borders, we must derive accurate motion vectors instead of relying on the dense motion map. We do this by averaging the motion vectors of connected nodal points in the Delaunay graph. This propagates the accurate motion of the vertices generated with the quadtree approach to the border vertices, ensuring no stretching or folding artifacts occur during the warping to intermediate frames.

Since we are warping from one reference frame to interpolate the motion to the other frame, the motion of a border point may be such that the warped frame does not completely cover the entire image. By mosaicing [Shum 97] the two reference frames, we obtain a complete image of the background. This mosaiced frame is then used for generating intermediate frames (figure 2.3).

2.4 Layered 2D Mesh Based Morphing

The morphing or metamorphosis of images or image parts is often used to generate special effects in films and animation. Within the setting of layered inbetweening, we use an adaptation of the basic notion of warping.

Previous work in mesh based morphing, as described in [Wolberg 90], assumes a uniform mesh of points in the source image, which define a grid of

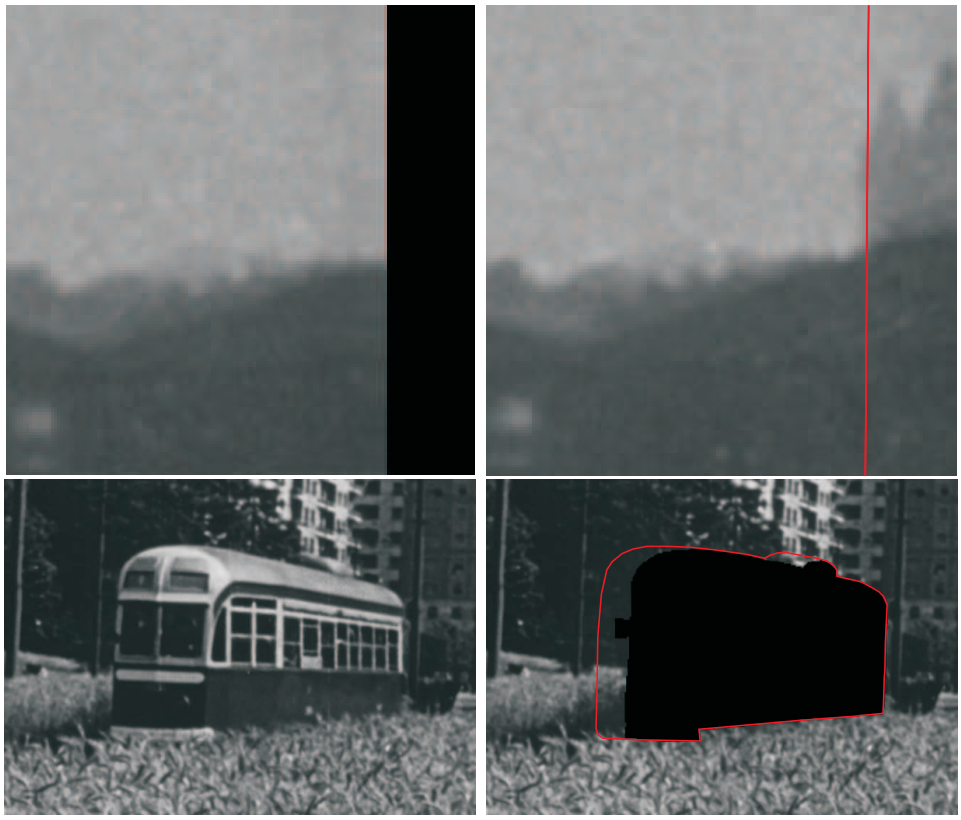


Figure 2.5: Reference frame mosaicing for image completion. Aligning the two leading and trailing frames into a common reference frame, missing data due to inward warping (top) or layer extraction (bottom) can be partially filled in.

Figure 2.6: Layered mesh specification. For each motion layer, a pair of corresponding meshes is defined in the reference frames, thereby establishing a layered motion trajectory.

control points. The correspondence with a target image is defined, by manually adjusting these control points to match the features between frames. Subsequent work addresses the tedious manual and error-prone feature relating, for instance by automating the process with active nets [Nishita 93], snakes [Lee 96], by using pairs of directed line segments [Lee 98], or by specifying outline curves [Tal 99][Johan 00].

The general problem of interpolating between two dimensional shapes is still an active domain of research. For our purposes, we adapt the approach of [Van den Bergh 02], in which the morphing between general objects with concavities, holes or protruding limbs is specifically addressed. Moreover, local control can be maintained by the use of subdivision techniques.

For each of the corresponding objects of interest, a motion layer is defined by segmenting the objects with an enclosing polygon. This polygon is then triangulated and refined with locally interpolating subdivision techniques. Note that this mesh is created for only one reference frame. The motion layer correspondence is established by placing the same mesh on the other reference frame, and manually adjusting the vertices to match the image features. Figure 2.4 shows an example of the motion layer mesh pairs.

Using the layer correspondences, intermediate frame layers are automatically generated by piecewise affine warping, as in the previous section. Intermediate frames can be further refined by the manual adjustment of vertices, thereby allowing more general object transformations.

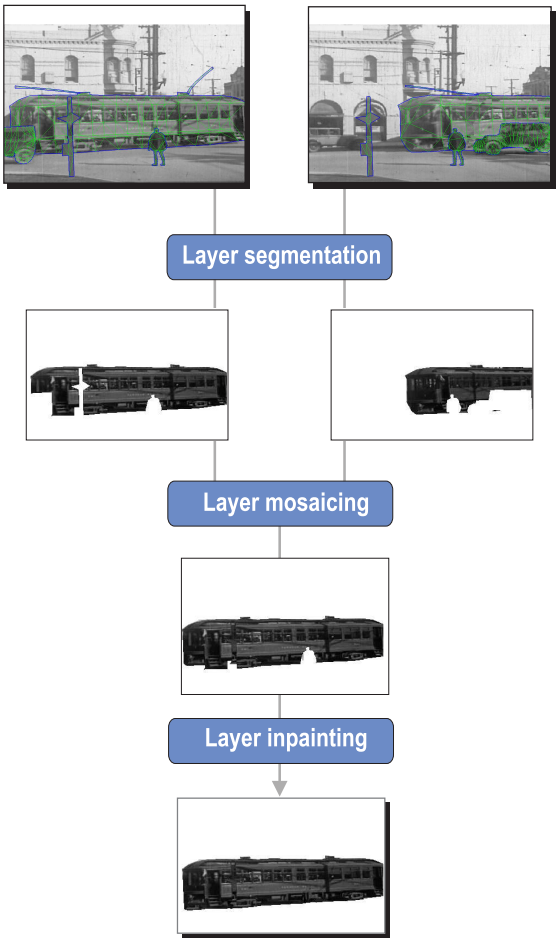


Figure 2.7: For each motion layer, a layer sprite is generated using mosaicing and image inpainting.

Figure 2.8: Warping error identification to aid in the definition of layers.

2.4.1 Layer Identification

Note that layers do not always need to be specified to generate a correctly interpolated sequence. Depending on the situation, the automatic background warping may generate the correct results. This is for instance true for a simple panning sequence. As a visual aid to the user, layers are indicated using a preliminary warp with the background mesh using the underlying motion field, thereby registering the leading frame with the trailing frame. A simple error map can be constructed taking the greyscale of the trailing frame, expanding it to RGB space, and replacing the R channel with the warped greyscale leading frame. This is illustrated in figure 2.8.

2.5 Compositing

The final step of the frame reconstruction consists of combining the various inbetweens of the layers into a composite sequence. After the background layer warps are generated with the quadtree mesh, the interpolations of the user defined layers are composited with these frames, in the order specified during the layer modeling step . Note that the background warping uses the mosaiced reference frames (see figure 2.3), for which the confidence map is set to zero for pixels which are occluded by a user defined layer. This way, the quadtree mesh design algorithm does not generate any nodal points on pixels belonging to these layers. The mosaicing step ensures that during the final compositing no borders of the cut out regions of the background layer are shown.

Figure 2.9: Simple alpha map construction, by convolution of a binary layer mask with a Gaussian kernel.

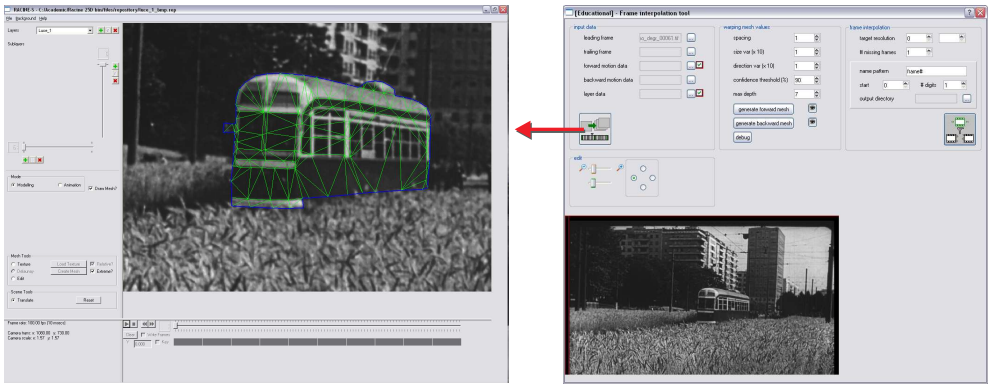


Figure 2.10: The GUIs for the frame reconstruction system. Depicted on the right is the main interface in which the leading and trailing frames, the layer specification, and interpolation parameters are specified. The left GUI shows the layer specification interface.

Also, each individual layer is completed using a combination of mosaicing and inpainting [Criminisi 03]. This is demonstrated in figure 2.7. This inpainting step is necessary to complete those regions of a layer that are occluded in the leading and trailing frame, but are exposed during the motion interpolation.

In order to smoothly blend the various layers onto a final frame, a simple alpha map is created for each segmented layer, by convolving the binary mask of that layer with a Gaussian kernel. This is demonstrated in figure 2.5.

2.6 Results

In figure 2.11 we present the results of our approach on a number of sequences from digitized archive footage. The examples are ordered in order of complex-

Figure 2.11: The results of our algorithm. The first and fourth row show the leading and trailing reference frames respectively. For each sequence, two intermediate frames out of 6(two leftmost columns) and 41(right column) missing frames are shown.

ity and amount of user interaction. The movies for these examples can also be downloaded from our interpolation project web page [J. Fransens 05].

The first example depicts the results for a simple panning sequence. In this example, 6 frames were fully automatically resynthesized, by warping the background layer. The resynthesis of the second example, for which the reference frames are also 6 frames apart, was generated by defining an extra layer containing the tramcar, moving in the opposite direction of the camera pan. For the last sequence, a total of 41 frames were reconstructed from the two reference frames and the definition of 3 layers, composited on the background. As can be seen from the figure, the recovered intermediate frames show the correct occlusion ordering.

Movie Restoration: Concluding Remarks

Summary

In the previous chapter, we have presented a layered approach for the complete resynthesis of entire sequences of historic film material. To this end, a hybrid system was proposed, with a tightly coupled interaction between fully automated reconstruction and manual intervention. A scene is decomposed into a number of motion layers, by manually establishing correspondences between objects of interest in the leading and trailing frame. Using this information, the motion layers are independently morphed and composited onto the automatically generated background layer. The background is piecewise affine warped, guided by a mesh structure that adapts itself to the underlying motion and the image features, in addition to an estimation of the relative motion accuracy. We have demonstrated the results of our approach on a number of actual archive films.

Future Work

In this section we identify some possible directions for further research on the layered frame interpolation.

Our current approach involves using two reference frames to guide the reconstruction of intermediate frames with a linear warp. The accuracy of this warping could be improved if more frames were to be taken into account, allowing for more general motion paths. Three dimensional view interpolation

[Zitnick 04] could in this case also be used to synthesize the motion compensated layers.

At the moment, our segmentation procedure employs polygonal meshes to cut out the individual layers. This might be refined using more advanced video segmentation tools, such as the recently proposed Video Cutout procedure [Wang 05]. This also relates to the layer compositing stage: the simple convolution mask might be refined using a more advanced alpha matting algorithm [Xiao 05].

Part II

Planar Segmentation for Point Clouds

Point Cloud Processing

In the last part of this dissertation we turn our attention to the segmentation of so called *point clouds*. A point cloud can be defined as a set of three dimensional points with no further connectivity information. The natural representation of point cloud data is an unstructured list of 3D point coordinates, possibly embellished with normal, color and size information.

The point cloud paradigm contrasts sharply with the more traditional polymesh representation, which describes a surface as a collection of higher order geometric primitives such as triangles or NURBS patches, whereas the scattered data representation of point clouds denotes a sampling of the underlying surface.

Although the use of point primitives for surface description was originally proposed by Levoy and Whitted as far back as 1985 [Levoy 85], point sets have only fairly recent become the subject of significant research efforts, due to a number of reasons: the advent of relatively inexpensive range and laser scanners, for which the point cloud representation is the natural output, has fueled a lot of research in the representation, modeling, compression, editing and filtering of 3D scattered data sets. These operations are also significantly simplified by the superfluity of topological bookkeeping, as opposed to the traditional polygon soup processing techniques. Furthermore, for highly detailed point models the rationale behind polygon rendering vanishes altogether, as the dense sampling makes the connectivity redundant, since a single point primitive contributes less than a pixel during rendering.

Figure 2.12: Some typical examples of scanning methods. From left to right: Airborne LIDAR sensing, touch probe acquisition and laser scanning. Photographs courtesy of NASA, Institut de Production et Robotique and Metris respectively.

Scanning

A number of different techniques exist for the acquisition of point clouds. The capturing of three dimensional structures can be subdivided into two different categories: on the one hand, *contact* or *tactile* scanners use a touching probe to sense the depth of individual surface points. These scanners have the advantage of being extremely accurate, at the cost however of operation speed and the inability of handling large or delicate surfaces, which makes this method unfit for scanning large buildings or archeological artifacts, amongst others.

On the other hand, *non-contact* scanners, operate on the principle of emission of radiation such as visible light, X-rays or sound. Non-contact scanners can be subdivided in *active* and *passive* systems. Active scanners deduct the depth of a scanning point by measuring the time-of flight of reflected radiation from the scanning device. Although this enables rapid scanning of large structures some precision is lost, particularly if the measuring apparatus is far away from the subject. Passive scanning does not rely on the emission of radiation of the scanner itself. Rather, the ambient light reflected of the scene is detected from which a model is to be created. Examples of this method include multi-camera photogrammetric approaches and user-assisted image based modeling techniques. Passive systems can be very cheap, because in many cases, no specialized hardware is required.

Applications

The three dimensional scanning of objects has important applications in such diverse areas as industrial inspection, terrestrial observation, the entertainment sector and cultural heritage preservation.

The digitizing of industrially manufactured parts has become an invaluable asset in the production pipeline. In many instances, the precision of the manufacturing process is assessed using a scanned representation of the object. The recorded point cloud is compared to a digital representation from which the object is produced, such as a CAD model, to measure how well it conforms. This enables a rapid quality assessment, from which the necessary steps can be taken to correct potential problems. *Reverse engineering* of parts is also considerably simplified using 3D scanners, as the point cloud of a model is readily obtained, which is then easily reproduced.

The acronym LIDAR (*Light Detection and Ranging*) is often used in airborne range sensing, in this context also referred to as ALSM (*Airborne Laser Swath Mapping*). This technology is used in a multitude of areas such as geology, seismology, remote sensing and atmospheric physics. For instance, airborne sensing enables researchers to monitor the evolution of the sea level or perform rapid change detection.

Also in the entertainment sector, three dimensional digitizing plays an increasingly important role. 3D scanners are employed to create models for both movies and video games. Frequently, video game artists first sculpt a physical object, from which a digital counterpart is created using scanning technology. For movies, actors and sets are often digitally cloned for the purpose of special effects.

3D scanning technology presents an immense opportunity to create digital copies of real world objects, which is crucial for cultural heritage preservation and dissemination. A notable cultural heritage project is the *Digital Michelangelo Project* [Levoy 00], for which a number of researchers set out to scanning Michelangelo's statues in Florence, which include the famous *David* statue.

Motivation and Overview

The last part of this dissertation discusses the segmentation of raw point clouds in meaningful subparts. More specifically, we concern ourselves with the *planar* segmentation of point clouds. The planar subset of the segmentation problem is of great interest, as many man-made objects and environments typically exhibit some form of planar regularity. For instance, buildings are usually largely made up of planar structures.

In the next chapters, we introduce our planar point cloud segmentation technique, which is capable of identifying planar subsets in unstructured point clouds. To this end we propose a novel hierarchical segmentation algorithm.

This planar segmentation serves as an initial analysis for three applications. First, we will discuss the inspection of so called *folded metal sheets*. Next, a surface reconstruction for piecewise planar point clouds will be presented. Finally, we will review a hybrid point and polygon rendering system, which exploits potential planar structures present in the point data.

Contributions

The next few chapters will introduce some new ideas and techniques for point cloud processing. Our major contribution is a novel hierarchical planar point cloud segmentation algorithm, which uses ideas from statistical processing and data clustering to arrive at the final segmentation.

Next, an addition to the well know Delaunay triangulation procedure is proposed, which successfully handles holes and concavities in the point data. To this end, a *Kernel Density Estimation* is performed. This technique is used for a novel planar surface reconstruction algorithm, which triangulates a planar point cloud by polygonizing each planar cluster in turn, after which the planar parts are *stitched* together. This separation allows for a separate triangulation in two dimensions for each plane, which significantly reduces the complexity of the surface reconstruction problem.

The final contribution is a novel hybrid rendering system, which employs both polygons and point primitives for displaying complex three dimensional models. This technique allows for an optimal trade-off between polygon rendering for perceptually planar parts in the point cloud, and point primitives for handling the more detailed point data regions.

CHAPTER 3

Perceptual Grouping of Planar Structures in Point Clouds

Contents

3.1 Introduction

The problem of data segmentation can be summarized as the decomposition into simpler constitutes, so that these subparts convey meaningful information about the data as a whole, which is a crucial first step towards higher level data understanding.

In the context of three dimensional data, the segmentation task usually involves the decomposition of the data into low-level primitives, which are then used for further analysis. The three dimensional segmentation problem has sprouted a huge volume of literature, because of the many computer vision tasks benefiting from this analysis.

Over the years, the particular problem of range image segmentation has received a lot of attention [Hoffman 87] [Hoover 96], mostly driven by robotics applications for industrial inspection. Another important problem in the structural understanding of three dimensional data is the analysis of airborne laser scanner data [Vu 01] [Sithole 05].

Of particular interest is the extraction of planar primitives from spatial data [Jiang 94] [Cobzas 01] [Nunes 05]. This is motivated by the high degree of planar regularity found in many man-made objects and environments such as buildings and industrially manufactured machine parts.

In this chapter, we propose an algorithm to efficiently extract planar regions from a general point cloud. Our technique is similar to that of Wahl et al. [Wahl 05] in the sense that we also use a hierarchical subdivision to determine local planarities. However, we extend this idea with a global planar clustering and segmentation refinement algorithm.

3.2 Related Work

3.2.1 2.5D Segmentation

The idiom 2.5D refers to three dimensional data which is elevated over a regular grid, and is also commonly denoted as height field data. Because of the gridded acquisition, segmentation techniques for these data often have close connections with image processing methods. The two most commonly used capturing devices for 2.5D data are range scanners and LIDAR sensors.

Range scanners are usually employed in an industrial setting and have become of fundamental importance in tasks such as autonomous navigation and reverse engineering [Gotardo 04]. Earlier work on range image segmentation was mainly focussed on robotic applications in assembly lines, for which the recognition of polyhedral objects was a main concern. Work on the evaluation

of these various range image segmentation methods includes the evaluations done by Hoover et al. [Hoover 96] and Jiang et al. [Jiang 00]. More recent work focuses on the segmentation of more complex scenes such as range data captured from outdoor scenes including trees and bushes [Han 04].

For planar segmentation of range images a number of approaches have been proposed in the literature. For instance, Jiang et al. [Jiang 94] segment individual scan-lines into line segments after which these segments are merged into planar regions. In contrast, Checchin et al. [Checchin 97] employ a more global approach and use a hybrid edge detection and region growing technique. Dell'Acqua et al. [Dell'Acqua 02] also determine an initial depth discontinuity map, after which planes are fitted to regions enclosed by the depth edges. Gottardo and coworkers [Gotardo 04] employ local neighborhood classification after which a robust estimator is used to extract the surfaces. In Lakaemper's work [Lakaemper 06] a modified EM iteration fits planar patches to $3D$ laser range data obtained by a mobile robot.

Automatic structure detection also play an essential role in the analysis of LIDAR data [Schuster 04], for which the automated extraction of meaningful parts such as roads, vegetation and roofs is of interest. Planar segmentation is used for the detection and reconstruction of buildings and roof tops [Vosselman 99] [Rottensteiner 03] [Peternell 04].

3.2.2 3D Segmentation

The segmentation of general point clouds, that is, point data without underlying structuring domain is of particular interest for reverse engineering applications [Várady 97] [Langbein 01]. Common techniques for general point set segmentation fall into the bottom-up or region growing category [Gelfand 04], but more global techniques such as direct extraction of parameterized shapes using a Hough Transform have also been proposed [Vosselman 04].

The fundamental importance of planar structures in real-world spatial data has also motivated some research to purely planar extraction from general point clouds. For instance, RANSAC based schemes have proven useful in the detection of planar structures. Bauer et al. [Bauer 03] use an iterated global RANSAC search around a seed point to extract the dominant planar features in a point cloud. This approach is combined with a hierarchical subdivision by Wahl et al. [Wahl 05] as a preprocessing stage for an efficient rendering algorithm.

3.3 Overview

In this section we give a general overview of our segmentation algorithm, see figure 3.1 for a graphical rundown. In the next section the details of this analysis will be discussed.

Given a set of samples, planar parts are extracted using principal component analysis (PCA) on an increasingly refining octree. At any stage of the recursion, the data contained in the octant under consideration is tested for planarity, comparing the eigenvalues of the covariance matrix. If the data is found to be planar, no further subdivision of the octant is performed, and its data is grouped into a planar region. Otherwise, the octant is split and the algorithm recurses on the children.

Next, planar regions having similar plane parameters are grouped together with a simple clustering technique. To this end, a minimum spanning tree (MST) is constructed, using the extracted planar regions as nodes. The edge weights are determined by the average Mahalanobis distance between the planar regions it connects. The MST will link coplanar regions with a low weight, while transitions between non coplanar regions are penalized with a larger weight. Planar regions are merged together by iteratively cutting away the higher weighted edges, until all resulting subtrees' nodes are coplanar.

The planar segmentation is finalized by further expanding the planar clusters with unsegmented samples that are sufficiently close with respect to the Mahalanobis metric and have similar normals.

The combination of techniques we employ to generate full planar clusters can be thought of as a hybrid top-down and bottom up approach. The top-down hierarchical analysis enables to quickly determine large scale planar patches, and the Mahalanobis refinement extends the clustered patches in a single iterated region-growing fashion.

The rest of this chapter is organized as follows: section 3.4.1 gives a detailed treatment of the hierarchical PCA analysis. In section 3.4.2 we discuss the MST based planar clustering. Section 3.4.3 discusses the refinement of the original planar segmentation. Section 3.5 concludes this chapter with some final remarks.

3.4 Planar segmentation of point clouds

3.4.1 Principal Component Analysis

Principal Component Analysis [Jolliffe 02] [Johnson 98] is used to assess the planarity of the point coordinates P contained in the octree cell under consid-

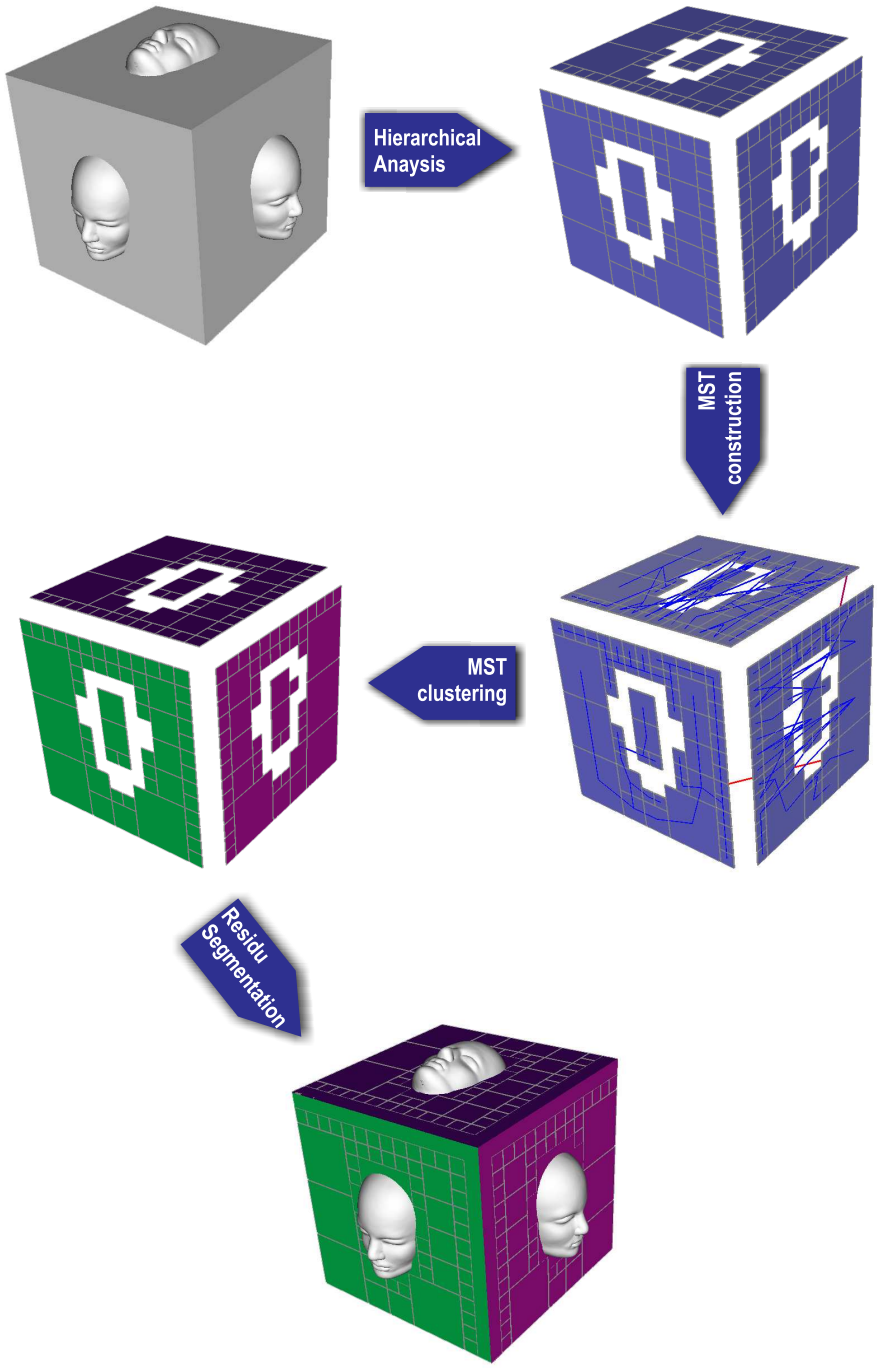


Figure 3.1: Overview of our planar segmentation procedure.

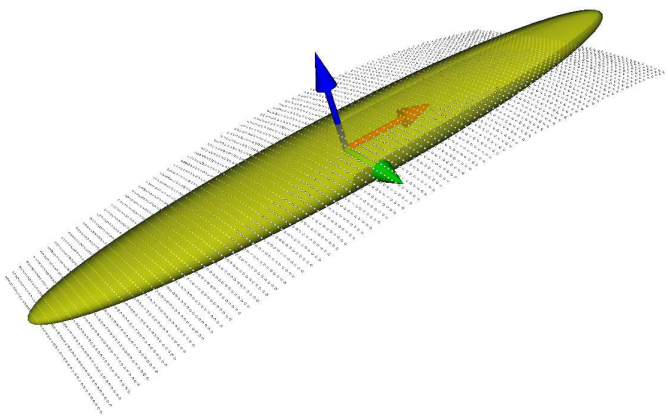


Figure 3.2: The PCA of a set of 3-dimensional points yields the principal vectors describing an orthonormal frame f. An equidistance ellipsoid with respect to the Mahalanobis metric is shown in yellow.

eration. The PCA of a set of 3-dimensional points yields the principal vectors describing an orthonormal frame $f = (f_1, f_2, f_3)$ and the principal values vector $\sigma = (\sigma_1, \sigma_2, \sigma_3)$ so that $\sigma_1 \geq \sigma_2 \geq \sigma_3$ [Duda 00]. The frame f implies a change of basis, in which the standard deviations along the respective axes are given by σ. The PCA decomposition of a set of points P is calculated as the SVD of the covariance C of the data samples, with

$$C = \frac{1}{n-1}\big(P - m\big)^\top \big(P - m\big) \tag{3.1}$$

where m denotes the mean of the data, and n the number of points in P.

For general n-dimensional problems, PCA is often used as a tool for dimensionality reduction, by projecting the original data on the subspace spanned by the first k principal vectors. The accuracy of this projection is expressed as the loss in variance, determined by the ratio of the unused variance and the total variance:

$$\frac{\sum \sigma_{k+1:n}}{\sum \sigma_{1:n}}$$

Reverting back to the 3-dimensional setting, we consider the points contained in a given octree cell as planar, if the total amount of variance lost due to the projection of the points onto the plane spanned by the first two

principal vectors satisfies the following relationship :

$$\frac{\sigma_3}{\sigma_1 + \sigma_2 + \sigma_3} < \tau$$

where $\tau \epsilon [0..1]$ is a threshold value, loosening the planarity condition with higher values of τ, which can be useful in analyzing noisy point cloud data.

3.4.2 Planar Clustering

The second step toward a full planar segmentation of the data points is the aggregation of the extracted planar patches into planar clusters. These planar clusters comprise planar patches with similar plane parameters. We approach the merging of planar patches as a clustering problem, which can be solved with minimal spanning tree analysis. A minimum spanning tree of a weighted graph G is the minimum-weight, connected, acyclic subgraph G' containing all nodes of G.
For patch merging, the initial graph G has one node per planar patch and is fully connected. The choice of an appropriate weight function is crucial for successful clustering, and should in this case account for the global plane parameters, as well as the average distance between the samples of any two patches. Instead of constructing a measure that conglomerates these parameters, we use the average linkage distance [Kaufman 90], in combination with the Mahalanobis metric:

$$w_{ij} = \frac{1}{n_i + n_j}\Big(n_j \sum_{s \in \mathcal{P}_j} d(s, \mathcal{P}_i) + n_i \sum_{s \in \mathcal{P}_i} d(s, \mathcal{P}_j)\Big) \tag{3.2}$$

with n_i and n_j the number of samples in the planar patches $\mathcal{P}_i$ and $\mathcal{P}_j$ respectively, and $d(s, \mathcal{P})$ the Mahalanobis distance [Jolliffe 02] between a sample s and a planar patch $\mathcal{P}$:

$$d^2(s, \mathcal{P}) = (s - m_{\mathcal{P}})^\top C_{\mathcal{P}}^{-1}(s - m_{\mathcal{P}}) \tag{3.3}$$

with $m_{\mathcal{P}}$ the mean of $\mathcal{P}$ and $C_{\mathcal{P}}$ the covariance of the samples in $\mathcal{P}$.

In contrast with the Euclidean distance, the Mahalanobis distance does take the sample variability into account. Instead of treating all values equally when calculating the distance from the mean point, it weighs the differences by the range of variability in the direction of the sample point. We can therefore consider the Mahalanobis distance as the inverse probability the sample s belongs to the $\mathcal{P}$ population.

```
//construct tree
Tree T;
T.nVertices = nPatches;
for i=1:p
    for j=i:p {
        weight = mahab_linkage(i,j);
        T.SetEdge(i,j,weight);
        T.SetEdge(j,i,weight);
    }
}
//compute mst
[edgesij,weights] = T.determineMST();

//sort weights, permute edges
sort(weights,edgesij);

//total planarity assessment
bool planar = IsPlanar(AllPatches);
nCuts = 1;

while(!planar){
    for c=1:nCuts
        cutlist.add(edgesij[c]);
    SubTrees = T.cut(cutlist);
    for s=1:nSubTrees
        planar = planar & IsPlanar(Subtrees[s])
    nCuts++;
}
```

Algorithm 3.1: Pseudo code listing for the planar clustering algorithm.

With the average Mahalanobis linkage weights, we can build the graph G and compute the minimum spanning tree G': A fully connected graph G is constructed with the planar patches $\mathcal{P}$ as vertices and with edge weights determined with equation (3.2). The MST of G is determined using Kruskall's algorithm [Skiena 98].

The planar clusters are determined by iteratively cutting the minimum spanning tree G' along the higher valued edges, until all resulting subgraphs consist of co-planar patches. At this stage, every subgraph defines a planar cluster. Listing 3.1 gives the pseudocode for the planar clustering algorithm.

Subgraph planarity assessment

Instead of recalculating the entire PCA for the samples referenced by a subtree, we can reuse the results of the hierarchical planarity assessment.

In the three dimensional setting, the eigendecomposition is performed on a 3×3 matrix. This decomposition is insignificant in terms of time and resource complexity, especially when compared to the evaluation of the covariance ma-

trix, so we focus on the combination of the covariances of the patches $\mathcal{P}$. In order to do so, we first have to rewrite the covariance formula (3.1):

$$C = \frac{P^\top P}{n-1} - n\frac{(m \cdot m^\top)}{n-1} \equiv S - n\frac{(m \cdot m^\top)}{n-1}$$

Lemma 3.4.1 *Suppose we have two planar patches $\mathcal{P}_1$ and $\mathcal{P}_2$ with known covariances C_1 and C_2, means m_1 and m_2 and with respective sizes n_1 and n_2. The union of the points contained in $\mathcal{P}_1$ and $\mathcal{P}_2$, denoted by $\mathcal{P}_1 \cup \mathcal{P}_2$ has $S_{\mathcal{P}_1 \cup \mathcal{P}_2}$ and $m_{\mathcal{P}_1 \cup \mathcal{P}_2}$ terms which are linear combinations of the covariance factors from the individual patches:*

$$\begin{aligned} m &= \frac{n_1 m_1 + n_2 m_2}{n_1 + n_2} \\ S &= \frac{(n_1 - 1)S_1 + (n_2 - 1)S_2}{n_1 + n_2 - 1} \end{aligned} \tag{3.4}$$

Proof For the global mean m we have

$$m = \frac{1}{n}\sum_{i=1}^{n} p_i$$

with p the set of n points in $\mathcal{P}$. This can easily be rewritten as a compound mean using substitution:

$$m = \frac{1}{n_1 + n_2}\Big(\sum_{i=1}^{n_1} p_{1i} + \sum_{j=1}^{n_j} p_{2j}\Big) = \frac{n_1 m_1 + n_2 m_2}{n_1 + n_2}$$

For the global S term we have:

$$
\begin{aligned}
S &= \frac{P^\top P}{n-1} \\
&= \frac{\left(\frac{P_1}{P_2}\right)^\top \left(\frac{P_1}{P_2}\right)}{n-1} \\
&= \frac{\left(\begin{array}{c|c} P_1^\top & P_2^\top \end{array}\right) \left(\frac{P_1}{P_2}\right)}{n-1} \\
&= \frac{\begin{pmatrix} \sum_{i=1}^{n_1} p_{1i}^1 p_{1i}^1 + \sum_{i=1}^{n_2} p_{2i}^1 p_{2i}^1 \\ \vdots \\ \sum_{i=1}^{n_1} p_{1i}^d p_{1i}^d + \sum_{i=1}^{n_2} p_{2i}^d p_{2i}^d \end{pmatrix}}{n-1} \\
&= \frac{P_1^\top P_1 + P_2^\top P_2}{n-1} \\
&= \frac{(n_1-1)S_1 + (n_2-1)S_2}{n_1+n_2-1} \qquad \blacksquare
\end{aligned}
$$

These formulas are easily generalized to more than 2 patches. Since the dominant calculation during three dimensional PCA is the determination of the covariance, the planarity of an entire subgraph can be determined very efficiently with formula (3.4).

3.4.3 Segmentation Refinement

As a result of the spatial subdivision of the octree-based PCA, not all points lying on a topological plane are included in an aggregated cluster. For instance, an octree leaf cell comprising the junction of two surfaces will not be classified as being planar. The planar segmentation is finalized by extending each cluster with points not contained in any planar cluster, if they conform to the statistics of the cluster.

To this end, we compute the Mahalanobis distance of every residue point p_r with every planar cluster using (3.3). The point p_r is added to the cluster $\mathcal{C}$ it has the minimal Mahalanobis distance d with iff d satisfies a threshold τ_r and the normal of p_r lies sufficiently close to the normal of $\mathcal{C}$, to avoid cutting through the point cloud (see figure (3.3)).

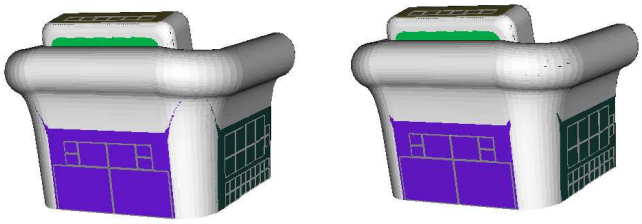


Figure 3.3: Effect of the normal constraint on the per point segmentation. The model on the left is segmented using the Mahalanobis distance metric, without normal constraint. In this case the segmentation cuts through the point data. This is solved using a normal similarity condition (right image).

3.4.4 Handling Noise

The algorithm presented so far works well, on the condition that the point cloud is essentially free of noise and outliers. This is because of the adverse affect noisy and contaminated data have on the principal component analysis, since PCA is in essence a least squares method. However, when working with real world scanned point sets, the data is often influenced by these unfavorable conditions.

To remedy this, we have opted to filter the point cloud in a preprocessing step as opposed to using for instance a robust PCA classifier [Hubert 05]. To this end, we employ Dey's NormFet software [Dey 05b] to estimate the normals for the potentially noisy point cloud. In doing so, not only reliable surface normals are generated, but potential outliers are also identified by the software's inability to generate a normal estimate for these points. Next, the same author's AMLS software [Dey 05a], based on the principle of a moving least squares surface [Alexa 01], is used to denoise the point data.

3.5 Results

In this section, we validate our planar segmentation algorithm on a number of different point clouds. Figure 3.5 shows the results on a variety of point data. The first two data sets are synthetically generated from polygonal models. The latter two are acquired with a three dimensional laser scanner. For each model, a graphical depiction is also given of the intermediate planar patches,

with the MST graph used for clustering overlaid.

The respective statistics for the different models are presented in table 3.1. These statistics include the number of points of each point cloud, along with the running times of the three consecutive stages of the segmentation algorithm:

- **Stage I** is the hierarchical planarity assessment, in which planar patches are extracted.
- **Stage II** consists of the MST graph construction and iterative clustering algorithm.
- **Stage III** refines the planar clusters with the Mahalanobis based region adjoining.

The timings are presented as they were recorded on a 1.6 Ghz. Pentium M machine with 1 Gigabyte of internal memory.

The proportional workload of the individual stages with respect to the total running time of the segmentation are given in figure 3.4. It can be clearly seen that the major complexity lies in the top-down planarity analysis. The partition of the workload depends mainly on the distribution of planar features compared to the total number of points.

Model	**n**	**Stage I**	**Stage II**	**Stage III**	**total**
Casting	372152	3261	182	931	4374
Block	546812	5748	389	970	7107
Wega	107974	1272	128	59	1459
A2	656247	6759	86	114	6959

Table 3.1: Statistics for the planar segmentation algorithm. The models appear in the same order as in figure 3.5. Timings are in milliseconds.

3.6 Conclusion

We have presented a novel planar segmentation algorithm for point clouds, which combines elements from top-down analysis and bottom-up gathering. The crux of the algorithm is a hierarchical planarity assessment using Principal Component Analysis. Planar patches are merged into planar clusters using a Minimum Spanning Tree pruning technique which iteratively cuts the edges of

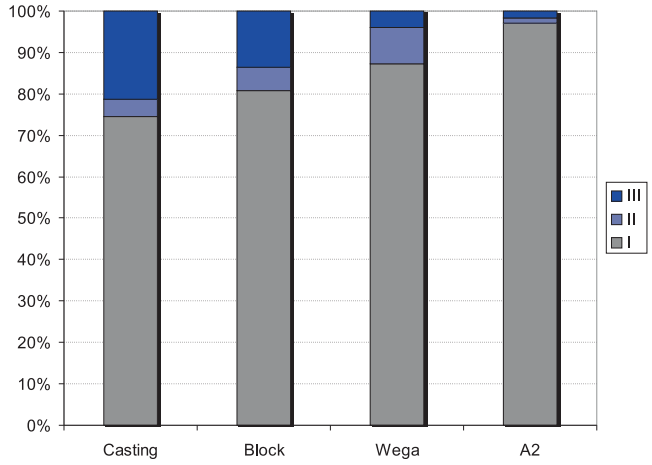
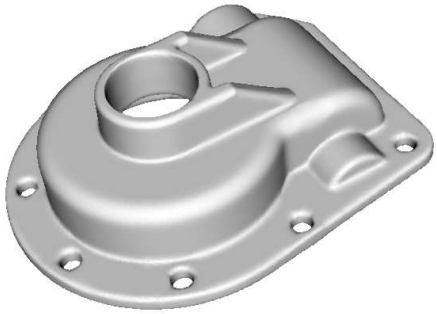
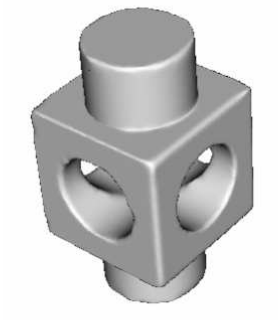
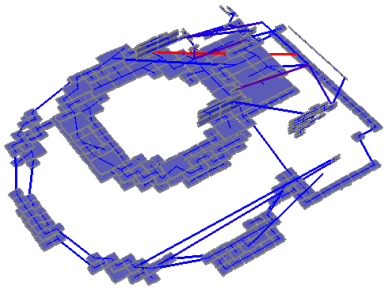
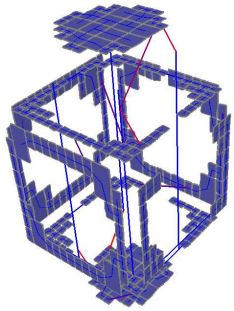
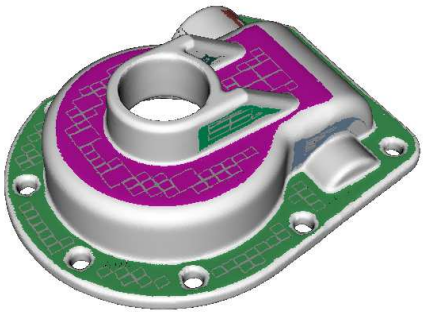
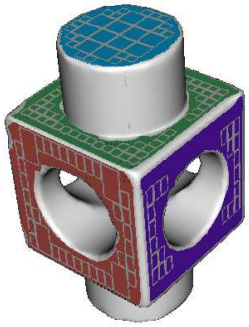


Figure 3.4: Proportional distribution of the different stages of table 3.1.

the graph until all resulting subgraphs link coplanar patches. The coplanarity is assessed using an efficient combination of the PCA results gathered from the octree analysis. The segmentation is finalized using per point inclusion based on the Mahalanobis metric. An initial filtering of the point data ensures the point cloud is essentially noise-free, so that the segmentation is a precise as possible. We have demonstrated the results of our approach on a number of different point sets to demonstrate the efficacy of the algorithm in a variety of situations.

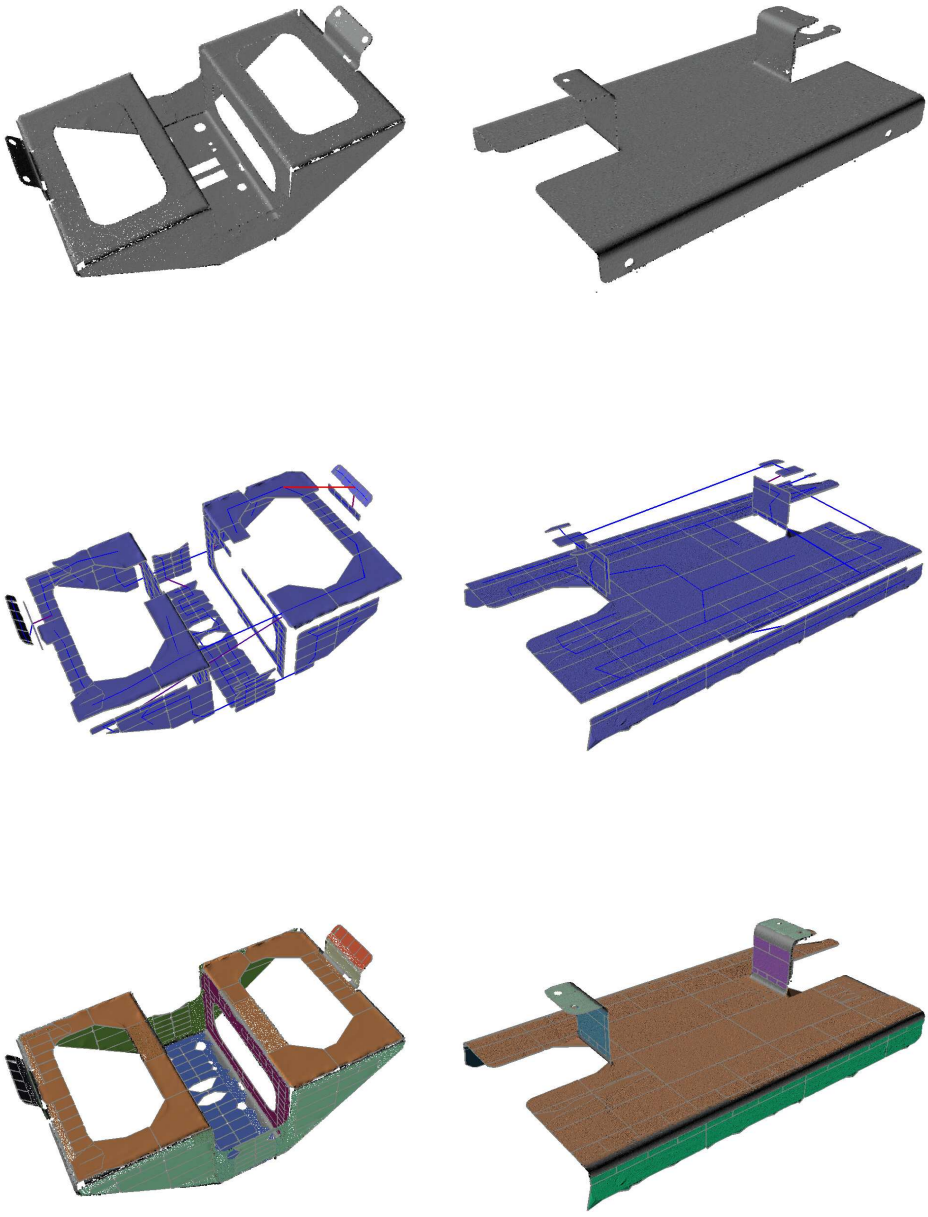


Figure 3.5: Planar segmentation results. For each point cloud, the intermediate patches and the clustering graph are depicted in the middle.

CHAPTER 4

Industrial Inspection

Contents

4.1 Introduction

The result of the segmentation algorithm introduced in the previous chapter serves as an initial analysis to gain more insight in the structure of the point cloud itself. More specifically, the planar subset of the segmentation with

Figure 4.1: A robot assisted sheet metal bending system. Image courtesy of FANUC robotics.

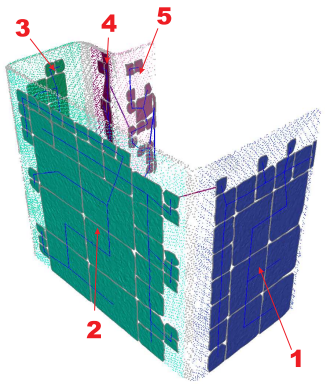
which we are concerned proves to be ultimately useful for point data gathered from manufactured objects and environments.

In this chapter, we discuss the analysis of the planar segmentation for industrial applications. In this context, we consider industrially bent sheet metal parts, consisting mainly of planar parts by default. We will show how the planar segmentation aids in the tolerance verification of these manufactured parts. We will also discuss a novel two dimensional triangulation algorithm, which is used to assess the precision of cut perforations.

The chapter is concluded with a section on surface reconstruction from piecewise planar point clouds. The reconstruction algorithm fits a polygonal model to point data sampled from a piecewise planar object using the planar segmentation and the triangulation procedure, after which the planar regions are merged together with a stitching procedure.

4.2 Tolerance Verification for Bent Sheet Metals

Bending is a commonly used shaping method in the manufacturing of various industrial products. It involves the uniform straining of a flat sheet or strip of metal round a straight axis [Duflou 05]. Sheet metal bendings or foldings have important applications in the aerospace, civil engineering and transportation industry. Typical applications include for example light-weight cores folded from aluminium, steel laminated roofing and sturdy trailer beds [Elsayed 04]. An example of a bending system is shown in figure 4.1.



$\angle_{12}$	90.0864
$\angle_{23}$	89.5667
$\angle_{34}$	90.5221
$\angle_{45}$	90.0000

Figure 4.2: Angular precision of a bent sheet metal. The tolerance was set to 0.2 degrees.

4.2.1 Bending Accuracy

Important criteria in assessing the quality of bent parts involve the precision of the fold angles, and the deviation of the planar elements from the ideal plane, information which is readily available from the hierarchical PCA analysis of laser scan data of the part: angle precision can be determined from the planar cluster normals, and the deviation criterium can be directly deduced from the σ_3 term of the cluster, which can be further corrected with the sensor noise statistics, which are known a priori. Figure 4.2 shows an example of a laser scan of a typical bent profile, and the deduced angles.

4.2.2 Cutting Precision

Next to the actual mechanical deformation of sheet metals, cutting perforations is another main process of sheet metal fabrication. The material can be cut using a variety of techniques such as laser and water jet cutting. The precision of a cut is influenced by a number of factors including material properties, speed of cutting and the heating properties of the cutting process itself.

The analysis of the cutting precision from a point cloud sampled from a perforated sheet metal involves the polygonal reconstruction of the sides of the industrial part, and the extraction of the boundaries of this polygonization. To this end, we have developed a novel two dimensional triangulation method for the boundary extraction, based on the *Delaunay* triangulation procedure

Delaunay triangulation is one of the fundamental topics in computational

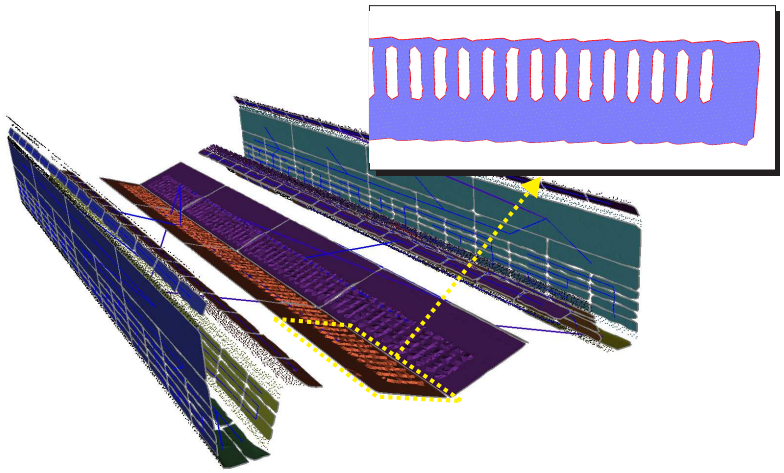


Figure 4.3: Cut analysis principle. A planar cluster is adaptively triangulated in the plane to extract the boundaries of perforations (red lines in the inset figure).

geometry and is applied in a large variety of problems because of its desirable properties [Edelsbrunner 01]. One of these attributes is for instance that the circumcenter of a Delaunay triangle does not contain any other vertex of the triangulated set. However, the Delaunay procedure is not directly applicable in the context of boundary detection in the extracted planar clusters, since it closes internal holes and non-convex parts of the clusters.

A solution to this could be first applying a boundary detection procedure such as the one proposed by Pauly et al. [Pauly 03a] or Volodine et al. [Volodine 03] after which a constrained Delaunay triangulation [Edelsbrunner 01] would produce the correct polygonization. In the next section we formulate an alternative triangulation procedure, based on a Kernel Density Estimation.

Density Based Triangulation

To triangulate a planar cluster with holes and concave regions, we want to retain those triangles from an initial Delaunay triangulation which are actually part of the physical surface. Figure 4.3 illustrates this concept. Our approach involves filtering triangles based on a *Kernel Density Estimate*(KDE) of the planar cluster under consideration.

Kernel Density Estimation is the construction of an estimate of the statistical density function from some observed data, by means of summing kernel

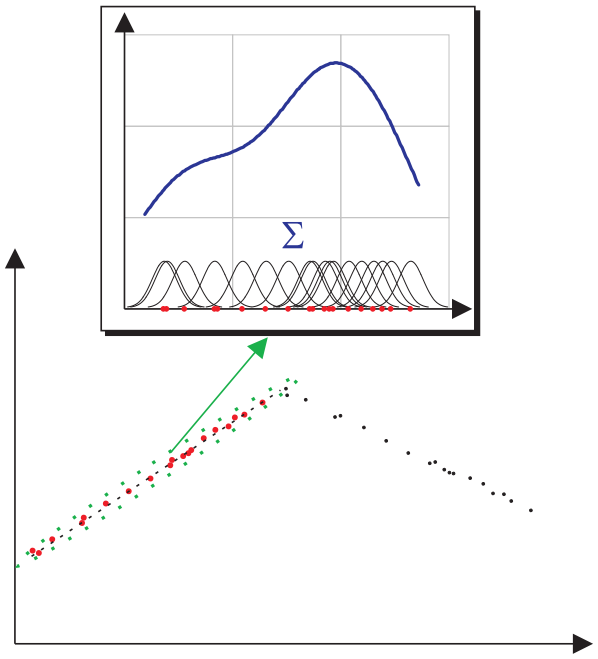


Figure 4.4: Kernel density estimate for a segmented planar cluster. The cluster (red points) is projected, after which the KDE is determined.

functions centered over the data points. Intuitively, the KDE function returns for a given point the likelihood it is a member of the population for which the KDE was determined.

Let $\mathcal{C}$ be a segmented planar cluster consisting of $n_{\mathcal{C}}$ points and let $\mathcal{C}_p$ be the projection of $\mathcal{C}$ onto the plane spanned by its first two principal components. Formally, the KDE of the two dimensional population $\mathcal{C}_p$ is given by:

$$\hat{f}(x) = \frac{1}{hw_p} \sum_{p \in \mathcal{C}} K\Big(\frac{x-p}{h}\Big) \tag{4.1}$$

with h denoting the window size for the kernel function K and w_p the weight the kernel reveives. For now, we assume these weights to be the constant $1/n_{\mathcal{C}}$. Popular kernel functions include the *Gaussian* and *Epanechnikov* kernels. See figure 4.4 for a graphical depiction of the determination of a planar cluster KDE.

Our Delaunay density based filtering algorithm evaluates for each Delaunay triangle of the triangulation of the projected planar cluster $\mathcal{C}_p$ the average

```
function T = triangulate(X,Y,tau)
//Delaunay triangulation
Tri = delaunay(X,Y); nT = size(Tri);
//Density estimation
f = kde(X,Y);
//filter delaunay triangles
for i=1:nT {
    t = Tri(i);
    //average density per unit area
    d = D(t,f);
    if (d > tau)
        T.add(t);
} return T;
```

Algorithm 4.1: Pseudo code listing for the density based Delaunay filtering.

density per unit area. If this density is sufficiently high, the triangle is retained. This can be expressed formally with the notion of an *internal triangle*:

Definition Let $\mathcal{C}_p = [X \mid Y]$ be a planar point set with a Delaunay triangulation T and a KDE $\hat{f}$ and let τ be a threshold value. We define a triangle $t \in T$ to be *internal* wrt τ if $D(t) = \frac{1}{\triangle(t)} \int_t \hat{f} > \tau$, with $\triangle(t)$ denoting the area of t.

In practice, the density integral $D(t)$ is approximated by evaluating the KDE at the vertices of t. For $t = (p_1, p_2, p_3)$ this gives:

$$D(t) \approx \frac{\hat{f}(p_1) + \hat{f}(p_2) + \hat{f}(p_3)}{3\triangle(t)}$$

The complete planar triangulation procedure is outlined in algorithm 4.1. All that remains for the evaluation of the density estimation given by equation 4.1 is the specification of the window size h. A window width which is too large will result in an oversmoothed density estimate, while an h-value which is too small causes the estimate to be just a collection of sharp pulses. Frequently used automatic bandwidth selectors are the plug-in method and the method of cross-validation, of which a thorough treatment is given for instance in [Park 92].

A reasonable choice of bandwidth selection is Silverman's *Rule of Thumb*, which is a member of the plug-in method family. For Gaussian kernels, the rule of thumb (*rot*) bandwidth is given by:

$$\hat{h}_{rot} = \left(\frac{4\hat{\sigma}^5}{3n_{\mathcal{C}}}\right)^{1/5} \approx 1.06\hat{\sigma}n_{\mathcal{C}}^{-1/5} \tag{4.2}$$

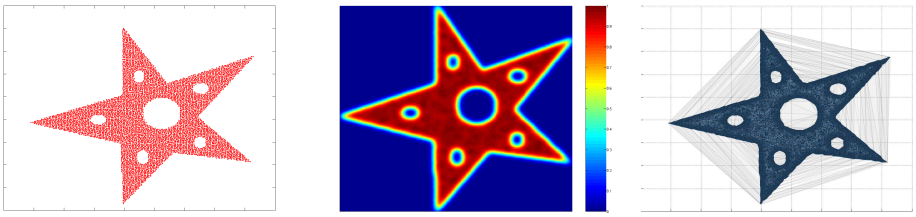


Figure 4.5: KDE based triangulation for planar clusters. From left to right: planar cluster, density estimation and the resulting triangulation. Retained triangles of the Delaunay algorithm are rendered shaded.

with $\hat{\sigma}$ the standard deviation estimator:

$$\sqrt{\frac{1}{n_{\mathcal{C}}-1}\sum_{i=1}^{n_{\mathcal{C}}}(x_i-\bar{x})^2}$$

The rot bandwidth selector assumes the kernel is Gaussian and that the underlying conditional distribution is Normal. Figure 4.5b shows the density estimate for a perforated star planar cluster. We can clearly see how this KDE has the desired property that internal points with respect to the holes and concavities have a high density associated with them, while the density slopes downwards at the *perceptual* boundaries of the cluster. The result of the density triangle retention procedure for the star cluster is shown in figure 4.5c.

Non-uniform Sampling

The triangulation procedure described in the previous section works well for surfaces that are sampled more or less uniformly. However, in practice most point clouds do not exhibit this optimally sampled characteristic. For instance, point sets which are merged from multiple partial scans, typically have regions of higher sample density in the overlapping regions (see figure 4.6). Also, a slanted surface with respect to the sensing device causes the samples to gradually spread out across the surface.

If there are large variations in sampling density and thus in the density estimation, the triangle retention scheme outlined above is prone to inaccurate triangulations, as is illustrated in figure 4.6. However, we are not interested in the precise density of a cluster as such to decide on the inclusion of the polygons. Rather, we want the density estimate to be more or less uniform in

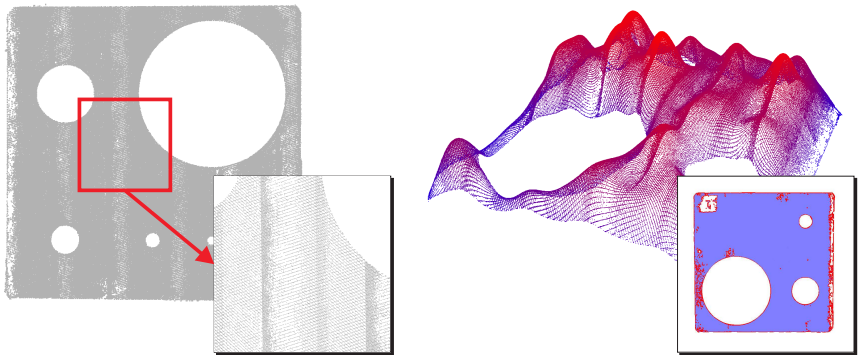


Figure 4.6: Non-uniform sampling density due to the merging of multiple overlapping scans. The right image shows the KDE evaluated at each point sample and the resulting triangulation.

the interior regions of the point cloud, approaching zero near the perceptual boundaries.

One approach could be using a variable bandwidth per kernel, for instance based on a nearest neighbor criterion, so that samples in densely populated regions would generate a more narrow kernel. However, in practice we found that this approach did not generate the desired density smoothness.

Instead, we choose to adjust the weights w_p. To this end, we employ a dual kernel density estimate: We first perform a density estimate using the h_{rot} bandwidth and uniform weights $1/n_{\mathcal{C}}$. A second density estimation, using $w_p = \hat{f}^{-1}(p)$ generates a density surface with the desired smoothness characteristics, as samples in dense areas receive less weight as opposed to those in relatively undersampled regions of the planar cluster.

Figure 4.7 shows the results of the dual density approach on the circular cut dataset. Note how the density is more evened out, thus resulting in a cleaner and more complete triangulation. However, for this particular dataset, some superfluous triangles are removed near the right boundary. As the inset figure indicates, this is due to the irregular sampling as the holes are removed between two strips of higher density. Another dataset and some of the reconstructed faces are depicted in figure 4.9.

To gain empirical insight in the dual KDE approach, figure 4.8 depicts a plot of the number of data points versus the KDE processing time using the MATLAB Kernel Density Estimation Toolbox [A. Ihler 03]. This shows clearly an exponential time complexity for the dual density estimation.

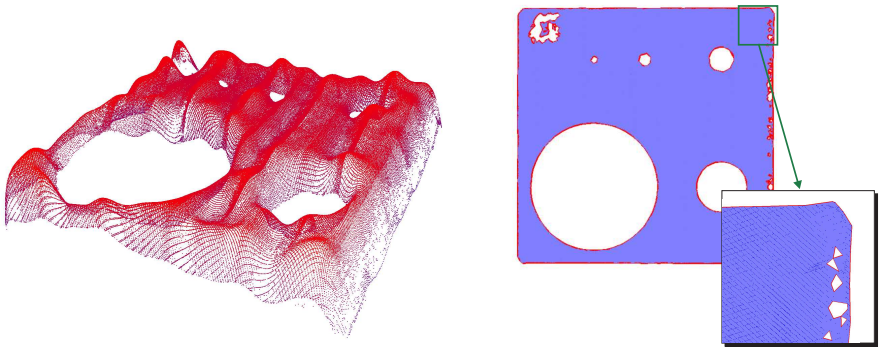


Figure 4.7: Dual KDE approach, with non-uniform weights attuned to the actual density. The right image shows the resulting triangulation (Compare with figure 4.6). Inset figure: see text.

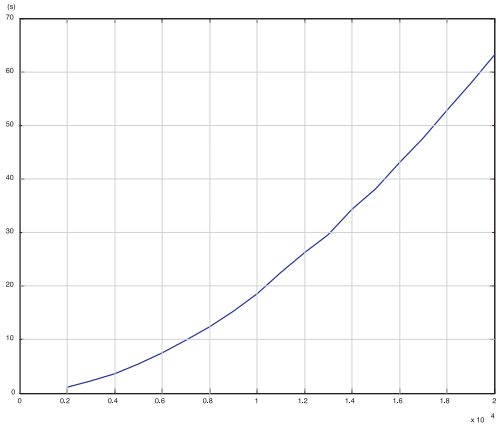


Figure 4.8: Timings for the dual KDE approach for an increasing number of data points (x axis).

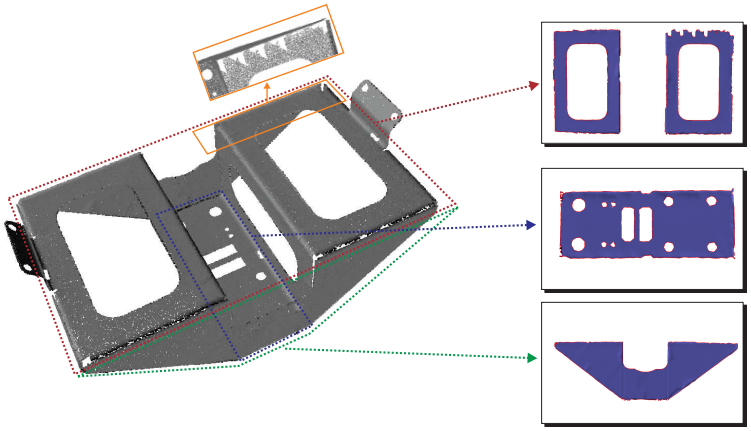


Figure 4.9: Some of the reconstructed faces of the *Wegra* dataset.

4.3 Reverse Engineering

Reverse engineering (RE) of geometric shapes is the conversion of a point cloud assumed to be sampled from the boundary of some physical shape to a concise geometric surface description such as a polygonal model. This is in fact the inverse of the traditional CAD/CAM paradigm, in which a tangible object is manufactured from computer designed models.

In this section, we will discuss the surface reconstruction of point clouds with the prior that the point set was scanned from an object which is entirely made of planar surfaces, a property which is also denoted by the term *piecewise planar*.

Over the years, reverse engineering physical shapes has become an invaluable asset in the production pipeline such as prototyping, mold design and automated inspection. The particular problem of surface reconstruction has therefore received a lot of attention in the scientific literature.

Among the first authors addressing the problem of reconstructing a surface from a scattered point set is Boissonnat [Boissonnat 84]. In his work, a nearest neighbor criterion is used, propagating a contour throughout a set of points, eventually converging to a correct solution, albeit under restrictive conditions.

Hoppe et al. [Hoppe 92] construct a signed distance function to the point cloud, with the help of graph techniques to ensure a consistent orientation. The zero isosurface of the distance function is triangulated using marching cubes [Lorensen 87], a well know algorithm often used in the context of surface reconstruction. An algorithm similar in spirit to the signed distance function

of Hoppe was developed by Curless and Levoy [Curless 96]. They propose a volumetric method for the integration of several range scans, scan-converting them onto a regular grid. The problem of combining several range scans is also addressed by the zippered meshes of Turk and Levoy [Turk 94], in which an iterated closest point algorithm is used for registering a pair of meshes created from range scans.

Edelsbrunner and Mücke [Edelsbrunner 92] generalize the notion of convex hull to that of α-hull, which is defined to be the complement of the union of α-spheres not containing any points of the input point set. A shape is reconstructed by substituting the curved boundary elements with simplicial complexes dubbed α-shapes.

Also, algorithms have been proposed which not only rely on empirical results, but also provide theoretical foundations on the geometric properties of the output surface. Bernardini et al. [Bernardini 99] propose a ball pivoting algorithm, related to the α-shapes method, which reconstructs a surface by essentially revolving a sphere on the point cloud. This algorithm is guaranteed to reconstruct a surface homomorphic to and within a bounded distance from the original manifold, given a sufficiently dense sampling.

The Crust algorithm by Amenta et al. [Amenta 01] [Amenta 00b] constructs a surface from the discrete approximation to the medial axis. The Cocone algorithm [Amenta 00a] and its incarnations Tight Cocone [Dey 03] and Super Cocone [Dey 01]draw upon the concepts of the crust, and guarantees an output homomorphic to the original surface.

The point set surface (PSS) proposed by Alexa et al. [Alexa 01] is widely used to reconstruct a continuous surface from scattered point data. The PSS method involves a Moving Least Squares projection [Levin 98] for the surface definition. An alternative projection operator is defined by [Reuter 05] et al. to handle sharp features.

Purely planar surface reconstruction is addressed by Schindler [Schindler 03] using uniform subdivision, and then applying RANSAC fitting to generate local planes, which are clipped to the individual cells, and then joined together using a local neighborhood *boundary melting*. However, precise boundaries are not extracted, nor is a closed polygonal model generated. In urban modeling, planar surface reconstruction plays an essential role to generate approximate planar models for buildings [Rottensteiner 03] [Peternell 04].

4.3.1 Surface Stitching

All that remains to generate a *closed* model from a piecewise planar point cloud is the joining of nearby boundaries from planar clusters: the density

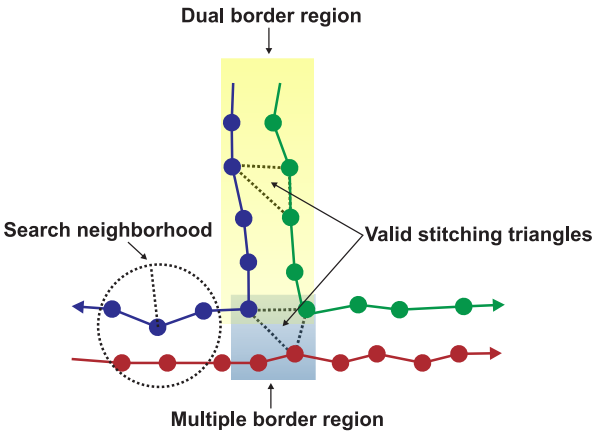


Figure 4.10: Schematic of the concepts used for surface stitching.

based triangulation procedure enables us to reconstruct each planar face of a point cluster in turn and independent from one another. In this section, we discuss how to triangulate the borders of a piecewise planar point cloud, given the triangulated planar clusters. To this end, we propose a simple surface stitching procedure based on a planar Delaunay procedure.

We define a valid *stitching triangle* as being one of the following (see figure 4.10):

- A triangle consisting of two *crossing* edges, that is edges that connect two cluster boundaries and one edge joining two consecutive vertices of either boundary.

- A triangle connecting three clusters.

These valid triangles are identified using a boundary sweeping procedure: for every boundary, a fixed radius search is performed around each vertex for other boundaries' vertices, thereby bookkeeping two sets for the vertices of every combination of boundaries that are close. One set holds the vertices of *dual border regions*, and another keeps track of *multiple border region* vertices. These concepts are illustrated in figure 4.10. Next, each boundary region is triangulated in their respective best fit plane, after which only valid stitching triangles are retained.

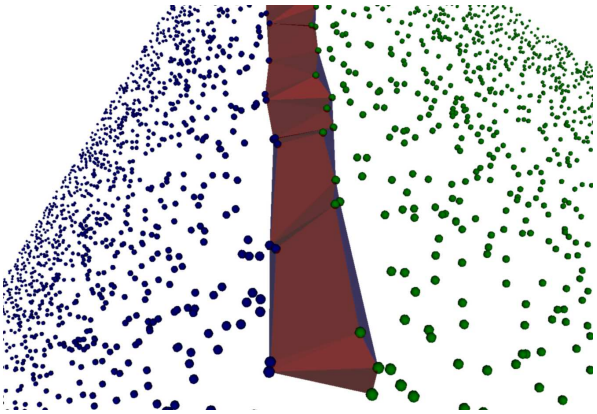


Figure 4.11: Stitching principle. Only valid stitching triangles (shaded red) are retained in the stitch.

4.3.2 Results

Figure 4.12 shows the result of the piecewise planar surface reconstruction for two perceptually piecewise planar point clouds which were generated as simulated range scans. The timings for the two main stages of the reconstruction - planar segmentation and (dual) density estimation - are given in table 4.1. Notice how even though the two point clouds have an equal number of data points, the density estimation for the *House* set took nearly twice as much time to complete as the *Blocks* point cloud. This stems from the non linear behavior of the dual KDE approach and the fact that the ground plane contains most of the points in the *House* point cloud.

Model	**n**	**Segmentation (ms)**	**Density Estimation (s)**
Blocks	52689	746	60.5
House	49658	676	105

Table 4.1: Performance of the piecewise planar surface reconstruction algorithm for the models depicted in figure 4.12.

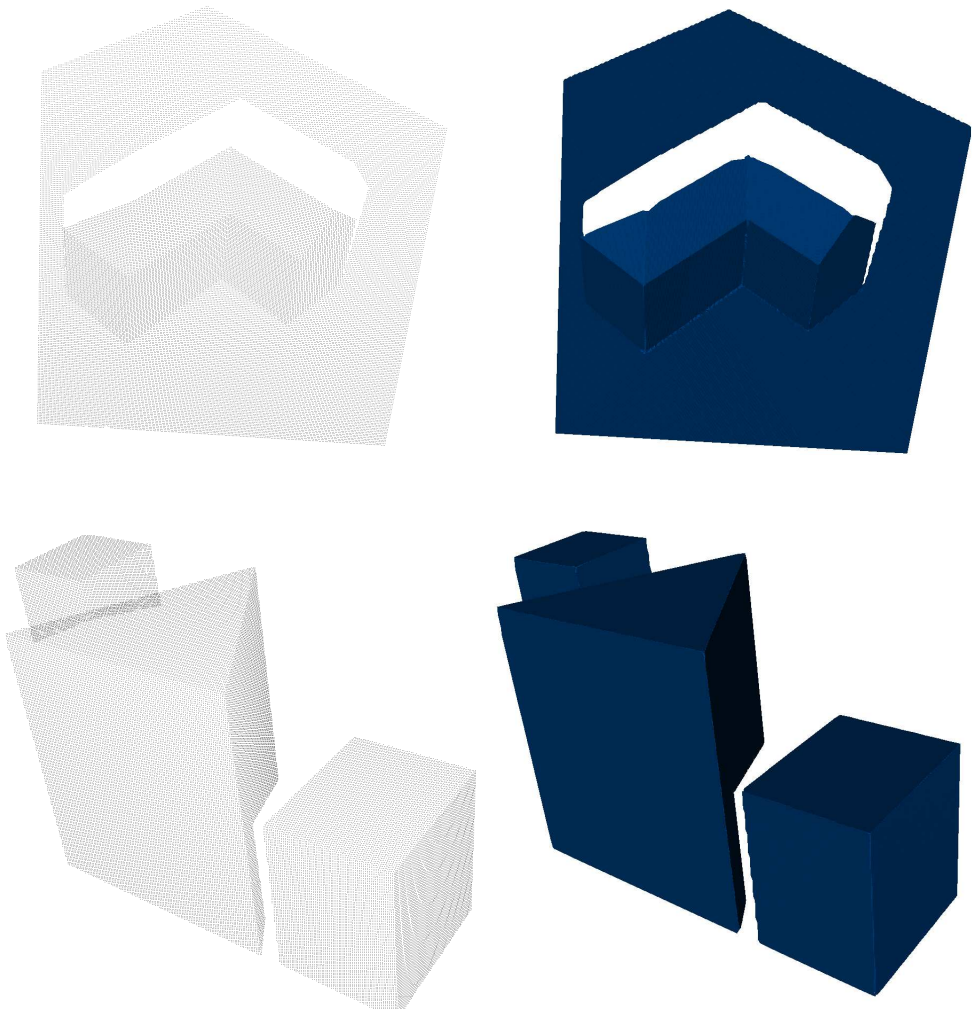


Figure 4.12: Piecewise planar surface reconstruction of the *House* and *Blocks* datasets.

4.4 Conclusion

In this chapter, we have discussed some important concepts in industrial inspection, applied to bent sheet metals. More specifically, we have shown how the planar segmentation algorithm can be used to gain more insight in the structure of these industrially manufactured parts.

Bending accuracy can be directly assessed using the planar analysis outlined in the previous chapter. Working from this segmentation, we have also discussed the more intricate problems of cut analysis and piecewise planar surface reconstruction. To this end, we have developed a novel density adaptive planar triangulation procedure, taking into account the possibly large inter-sample distance variations across a planar cluster. As an addition to the cluster triangulation we have also introduced a new stitching procedure, which enables the joining of the reconstructed clusters into a closed model.

CHAPTER 5

Hybrid Rendering

Contents

5.1 Introduction

During the last couple of years, point sets have emerged as a new standard for the representation of largely detailed models. This is partly due to the fact that range scanning devices are becoming a fast and economical way to

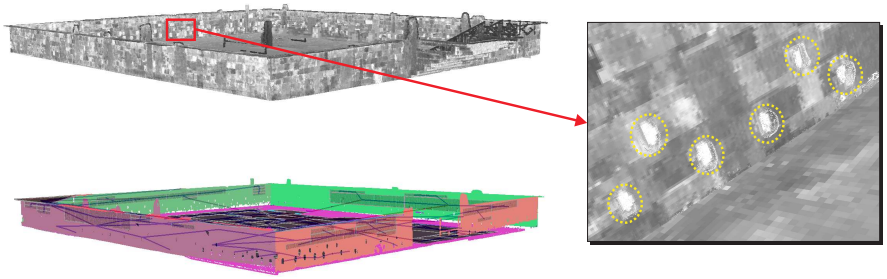


Figure 5.1: Hybrid rendering principle, demonstrated on the Templeté dataset. An input point cloud (not shown) is segmented (bottom figure) and rendered as a combination of planar geometry and points. The inset figure shows a detail of the left wall, depicting the planar structures (wall and ground) and detail geometry (extruding stones), encircled yellow.

capture dense point clouds. Also, the complexity of a traditional polygon model may be such, that a single polygonal primitive actually contributes less than a pixel during rendering.

However, for large, flat areas in point sets, there is a considerable redundancy in the representation as well as a rendering overhead when compared to polygon rasterization. Man-made objects and environments in particular, typically exhibit a significant degree of planar regularity.

We conclude the part on point cloud segmentation with a point cloud rendering technique, using the planar segmentation as a preprocessing step towards a hybrid rendering paradigm, which employs both triangles and polygons for efficient rendering of point models. Figure 5.1 illustrates this principle.

The central idea is to replace the planar parts of the point cloud with textured quads, while the parts of the point cloud capturing detailed geometry remain unaffected. This representation makes sense, not only because textured geometry is more efficient in terms of storage requirements, but also because of the fact that modern graphics hardware is specially targeted towards rapid visualization of polygons.

In a sense, this approach can also be seen as a lossless compression scheme for point models. However, the hybrid processing and rendering technique is only useful for point sets exhibiting a significant degree op planar regularity, as is often the case in man-made objects and environments.

5.2 Related Work

5.2.1 Point-based Rendering

Points were introduced as geometric rendering primitives by Levoy and Whitted [Levoy 85]. In their work, various rendering algorithms for different geometry representation are unified by the subdivision of the geometry into a dense set of sample points.

In recent years, the use of point primitives for rendering highly detailed models has been revisited and have been the subject of extensive research [Pfister 04] [Kobbelt 06]. The requirement of hole-free renderings has been met by image-space reconstruction techniques [Grossman 98] [Pfister 00] or by object-space resampling [Adamson 03].

The idea of rendering points was extended to object-space disks or ellipses with surface splatting [Zwicker 01]. The overlap of the splats guarantees the hole free rendering, and the shading discontinuities resulting from the splat intersections are filtered using an anisotropic anti-aliasing method [Zwicker 01]. A splat clipping algorithm has been proposed by Pauly et al. [Pauly 03b] to handle sharp features.

The increasing efficiency and programmability of graphic cards has inspired a number of splatting techniques exploiting the benefits of graphical hardware acceleration [Botsch 05]. With sequential point trees [Dachsbacher 03], Dachsbacher et al. present a hierarchical GPU point cloud visualization algorithm, based on the well-known Q-Splat system [Rusinkiewicz 00].

5.2.2 Imposters and Billboards

A popular approach to reduce the geometric complexity of computer models, without affecting the visual quality of the rendered output is the clustering of large parts of the scene as textured planes, as was originally proposed by Maciel et al. [Maciel 95]. This technique is for example commonly used in video games [Gamasutra 06].

Over the years various extensions to the basic principle, such as viewpoint-adaptive texturing [Shade 96], image-based interpolation techniques such as the layered depth image (LDI) approach of Shade et al. [Shade 98] and layered imposters [Schaufler 98] were proposed to more satisfactory handle parallax-effects.

More recently, so called *billboard clouds* have been proposed by Décoret and coworkers [Décoret 03]. In their work, a given 3D model is converted into a sparse set of textured billboards, thus resulting in a dramatic view-independent

simplification, at the cost however of quite considerable requirements for the texture memory.

All of the works cited above expect polygonal inputs for the simplification. The renaissance of the point based paradigm has also inspired some work on the simplification of point cloud models with geometric primitives. Wicke et al. [Wicke 05] approximate a given point cloud with a decimated version of a reconstructed mesh. Visual detail is maintained employing normal maps.

Subdivision surfaces too have proven useful candidates for the efficient representation and visualisation of point sets. For instance, Cheng et al. [Cheng 04] use an initial control mesh which is then refined to match the underlying point cloud. In contrast, Boubekeur et al. [Boubekeur 06] use a set of local triangulations which are glued together using a subdivision scheme.

5.2.3 Hybrid Rendering

The combination of both point and polygon primitives into a single rendering framework was first addressed by the POP rendering system [Chen 01]. The POP system builds a hierarchy of points, and uses triangles at the lowest level. Hybrid Simplification [Cohen 01] also incorporates triangles and points for model display, and uses a triangle hierarchy, in which points are generated based on triangle size and an error metric. While the POP and the Hybrid Simplification build the hierarchy from an input triangular mesh, the Point to Mesh Rendering (PMR) system [Dey 02] constructs a hierarchy of points and triangles from an input point cloud, based on intrinsic model features.

The previous algorithms focus on on-the-fly transitions between point and polygons, while maintaining the underlying point set representation. In contrast, we propose a point cloud processing technique, that extracts planar regions and replaces them with texture mapped polygons *a priori*. In that sense, our approach is similar to the hybrid rendering system of Wahl et al. [Wahl 05], which identifies planes at different scales using RANSAC plane fitting. At every level of an octree hierarchy Wahl et al. construct a best fitting plane for each leaf node individually.

Albeit similar, the algorithm presented in this chapter differs fundamentally from the approach of Wahl et al. in the planar taxonomy, as planes are identified at different scales after which the extracted planar regions are clustered using a novel *minimal spanning tree* clustering algorithm, resulting in globally optimal planes, independent from scale.

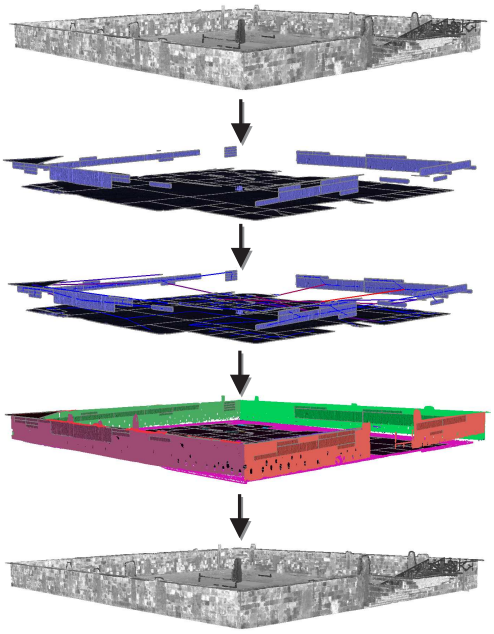


Figure 5.2: Planar analysis and hybrid rendering overview. The input point cloud is segmented, after which the planar clusters are represented as textured quads. The final hybrid model has the same appearance as the original point set, yet is more efficient in terms of representation and visualization requirements.

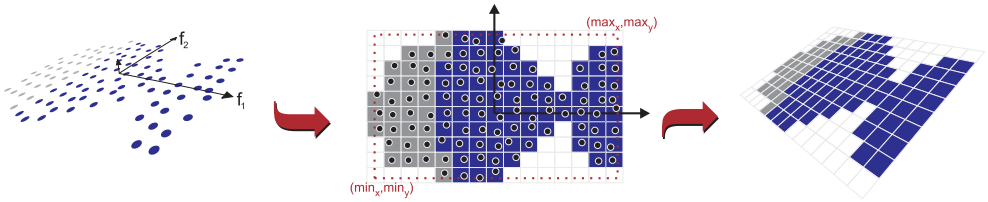


Figure 5.3: Texture generation. First the planar cluster is projected onto the plane spanned by the first two eigenvectors, after which the points are splat. Finally, the planar impostor is projected back into 3D space.

5.3 Algorithm

Our hybrid representation and rendering scheme is built upon the hierarchical planar analysis described in chapter 3. The input is a point cloud, possibly adorned with per point color information, consisting of both planar regions and more detailed geometry. Perceptually planar regions are segmented and replaced with textured quadrilaterals. In addition to the efficiency gained in terms of representation and visualization requirements this technique is also completely complementary to any existing point cloud processing and visualization technique which might be used to analyze or render the unaffected detail part of the point set, as these points are not contingent on the planar geometry. The whole process is summarized in figure 5.2.

5.3.1 Texture Generation

The first step in using textured impostors replacing planar segments of point clouds is determining the geometrical parameters for the quads. To this end, we use the PCA results for a given cluster: the points are projected onto the plane spanned by the first two eigenvectors f_1 and f_2. On this plane, the bounding rectangle enclosed by (min_x, min_y) and (max_x, max_y) is determined and projected back into 3D, thereby establishing the extends of the plane on which the texture is applied (see figure 5.3).

Next, we need to determine the resolution of the texture for a projected quad. Wahl et al. [Wahl 05] choose the resolution so that the inter-texel distance equals the approximation error of the plane. Wicke and coworkers [Wicke 05] propose an alternative iterative procedure, in which an error function between the surfel representation of a planar patch and the projected texture is evaluated.
We determine the texture resolution, based on the following observations:

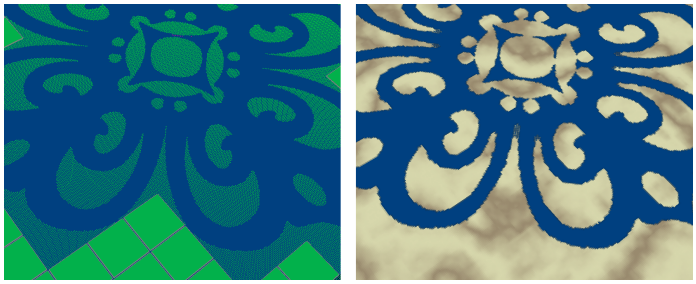


Figure 5.4: Texture projection. The planar cluster (left) is splat onto an RGBA texture(right).

- The ration between the number of columns and rows in the texture should equal the ratio of the spatial extend of the planar impostor. More formally, let the texture resolution be denoted as $r_x \times r_y$ and let $e_x = max_x - max_x$ and $e_y = max_y - min_y$, then
$$\frac{r_x}{r_y} = \frac{e_x}{e_y}$$

- The number of occupied texels after texture projection should reflect the number samples $n_{\mathcal{C}}$ of the planar cluster under consideration. If B is a binary $r_x \times r_y$ texture on which the cluster is projected, we can express this as
$$\begin{aligned} n_{\mathcal{C}} &\approx \sum_{i,j} B(i,j) \\ &= \frac{r_x r_y}{r_x r_y} \sum_{i,j} B(i,j) \\ &= r_x r_y \gamma \end{aligned}$$

  in which γ denotes the texture occupancy : the percentage of texels which receive a contribution from the point samples. In practice, this quantity is approximated by first projecting the samples on a small binary texture.

From these formulas, the texture resolution can be easily derived as:
$$\begin{aligned} r_y &= \sqrt{n_{\mathcal{C}} \gamma \frac{e_y}{e_x}} \\ r_x &= r_y \frac{e_y}{e_x} \end{aligned}$$

after which the texture is constructed by splatting the projected planar cluster kernel. In order to maintain the holes in the original cluster, this texture is initially set completely transparant. Figure 5.4 shows an example of a textured planar impostor.

The resolution analysis presented above implicitly assumes a uniform density, but in practice the heuristic we developed allows for some variance across the sampling density, without affecting the visual quality of the textures.

5.3.2 Results

We validate the results of our hybrid approach on a number of models, depicted in figure 5.5. The performance statistics are presented in table 5.1. For each point cloud, the number of raw data points n_r, the unprocessed rendering performance fps_r in Hz., the number of quad impostors $quads_h$ for the hybrid rendering, the number of residue detail points n_h and the visualization speed fps_h are given. The timings are presented as they were recorded on a 1.6 Ghz. Pentium M machine with 1 Gigabyte of internal memory and an on-board Intel graphics accelerator with 64 MB of memory.

Rather than being interested in the absolute performance of the different visualizations, the important observation is that there is an inversely proportional performance gain with the number of point primitives which are retained in the hybrid rendering. That is, the planar impostors have a negligible impact on rendering performance.

Model	$\mathbf{n}_r$	$\mathbf{fps}_r$	$\mathbf{quads}_h$	$\mathbf{n}_h$	$\mathbf{fps}_h$
tile	570K	7	6	161K	23
elk	332K	9	4	227K	13
templeté	1.4M	6	13	200K	49

Table 5.1: Statistics for the hybrid rendering algorithm.

5.4 Conclusions

We have presented an efficient algorithm to extract planar parts of a given point cloud. A hybrid model is composed of the segmentation residue points and textured quads, generated from the planar clusters. This representation not only allows for a more efficient rendering of the point data, but is also more compact in terms of memory requirements.

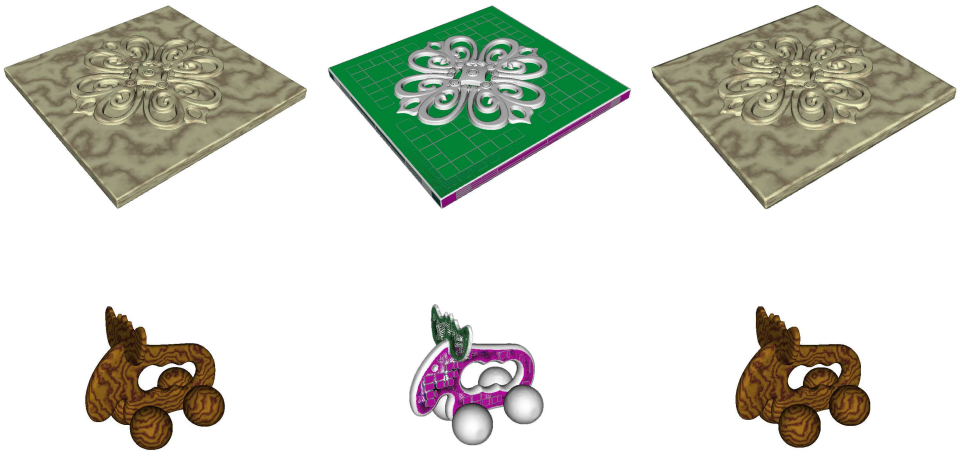


Figure 5.5: The results of the hybrid analysis and rendering algorithm. From left to right: (a) unprocessed point cloud , (b) planar segmentation results with residue points in white and (c) hybrid model.

The core of the algorithm consists of the planar segmentation presented in chapter 3. Each of these planar clusters are rendered as a textured quad, while the high frequency detail point geometry remains as a separate point cloud. To this end, we have developed a simple heuristic to determine the texture resolution, after which each planar cluster is splat onto an initially fully transparent planar impostor.

As we have shown, this approach is particularly useful for point data with sufficiently large planar subsets, as is often the case with man-made objects and environments.

Point Cloud Processing: Concluding Remarks

Summary

In the second part of this dissertation, we have presented a novel planar segmentation algorithm for point clouds, which combines elements from top-down analysis and bottom-up gathering. The crux of the algorithm is a hierarchical planarity assessment using Principal Component Analysis. Planar patches are merged into planar clusters using a Minimum Spanning Tree pruning technique which iteratively cuts the edges of the graph until all resulting subgraphs link coplanar patches. The coplanarity is assessed using an efficient combination of the PCA results gathered from the octree analysis. The segmentation is finalized using per point inclusion based on the Mahalanobis metric. An initial filtering of the point data ensures the point cloud is essentially noise-free, so that the segmentation is a precise as possible. We have demonstrated the results of our approach on a number of different point sets to demonstrate the efficacy of the algorithm in a variety of situations.

Next, we have discussed some important concepts in industrial inspection, applied to bent sheet metals. More specifically, we have shown how the planar segmentation algorithm can be used to gain more insight in the structure of these industrially manufactured parts.

Bending accuracy can be directly assessed using the planar analysis outlined in chapter 3. Working from this segmentation, we have also discussed the more intricate problems of cut analysis and piecewise planar surface reconstruction. To this end, we have developed a novel density adaptive planar

triangulation procedure, taking into account the possibly large inter-sample distance variations across a planar cluster. As an addition to the cluster triangulation we have also introduced a new stitching procedure, which enables the joining of the reconstructed clusters into a closed model.

Finally, we have presented an efficient algorithm to extract planar parts of a given point cloud. A hybrid model is composed of the segmentation residue points and textured quads, generated from the planar clusters. This representation not only allows for a more efficient rendering of the point data, but is also more compact in terms of memory requirements.

The core of the algorithm consists of the planar segmentation presented in chapter 3. Each of these planar clusters are rendered as a textured quad, while the high frequency detail point geometry remains as a separate point cloud. To this end, we have developed a simple heuristic to determine the texture resolution, after which each planar cluster is splat onto an initially fully transparent planar impostor.

As we have shown, this approach is particularly useful for point data with sufficiently large planar subsets, as is often the case with man-made objects and environments.

Future Directions

In this section we give some suggestions for future work which can improve the results and the generality of our approach.

Planar Analysis

The planar segmentation algorithm provided in chapter 3 currently operates independent from scale, that is, at every stage of the recursion the same threshold τ is used. A useful approach would be to vary this threshold in function of the octree level, so that small scale deviations from the ideal plane can be detected and appropriately dealt with.

Also, the fine segmentation which assigns points iteratively to clusters currently uses a single region growing pass. Some additional passes might further improve the segmentation results. This is also closely related to the graph clustering algorithm itself: although the Mahalanobis weights penalizes large distances between individual patches, it usually connects coplanar regions with low weights, without taking perceptual adjacency into account. In some cases it might be useful to only cluster those patches which are not only coplanar but also spatially adjacent.

As an addition to the purely planar identification, it would also be a useful addition to subject the residue points to a higher order segmentation to extract for instance cylinders or spheres, as these surface types also play an important role in object manufacturing. This segmentation could for example use the same hierarchical strategy as the plane detection, working with the per-point curvature rather than the coordinates.

Industrial Inspection

At the moment, we use the dual KDE approach to generate a more uniform density, so that internal regions have similar density values associated with them. It might be worthwhile to investigate alternative density estimation routines, using a combination of non uniform weights and kernel windows to accomplish this uniformity in a single pass.

The surface reconstruction assumes a piecewise planar input point set. A stitching procedure joins the independently reconstructed planar clusters. In the context of bent sheet metals, which mainly consist of planar regions and the actual cylindrical bendings, a useful addition to the stitching would be a cylindrical surface reconstruction, to adequately handle the non-linear transitions between clusters.

Hybrid Rendering

For the hybrid rendering paradigm, we currently use one quad impostor per planar cluster. Relating to the graph clustering method, this might result in large quadrilaterals with many transparent regions. To avoid this, we could for instance use the planar reconstruction procedure and triangulate the simplified borders, and project on these impostors instead.

Also, currently the presence of large planes is not exploited for rendering the residue part of the point cloud. The rendering speed can be further improved using a form of occlusion culling with the extracted impostors.

CHAPTER 6

Conclusion

6.1 Summary

This dissertation dealt with the general topic of segmentation in Computer Graphics applied to historical film restoration and point cloud processing. The term *segmentation* in the general context refers to the partitioning of data into semantically distinct, non-overlapping groups or clusters, after which these meaningful components can be subjected to further manipulation. In most cases, data segmentation signifies a crucial first step towards a higher order understanding of the data as a whole.

The first part of this thesis discussed motion segmentation, in the context of historical film restoration. Previous work on movie restoration mainly focusses on the repair of individual frame defects. In contrast, we have addressed the most severe form of degradation, in which an entire sequence of frames is lost permanently, or degraded beyond local repair. To this end we have introduced a semi-automatic system, in which the user manually segments motion layers from input motion on both sides of the missing film. From this segmentation, lost frames are automatically regenerated by interpolating the motion layers and compositing the layers with an automatically generated and motion-compensated background layer. We have shown how this hybrid

approach results in a robust and general system for the reconstruction of missing film frames. The results of our approach have been demonstrated on a number of sequences from digitized archive film material.

The second part of this dissertation discussed the segmentation of three dimensional point clouds and some applications built upon this analysis. More specifically, we concerned ourselves with the *planar* segmentation of point clouds. The planar subset of the segmentation problem is of great interest, as many man-made objects and environments typically exhibit some form of planar regularity.

To this end, we have introduced a novel planar point cloud segmentation technique, which is capable of identifying planar subsets in unstructured point clouds. A novel hierarchical segmentation algorithm underpinned this analysis.

This planar segmentation serves as an initial analysis for three applications. First, the inspection of so called *folded metal sheets* was discusses. Next, a surface reconstruction for piecewise planar point clouds was presented. Finally, a new hybrid point and polygon rendering system was introduced, which exploits potential planar structures present in the point data to increase the performance of the visualization and representation.

6.2 Contributions

This dissertation has introduced a number of novel ideas and techniques for statistical data segmentation in computer graphics. In addition, for each of the major topics a number of suggestions were made for future research,

- Chapter 2 introduced a novel motion and feature adaptive mesh warping algorithm, which is tightly coupled with a user interactive layer segmentation system.
- In chapter 3, we presented a new point cloud segmentation algorithm, which identifies planar clusters from 3D point sets. Our approach involves a combination of techniques from statistical multivariate analysis and graph theory.
- A novel 2D triangulation procedure was proposed in chapter 4. Our approach filters Delaunay triangles for a topologically correct reconstruction, in presence of variations in sample density .
- Chapter 4 also established a new planar surface reconstruction algorithm. To this end, the density adaptive triangulation is supplemented with a new surface stitching algorithm.

- Hybrid rendering was introduced in chapter 5, replacing parts of point clouds with planar geometry to obtain an efficient representation and visualization paradigm.

6.3 Acknowledgements

I gratefully express my gratitude to the European Fund for Regional Development and the Flemish Government, which are kindly funding part of the research reported in dissertation. Part of this work was also funded by the European Commission under project IST-2001-37117 (Racine- S).

Appendices

APPENDIX A

Scientific Contributions and Publications

Publications at international conferences

[Fransens 03], 2003 *J. Fransens, P. Bekaert & F. Van Reeth.* A Hierarchical PCA-based Piecewise Planar Surface Reconstruction Algorithm. *In Proceeding of the Eighth SIAM Conference on Geometric Design and Computing, pages 203 – 214, Brentwood, TN, USA, November 2003. Nashboro Press*

[Fransens 05], 2005 *J. Fransens, F. Di Fiore & F. Van Reeth.* The Reconstruction of Missing Frames in Historical Films, a Layered Approach. *In Proceeding of the International Conference on Computer Graphics and Vision 2005 (GraphiCon05), pages 63 – 69, June 2005*

[Fransens 06c], 2006 *J. Fransens & F. Van Reeth.* Hierarchical PCA Decomposition of Point Clouds. *In Proceedings of the Third International Symposium on 3D Data Processing, Visualization and Transmission , to appear, june 2006*

Technical Reports

[Fransens 06b], 2006 *J. Fransens.* Planar Segmentation of Point Clouds for Hybrid Rendering. *Technical Report TR-UH-EDM-0602, University of Hasselt, 2006*

[Fransens 06a], 2006 *J. Fransens.* Density Based 2D Delaunay Sculpting. *Technical Report TR-UH-EDM-0602, University of Hasselt, 2006*

APPENDIX B

Nederlandstalige Samenvatting (Dutch Summary)

B.1 Inleiding

Algemeen gesproken houdt datasegmentatie de analyse en de groepering van conceptueel zinvolle onderdelen uit die data in, zodat de individuele segmenten bijdragen tot een hogere orde begrip van de data als een geheel. Segmentatie is vaak een cruciale eerste stap naar verdere dataverwerking toe. In deze thesis worden twee segmentatietechnieken besproken, beiden gestoeld op het principe van een hirarchische statistische analyse.

In het eerste deel wordt een techniek ontwikkeld die semi-automatisch bewegingsvectoren bekomen uit filmmateriaal segmenteert in individueel bewegende entiteiten. Deze segmentatie wordt dan gebruikt om verloren gegane film automatisch te reconstrueren, op basis van het filmmateriaal dat wel nog voorhanden is. Dit is vooral nuttig in de context van historische films, omdat het pelliculemedium waarop film wordt opgeslagen onderhevig is aan ernstige degradatie over de jaren heen.

Het tweede deel van de thesis handelt over de planaire segmentatie van zogeheten puntenwolken, een collectie van driedimensionele cordinaten die

tesamen het oppervlak van een fysiek object beschrijven. Puntenwolken kunnen bijvoorbeeld bekomen zijn uit 3D laserscanners of met photogrammetrische methodes. Met planaire segmentatie worden vlakstructuren in een puntenwolk gedetecteerd. Hiertoe werd een statistische methode ontwikkeld, gebaseerd op Principale Componenten Analyse. Deze vlakdetectie wordt dan gebruikt voor industrile inspectie van complexe plooiplaatproductie en voor een weergaveparadigma dat zowel polygonen als puntprimitieven gebruikt voor de weergave van complexe puntenwolken.

B.2 Domein

De titel van deze thesisdissertatie is *Statistische Datasegmentatie in Computer Graphics* aangezien we twee segmentatieproblemen binnen de computer graphics behandelen: de segmentatie van bewegingsvectoren enerzijds en de planaire segmentatie van puntenwolken anderzijds. Hoewel statistische principes in het algemeen altijd gebruikt worden om data te segmenteren, hebben de rigoureuze statistische concepten die de grondslagen vormen voor onze ontwikkelde technieken ons gemotiveerd dit onderzoek binnen de categorie *Statistische Segmentatie* te classificeren.

B.3 Contributies

Deze thesistekst introduceert een aantal nieuwe ideeën en technieken voor statistische datasegmentatie in computer graphics.

- Er wordt een nieuw mesh warping algoritme voorgesteld voor bewegingsgecompenseerde beeldinterpolatie, dat adaptief is aan sterke features binnen het beeld en de uniformiteit van het onderliggend bewegingsveld. Dit warping algoritme is geïntegreerd in een volledig systeem voor filmreconstructie.

- In het tweede deel wordt een nieuw planair segmentatie algoritme ontwikkeld, die clusters van vlakke data identificeert in ongestructureerde 3D data. Onze aanpak houdt een combinatie van multivariate statistische technieken en grafetheorie in.

- In de context van industriële inspectie wordt een nieuw 2D triangulatie algoritme voorgesteld, waarmee de precisie van uitsnijdingen binnen plaatproductie kan worden geanalyseerd. Onze techniek houdt rekening met de bemonsteringsvariabiliteit die gebruikelijk is bij industriële scans.

- De planaire segmentatieprocedure tesamen met het triangulatie-algoritme stelt ons in staat om het oppervlak te reconstrueren van stuksgewijs vlakke puntewolken, door toevoeging van een eenvoudige techniek die de individueel gereconstrueerde vlakken met elkaar verbindt om zo een gesloten oppervlak te bekomen.

- De thesis wordt afgesloten met een toepassing van vlaksegmentatie voor de visualisatie van puntenwolken. Hiertoe werd een nieuw visualisatieparadigma ontwikkeld die de rendering van punten en vlakken in een enkel raamwerk combineert.

B.4 Filmreconstructie

Sinds het begin van de cinematografie werd er een culturele en geschiedkundige schat aan informatie opgenomen en opgeslagen op film. Helaas is het pelliculemedium dat gebruikt wordt voor de opslag uitermate gevoelig aan externe degradaties. Zo kunnen er bijvoorbeeld diepe krassen op de film ontstaan door het mechanische projectieproces of kunnen kleurverschillen optreden door schimmelvorming.

Deze verouderingssymptomen moeten uiteraard zoveel mogelijk worden tegengegaan en hersteld waar mogelijk. Ook zou dit proces bij voorkeur automatisch moeten gebeuren omdat manuele restauratie zeer tijdrovend en bijgevolg duur is. Er bestaat reeds vergevord onderzoek naar de automatische restauratie van individuele frames van historische films. In tegenstelling tot deze technieken ontwikkelen we een volledig systeem om verloren gegane sequenties van beelden te hergenereren aan de hand van het filmmateriaal dat wel nog voor handen is.

B.4.1 Overzicht

Voor het genereren van volledige sequenties gebruiken we een gelaagd bewegingsmodel, waarin elke laag een onafhankelijk bewegende entiteit voorstelt. We maken hierin een onderscheid tussen de achtergrond en voorgrond in een beeld. Figuur 2.1 geeft een schematisch overzicht van onze aanpak.

De invoer bestaat uit de twee beelden die de te hergeneren sequentie omsluiten, en de bewegingsvectoren die deze beelden met elkaar relateren. Vervolgens worden eventuele voorgrondobjecten van de achtergrond gescheiden met behulp van manuele segmentatie. Deze segmentatie, die gebeurt op basis van planaire subdiviseoppervlakken bepaalt de bewegingslagen en de de onderliggende volgorde van deze lagen. Vervolgens worden de tussen-

liggende beelden gesynthetiseerd uit de gelaagde segmentatie en de automatisch gegenereerde achtergrond.

B.4.2 Achtergrondmesh

De achtergrond wordt vervormd aan de hand van een mesh controlestructuur die automatisch bepaald wordt, rekening houdend met de twee omsluitende beelden en de beweginsgvectoren die per pixel zijn berekend. Elke vertex van de mesh wordt dan verplaatst aan de hand van de onderliggende bewegingsvector. Dit is te vergelijken met de vervorming van een elastisch membraan. We willen dat deze mesh voldoet aan de volgende eigenschappen:

- De vertices van de mesh zijn bij voorkeur gelegen op sterke beeldfeatures. Een mogelijk maat hiervoor is de beeldgradiënt.
- De mesh zou aangepast moeten zijn aan het onderliggende bewegingsveld. Dit wil zeggen dat in gebieden waar de variatie in bewegingsrichting en -grootte hoog is, er meer vertices geplaatst moeten worden, in tegenstelling tot gebieden die een meer uniforme vervorming ondergaan.
- De precisie van de onderliggende verplaatsing moet hoog zijn, om een correcte warping te garanderen.

Deze eigenschappen worden voldaan met een nieuw meshgeneratie-algoritme. Het beeld wordt volgens een quadtreestructuur opgesplitst totdat alle quadranten een uniform bewegingsveld omvatten. Dan wordt per quadrant een vertex geplaatst op de pixel waarvan de verplaatsing door de onderliggende bewegingsvector het meest accuraat is. Met een Delaunay triangulatie wordt dan de uiteindelijke controlemesh bekomen.

B.4.3 Gelaagde Segmentatie

Voor de segmentatie van de voorgrondobjecten maken we gebruik van een subdivisietechniek waarbij de gebruiker het te segmenteren object omsluit met een controlepolygoon. Deze polygoon wordt dan onderverdeeld en verfijnd met lokaal interpolerende subdivisietechnieken. Deze meshstructuur wordt voor een enkele frame gecreëerd waarna dezelfde structuur over de andere frame wordt geplaatst zodat de gebruiker het voorgrond object over de twee frames heen kan relateren. Tesamen met de segmentatie geeft de gebruiker ook de onderlinge volgorde tussen de lagen aan zodat bij het hergenereren van de frames de correcte occlusies tussen de individuele lagen kunnen bekomen worden.

B.4.4 Beeldsamenstelling

De gereconstrueerde sequentie kan nu bekomen worden door het samenstellen van de tussenliggende achtergrondlaag en voorgrondlagen. Aan de hand van de achtergrondmesh en de bewegingsvectoren wordt de achtergrond voor elke te synthetiseren frame gegenereerd. Hierop worden de tussenliggende voorgrondlagen in de aangegeven volgorde geplaatst. De tussenliggende positie voor elk van deze lagen wordt bepaald aan de hand van de expliciete relatie tussen de meshvertices in de omsluitende beelden.

B.5 Analyse van Puntenwolken

Een puntenwolk (point cloud) is een set van driedimensionale punten die tesamen het oppervlak van een object beschrijven. Mogelijk wordt per punt ook kleur- of normaalinformatie meegegeven. Het puntenwolkparadigma constrasteert scherp met de traditionele poygonale representatie waarmee driedimensionale objecten doorgaans worden voorgesteld.

De laatste jaren hebben puntenwolken een flinke opmars gemaakt voor het modelleren, weergeven en analyseren van objecten. Dit is deels te wijten aan de opkomst van steeds goedkopere driedimensionele scanners maar ook aan de overbodigheid van veel topologisch rekenwerk, aangezien puntenwolken in essentie ongestructeerd zijn en dus enkel een impliciete topologie met zich meedragen. Een ander aspect is dat voor uiterst gedetailleerde modellen het principe van polygonale modellering niet meer zinvol is omdat individuele polygonen minder dan een pixel zouden beslaan tijdens de weergave.

Het tweede deel van deze doctoraatsdissertatie handelt over de planaire segmentatie van puntenwolken, en een aantal toepassingen hierop. Deze vlaksegmentatie is van groot belang omdat vele door de mens geconstrueerde objecten en omgevingen voor een groot deel uit vlakke structuren bestaan.

Deze planaire segmentatie vormt de basis voor een aantal toepassingen binnen de industriële analyse en datavisualisatie. In deze optiek wordt de analyse van plooiplaatproducten besproken en wordt een hybride weergaveparadigma ontwikkeld.

B.5.1 Planaire Segmentatie

Het segmentatie-algoritme dat we ontwikkeld hebben is gebaseerd op een hiërarchische Pincipiële Componenten Analyse (PCA), een welgekende techniek uit de multivariate data analyse, tesamen met technieken uit de grafeclustering.

Een gegeven puntenwolk wordt volgens een octree principe opgesplitst totdat elk octant planaire data omvat, of tot het maximale splitsingsniveau is bereikt. De planariteit van een octant wordt geanalyseerd aan de hand van de proportionele variantie opgevangen in de eerste twee principiële componenten ten aanzien van de totale datavariantie.

Vervolgens worden deze planaire delen gegroepeerd in clusters door toepassing van een grafeclustering op de minimal spanning tree (MST) van de vlakke delen, waarbij het gewicht bepaald wordt door een cummulatieve Mahalanobis afstand. Een speciaal ontwikkeld snoeialgoritme identificeert dan de coplanaire onderdelen van de grafe, en groepeert de data in vlakclusters. De segmentatie wordt gefinaliseerd door een groeialgoritme, dat iteratief de clusters uitbreidt met ongesegmenteerde punten op basis van de Mahalanobis metriek.

B.5.2 Industriële Inspectie

Het planaire segmentatie-algoritme vormt de basis voor een structurele analyse van plooiplaatproducten. Plooiplaatproductie is een gangbaar fabricatieproces waarbij een doorgaans metalen plaat op een ronde as gebogen wordt. Plooiplaten kennen veel toepassingen in de luchtvaart, bouwtechniek en de transportsector.

Eén van de criteria om de kwaliteit van een plooiplaatproduct in te schatten is de analyse van de plooihoeken. Deze plooihoekinformatie kan rechtsreeks worden bekomen uit de normalen van de clusters die bekomen zijn met de planaire segmentatie.

Naast het buigen is het uitsnijden van gaten met behulp van laser of water een van de belangrijke vormingsprocessen bij plooiplaatproductie. De precisie van deze uitsnijdingen hangt af van een aantal factoren zoals materiaaleigenschappen, snijsnelheid en de verwarming door het snijproces zelf. De snijprecisie houdt de polygonale reconstructie van een vlakke cluster in, waarna de randen van de uitsnijdingen gemakkelijk geëxtraheerd kunnen worden. Hiertoe hebben we een triangulatie-algoritme ontwikkeld, gebaseerd op de welbekende Delaunay triangulatie.

De Delauny procedure kan niet rechtstreeks worden toegepast omdat convexe stukken en gaten in de triangulatie worden opgenomen. Ons algoritme neemt deze triangulatie als input en weerhoudt enkel die driehoeken die deel uitmaken van het fysieke oppervlak. Deze filtering gebeurt op basis van een *Kernel Density Estimation* (KDE) van de planaire cluster die dan geëvalueerd wordt over elke Delaunay driehoek. Wanneer deze densiteit onder een bepaalde drempelwaarde valt, verwijderen we de driehoek.

Door toevoeging van een procedure die de afzonderlijk gereconstrueerde

clusters polygonaal met elkaar verbindt kunnen we ook oppervlaktereconstructie doen voor stuksgewijs planaire puntenwolken.

B.5.3 Hybride Rendering

Puntenwolken kennen de laatste jaren veel succes voor het weergeven van gedetailleerde driedimensionale modellen. Grote vlakke stukken in puntenwolken betekenen echter een significante redundantie voor de voorstelling en weergave van grote puntenwolken. Het is namelijk zo dat moderen grafische hardware speciaal ontwikkeld is om efficiënt polygonale modellen weer te geven.

Ons planaire segmentatiealgoritme kan gebruikt worden om vlakke structuren te identificeren en te vervangen door getexturede vlakken, zonder dat er een visueel verschil tussen het puur puntenmodel en het hybride model is. Deze aanpak houdt een textureprojectie op een initieel volledig transparante texture in. De hybride voorstelling blijkt zowel op vlak van geheugenvereisten als rendersnelheid goede resultaten te bekomen, vooropgesteld dat er voldoende vlakstructuren in de originele puntenwolk zijn.

B.6 Conclusie

In deze thesis hebben we statistische segmentatie en de toepassing daarvan besproken binnen computer graphics. In het eerste deel werd een techniek ontwikkeld die semi-automatisch bewegingsvectoren bekomen uit filmmateriaal segmenteert in individueel bewegende entiteiten. Deze segmentatie wordt dan gebruikt om verloren gegane film automatisch te reconstrueren, op basis van het filmmateriaal dat wel nog voorhanden is. Dit is vooral nuttig in de context van historische films, omdat het pelliculemedium waarop film wordt opgeslagen onderhevig is aan ernstige degradatie over de jaren heen.

Het tweede deel van de thesis handelt over de planaire segmentatie van zogeheten puntenwolken, een collectie van driedimensionele cordinaten die tesamen het oppervlak van een fysiek object beschrijven. Met planaire segmentatie worden vlakstructuren in een puntenwolk gedetecteerd. Hiertoe werd een statistische methode ontwikkeld, gebaseerd op Principale Componenten Analyse. Deze vlakdetectie wordt dan gebruikt voor industrile inspectie van complexe plooiplaatproductie en voor een weergaveparadigma dat zowel polygonen als puntprimitieven gebruikt voor de weergave van complexe puntenwolken.

Bibliography

[A. Ihler 03] A. Ihler. World Wide Web, `http://www.ics.uci.edu/~ihler/code/kde.shtml`, 2003.

[Adamson 03] A. Adamson & M. Alexa. *Ray Tracing Point Set Surfaces*. In SMI '03: Proceedings of the Shape Modeling International 2003, page 272. IEEE Computer Society, Washington DC, 2003.

[Alexa 01] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin & C. T. Silva. *Point Set Surfaces*. vis, vol. 00, pages 21 – 28, 2001.

[Altunbasak 97] Y. Altunbasak & A. Tekalp. *Occlusion-adaptive, content-based mesh design and forward tracking*, 1997.

[Amenta 00a] N. Amenta, S. Choi, T. Dey & N. Leekha. *A Simple Algorithm for Homeomorphic Surface Reconstruction*. In ACM Symposium on Computational Geometry, pages 213–222, 2000.

[Amenta 00b] N. Amenta & R. K. Kolluri. *Accurate and efficient unions of balls*. In Symposium on Computational Geometry, pages 119–128, 2000.

[Amenta 01] N. Amenta, S. Choi & R. Kolluri. *The power crust, unions of balls, and the medial axis transform*. International

Journal of Computational Geometry and its Applications, vol. 19, no. 2-3, pages 127–153, 2001.

[Bauer 03] J. Bauer, K. Karner, K. Schindler, A. Klaus & C. Zach. *Segmentation of Building Models From Dense 3D Point-Clouds*. In 27th Workshop of the Austrian Association for Pattern Recognition, Laxenburg, Austria, 2003.

[Bernardini 99] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva & G. Taubin. *The Ball-Pivoting Algorithm for Surface Reconstruction*. IEEE Transactions on Visualization and Computer Graphics, vol. 5, no. 4, pages 349–359, 1999.

[Besserer 04] B. Besserer & C. Thire. *Detection and Tracking Scheme for Line Scratch Removal in an Image Sequence*. In Proceedings of ECCV 2004, pages 264–275, 2004.

[Biemond 99] J. Biemond, P.M.B. Van Roosmalen & R.L. Lagendijk. *Improved Blotch Detection by Postprocessing*. In Proceedings of ICASSP 1999, 1999.

[Blume 02] G. Blume, G. Herczeg, O. Erdler & T.G. Noll. *Object Based Refinement of Motion Vector Fileds Applying Probabilistic homogenization rules*. IEEE Transactions on Consumer Electronics, vol. 48, no. 3, pages 694–701, August 2002.

[Boissonnat 84] J.D. Boissonnat. *Geometric structures for three-dimensional shape representation*. ACM Trans. Graph., vol. 3, no. 4, pages 266–286, 1984.

[Botsch 05] M. Botsch, A. Hornung, M. Zwicker & L. Kobbelt. *High-Quality Surface Splatting on Today's GPUs*. In Proceedings of the Eurographics Symposium on Point-Based Graphics, pages 17–24, June 2005.

[Boubekeur 06] T. Boubekeur, P. Reuter & C. Schlick. *Local Reconstruction and Visualization of Point-Based Surfaces Using Subdivision Surfaces*. Computer Graphics and Geometry, vol. 8, no. 1, 2006.

[Bretschneider 00] T. Bretschneider, O. Kao & P. Bones. *Removal of Vertical Scratches in Digitised Historical Film Sequences using*

Wavelet Decomposition. In Proceedings of the Image and Vision Computing New Zealand, pages 37–42, 2000.

[Checchin 97] P. Checchin, L. Trassoudaine & J. Alizon. *Segmentation of Range Images into Planar Regions.* 3dim, vol. 00, page 156, 1997.

[Chen 01] B. Chen & X. Nguyen. *POP: A Hybrid Point and Polygon Rendering System for Large Data.* In Proceedings Visualization 2001, pages 45–52, 2001.

[Cheng 04] K. D. Cheng, W. Wang, H. Qin, K. K. Wong, H. Yang & Y. Liu. *Fitting Subdivision Surfaces to Unorganized Point Data Using SDM.* In PG '04: Proceedings of the Computer Graphics and Applications, 12th Pacific Conference on (PG'04), pages 16–24, Washington, DC, USA, 2004. IEEE Computer Society.

[Cobzas 01] D. Cobzas & H. Zhang. *Planar Patch Extraction with Noisy Depth Data.* Proceedings of 3-D Digital Imaging and Modeling, pages 240 – 245, 2001.

[Cohen 01] J. D. Cohen, D. G. Aliaga & W. Zhang. *Hybrid Simplification: Combining Multi-resolution Polygon and Point Rendering.* In Proceedings Visualization 2001, pages 37–44, 2001.

[Criminisi 03] A. Criminisi, P. Pérez & K. Toyama. *Object Removal by Exemplar-Based Inpainting.* In CVPR (2), pages 721–728, 2003.

[Curless 96] B. Curless & M. Levoy. *A volumetric method for building complex models from range images.* In SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques, pages 303–312, New York, NY, USA, 1996. ACM Press.

[Dachsbacher 03] C. Dachsbacher, C. Vogelgsang & M. Stamminger. *Sequential point trees.* ACM Transactions on Graphics, vol. 22, no. 3, pages 657–662, 2003.

[Dane 04] G. Dane & T. Nguyen. *Motion Vector Processing for Frame Rate Up Conversion.* In Proceedings of IEEE 2004

International Conference on Acoustics, Speech and Signal Processing, 2004.

[Décoret 03] X. Décoret, F. Durand, F. Sillion & J. Dorsey. *Billboard Clouds for Extreme Model Simplification.* In Proceedings of the ACM Siggraph. ACM Press, 2003.

[Dell'Acqua 02] F. Dell'Acqua & R.t Fisher. *Reconstruction of Planar Surfaces Behind Occlusions in Range Images.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 4, pages 569–575, 2002.

[Dey 01] T. K. Dey, J. Giesen & J. Hudson. *Delaunay based shape reconstruction from large data.* In PVG '01: Proceedings of the IEEE 2001 symposium on parallel and large-data visualization and graphics, pages 19–27, Piscataway, NJ, USA, 2001. IEEE Press.

[Dey 02] T. K. Dey & J. Hudson. *PMR: Point to Mesh Rendering, A Feature-Based Approach.* In VIS '02: Proceedings of the conference on Visualization '02. IEEE Computer Society,Washington, DC, 2002.

[Dey 03] T. K. Dey & S. Goswami. *Tight cocone: a water-tight surface reconstructor.* In SM '03: Proceedings of the eighth ACM symposium on Solid modeling and applications, pages 127–134, New York, NY, USA, 2003. ACM Press.

[Dey 05a] T. K. Dey & J. Sun. *An Adaptive MLS Surface for Reconstruction with Guarantees.* In Symposium on Geometry Processing, pages 43–52, 2005.

[Dey 05b] T. K. Dey & J. Sun. *Normal and feature approximations from noisy point clouds.* Rapport technique OSU-CISRC-7/50-TR50, Ohio State University, Department of Computer Science and Engineering, July 2005.

[Duda 00] R. O. Duda, P. E. Hart & D. G. Stork. Pattern classification. Wiley-Interscience Publication, 2000.

[Duflou 05] J. R. Duflou, J. Váncza & R. Aerens. *Computer aided process planning for sheet metal bending: a state of the*

art. Computers in Industry, vol. 56, no. 7, pages 747–771, 2005.

[Edelsbrunner 92] H. Edelsbrunner & E. P. Mücke. *Three-dimensional alpha shapes*. In VVS '92: Proceedings of the 1992 workshop on Volume visualization, pages 75–82, New York, NY, USA, 1992. ACM Press.

[Edelsbrunner 01] H. Edelsbrunner. Geometry and topology for mesh generation. Cambridge University Press, New York, NY, USA, 2001.

[Elsayed 04] E. A. Elsayed & B. B. Basily. *A Continuous Folding Process for Sheet Materials*. International Journal of Materials and Product Technology, vol. 21, no. 1/2/3, pages 217 – 238, 2004.

[Fransens 03] J. Fransens, P. Bekaert & F. Van Reeth. *A Hierarchical PCA-based Piecewise Planar Surface Reconstruction Algorithm*. In Proceeding of the Eighth SIAM Conference on Geometric Design and Computing, pages 203 – 214, Brentwood, TN, USA, November 2003. Nashboro Press.

[Fransens 05] J. Fransens, F. Di Fiore & F. Van Reeth. *The Reconstruction of Missing Frames in Historical Films, a Layered Approach*. In Proceeding of the International Conference on Computer Graphics and Vision 2005 (GraphiCon05), pages 63 – 69, June 2005.

[Fransens 06a] J. Fransens. *Density Based 2D Delaunay Sculpting*. Technical Report TR-UH-EDM-0602, University of Hasselt, 2006.

[Fransens 06b] J. Fransens. *Planar Segmentation of Point Clouds for Hybrid Rendering*. Technical Report TR-UH-EDM-0602, University of Hasselt, 2006.

[Fransens 06c] J. Fransens & F. Van Reeth. *Hierarchical PCA Decomposition of Point Clouds*. In Proceedings of the Third International Symposium on 3D Data Processing, Visualization and Transmission , to appear, june 2006.

[Gamasutra 06] Gamasutra. World Wide Web, `http://www.gamasutra.com/features/20060105/davis_01.shtml`, 2006.

[Gelfand 04] N. Gelfand & L. J. Guibas. *Shape segmentation using local slippage analysis*. In SGP '04: Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing, pages 214–223, New York, NY, USA, 2004. ACM Press.

[Gotardo 04] P. F. U. Gotardo, K. L. Boyer, O. R. P. Bellon & L. Silva. *Robust Extraction of Planar and Quadric Surfaces from Range Images.* In ICPR (2), pages 216–219, 2004.

[Grossman 98] J. P. Grossman. Point sample rendering. Master's thesis, Dept. of Electrical Engineering and Computer Science, MIT, 1998.

[Han 97] S. Han & J.W. Woods. *Frame-Rate Up-Conversion Using Transmitted Motion and Segmentation Fields for Very Low Bit-Rate Video Coding*. In Proceedings of ICIP 1997, pages 747–750, 1997.

[Han 04] F. Han, Z. Tu & S. Zhu. *Range Image Segmentation by an Effective Jump-Diffusion Method*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 26, no. 9, pages 1138–1153, 2004.

[Hoffman 87] R. Hoffman & A. K. Jain. *Segmentation and classification of range images*. IEEE Trans. Pattern Anal. Mach. Intell., vol. 9, no. 5, pages 608–620, 1987.

[Hoover 96] A. Hoover, G. Jean-Baptiste, X. Jiang, P. J. Flynn, H. Bunke, D. B. Goldgof, K. Bowyer, D. W. Eggert, A. Fitzgibbon & R. B. Fisher. *An Experimental Comparison of Range Image Segmentation Algorithms*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 18, no. 7, pages 673–689, 1996.

[Hoppe 92] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald & W. Stuetzle. *Surface reconstruction from unorganized points*. In SIGGRAPH '92: Proceedings of the 19th annual conference on Computer graphics and interactive techniques, pages 71–78, New York, NY, USA, 1992. ACM Press.

[Hubert 05] M. Hubert, P. J. Rousseeuw & K. V. Branden. *ROBPCA: A New Approach to Robust Principal Component Analysis*. Technometrics, vol. 47, no. 1, pages 64 – 79, February 2005.

[J. Fransens 05] J. Fransens. World Wide Web, `http://research.edm.uhasselt.be/~fdifiore/research/GraphiCon2005/`, 2005.

[Jepson 02] A.D. Jepson, D.J. Fleet & M.J. Black. *A Layered Motion Representation with Occlusion and Compact Spatial Support*. In Proceedings of ECCV 2002, pages 692–706, 2002.

[Jiang 94] X. Jiang & H. Bunke. *Fast segmentation of range images into planar regions by scan line grouping*. Machine Vision Applications, vol. 7, pages 115–122, 1994.

[Jiang 00] X. Jiang, K. W. Bowyer, Y. Morioka, S. Hiura, K. Sato, S. Inokuchi, M. Bock, C. Guerra, R. E. Loke & J. M. Hans du Buf. *Some Further Results of Experimental Comparison of Range Image Segmentation Algorithms*. In ICPR '00: Proceedings of the International Conference on Pattern Recognition, pages 4877–4882, 2000.

[Johan 00] H. Johan, Y. Koiso & T. Nishita. *Morphing using Curves and Shape Interpolation Techniques*. In Proceedings of Pacific Graphics 2000, 8th Pacific Conference on Computer Graphics and Applications, pages 348–358, 2000.

[Johnson 98] R. A. Johnson & D. W. Wichern. Applied multivariate statistical analysis, 4th edition. Prentice Hall, 1998.

[Jolliffe 02] I. T. Jolliffe. Principal component analysis, 2nd edition. Springer-Verlag, New York, 2002.

[Joyeux 99] L. Joyeux, O. Buisson, B. Besserer & S. Boukir. *Detection and Removal of Line Scratches in Motion Picture Films*. In Proceedings of CVPR 1999, pages 548–553, 1999.

[Kaufman 90] L. Kaufman & P. J. Rousseeuw. Finding groups in data: An introduction to cluster analysis. John Wiley, 1990.

[Kobbelt 06] L. Kobbelt & M. Botsch. *A Survey of Point-Based Techniques in Computer Graphics, To Appear*. Computers & Graphics, 2006.

[Kokaram 98] A.C. Kokaram. Motion picture restoration: Digital algorithms for artefact suppression in degraded motion picture film and video. Springer Verlag, 1998.

[Lakaemper 06] R. Lakaemper & L. J. Latecki. *Extended EM for Planar Approximation of 3D Data*. In Proceedings of IEEE International Conference on Robotics and Automation (ICRA), May 2006.

[Langbein 01] F. Langbein, B. I. Mills, A. D. Marshall & R. R. Martin. *Finding approximate shape regularities in reverse engineered solid models bounded by simple surfaces*. In SMA '01: Proceedings of the sixth ACM symposium on Solid modeling and applications, pages 206–215, New York, NY, USA, 2001. ACM Press.

[Lee 96] S. Lee, G. Wolberg, K.-Y. Chwa & S. Y. Shin. *Image Metamorphosis with Scattered Feature constraints*. IEEE Transactions on Visualization and Computer Graphics, vol. 2, no. 14, pages 337–354, 1996.

[Lee 98] S. Lee, K.-Y. Chwa, J. Hahn & S. Y. Shin. *Fast Feature-based Volume Metamorphosis and Operator Design*. In Computer Graphics Forum, volume 17(3), pages 15–22, 1998.

[Levin 98] D. Levin. *The approximation power of moving least-squares*. Math. Comput., vol. 67, no. 224, pages 1517–1531, 1998.

[Levoy 85] M. Levoy & T. Whitted. *The use of points as a display primitive*. Rapport technique TR 85-022, University of North Carolina at Chapel Hill, 1985.

[Levoy 00] M. Levoy, 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 SIGGRAPH '00: Proceedings of the 27th annual conference on Computer

graphics and interactive techniques, pages 131–144, New York, NY, USA, 2000. ACM Press/Addison-Wesley.

[Liu 00a] S. Liu, J. Kim & C.C.J. Kuo. *MCI-embedded Motion Compensated Prediction for Quality Enhancement of Frame Interpolaton*. In SPIE Proceedings of International Symposium on Voice, Video, and Data Communications, Multimedia Systems and Applications III, 2000.

[Liu 00b] S. Liu, J. Kim & C.C.J. Kuo. *Non-Linear Motion-Compensated Interpolation for Low Bit Rate Video*. In SPIE Proceedings of International Symposium on Optical Science, Engineering, and Instrumentation 2000, Applications Digital Image Processing XXIII, 2000.

[Lorensen 87] W. E. Lorensen & H. E. Cline. *Marching cubes: A high resolution 3D surface construction algorithm*. In SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques, pages 163–169, New York, NY, USA, 1987. ACM Press.

[Lucas 81] B. D. Lucas & T. Kanade. *An Iterative Image Registration Technique with an Application to Stereo Vision (DARPA)*. In Proceedings of the 1981 DARPA Image Understanding Workshop, pages 121–130, April 1981.

[Maciel 95] P. W. C. Maciel & P. Shirley. *Visual navigation of large environments using textured clusters*. In SI3D '95: Proceedings of the 1995 symposium on Interactive 3D graphics, pages 95–106., New York, NY, USA, 1995. ACM Press.

[Nicolescu 03] M. Nicolescu & G. Medioni. *Layered 4D representation and voting for grouping from motion*. PAMI, vol. 25, no. 4, pages 492–501, April 2003.

[Nishita 93] T. Nishita, T. Fujii & E. Nakamae. *Metamorphosis using Bézier Clipping*. In Proceedings of 1st Pacific Conference on Computer Graphics and Applications, pages 162–173, 1993.

[Nunes 05] S. Nunes, D. Almeida, E. Loke & H. du Buf. *Polygon Optimisation for the Modelling of Planar Range Data*. In

Proceedings of 2nd Iberian Conference on Pattern Recognition and Image Analysis, pages 128 – 136, 2005.

[Park 92] B. U. Park & B. A. Turlach. *Practical performance of several data driven bandwidth selectors (with discussion).* Computational Statistics, vol. 7, no. 3, pages 251–270, 1992.

[Pauly 03a] M. Pauly, R. Keiser & M. H. Gross. *Multi-scale Feature Extraction on Point-sampled Surfaces.* Computer Graphic Forum, vol. 22, no. 3, pages 281–290, 2003.

[Pauly 03b] M. Pauly, R. Keiser, L. P. Kobbelt & M. Gross. *Shape modeling with point-sampled geometry.* ACM Transactions on Graphics, vol. 22, no. 3, pages 641–650, 2003.

[Peternell 04] M. Peternell & T. Steiner. *Reconstruction of piecewise planar objects from point clouds.* Computer-Aided Design, vol. 36, no. 4, pages 333–342, 2004.

[Pfister 00] H. Pfister, M. Zwicker, J. van Baar & M. Gross. *Surfels: Surface Elements as Rendering Primitives.* In SIGGRAPH 2000 Proceedings, pages 335–342, 2000.

[Pfister 04] H. Pfister & M. H. Gross. *Point-Based Computer Graphics.* IEEE Computer Graphics and Applications, vol. 24, no. 4, pages 22–23, 2004.

[Reuter 05] P. Reuter, P. J., J. Trunzler, T. Boubekeur & C. Schlick. *Surface Reconstruction with Enriched Reproducing Kernel Particle Approximation.* In Proceedings of the IEEE/Eurographics Symposium on Point-Based Graphics, pages 79–87. Eurographics/IEEE Computer Society, 2005.

[Rottensteiner 03] F. Rottensteiner. *Automatic Generation of High-Quality Building Models from Lidar Data.* IEEE Comput. Graph. Appl., vol. 23, no. 6, pages 42–50, 2003.

[Rusinkiewicz 00] S. Rusinkiewicz & M. Levoy. *QSplat: A Multiresolution Point Rendering System for Large Meshes.* pages 343–352. ACM Press / ACM SIGGRAPH / Addison Wesley Longman, 2000.

[Schallauer 99] P. Schallauer, A. Pinz & W. Haas. *Automatic Restoration Algorithms for 35mm Film*. Videre, vol. 1, no. 3, pages 60–85, 1999.

[Schaufler 98] G. Schaufler. *Per-Object Image Warping with Layered Impostors*. In Proceedings of the 9th Eurographics Workshop on Rendering, pages 145 –156, 1998.

[Schindler 03] K. Schindler. *Spatial subdivision for piecewise planar object reconstruction*. Videometrics VII, vol. 5013, no. 1, pages 194–201, 2003.

[Schuster 04] H. F. Schuster. *Segmentation of Lidar Data Using The Tensor Voting Framework*. In Proceedings of the 10th ISPRS Congress, volume 25, pages 1073 – 1078, July 2004.

[Shade 96] J. Shade, D. Lischinski, D. H. Salesin, T. DeRose & J. Snyder. *Hierarchical image caching for accelerated walkthroughs of complex environments*. In SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques, pages 75–82, New York, NY, USA, 1996. ACM Press.

[Shade 98] J. Shade, S. Gortler, L. He & R. Szeliski. *Layered depth images*. In SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques, pages 231–242, New York, NY, USA, 1998. ACM Press.

[Shum 97] H. Shum & R. Szeliski. *Panoramic Image Mosaics*. Rapport technique MSR-TR-97-23, Microsoft Research, 1997.

[Sithole 05] G. Sithole & G. Vosselman. *Filtering of airborne laser scanner data based on segmented point clouds*. International Archives of Photogrammetry and Remote Sensing, vol. 36, pages 66 –71, 2005.

[Skiena 98] S. S. Skiena. The algorithm design manual. Springer-Verlag New York, Inc., New York, NY, USA, 1998.

[Smith 04] P. Smith, T. Drummond & R. Cipolla. *Layered Motion Segmentation and Depth Ordering by Tracking Edges*. IEEE Trans. Pattern Anal. Mach. Intell., vol. 26, no. 4, pages 479–494, 2004.

[Tal 99] A. Tal & G. Elber. *Image Morphing with Feature Preserving Texture*. In Proceedings of Eurographics (EG1999), volume 18(3), pages 339–348, 1999.

[Tekalp 95] A.M. Tekalp. Digital video processing. Prentice-Hall, 1995.

[Tenze 00] L. Tenze, G. Ramponi & S. Carrato. *Blotches Correction and Contrast Enhancement for Old Film Pictures*. In Proceedings of International Conference on Image Processing 2000, pages 660–663, 2000.

[Turk 94] G. Turk & M. Levoy. *Zippered polygon meshes from range images*. In SIGGRAPH '94: Proceedings of the 21st annual conference on Computer graphics and interactive techniques, pages 311–318, New York, NY, USA, 1994. ACM Press.

[Valette 04] S. Valette, I. Magnin & R. Prost. *Mesh-based video objects tracking combining motion and luminance discontinuities criteria*. Signal Processesing, vol. 84, no. 7, pages 1213–1224, 2004.

[Van den Bergh 02] J. Van den Bergh, F. Di Fiore, J. Claes & F. Van Reeth. *Interactively Morphing Irregularly Shaped Images Employing Subdivision Techniques*. In Proceedings of 1st Ibero-American Symposium in Computer Graphics (SIACG2001), pages 315–321, July 2002.

[Várady 97] T. Várady, R. R. Martin & J. Cox. *Reverse engineering of geometric models - an introduction*. Computer-Aided Design, vol. 29, no. 4, pages 255–268, 1997.

[Volodine 03] T. Volodine, D. Roose & D. Vanderstraeten. *Efficient Triangulation of Point Clouds Using Floater Parameterization*. In Proceeding of the Eighth SIAM Conference on Geometric Design and Computing, pages 523 – 536, Brentwood, TN, USA, November 2003. Nashboro Press.

[Vosselman 99] G. Vosselman. *Building Reconstruction using Planar Faces in Very High Density Height Data*. In Proceedings of ISPRS Automatic Extraction of GIS Objects from Digital Imagery, pages 87–92, September 1999.

[Vosselman 04] G. Vosselman, B.G.H. Gorte, G. Sithole & T. Rabbani. *Recognising structure in laser scanner point clouds.* In International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, volume 46, pages 33 – 38, Freiburg, Germany, October 2004.

[Vu 01] T.Thuy Vu & Mitsuharu Tokunaga. *Wavelet and Scale-Space Theory in Segmentation of Airborne Laser Scanner Data.* Proceedings of the Asian Conference on Remote Sensing, vol. 1, pages 176 – 180, 2001.

[Wahl 05] R. Wahl, M. Guthe & R. Klein. *Identifying Planes in Point-Clouds for Efficient Hybrid Rendering.* In The 13th Pacific Conference on Computer Graphics and Applications, October 2005.

[Wang 94a] J. Y.A. Wang & E. H. Adelson. *Representing Moving Images with Layers.* IEEE Transactions on Image Processing Special Issue: Image Sequence Compression, vol. 3, no. 5, pages 625–638, September 1994.

[Wang 94b] Y. Wang & O. Lee. *Active Mesh - A Feature Seeking and Tracking Image Sequence Representation Scheme.* IEEE Trans. Image Processing, vol. 3, pages 610–624, 1994.

[Wang 96] Y. Wang & O. Lee. *Use of two-dimensional deformable mesh structures for video coding.* IEEE Transactions on Circuits and Systems for Video Technology, vol. 6, no. 6, pages 636 – 646, 1996.

[Wang 05] J. Wang, P. Bhat, R. A. Colburn, M. Agrawala & M. F. Cohen. *Interactive video cutout.* ACM Transactions on Graphics, vol. 24, no. 3, pages 585–594, 2005.

[Wicke 05] M. Wicke, S. Olibet & M. Gross. *Conversion of Point-Sampled Models to Textured Meshes.* In Proceedings of the Eurographics Symposium on Point-Based Graphics '05, pages 119–124, 2005.

[Wolberg 90] G. Wolberg. Digital image warping. IEEE Computer Society Press, 1990.

[Xiao 04] J. Xiao & M. Shah. *Motion layer extraction in the presence of occlusion using graph cut.* In Proceedings of CVPR 2004, pages 972–979, 2004.

[Xiao 05] J. Xiao & M. Shah. *Accurate Motion Layer Segmentation and Matting.* In CVPR, pages 698–703, 2005.

[Zhang 05] Y. Zhang, J. Xiao & M. Shah. *Motion Layer Based Object Removal in Videos.* In Proceedings of the IEEE Workshop on Applications of Computer Vision, Breckenridge, CO, USA, January 2005. to appear.

[Zitnick 04] L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder & R. Szeliski. *High-quality video view interpolation using a layered representation.* ACM Transactions on Graphics, vol. 23, no. 3, pages 600–608, 2004.

[Zwicker 01] M. Zwicker, H. Pfister, J. van Baar & M. Gross. *Surface Splatting.* In SIGGRAPH 2001 Proceedings, 2001.