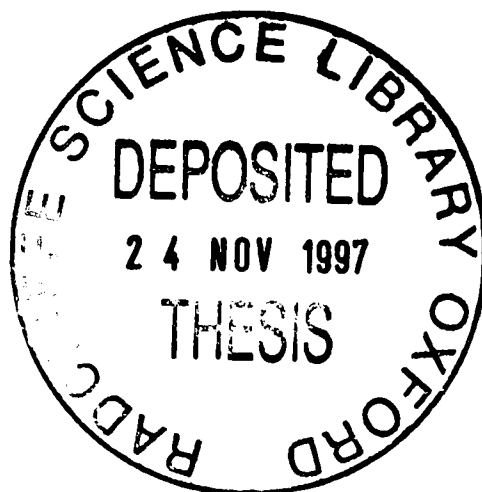


Characterisation of Foams Using Vision Systems

PD Thomas
Balliol College

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
at the University of Oxford
Hilary Term 1997



Abstract

The macroscopic behaviour of a two-phase foam depends on chemical properties such as surface tension, and physical properties such as the shape and size of the bubbles in the foam. The chemical properties of a foaming material may be deduced from experiments on the material in the single phase, for example surface tension or viscosity measurements. However, in order to measure structure the foam must clearly be present. This presents a difficult problem, especially for liquid foams which are often fast-moving and liable to collapse or rearrange on coming into contact with a probe.

Optical techniques for examining foam structures can be non-invasive and take advantage of the semi-transparent nature of many liquid and solid foams. In particular, the application of confocal and axial tomography systems to real three-dimensional cellular foams can resolve their local geometric structure. This thesis covers the application of new optical tomographic techniques to the imaging of foams and presents the first three-dimensional models of bubble structure in liquid and solid foams. Complete descriptions of the hardware and software are included; imaging systems are based on a personal computer with an inexpensive video digitising card and a CCD camera.

A clear advantage of optical systems is that the high data bandwidth required for tomography is available even on relatively slow computers by modern standards. Typically, all the data for a sixteen-million (256^3) voxel model covering a world volume of 64mm^3 can be acquired in a period of 30 seconds.

Results from this work include the first three-dimensional solid models of observed foams, both aqueous and polyurethane-based, including volume models of minimal energy cellular foam configurations. In particular, Kelvin's proposed minimal cell and also bubbles from the Weaire-Phelan structure have been resolved by the system.

Contents

1	Introduction	1
1.1	Foams	1
1.2	Soap foams	3
1.3	Foam structure and research motivation	8
2	Foam research literature	11
2.1	Cellular foam structure fundamentals	11
2.1.1	Two-dimensional foams	12
2.1.2	Three-dimensional foams	15
2.2	Foam models	19
2.2.1	Two-dimensional cellular models	21
2.2.2	Three-dimensional models	23
2.3	Experimental work	27
2.4	Optical methods for three dimensions	36
3	A short chapter about the Z Notation	37
3.1	Introduction	37
3.2	Z fundamentals	38
3.3	Schemas	40

3.4	Other expressions	41
4	Focal plane photography	43
4.1	Introduction	43
4.2	Experimental apparatus	44
4.3	Image processing	47
4.4	Image analysis	53
4.5	Reconstruction and Visualisation	58
5	Optical axial tomography	73
5.1	Tomographic principles	73
5.2	Interferometry	76
5.3	Three-dimensional X-ray angiography	78
5.4	Microtomography	83
5.4.1	Optical microscope tomography	83
5.4.2	Optical tomography of a cochlea	84
5.5	Process tomography	86
5.6	Optical foam tomography	87
5.6.1	Reconstruction of solid foams	96
5.6.2	Reconstruction of liquid foams	106
5.7	Results	108
6	Optimised method for liquid foam	118
6.1	Line models	119
6.2	Experimental apparatus and method	119
6.3	A reconstruction algorithm	122
6.3.1	Calculation of ray paths	123

6.3.2	Resolving ray intersections	125
6.3.3	Locating Plateau border vertices	129
6.3.4	Connecting vertices	131
6.3.5	Model reality-check	133
6.3.6	Complete reconstruction specification	135
6.4	Results	139
7	Conclusions	145
7.1	Reconstruction method pros and cons	145
7.1.1	Focal plane photography	146
7.1.2	Optical axial tomography	148
7.1.3	Optimised tomographic method	149
7.1.4	Software execution times	150
7.2	Improvements and future work	152
7.3	Concluding remarks	155
A	Published papers	156

List of Figures

1.1	Air/water foam structure; gas fraction increases from left to right.	4
1.2	Cross-section through a Plateau border.	6
1.3	Fluid flow between lamella and Plateau border.	7
2.1	Foam lamellae undergo a $T1$ change to form two stable 3-vertices.	13
2.2	A $T2$ change occurs when a small bubble shrinks to zero size.	15
2.3	Foam topological transformations in three dimensions.	16
2.4	Kelvin's minimal tetrakaidecahedron.	24
2.5	Weaire-Phelan minimal cell geometry (actual structure has slightly curved faces) consisting of six 14-sided polyhedra and two regular pentagonal dodecahedra.	26
2.6	Stereology grid points on a two-phase material sample.	28
2.7	Bubbles next to vessel walls in a soap foam, drawn by Matzke.	32
2.8	Bubbles in the centre of a soap foam, drawn by Matzke.	33
2.9	Simple X-ray tomography arrangement.	35
4.1	Lens configurations for varying depths of field.	44
4.2	The experimental foam rig.	45

4.3	7×7 convolution kernels for edge detection.	54
4.4	The same image before (upper) and after segmentation.	59
4.5	The skeletonisation operator.	60
4.6	Configurations where the central pixel is deleted during the skeletonisation process.	60
4.7	Three-dimensional line model of aqueous foam.	64
4.8	Three-dimensional voxel model of aqueous foam, after dilation (top) and after annealing (bottom).	67
4.9	Foam line model after skeletonisation and padding.	68
4.10	Connection of line segments in the Plateau border model.	68
4.11	Three-dimensional line model of aqueous foam after smoothing.	71
5.1	A two-dimensional tomographic system with two orthogonal projections, sampling (left) and reconstruction (right).	74
5.2	An optical tomography system based on laser interferometry.	77
5.3	Simulated annealing tomographic reconstruction.	82
5.4	Optical microscope configuration.	84
5.5	Optical tomography system configuration.	85
5.6	Tomographic reconstruction, using visible light, of the inner ear of a guinea pig	86
5.7	Experimental rig for foam tomography.	87
5.8	Determination of image centre of rotation.	89
5.9	One frame from a typical foam video, with image centre line and cropping lines marked.	90
5.10	Bubble voxel model, viewed as a transparent solid (left), and as an opaque isosurface (right).	91

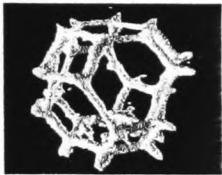
5.11	Voxel model of a Kelvin bubble, viewed as a transparent solid and as an opaque isosurface.	92
5.12	Results of applying optical tomography to a bent wire phan- tom with differing camera orientations (left to right and top to bottom, 0° , 10° , 20° , 30° , 40° and 45° . Angle is measured from horizontal, axis of rotation of object is vertical.	94
5.13	A selection of solid, open polyurethane foams.	97
5.14	Actual camera geometry for tomography.	98
5.15	Solving for x and y gives the value of the pixel in frame F , and hence the intensity for a ray passing through F to a voxel v .100	
5.16	Voxel and frame position after rotations.	104
5.17	Unfiltered (top) and filtered (bottom) Weaire-Phelan bubble. .	109
5.18	Varying thresholds for a 16-sided bubble model. Threshold values are (in rows, top to bottom) 90, 100, 110, 120, 130 and 140.	110
5.19	Close-up of a Plateau border junction. Threshold values are (in rows, top to bottom) 90, 100, 110, 120, 130 and 140.	112
5.20	The Kelvin Cell and close-up sections.	114
5.21	Solid foam 1.	115
5.22	Solid foam 2.	115
5.23	Solid foam 3.	116
5.24	Solid foam 4.	116
6.1	Two digitised images—left, at 160×120 , right, at 640×480 . .	121
6.2	User selection of Plateau border junctions in an image.	122
6.3	User selection of Plateau borders in an image.	136

6.4 Diamond foam lamellae (video top, wire model bottom). Each diamond is of side length approximately 2mm. 141

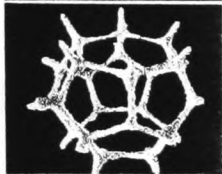
6.5 Weaire-Phelan foam bubble. The bubble diameter is approximately 5mm. 142

6.6 Kelvin foam bubbles. Individual bubble diameter is approximately 5mm. 144

List of Thumb-movies



A Kelvin cell



A fourteen-sided Weaire-Phelan cell



A sixteen-sided bubble



A Kelvin cell fragment



A polyurethane solid foam fragment



A denser polyurethane foam



A polyurethane foam



A polyurethane foam

To view the movies, hold this book in the right hand and allow pages to fall in reverse order from the left hand while looking at the right margin.

Chapter 1

Introduction

1.1 Foams

Foam (fōum), *sb.* [OE. *fám* :—WGer. **faimo-* :—pre-Teut. **poimo-* or **phoimo-*.] **1.** The aggregation of minute bubbles formed in water or other liquid by agitation, fermentation effervescence, etc. **b. spec.** The foaming saliva issuing from the mouth in epilepsy, rabies, etc. OE. Also *fig.* **2.** Foaming water, the sea. *arch.* OE. **3.** *Min.* = APHRITE.

The Shorter Oxford English Dictionary, 1959.

It might seem, given the simple dictionary definition of *foam*, that foaming research would feature in only a small number of modern scientific disciplines and have a similarly-sized impact on our everyday lives. In fact, the opposite is true—research into foams and multi-phase mixtures is essential for the development of many domestic items we take for granted. Moreover, the same work has links with a wide range of theories on subjects as diverse as cucumber cell shapes and the placement of matter in the universe.

Anybody who wet-shaves in the morning will appreciate the enhanced stability of shaving foam which enables it to act as a better lubricant for the

blades. We have foam padding in chairs and mattresses which is rather more elastic and deformable than equivalent single-phase solid material. Foams abound in the food and drinks industry, from chocolate bars to ice-cream and carbonated soft drinks to beer [1]. The Germans and Danes prefer their beer to have a thick, stable head of froth, whereas in Britain we are happier when the head collapses quickly and the barman can fill our glass right up to the brim.

The different physical properties of all these foams have been tuned by extensive experimentation based on some theoretical work. However, it is not yet possible to deduce the complete foaming behaviour of a mixture from the physical and chemical properties of its components. It is possible to make a reasonable guess, but even in two-component mixtures the interactions between each component can be complex, and these lead to sometimes surprising foaming behaviour.

An idea of the extent of foam structure research in other fields is given by Glazier and Weaire [2] in their paper on cellular patterns, from which is taken table 1.1. Each of the cellular systems listed here share a common structure of homogeneous domains separated by boundaries of varying thickness, depending on the volume ratio of the component materials.¹ In many cases, the structure of the cellular network is defined by minimum boundary energy constraints—the presence of a surface tension force in the domain borders suggests that a stable low-energy state is formed when the area of these borders is minimised. Even the wax used by bees in the manufacture of their honeycombs is subject to cellular boundary constraints [4].

¹A nice graphical depiction of metal grain boundaries is given in [3].

System	Type
Soap froth	Bubbles
Lipids	Bubbles
Metal	Grains
Ceramics	Grains
Potts model	Domains
Iron garnets	Domains
Cucumber	Cells
Cornea of eye	Cells
Geography	Territories
Universe	Voids

Table 1.1: Cellular systems [2].

1.2 Soap foams

Much time and effort has been devoted to researching one of these cellular systems, that of the air and water mixture stabilised by a soap or detergent surfactant. The structure of this system varies between a bubbly froth and a cellular network depending on the fraction of air in the mixture. For a low gas fraction, independent spherical bubbles are free to move through the liquid under the action of gravity until they come close to other bubbles, when they might coalesce. As the gas fraction is increased, the movement of bubbles becomes increasingly constrained by their neighbours and the liquid boundaries between bubbles become thinner and less mobile. Eventually, the bubbles are so large that motion becomes nearly impossible and a stable cellular network begins to form. At the limit, where hardly any liquid is present, the

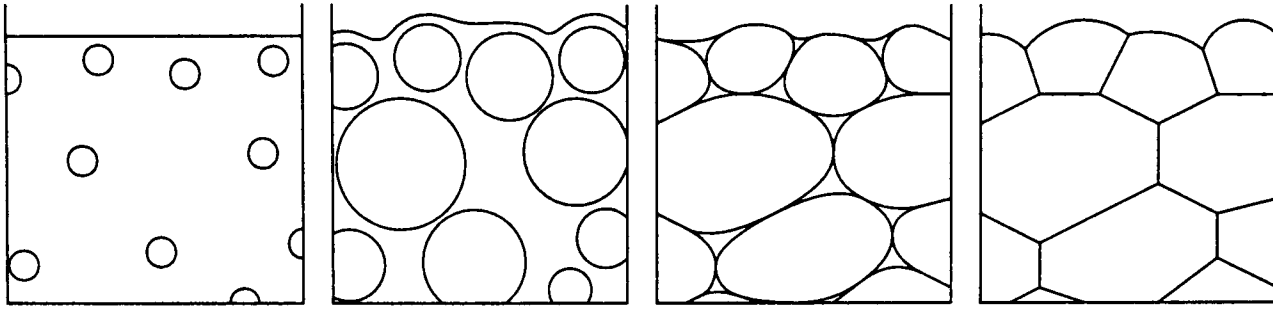


Figure 1.1: Air/water foam structure; gas fraction increases from left to right.

foam cells are separated by extremely thin (of the order of microns or less) liquid lamellae. Figure 1.1 shows this evolution in two dimensions. Hrma [5] uses the terms ‘bulk foam’ and ‘surface foam’ to distinguish between foams with low and high gas fractions respectively, and his list of some typical characteristics of each type is shown in table 1.2.

It is easy to understand the physics governing the stability of the low-gas-fraction case above. The free liquid surface surrounding each bubble pulls the gas into a spherical shape which minimises surface area and hence surface energy. The surface tension is constant throughout.

But at the other extreme, where the large gas bubbles are separated from each other by thin liquid lamellae, what prevents the liquid structure collapsing into droplets or draining away under the action of gravity? The answer lies in the uneven distribution of surface forces along the gas-liquid interfaces. The soap and water system was studied in detail by Plateau [6] in the nineteenth century, and from his work the name *Plateau border* is given to the liquid channel which lies between two lamellae. The variation of surface forces between the Plateau border and lamella provides stability for the foam and hinders immediate collapse.

²If the bubble rising velocity is v_r and the bubble radius is r then $v_r \gg dr/dt$.

	Surface foam	Bulk foam
Example	Soap foam	Foam glass
Mode of generation	Accumulation of bubbles transported by buoyancy to liquid surface	Growth of bubbles within liquid due to gas pressure
Conditions for generation	Bubble rising velocity ² $\gg$ bubble growth rate (low viscosity, high gravity)	Bubble rising velocity $\ll$ bubble growth rate (high viscosity, low gravity)
Steady state	Balance between bubble supply through foam bottom and burst at foam surface	Balance between bubble nucleation and collapse vent holes
Upper limit for steady dynamic foam	Either critical gas flux or transition to bulk foam	
Lower limit for steady dynamic foam	Minumum gas flux	Critical gas holdup
Film thinning mechanism	Gravity and capillary drainage	Bubble expansion

Table 1.2: Differences between ‘surface’ and ‘bulk’ foams [5].

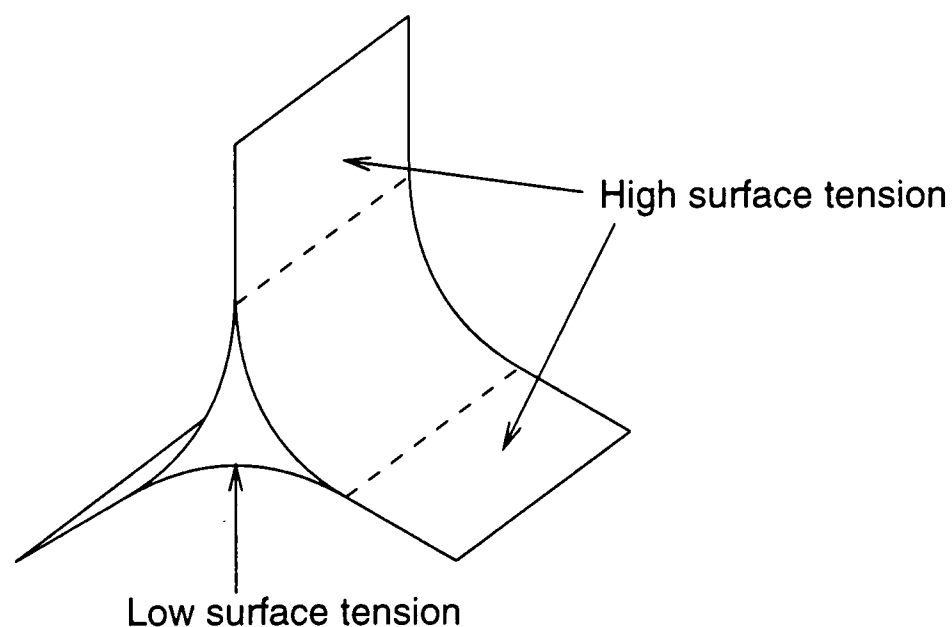


Figure 1.2: Cross-section through a Plateau border.

A cross-section of a Plateau border is shown in figure 1.2. In a stable foam, the Plateau border surface is virtually stationary and liquid drains from the lamellae into the border. Liquid then flows along the centre of the Plateau border into its neighbours. Often, the Plateau border is much thicker than the lamellae and contains the vast majority of the liquid in the foam.

The rate of drainage of liquid from the lamellae depends on the surface tension gradient between the lamellae and the Plateau border. This difference in surface forces arises because of the uneven distribution of surfactant molecules on the lamellae and Plateau border. The higher concentration of surfactant in the Plateau border causes the surface tension to be significantly reduced in this region compared to that in the lamellae. Figure 1.3 shows a common model for the liquid flow between lamella and Plateau border. The lamella is assumed to be flat-sided and therefore has no surface shear stress; the Plateau border surface is of constant curvature and there is a very small

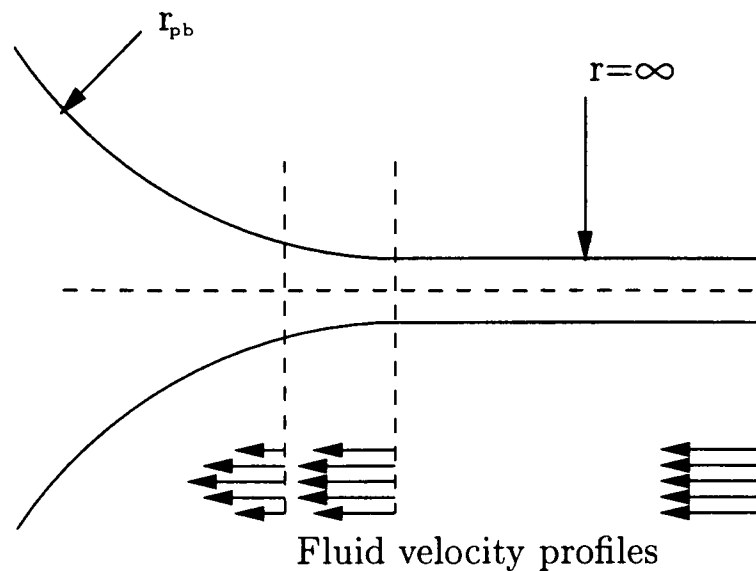


Figure 1.3: Fluid flow between lamella and Plateau border.

liquid section between these regimes where the change in curvature occurs.

The increased curvature of the Plateau border implies a lower liquid pressure compared to the pressure in the lamellae. This pressure gradient causes liquid to drain into the Plateau border. In the lamella the liquid moves by plug flow, but the surface tension gradient which inhibits liquid flow into the Plateau border causes this to distort to a parabolic flow profile where the surface liquid travels at a much reduced velocity. It is this effect which prevents liquid draining rapidly out of the lamella, and therefore improves the stability of the foam. Presumably the larger the surface tension gradient the more stable the foam becomes as the resistance to drainage increases.

This model is a popular one but it has a few problems. It seems unlikely that a liquid surface would have a relatively sudden gradient discontinuity for any reason. This would mean a very high local concentration of surfactant molecules at the gradient step and also the constant radius shape does not give an overall minimal length surface. There is also a problem with the plug flow assumption which must follow if a parallel-sided lamella is assumed.

What happens in the centre of the lamella, where it would seem there must be a liquid source? And how can the lamella become thinner otherwise if the flow is perpendicular to the direction of thinning? Foam lamellae which are draining cannot have parallel sides in nature but the difference between this and the model is small enough that the extra complexity needed for a more accurate model often does not make the latter worthwhile.

Although this thesis is concerned with two-phase liquid/gas or solid/gas foams, three-phase foams are possible. Solid particles may be suspended at the interfaces of a gas/liquid foam, see [7], and this can be useful in separation processes relying on flotation of solids. A group in Manchester is using optical techniques to resolve bubbles in a water/air/coal-dust foam flotation system, see [8].

1.3 Foam structure and research motivation

Foaming research neatly divides into two main subject areas concentrating either on the low-level chemical processes responsible for foam formation and stability or on the physical structure of foam bubbles and other cellular networks. A thorough understanding of both these areas is needed for a complete model of foaming behaviour and bulk properties. A chemical engineer designing a distillation vessel (for example [9]) would very much like to know what chemicals in the reactants produce foam and how, and also what are the values of bulk properties such as interfacial area per cubic metre. A designer of solid foam filling for cushions needs to know how the production process affects the size and shape of cells in the foam so that its bulk elastic properties can be varied.

The scope of this thesis concerns the structural half of the foam problem. Less attention is paid to the chemical nature of foam formation, see [10], instead focusing on the determination of foam geometry using non-invasive optical methods. The breakdown by chapters is

- **Chapter two** outlines the state of structural research so far, including computer simulation of random two-dimensional foams and special cases of three-dimensional foam cells, measurements of real foams and other systems with a cellular arrangement. Also, some other non-invasive techniques such as gamma or X-ray analysis and their reconstruction software are discussed.
- **Chapter three** is a short overview of the *Z Notation*, a mathematical framework based on set theory which is used to describe the software developed for the following tomographic projects.
- **Chapter four** details the experimental work done to develop an optical measuring system based on confocal imaging of a foam. Although simple to set up, it is difficult to get good results due to the asymmetry of the data acquisition system.
- **Chapter five** describes the implementation of an optical tomography scanner which solves many of the problems of the confocal system. Much better experimental results are presented, however, the cost is increased computation complexity.
- **Chapter six** is a description of an alternative video-based method which reduces the computational burden of tomography whilst still

producing a model of foam cells. Unfortunately, though, some human help is needed in this process.

- **Chapter seven** concludes the work with a discussion of the results from chapters 4–6 and offers some insight into what research might be feasible and useful in the future.

Chapter 2

Foam research literature

In the past many researchers have focussed on macroscopic models of foam behaviour, such as those for flow through pipes and foam heights in stirred vessels. One reason for this is that the microscopic details of foam structure, particularly in cellular foams, are very hard to observe and model. However, since a thorough understanding of the bulk properties of foams must eventually take into account the cellular network geometry, more attention is now being concentrated in this area. In this chapter, an overview of previous work is presented along with work in related fields which may be applicable to foaming research.

2.1 Cellular foam structure fundamentals

When a two-phase mixture forms, what is the driving force which governs how the bubbles arrange themselves in a foam? The answer is that, as might be expected, a foam settles into a minimum energy configuration. The short-term minimum is when the interfacial surface area is minimised since surface forces are initially the largest and quickest-acting. Eventually,

gravity becomes the dominant force in the system, and its downward pull on the liquid drains the foam, possibly causing further rearrangement of cells as the minimum surface energy constraint always applies.

It is clear that bubbles in a minimal configuration in a foam of low gas fraction would be spherical. There is little interaction between bubbles and no constraint on the shape other than the minimum surface area consideration. However, this spherical packing is not possible in a cellular foam of significantly high gas content, as the liquid channels separating the bubbles are thin enough to permit interaction between the latter. Bubbles may coalesce, or stay separate and deform to fit closely together. It is still not clear exactly what cell shape, or shapes, which must tessellate perfectly, minimises the surface area of a random foam—but this has been the subject of intense speculation. It is important because many researchers have assumed one cellular structure or another in their foam theory and computational simulation. If the reality were to differ widely from the theory, some empirically-derived relationships might be suspect. The topic of minimal surfaces and the search for a cellular structure which could be successfully used in foam theories is discussed later in this chapter. But first, we should consider some basic characteristics shared by many cellular networks.

2.1.1 Two-dimensional foams

In its simplest form, the foam structure problem applies to a two-dimensional cellular network where the only force acting on the system is a uniform surface tension which tends to pull the material into a minimum energy arrangement. This is a much better understood system than the equivalent

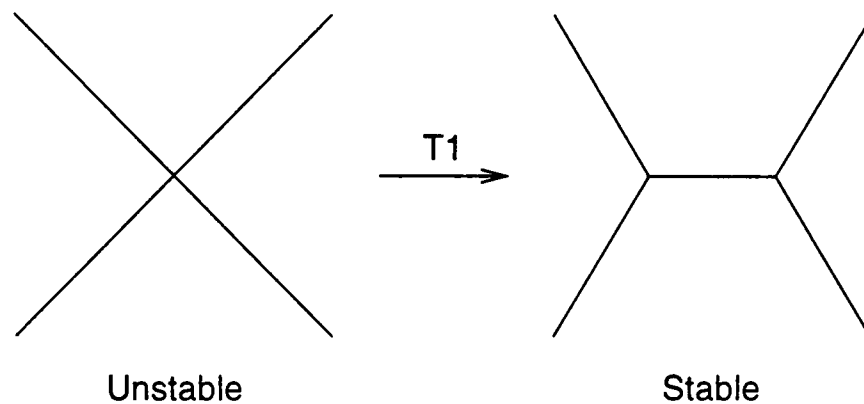


Figure 2.1: Foam lamellae undergo a $T1$ change to form two stable 3-vertices.

three-dimensional case, mainly because the underlying mathematical theory is reasonably easy to understand and numerical simulations are feasible on modern computers.

A theoretical two-dimensional foam with infinitesimally small liquid content can be represented by an undirected graph containing lines of zero width. The total energy of the foam is simply proportional to the total line length. If the foam is in an equilibrium state, there are several geometrical rules which must be obeyed,

- Exactly three lamellae meet at a Plateau border.
- Lamellae join at local angles of 120° .
- Lamellae are either straight lines or arcs of constant curvature.

Figure 2.1 gives a clue as to why there would never be more than three lamellae joining together. The total side length of the lamellae joining at the four-fold vertex is 3.5% greater than that of those at joining at the two three-fold vertices, as pointed out by Glazier and Weaire [2]. Hence the higher energy 4-vertex splits into two lower energy 3-vertices. This decay

involves a topological change, and could happen in a foam if two 3-vertices come together as a result of their shared lamella shrinking to zero length. The accepted notation for this is a ‘ $T1$ ’ process, and it is the most common type of rearrangement in a foam. The ‘ $T2$ ’ change is discussed later. Topological changes often occur on a much faster time scale than the general evolution of the foam.

The surface tension along each lamella must be constant and identical for all. This means that for a vertex to be stable the three lamellae must make angles of 120° with each other, as each is exerting a force of the same magnitude on the other two. The only way to balance these forces is by ensuring that they meet at equal angles.

Even though locally lamellae meet at equal angles, they must still be curved if the pressure in bubbles on their faces is unequal. The pressure difference is constant across a lamella so the curvature must also be constant, and the radius of curvature, r , is related to the pressure difference by

$$r = 2\sigma/\Delta P$$

where ΔP is the pressure difference across the lamella and σ is the surface tension.

The other type of topological change, the ‘ $T2$ ’ process, occurs in foams which have gases diffusing between bubbles across lamellae (figure 2.2). If all the gas diffuses out of a cell, then it collapses and its exterior lamellae become joined. Further $T1$ transformations might also occur if the collapse of the original bubble created a greater than three-fold vertex.

An interesting rule was found by von Neuman [11], who showed that for

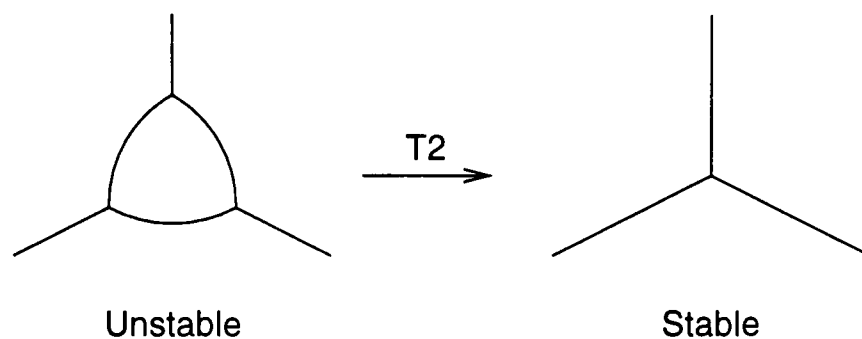


Figure 2.2: A $T2$ change occurs when a small bubble shrinks to zero size.

a two-dimensional bubble of area A and number of sides n ,

$$\frac{dA}{dt} = K(n - 6)$$

where K is a constant value. This surprising result claims that bubbles with less than six sides will shrink in a diffusive system, whereas bubbles with more than six sides will grow. However, it correctly predicts the state of play for a network consisting entirely of equal-sided hexagons—no change occurs. This is a special case, a *monodisperse* foam, where all the bubbles have equal size, which is the minimum energy configuration for a two-dimensional foam. Some researchers have used the hexagonal bubbles as a model for studying macroscopic properties, for example Kraynik and Reinelt in their studies on flowing foams [12, 13, 14].

2.1.2 Three-dimensional foams

The step up from theory and experiment on two-dimensional foams to three dimensions is a large one. Much of the mathematical simplicity of two-dimensional network modelling vanishes in a haze of complex curvatures and non-tessellating three-dimensional shapes. Plateau borders in three dimensions lie between three bubble facets which meet at 120° and join with three

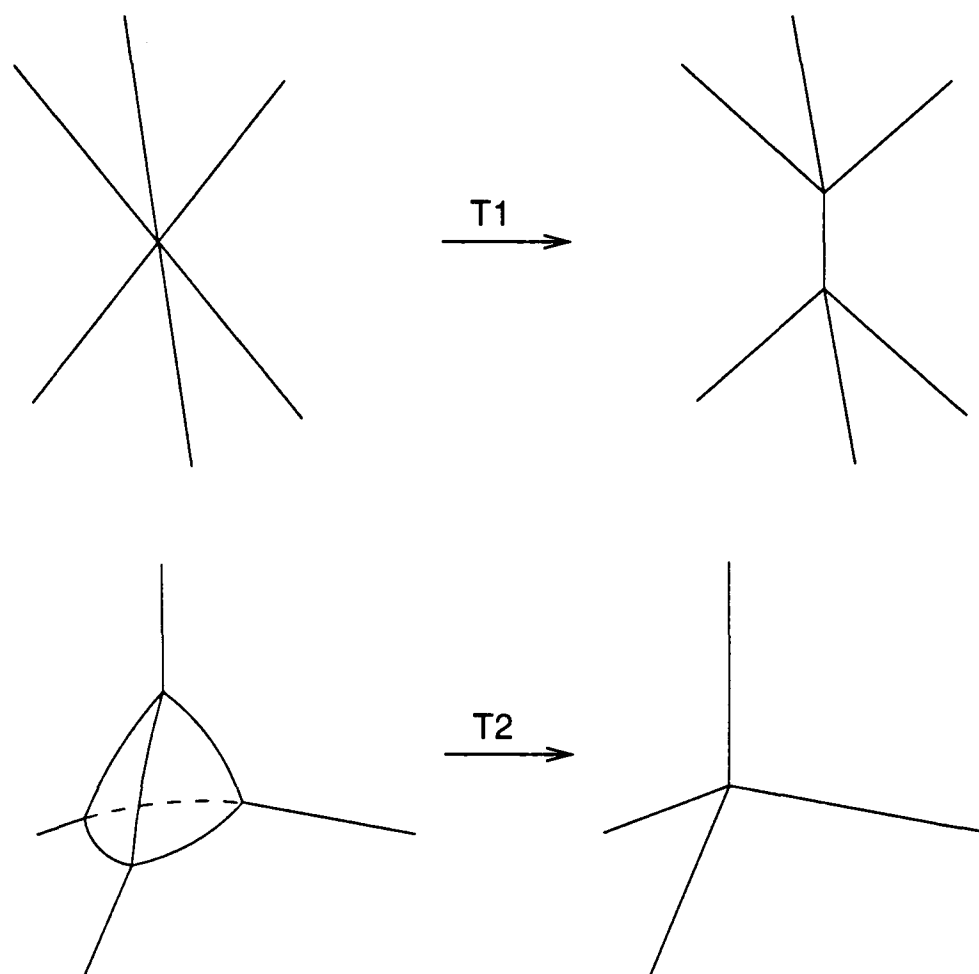


Figure 2.3: Foam topological transformations in three dimensions.

other Plateau borders at local angles of 109.47° , the tetrahedral angle. Each Plateau border is then at the same angle to its neighbours in a junction.

It follows that the $T1$ and $T2$ processes also change to accommodate the extra dimension, as shown in figure 2.3. $T1$ processes occur when two 4-vertices join to briefly become 6-vertices before separating into two different 4-vertices. The simplest case for a $T2$ bubble disappearance happens with a tetrahedral bubble, although as with the two-dimensional case other small bubbles may also vanish in an instant. Unfortunately, von Neumann's growth law has no simple equivalent in three dimensions. DeHoff and Liu [15], working with grains in polycrystals, suggested that the number of faces of a grain was proportional to some measure of the grain diameter. Avron and

Levine [16] applied von Neuman's reasoning to three-dimensional foams but found that there was no minimum number of facets for a bubble above which it would be certain to grow. Sire [17] concluded that

$$V_f^{-1/3} \frac{dV}{dt} = \mathcal{F}(F)$$

where V is the volume of a bubble and F is its number of faces. This function is linear in F and vanishes for $F = 15.7..16.1$, and is true for averages of bubble clusters only. It does not hold for individual bubbles.

Here the awkwardness of the three-dimensional case becomes apparent. For a cellular network containing flat faces, the average number of faces on a cell is 13.7 compared with six edges in the two-dimensional system. Clearly this is an impossible number of sides for a single bubble, and many bubbles end up with twelve or fourteen sides. So why does Sire's function have a zero value for the range $F = 15.7..16.1$? Weaire and Glazier [18] suggest that the $V_f^{-1/3}$ which results from the derivation of the equation skews the mean number of faces upwards.

The general message is that simulation of two-dimensional networks is straightforward, partly because of von Neumann's Law, whereas in three-dimensions a thorough understanding of the mathematics is not yet possible. Also, there is a simple minimal network composed of hexagons which can be used in computational experiments on two-dimensional foams, but an equivalent in three dimensions is harder to find.

In 1988, Kraynik [19] commented,

Current theories cannot provide definitive relationships between foam structure and rheology because they are restricted by two dimensionality in all cases and perfect order in most cases.

Four years later, Glazier and Weaire [2] introduced their review paper with

We will focus primarily on the evolution of two-dimensional rather than three-dimensional cellular patterns because the experiments and theory are more tractable and therefore better understood. On the experimental side, this results from the difficulty of serial sectioning and three-dimensional reconstruction in the direct ‘brute force’ approach to structural analysis.

It is easy to see why the experimentalists concentrated their efforts on two-dimensional research. The study of a single bubble layer nestling between glass plates makes an easily visible and slow-moving target for analysis. In three dimensions, there are problems even getting a foam to stay still for long enough to make an observation, let alone being able to devise a system which can gather enough useful data in that time. Foams do not respond well to invasive techniques, being liable to collapse or rearrange on coming into contact with a probe, and taking visual measurements can be thwarted by the scattering or reflection of light, coherent or not, incident on Plateau borders. Macroscopic measurements of ensemble foam properties can sometimes be found using diffusing-wave spectroscopy [20], but this gives no local geometric data and is only applicable to foams which scatter light at laser wavelengths.

There are problems for the theoreticians and modellers, too. A lamella in a two-dimensional foam is a simple arc, whereas in three-dimensions the total mean radius of curvature of the surface is constant but local curvatures in perpendicular directions may differ. The amount of storage required just to maintain a mesh model of a three-dimensional foam to a reasonable accuracy could run to hundreds of megabytes, even for simple cellular configurations,

and the calculations on the model needed for diffusion or relaxation problems would consume more processing power than is currently available at all but the largest research establishments.

Because of these problems, researchers have concentrated mainly on studying three-dimensional structures containing either only one type of bubble or only a handful of different cells. An account of modelling and measurement efforts in these cases and two-dimensional networks is given in the following sections.

2.2 Foam models

A large proportion of work concerning the drainage of foams has been performed without reference to foam cellular structure. Some, like Leonard and Lemlich [21], Barigou and Davidson [22] or Barber and Hartland [23, 24], have considered liquid flow through a single idealised Plateau border or lamella. Other researchers have simulated a random foam network with a mathematically simpler device, such as Weaire *et al.* [25], Verbist and Weaire [26], Hutzler *et al.* [27] and Verbist *et al.* [28] who assumed all Plateau borders were vertical tubes in their theories on drainage. But the latter is careful to point out that

It can hardly be claimed that the validity of these assumptions and approximations is self-evident, and we cannot yet map out clearly their regimes of reliability.

It is, of course, possible to treat a foam as a more homogeneous material and consider bulk properties with even fewer structural details. Barber and

Hartland [29] have derived a relation for foam collapse time t which depends on material properties such as density ρ , viscosity μ and surface tension σ ,

$$t = \frac{0.55d^{9/5}h_0^{1/5}\mu}{(\rho g)^{1/5}\sigma^{4/5}\delta_c^{7/5}} \left[\left(\frac{h}{h_0} \right)^{-\frac{1}{7}} - \frac{h}{h_0} \right]$$

where d is the bubble diameter, h_0 is the initial foam height and δ_c is the critical film thickness. In order to produce this, the authors assumed that their foams consisted of identical dodecahedral bubbles and that the foam was initially in a steady state where the height was unchanging. Unfortunately, although dodecahedral bubbles are mathematically friendly in geometry and hence surface area calculations, they do not actually tessellate in space (by an error of a few percent) and therefore could not be the only cellular component of a foam.

Also, using an average bubble diameter in this way eliminates the consideration of internal $T1$ and $T2$ changes in the foam which would cause the foam to coarsen. This will affect the collapse time, which depends on a critical lamella thickness, as the overall surface area of the foam will decrease and presumably allow more liquid to thicken the remaining Plateau borders.

Hrma [5] has skirted round the cellular geometry problem by devising a model which relates foam height to gas flux and ‘bubble radius,’ which again is an average or effective quantity,

$$\frac{h}{2r} = \frac{j}{j_m}$$

Here, j_m is the minimum gas flux required for foaming to occur. This model seems to be too simple for most foaming situations, though.

Empirical models such as these can be accurate in a range of conditions, as Barber and Hartland point out, and may be useful in industry if a good

enough fit can be obtained. However, no model is even available which covers all cases for one liquid system and certainly not for all systems in all conditions.

2.2.1 Two-dimensional cellular models

Weaire and Kermode [30] developed a computer simulation of a two-dimensional foam of infinitely low liquid fraction. The authors aimed to study the variation in bubble size as the foam coarsened, and therefore needed to model the diffusion of gas across cell boundaries. Here, von Neumann's law spares the expense of calculating the actual movement of gas molecules into and out of a bubble, because the rate of bubble size change is simply a function of its number of sides.

The authors used a local iterative scheme to save on processing power and memory, where each edge vertex was considered in turn. The positions of the vertices and the pressures of their three neighbouring cells were relaxed until equilibrium for the whole network was reached. During this process, the software had to check continually for possible rearrangements of the structure by *T1* transformations. This topological change would occur if

$$\mathbf{D} \cdot \mathbf{R}_i > \mathbf{R}_i \cdot \mathbf{R}_i$$

where $\mathbf{D}$ is the displacement vector and $\mathbf{R}_i$ is the vector to the i^{th} nearest neighbour vertex which might be involved in the transformation.

After equilibrium had been reached, the program then calculated a diffusion step using von Neumann's law to decide which bubbles had grown or shrunk and by how much. It is during this step that a *T2* process may occur and a small bubble shrinks to zero area. In practice, small bubbles and their

very high pressures caused numerical problems for the simulation and were deleted when they assumed an area of less than a preset small value.

The initial network for the program was generated by randomly adjusting a hexagonal network and forcing a few $T1$ processes, and then removing the ten shortest sides to give a structure containing 100 bubbles.

The overall effect of the program was to simulate foam coarsening apparently quite well. Occasionally, two-sided bubbles were created despite the fact that this could never happen in a ‘real’ two-dimensional foam. It seems that the discrete time steps used by the iterative solver sometimes caused the program to deviate some way from equilibrium and allowed $T1$ processes to occur when they would not normally have done so.

Whereas Weaire and Kermode had deliberately avoided a simultaneous solution due to computing power limitations, Herdtle and Aref [31] used a supercomputer to find a global matrix solution. Defining the problem as a set of equations relating the bubble pressures, positions and liquid film tensions and then using a sparse matrix solver to find a solution enabled the authors to be more sure about the numerical accuracy of their answers. The extra computing power also meant that a network of 1024 bubbles could be used instead of the 100 of the previous model. Unfortunately, Herdtle and Aref did not ever run the same initial conditions as Weaire and Kermode, so it is not known whether the two models would have either deviated or produced the same result.

Bolton and Weaire [32] then extended the simulation so that the liquid fraction in the foam could be varied. Also, they used a more common method among cellular researchers to generate the initial network, by creating a

Voronoi tessellation [33] of a set of random points on a plane. The Voronoi mesh has vertices at the centres of each triangle of nearest neighbouring initial points; it is a property of this network that each vertex joins three edges in the general case. They found that it was impossible to equilibrate the network for gas fractions of less than 0.84, because the Plateau borders were large enough to allow movement of bubbles. At this point the bubbles were circular, as would be expected. Perhaps less obvious is the fact that this arrangement of bubbles corresponds with the packing fraction of random hard discs, as determined by Bideau and Troadec [34]. Is there a similar relationship in three dimensions? It seems that this question will remain unanswered for some time, until three-dimensional models become more realistic. The efforts undertaken so far are documented next.

2.2.2 Three-dimensional models

The search for possible candidates for bubble shapes which would tessellate in three-dimensions and minimise surface area, and hence be of use in foam models, was led by Thomson (Lord Kelvin) [35] back in 1887. In an elegant piece of analysis, Thomson showed how the pattern of hexagonal bubbles in the two-dimensional minimum solution could be created from a grid of squares by a set of $T1$ transformations, and that similarly in three-dimensions a minimal tetrakaidecahedral network could be created by starting with a cubic structure. A single tetrakaidecahedron is shown in figure 2.4. The six-sided facets on Kelvin's cell are slightly curved in orthogonal directions to allow for local tetrahedral angles at the Plateau border junctions, although each surface has zero mean curvature. This curvature means that adjacent

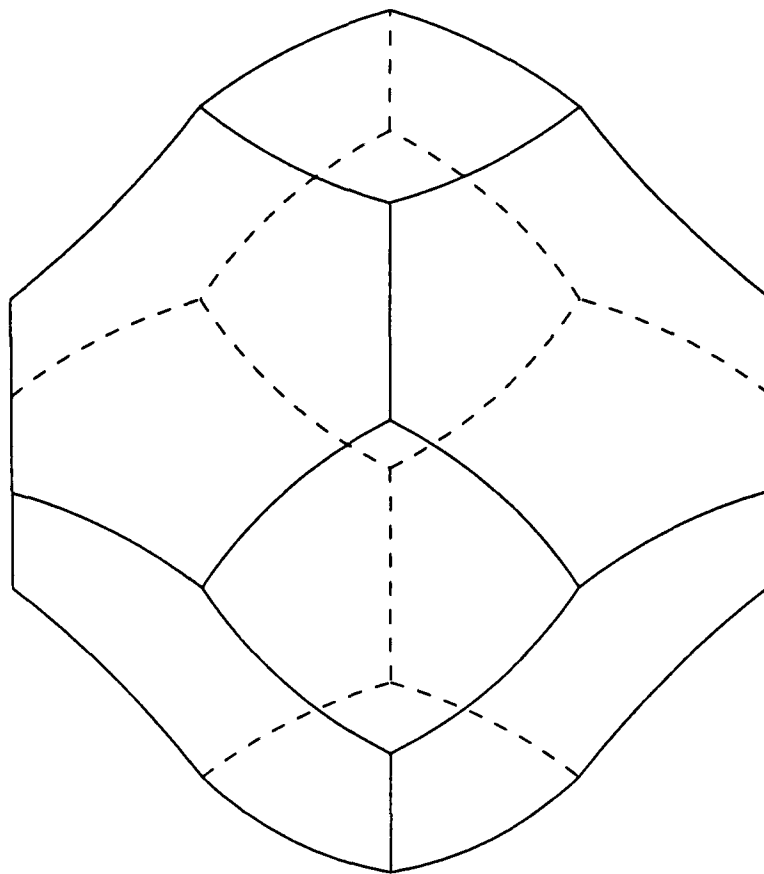


Figure 2.4: Kelvin's minimal tetrakaidecahedron.

tetrakaidecahedra are actually rotated 90° to each other along an axis through their joining hexagonal faces.

Kelvin produced approximate expressions for the curved edges and surfaces on his model, but did not derive an accurate value for the surface area on a cell of unit volume. This was left to Princen and Levinson [36] who showed that Kelvin's cell had a surface area which was 18% smaller than the orthic (planar-faced) tetrakaidecahedron. If the distance between the centre and middle of a side of a hexagon in the orthic cell is taken as one distance unit, the total surface area of the Kelvin cell is 35.6488 and its volume is 17.4186.

Princen and Levinson also calculated the *relative surface area* and the *isoperimetric quotient* of the Kelvin cell and other polyhedra. The relative

Body	S/S_0	IQ
Tetrahedron	1.4900	0.30230
Cube	1.2407	0.52360
Octahedron	1.1826	0.60459
Pentagonal dodecahedron	1.0984	0.75471
Rhombic dodecahedron	1.1053	0.74048
Orthic tetrakaidecahedron	1.0990	0.75337
Kelvin tetrakaidecahedron	1.0970	0.75746
Icosahedron	1.0646	0.82881
Sphere	1.0000	1.0000

Table 2.1: Relative surface area and isoperimetric quotient for several solid shapes [36].

surface area (S/S_0) is the ratio of the surface area to that of a sphere of the same volume. The isoperimetric quotient is the reciprocal of the relative surface area cubed,

$$\frac{S}{S_0} = \left(\frac{1}{IQ}\right)^{1/3}$$

These values are tabulated in table 2.1.

For a very long time, over 100 years, it was thought that the Kelvin cell had the minimal surface area for a space-filling shape, and researchers have used foams containing Kelvin cells in work on foam properties, for example the papers by Reinelt and Kraynik [37, 38, 39]. Although it is true that Kelvin's cell is the minimum single-shape solution found so far, Weaire and Phelan [40] have discovered a composite structure containing two different polyhedral shapes which has a lower surface area than Kelvin's solution by

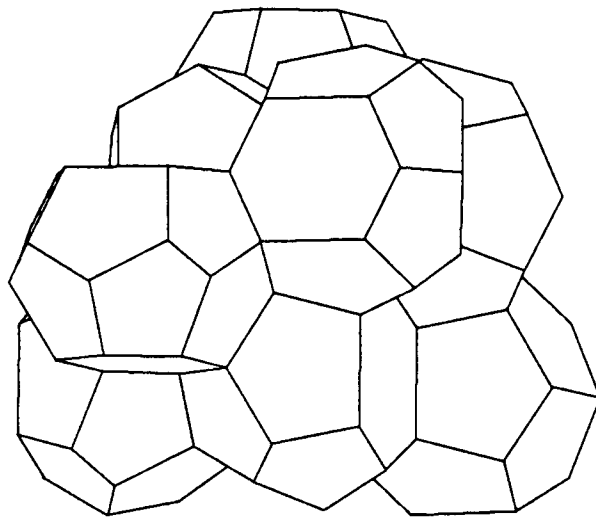


Figure 2.5: Weaire-Phelan minimal cell geometry (actual structure has slightly curved faces) consisting of six 14-sided polyhedra and two regular pentagonal dodecahedra.

about 0.3%. This newer structure has a unit cell composed of two pentagonal dodecahedra and six tetrakaidecahedra consisting of two hexagonal and twelve pentagonal faces; see figure 2.5. Weaire and Phelan discovered this structure by mimicking the arrangement of atoms in the $\text{Ni}_8\text{Si}_4\text{6}$ cubic periodic clathrate solid on computer and then using a surface minimising program called the ‘Surface Evolver,’ written by Kenneth Brakke [41, 42], to produce an optimal cell.¹ The minimal surface formed by the Weaire-Phelan cell is composed of slightly-curved faces due to the pressure differences in the component bubbles. The criterion of equal volumes is preserved, though.

Although the Surface Evolver is very good at minimising shapes of a given starting topology, it does not allow topological changes like the $T1$ and $T2$ transformations which occur in foams, and is therefore not able on its own to perform an exhaustive search for a minimal solution. This is a difficult

¹Kraynik and Reinelt [43, 44] have already reported results of studying the elastic behaviour of foams composed of Weaire-Phelan cells.

problem to solve in three dimensions and there is no known software yet written to cope with a general foam system.

2.3 Experimental work

The study of real foams has proved to be just as difficult as constructing convincing mathematical and computational models. Frothy liquid foams are the most interesting to researchers, but there are no simple ways to capture the structure of a three-dimensional foam in order to study its geometric and topological properties.

Liquid foams scatter light, making many optical observations hard at best and impossible in many cases. Attempts to illuminate foam cross-sections with a laser sheet are thwarted by the light-scattering effect of the foam Plateau borders. This, and the inability of many foams to support even small particles mean that particle image velocimetry methods [45], which are among the commonest ways to visualise liquid flow, are not suitable for this task.²

Foams are generally short-lived phenomena, meaning that techniques which cannot sample the foam structure during a much shorter time scale than that by which the foam is rearranging or moving will be useless. The structure of two-dimensional foams can be recorded by placing the foam vessel on a photocopier or by taking a photograph. In three dimensions, only bubbles next to the foam vessel wall are easily photographable. There has been much debate about whether this is of any value, too.

² Chen and Fan [46] and MacCuaig *et al.* [47] have reported particle velocimetry results on a related system—the fluidised bed.

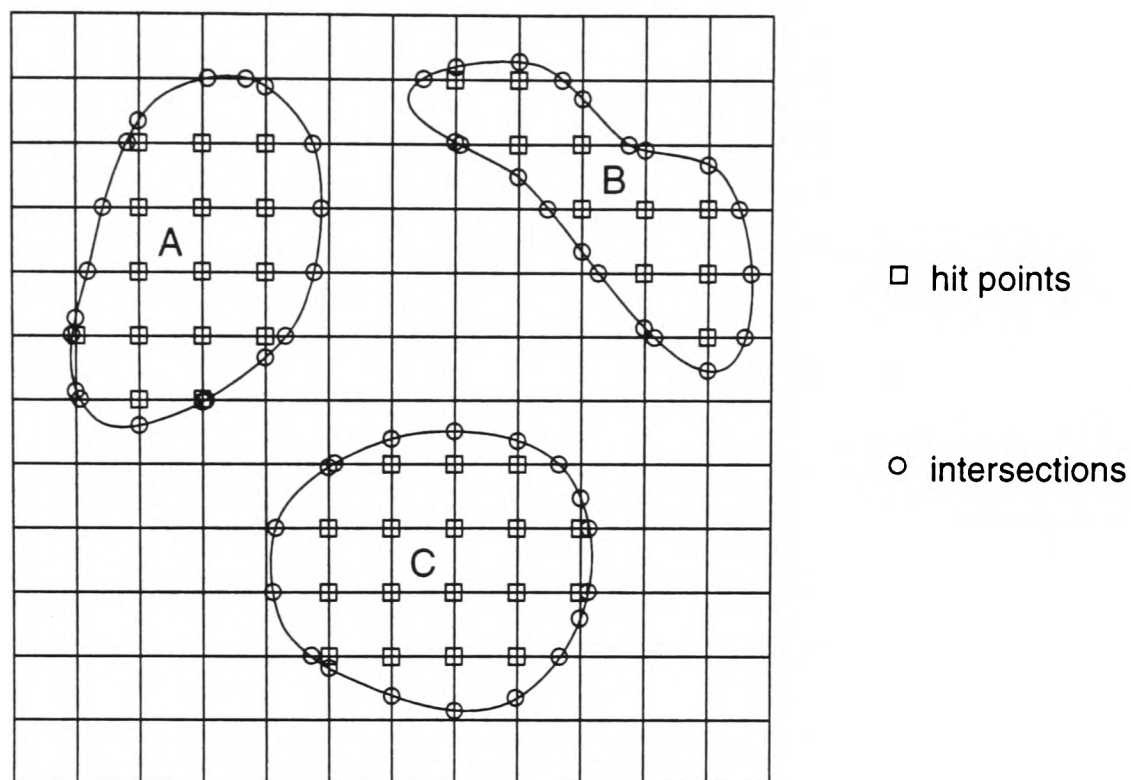


Figure 2.6: Stereology grid points on a two-phase material sample.

Calderbank and Rennie [48] have presented an analysis of the froth formed on sieve-plates and have deduced the bulk foam properties by considering the bubbles in contact with the perspex walls of their container vessel. Using stereological techniques (see below), the researchers related bulk bubble volume to the two-dimensional bubble area next to the vessel walls. However, this work has been criticised [49] by others who believe that wall effects distort the accuracy of this technique.

More recent work on the stereological analysis of foam samples was undertaken by Rhodes and Khaykin [50]. A multi-phase material is shown in figure 2.6 with a superimposed regular grid. By counting the numbers of intersections and enclosed mesh vertices, the stereological parameters in table 2.2 are calculated. The two-phase material in figure 2.6 contains three regions, A, B and C, of one phase, in a substrate of a different phase. The test

P_T	total number of test points	$P_T = m^2$
d	distance between test points	
P_H	total number of hit points	
P_I	total number of intersections	
P_{L1}	total number of horizontal intersections	
P_{L2}	total number of vertical intersections	
V_V	volume density	$V_V = P_H/m^2$
S_V	surface density	$S_V = 2P_I/m^2d$
S/V	specific surface	$S/V = 2P_I/P_Hd$
$\bar{L}_3$	mean intercept length	$\bar{L}_3 = 2P_Hd/P_I$
λ	mean free distance between features	$\lambda = \bar{L}_3(1 - V_V)/V_V$
$\bar{D}$	mean diameter	$\bar{D} = 3P_Hd/P_I$
Q	elongation ratio	$Q = P_{L1}/P_{L2}$
Ω_{12}	degree of orientation	$\Omega_{12} = (P_{L1} - P_{L2})/$ $(P_{L1} + 0.571P_{L2})$

Table 2.2: Stereological parameters for foam characterisation [50].

grid consists of a matrix of 13^2 points, and there are 42 hit points (P_H) and 57 intersections (P_I). If the test points are 0.01m apart, then for example,

$$V_V = P_H/m^2 = 42/13^2 = 0.249$$

$$S_V = 2P_I/m^2d = 2 \times 57/(13^2 \times 0.01) = 67.5m^{-1}$$

$$S/V = 2P_I/P_Hd = 2 \times 57/(42 \times 0.01) = 271m^{-1}$$

Other material properties follow in a similar manner.

For stereological measurements to be successful the following requirements should hold,

- the foam structure should be distributed homogeneously
- the number of measurements must be statistically sufficient

In examining the bubbles near to a container wall, the former criterion may not be satisfied and, if only a limited section of foam can be viewed, nor may the latter.

Matzke [51] had read Kelvin's paper on minimal surfaces, since he had an interest in correlations between foam structure and the shapes of plant cells. Believing that the forces governing the creation and stabilisation of cells and bubbles were similar, he embarked on an extensive experimental program to try and create Kelvin's cells in a monodisperse soap foam.

Taking a small syringe with 0.01ml gradations, Matzke carefully blew individual bubbles of volume 0.1ml by dipping the tip of the syringe needle into the soap solution and expelling the required amount of air. He placed these bubbles one by one into a beaker which could hold about 1900 bubbles in total and left them to drain and settle for a day or more. To prevent excessive drainage and bubble collapse, Matzke left the beaker on a rotating

clinostat which continually inverted the entire foam. This whole procedure was repeated for sixteen beakers' worth of foam. Sadly, the foams created in this way were not likely to stay monodisperse for long—gaseous diffusion between bubbles would have caused volume changes throughout over the long period of time the foams were allowed to settle.

However, Matzke took photographs of the foams through the beaker. He made drawings of bubbles next to the vessel wall (see figure 2.7) and of those in the bulk of the foam a few bubble-widths away from the wall (see figure 2.8). At the periphery of the foam the bubbles had between seven and thirteen faces, whereas in the bulk of the foam the lower and upper numbers were eleven and sixteen respectively. Matzke could see no order or regular structure anywhere in the foams.

Matzke noted that peripheral bubbles averaged a different number of faces to those in the bulk of the foam, 10.99 to 13.7. More than two-thirds of all the bubble faces were pentagonal, and almost all were either quadrilateral, pentagonal or hexagonal. The most common type of bulk foam cell was a 13-faceted bubble with one quadrilateral, ten pentagonal and two hexagonal faces.

Matzke's observations of a possibly polydisperse foam of similarly-sized bubbles give a clue as to why the Weaire-Phelan minimal cell is better than Kelvin's original suggestion. Despite the elegant symmetry of Kelvin's cell, where each vertex and Plateau border is identical, it is unusual because it contains no pentagonal faces. The Weaire-Phelan model requires two different bubble shapes in order to tessellate, but pentagonal faces dominate the structure.

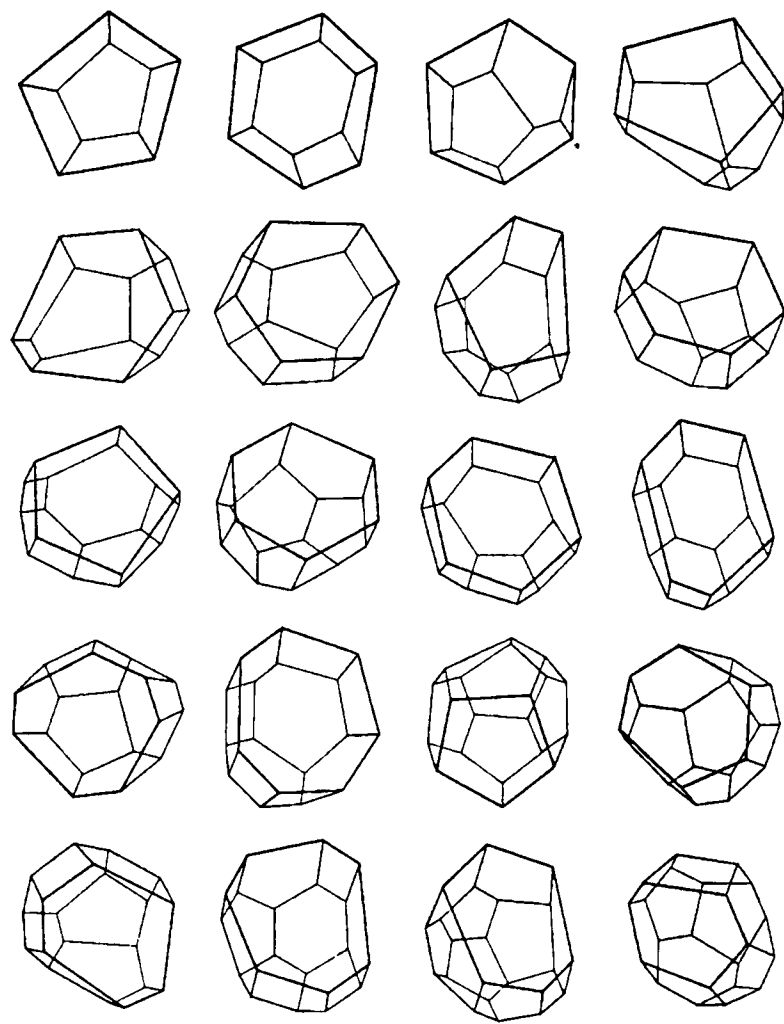


Figure 2.7: Bubbles next to vessel walls in a soap foam, drawn by Matzke.

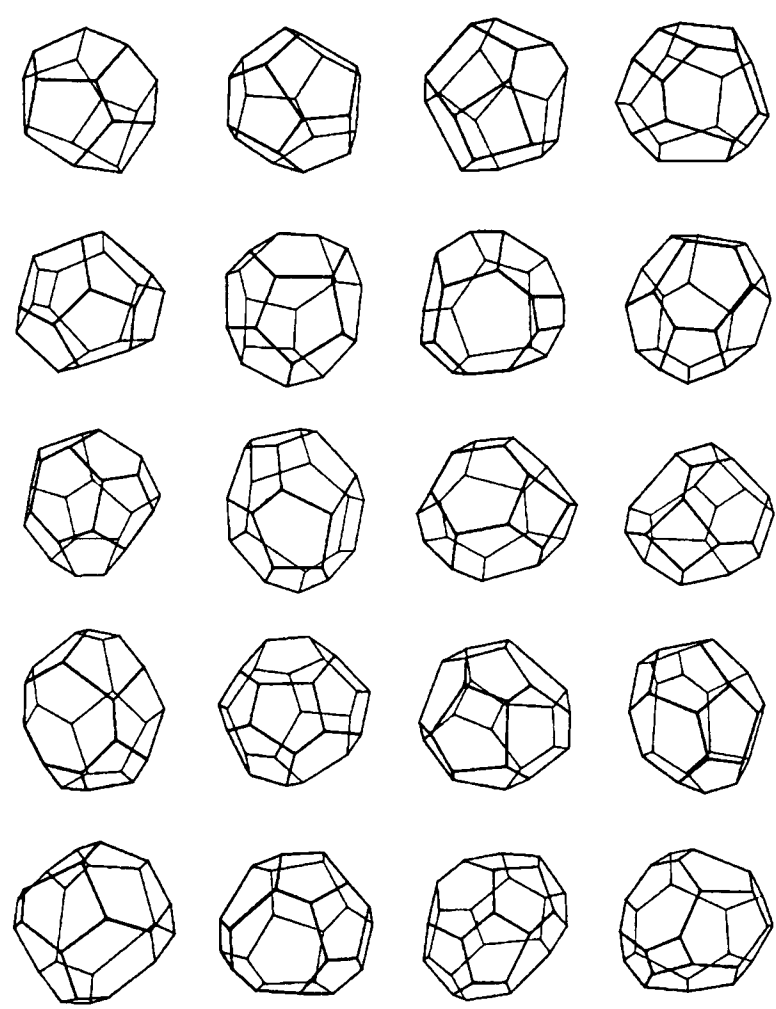


Figure 2.8: Bubbles in the centre of a soap foam, drawn by Matzke.

The problem of accurately recording the three-dimensional structure of foams is now beginning to be addressed. It is possible with care to freeze a liquid foam so that the bubbles may be studied over a much longer time period than otherwise possible. Foot *et al.* [52] examined the structure of the thick stable foams formed on sludge plants by quench-freezing small sections of sludge in liquid nitrogen. These were then freeze-dried and allowed to warm up to room temperature before coating with gold for electron microscope analysis. The results are good, but sludge foams are not very good examples of cellular foam structures, looking rather more like solid foams and therefore not obeying surface area minimisation rules. Unfortunately, a less stable surfactant foam would be unlikely to withstand the trauma of rapid freezing without undergoing some rearrangement.

The development of medical imaging technology, in particular X-ray tomography and magnetic resonance imaging, has led some researchers to try to apply these techniques to foams in the hope of resolving three-dimensional liquid foam structure. Probably the earliest record of a tomographic study of a liquid foam was of that carried out by Morris and Morris [53] in 1965, who used an X-ray photographic method to resolve a series of horizontal slices in a liquid foam, figure 2.9. The rotation of the foam vessel and the photographic plate are synchronised which will cause a two-dimensional image of the foam cross section to be formed on the film. Some blurring occurs due to the presence of liquid above and below the imaging plane, but the results are good for a system developed before digital imaging became available.

Unfortunately, Morris and Morris' images are too blurred to be reprinted here, but it is possible to make out individual bubbles from their photographic

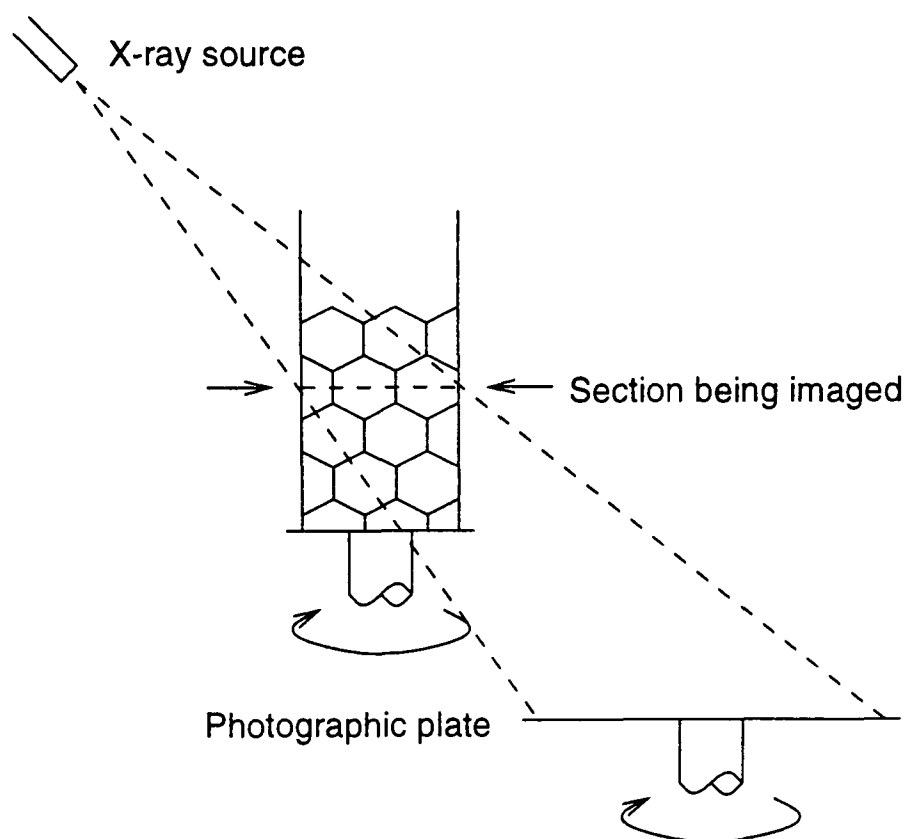


Figure 2.9: Simple X-ray tomography arrangement.

plates and even to measure cell areas. Unfortunately though, the process of moving the foam vertically and changing films meant that capturing enough data for a three-dimensional model was infeasible.

Gonatas *et al.* [54] have completed very recent work using magnetic resonance imaging to resolve foam structure. The cross section through a cylindrical foam sample 1.4cm in diameter can be resolved to an accuracy of $100\mu\text{m}$, which is enough to allow measurement of the areas of 80% of the bubbles in the foam. In a similar piece of work, Prause *et al.* [55] have produced a three-dimensional magnetic resonance image of a foam. After processing, the final result is a voxel model of resolution $128 \times 128 \times 128$ voxels.³ However, it seems that this resolution is not high enough for detailed

³A voxel is the smallest cubic section of material in a three-dimensional model, analogous to a pixel in two dimensions.

analysis of many bubbles in the foam.

2.4 Optical methods for three dimensions

After a brief chapter on the Z Notation, there follow three chapters describing work done on different optical techniques for resolving liquid and solid foam structure—confocal imaging, optical tomography and a hybrid method which requires human assistance.

Chapter 3

A short chapter about the Z Notation

3.1 Introduction

A substantial part of this thesis is concerned with the development of software which will implement various tomographic reconstruction algorithms. A simple aid in the description of this process for researchers who wished to duplicate the work might be the publication of the computer source code for each program, written in the C programming language. However, this would be worthless for any programmers who used another language such as FORTRAN, for example. Even C programmers using a different operating system to the UNIX clones used for this project might have difficulty porting the code.

A much better solution is to describe the software with an abstract mathematical framework, which focusses on what each program does rather than how it does it. Fortunately, *The Z Notation* [56] has been developed at the University of Oxford with the specification of computer systems in mind, and is ideal for describing computer software. The aim of this chapter is



to give a brief introduction to Z, to enable the reader to follow the general descriptions of the software given later. For more detailed information refer to Spivey [56].

3.2 Z fundamentals

Z uses mathematical sets to represent the data in a system specification. An example of a type which is always predefined for use in Z is the set of all integers, $\mathbb{Z}$. A variable z , which is of type integer, is then declared as $z : \mathbb{Z}$. Set operations may be applied to create new set types, such as the set of *natural numbers*,¹ $\mathbb{N}$,

$$\mathbb{N} == \{ n : \mathbb{Z} \mid n \geq 0 \}$$

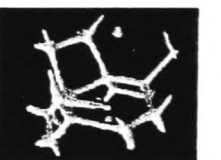
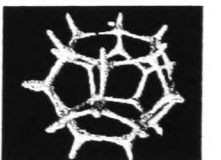
The $\{ \mid \}$ construct is a *set comprehension*; in this case the variable n on the left satisfies the constraint $n \geq 0$ on the right hand side. $\mathbb{N}$ is the set of all possible values of n . The ‘==’ symbol is used for a type definition.

The choice of basic data types for a specification depends on the system being described. Image processing software often deals with the manipulation of pixels which have an 8-bit value, so it will be useful to define the type *PIXEL8* as follows,

$$PIXEL8 == \{ x : \mathbb{N} \mid x < 256 \}$$

In English, *PIXEL8* is defined as the set of all non-negative integers less than 256. Any variable of type *PIXEL8* may have any value in the range 0..255.

¹Non-negative integers.



Compound types are possible in the form of *tuples*. An example of a tuple is a pair, $p : \mathbb{Z} \times \mathbb{Z}$. Legal values for p could be (1,2) or (-3,7), for example. Tuples must have two or more components, and each component may have a different type. An example used later on is

$$VECT == \mathbb{R} \times \mathbb{R} \times \mathbb{R}$$

where $\mathbb{R}$ is the set of real (floating point) numbers. *VECT* describes the set of all real three-dimensional vectors.

Another important type construct is the sequence, *seq*. The following line defines a variable s to be a sequence of five natural numbers,

$$s : \text{seq } \mathbb{N} \mid \#s = 5$$

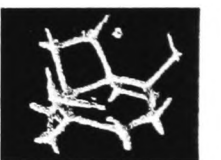
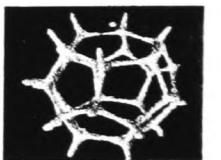
The $\#$ operator gives the number of items in a sequence. The first number in the sequence is referred to as $s(1)$ and the last is $s(5)$. The specification of s is global when presented in this manner.

Z offers a way of compartmentalising the mathematics in a way similar to the system of subroutines and functions in software. A function, *square*, which would return the square of a real number in software, could be described by

$$\text{square} : \mathbb{R} \rightarrow \mathbb{R} \mid \forall r : \mathbb{R} \bullet \text{square}(r) = r * r$$

In English, for all real numbers r , the square of r is r multiplied by r .² The $\mid$ and $\bullet$ are separators, and the function type $\mathbb{R} \rightarrow \mathbb{R}$ means ‘a function which operates on a value of type real and returns a value of type real.’ Such definitions may become quite complex and it is often the case that the

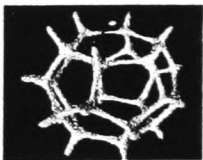
²Arithmetic multiplication is represented by $*$ in Z.



function may not fit on one line. In order to improve legibility in these cases, the definition is normally rewritten vertically as

$square : \mathbb{R} \rightarrow \mathbb{R}$

$\forall r : \mathbb{R} \bullet square(r) = r * r$



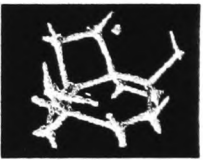
3.3 Schemas

Local definitions in Z are often encapsulated in *schemas* of the form

$$ExampleSchema \hat{=} [D_1; D_2; D_3 \mid P_1; P_2; P_3]$$



where D_1 , D_2 and D_3 are definitions, and P_1 , P_2 and P_3 are predicates involving those definitions. Again, this is normally presented vertically,



$ExampleSchema$

D_1

D_2

D_3

P_1

P_2

P_3



Each line in the vertical schema is part of a conjunction with its neighbours.

There is a mechanism for transferring data between schemas. Any variables decorated with a ‘?’ are designated inputs to a schema and require no initialisation. Variables decorated with a ‘!’ are outputs from a schema. If



two schemas share one or more common variables with the appropriate decorations, the output of one schema can be *piped* to the input of the other. Consider these schemas, *One* and *Two*,

<i>One</i>	
$a! : \mathbb{Z}$	
$a! = 1$	

<i>Two</i>	
$a?, b : \mathbb{Z}$	
$b = a?$	

The expression *One* >> *Two* is the combination of *One* and *Two* joined by a pipe. The value of *b* will be 1, as *a?* assumes the value of *a!* by default.

3.4 Other expressions

To present a set of a given set type, the *power set* operator $\mathbb{P}$ is used, for example a set of integers, $\mathbb{P}\mathbb{Z}$. The following line is mathematically correct,

$$s : \mathbb{P}\mathbb{Z} \mid s = \{1, 2, 3, 4, 5\}$$

The full set comprehension $\{ \mid \bullet \}$ construct allows complex sets to be described, such as the set of all positive even numbers,

$$\{ z : \mathbb{Z} \mid z \geq 0 \bullet z * 2 \}$$

There are other set constructs,

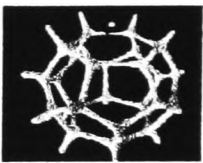
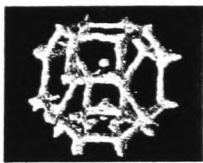
$\emptyset$ the empty set

$\in$ set membership

$\cup$ set union

$\cap$ set intersection

$\setminus$ set difference



There are several predicate expressions possible in Z,

$\neg$ logical negation

$\wedge$ logical conjunction

$\vee$ logical disjunction

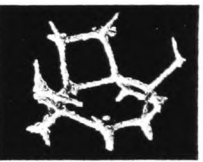
$x \Rightarrow y$ implication; y is true if x is true

$x \Leftrightarrow y$ equivalence; y is true if and only if x is true

$\forall x : X$ for all x of type $X \dots$

$\exists x : X$ There is an x of type $X \dots$

$\exists_1 x : X$ There is only one x of type $X \dots$



Other symbols are documented with a footnote when they first appear in the text.



In general, a Z specification consists of concise mathematical descriptions accompanied by English documentation. This is how the reconstruction software, developed as part of this project, is described in the following chapters.



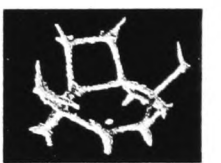
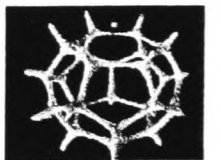
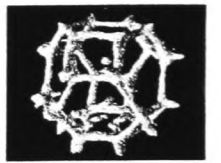
Chapter 4

Focal plane photography

4.1 Introduction

The property of lenses known as *depth of field* provides a method of resolving the three-dimensional structure of a cellular foam network, such as a stable liquid foam of high gas fraction. A photograph taken of a foam shows Plateau borders in sharp focus only at distances from the camera lens within the focal plane. All other Plateau borders are less sharply focussed depending on their distance away from this plane—a dark point object becomes a fainter and larger circle as it moves further out of focus.

By altering the configuration of the lens and aperture, it is possible to change the depth of field, or the range of distances from the camera in which an object is considered in focus. For a large depth of field, the aperture is reduced and the lens moved as far away from the image plane (film or CCD array) as possible. This means that light rays which hit the image plane will be nearly parallel to the axis of the lens, and any change in distance from the camera of an object means that its image changes only slightly. Conversely, if the aperture is widened and the object (and thus also the image plane)



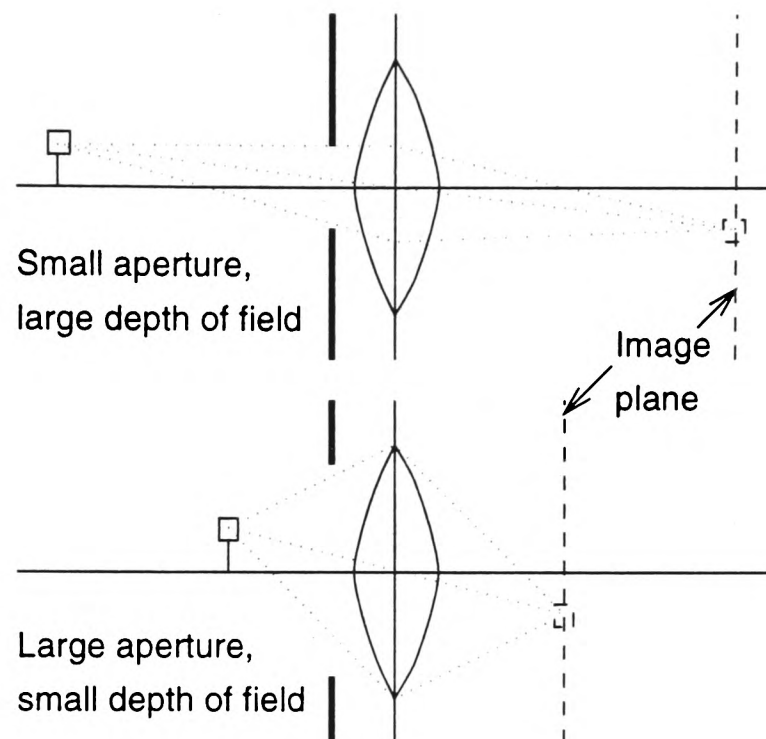


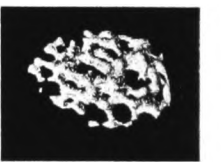
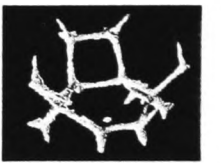
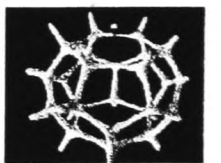
Figure 4.1: Lens configurations for varying depths of field.

brought closer to the lens, the light rays converge at a larger angle to the lens axis, and any movement of object or image plane results in a loss of focus and a small depth of field. This is shown in figure 4.1.

4.2 Experimental apparatus

A system is devised which uses a lens with a small depth of field to attempt to resolve the structure of an aqueous cellular foam. This consists of a transparent perspex box containing foam and a Nikon F camera mounted on a movable platform.

The foam is formed from an aqueous solution of domestic detergent, presumably essentially Teepol—an ionic surfactant composed mainly of alkyl sodium sulphates. A container consisting of two perspex sheets, 6mm thick and held apart by 50mm spacers, has a brass tube of diameter 15mm running



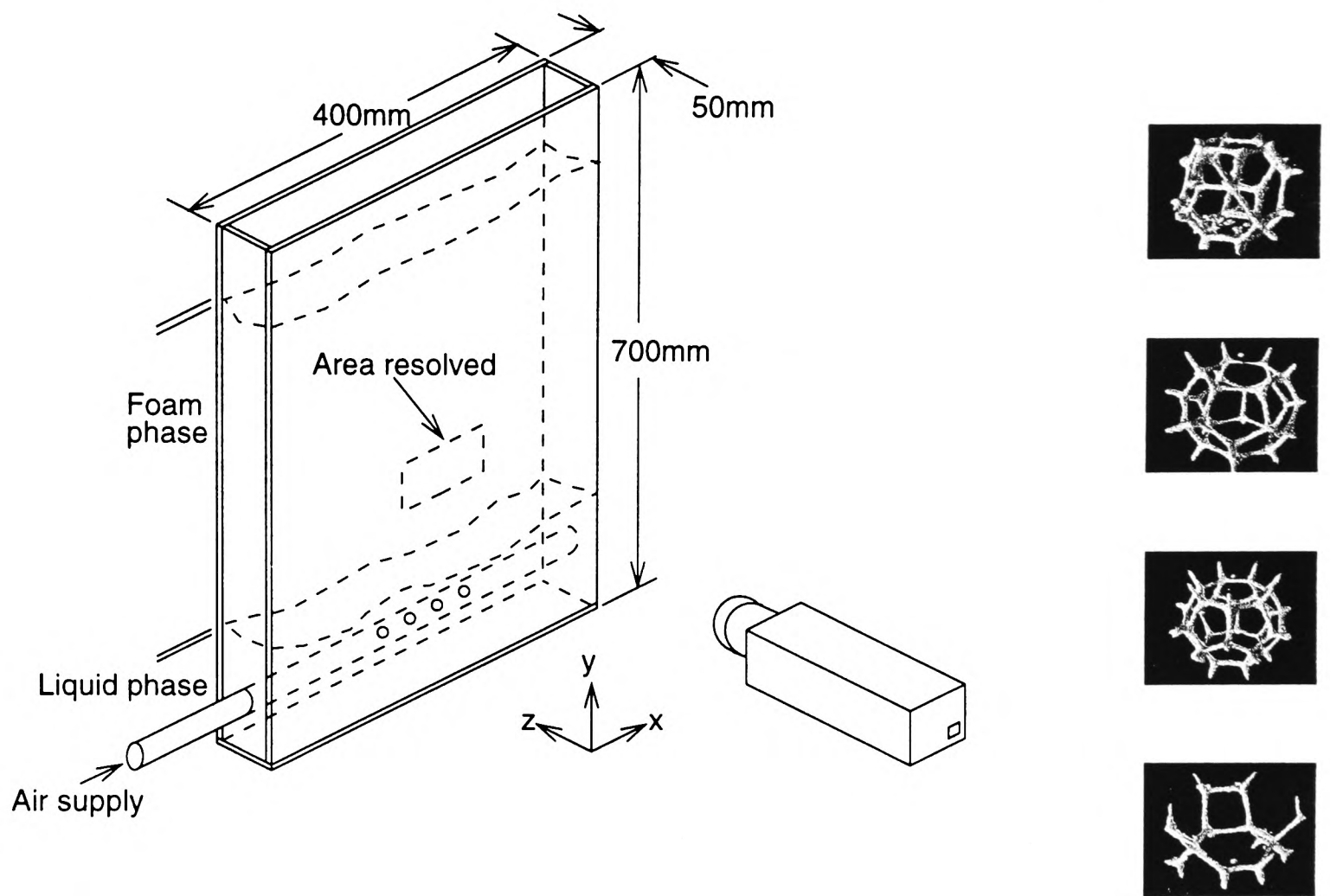


Figure 4.2: The experimental foam rig.

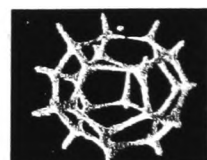
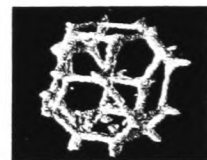
from wall to wall lengthways with a series of small holes drilled through it. The box is filled with enough liquid to cover the tube, which then has air blown through it to generate the foam. Depending on the air flow rate and the hole diameters, foams with various bubble sizes and liquid hold-ups can be created. However, for a foam to be stable enough to photograph it needs small bubbles and has to be left to stand for a few minutes so that most of the liquid can drain away. The box is tall enough that even though foam at the top may collapse, there is a section of stable foam at the bottom which can be photographed—see figure 4.2. The smallest depth of field available was obtained with a set of Nikon 185mm bellows and a Nikkor 135mm lens. The camera is mounted on a specially-made platform which can be moved

towards the foam in 0.25mm increments using a screw-threaded device with a pitch of 1mm, operated by hand. Hence quarter turns yield a 0.25mm movement provided the play in the screw thread is taken up.

The foams tend to remain stable long enough for one reel of film to be taken. For this camera, that means a maximum of 36 exposures covering a depth through the foam of 8.75mm. Lighting is arranged carefully so that specular reflections in the foam are reduced as much as possible by means of a diffuser. As with all image processing problems, it is worth spending time to get the image quality as good as possible in order to save much effort in the processing stages later on. Care was taken to ensure a reasonably uniform background illumination was provided, although small variations in intensity can be corrected easily by image processing. This enables an aperture of f3.5, the largest available, to be used with a $\frac{1}{60}$ second exposure time.

The camera has a field of view which covers about 20mm in the horizontal direction at the focal plane. The foams used tend to have bubbles of diameter about 6–8mm, hence several fit into each frame.

After shooting, the film is developed and each picture converted to digital form by scanning the prints with an A4 flatbed scanner. This gives a sequence of images having 16 grey levels and an aspect ratio of roughly 5:3. To reduce noise in each image, the prints were scanned at 300 dots per inch and then compressed to 75 dots per inch by averaging over 16 pixel squares. This gives images with 256 grey levels at $\frac{1}{16}$ th the original area. The extra information which could be obtained by scanning at a higher resolution is small and offset by the extra storage and processing requirements needed for larger images. Moreover, photographic prints in general are noisy, grainy and exhibit poor



contrast. The use of images at greater than 75 dots per inch is questionable.

4.3 Image processing

There are several problems with this system which must be corrected by digital image processing before any reconstruction of a computer model representing the foam Plateau border geometry. Firstly, the developing process is not entirely regular due to inconsistencies in the photographic dyes, and this results in differing average grey levels between each picture. Since each image consists mainly of a light background hue, with only small sections of foam shown as dark regions, it is reasonable to simply normalise each image by directly modifying the values of individual pixels by the same amount in order to produce a uniform average image intensity. In practice, the changes are relatively small and do not cause pixel values to either over- or underflow.

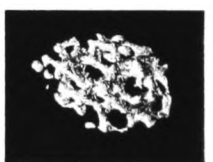
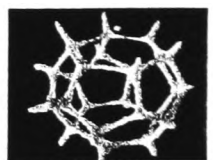
The first step in describing this and the other image processing algorithms with the Z specification language is to define the basic data types upon which the algorithms operate. The set of eight-bit (one byte) pixels is represented by *PIXEL8*, and hence an image raster is an ordered sequence of columns, which in turn are ordered (equal length) sequences of pixels,

$$PIXEL8 == BYTE == \{x : \mathbb{N} \mid x < 256\}$$

$$RASTER == \{r : \text{seq seq } PIXEL8 \mid \forall n : \text{dom } r \bullet \exists_1 l : \mathbb{N}_1 \bullet \#r(n) = l\}^1$$

since all the images used are 8-bit-deep greyscale, at least during the image processing steps. $\#r$ is the length of the sequence of columns in r , and

¹ $\mathbb{N}_1 == \mathbb{N} \setminus \{0\}$.



$\#r(n)$ is the number of pixels in the n^{th} column of r .² Since an image must be rectangular, each column contains the same non-zero number of pixels.

The origin of a raster is taken to be in the upper left corner, so the pixel $p : \text{PIXEL}$ at the top left of raster $r : \text{RASTER}$ is referenced by

$$p = r(1)(1)$$

where the first index is the x-coordinate and the second is the y-coordinate.³

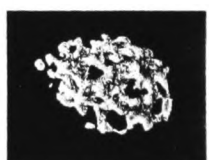
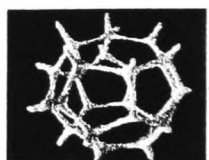
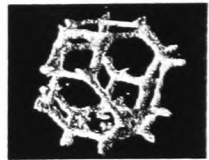
Several operations on rasters occur frequently in this specification, so they are defined here as a convenient shorthand, a small raster toolkit. Firstly, $\square$ is defined as a function on an image giving an integer tuple of its width and height,

$$\begin{array}{|l} \square : \text{RASTER} \rightarrow \mathbb{N}_1 \times \mathbb{N}_1 \\ \hline \forall r : \text{RASTER} \bullet \square(r) = (\#r, \#r(1)) \end{array}$$

Secondly, the specification is simplified if there are operators which are relations on the dimensions of rasters. If an image $r_1 : \text{RASTER}$ is smaller than another, $r_2 : \text{RASTER}$, in both dimensions, then the pair satisfy $r_1 <_{\square} r_2$. The $\leq_{\square}$ and $=_{\square}$ relations are similarly defined according to the dimensions of their arguments,

²In Z, $\#$ is the number of elements in a set; but as $\text{seq } X == \{1 \mapsto x_1, \dots, n \mapsto x_n\}$, $\#X$ is also the number of elements in the sequence.

³In the Z language, sequences start at index 1, although in a C program implementation arrays start at index 0.



$$- <_{\square} -, - \leq_{\square} -, - =_{\square} - : RASTER \leftrightarrow RASTER$$

$$\forall r, s : RASTER \bullet$$

$$r <_{\square} s \Leftrightarrow \#r < \#s \wedge \#r(1) < \#s(1)$$

$$r \leq_{\square} s \Leftrightarrow \#r \leq \#s \wedge \#r(1) \leq \#s(1)$$

$$r =_{\square} s \Leftrightarrow \#r = \#s \wedge \#r(1) = \#s(1)$$

This may seem slightly odd, because two images are considered ‘equal’ if and only if their dimensions are the same, regardless of the values of their component pixels. However, this specification of equality is much more useful than a simpler definition $r = s$, which would mean that $\square(r) = \square(s)$ and the pixel values of r and s were identical.

The global variable *AverageGreyLevel* : *BYTE* is defined to be the required mean intensity value of each image in the dataset. The algorithm for this image equalisation function can be described by,

$$EqualiseGreymap : RASTER \rightarrow RASTER$$

$$\forall r, r' : RASTER \bullet r' = EqualiseGreymap(r) \Rightarrow$$

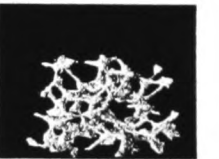
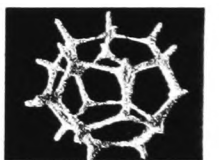
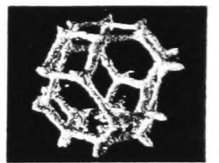
$$r' =_{\square} r$$

$$\forall x : \text{dom } r; y : \text{dom } r(1) \bullet$$

$$\exists c : PIXEL8 \bullet c = r'(x)(y) - r(x)(y)$$

$$AverageGreyLevel = \frac{\sum_{x=1}^{\#r'} \sum_{y=1}^{\#r'(1)} r'(x)(y)}{\#r' * \#r'(1)}$$

There is a small problem with arithmetic overflow in the definition of this function, but this is ignored as the result is assumed to be within the correct 0..255 range. This is handled properly in the implementation. The correction factor, c , can be added to every pixel in raster r to produce an image with a



normalised greymap.

Another problem is that each photographic print is not aligned correctly in the focal plane. The developer of the prints was asked to ensure that each print was orientated the same way, with no rotational error, but there is a translational error caused by inconsistent trimming of the prints. This is solved by cross-correlating images with their immediate neighbours and looking for a position with minimal error in intensity correlation. This trial-and-error process, where the computer searches through all possible displacements of an image with its neighbour within a range of about 20 pixels along perpendicular axes in the images, is the most intensive preprocessing stage.

The correlation function is just a two-dimensional difference operator between images. The minimum absolute value of the correlation between two images gives the correct offset of one image to the next. For completeness, the absolute value function on integers is,

$abs : \mathbb{Z} \rightarrow \mathbb{N}$

$\forall n : \mathbb{Z} \bullet$

if $n < 0$ then $abs(n) = -n$

else $abs(n) = n$

The correlation function on two rasters, each not necessarily square but the first having smaller dimensions than the second, takes four arguments: two rasters and a horizontal and vertical offset in pixels. This returns a normalised integer representing the total difference per pixel,



$$\text{correlate} : \text{RASTER} \times \text{RASTER} \times \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$$

$$\forall a, b : \text{RASTER}; r, i, j : \mathbb{N} \mid a \leq_{\square} b \bullet$$

$$r = \text{correlate}(a, b, i, j) \Rightarrow$$

$$r = \frac{\sum_{x=1}^{\#a} \sum_{y=1}^{\#a(1)} \text{abs}(a(x)(y) - b(x+i)(y+j))}{\#a * \#a(1)}$$

What happens when a doesn't completely overlap b is not specified. In the implementation it is assumed that checks are performed by the calling routine to prevent an error occurring here. Before that schema is presented, one more function is necessary. The first image in the sequence is cropped slightly around all edges to allow some room for the correlation process. All subsequent images are similarly cropped after the best fit is found. *Crop* takes five arguments, an image and coordinates of the cropping region: minimum x, minimum y, maximum x and maximum y,

$$\text{crop} : \text{RASTER} \times \mathbb{N}_1 \times \mathbb{N}_1 \times \mathbb{N}_1 \times \mathbb{N}_1 \rightarrow \text{RASTER}$$

$$\forall r, r' : \text{RASTER}; x_{\min}, y_{\min}, x_{\max}, y_{\max} : \mathbb{N}_1 \mid x_{\min} < x_{\max} < \#r$$

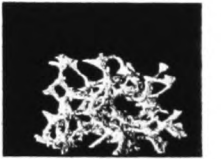
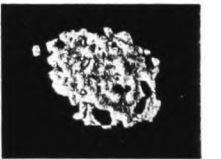
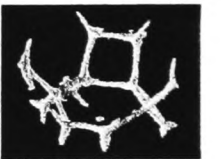
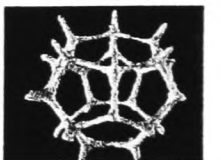
$$y_{\min} < y_{\max} < \#r(1) \bullet$$

$$\forall x, y : \mathbb{N}_1 \mid x \leq x_{\max} - x_{\min} \wedge y \leq y_{\max} - y_{\min} \bullet$$

$$r' = \text{crop}(r, x_{\min}, y_{\min}, x_{\max}, y_{\max}) \Rightarrow$$

$$r'(x)(y) = r(x + x_{\min})(y + y_{\min})$$

A full definition of the algorithm required for the complete data pre-processing stage is now possible. Here, the sequence of photographic images is normalised with respect to intensity and then cross-correlated and clipped to produce a suitable dataset for image analysis.



ImagePreprocess

dataset?, *dataset*, *dataset!* : seq *RASTER*

$$dataset!(1) = crop(EqualiseGreymap(dataset?(1)), X_1?, Y_1?, X_2?, Y_2?)$$

$$\forall n : \text{dom } dataset? \setminus \{1\} \bullet$$

$$dataset(n) = EqualiseGreymap(dataset?(n))$$

$$\exists x, y : \mathbb{Z} \bullet \forall i, j : \mathbb{Z} \bullet$$

$$correlate(crop(dataset(n-1), X_1?, Y_1?, X_2?, Y_2?),$$

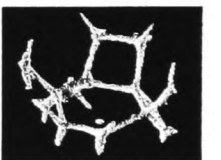
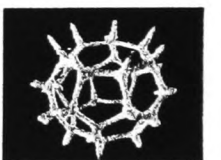
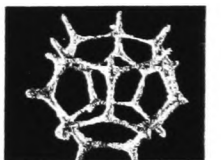
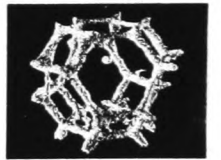
$$dataset(n), x, y)$$

$$\leq correlate(crop(dataset(n-1), X_1?, Y_1?, X_2?, Y_2?),$$

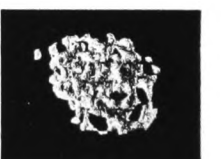
$$dataset(n), i, j)$$

$$dataset!(n) = crop(dataset(n), X_1? + x, Y_1? + y,$$

$$X_2? + x, Y_2? + y)$$



The edges are cropped when all images match to give a set of correctly registered images, such that the centre of each is aligned on the axis of the lens. The values $X_1?, \dots, Y_2?$ depend on the size of the photographic prints and the scanning resolution. For a typical dataset, $X_2? - X_1? = 500$ and $Y_2? - Y_1? = 300$.



This tends to work well in practice, although there are certain pathological images where it will fail. This is due to the problem of the expanding motion field caused by the camera moving towards the foam. Instead of parts of the foam just becoming more or less distinct as they move through the focal plane, they also move slightly outwards as they come closer to the lens. This is not a problem for objects in focus all over the image, because the best correlation will be found when the smaller image fits over the larger one

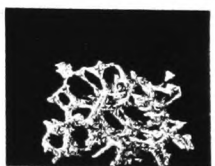
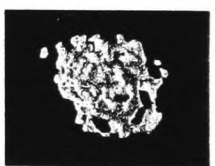
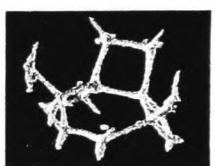
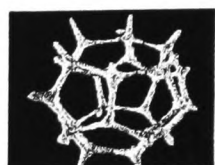


symmetrically. However, suppose a small section of foam occupies one corner of two images with nothing else in the frame. The cross-correlation function will match the images without considering that one has been displaced away from the centre by an enlargement transformation and not by translation. Hence the resulting match will move the second image further away from the image centre than it should be. In practice this is not a problem, mainly because the enlargement of each frame relative to its predecessor is small and also because the foam tends to occupy most parts of the frame and is rarely confined to just one half or quadrant.

The cross correlation operation also solves another problem caused by the length of time required to take 36 exposures. Once the foam is formed, the air supply is cut off but this does not prevent liquid seeping back into the air delivery tube. This results in a steady decrease in the liquid level in the box, and hence the foam also starts to sink. The correlation function corrects this by moving the clipping window downwards with the moving foam. Whilst this reduces the size of the final images in the vertical direction, it does prevent the steady descent of the foam from affecting the results.

4.4 Image analysis

Each registered image typically contains areas of blurred, defocussed foam Plateau borders and also thin, dark line segments of Plateau borders which lie in the plane of focus of the camera. A line detector can be used to enhance the Plateau borders in focus and to reject the defocussed background so that thresholding the result will give a binary line image representing the cellular structure at the focal plane of the original photograph.



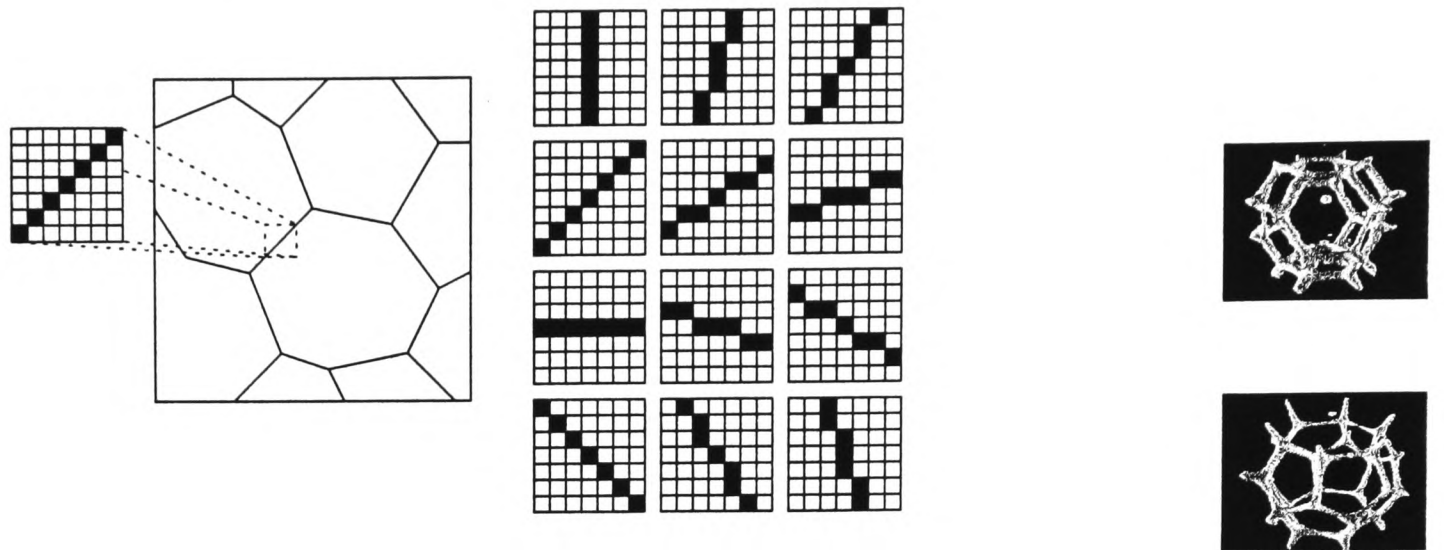
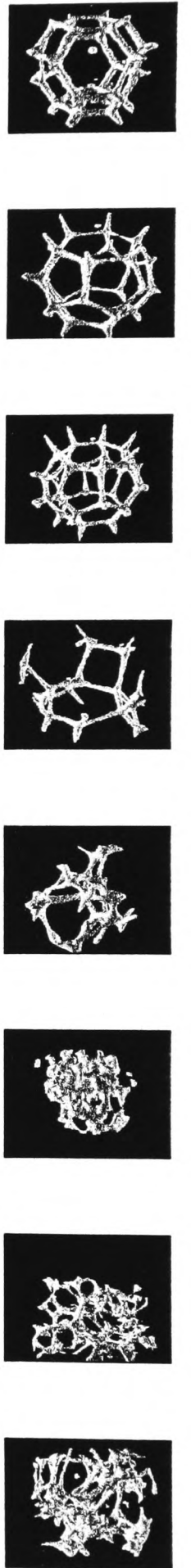


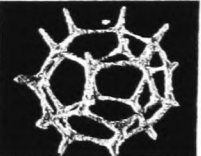
Figure 4.3: 7×7 convolution kernels for edge detection.

The best algorithm for detecting lines in an image is to use multiple two-dimensional convolutions of the image with a selection of 7×7 -pixel bitmaps containing lines in all possible orientations, as shown in figure 4.3. The raster produced as a result of each convolution has high values where the smaller line kernel matches with a line in the image being tested. By taking maximal values for each pixel out of the rasters produced for each convolution operation a complete line image can be created. It is then possible to threshold this image using Canny's upper and lower hysteresis threshold bounds [57] to prevent the 'streaking' effect of just using a single threshold value.

Convolution is defined here as a function which takes two rasters and returns a raster of the same dimensions as the second argument. Each raster may not necessarily be square but the first raster must be smaller than the second in both dimensions. Initially, a function which gives the value of the pixel in the i^{th} column and j^{th} row of the result is defined,⁴

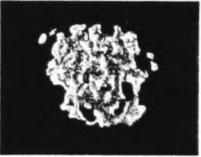
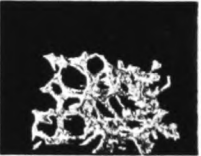
⁴Div is the integer division operator.



$conv : RASTER \times RASTER \times \mathbb{N}_1 \times \mathbb{N}_1 \rightarrow PIXEL8$	
$\forall a, b, r : RASTER; i, j : \mathbb{N}_1 \mid a <_{\square} b \wedge r =_{\square} b \bullet$	
if $(\#a \text{ div } 2 \leq i \leq \#b - \#a \text{ div } 2) \wedge$ $(\#a(1) \text{ div } 2 \leq j \leq \#b(1) - \#a(1) \text{ div } 2)$	
then $r(i)(j) = conv(a, b, i, j) \Rightarrow$ $r(i)(j) = \frac{\sum_{x=1}^{\#a} \sum_{y=1}^{\#a(1)} a(x)(y) * b(i+x-\#a \text{ div } 2)(j+y-\#a(1) \text{ div } 2)}{\#a * \#a(1)}$	
else $r(i)(j) = conv(a, b, i, j) \Rightarrow r(i)(j) = 0$	

It is easily seen from this definition that if the a does not entirely overlap b , no convolution is attempted and a zero-valued pixel is returned for that (i, j) . Although this would be unnecessary if periodic boundary conditions were used, there is no useful meaning to those on non-periodic images. It is better to leave a blank border around the resulting image to show the area where no information is available.

The set of twelve convolution kernels for line detection in the image is specified by the schema,

$LineKernels$	
$kernels : \mathbb{P} RASTER$	
$\forall r : kernels \bullet \square(r) = (7, 7)$	
$\#kernels = 12$	

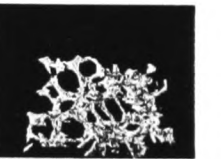
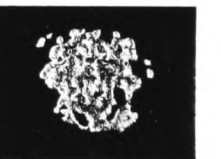
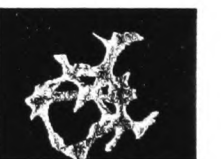
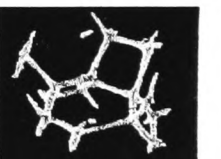
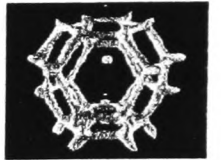
These form part of the specification for the first phase of the line-detecting algorithm. Initially, each foam image is convolved with the set of line kernels and the maximum response at each pixel is taken to form a line-enhanced

image. This gives the best chance of detecting faint lines in each image reliably, and is described by,⁵

$$\begin{array}{l}
 \text{EdgeEnhance} \\
 \hline
 \exists \text{LineKernels} \\
 \text{dataset?}, \text{dataset!} : \text{seq RASTER} \\
 \hline
 \forall n : \text{dom dataset?}; x : \text{dom dataset?}(n); y : \text{dom dataset?}(n)(1) \bullet \\
 \quad \text{dataset!}(n)(x)(y) = \\
 \quad \quad \max\{c : \text{kernels} \bullet \text{conv}(c, \text{dataset?}(n), x, y)\}
 \end{array}$$

The output of this schema, *dataset!* is a sequence of continuous tone (that is, 256 grey levels rather than a binary image) line-enhanced images. These must be *segmented* to be any use to a reconstruction algorithm; a binary image is required which should optimally contain only line information. To achieve this, Canny [57] suggests a hysteresis thresholding method which uses two threshold levels, *upper?* and *lower?*. All pixels in a raster having a value above *upper?* are immediately copied to the final image, while all neighbouring pixels to these and their neighbours are copied provided they have a value greater than *lower?*. This tends to prevent ‘streaking’, where noisy lines are only half detected resulting in a dotted line on the output image. The simplest programming solution involves using an algorithm which traverses an image iteratively, outputting pixels in each step which are in the immediate neighbourhood of pixels output in the previous traversal. This would be inefficient to describe in Z at a low level of abstraction. Instead, the problem can be formulated of in terms of outputting every pixel which is

⁵The $\exists$ line includes the variables and predicates of *LineKernels* in *EdgeEnhance*.



connected to a pixel of value greater than *upper?* along a path composed of pixels of values greater than *lower?*. A description of the *PATH* datatype is

$$PATH == \text{seq}(\mathbb{N}_1 \times \mathbb{N}_1)$$

$$\forall p : PATH; n : \text{dom } p \setminus \{1\} \bullet$$

$$abs(first\ p(n) - first\ p(n-1)) < 2$$

$$abs(second\ p(n) - second\ p(n-1)) < 2$$

Pixels along a $p : PATH$ are therefore eight-connected—they can be linked to any of their nearest neighbours, including the diagonal ones. Given this, the segmentation step is described by

$$ImageSegment$$

$$dataset?, dataset! : \text{seq } RASTER$$

$$upper?, lower? : \mathbb{N}$$

$$\forall n : \text{dom } dataset?; x : \text{dom } dataset?(n); y : \text{dom } dataset?(n)(1) \bullet$$

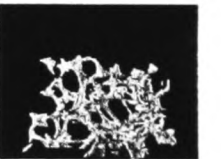
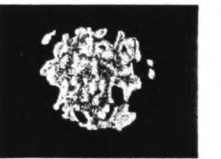
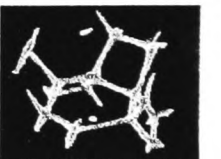
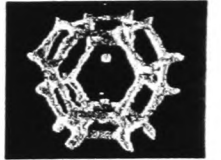
$$\text{if } \exists p : PATH \bullet$$

$$(x, y) \in p$$

$$\forall lpair : \text{ran } p \bullet dataset?(n)(first\ lpair)(second\ lpair) > lower?$$

$$\exists upair : \text{ran } p \bullet dataset?(n)(first\ upair)(second\ upair) > upper?$$

$$\text{then } dataset!(n)(x)(y) = 1$$

$$\text{else } dataset!(n)(x)(y) = 0$$


All the components of the iterative image processing system have now been described. Because each schema uses the variable *dataset?* as an input and

dataset! as an output, and because the types of these are identical, a full description is just each step connected by *pipes*,

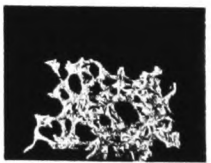
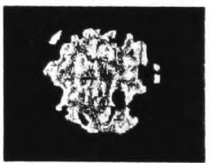
```
ImagePipeline
ImagePreprocess >> EdgeEnhance >> ImageSegment
```

The final result is a sequence of binary images where line pixels in each image have value one and all other pixels have value zero. Ideally, the lines in each image represent parts of the foam cellular structure that are in focus at that camera position. Figure 4.4 shows a typical example of segmentation of an image of an aqueous detergent foam.

4.5 Reconstruction and Visualisation

An efficient computer model for the Plateau border network in a dry foam is simply a list of zero-thickness lines in space. A model close to this, which can be generated from the image data, is a set of lines in space where each line vertex must lie in the plane of one of the images. Vertex z-coordinates are quantised into 36 positions, one for each of the original photographs, 0.25mm apart. There is clearly an inhomogeneity in the spatial resolution of a vertex position, as the resolution in the plane of an image is roughly 0.04mm in horizontal and vertical directions.

However, it is reasonably straightforward to generate this type of computational model by the morphological operation of skeletonisation on each binary image followed by tracing the remaining edges to detect short line segments between vertices. The skeletonisation step is necessary because an efficient line-tracing algorithm needs lines of single-pixel thickness which will



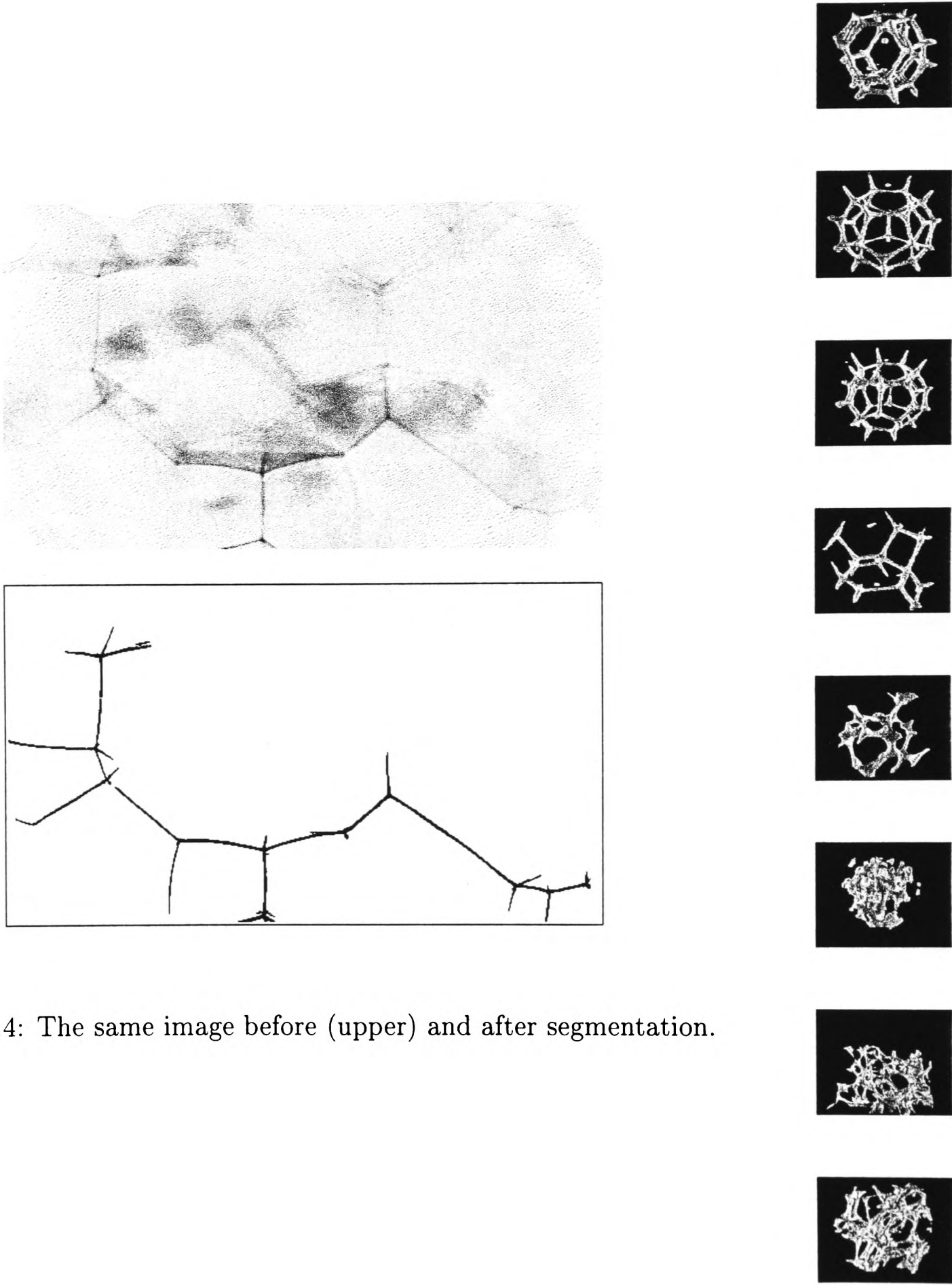


Figure 4.4: The same image before (upper) and after segmentation.

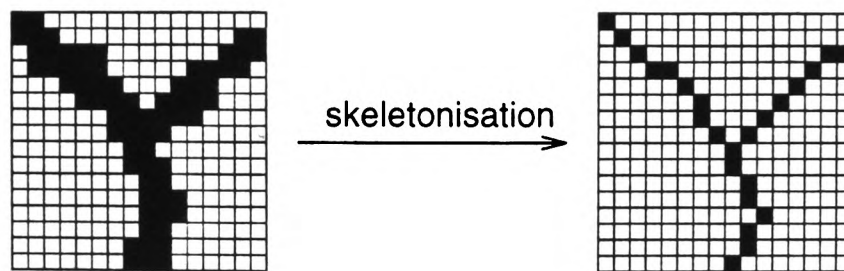


Figure 4.5: The skeletonisation operator.

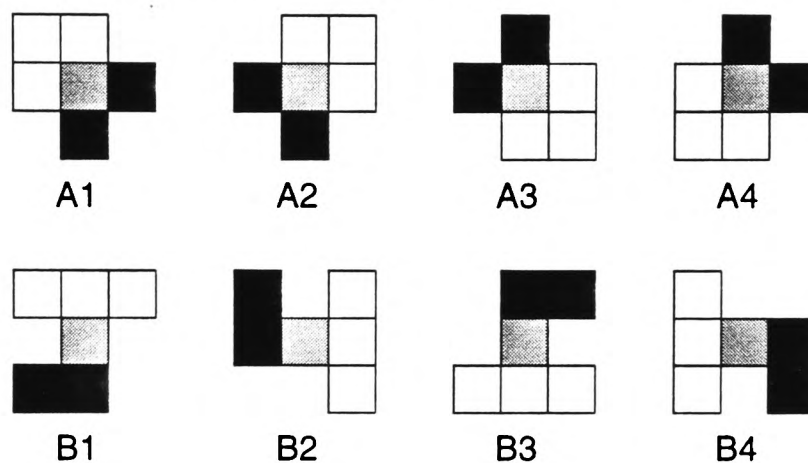
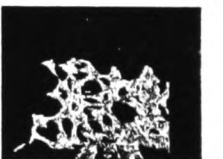
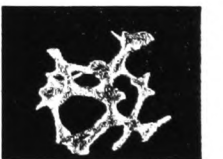
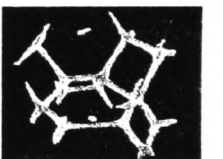
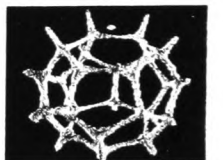


Figure 4.6: Configurations where the central pixel is deleted during the skeletonisation process.

not necessarily result from the image processing operations. Figure 4.5 shows the operation of the skeletonisation algorithm.

The skeletonisation algorithm used is by Arcelli *et al.* [58], and it produces images containing lines where pixels are connected to only two neighbours, except for pixels at junctions where lines join, which have three neighbours, and end pixels which only have one neighbour. Figure 4.6 shows pixel configurations which must occur for the central pixel to be removed. These are tested for in order, A1, B1, A2, B2, A3, B3, A4 and B4, and this results in proper thinning of the image.



$$thin : RASTER \times RASTER \rightarrow RASTER$$

$$\forall a, b, r : RASTER \bullet thin(a, b) = r \Rightarrow$$

$$\forall x : \text{dom } b; y : \text{dom } b(1); dx, dy : \{1, 2, 3\} \mid$$

$$\square(a) = (3, 3) \wedge r =_{\square} b \bullet$$

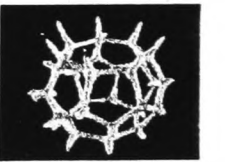
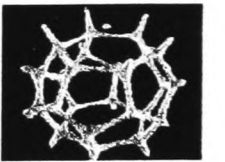
$$\text{if } 2 < x < \# \text{dom } b - 1 \wedge 2 < y < \# \text{dom } b(1) - 1$$

$$\text{then}$$

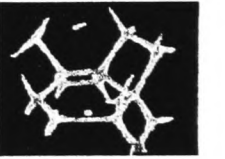
$$\text{if } a(dx)(dy) = b(x + dx - 2)(y + dy - 2) \vee a(dx)(dy) = 2$$

$$\text{then } r(x)(y) = 0$$

$$\text{else } r(x)(y) = b(x)(y)$$

$$\text{else } r(x)(y) = 0$$


Each application of *thin* removes pixels from one corner or edge of the lines in an image. It must be applied eight times, once using each skeletonisation kernel, before testing for termination of the process. *Skeletonise* specifies one complete iteration,



$$Skeletonise$$

$$kernel? : \text{seq } RASTER$$

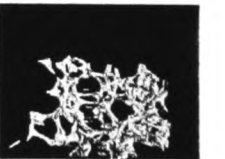
$$dataset?, dataset! : \text{seq } RASTER$$

$$\forall n : \text{dom } dataset? \bullet$$

$$dataset!(n) = thin(kernel(8), thin(kernel(7), thin(kernel(6),$$

$$thin(kernel(5), thin(kernel(4), thin(kernel(3), thin(kernel(2),$$

$$thin(kernel(1), dataset?(n))))))))$$



The algorithm described by *Skeletonise* is executed until $dataset! = dataset?$. The resulting sequence of binary rasters contains only single-pixel-wide lines

which were originally the medial axes, the central lines in each region, of the original edges.

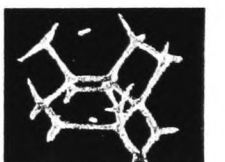
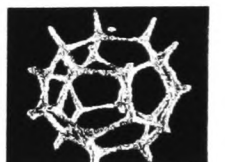
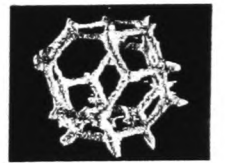
There are several steps in converting the sequence of binary images into a line model. The desired result is a set of lines in space which form a piecewise approximation to the lines in each raster, and there is an upper limit imposed on the length of each line. This is so that a curved line will be closely approximated by a series of short straight lines—if the line length is unlimited, a less accurate model will result. However, the shorter the maximum allowable length, the more lines are needed and hence the more memory is required to store them. An acceptable compromise is that lines are allowed to be up to 20 pixels long.⁶ In the specification, the lines have the set type *LINE*, which is a pair of coordinates in space, *COORD*,

$$COORD == \mathbb{Z} \times \mathbb{Z} \times \mathbb{Z}$$

$$LINE == COORD \times COORD$$

The operation of converting the images to lines is described by the schema,

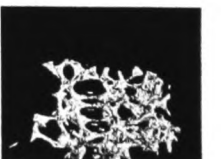
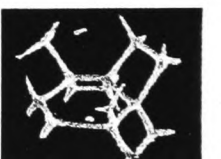
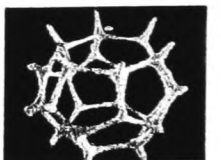
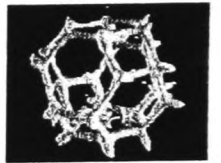
⁶Strictly, the maximum line length is $20\sqrt{2}$ pixel units, since n -pixel lines at 45° in the image have a length which is $n\sqrt{2}$.



Data2lines

dataset? : seq *RASTER*
model! : $\mathbb{P}$ *LINE*

 $\forall n : \text{dom } dataset? \bullet$
 $\forall (start, end) : LINE \bullet (start, end) \in model! \Rightarrow$
 $\exists p : PATH \bullet (\forall (x, y) : \text{ran } p \bullet dataset?(n)(x)(y) = 1)$
 $start = (first\ p(1), second\ p(1), n * 6)$
 $end = (first\ p(\#p), second\ p(\#p), n * 6)$
 $\#p \leq 20$
 $\forall q : PATH \mid q \neq p \bullet \forall coord : \text{ran } q \bullet coord \notin \text{ran } p$



The implementation of this involves simply looking for 1-valued pixels in an image and then following their 8-neighbours until a vertex, an endpoint or the maximum line length is reached, then the coordinates of the line are stored in a file.

Visualisation of this computational model is straightforward. Figure 4.7 shows an example of a view of the line map across all 36 image planes. It is obvious that each Plateau border remains in focus across a range of images. This is because the depth of field of the system is not small enough to blur Plateau borders close to, but not in, the focal plane.

An alternative way to visualise this model is by attempting to coalesce the separate lines in each plane into one Plateau border, almost like stretching an elastic skin over the line framework. Here, a voxel model is used to represent the Plateau border structure. A voxel model is generated from the line model by using the morphological operation enlargement followed by a smoothing

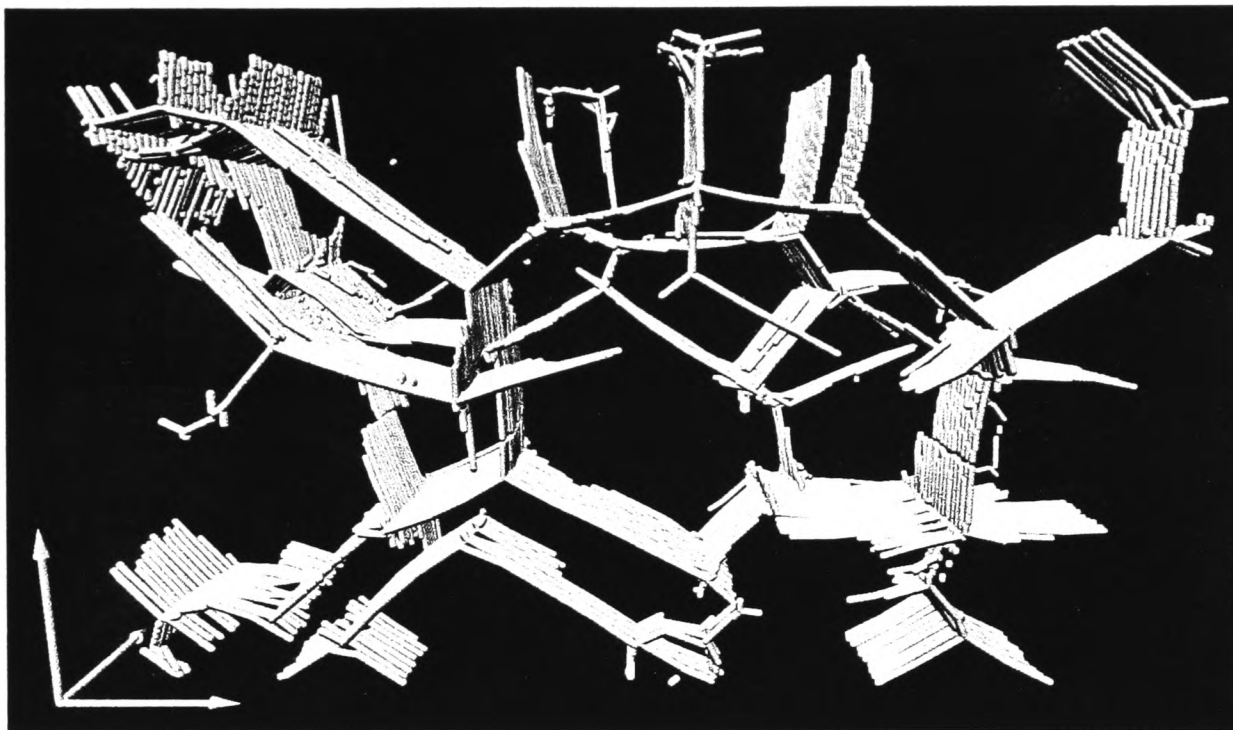


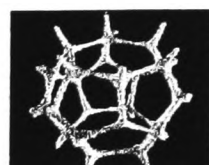
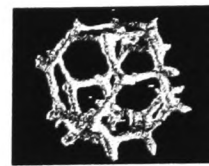
Figure 4.7: Three-dimensional line model of aqueous foam.

process. Converting the set of binary images to a voxel model is described by,

$$\begin{array}{l}
 \text{Data2voxel} \text{ ---} \\
 \text{dataset?} : \text{seq RASTER} \\
 \text{vmodel!} : \text{seq RASTER} \\
 \hline
 \forall n : \mathbb{N}_1 \mid n \leq \text{dom dataset?} * 6 \bullet \\
 \quad \text{vmodel!}(n) = \text{dataset?}(n \text{ div } 6 + 1)
 \end{array}$$

The sequence of binary images is ‘stretched’ along it’s axis and padded to ensure a correct aspect ratio for the whole voxel.

The dilation operator returns the maximum value of a group of 27 pixels as the centre pixel’s new value,



Dilation

$vmodel? : \text{seq } RASTER$

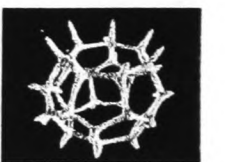
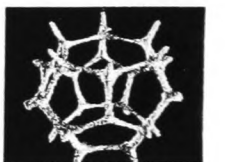
$vmodel! : \text{seq } RASTER$

$\forall x : \text{dom } vmodel?(1); y : \text{dom } vmodel?(1)(1); z : \text{dom } vmodel? \bullet$

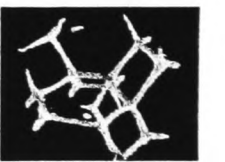
if $1 < x < \#vmodel?(1) \wedge 1 < y < \#vmodel?(1)(1) \wedge$
 $1 < z < vmodel?$

then $vmodel!(z)(x)(y) = \max\{\forall dx, dy, dz : \{-1, 0, 1\} \bullet$
 $vmodel?(z + dz)(x + dx)(y + dy)\}$

else $vmodel!(z)(x)(y) = 0$



This operator substitutes a $3 \times 3 \times 3$ voxel cube for every single voxel in $vmodel?$. This is repeated to thicken the voxel model, and thus to connect line segments in adjacent image planes. To stretch it taut, the annealing operator⁷ is used. This is a cellular automaton which works like a simple majority function over the 27-neighbourhood except that the middle two values are inverted. So, if the sum of values of voxels in the 27-neighbourhood is $s : \mathbb{N}$, a zero result is produced if $s \leq 12 \wedge s = 14$ and unity otherwise. The macroscopic effect is one of surface tension, so the skin begins to deform and shrink snugly over the line framework,



⁷ See Toffoli and Margolus [59] for the equivalent two-dimensional annealing algorithm.



Anneal

 $vmodel? : \text{seq } RASTER$
 $vmodel! : \text{seq } RASTER$

 $\forall x : \text{dom } vmodel?(1); y : \text{dom } vmodel?(1)(1); z : \text{dom } vmodel? \bullet$

if $1 < x < \#vmodel?(1) \wedge 1 < y < \#vmodel?(1)(1) \wedge$
 $1 < z < vmodel?$

then

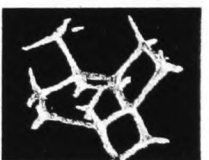
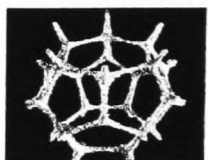
$\exists sum : \mathbb{N} \bullet$

$$sum = \sum_{dx=-1}^1 \sum_{dy=-1}^1 \sum_{dz=-1}^1 vmodel(z + dz)(x + dx)(y + dy)$$

$$vmodel!(z)(x)(y) = 0 \Rightarrow sum < 13 \vee sum = 14$$

$$vmodel!(z)(x)(y) = 1 \Rightarrow sum > 14 \vee sum = 13$$

else $vmodel!(z)(x)(y) = 0$



The repeated application of *Anneal* results in the solid model shown in figure 4.8.



Assuming that a depth of field can be chosen which is small relative to the distance between photographs, there is a further post-processing step which can be applied to the line list model of the Plateau border geometry. Although this system cannot produce a small depth of field, it can be simulated by passing a three-dimensional skeletonisation filter over the voxel model and then padding this in the z direction with space. The result of this operation is shown in figure 4.9. Now consider one Plateau border, which crosses several image frames as in figure 4.10. The whole line in the model is represented by a series of small segments lying in the focal planes of the original photographs. A better model would connect each of these loose ends



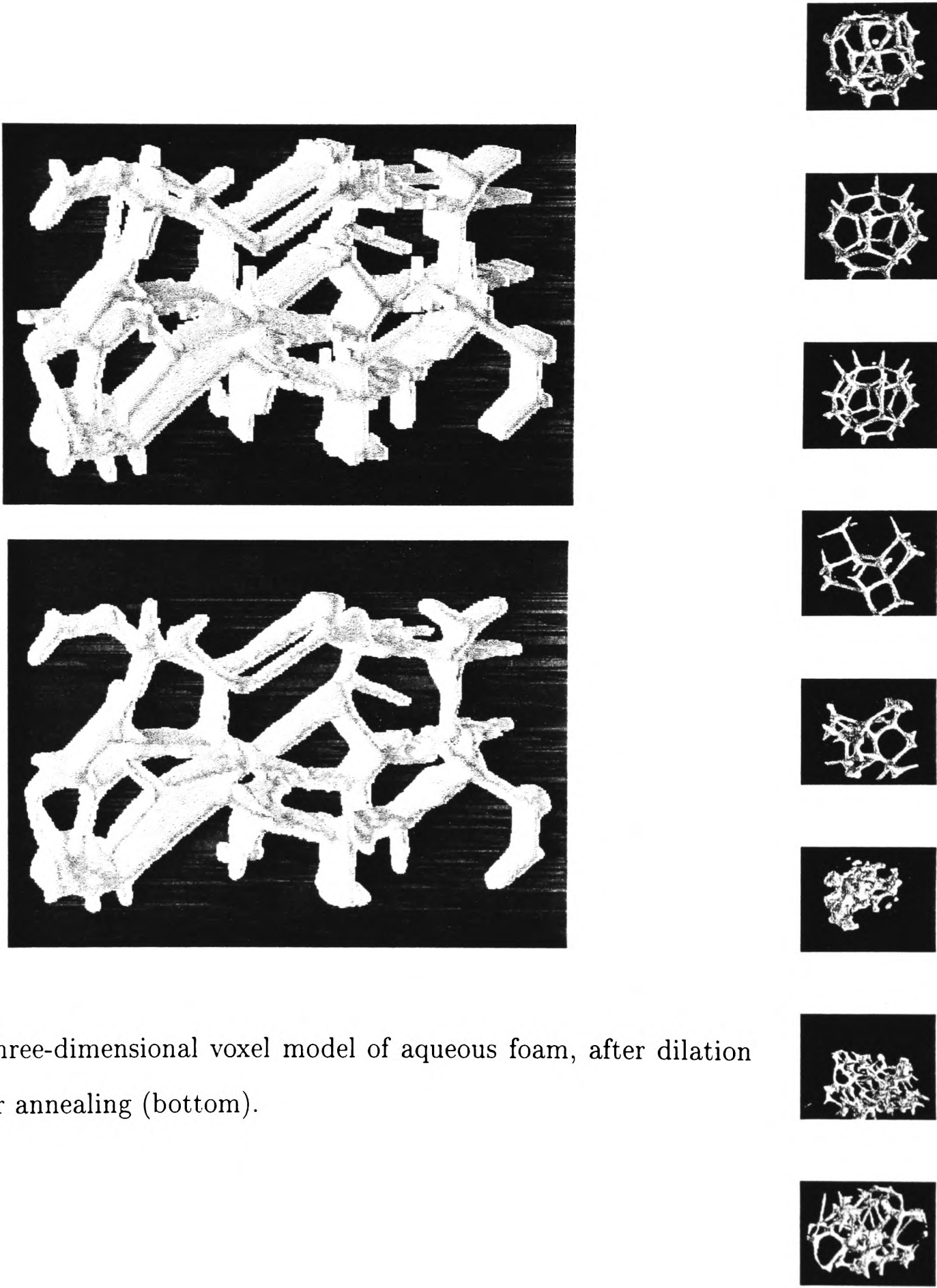


Figure 4.8: Three-dimensional voxel model of aqueous foam, after dilation (top) and after annealing (bottom).

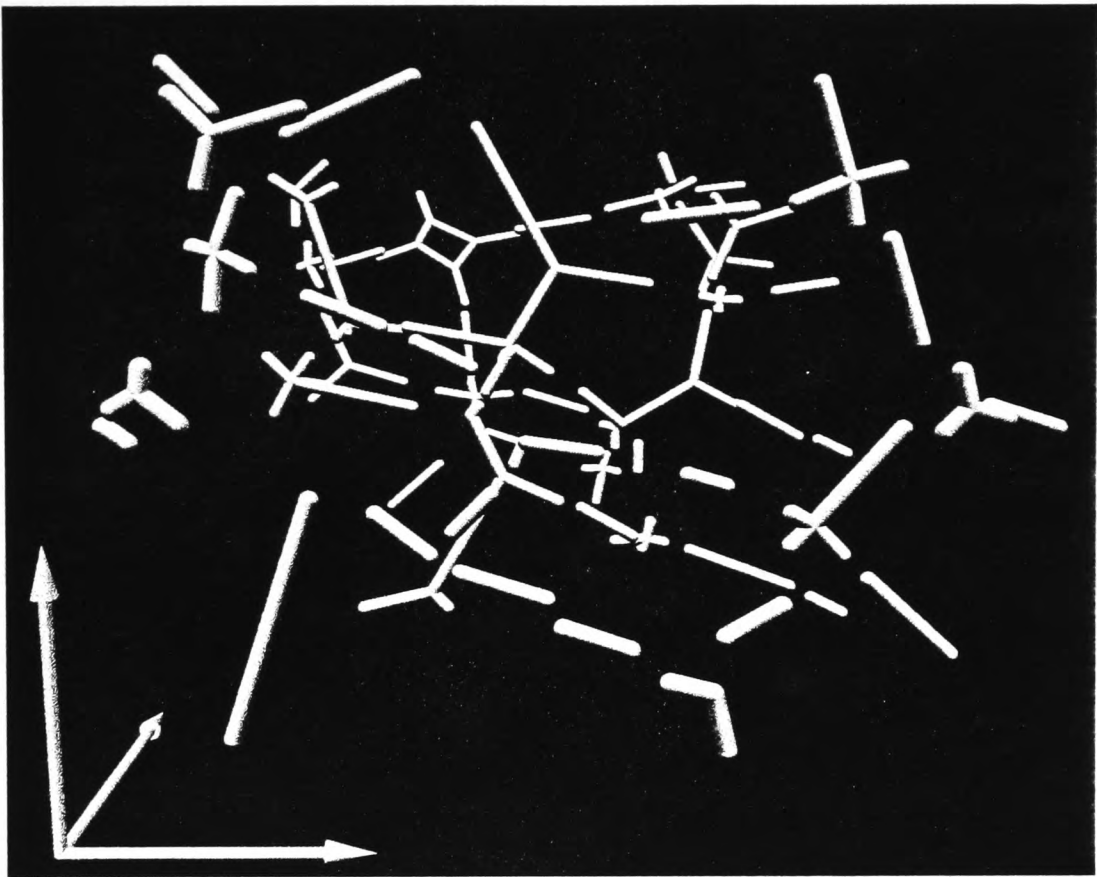


Figure 4.9: Foam line model after skeletonisation and padding.

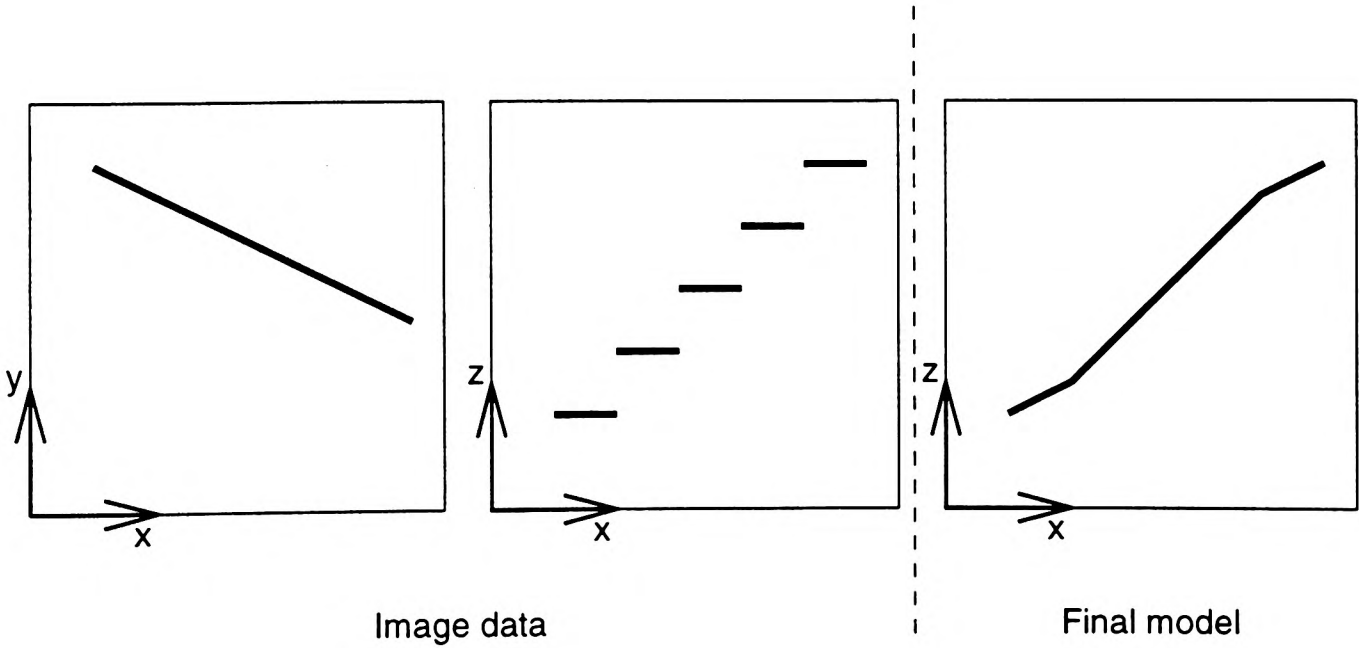
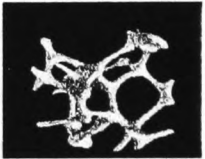
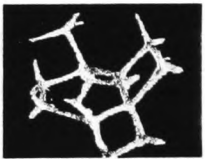
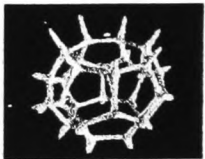
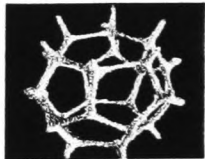
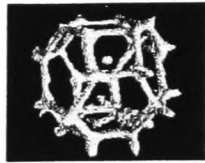


Figure 4.10: Connection of line segments in the Plateau border model.



to produce a single line segment across the model, interpolating between the images where there is no data.

Given an unordered list of straight lines in space, each line end point may either be a free end which should be linked to another line segment to form one uniform segment, or the end point may be part of a vertex in an image where two or more lines meet. In the former case, a search is required to find a matching line to be attached, but otherwise the point is left alone. Three functions are defined, *IsVertex*, *Near*, and *CoAverage*. The first of these tests a point to see if it has any duplicates in a frame, and hence is a vertex. The second is a relation on whether two points are near neighbours, and the last joins two vertices at the midpoint of the line which joins them,⁸

$$IsVertex : \mathbb{P} LINE \leftrightarrow COORD$$

$$Near : \mathbb{P} COORD \times COORD$$

$$CoAverage : \mathbb{P} COORD \times COORD \rightarrow COORD$$

$$\forall lmodel : \mathbb{P} LINE; x, y, z : \mathbb{Z} \bullet$$

$$IsVertex(lmodel, x, y, z) \Leftrightarrow$$

$$\exists u, v : lmodel \bullet (fst3(u) = fst3(v) \vee$$

$$fst3(u) = snd3(v)) \wedge u \neq v$$

$$\forall a, b : COORD \bullet Near(a, b) \Leftrightarrow ((fst3(a) - fst3(b))^2 +$$

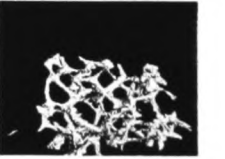
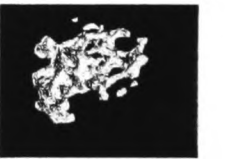
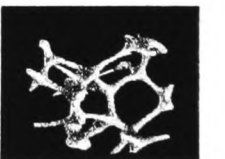
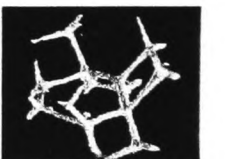
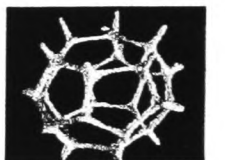
$$(snd3(a) - snd3(b))^2 + (thd3(a) - thd3(b))^2)^{\frac{1}{2}} < 20$$

$$CoAverage(a, b) = ((fst3(a) + fst3(b)) \text{ div } 2,$$

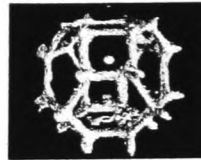
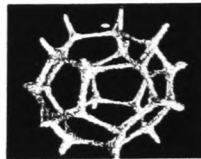
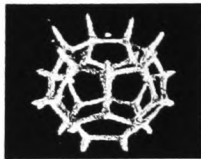
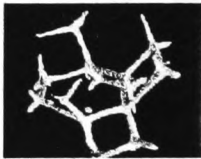
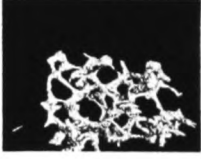
$$(snd3(a) + snd3(b)) \text{ div } 2, (thd3(a) + thd3(b)) \text{ div } 2)$$

These functions are required by *JoinLines*, which scans the wire model and

⁸ The functions $fst3, snd3, thd3 : COORD \rightarrow \mathbb{R}$ return the first, second and third components of a vector triple. These are the triple equivalents of *first* and *second*.



joins edges which are near neighbours in adjacent image planes, provided none are vertices,

<i>JoinLines</i>	
$lmodel?, lmodel! : \mathbb{P} LINE$	
$\forall l_1, l_2 : lmodel? \mid l_1 \neq l_2 \wedge \neg (IsVertex(first(l_1)) \vee$ $IsVertex(second(l_1)) \vee IsVertex(first(l_2)) \vee$ $IsVertex(second(l_2))) \bullet$ $Near(first(l_1), first(l_2)) \Rightarrow$ $(CoAverage(first(l_1), first(l_2)), second(l_1)) \in model!$ $(CoAverage(first(l_1), first(l_2)), second(l_2)) \in model!$ $Near(first(l_1), second(l_2)) \Rightarrow$ $(CoAverage(first(l_1), second(l_2)), second(l_1)) \in model!$ $(first(l_2), CoAverage(first(l_1), second(l_2))) \in model!$ $Near(second(l_1), second(l_2)) \Rightarrow$ $(first(l_1), CoAverage(second(l_1), second(l_2))) \in model!$ $(first(l_2), CoAverage(second(l_1), second(l_2))) \in model!$	      

The set *model!* is a continuous model of the foam Plateau borders. Depending on the quality of the photographic data, this may be very representative of the real foam, especially if depths of field and distances between image planes are small. Figure 4.11 shows an example model. No explicit tests of the fidelity of the reconstruction algorithm were performed other than visual inspection. Given that no other mechanism capable of resolving the structure of liquid foams with either temporal or spatial resolution matching the optical system exists, testing the system with real foams is probably

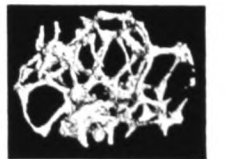
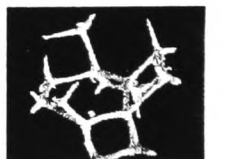
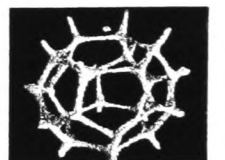
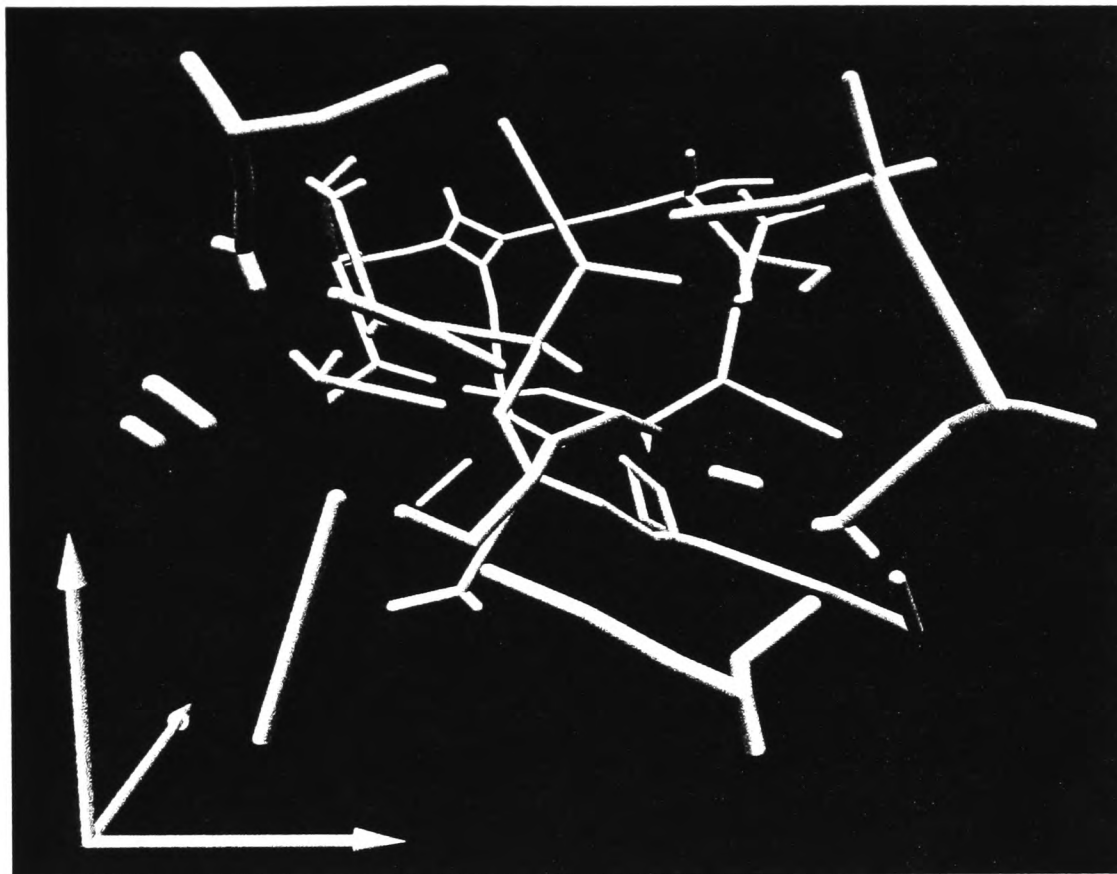


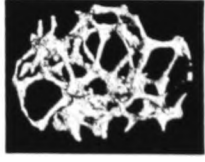
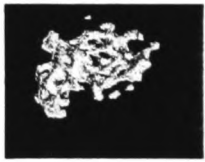
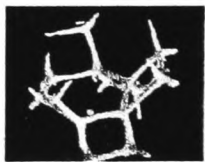
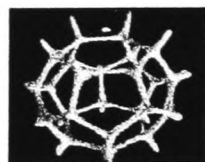
Figure 4.11: Three-dimensional line model of aqueous foam after smoothing.

impossible. However, it is possible to produce ‘photorealistic’ images with raytracing software on a PC. With considerable effort, a set of test images could be prepared artificially and then used by the reconstruction software.

Traditional film processing followed by scanning prints is a very inefficient method of providing photographic data for these experiments. The entire *ImagePreprocess* step, which is computationally expensive for a large number of images, can be avoided if a CCD camera and framegrabber are used to supply the images. This is because all the problems caused by the slow capture rate and uneven printing of the photographic technique are solved by a reasonably fast digital system. Each digital image is properly registered with respect to its neighbours, the grey level palette is uniform across the set of images and the foam does not have time to sink or rearrange during

the much shorter capture period.

Chapters five and six describe axial tomographic methods of resolving the structure of a foam with a digital image system. Further discussion of the relative merits of all three methods can be found in chapter seven.



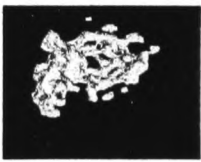
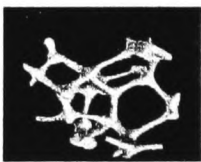
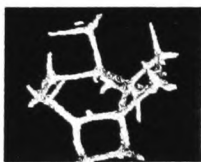
Chapter 5

Optical axial tomography

A method of capturing three-dimensional foam structure information, which does not suffer from the limitations of the small depth-of-field system described in chapter four, is provided by a variant of traditional X-ray tomography. This chapter explains how simple tomographic machines work and briefly describes other work in related fields to foam study, before ending with an account of the development of the author’s optical tomography system.

5.1 Tomographic principles

The earliest work on the basic mathematics for tomography was done by Radon in 1917 [60]. Tomography derives a relation between the intensities of samples of radiation attenuated by passing through an object of interest, and the reconstructed model of the object. For example, when a ray of x-radiation passes through a bone, the fraction of radiation which passes through the bone gives an indication of the integral of the bone density along the ray path. By shooting many rays through the bone at different angles



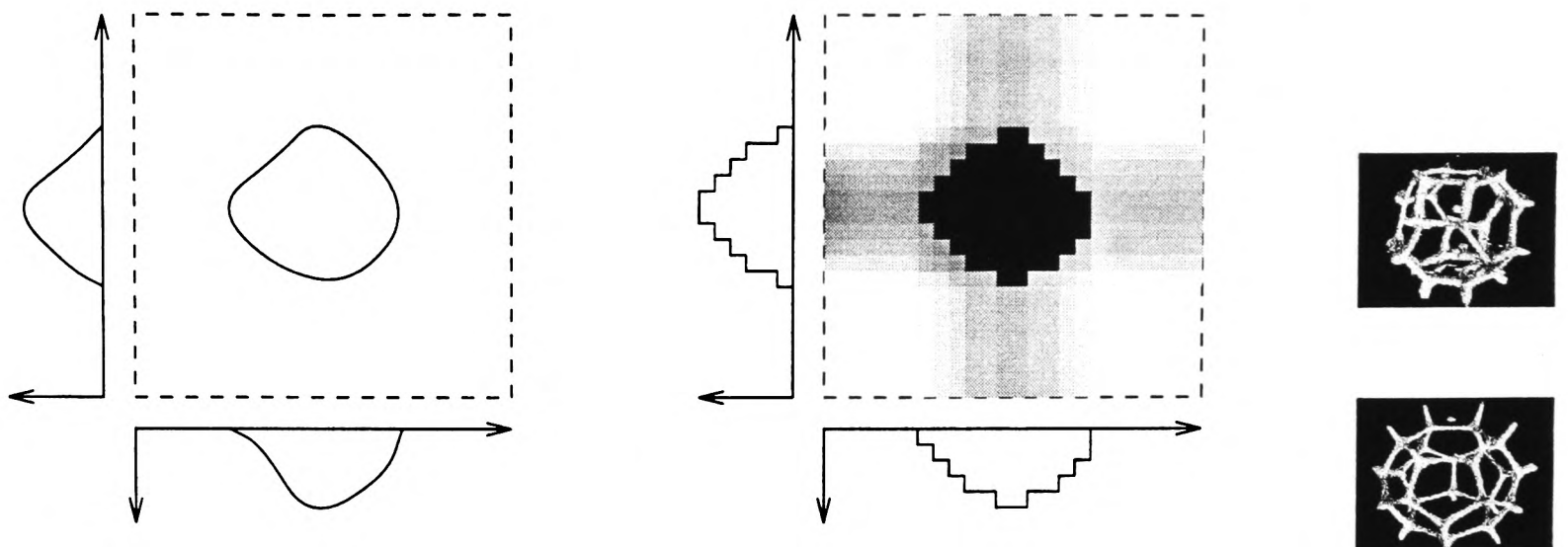


Figure 5.1: A two-dimensional tomographic system with two orthogonal projections, sampling (left) and reconstruction (right).

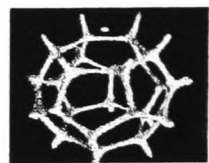
and observing how much radiation is absorbed, a two- or three-dimensional image of the bone structure can be determined. If all the rays lie in one plane, a two-dimensional image is produced [61]. With added instrument complexity, if rays are unconstrained in three-dimensions, a volume model of the bone sample may be reconstructed. This principle has been extended to other fields and other imaging radiations [62].

Early tomographic systems used simple X-ray sources and photographic plates to resolve a cross-section through an object [63]. With the development of computers came more accurate digital tomographic systems; a simple example of a two-dimensional digital tomographic system is shown in figure 5.1. This has two pairs of emitter and detector linear arrays arranged at 90° to each other. The left-hand picture shows the attenuation in radiation intensity perceived by the detector arrays, which are assumed to contain an infinite number of elements, caused by a semi-opaque solid object which blocks radiation to some degree. The right-hand image shows

a reconstruction using a matrix of 24×24 pixels. The reconstructed model is generated by *back-projecting* discrete values of ray absorption into the pixel matrix. For each ray, the average density value along its path is placed into the pixels along that path in the reconstructed model. As more rays are traced through the reconstructed matrix, the model begins to become more defined and the signal to noise ratio increases. Finally, when all rays have been back-projected into the matrix, the values are normalised and a *threshold* is chosen such that all pixels above the threshold value are tagged as belonging to the object and all other pixels are considered to be part of the space around it. Another way to look at the problem is that the pixel matrix holds a set of probability values of the chances of material being present at a particular pixel. By setting a threshold value, it is possible to decide what likelihood of being solid a pixel should have before it is considered to be part of the reconstructed model.

It is clear that a two-projection system like this will not provide very accurate tomographic reconstructions. The quality of reconstruction depends on the number of unique ray samples, where more is better. A commercial medical scanner will have over a thousand detectors in a linear array which can be rotated around a person in steps of fractions of a degree, giving many hundreds of unique projections. It is then possible to obtain a reconstructed cross-section at resolutions of 1000×1000 pixels, which are often needed to see the small symptoms of ailments accurately.

The two-dimensional system described here can be extended to three dimensions. The principles are the same, but there is extra complexity when designing emitter and detector arrays for three-dimensional use. A projection



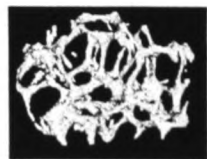
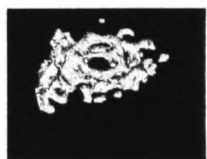
now becomes a two-dimensional image, and the reconstruction takes place using a cubic matrix of voxels. As will be described later, an elegant solution is to use a CCD camera as the detector array in an optical tomography experiment. The data bandwidth is relatively high compared to X-ray systems, and it is very much cheaper and safer to use. Fortunately, the study of foam cellular structures is possible with optical tomographic methods, although there are some drawbacks.

Other researchers have also experimented with optical methods. In order to understand the advances made in the foam work, it is useful briefly to cover some previous work in related fields. This material is covered in the next few sections.

5.2 Interferometry

Early optical tomography experiments have exploited the interference properties of coherent light rays passing through materials of varying refractive index, and hence varying optical path length. One such experiment involved the tomographic determination of refractive index variation across the cross-section of a Pyrex rod.

Snyder and Hesselink [64] used the apparatus shown in figure 5.2 to resolve ten 360×120 -pixel projections of the Pyrex rod. A semi-silvered mirror is used to split a laser beam into a reference component and a component which will pass through the object being studied, the Pyrex rod. Two holographic optical elements (HOEs) have been specially made for the experiment—these are essentially accurate converging mirrors, designed so that a parallel beam of light passes through the rod. The two separate laser beams are recombined



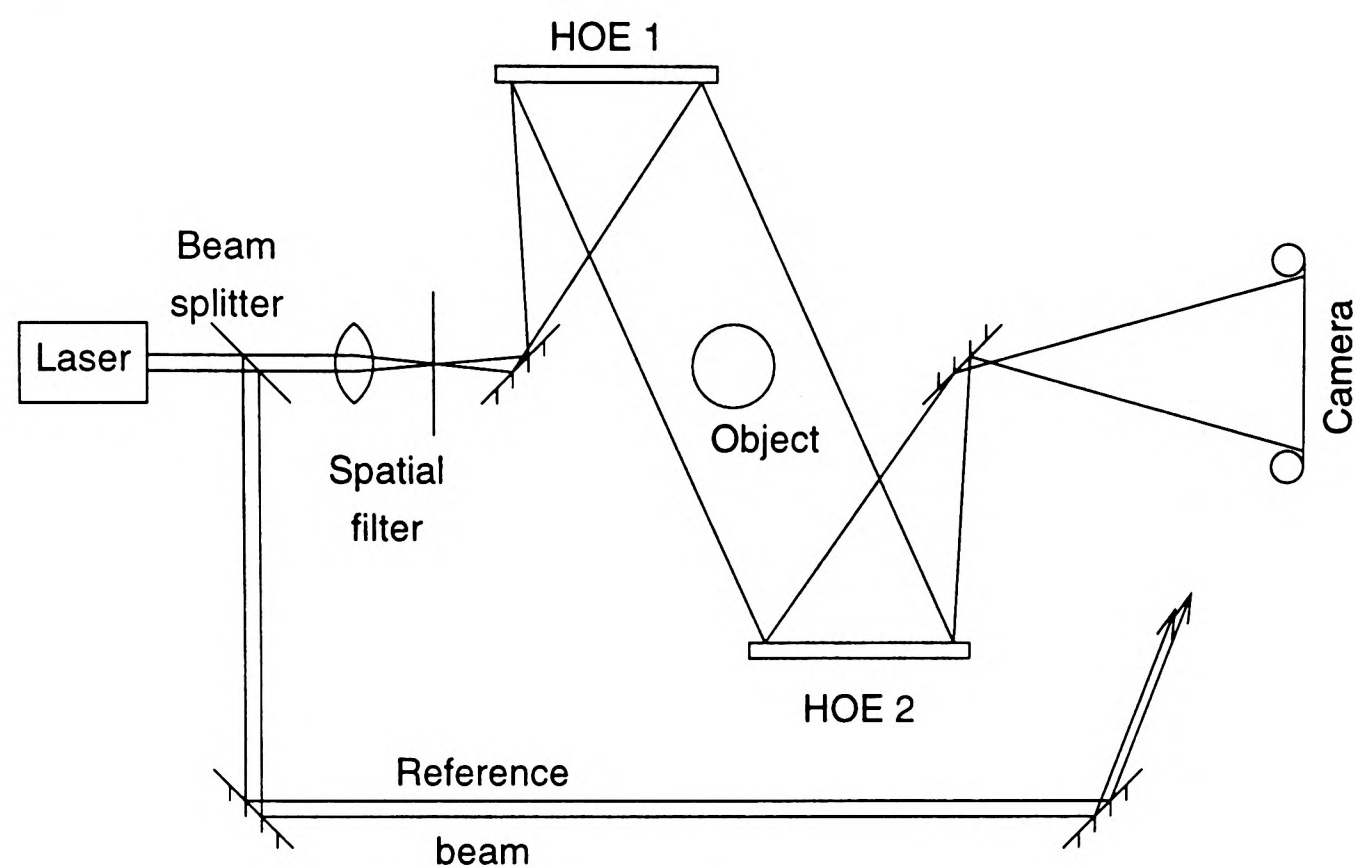


Figure 5.2: An optical tomography system based on laser interferometry.

at the camera imaging plane to form an interference pattern consisting of fringes aligned nearly perpendicular to the axis of the rod. As the sample ray passes through the rod the change in path length which occurs is shown by a vertical movement of the fringes at the corresponding horizontal position of the beam.

By measuring the displacement of the fringes at each horizontal sample point, and resolving the phase wraparound which occurs for path length differences greater than one wavelength, a linear projection can be formed which can in turn be used in a tomographic reconstruction. The refractive index variation of a simple object such as a glass rod can be reasonably well resolved from only ten projections.

The reconstruction of the Pyrex rod leads on to a popular target for



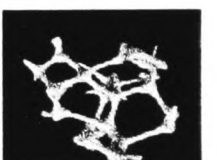
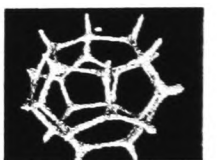
optical tomographers—the gas jet. Various researchers [65, 66] have used a similar set up, perhaps with changes designed to speed up data collection and optimise the reconstruction algorithm, but the principles are the same. Modifications include using many mirrors to produce multiple projections from one laser beam simultaneously, or using a calcite crystal prism at various orientations to the sample beam to vary the fringe width and hence the resolution of an interferometer [67], or even using a fixed fibre-optic and photodiode array time-multiplexed to give 100 projections of 100 samples in less than 200ns [68].

Another popular area of study using optical tomographic systems is that of combustion and flame analysis. For a rotationally-symmetric flame, one image is enough to reconstruct a three-dimensional representation [69]. It is also possible to reconstruct in three dimensions a temperature map of a group of flame jets [70] and to control a combustion process using tomographic data [71].

5.3 Three-dimensional X-ray angiography

Computerised angiography seeks to resolve the geometry of the network of blood vessels in living tissues. Using *digital subtraction angiography* (DSA), where two X-ray images taken within a short time period are numerically subtracted to reveal which parts of each image have changed, blood carrying an iodine tracer through arteries and veins shows up with very high contrast.

Researchers at Thomson [72] and General Electric [73, 74, 75, 76] are using several digital angiograms to reconstruct the three-dimensional structure of blood vessels in living tissues, with a method based on cone-beam



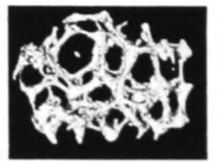
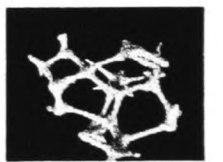
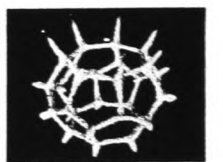
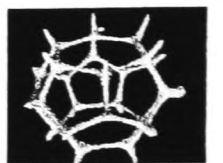
tomography. The resolution of each angiogram is good, but there is only a limited time in which to gather the information, so the number of images is small, typically less than 50.

The reconstruction is a two-stage process, after each projection image has been prepared. Firstly, a *region of support* is determined which reduces the number of voxels needing to be considered for further processing. A voxel is eliminated from consideration if there is at least one frame in which the voxel is clearly empty, and that voxel is not immediately adjacent to a non-empty voxel. Then, the model is reconstructed using a variant of the algebraic reconstruction technique with constraints on the solution, such as non-negativity of voxel values.

The team have noted the need for rapid data acquisition, and hence have built an X-ray camera system which can sample several tens of high-resolution images (512×512 pixels) in the few seconds that the blood vessels are opaque to the X-rays. The reconstructed model resolution is either $256 \times 256 \times 256$ or $512 \times 512 \times 512$ voxels, the processing being carried out by a parallel 'mini-supercomputer'.

A group in Japan [77] has tried to apply a *simulated annealing* [78] algorithm to the reconstruction of DSA images. Working with computer simulations only, they reconstruct a model of the heart and surrounding arteries using only eight projections.

Starting with an arbitrary voxel set, random perturbations are introduced to each voxel value, and if a specified cost function is minimised the changes are allowed to remain. Or, if the cost function is increased by less than a value related to the system 'temperature' in a Boltzmann probability function,



$e^{\frac{-\Delta E}{kT}}$, the changes also stand. The temperature is slowly reduced during the calculation, so that eventually the system settles into an optimal solution. This is the principle of simulated annealing.

In order for this method to work at all, the cost function must be carefully chosen so that minimizing it accurately reflects the desired result. The authors have noted that the intensity distribution in a DSA image is bimodal—essentially the images are high contrast and have a high signal-to-noise ratio; blood vessels marked with a tracer show up strongly against a uniform background. Each voxel in the reconstructed model therefore has a value of either zero or one. In order to perturb the system, a statistical model is set up, where each voxel has a 20% chance of ‘flipping’ or inverting its state.

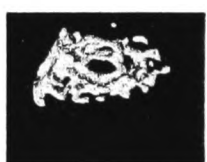
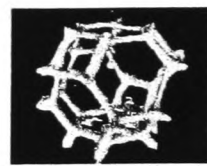
It is not necessary to devise a global cost, or energy, function, since only the relative change after an alteration to the voxel configuration is needed. The authors, Haneishi and Ohyama, propose,

$$\Delta E = \Delta E_s + w_h \Delta E_h + w_c \Delta E_c$$

ΔE_s is the mean square error in the projection domain. For each voxel alteration, the effect on the set of projections is calculated and compared to the original projections to form ΔE_s . There are two weighting terms, w_h and w_c , which the authors only describe as ‘positive’.

ΔE_h is effectively a bounded reciprocal of the projection pixel intensity histogram, such that maximum values in the histogram are minimal energy values. The value of E_h is then determined in conjunction with the statistical calculation of the size of voxel perturbation as mentioned above.

The E_c term is an adjustment which takes into account the continuity of



space around each voxel,

$$\Delta E_c = p \cdot s$$

where p depends on the direction of voxel value change, and s is a weighting factor which depends on the spatial continuity of the local voxel neighbourhood. The values of p are taken as follows,

$$p = \begin{cases} 0 & \text{no change} \\ 1 & \text{jump from one to zero} \\ -1 & \text{jump from zero to one} \end{cases}$$

The parameter s is more interesting. The reconstructed model is expected to have most of its voxels ‘well’ connected together, and spurious individual voxels, possibly caused by noise, are unwanted. In three dimensions, this can be represented by s as follows, where n is the number of one-valued voxels in the 26 immediate neighbours of the voxel being considered,

$$s = \begin{cases} n/26 & n > 0 \\ 1 & n = 0 \end{cases}$$

Overall, minimising E_c will produce a tidier, less noisy reconstruction.

Figure 5.3 shows the result of applying the simulated annealing algorithm to a computer-generated phantom consisting of a cylinder surrounded by four spirals (this is meant to be a heart and arteries). The reconstructed model is a good representation of the original, with only 25 error voxels out of 32768 (32^3) total voxels. However, the model resolution of $32 \times 32 \times 32$ is too low to be applicable to most problems. The simulated annealing algorithm is unfortunately too computationally intensive, and machines which can efficiently run such programs on large data sets are still not feasibly cheap. However,



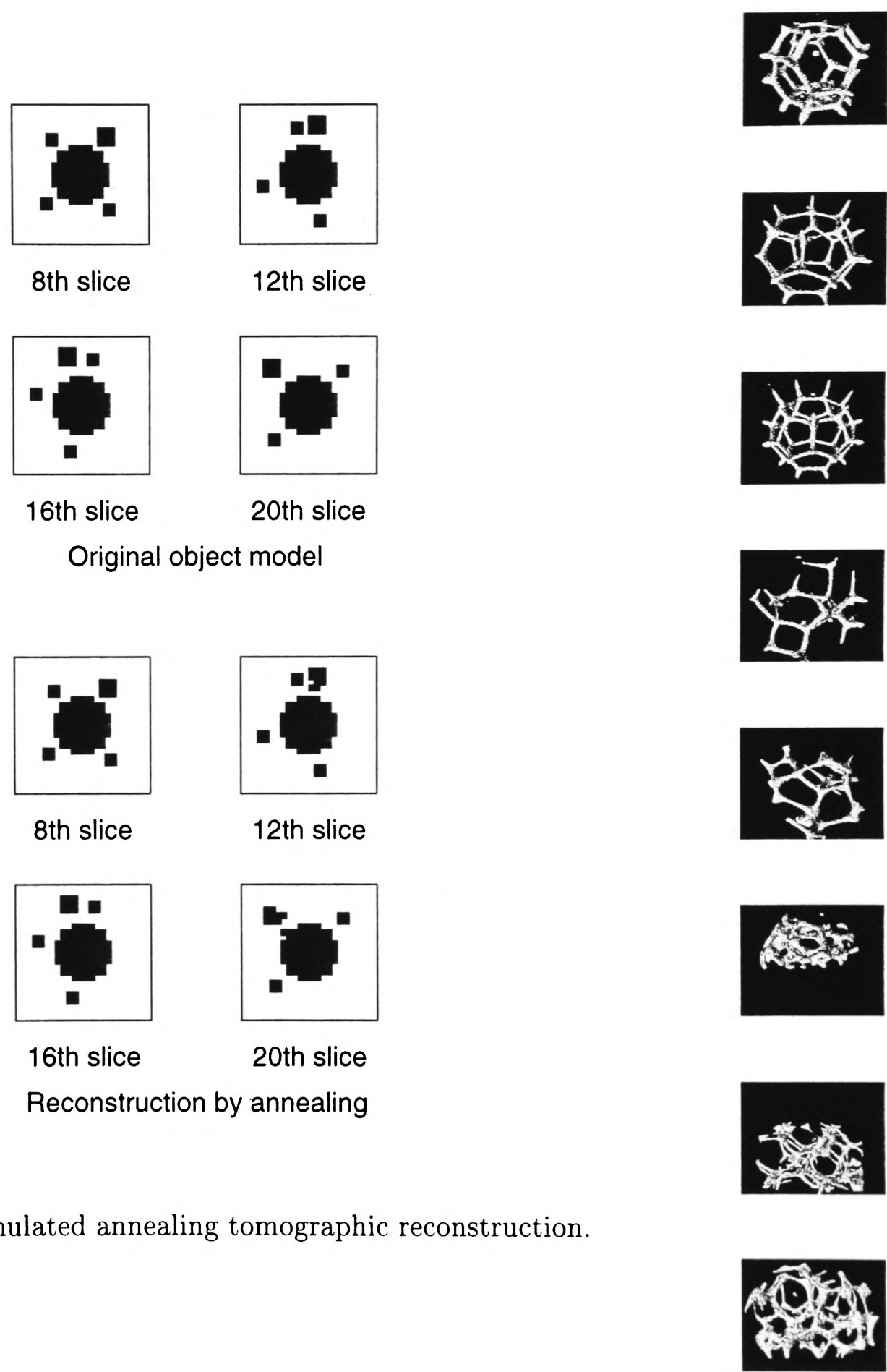


Figure 5.3: Simulated annealing tomographic reconstruction.

the general idea of optimisation in cone-beam tomography is a good one, given the lack of data provided by most sampling geometries.

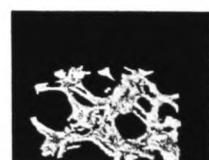
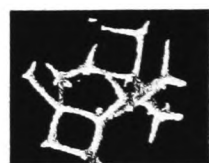
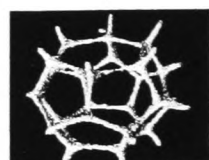
5.4 Microtomography

There have been attempts to scale down the tomography hardware to study small objects in the laboratory. Some of these are based on gamma-ray systems [79] and others on X-rays [80]. The latter method converts X-rays to visible light using a scintillator so that video images of the X-ray projections may be downloaded into a computer.

There have been a couple of attempts to use light rather than gamma- or X-rays in computerised tomography experiments. Both of the sets of researchers used modified microscope apparatus to study small objects—no known work exists for the macroscopic case.

5.4.1 Optical microscope tomography

Japanese researchers [81, 82, 83] have developed an optical tomography machine based on a standard optical microscope. They have added a mask with an off-axis pupil in front of a transmitted-light source, which masks all light rays except for a small subset that pass through the sample at an oblique angle. By rotating this mask, the direction of light passing through the sample can be altered. The configuration is shown in figure 5.4. An important parameter in the system is the angle the ray paths make with the axis of the microscope lens. Here, this is 18° ; although a larger angle would probably give better results, the constraint is the geometry of the microscope viewing



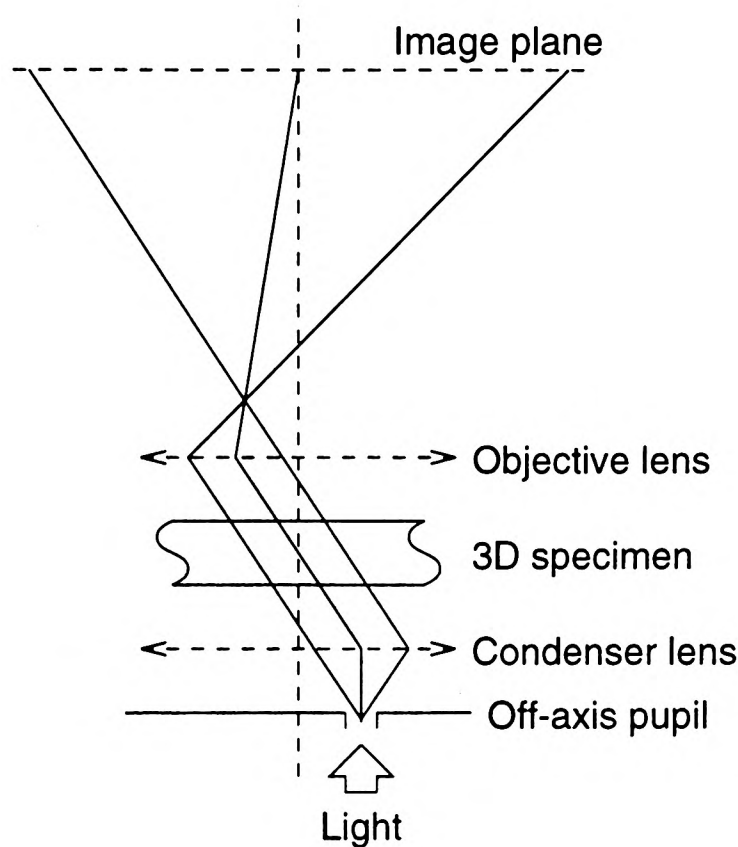


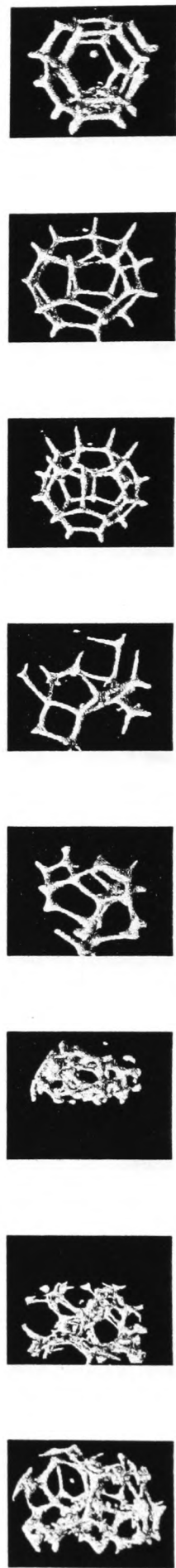
Figure 5.4: Optical microscope configuration.

system, in particular the size of the objective lens and its distance from the off-axis pupil.

The researchers have used the system to study the single-celled plant, *Spirogyra*. This is a cylindrical organism characterised by a helical chloroplast coiled around the inner cell wall and a dark central nucleus, and hence makes a good test subject for tomographic analysis.

5.4.2 Optical tomography of a cochlea

A group at the University of Washington in Seattle [84] has done some work on optical tomography of translucent objects, including a guinea pig cochlea. After decalcification and staining with a blue dye, the cochlea is a suitable test object for an optical tomography experiment, being semi-opaque and



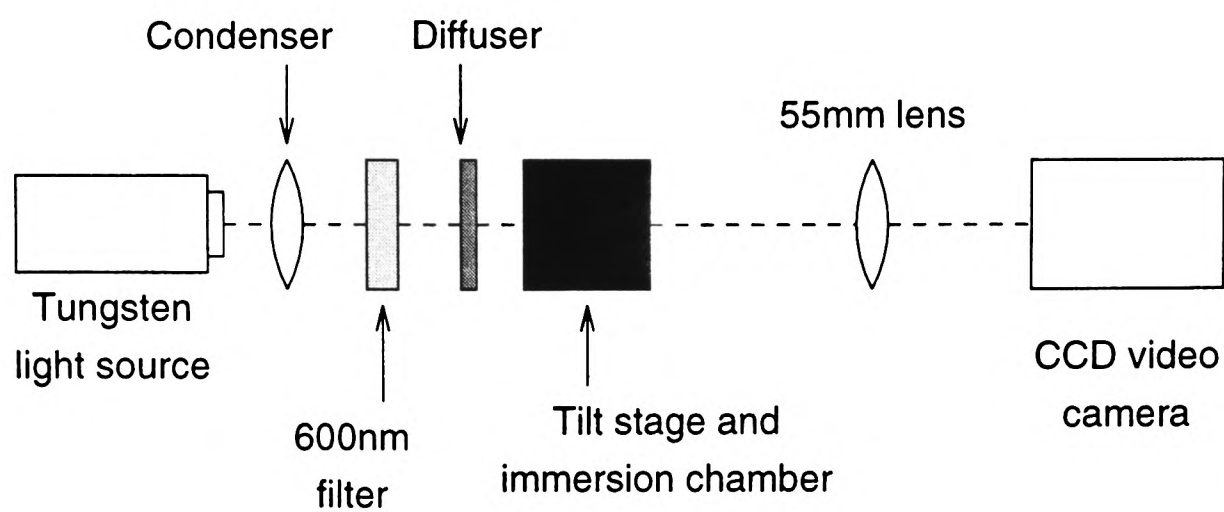
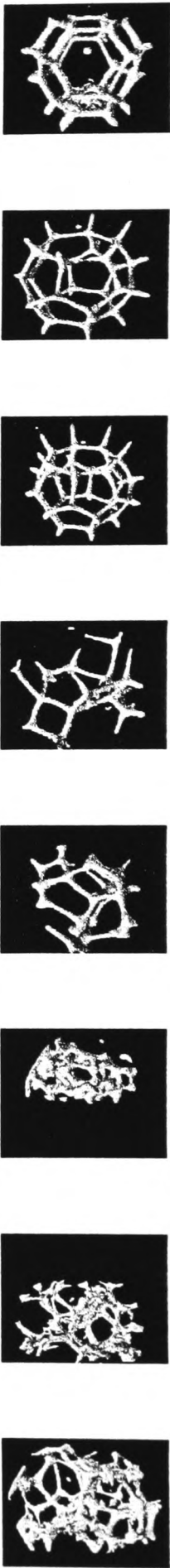


Figure 5.5: Optical tomography system configuration.

having a spiral tectorial membrane running through the centre of its canal.

A block diagram of the experimental system is shown in figure 5.5. The cochlea is placed at the end of a probe on a microscope tilt stage and immersed in Spalteholtz fluid to reduce scattering, which is also the reason for using a 600nm light source. Using an 8-bit digitiser, one projection is formed by summing 16 240×240 transmission images into a 16-bit buffer to reduce the effects of background noise. After correcting for detector non-uniformity, 96 evenly-spaced projections are captured over a 180° arc. A reconstruction of the cochlea is performed by convolution back projection, considering each voxel plane in turn, since the authors are making the assumption that each projection is parallel. Given the quality of the results, this seems reasonable although the reality is that the light rays forming the projection do diverge slightly.

Figure 5.6 shows a maximum voxel view of the cochlea reconstruction and a vertical section through the centre point. The tectorial membrane, which shows up strongly in the model, is arrowed in the second picture.



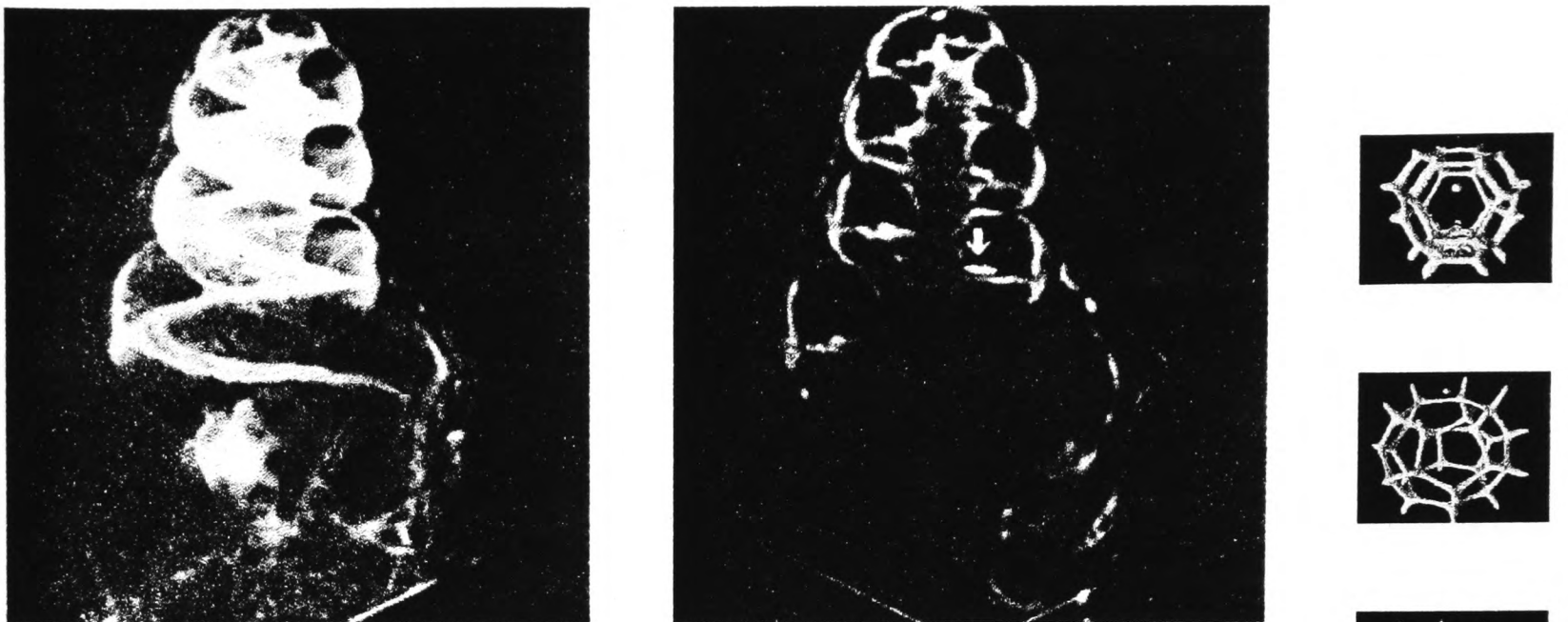
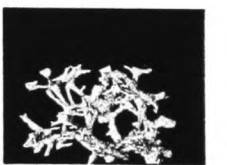
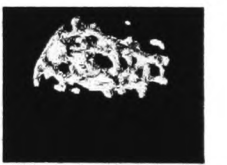
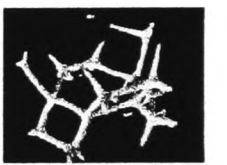
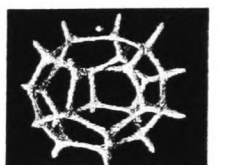
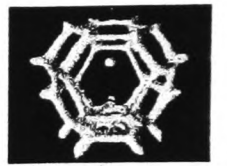


Figure 5.6: Tomographic reconstruction, using visible light, of the inner ear of a guinea pig; left, one projection, right, a cross-section through the tomographic model (Brown *et al* [84]).

The experiment shows that it is possible to use visible light to resolve the dimensions of relatively solid objects, as long as they are not totally opaque. Also, it demonstrates the usefulness of a CCD camera as a fixed detector. The authors note the lack of slice registration problems because the CCD array elements are all fixed relative to each other, and no movement is required, as it would be with a point detector or linear array.

5.5 Process tomography

Process vessels tend to be opaque, so X-ray tomography is often used to resolve multiphase flows in pipes, for example. However, other forms of tomography using combinations of ultrasound, electrical capacitance or impedance, gamma-ray and positron-emission imaging systems have been developed. See



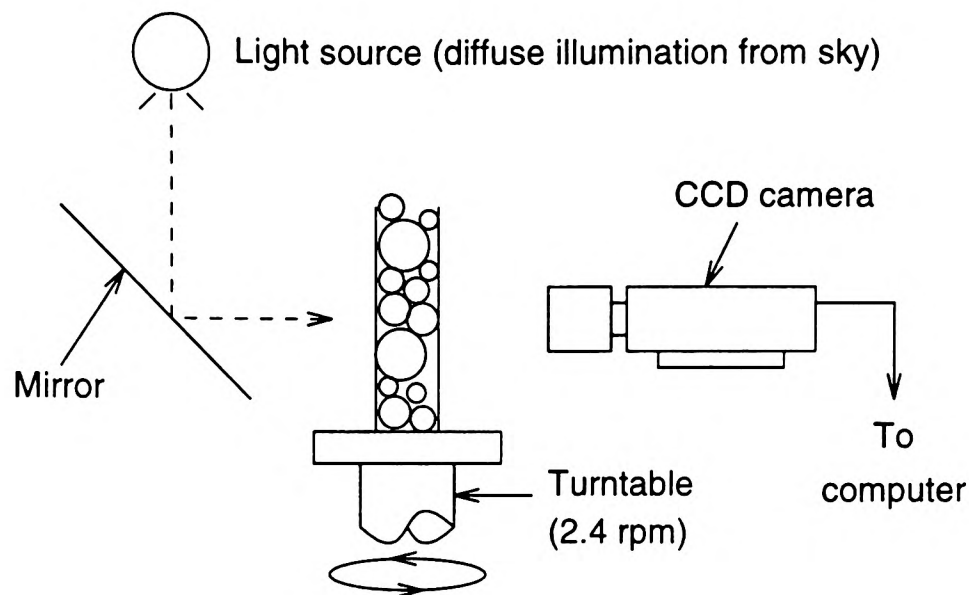


Figure 5.7: Experimental rig for foam tomography.

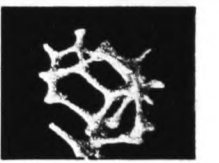
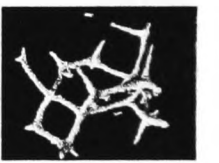
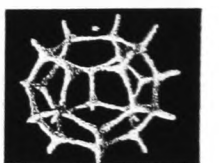
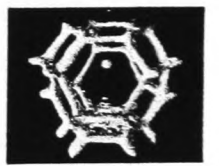
the proceedings of the recent European Communities Workshop on Process Tomography [85] for a review of the state of the art.

Optical systems are currently rare. However, new applications which could be resolved with optical tomography are being discovered all the time; foaming processes are an example.

5.6 Optical foam tomography

Several types of foam are suitable for analysis with optical tomography. There are some limitations to the technique, since the foam must be transparent to some degree and must remain stable for the duration of the data-collecting phase. During the course of this work, both solid and liquid foams have been studied, and examples of both are shown to be suitable to this type of analysis.

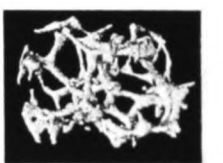
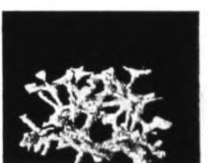
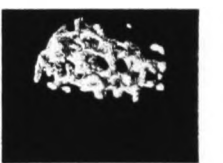
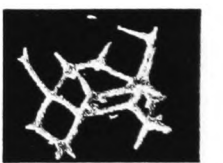
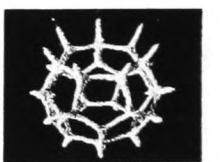
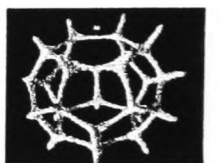
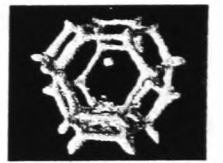
The experimental rig used initially is shown in figure 5.7. This configuration is for studying liquid foams, which are contained in a glass tube on



the turntable. For analysing solid foams, this is replaced with a simple wire support for the solid foam sample. The best source of diffuse illumination is created by reflecting an image of the sky outside the laboratory off a mirror placed behind the foam. For either clear blue sky, or complete cloud cover this gives a uniform bright background which produced a crisp image of the foam on the camera CCD array.

The turntable rotates at about 2.4 revolutions per minute. This gives a sampling time of roughly 25 seconds in which to digitise as many images of the foam as possible. There is a trade-off between sampling time, image resolution and number of images for a given video bandwidth. In tomography applications, it is generally thought that the optimal ratio of image dimension to number of images is 1:1. So for a set of 200 video images, each should have a resolution of 200×200 pixels, and this data should be used to reconstruct a voxel model of resolution $200 \times 200 \times 200$ voxels.

The complete video subsystem consists of a Dage-MTI 72S charge-coupled device camera, of array resolution 750×580 elements, connected by a PAL interface to a VideoLogic Captivator framegrabber sitting in a 486/66 personal computer. Video for Windows is the software used to drive the Captivator. This is an extremely low cost system, and as a result the available video bandwidth is relatively poor. The Captivator resides on the ISA backplane of the PC, and is capable of transferring 500kB per second to the processor and computer memory. The system is able to capture images at the rate of 25 frames per second (the camera limit) but only at a resolution of 160×120 pixels. At higher resolutions the frame rate degrades rapidly. Even so, this is still enough to enable tomographic experiments on small foam samples.



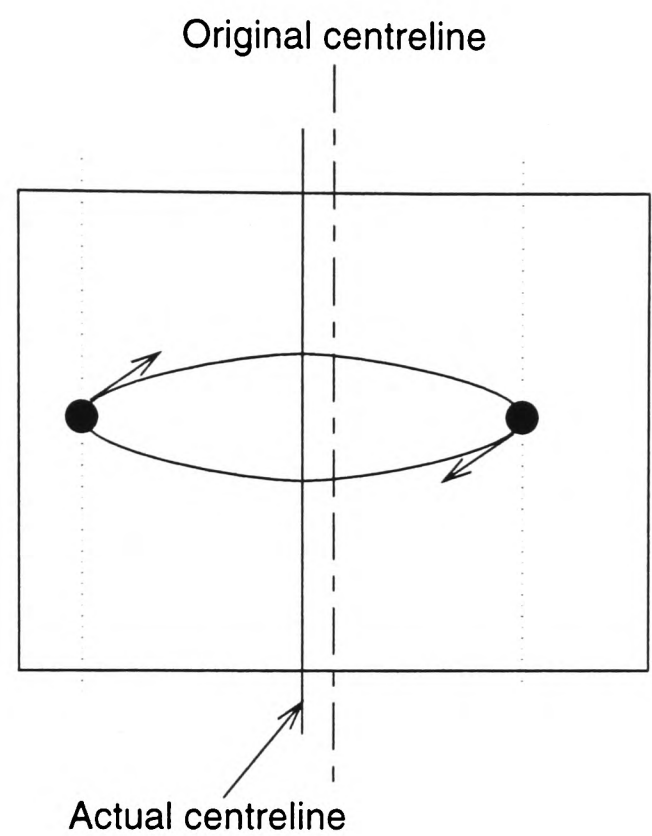
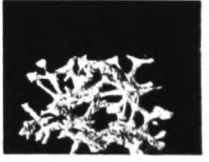
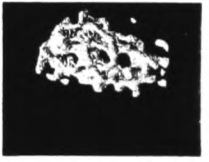
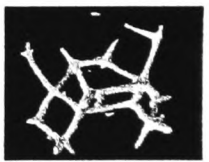
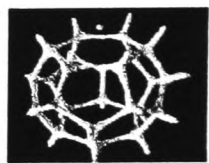
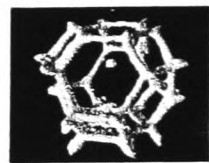


Figure 5.8: Determination of image centre of rotation.

The camera mount guarantees that a vertical line in the image will be in the same vertical plane as the axis of rotation of the turntable, but there is no alignment which ensures the lens axis intersects the axis of rotation. Since it is essential for accurate tomographic reconstruction that the centre of the image be aligned with the centre of rotation, an experimental procedure is devised as follows: initially the camera is aligned by eye so that the axis of rotation of the foam sample is as near to the centre of the image as possible. Then, after all necessary video has been taken, the real centre of rotation is accurately calculated from the set of images and these are then cropped to give correctly aligned projections. Figure 5.8 shows how this is done. A point on the foam which can be seen at the left and right extremities of its arc is used to calculate the real centre line, which is just the midline between



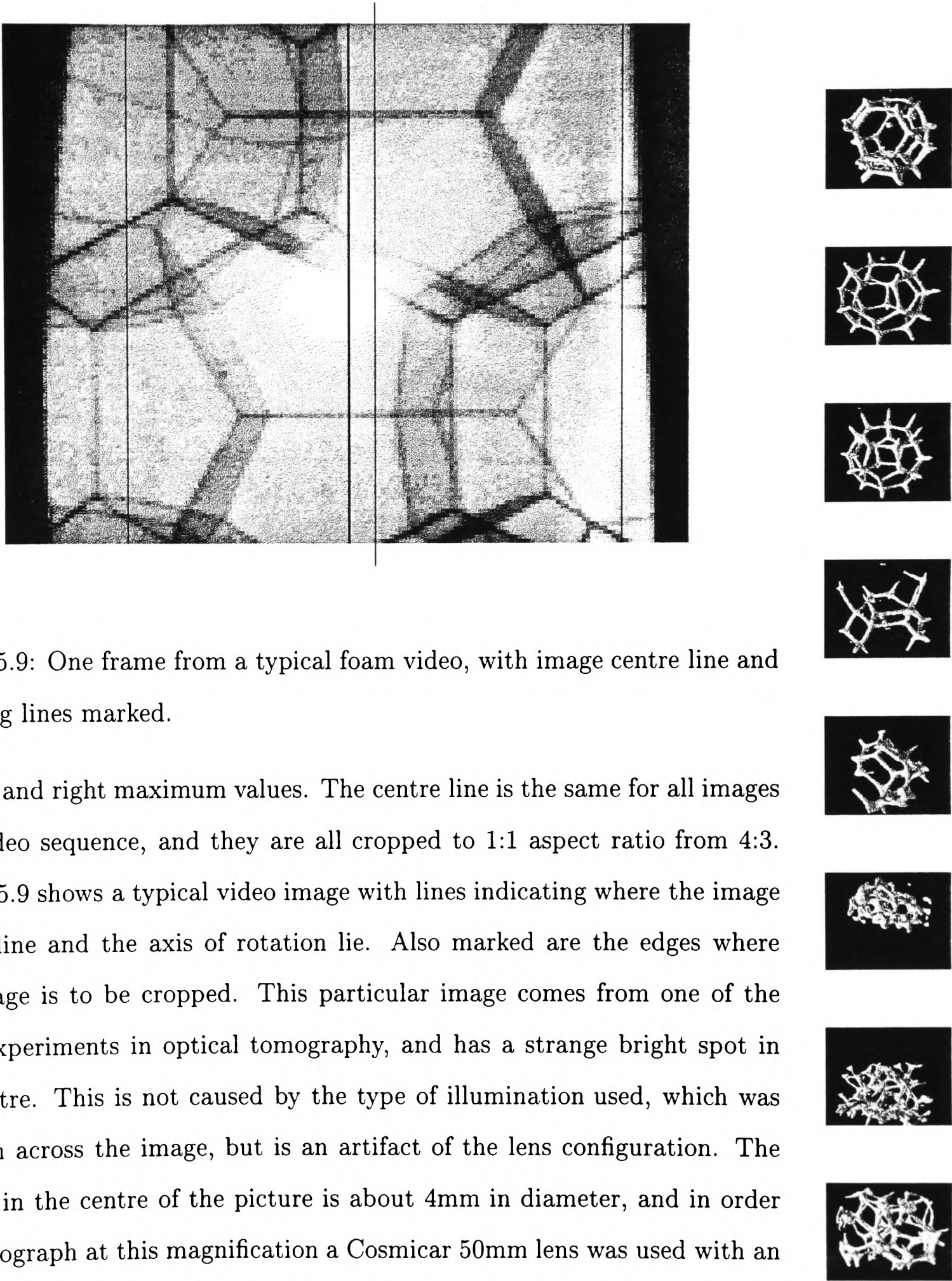


Figure 5.9: One frame from a typical foam video, with image centre line and cropping lines marked.

the left and right maximum values. The centre line is the same for all images in a video sequence, and they are all cropped to 1:1 aspect ratio from 4:3. Figure 5.9 shows a typical video image with lines indicating where the image centre line and the axis of rotation lie. Also marked are the edges where the image is to be cropped. This particular image comes from one of the early experiments in optical tomography, and has a strange bright spot in the centre. This is not caused by the type of illumination used, which was uniform across the image, but is an artifact of the lens configuration. The bubble in the centre of the picture is about 4mm in diameter, and in order to photograph at this magnification a Cosmimar 50mm lens was used with an extension tube of length 40mm. For some reason, this causes a bright spot

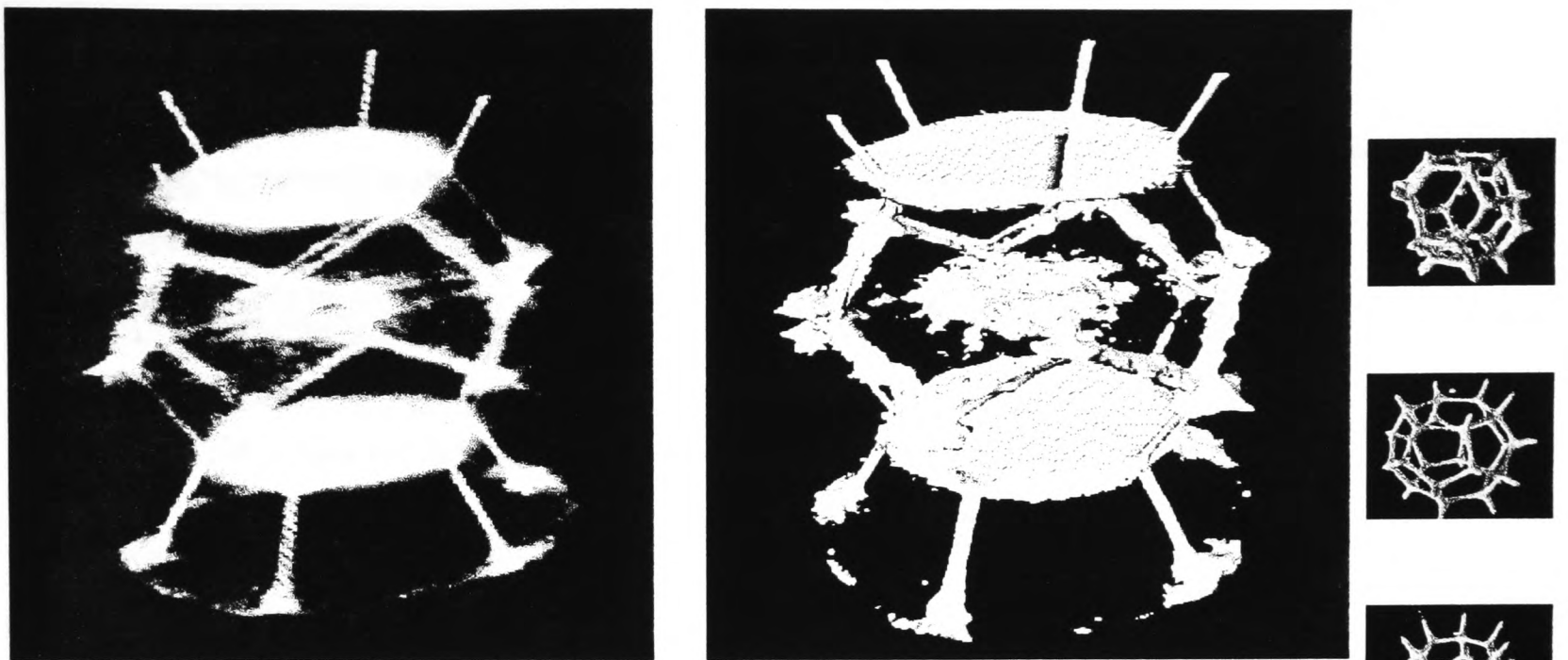


Figure 5.10: Bubble voxel model, viewed as a transparent solid (left), and as an opaque isosurface (right).

to appear in the centre of the image. This lens was later replaced with a newer Cosmincar 50mm lens and the problem did not occur.

The experimental rig described above does not work particularly well. Using the video clip, from which figure 5.9 comes, to reconstruct a solid model of the Plateau borders shows a flaw in the alignment of the camera. Figure 5.9 shows a dodecahedral bubble consisting of twelve pentagonal facets, where two of those facets are horizontal. These lie nearly parallel to the lens axis for the whole 360° camera arc, and it is therefore very difficult to determine their structure. Figure 5.10 shows two views of a tomographic voxel reconstruction of a dodecahedral bubble in an aqueous foam (details of the reconstruction methods will be presented later in this chapter). The horizontal bubble radius is approximately 4mm. Visualisation of volume models is always difficult, so here are presented two different types of three-dimensional model image.¹

¹Each of these images was produced with *Iris Explorer 2.2* running on a Silicon Graph-

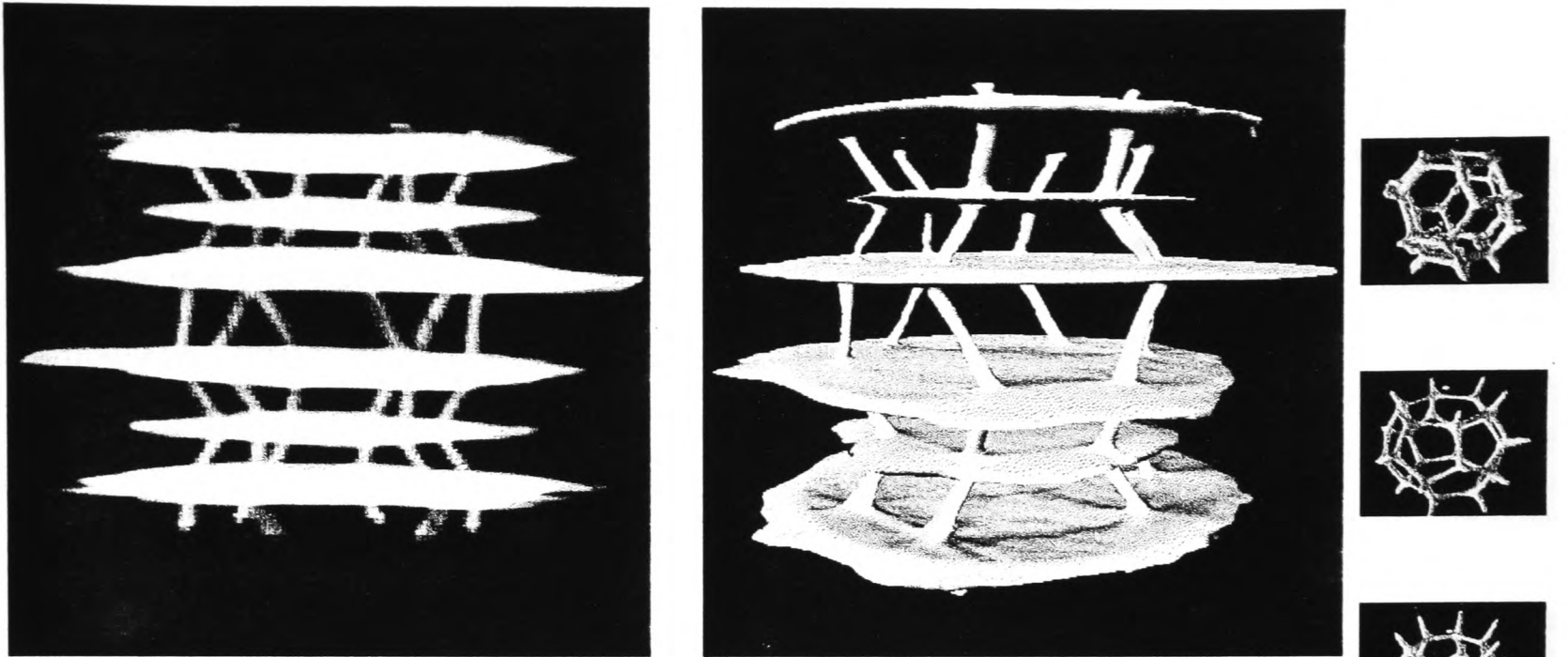


Figure 5.11: Voxel model of a Kelvin bubble, viewed as a transparent solid and as an opaque isosurface.

Firstly, treating each voxel as being partly transparent, it is possible to see the outlines of a cellular structure and to see how well ‘focussed’ the tomographic model appears. An alternative visualisation scheme is to choose a threshold voxel value and create an *isosurface* enclosing all voxels above threshold value. This is then rendered as an opaque solid.

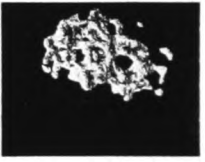
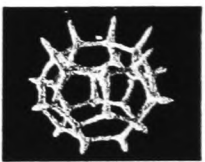
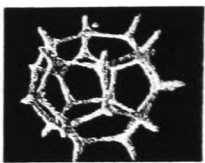
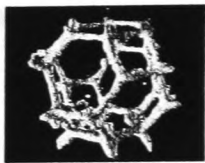
In this case, the reconstructed model is of poor quality, and although some of the more vertical Plateau borders have been resolved fairly accurately, the more horizontal ones are blurred or indistinguishable from noise. Also, the horizontal pentagonal facets appear solid, instead of hollow as they should be. Figure 5.11 is another example which shows the problems more clearly. In this figure of the reconstruction of an aqueous bubble resembling Kelvin’s suggested minimal shape [35, 36], the horizontal Plateau borders give rise to large horizontal plates which should not have been formed. The

ics Indy machine, as were all other three-dimensional images in this chapter.

intention is to resolve Plateau borders only, and not foam lamellae. There is a straightforward solution to these problems, which is that the camera should be angled away from perpendicular to the axis of rotation.

This is an example of the sampling problem which occurs in *cone-beam* tomography, and it is further complicated by the fact that the optical system used here does not have a continuous Radon transform. In both the cases of solid and liquid foams, the shadow cast by the Plateau borders is almost completely dark. With the solid foams, this is because the Plateau borders are totally opaque, whereas the liquid foam Plateau borders, although highly transparent, scatter light so well that none passes through to the camera. For X-ray tomographic scanners, the attenuation of radiation depends on the amount and density of material a ray passes through before reaching a detector. This gives added information which is not available if rays are either attenuated completely or not at all, giving a binary Radon transform.

The effect of varying the angle of the camera to the horizontal can be seen in figure 5.12. A phantom target was made from a piece of bent solder wire and then filmed from six different camera elevations, capturing about 140 images for each elevation. A reconstruction is made in each of the six cases and these are then rendered from the same viewpoint. Clearly, at low elevations the wire voxel models suffer from the same problem as the foam ones. The wire has been elongated in the horizontal plane, and the lower loop has become one flat surface. At angles of 30° or so, the distortion is much less, and the correct structure of the phantom can be seen. The elevation of 40° seems to give the best reconstruction; however, at this angle the view of the camera is restricted by the turntable and it is virtually impossible



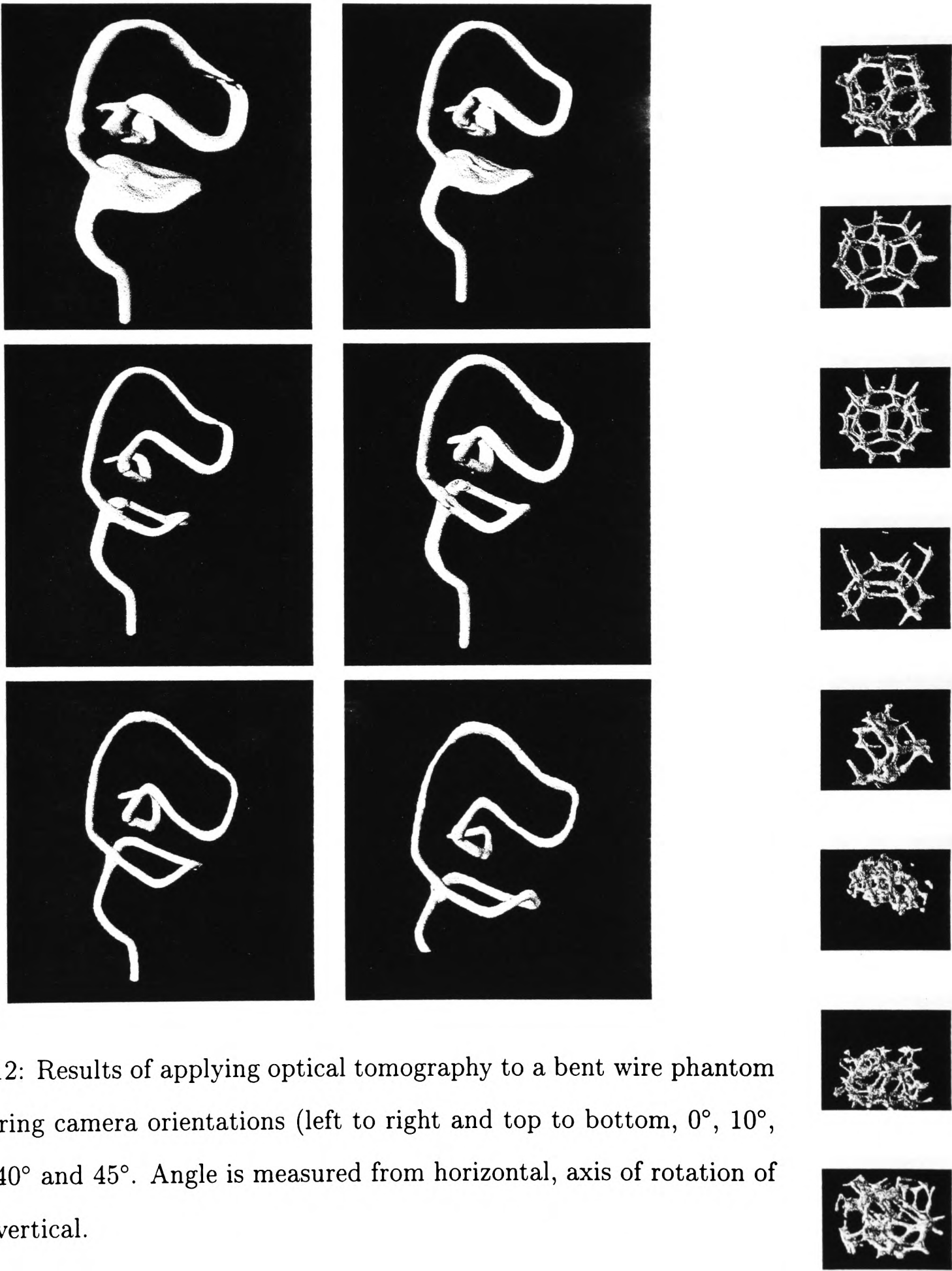


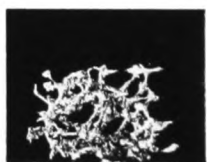
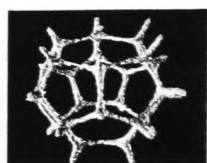
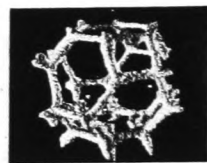
Figure 5.12: Results of applying optical tomography to a bent wire phantom with differing camera orientations (left to right and top to bottom, 0°, 10°, 20°, 30°, 40° and 45°. Angle is measured from horizontal, axis of rotation of object is vertical.

to illuminate the subject properly from behind. Because of this, the foam experiments were carried out with a camera elevation of 30° , as this gives good reconstruction accuracy and allows even lighting of a large section of foam.

The choice of scanning geometry for cone-beam tomography is a compromise between reconstruction accuracy and simplicity of mechanical design. Many commercial medical scanners avoid the problem by using multiple two-dimensional tomographic slices to form a three-dimensional image. In two dimensions, the Radon transform in the plane is *complete* if an infinite number of projections of infinite resolution are taken at all angles around the target. An approximation to this is easily made by using many high-resolution projections on a circular arc. In three dimensions, however, the complete Radon transform is an infinite set of planar projections taken at all angles on a sphere around the target. It is difficult to design a mechanical system which can provide this, so approximations are made which can mean that the Radon transform is not mathematically complete and reconstruction accuracy will be compromised. Smith explains the details [86, 87, 88] and provides complex algorithms for several scanning geometries.

In the interests of mechanical simplicity, this project uses only the circular scanning geometry described above. Fortunately, despite the theoretical mathematical shortcomings of this system, useful results are still obtainable.

Further complications arise in the ratio of emitters and detectors best suited to efficient reconstruction. Natterer [89] suggests that the number of emitter positions in a two-dimensional fan-beam tomography system should greatly exceed the number of detectors per projection for efficient recon-



struction. In the three-dimensional optical case, this corresponds to a large number of low resolution images. In practice, limiting the resolution of a projection image can mean that the fine Plateau border filaments of a foam are not resolved. A high enough resolution is required that this is not the case, and since overall video bandwidth is restricted, the maximum number of images which may be sampled is less than optimal.

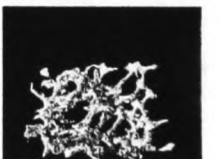
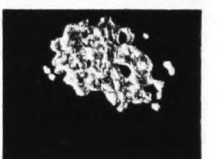
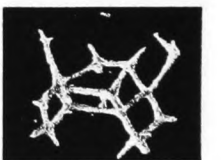
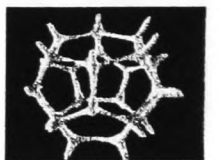
Fast reconstruction algorithms for two-dimensional tomography are well-documented, for example Peters [90]. The more complex three-dimensional problem is proving more difficult to implement in software. Feldkamp *et al.* [91] produce an approximate algorithm, noting that at that time no commercial scanners were known to use direct three-dimensional methods.

The algorithm presented in the next sections differs from those mentioned above by having a complexity based on the number of voxels, rather than the number or resolution of projections. It is not exact, but since the scanning geometry prevents an exact reconstruction this is not considered important. The software is reasonably efficient, bearing the above in mind.

5.6.1 Reconstruction of solid foams

The description of the reconstruction algorithms is divided into two categories, solid and liquid foams. The difference between the two is the value of the gas fraction in each type of foam. Solid foams in this project have a lower gas fraction than the liquid foams, and this simplifies the reconstruction process slightly.

Some examples of the foams are shown in figure 5.13. These are open polyurethane foams, and hence have no lamellae between Plateau borders.



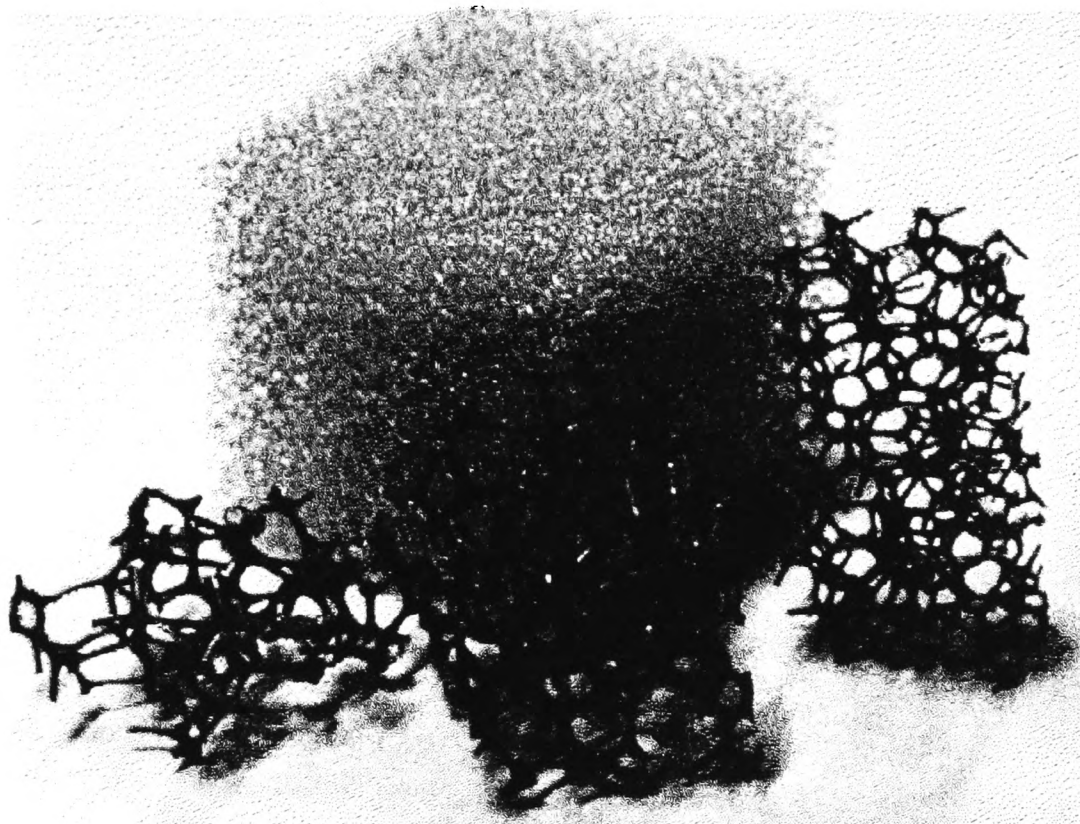
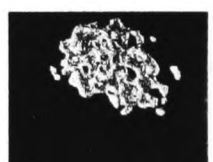
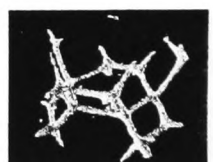
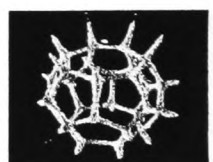
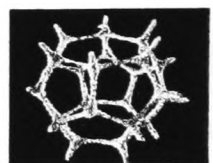


Figure 5.13: A selection of solid, open polyurethane foams.

Because of this, these foams are well suited to analysis by optical tomography, as light passes through several layers of bubbles without much overall attenuation.

For accurate reconstruction, the position of the camera relative to the foam must be known. The details are shown in figure 5.14. Angle a is given accurately by a scale on the camera rig. The height of the CCD matrix, h , is taken from the camera documentation to be 6.6mm. The quantity $D + d$ is measured with a ruler. This is roughly 300mm, depending on the magnification required to resolve individual bubbles in the foam sample. The distance H , the image height, is found by putting a ruler in front of the camera instead of the foam and measuring the size of the image directly.

The values of D and d can easily be calculated as they are in the same ratio as H and h , since all four variables are on the sides of similar triangles. Given



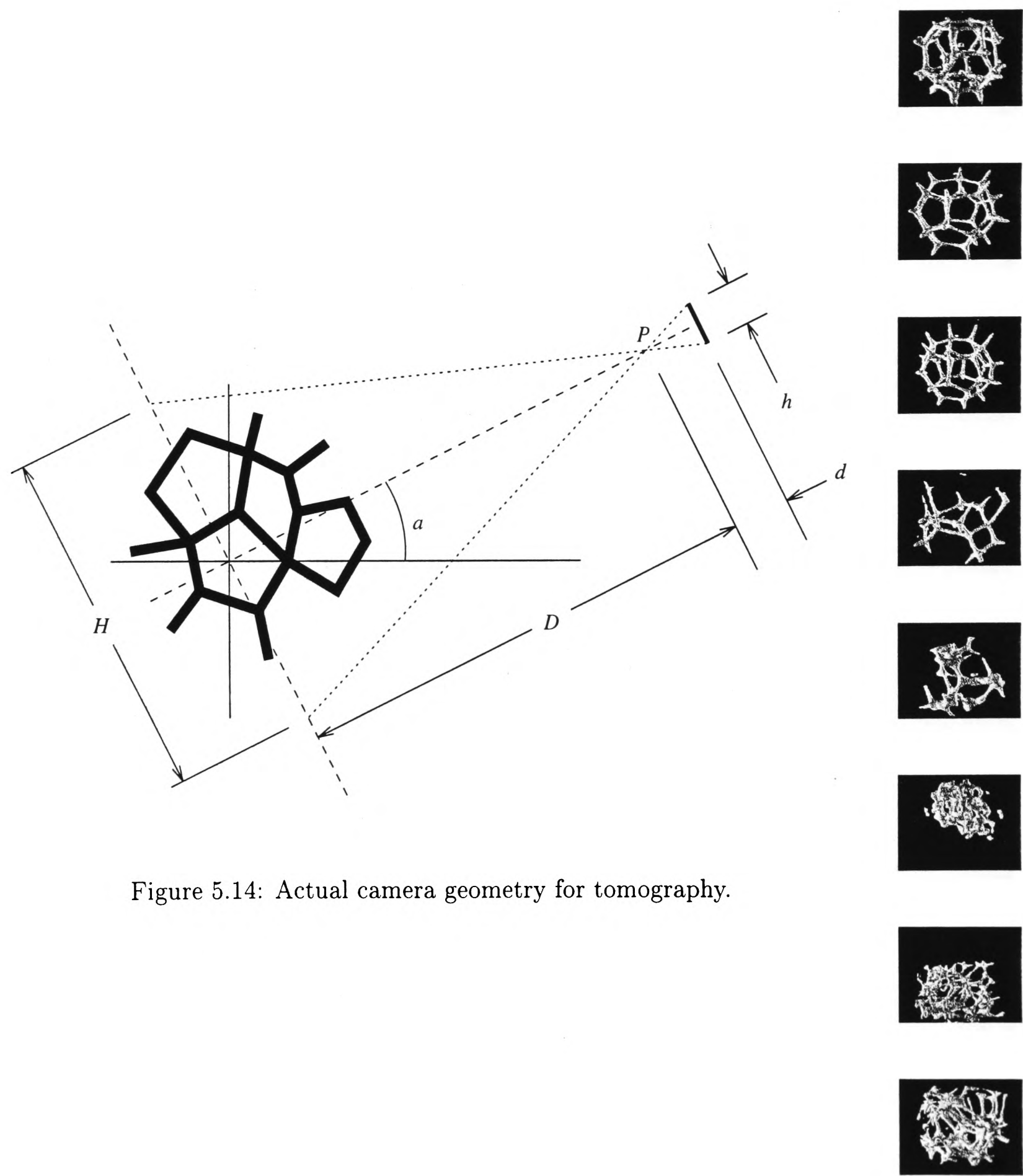
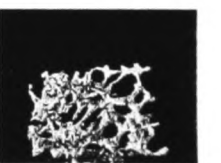
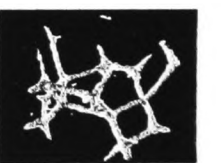
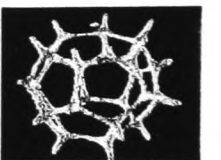
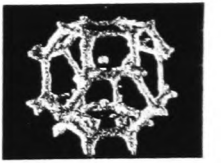


Figure 5.14: Actual camera geometry for tomography.

D and a , the location of the point P is known. Successful reconstruction depends on knowing a , D and H , and also the azimuthal angle of rotation of the foam sample for each projection frame. The video system is assumed to be sampling frames at a regular rate; this is impossible to guarantee in computer terms without using a real time operating system. Microsoft Windows 3.11 and the Captivator software provide no indication of the elapsed time between sampled images, although if a frame cannot be digitised quickly enough and it is skipped, the software is supposed to inform the user of this. In practice, Video for Windows version 1.0 sometimes skips frames quietly. The only way to be sure this has not happened is to go through the digitised video and check for duplicate frames, since if a frame is skipped the software copies the previous frame to the output file a second time.

To determine the angle the turntable rotates between frames, the total number of frames for one revolution are counted and divided into 360° . In order to make this as reliable as possible, all video was digitised to a buffer in memory first, then spooled to disk during moments when the processor was idle. Knowing the inability of Windows 3.11 to multitask efficiently, the accuracy of this procedure can only be guessed at. In general, recording at a given frame rate usually gave a variation of less than one percent in the number of frames required to store one revolution of the turntable. This margin of error could have been caused by a slightly binding bearing in the turntable hub. However, the error in frame rate was larger, such that the system never delivered the exact frame rate specified by the user.

The unit used internally by the reconstruction algorithm is the width of one pixel. Since the image height is known both in pixels and millimetres, a



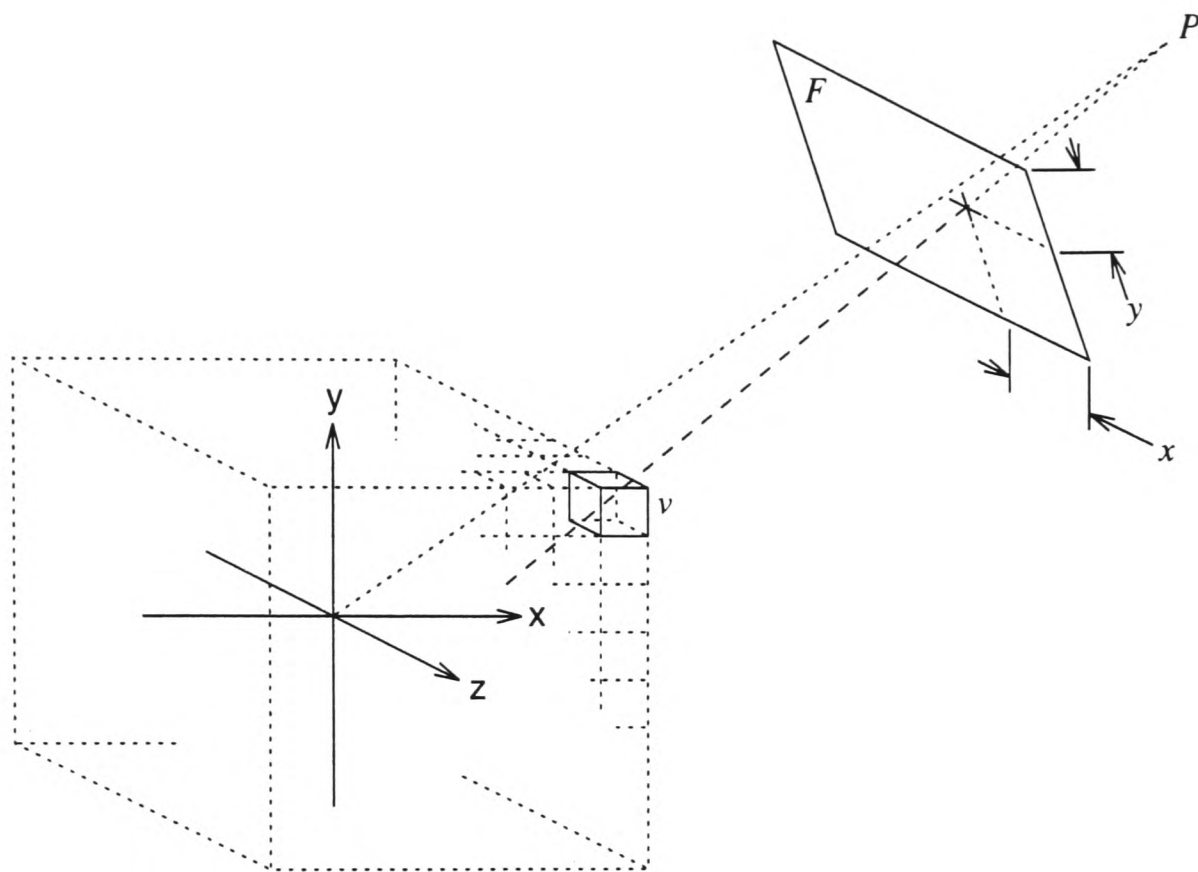
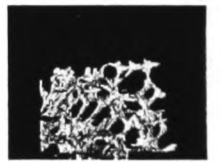
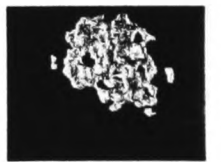
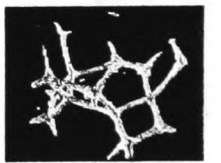
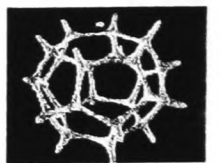
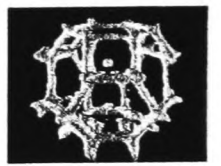


Figure 5.15: Solving for x and y gives the value of the pixel in frame F , and hence the intensity for a ray passing through F to a voxel v .

conversion between the two units is possible. The distance along the side of a voxel is also one unit.

The reconstruction algorithm implements the summation of all ray samples through each model voxel. It is assumed that the foam stays fixed in space and the camera rotates in a 360° circular arc around it. Figure 5.15 shows how a reasonably efficient algorithm might work. For every voxel in the model, the intersection of a ray passing through it and the point P previously calculated with a frame F of the video gives the pixel value associated with that voxel. Since the pixel value represents the attenuation of light through the real foam at that point, inverting it by subtracting from 255, the maximum value, gives a probability of the voxel being part of a foam Plateau



border. This value is added to the initial voxel value (zero) and the process is repeated for every frame. Hence, for a video sequence of 100 frames, the final value of the voxel v will be the sum of all the 100 probabilities derived from the frames with suitable normalisation to prevent overflow. A simple threshold, its value chosen depending the data, can then be used to decide if a voxel should represent gas or part of the foam Plateau border network.

The reconstruction system is most concisely and unambiguously described by a Z specification. The structure used to store the tomographic model is a voxel set, which is a sequence of images containing sequences of rows and columns, and is represented by the types,

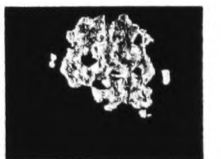
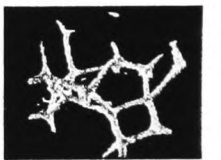
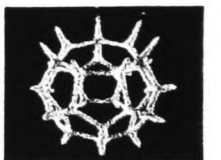
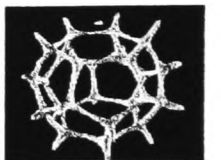
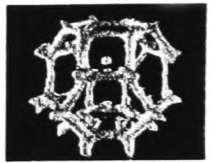
$$VOXEL == BYTE$$

$$VOXELSET == \text{seq seq seq } VOXEL$$

The dimension of the voxel set is d_v , so it contains d_v^3 voxels arranged in a cube. There is also a sequence of images, in similar format to that in chapter four, $\text{seq } RASTER$. These images are inverted to form projections for use in the reconstruction by,

$\begin{array}{l} \text{Invert} \\ \hline \text{dataset?}, \text{dataset!} : \text{seq } RASTER \\ \hline \forall n : \text{dom dataset?}; x : \text{dom dataset?}(1); y : \text{dom dataset}(1)(1) \bullet \\ \quad \text{dataset!}(n)(x)(y) = 255 - \text{dataset?}(n)(x)(y) \end{array}$

After this operation, each pixel in the image has a value in the range 0..255 where higher numbers mean an increased likelihood of foam material along the ray through the pixel.



Several parameters are required for the tomography software, as well as d_v ,

$n_i : \mathbb{N}_1$ the number of projection images

$x_i : \mathbb{N}_1$ the image width (pixels)

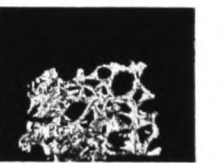
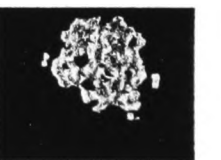
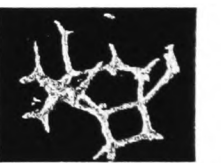
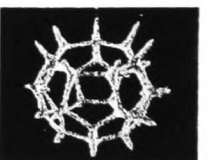
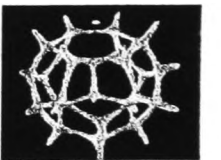
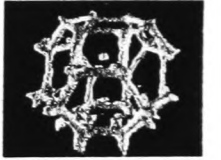
$y_i : \mathbb{N}_1$ the image height (pixels)

$\alpha_{cam} : \mathbb{R}$ the camera elevation (degrees)

The trigonometric functions *Sin* and *Cos* are assumed to exist without needing specification, other than being of type $\mathbb{R} \rightarrow \mathbb{R}$. From these functions, the implementation of the rotation of a point about an axis can be defined. In particular, functions are required which describe the rotation of an arbitrary point about the x and y axes, and these will operate with real coordinates. An advantage of the small depth of field method described in chapter four is that it uses integer arithmetic throughout. The tomographic method requires floating point arithmetic, because of the trigonometric transformations involved, and hence executes more quickly on a processor with fast hardware floating point arithmetic acceleration. The definition of a real three-dimensional coordinate type is the set of tuples of the form,

$$VECT == \mathbb{R} \times \mathbb{R} \times \mathbb{R}$$

And from this, the specifications for rotation functions follow, where each function takes the coordinates of a point in space and an angle through which to rotate the point,



$$rotX, rotY, rotZ : VECT \times \mathbb{R} \rightarrow VECT$$

$$\forall x, y, z, \theta : \mathbb{R} \bullet$$

$$rotX((x, y, z), \theta) =$$

$$(x, y * Cos(\theta) - z * Sin(\theta), z * Cos(\theta) + y * Sin(\theta))$$

$$rotY((x, y, z), \theta) =$$

$$(x * Cos(\theta) - z * Sin(\theta), y, z * Cos(\theta) + x * Sin(\theta))$$

$$rotZ((x, y, z), \theta) =$$

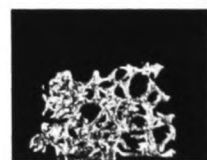
$$(y * Cos(\theta) - x * Sin(\theta), y * Cos(\theta) + x * Sin(\theta), z)$$



Initially, the voxel set lies with its sides aligned with the x, y and z axes. The camera axis has azimuth and elevation offset angles relative to the z axis initially. An efficient way to proceed is to rotate the camera axis so that it lies along the z axis, and this requires a rotation about the y axis followed by a rotation about the x axis. The same transformations are applied to each voxel in turn, and it is then a simple matter to determine the intersection of the line joining the voxel and point P with the projection plane, which is now parallel to the xy plane and centred on the z axis. Figure 5.16 shows the situation after both rotations have been applied. It is a simple matter to find x and y by similar triangles, and hence the value which should be added to voxel v from the frame F .



There is a function $RTransform$, which applies both rotations to a real coordinate,



$$RTransform : VECT \times \mathbb{R} \times \mathbb{R} \rightarrow VECT$$

$$\forall c : VECT; \theta_x, \theta_y : \mathbb{R} \bullet$$

$$RTransform(c, \theta_x, \theta_y) = rotX(rotY(c, \theta_y), \theta_x)$$



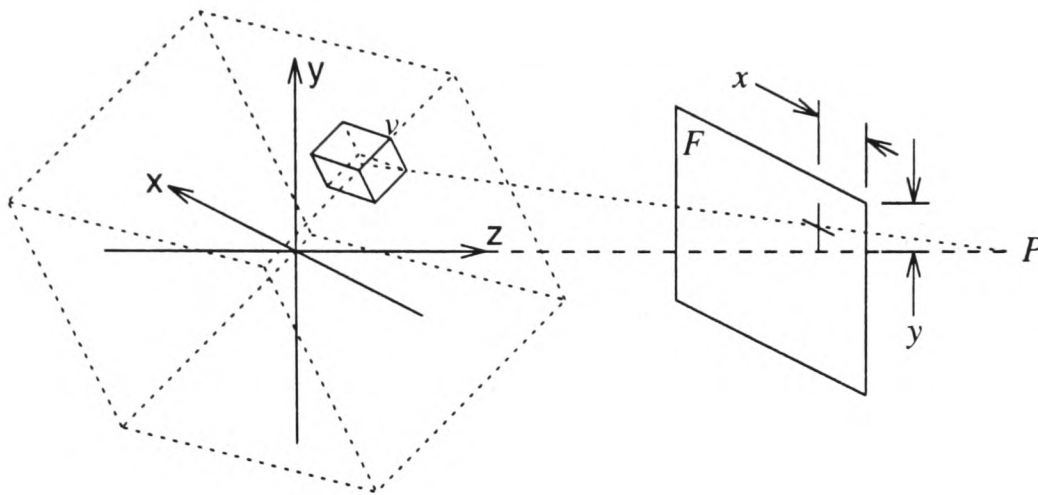
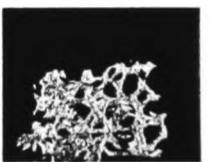
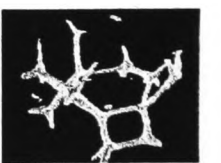
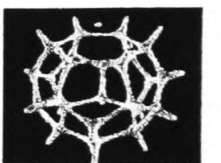


Figure 5.16: Voxel and frame position after rotations.

The order of rotation is important. First the azimuthal rotation about the y axis is applied so that the camera axis is in the yz plane, then the x axis rotation aligns this axis with the z axis.

Given the constraint mentioned previously that the individual voxel side length is also the width of a pixel in F , the coordinates, relative to the axes of F , of the intersection of a ray through P and a given coordinate with F are found by applying the function *Intersect*. There is a problem with the specification of this function, as it seems to require a *type cast*, as indeed does the C language implementation. Type casting is the conversion of a data item of one type into another, usually done in software because storage requirements are different for each type. In Z, type casting is necessary to satisfy the strongly-typed nature of the language [56]. Z makes no provision for changing the types of variables in an expression, which is sometimes necessary in the description of a software module. In particular, *Intersect* needs to truncate real values to form integers. This is done by using $()$ as the cast operator,² as in C, with the type of the expression between the

²Z does not ordinarily support overloading of operators either.



parenthesis. Although this may seem improbable in $\mathbb{Z}$, the definition of a real to integer³ cast is straightforward,⁴

$$\begin{array}{|l}
 r2z : \mathbb{R} \times \mathbb{Z} \rightarrow \mathbb{Z} \\
 \hline
 \forall r : \mathbb{R}; z : \mathbb{Z} \bullet \\
 \quad \text{if } -1 < r < 1 \text{ then } r2z(r, z) = z \\
 \quad \text{if } r \geq 1 \text{ then } r2z(r, z) = r2z(r - 1, z + 1) \\
 \quad \text{if } r \leq -1 \text{ then } r2z(r, z) = r2z(r + 1, z - 1)
 \end{array}$$

Then the typecast operator to integers uses $r2z$ as follows,

$$\begin{array}{|l}
 (\mathbb{Z})_- : \mathbb{R} \rightarrow \mathbb{Z} \\
 \hline
 \forall r : \mathbb{R} \bullet (\mathbb{Z})r = r2z(r, 0)
 \end{array}$$

The definition of *Intersect* follows,

$$\begin{array}{|l}
 Intersect : VECT \rightarrow \mathbb{Z} \times \mathbb{Z} \\
 \hline
 \forall x, y, z : \mathbb{R} \bullet \\
 \quad Intersect(x, y, z) = \\
 \quad \left((\mathbb{Z}) \left(\frac{x}{1 + \frac{z}{D}} \right) + x_i \text{ div } 2, (\mathbb{Z}) \left(\frac{y}{1 + \frac{z}{D}} \right) + y_i \text{ div } 2 \right)
 \end{array}$$

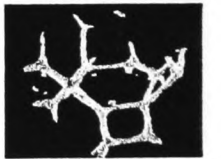
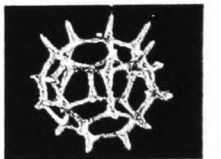
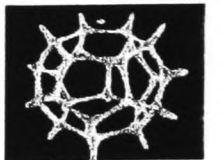
So, this function returns the offset, in image space coordinates, of the pixel associated with the voxel at (x, y, z) .

Bringing all these functions together, the complete tomographic part of the reconstruction algorithm can now be defined,⁵

³ $(\mathbb{Z})$ rounds towards zero.

⁴Note that $\{0, 1\} \in \mathbb{Z} \wedge \{0, 1\} \in \mathbb{R}$.

⁵There is a shorthand notation here where if there is a tuple $p : (\mathbb{N} \times \mathbb{N}) = (a, b)$, then $dataset?(f)(a)(b) = dataset?(f)(p)$.



TomographicReconstruction

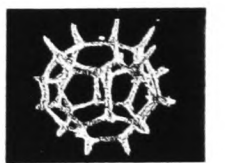
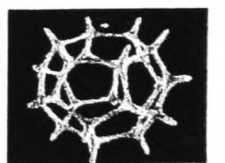
dataset? : seq *RASTER*
vmodel! : seq seq seq *VOXEL*

 $\forall x, y, z : \mathbb{N}_1 \mid x \leq d_v \wedge y \leq d_v \wedge z \leq d_v \bullet$
 $vmodel!(x)(y)(z) =$

$$\sum_{f=1}^{n_i} dataset(f)(Intersect($$

$$RTransform(x - d_v \text{ div } 2, y - d_v \text{ div } 2, z - d_v \text{ div } 2),$$

$$\alpha_{cam}, f * 360/n_i))$$



TomographicReconstruction outlines the transformations required for the correct reconstruction of the tomographic data. The resulting *vmodel!* is a three-dimensional array of eight-bit density values, to which a threshold can be applied to distinguish the polyurethane foam material from the surrounding ‘air.’ The complete reconstruction system, going from video stream to solid model is,

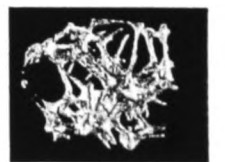
SolidTomography

Invert >> *TomographicReconstruction*



5.6.2 Reconstruction of liquid foams

Liquid/gas mixtures form closed foams, which may have either a high gas fraction and be polygonal, or a low gas fraction and be bubbly. The polygonal foams are better suited to analysis by optical tomographic means, since the cell lamellae are very thin compared to the Plateau border thickness, and almost completely transparent. The structure of foams of low gas fraction

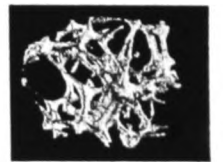
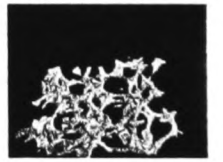
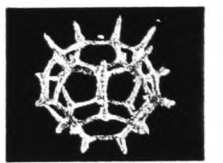
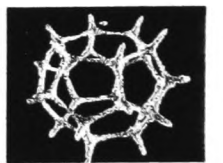


is more like that of spherical or elliptical bubbles in liquid, and the absence of thin lamellae means that virtually no light rays can pass through without being diffracted greatly. Since this optical tomography system depends on light rays passing through the foam sample in straight lines, low gas fraction foams are not easily studied with this method.

A problem with the dry polygonal foams is that, although the cellular structure is clear, individual Plateau borders are very thin (possibly tens of microns or less) and a typical CCD array does not have high enough resolution to properly resolve these thin strands of liquid. The effect is that what should be a crisp dark line in a video frame becomes a blurred grey line in the digital image. This effect can be seen in figure 5.9, where the already low resolution of the image is reduced by a greying and blurring of lines.

The ideal image which would be suitable for use as a tomographic projection would be a set of dark shadows from the Plateau borders on a uniform light background. Unfortunately, even with very thin liquid foams, the liquid lamellae can absorb a significant amount of light if they are oriented nearly parallel to the lens axis. This is also visible in figure 5.9. Lamellae which are almost ‘edge on’ to the camera appear grey and somewhat opaque, rather than completely clear. If something is not done to fix this, the lamellae will adversely affect the reconstruction, giving the appearance of liquid where none is intended.

A solution to both problems is to use the edge enhancement routines described in chapter four. Firstly, this has the effect of sharpening grey, defocussed lines into darker, sharper edges. But also, because it acts a bit like a two-dimensional differentiator, the darker lamellae are brightened up



to match the rest of the background.

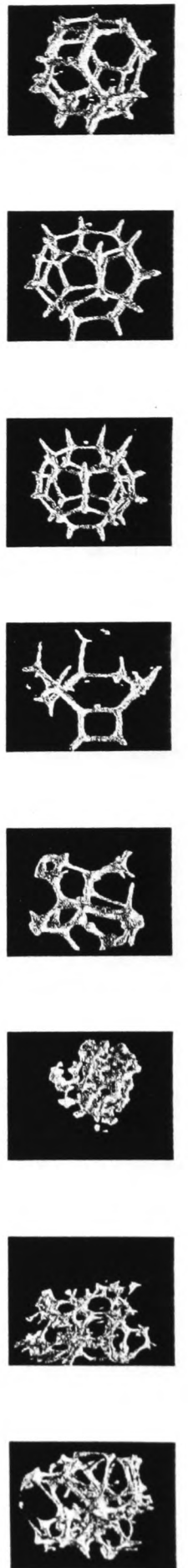
5.7 Results

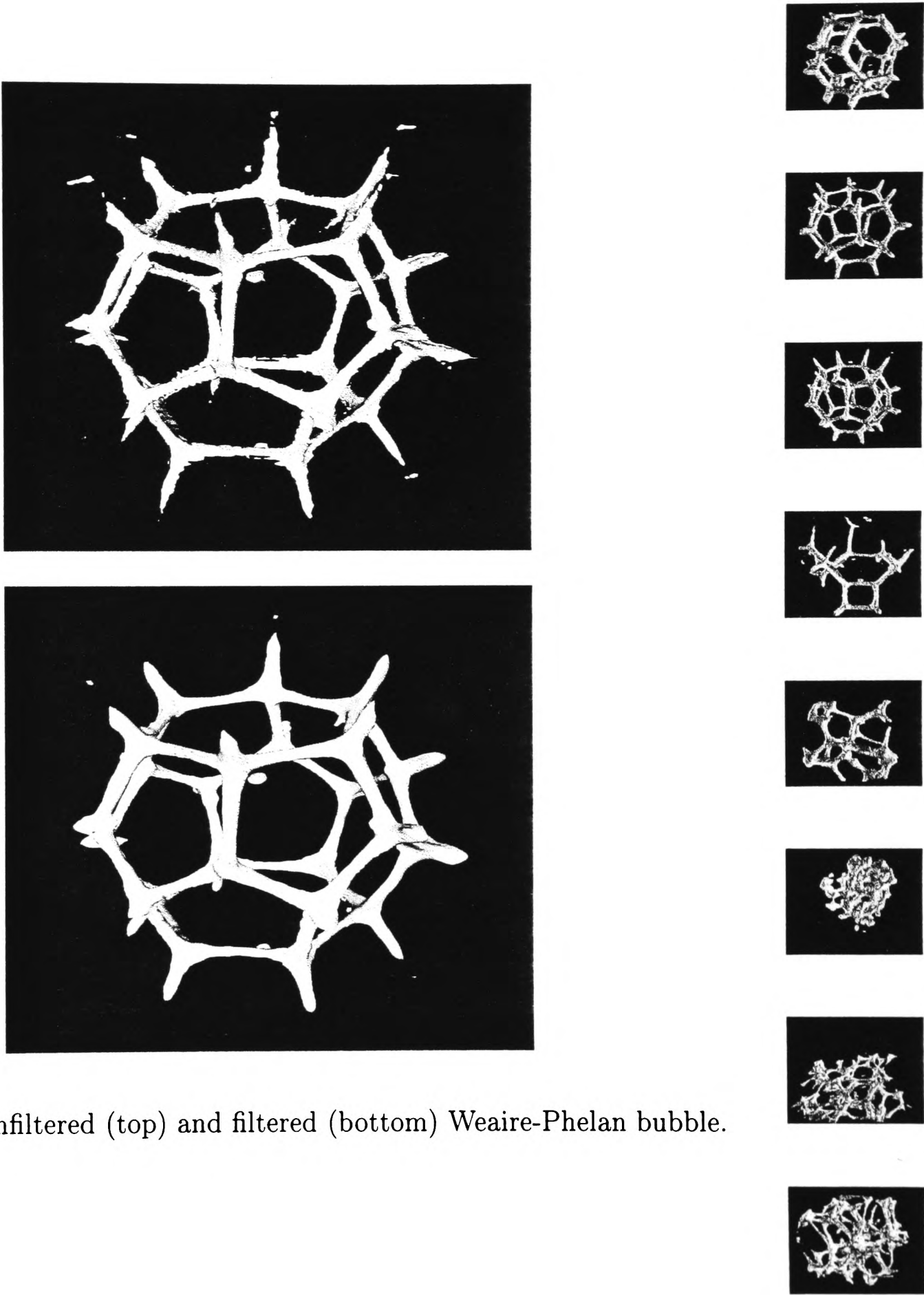
The voxel models benefit from a low-pass filtering operation, as can be seen from figure 5.17. There are high spatial frequency noise artifacts in the reconstructed model but these are greatly diminished after filtering. The filter is a simple 27-neighbourhood averaging kernel which is applied in one pass to the voxel model. More passes only blur the data whilst adding no improvement.

The geometry of the Weaire-Phelan bubble [40] is clearly visible in figure 5.17. The regular 14-sided bubble has two hexagonal faces and twelve pentagonal faces. These monodisperse bubbles form a column in the foam tube such that they are surrounded by a single layer of bubbles next to the tube wall [92]. If the bubble volume is changed by a small amount, other regular bubble columns may be formed such as those from the bubbles shown in figure 5.18.

In figure 5.18 the bubble volume has been reduced, which means that more bubbles will fit into the tube. The 16-sided bubble is surrounded in the horizontal plane by fourteen bubbles, whereas the Weaire-Phelan bubble has twelve horizontal neighbours. The extra bubbles mean that the 16-sided shape has seven-sided horizontal faces and fourteen pentagonal faces. This leads to a greater distortion of the perimeter of the horizontal faces, which are planar, in order for the local Plateau border angle constraint to be satisfied. The extra curvature can be seen on the model.

Figure 5.18 also shows how the liquid threshold value affects the voxel





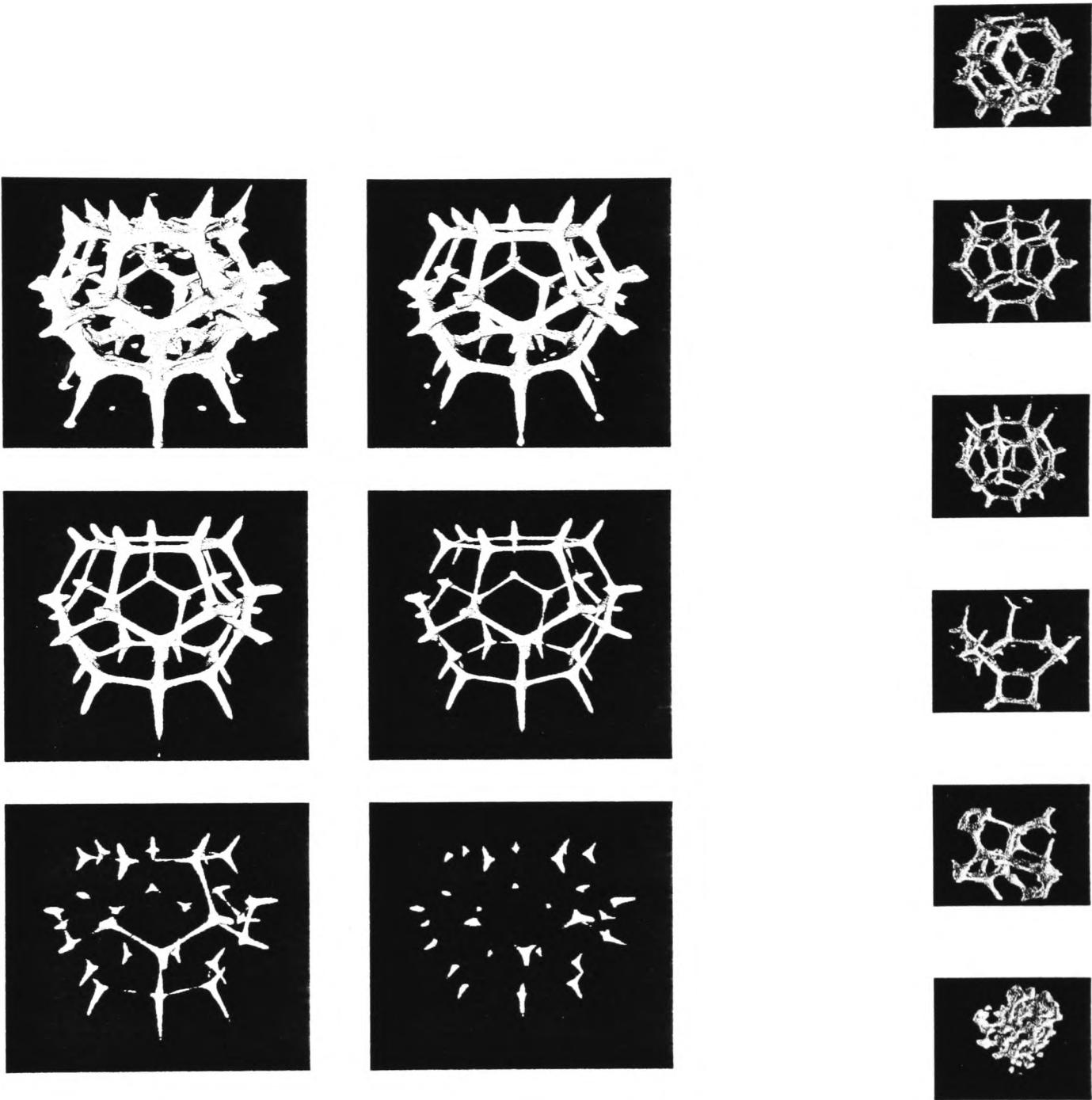
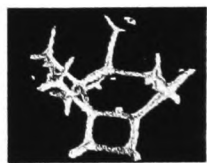
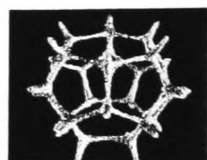


Figure 5.18: Varying thresholds for a 16-sided bubble model. Threshold values are (in rows, top to bottom) 90, 100, 110, 120, 130 and 140.

model. The tomography experiment produces a continuous voxel result where varying the threshold changes the liquid content of the model. Tomographic models by their nature can never be black and white, but can only indicate grey levels. The high end medical tomography scanners employ very sophisticated image processing techniques in order to produce useful images of tissue sections, but the raw tomographic data must be good enough for image processing to be worthwhile. In these foam models, there is no simple algorithm for determining the liquid threshold value automatically, however, if the foam density were known, the calculation of the correct threshold value would be trivial. In these experiments, the best situation is to scale the reconstruction so that a 'good' threshold value is somewhere near the middle of the 0..255 voxel value range.

Several magnified images of a Plateau border junction at different threshold values are shown in figure 5.19. It is clear that the resolution of the voxel model at this level of detail is inadequate, and it is not reasonably possible to infer the exact junction geometry or even radii of curvature from the model. This should not be surprising, though, as the features are so small that very few voxels are needed to represent the curved surfaces. It is also true that if a much higher magnification were used (for example, a powerful microscope) to create pictures of the Plateau border, the tomography becomes less useful as a means of analysing liquid structures. This is because the scattering of light by the Plateau border is unpredictable, and only on a macroscopic scale does this produce a uniform dark shadow. At the microscopic level, some light will almost certainly pass through the Plateau border and be refracted towards the camera, producing a lighter area in the image than should be



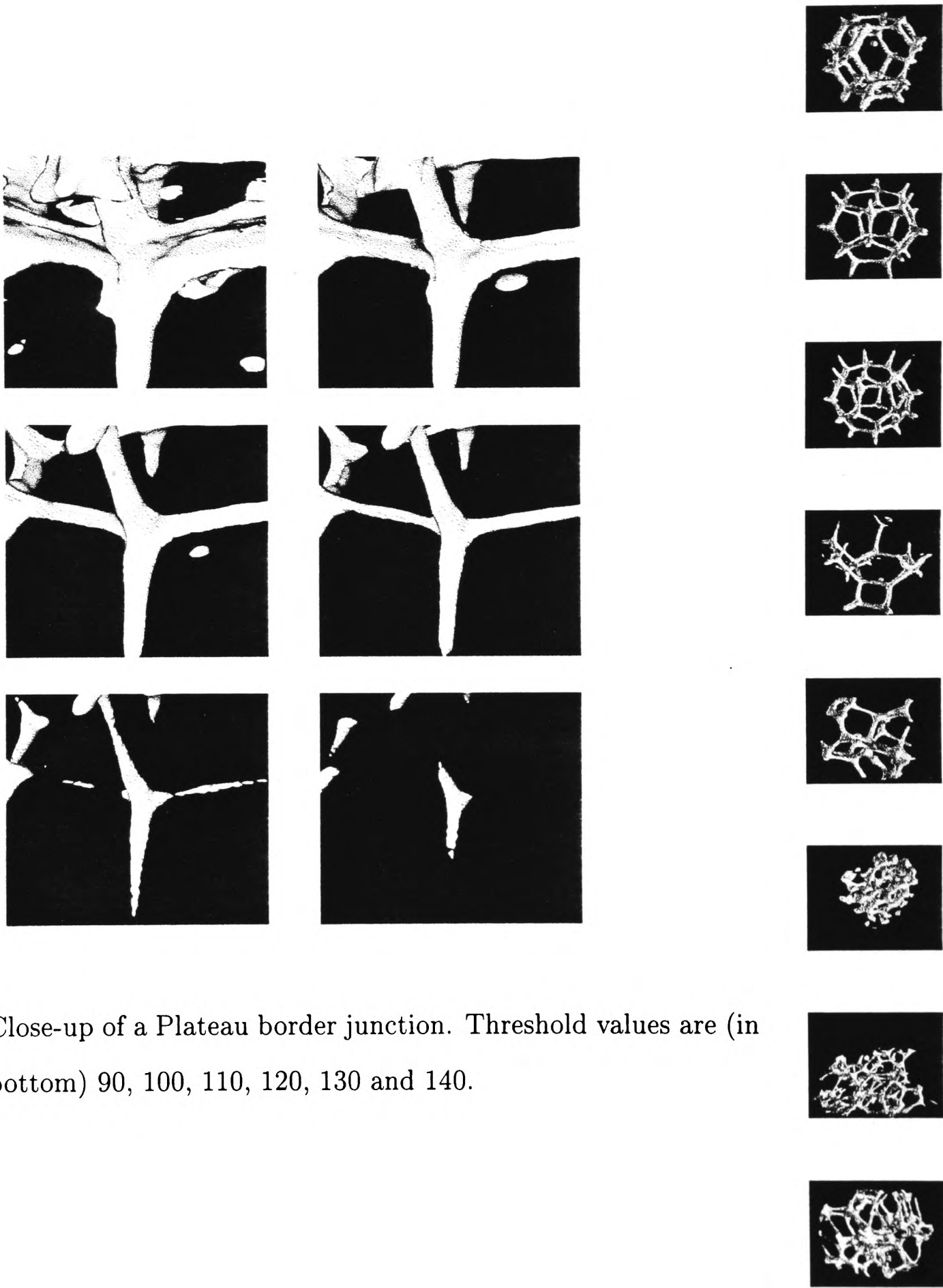


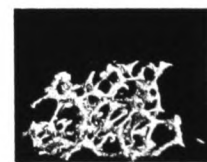
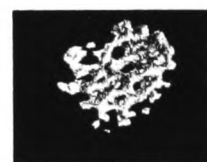
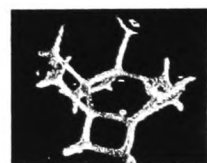
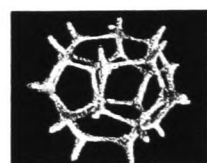
Figure 5.19: Close-up of a Plateau border junction. Threshold values are (in rows, top to bottom) 90, 100, 110, 120, 130 and 140.

predicted by a simple application of the Radon transform.

Whilst the voxel models lack resolution, plenty of detail is captured by tomography. Figures 5.20 and 5.21 demonstrate this well. From reconstructions of aqueous Kelvin bubbles (figure 5.20) it is possible to see the slight curvature of the Plateau borders which Kelvin described in his paper. The quadrilateral faces are planar but have their edges bowed outwards by surface tension forces—this is visible in the model.

The solid foam in figure 5.21 shows a very good approximation of a portion of a polyurethane foam. There is a broken Plateau border on the lower right part of the model. This is an accurate representation of the real foam which also has a broken Plateau border there. Since many uses of solid foams depend on foam structural properties, a more powerful version of the tomography system could help to find weaknesses in foams caused by wear and tear or faulty manufacturing processes. If the foam contains many incomplete or broken links it will be less mechanically strong and resilient. Tomography could help in the quality assurance process.

Figures 5.22, 5.23 and 5.24 give an idea of the limitations of this system. The solid foam tomography has no limitations on data gathering as the foams are clearly motionless. With any system, though, it is best to capture approximately the same number of frames as the dimension of the image or voxel set, as described earlier. Using a frame size of about 256×256 with 256 frames, it is possible to reconstruct a voxel cube of side 256, or sixteen million voxels in total. For a target volume of about 64mm^3 , a single voxel represents a volume of $3.8 \times 10^{-15}\text{m}^3$. The three figures represent a rough limit for foam complexity with this amount of data and the reconstruction



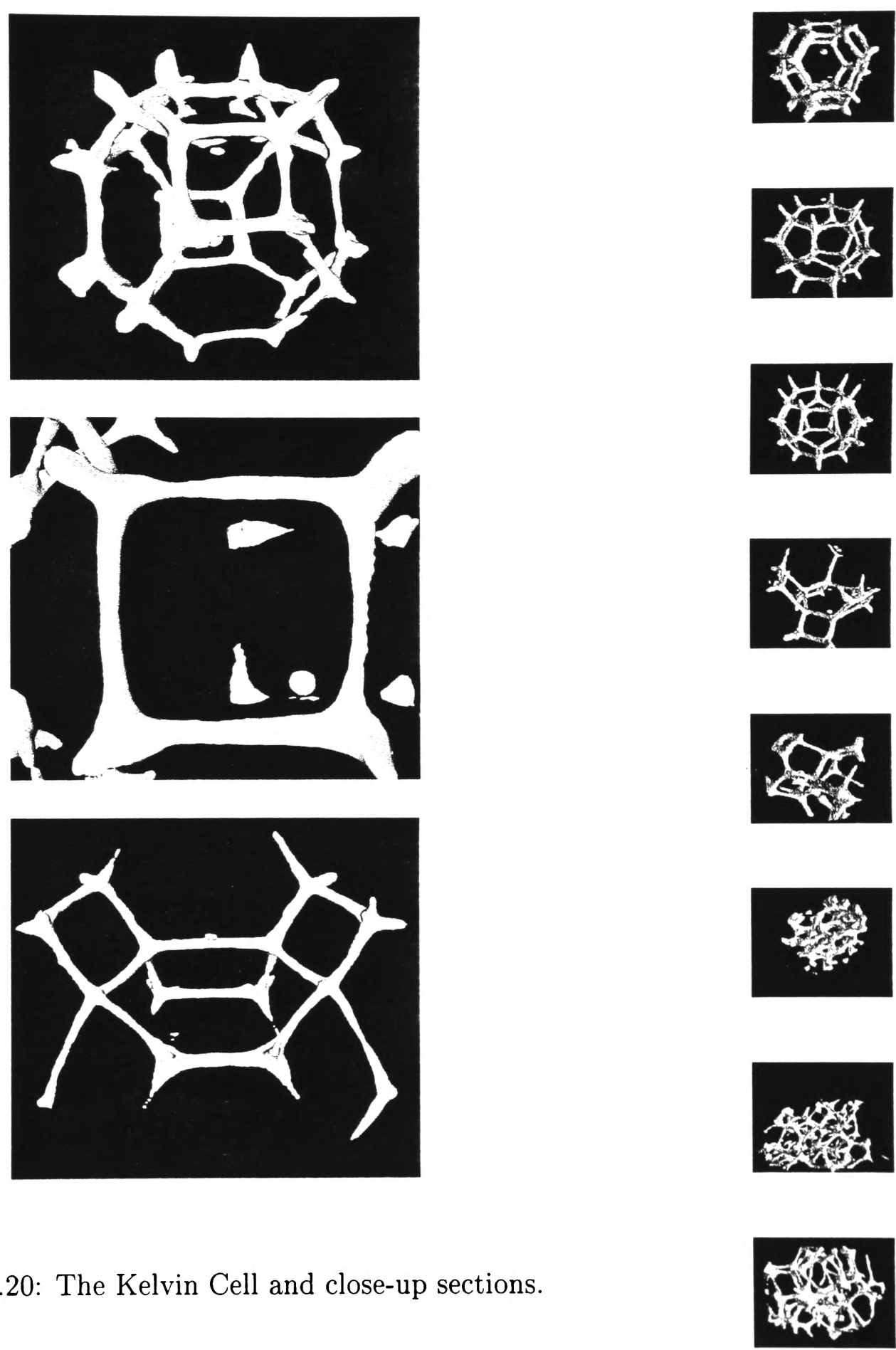


Figure 5.20: The Kelvin Cell and close-up sections.

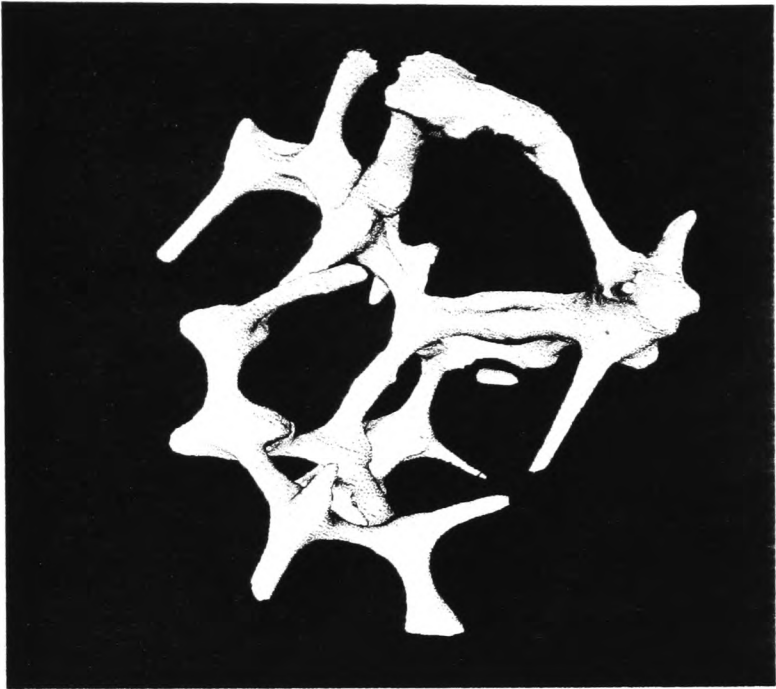


Figure 5.21: Solid foam 1.

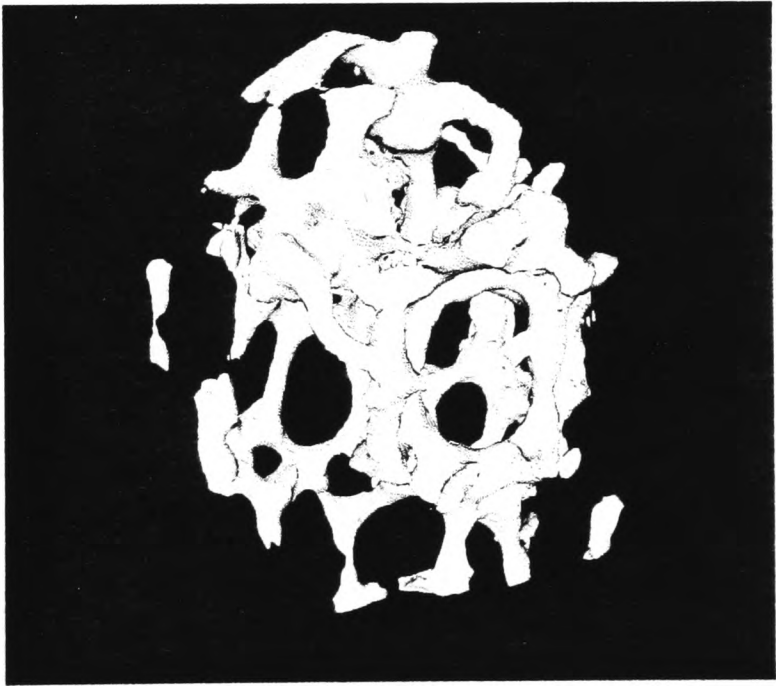
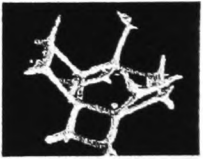
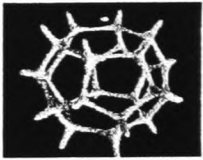
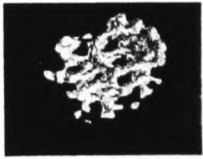


Figure 5.22: Solid foam 2.



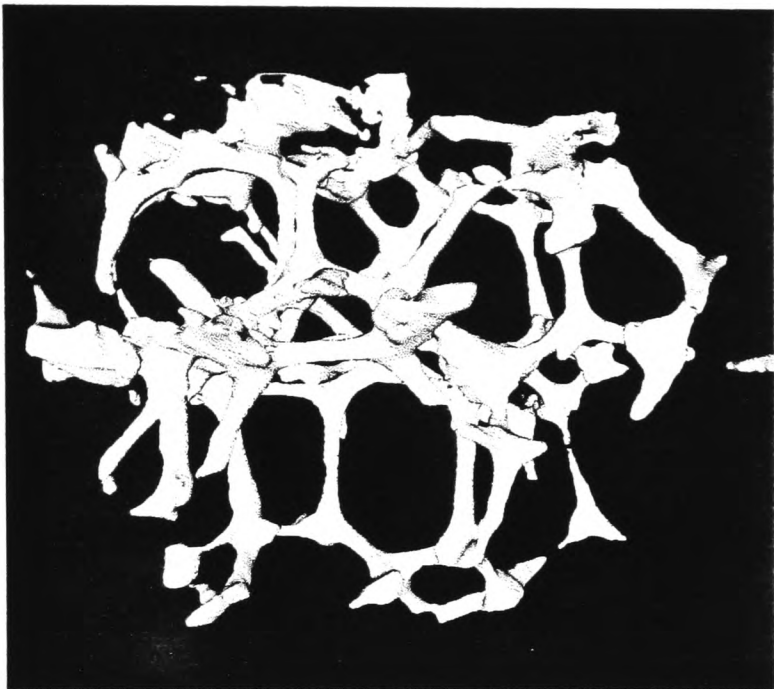


Figure 5.23: Solid foam 3.

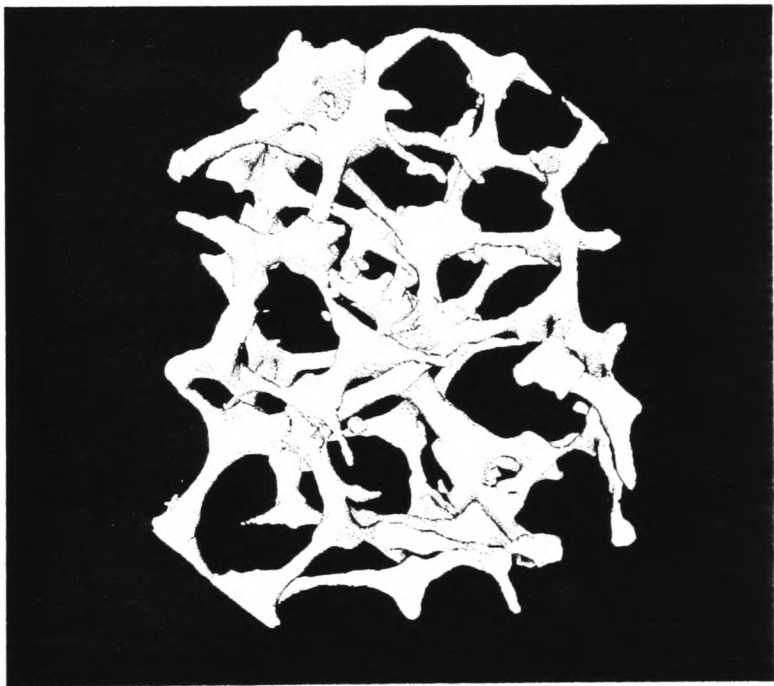
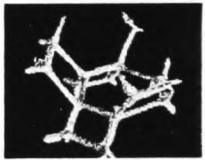
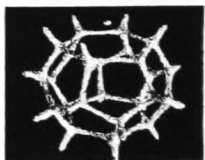
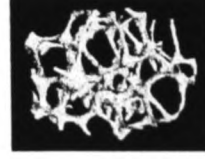
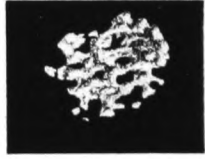
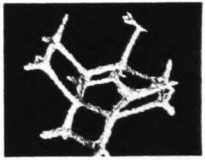
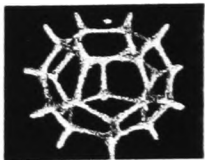


Figure 5.24: Solid foam 4.



software being used. At the expense of execution time, it may be possible to enhance the software to give minor improvements in model quality. It may also be that some other more complex scanning geometry may give some improvement; however, it is unlikely that either of these will lead to a big enhancement in image quality. What would probably be more helpful would be an increase in image and voxel resolution, which would also require a larger number of frames.

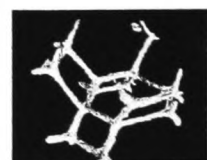


Chapter 6

Optimised method for liquid foam

Any kind of tomography operation is a computationally expensive process and the optical method described in chapter five is no exception. There are large amounts of data to be dealt with at every stage, from the initial set of images to the final voxel dataset. A corresponding volume of processing power is required to complete the inversion of the Radon transform in reasonable time. These computational requirements will not change in the future even though more powerful machines are becoming cheaper and more readily available.

Considering the special case of liquid foams with a high gas content, it seems that the full tomography operation could be unnecessary if a simpler model of the foam is desired. In this chapter, efforts to produce a representative model of liquid cellular foam structure without using optical tomography are documented.



6.1 Line models

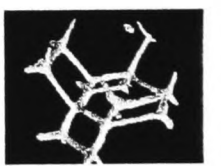
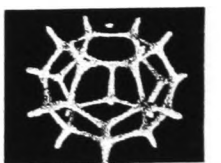
A simple cellular foam model could be just a list of lines in space. Each line has no thickness or curvature and the whole list simply connects the Plateau border vertices of the foam. The amount of storage required to contain such a model could easily be less than one thousandth that needed for a voxel dataset. The processing power needed to manipulate these *wire frame models* is similarly reduced.

It is possible to convert a voxel model into a line model by applying a skeletonisation filter and then searching for and joining the vertices in the resulting voxel skeleton with straight lines. However, it is also possible to go directly to a wire model by using a simplified optical method using less storage and processing resources¹ than tomography at every stage. The downside is that a line model contains much less information about foam structure than a voxel model. Whereas it is possible to see Plateau border curvature in the voxel model, for example, this information is lost in the simple line model. The payoff is in the speed of execution and reduced computational requirements of the method.

6.2 Experimental apparatus and method

Essentially, the same rig as was used for the optical tomography experiments previously can be used here (figure 5.7). A sequence of video images is digitised as before, only for this experiment it will be important to use the highest image resolution possible rather than the highest frame rate for

¹But some human assistance is required, as explained later.



reasons which will become clear later.

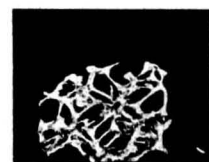
There are five steps to the method,

- Digitise video sequence
- Locate Plateau border vertices in each video image
- Resolve the position of each vertex in space
- Connect neighbouring vertices
- Check for 'unlikely' incorrect links

The most computationally difficult step is probably the second one listed here. Even for simple foam networks, devising a system for resolving every link and node reliably is rather difficult. To make this as easy as possible, the frame grabber configuration is altered so that fewer frames are captured, but at a higher resolution. In the same bandwidth that six frames per second at 160×120 can be digitised, we can get one frame per second² at 640×480 pixels, both at a depth of six bits per pixel. The extra resolution is useful in resolving more accurately the positions of vertices in particular. A comparison of the two image resolutions is shown in figure 6.1. It is clear that the higher resolution image will be a better subject for edge analysis than the lower resolution one.

However, edge detection is still difficult. The pictures in figure 6.1 show a vertical column of Kelvin cells on the central axis of a glass cylinder of internal radius 12mm and wall thickness 1mm. This is about the simplest

²With the Video for Windows software it is not possible to specify an exact frame rate, but after capture the system reports the average frame rate for the whole capture period.



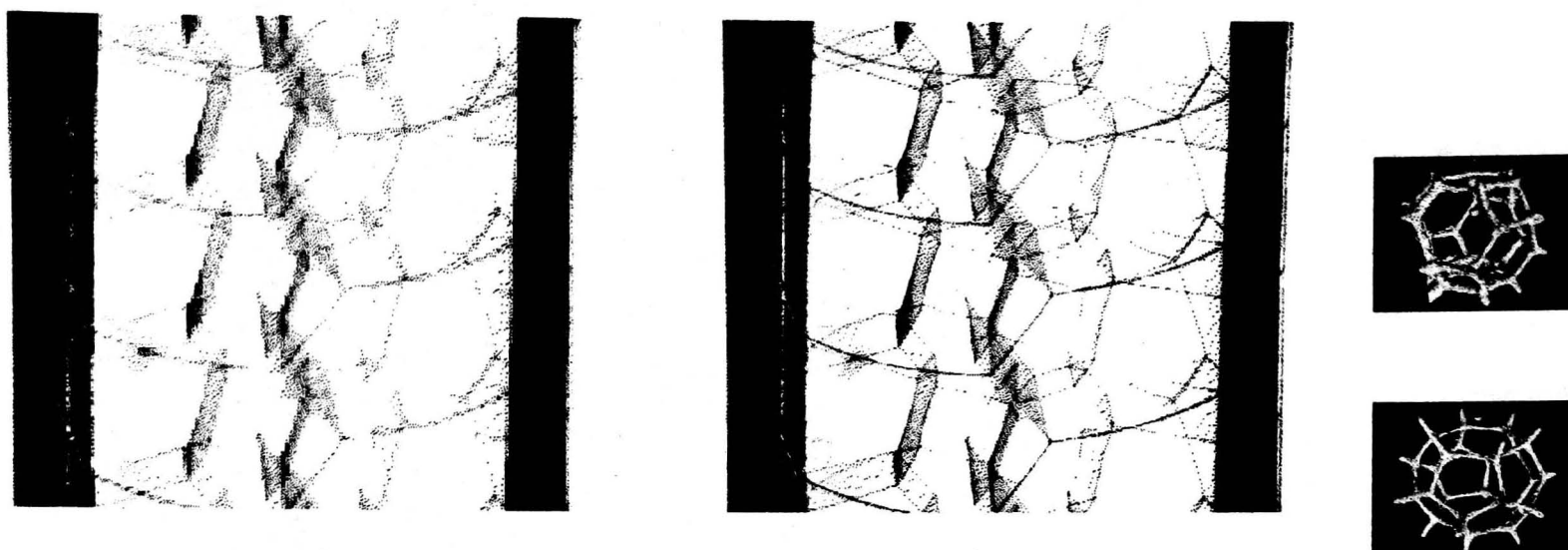


Figure 6.1: Two digitised images—left, at 160×120 , right, at 640×480 .

interesting configuration it is possible to analyse, but even so, the configuration of Plateau borders is very difficult to detect automatically by computer. Even by eye, it is hard to spot exactly where some Plateau border vertices lie. It seems that at this resolution and image complexity, it will not be possible to devise an algorithm which can reliably detect a large enough number of Plateau borders in each image to make reconstruction possible.

The best solution to this problem, given the resources available, is to use a piece of software which allows a user to locate the Plateau border vertices by hand. This is not ideal, but even a human often has trouble trying to see exactly where links connect to vertices in an image. If it is a difficult recognition problem for a human, an automatic solution is probably going to be very hard to produce.

The software displays one frame of video on the screen, magnified by a factor of two. A user can then use the mouse to highlight vertices in the image. A screen shot of the system in action is shown in figure 6.2. A user has placed markers on each visible Plateau border junction in the image. Inevitably, some vertices are obscured and can't be seen but over all the

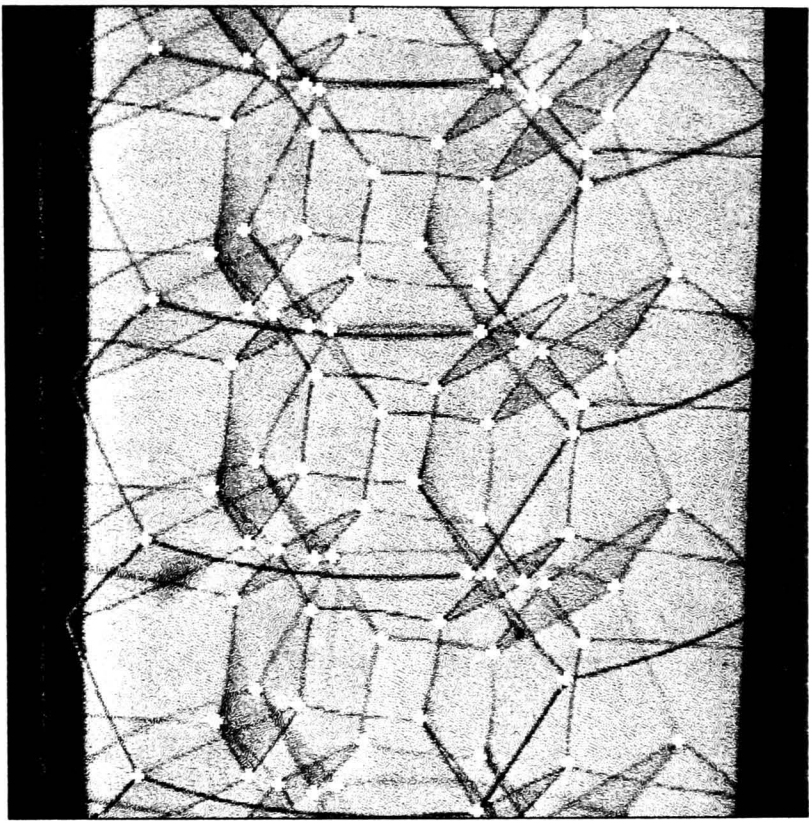


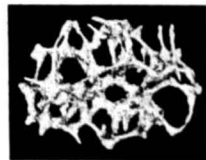
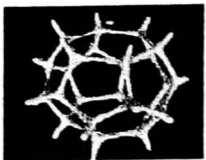
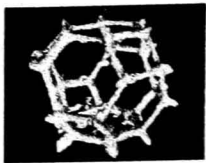
Figure 6.2: User selection of Plateau border junctions in an image.

images enough points are identified that a model can be constructed. The software remembers all the vertices marked by the user, and stores them in a file which will be used by the reconstruction program. This process is repeated for every image; on completion, a set of vertex files have been created which can be used to reconstruct a wire frame representation of the foam cellular structure.

6.3 A reconstruction algorithm

In detail, the reconstruction process comprises the following steps,

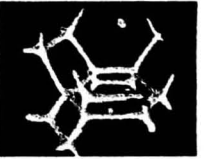
- Digitise video sequence
- Locate Plateau border vertices in each video image



- Resolve the position of each vertex in space
 - Check for projection rays mutual closest approach less than minimum $iDist?$
 - Cluster closest approach midpoints
 - Average cluster points to form vertices
- Calculate the nearest three neighbouring vertices of a vertex
- Check for and reject ‘unlikely’ incorrect links such as links not connected at about 109.47°



These steps are described below with Z ; where identifiers are not defined, they are assumed to be taken from definitions in the preceeding chapters.



6.3.1 Calculation of ray paths

Each vertex in a frame from the digitised video must lie on a line in three-dimensional space which passes through the vertex and through the centre of the camera lens, when the image is correctly aligned in space. In similar fashion to the earlier experiments, the calibration of the camera is known from measurements taken from the foam rig, hence the sizes and positions of the projections and the reconstructed model can be deduced.



As before, the real foam rotates about a vertical axis which lies in the same plane as the lens axis, which subtends an angle α_{cam} to the horizontal. Each projection is perpendicular to the lens axis, and the x axis of the projection is horizontal.



The reconstruction software takes a list of triples, $dataIn? : seq(\mathbb{Z} \times \mathbb{Z} \times \mathbb{R})$, which contain two integer coordinates and a real angle. The coordinates

are offsets of a Plateau border vertex relative to the centre of its projection (through which the lens axis passes), and the angle is the azimuthal rotation of the foam about its vertical axis. The order of triples does not matter, although in the implementation projections are read in one at a time, so the rotations increase uniformly.

The schema *Points2Rays* describes the process of converting the vertex coordinates in the set of projections to a sequence of three-dimensional lines in the vector form $a + \lambda b$, which has type $VECT \times VECT$. Here, the z axis is vertical, and the x axis is horizontal and perpendicular to the axis of the camera lens. When transforming a vertex, the actual location in space of the vertex and the location of the camera lens must be determined. Initially, the lens lies on the negative y axis, at a distance $lDist?$ from the origin, and the projection containing the vertex being transformed lies perpendicular the negative y axis with its centre on the origin.

Points2Rays

$dataIn? : seq(\mathbb{Z} \times \mathbb{Z} \times \mathbb{R})$

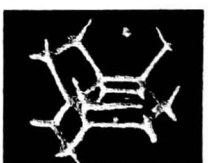
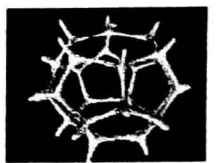
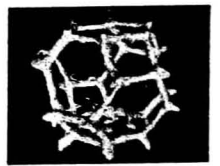
$lDist? : \mathbb{R}$

$rays! : seq(VECT \times VECT)$

$\forall n : \mathbb{N}_1 \mid n \leq \#dataIn? \bullet$

$rays!(n) = (rotZ(rotX((0, -lDist?, 0), \alpha_{cam}), thd3(dataIn?(n))),$
 $rotZ(rotX(((\mathbb{R}) - fst3(dataIn?(n)), -lDist?,$
 $(\mathbb{R}) - snd3(dataIn?(n))), \alpha_{cam}), thd3(dataIn?(n))))$

A rotation about the x axis by α_{cam} sets the correct elevation for the vertex and lens, followed by a rotation about the vertical z axis to set the azimuth



to that specified in the third field of the vertex triple. The projection ray passing from the camera lens through the vertex is then found by simple vector subtraction. It follows that the true three-dimensional vertex must lie somewhere along this ray.

As noted previously, type casts are sometimes needed in real software. The next section requires an integer to real conversion, which is described below,

$$\begin{array}{|l}
 \hline
 z2r : \mathbb{Z} \times \mathbb{R} \rightarrow \mathbb{R} \\
 \hline
 \forall z : \mathbb{Z}; r : \mathbb{R} \bullet \\
 \quad \text{if } z = 0 \text{ then } z2r(z, r) = r \\
 \quad \text{if } z > 0 \text{ then } z2r(z, r) = z2r(z - 1, r + 1) \\
 \quad \text{if } z < 0 \text{ then } z2r(z, r) = z2r(z + 1, r - 1)
 \end{array}$$

Then the typecast operator to reals is

$$\begin{array}{|l}
 \hline
 (\mathbb{R})_- : \mathbb{Z} \rightarrow \mathbb{R} \\
 \hline
 \forall z : \mathbb{Z} \bullet (\mathbb{R})z = z2r(z, 0)
 \end{array}$$

Ray paths are calculated for each vertex in *dataIn?* so that a complete set of projection rays is calculated and stored in the sequence *rays!* : seq(*VECT* × *VECT*).

6.3.2 Resolving ray intersections

From the list *rays!*, the location of Plateau border vertices in space can be determined. A simple way to do this is as follows,

- Test all pairs of rays for mutual closest approach

- Discard all pairs greater than Δ_{min} apart
- Cluster midpoints of remaining ray pairs to form vertices

It is unreliable to test for exact ray intersections, because of errors in the camera calibration, in the user-supplied vertex coordinates or in the floating-point machine arithmetic, hence the closest approach of ray pairs is used to find intersections. The value of Δ_{min} is large enough to cover for errors of a percent or so, but small enough so that most rays which clearly do not nearly intersect can be rejected.

Considering two rays, $\underline{r}_1 = \underline{a} + \lambda \underline{b}$ and $\underline{r}_2 = \underline{c} + \mu \underline{d}$, the formula for calculating their distance apart at the closest point is,

$$distance = \frac{(\underline{a} - \underline{c}) \cdot \underline{b} \wedge \underline{d}}{|\underline{b} \wedge \underline{d}|} \quad (6.1)$$

At the closest approach, the rays are mutually perpendicular. But a unit vector in this perpendicular direction is,

$$\hat{n} = \frac{\underline{b} \wedge \underline{d}}{|\underline{b} \wedge \underline{d}|} \quad (6.2)$$

Hence, forming an equation for the vector joining the two rays at the closest approach,

$$\underline{a} + \lambda \underline{b} + \underline{c} + \mu \underline{d} = \left(\frac{(\underline{a} - \underline{c}) \cdot \underline{b} \wedge \underline{d}}{|\underline{b} \wedge \underline{d}|} \right) \left(\frac{\underline{b} \wedge \underline{d}}{|\underline{b} \wedge \underline{d}|} \right) \quad (6.3)$$

From this, the values of λ and μ can be found at the closest point. Then, the endpoints of the connecting line between closest points can be determined and hence its midpoint, which is the value used for the ‘intersection’ point of the two rays.

The specification of this procedure requires a modulus function for calculating the length of a vector, *length*,

$$\begin{array}{|l}
length : VECT \rightarrow \mathbb{R} \\
\hline
\forall (x, y, z) : VECT \bullet \\
\quad length(x, y, z) = \sqrt{x^2 + y^2 + z^2}
\end{array}$$

Slightly more complex are the descriptions of the closest approach and midpoint functions, which follow from the equations above,

$$\begin{array}{|l}
dist : (VECT \times VECT \times \mathbb{R}) \times (VECT \times VECT \times \mathbb{R}) \rightarrow \mathbb{R} \\
midpoint : (VECT \times VECT \times \mathbb{R}) \times (VECT \times VECT \times \mathbb{R}) \rightarrow VECT \\
\hline
\forall (x_1, y_1, z_1), (a_1, b_1, c_1), (x_2, y_2, z_2), (a_2, b_2, c_2) : VECT; \lambda, \mu : \mathbb{R} \bullet \\
\quad dist(((x_1, y_1, z_1), (a_1, b_1, c_1)), \lambda, ((x_2, y_2, z_2), (a_2, b_2, c_2)), \mu) = \\
\quad \quad length(x_2 - x_1 + \mu a_2 - \lambda a_1, y_2 - y_1 + \mu b_2 - \lambda b_1, \\
\quad \quad \quad z_2 - z_1 + \mu c_2 - \lambda c_1) \\
\quad midpoint(((x_1, y_1, z_1), (a_1, b_1, c_1)), \lambda, ((x_2, y_2, z_2), (a_2, b_2, c_2)), \mu) = \\
\quad \quad ((x_1 + \lambda a_1 + x_2 + \mu a_2)/2, (y_1 + \lambda b_1 + y_2 + \mu b_2)/2, \\
\quad \quad \quad (z_1 + \lambda c_1 + z_2 + \mu c_2)/2)
\end{array}$$

The schema *GetIntersections* produces a list of midpoints of all possible ray pairs from the sequence of rays *rays?*.³ The output sequence *intersections!* is composed of tuples containing the intersection point, or mutual midpoint of closest approach, of its two parent rays. If two rays are never closer than *iDist?* apart then they are outside the margin of error for intersection and their mutual midpoint is not included in the output sequence of intersections.

³Clearly *rays!* and *rays?* will form a pipe in *Points2Rays >> GetIntersections*.

GetIntersections

$rays? : \text{seq}(VECT \times VECT)$

$iDist? : \mathbb{R}$

$intersections! : \text{seq } VECT$

$\forall m, n : \mathbb{N}_1 \mid m \leq \#rays? \wedge n \leq \#rays? \wedge m \neq n \bullet$

$\exists \lambda, \mu : \mathbb{R} \bullet \forall \alpha, \beta : \mathbb{R} \mid \lambda \neq \alpha \wedge \mu \neq \beta \bullet$

$dist(rays?(m), \lambda, rays?(n), \mu) \leq dist(rays?(m), \alpha,$

$rays?(n), \beta)$

$dist(rays?(m), \lambda, rays?(n), \mu) < iDist?$

$midpoint(rays?(m), \lambda, rays?(n), \mu) \in intersections!$

The power of Z enables the description of *what* happens when an algorithm is executed to occupy only a few lines of mathematics, even if *how* the goal is achieved typically takes many more lines of code. The specification of *GetIntersections* is a case in point; the solution of equation 6.3 in order to find λ and μ requires some modifications to account for the possible error conditions. Firstly, if the following occurs,

$$|\underline{b} \wedge \underline{d}| = 0$$

then there must be an exact intersection between two rays, and it is also true that

$$\frac{\underline{b} \wedge \underline{d}}{|\underline{b} \wedge \underline{d}|} = 0$$

but evaluating this expression will cause an arithmetic error for dividing by zero, which must be trapped. A more likely error is that the solution of

equation 6.3 which looks like

$$\begin{aligned} b_i \lambda - d_i \mu &= z_i - a_i + c_i \\ b_j \lambda - d_j \mu &= z_j - a_j + c_j \\ b_k \lambda - d_k \mu &= z_k - a_k + c_k \end{aligned}$$

will yield one or more of the following to be zero,

$$\begin{vmatrix} b_i & d_i \\ b_j & d_j \end{vmatrix} \quad \begin{vmatrix} b_j & d_j \\ b_k & d_k \end{vmatrix} \quad \begin{vmatrix} b_k & d_k \\ b_i & d_i \end{vmatrix}$$

but since at least one will always be non-zero if $|\underline{b} \wedge \underline{d}| \neq 0$, a solution can always be found if care is taken to check the validity of divisions. Fortunately, in the specification, the level of abstraction chosen is such that essential, but distracting, implementation details can be ignored.

6.3.3 Locating Plateau border vertices

The set of ray intersection points form clusters around the Plateau border vertices. By identifying these clusters and taking the average of their component vertices the centre of a Plateau border junction can be obtained. There is a function *mean* which finds the average of a set of real values,⁴

$$\begin{array}{|l} \text{mean} : \mathbb{P}_1 \mathbb{R} \rightarrow \mathbb{R} \\ \hline \text{mean}(\emptyset) = 0 \\ \text{mean}(s) = \frac{\min s}{(\mathbb{R})\#s} + \frac{(\mathbb{R})\#s - 1}{(\mathbb{R})\#s} \text{mean}(s \setminus \min s) \end{array}$$

⁴This requires the $\mathbb{Z}$ function *min* to be overloaded with a real variant, $\min : \mathbb{P}_1 \mathbb{R} \rightarrow \mathbb{R}$.

This function is part of the schema *MakeNodes* which describes the agglomeration of the ray intersection points produced by *GetIntersections*. A *blob* is a set of points which satisfy the criterion that no point in the blob is more than $nDist?$ from at least one other point in the same blob. Searching for blobs in the set *intersections?* should give a blob located at each Plateau border vertex, where there should be a high density of intersection points. In practice, $ndist$ is only one percent or so of the width of the whole point space under consideration. Each blob has a centroid which is simply the average of its component points; this is taken to be the location of the Plateau border junction represented by the blob.

<i>MakeNodes</i> $intersections? : seq VECT$ $nDist? : \mathbb{R}$ $nodes! : \mathbb{P}_1 VECT$ $\exists blobs : \mathbb{P}_1 \mathbb{P}_1 (VECT \times \mathbb{N}_1) \bullet$ $\forall s : blobs \bullet \forall (x_1, y_1, z_1, n_1) : s \bullet \exists (x_2, y_2, z_2, n_2) : s \bullet$ $length(x_1 - x_2, y_1 - y_2, z_1 - z_2) < nDist?$ $\forall (x, y, z, n) : s \bullet (x, y, z) = intersections?(n)$ $(mean(\{(x, y, z, n) : s \bullet x\}), mean(\{(x, y, z, n) : s \bullet y\}),$ $mean(\{(x, y, z, n) : s \bullet z\})) \in nodes!$

An implementation of this algorithm could be a recursive procedure which would flag pairs of points less than $nDist?$ apart with a unique group flag. Successive iterations could then join vertices such that they both assumed the lowest group number of either vertex. Probably a more efficient implemen-

tation would sort the points first in order to reduce the number of necessary comparisons.

6.3.4 Connecting vertices

There is no information about the Plateau border geometry fed into the reconstruction software. However, in the special case of monodisperse foams there seems to be a convenient heuristic which applies to the interconnection of *most* vertices. In all foams observed in the course of this work, a vertex is joined by Plateau borders to the closest four neighbouring vertices. Clearly though, vertices near the edge of a reconstructed model may join with fewer than four other vertices, as the model will be incomplete here.

Connecting the vertices is then achieved by searching for the nearest neighbours of any particular vertex. A routine for sorting real numbers is needed,

$$\begin{array}{|l}
 \text{sort} : \mathbb{P} \mathbb{R} \rightarrow \text{seq } \mathbb{R} \\
 \hline
 \text{sort}(\emptyset) = \langle \rangle \\
 \forall r : \mathbb{R} \bullet \text{sort}(\{r\}) = \langle r \rangle \\
 \forall R : \mathbb{P}_1 \mathbb{R} \mid \#R > 1 \bullet \\
 \quad \text{sort}(R) = \langle \min R \rangle \frown \text{sort}(R \setminus \min R)
 \end{array}$$

There is also a convenience function, *vdiff*, which calculates the difference of two vectors,

$$\begin{array}{|l}
 \text{vdiff} : \text{VECT} \times \text{VECT} \rightarrow \text{VECT} \\
 \hline
 \forall (x_1, y_1, z_1), (x_2, y_2, z_2) : \text{VECT} \bullet \\
 \quad \text{vdiff}((x_1, y_1, z_1), (x_2, y_2, z_2)) = (x_1 - x_2, y_1 - y_2, z_1 - z_2)
 \end{array}$$

Both these functions are used in the schema *LinkNodes*, which produces an output list of pairs of connected nodes. The set *dists* contains the values of the distance of a vertex *m* to every other vertex in the model; hence when *dist* is sorted in ascending order, its fifth element will be further away from *m* than all of the vertices *m* could be connected to. All four vertices within this range (*maxDist*) are members of the output set *links!*.

<i>LinkNodes</i>
$nodes? : \mathbb{P}_1(VECT \times \mathbb{P}_1 \mathbb{N}_1)$ $links! : \mathbb{P}_1(VECT \times VECT)$
$\forall m : nodes? \bullet \exists dists : \mathbb{R} \bullet$ $\quad \forall n : nodes? \mid m \neq n \bullet$ $\quad \quad length(vdiff(m, n)) \in dists$ $\quad \quad \exists maxDist \mid maxDist = sort(dists)(5) \bullet$ $\quad \quad \text{if } length(vdiff(m, n)) < maxDist$ $\quad \quad \text{then } (m, n) \in links!$

Even though *links!* is a set, it is possible for duplicate links to exist because the order of points in a link tuple of vertices could be reversed. *RemDups* sorts each link pair by the vector length of each vertex vector—a completely arbitrary criterion, but this enables duplicate links to be removed from the output set.

RemDups

 $links?, links! : \mathbb{P}(VECT \times VECT)$

 $\forall(a, b) : links? \bullet$

 if $length(a) < length(b)$ then $(a, b) \in links!$

 else $(b, a) \in links!$

6.3.5 Model reality-check

The wire model reconstruction may not exactly resemble the foam in the video—errors in the reconstruction caused by noisy data, nodes which have been hidden in several images and are hence poorly located, or just errors in the human input stage of the method can lead to vertices being created which should not exist or to links being made where there are no Plateau borders in the real foam. Both these types of error can sometimes be corrected by checking that the Plateau borders at a junction meet at angles of about 109.47° . Small errors in the locations of the Plateau border vertices and also the small curvatures of the Plateau borders not represented in the wire model mean that angles of exactly 109.47° do not exist, even if they did in the real foam. By rejecting links which do not satisfy the criterion that both ends of a link must make roughly 109.47° with at least two other adjacent links, the model is more likely to be correct. This allows for the case at the edges of a model where there are often only three links at a vertex.

A basic component of the link-checking system is a schema which describes the operation of calculating the angle between two vectors,

$$\begin{array}{|l}
 \text{vectAngle} : VECT \times VECT \rightarrow \mathbb{R} \\
 \hline
 \text{vectAngle}((x_1, y_1, z_1), (x_2, y_2, z_2)) = \\
 \cos^{-1} \frac{x_1 \times x_2 + y_1 \times y_2 + z_1 \times z_2}{\sqrt{x_1 \times x_1 + y_1 \times y_1 + z_1 \times z_1} \times \sqrt{x_2 \times x_2 + y_2 \times y_2 + z_2 \times z_2}}
 \end{array}$$

The *vectAngle* function is used in *CheckLinks* which describes the complete check on the set of links for foam-like Plateau border angles. An input here is Δ_{angle} ? which is the permitted variation in line angles. In the implementation, this has to be 20° or more so that correct links are with $109.47^\circ \pm 20^\circ$.

The two sets of pairs, *lines1* and *lines2*, contain the links which attach at the first and second ends of a link (s, e) . If there are two links in each set which make the right angles with the test link, then that link is added to the output set.

CheckLinks

 $links? : \mathbb{P}_1(VECT \times VECT)$
 $\Delta_{angle?} : \mathbb{R}$
 $links! : \mathbb{P}(VECT \times VECT)$

 $\forall (s, e) : links? \bullet$
 $\forall (s, u), (v, s), (w, e), (e, x) : VECT \times VECT \mid u \neq e \wedge w \neq s \bullet$
 $\exists lines1, lines2 : VECT \bullet$
 $vdiff(u, s) \in lines1 \wedge vdiff(v, s) \in lines1$
 $vdiff(w, e) \in lines2 \wedge vdiff(x, e) \in lines2$
 $\text{if } \exists l_1, l_2 : lines1; l_3, l_4 : lines2 \mid l_1 \neq l_2 \wedge l_3 \neq l_4 \bullet$
 $vectAngle(vdiff(e, s), l_1) = 109.47^\circ \pm \Delta_{angle?}$
 $vectAngle(vdiff(e, s), l_2) = 109.47^\circ \pm \Delta_{angle?}$
 $vectAngle(vdiff(s, e), l_3) = 109.47^\circ \pm \Delta_{angle?}$
 $vectAngle(vdiff(s, e), l_4) = 109.47^\circ \pm \Delta_{angle?}$
 $\text{then } (s, e) \in links!$

6.3.6 Complete reconstruction specification

Now that the component parts of the method are defined, the whole system is easily described by

$$Points2Rays \gg Getintersections \gg MakeNodes \gg$$

$$LinkNodes \gg RemDups \gg CheckLinks$$

What can be done if the assumption that each Plateau border vertex is connected to its nearest neighbours is not applicable? Although this pathological case is rare and not seen in the experimental foams studied here, it will cause

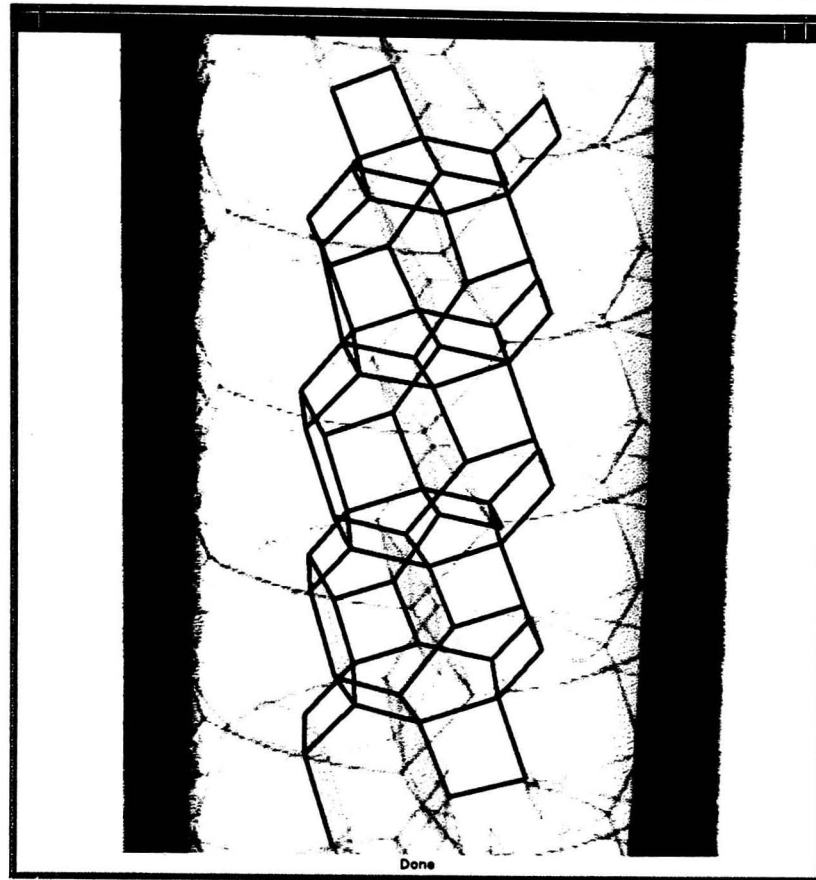


Figure 6.3: User selection of Plateau borders in an image.

the reconstruction software to fail.

Modifications to the user input step and the reconstruction algorithm can get around this problem. If the user can also specify which vertices are connected as well as their locations in an image then reconstruction can proceed with a simpler vertex-linking stage. Figure 6.3 shows a modified version of the image processing program used to select features in an image. This time, the user selects lines rather than points and can therefore add the link information to each frame as well as vertex points.

As the link information exists only for the two-dimensional network in each individual frame, it is essential for the reconstruction system to keep track of which points end up in which cluster of intersection points. Then link information can be converted into three dimensions by understanding how the components of each vertex are connected to the components of other

vertices. Three schemas in the definition need modification to include this change. Firstly, a replacement for *GetIntersections* now produces an output list which contains tuples of type $VECT \times \mathbb{N}_1 \times \mathbb{N}_1 \times \mathbb{R}$ instead of $VECT \times \mathbb{R}$. The extra pair of natural numbers are the offsets in the list *rays?* of the component vertices making the intersection point.

GetIntersections2

rays? : seq($VECT \times VECT$)

iDist? : $\mathbb{R}$

intersections! : seq($VECT \times \mathbb{N}_1 \times \mathbb{N}_1 \times \mathbb{R}$)

$\forall m, n : \mathbb{N}_1 \mid m \leq \#rays? \wedge n \leq \#rays? \wedge m \neq n \bullet$

$\exists \lambda, \mu : \mathbb{R} \bullet \forall \alpha, \beta : \mathbb{R} \mid \lambda \neq \alpha \wedge \mu \neq \beta \bullet$

$dist(rays?(m), \lambda, rays?(n), \mu) \leq dist(rays?(m), \alpha, rays?(n), \beta)$

$dist(rays?(m), \lambda, rays?(n), \mu) < iDist?$

$(midpoint(rays?(m), \lambda, rays?(n), \mu), m, n,$

$dist(rays?(m), \lambda, rays?(n), \mu)) \in intersections!$

In a similar way, each *node* requires a set of numbers which represent its component points. So the output set from *MakeNodes2* has type $\mathbb{P}_1(VECT \times \mathbb{P}_1 \mathbb{N}_1)$ and this extra information is provided by the new fields in each member of *intersections?*.

MakeNodes2

intersections? : seq(*VECT* × $\mathbb{N}_1$ × $\mathbb{N}_1$ × $\mathbb{R}$)

nDist? : $\mathbb{R}$

nodes! : $\mathbb{P}_1(\text{VECT} \times \mathbb{P}_1 \mathbb{N}_1)$

$\exists \text{blobs} : \mathbb{P}_1 \mathbb{P}_1(\text{VECT} \times \mathbb{N}_1) \bullet$

$\forall s : \text{blobs} \bullet \forall (x_1, y_1, z_1, n_1), (x_2, y_2, z_2, n_2) : s \bullet$

$\text{length}((x_1 - x_2, y_1 - y_2, z_1 - z_2)) < nDist?$

$\forall (x, y, z, n) : s \bullet \exists n_1, n_2 : \mathbb{N}_1; r : \mathbb{R} \bullet$

$\text{intersections!}(n) = ((x, y, z), n_1, n_2, r)$

$(\text{mean} \{((x, y, z, n) : s \bullet x\}, \text{mean} \{(x, y, z, n) : s \bullet y\},$

$\text{mean} \{(x, y, z, n) : s \bullet z\}), \{(x, y, z, n) : s \bullet n\}) \in \text{nodes!}$

From the user-supplied edge information comes the set $\text{linkData?} : \mathbb{P}_1(\mathbb{N}_1 \times \mathbb{N}_1)$, which is a set of pairs of offsets in the *intersections?* point list. The new schema *LinkNodes2* uses this information to connect the Plateau border vertices rather than searching for nearest vertex neighbours. If there are two points in separate nodes which are in the link list, then the parent nodes are added to the output set of Plateau borders.

LinkNodes2

 $nodes? : \mathbb{P}_1(VECT \times \mathbb{P}_1 \mathbb{N}_1)$
 $intersections? : seq(VECT \times \mathbb{N}_1 \times \mathbb{N}_1 \times \mathbb{R})$
 $linkData? : \mathbb{P}_1(\mathbb{N}_1 \times \mathbb{N}_1)$
 $links! : \mathbb{P}_1(VECT \times VECT)$

 $\forall((x_1, y_1, z_1), N_1), ((x_2, y_2, z_2), N_2) : nodes? \mid (x_1, y_1, z_1) \neq (x_2, y_2, z_2) \bullet$
 $\forall((p_1, q_1, r_1), a_1, b_1, d_1) : \{n_1 : N_1 \bullet intersections?(n_1)\},$
 $((p_2, q_2, r_2), a_2, b_2, d_2) : \{n_2 : N_2 \bullet intersections?(n_2)\} \bullet$
 $(a_1, b_2) \in linkData? \vee (b_2, a_1) \in linkData? \vee$
 $(b_1, a_2) \in linkData? \vee (a_2, b_1) \in linkData? \Rightarrow$
 $((x_1, y_1, z_1), (x_2, y_2, z_2)) \in links!$

The output from *LinkData2* has exactly the same signature as that from *LinkData* and can be piped on into *RemDups* as before. In practice, on the experimental data used in this project, the final model output is identical in most cases whichever method of linking the vertices is used. Rarely, a difference occurs when the software incorrectly locates vertices due to the reasons described earlier. In this case, the failure modes of each algorithm are different.

6.4 Results

Some computer models of foam bubbles seen in the experimental apparatus are shown in this section. In each case, the foam geometry is represented by a network of straight, thin lines in space even though the actual foam contains Plateau borders with a small curvature.

Figure 6.4 is an example of one of the simplest liquid foam structures. The bubbles in this foam are large enough to occupy a quadrant of the containing tube cross-section and meet in the centre of the tube with quadrilateral interfaces. This forms a vertical ‘ladder’ of squares linked by their diagonally opposite corners at almost mutually perpendicular angles. These quadrilateral faces are very similar to those in Kelvin’s bubble, being planar and having outwardly-curved edges.

It is clear from the wire model that most of the visible structure has been correctly resolved although there are errors at the lower end. These are probably caused by the increasing refraction of light as the path through the glass tube becomes longer, lower down the video image. It is not possible to check for two legal Plateau border angles at each vertex in this model, because some Plateau borders only meet one other at one of their vertices.

As with the tomography experiments, it is possible to study some of the Weaire-Phelan bubble shapes. Figure 6.5 shows a highly regular fourteen-sided bubble with two hexagonal and twelve pentagonal faces. Unfortunately the video hardware used in this project is not suitable for resolving large numbers of bubbles, so it is impossible to try and resolve a cluster of bubbles which might be organised in the way Weaire and Phelan suggest. This, as well as the tomographic work, does however provide proof that the individual bubbles do exist in real foams.

Figure 6.6 shows a group of three near-perfect Kelvin bubbles arranged in a vertical column. There are a couple of flaws at the extremities of the wire model, again where refraction effects are more disruptive. However, two of the three bubbles are perfect representations of the real foam geometry and it

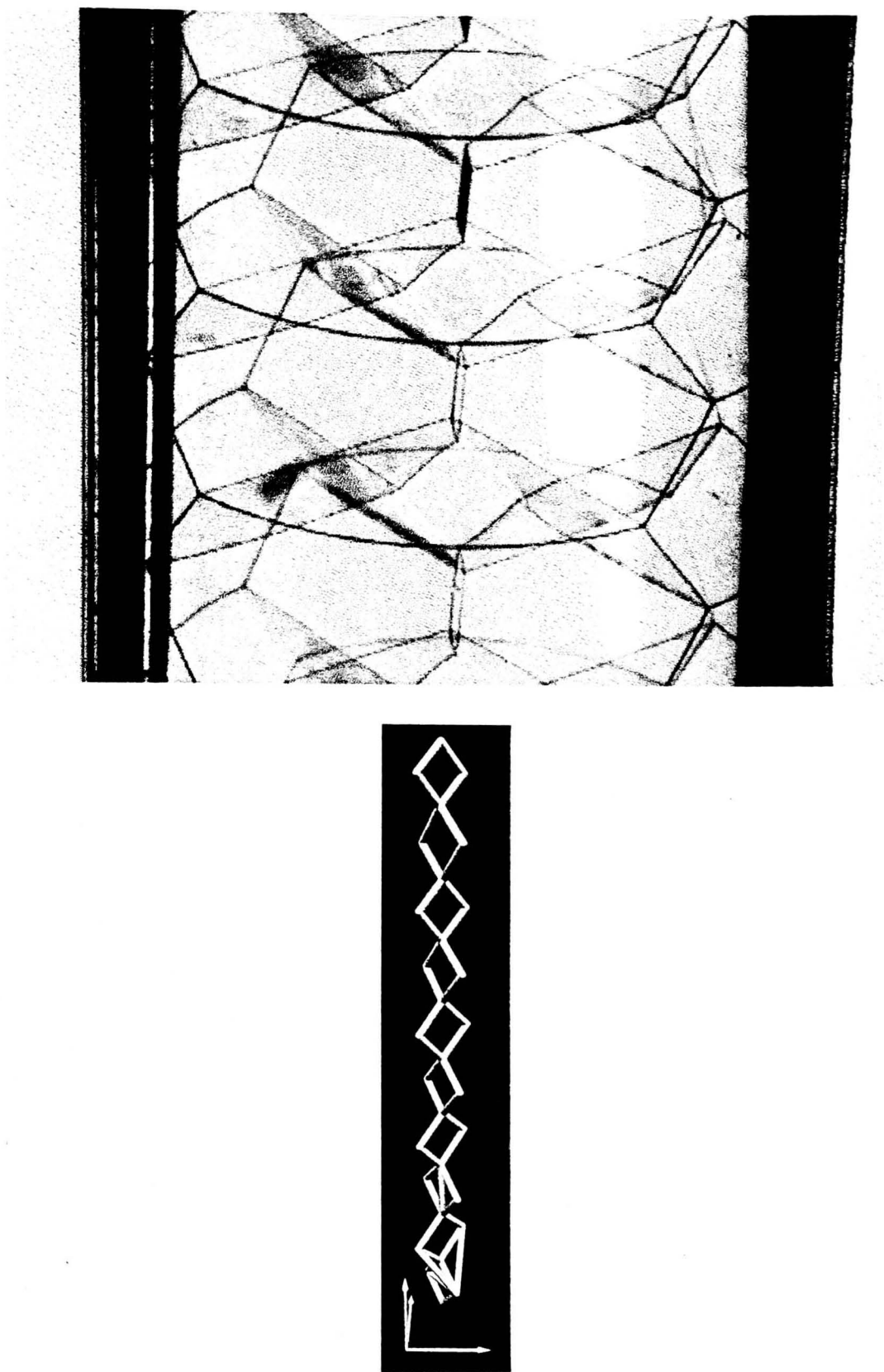


Figure 6.4: Diamond foam lamellae (video top, wire model bottom). Each diamond is of side length approximately 2mm.

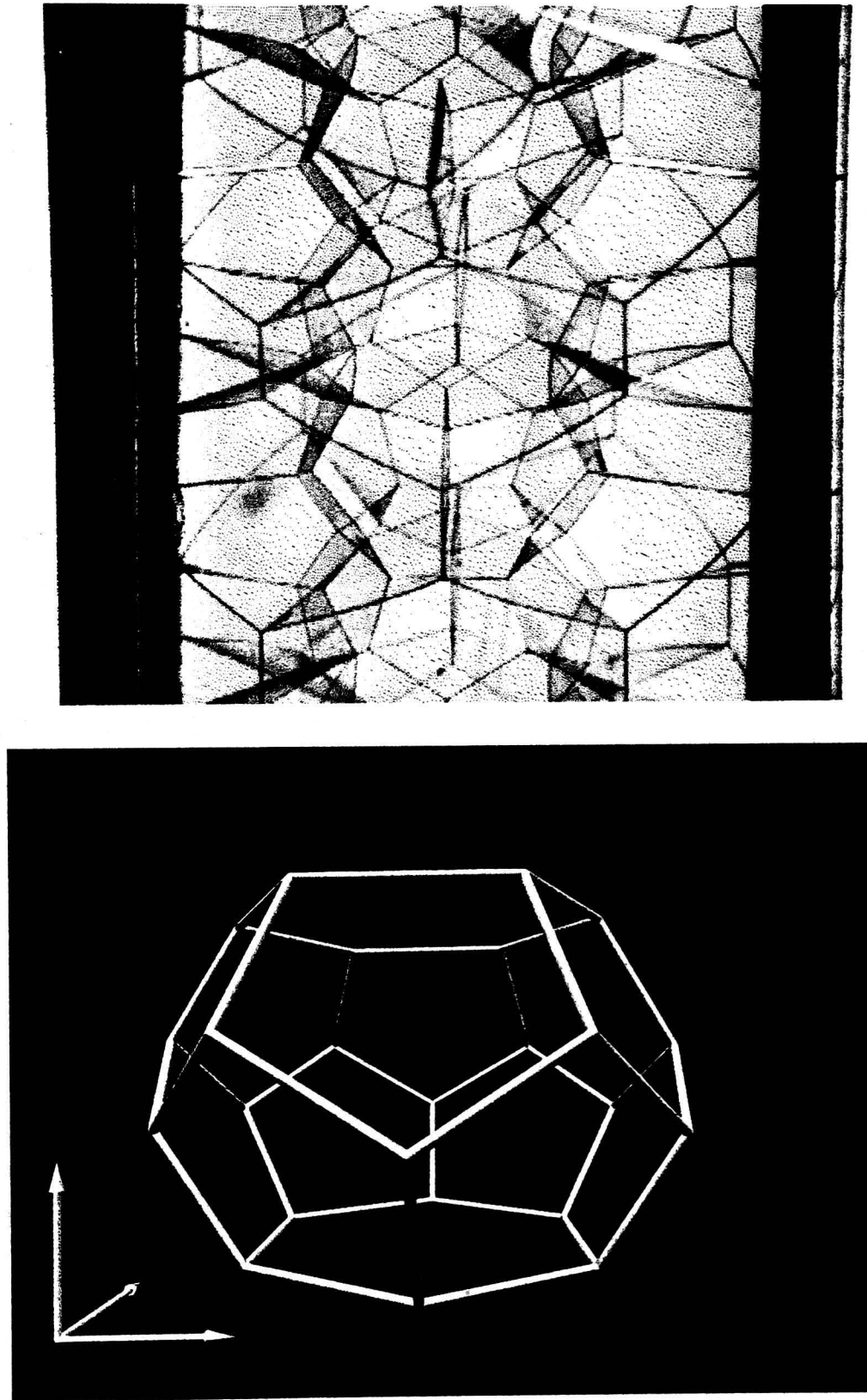


Figure 6.5: Weaire-Phelan foam bubble. The bubble diameter is approximately 5mm.

is easy to see the overall bubble structure. Unfortunately it is not possible to resolve the surrounding layer of bubbles around this central bubble column.

Liquid foams such as the aqueous soap foams studied here are amenable to the use of *a priori* information to assist in the reconstruction process due to their strict adherence to geometric rules. However, the solid polyurethane foams by the nature of their production process do not conform so rigorously to the tetrahedral angle constraints, for example. In this case, there is little which can be done to assist reconstruction other than to apply very general constraints such as voxel continuity (see [77] as described in chapter five).

The next chapter reviews the whole foam visualisation project and adds some suggestions as to where future research should be directed.

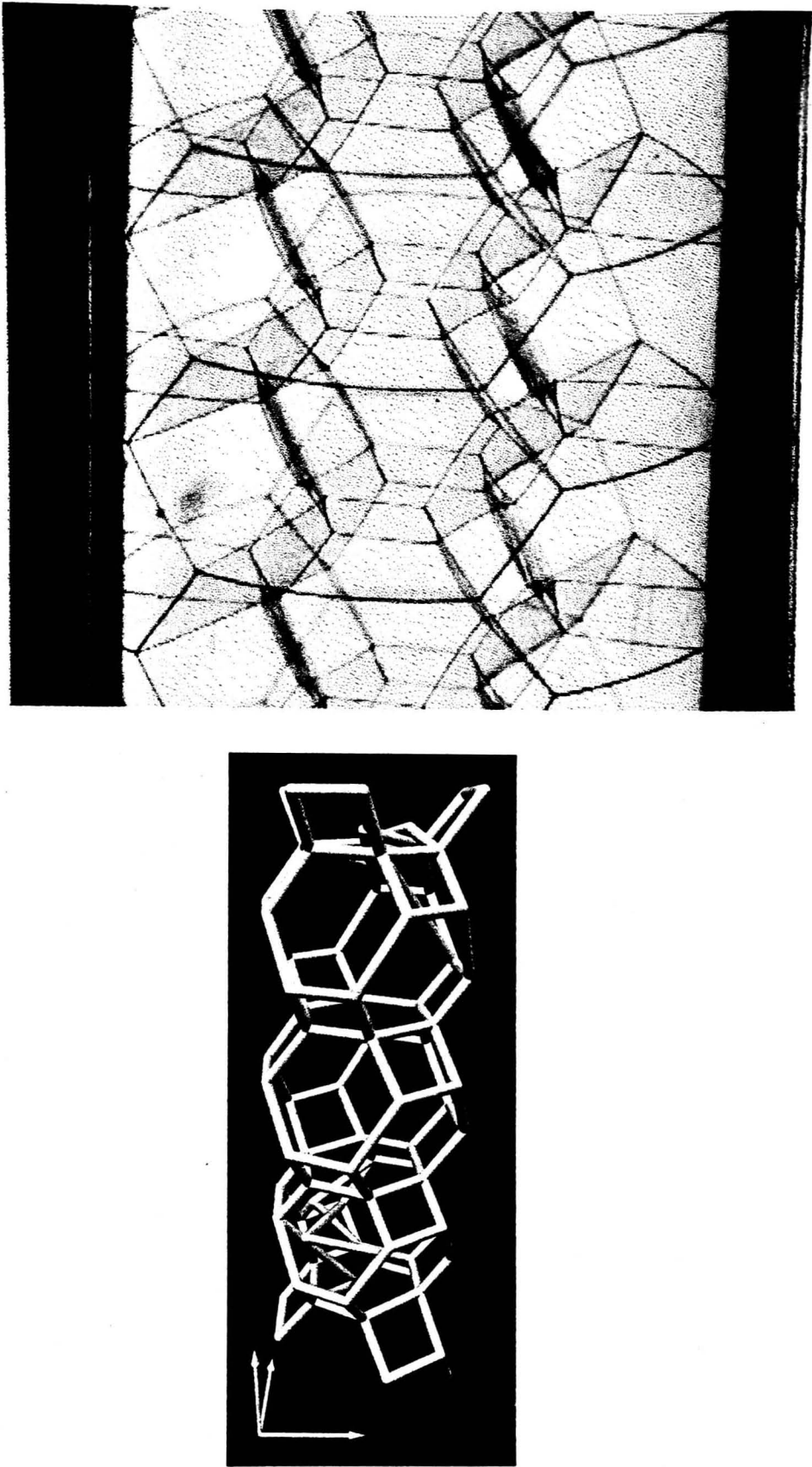


Figure 6.6: Kelvin foam bubbles. Individual bubble diameter is approximately 5mm.

Chapter 7

Conclusions

This thesis has described three methods of using optical systems to resolve the geometric structure of cellular foam networks. Probably for the first time, voxel and wire frame models of real foam bubbles have been presented. Although the foam samples which have been studied are small and not very complex, these methods are scalable with the availability of more processing power and superior video systems.

7.1 Reconstruction method pros and cons

There are some general considerations when using photographic systems on foams. One of the difficult problems is the illumination of the foam sample, especially in the liquid foam case. Any light sources can reflect off shiny aqueous interfaces and cause specular highlights which disrupt the reconstruction processes. This tends to rule out the use of lasers, either as a parallel light source or as a plane illumination.

All the work presented here was done with foams lit by a diffuse back-light, which gives good shadows for both liquid and solid foams and does

not cause significant specular highlights. This does add an extra constraint; obviously the foam sample must be thin or sparse enough that some light travels through to the camera. Unfortunately, the quality of the reconstruction becomes worse when less light is let through, which does not bode well for the study of complex or very dense cellular structures.

Another important consideration is the time required for the capture of enough images for the reconstruction to be worthwhile. The confocal method using a photographic camera needed several minutes to capture 36 exposures. During this time the foam sunk slightly in the containing vessel, although image processing was able to correct this error. It is much harder to determine if the foam has undergone any structural rearrangement, such as a $T1$ or $T2$ transformation. The use of a digital video system enables five times as many high resolution images to be digitised in the same time frame as the 36 exposures, which reduces the total time of capture considerably. Now it is possible to get image digitisers for pentium class computers which can grab twenty-five times as many high resolution frames per second as the system used in this work.

In the next sections, the advantages and disadvantages of each of the three optical methods presented here are discussed in more detail.

7.1.1 Focal plane photography

The confocal method uses what is probably the simplest experimental rig and requires the least processing power to reconstruct a model. However, this is only true if the image data set is of high quality. In particular, if the lens cannot produce a small depth of field, the computational requirements

increase and even then it may not be possible to reconstruct an accurate model.

Even with a high quality lens, it seems unlikely that the spatial resolution in the direction parallel with the lens axis will be anywhere near as high as the resolution in the plane of each image. In the work described here, the depth resolution is well over 100 times lower than the resolution in the image plane. Any measurements on the foam as a whole would have to be carried out at the lower resolution.

Confocal imaging works more effectively when the (dark) features in an image are thin or sparse compared to the (uniform) background. If this is the case, other features which do not lie in the focal plane are more easily rejected because the blurring effect lightens them as well as reducing sharpness. For dense cellular networks, this effect is lessened by the darkening of the background caused by many defocussed features. The lower contrast between focal plane features and others makes detection of edges more difficult.

There are some applications, though, where this optical method may be the best possible method. Each of the axial tomography systems requires a foam sample small enough to be rotated in front of a camera; it is not hard to imagine an industrial foam analysis machine which could only be presented with a large foam sample and must conduct the survey from one edge. Provided suitable illumination can be found, it seems that confocal imaging can give results of foam structure several layers into the bulk of the material.

The confocal software developed here produces a wire frame model of the foam geometry. Given sufficient accuracy, there is no reason why this could

not be extended to include Plateau border width information. It may be possible to assume that the properties of Plateau borders in the focal plane, whose widths are easy to measure, could be added to the Plateau borders parallel to the lens axis which are more difficult to measure. It is also possible to write software which could guess the locations of some Plateau borders using the geometric rules which must be satisfied by liquid foams. As always, the success of these programs depends on the quality of the initial image data.

7.1.2 Optical axial tomography

The axial tomography scanner adds complexity to the scanning geometry and experimental rig and increased computational requirements but produces a more detailed and useful end result. It is also more robust when the image data is of poorer quality, having lower signal to noise ratio for example.

The more complicated scanning system of axial tomography may restrict its use to small foam samples only. The liquid foams studied here were contained in a glass tube as any non-symmetrical vessel will refract light differently depending on its orientation to the camera. This will clearly affect the reconstruction, unless some way of equating the refractive indices of the vessel and its surroundings can be found. Even with a tubular vessel, the only useful part of the digitised images is in the central portion where the refraction is minimised.¹

Axial tomography does appear to cope well with the denser foams, for example the polyurethane solid foams. However, this method does have problems in reconstruction of high spatial frequencies such as the very thin

¹ Attempts to add the refraction calculations to the reconstruction software resulted in programs which took an infeasibly long time to execute.

Plateau borders found in some images. These invariably become blurred in the reconstructed voxel model even before low-pass filtering to remove noise.

The voxel models produced by axial tomography are much more homogeneous than those produced by the confocal system. The resolution in all directions is approximately the same, with no noticeable variation. Measurements of the bubble sizes and shapes could now be taken with much more accuracy than in the confocal case.

7.1.3 Optimised tomographic method

This is an attempt to overcome the focal plane system's problem with inhomogeneous result resolution and the axial tomographic method's poor performance with very thin Plateau borders. The final model, a wire frame, contains less information than the voxel model but is computed very much more cheaply and from very many fewer frames. However, the image processing step is much more complicated in this case and the very high resolution necessary for this to be completed automatically was not available during the course of this work. Instead of working at a resolution of about 500×500 pixels, which requires a human to mark the edges and Plateau border vertices, double or triple this resolution might allow an image processing algorithm to perform this step. The extra resolution means that lines are more clearly defined and an edge-detection program is more likely to find all of the Plateau borders and intersections.

Even though human assistance is required in the image processing stages, the relatively simple reconstruction software does well to reconstruct accurately the cellular geometry. The wire models show the regularity and uni-

formity of real monodisperse foam bubbles for the first time although information about the curvature of Plateau borders is lost. The image processing stage could be modified to include curvature information, though.

7.1.4 Software execution times

The performance of all three reconstruction methods depends the size of the input data. A brief summary follows for typical runs of each program on an Iris 4D workstation with a 50MHz R4000 MIPS processor.²

A major part of the execution time for the confocal system is taken in the image preprocessing stage. The lack of consistency in scanned photographs necessitates considerable computational time to correct, in particular the cross-correlation between images needed for correct registration. This takes approximately twenty minutes. Similarly, the convolution operations used for the edge detection are also processor-intensive. On the 36-frame data set this will take 10–15 minutes. Converting the images to a line model is much faster, the complete set being finished in under one minute. Visualisation of the finished model may take anywhere between milliseconds to minutes depending on the quality of the final images. The high-quality pictures in chapter four were produced by the POVray raytracer which takes tens of minutes.

The axial tomography experiments uses more images and the more sophisticated algorithm requires much more processor time. The preprocessing step of inverting the digital images takes negligible time and is conveniently folded into the main reconstruction program, but liquid foams incur the

²The 64MB main memory is sufficient to contain the largest data sets shown in chapter five.

Method	Image acquisition	Image processing	Reconstruction time	Visualisation
Confocal imaging (with photography)	several mins	20 mins	15 mins	fast
Axial tomography (digital imaging)	30 secs	5 mins	30 mins	slow
Optimised method (human assistance)	< 30 secs	slow	1-2 mins	fast

Table 7.1: A summary of tomographic system image acquisition and processing times.

added burden of edge enhancement as above. Reconstruction time depends on the resolution of the final model; a voxel set of resolution 130^3 takes about 30 minutes to prepare. However, doubling the resolution in each direction results in an eight-times slowdown.

The optimised method in chapter six is more efficient with computer time. However, the time required for a person to outline each Plateau border in the projection images takes far longer. Reconstruction from these typically takes several minutes, although the algorithm is polynomial³ and complex foams may take up to ten times longer to reconstruct.

Table 7.1 summarises the differences between each method.

³This is a simple-minded implementation which makes most of the n^2 comparisons between n rays.

7.2 Improvements and future work

The ultimate goal of these optical tomography systems is to analyse complex foam behaviour in real time. This could have widespread use in any of the many fields where the characteristics of foam systems are important, as mentioned in chapter one. It is clear that this goal is infeasible for now, but as computers and video subsystems in particular become ever more powerful it is highly likely that much progress could be made towards this end in the next five to ten years. Given sufficient resources, it is probably possible today for a determined organisation to study a small liquid foam sample (of the order of ten bubbles) in real time by using a version of the optimised tomographic method described in chapter six.

No doubt this would require substantial computing power and probably several cameras and video digitisers. It may be too expensive for the results which such a system can achieve at the moment, though.

As far as smaller-scale research is concerned, there are many improvements which could be made to the systems described here. In fact, any one would be a good starting point for further research. Even for the same capital outlay, the equipment which can be supplied now for the same price as the 486/66 PC and video system used here is capable of five times the processing performance and full frame full motion video; this will improve the performance of the axial tomography method even with the same software. The confocal and optimised axial methods will benefit from the new very-high-resolution digital cameras which give superior image quality.

Tomographic reconstruction in general lends itself to parallel computation and the speed of reconstruction in the optical systems presented here

could be increased by the use of a parallel computing system. Traditionally transputers have been used for this sort of task, but with cheap PC hardware and fast ethernet cards readily available it seems that the most power for the least money could come from a PC cluster. Multiprocessor Pentium Pro motherboards perhaps running a free UNIX clone designed for symmetric multiprocessing can easily challenge much more expensive workstations for raw numeric performance.

There are improvements to be made in the software also. The back projection algorithm used for axial tomography is effective, but simple-minded and inefficient. A better algorithm could take account of the sparse nature of images of cellular networks. If the projection could be made orthographic by some means⁴ then there are some optimisations which could significantly increase the speed of the back projection algorithm. The ray paths in a parallel projection can be calculated much more quickly than those in a cone-beam because for each projection the direction of a ray only needs to be calculated once. Cone-beam rays must be recalculated on an individual ray basis. Also, the three-dimensional system can be simplified to a series of two-dimensional reconstructions for which fast algorithms are known. Since back-projection is essentially a convolution operation, it is possible to fast Fourier transform the projections and complete the convolution as a multiplication in Fourier space before transforming back to world space. The implementation of this algorithm in the cone-beam case has been hampered by the difficulty of efficiently performing the three-dimensional Fourier transform.

It is also possible to improve on the optimised axial tomography method. Instead of a user fitting points or straight lines to a foam image, parametric

⁴By using a parallel light source, for example.

curves like Bezier or B-splines which could capture the curvature of each Plateau border could be fitted to the image. As mentioned above, given an image of sufficient resolution and clarity this could be done automatically by computer. The optimised method seems to be the best candidate for use in a real time system which might be able to resolve types of dynamic foam. Similarly to the raw tomographic method, parallel computing hardware such as the PC cluster described above will greatly increase the speed of both image processing and also model reconstruction. Performance of the system will also be improved by the removal of a software bottleneck which exists in the implementation used here. As can be seen from the specification in chapter six, the search for clusters of Plateau border vertices is very inefficient. The search space may be reduced considerably by considering coherence between adjacent frames in similar fashion to the stereo viewing problem. However, any stereo algorithm would need to be modified to account for the angle of elevation of the camera. Stereo problems are often simplified by maintaining scan line coherence such that a plane through any scan line in one image also contains the same scan line from the other stereo image. This is not true in the axial tomography system, where scan lines from two adjacent images are not coplanar. This apparently minor difference adds considerable complexity to the cluster search problem.

Finally, the experimental rig used in axial tomography could be improved. Currently, the liquid foam resolution is adversely affected by diffraction effects from the glass tube in which it resides. Whereas in this project the bubbles studied have been located in the central region of the tube, it would be possible to resolve a cross-section of the whole tube if diffraction problems

were removed. Possibly this could be achieved by using a refractive index-matching liquid around the exterior of the tube, but the internal glass-air interface will still remain.

7.3 Concluding remarks

This work has demonstrated that optical tomographic systems can resolve the bubbles in small foam samples. The presence of Kelvin cells in real foams, as seen by the physicists of Trinity College Dublin [92], has finally been confirmed by tomographic analysis, as have the individual shapes in the Weaire-Phelan foam cell.

This work is still in the early stages. Not much is known about the behaviour or performance of optical systems, even though X-ray and gamma-ray tomography have been studied for decades. However, the optical systems have clear potential and it may eventually be possible to observe the structure of sufficiently complex foams to give further insight into the process of foam formation and collapse.

Paul Thomas June 9, 1997

Appendix A

Published papers

These papers and articles have been published during the course of this work [93, 94, 95, 96, 97].

Acknowledgements

The author would like to thank his supervisors, Dr. RC Darton and Dr. PB Whalley, Helene Grunnet-Jepsen and John Mooney for help and advice during the course of this work.

The author would also like to thank Shell Research BV for funding the project.

Bibliography

- [1] D Weaire.
Froths, foams and heady geometry.
New Scientist, pages 34–37, May 1994.
- [2] JA Glazier and D Weaire.
The kinetics of cellular patterns.
Journal of Physics—Condensed Matter, 4(8):1867–1893, 1992.
- [3] WM Williams and CS Smith.
A study of grain shape in an aluminium alloy and other applications of
stereoscopic microradiography.
Journal of Metals, pages 755–765, July 1952.
- [4] D Weaire and R Phelan.
Optimal design of honeycombs.
Nature, 367:123, January 1994.
- [5] P Hrma.
Model for a steady state foam blanket.
Journal of Colloid and Interface Science, 134(1):161–167, January 1990.
- [6] J Plateau.
*Statique Experimentale et Theoretique des Liquides Soumis aux Seules
Forces Moléculaires*.
Ghent, Paris: Gauthier-Villars, 1873.
- [7] R Aveyard, BD Beake, and JH Clint.
Wettability of spherical-particles at liquid surfaces.
Journal of the Chemical Society-Faraday Transactions, 92(21):4271–
4277, 1996.
- [8] ET Woodburn, LG Austin, and JB Stockton.
A froth-based flotation kinetic model.
Transactions of the Institution of Chemical Engineers A, 72:211, 1994.
- [9] AD Barber and EF Wijn.
Foaming in crude distillation units.

- In *Distillation 1979, proceedings of the Third International Symposium*, pages 30-1-30-21, 1979.
- [10] H Grunnet-Jepsen RC Darton and PB Whalley.
A correlation of foaming behaviour with surface properties.
In *The 1995 IChemE Research Event*, 1995.
- [11] C Herring, editor.
Metal Interfaces, page 108.
American Society for Metals, Cleveland, Ohio, 1952.
- [12] DA Reinelt and AM Kraynik.
Viscous effects in the rheology of foams and concentrated emulsions.
Journal of Colloid and Interface Science, 132(2):491-503, October 1989.
- [13] DA Reinelt and AM Kraynik.
On the shearing flow of foams and concentrated emulsions.
Journal of Fluid Mechanics, 215:431-455, 1990.
- [14] AM Kraynik, DA Reinelt, and HM Princen.
The nonlinear elastic behaviour of polydisperse hexagonal foams and concentrated emulsions.
Journal of Rheology, 35(6):1235-1253, August 1991.
- [15] RT DeHoff and GQ Liu.
On the relation between grain size and grain topology.
Metallurgical Transactions A, 16A:2007-2011, November 1985.
- [16] JE Avron and D Levine.
Geometry and foams: 2d dynamics and 3d statics.
Physical Review Letters, 69(1):208-210, July 1992.
- [17] C Sire.
Growth laws for 3D soap bubbles.
Physical Review Letters, 72(3):420-423, January 1994.
- [18] D Weaire and JA Glazier.
Relation between volume, number of faces and three-dimensional growth laws in coarsening cellular patterns.
Philosophical Magazine Letters, 68(6):363-365, 1993.
- [19] AM Kraynik.
Foam flows.
Annual Review of Fluid Mechanics, 20:325-357, 1988.
- [20] DJ Durian, DA Weitz, and DJ Pine.
Dynamics and coarsening in three-dimensional foams.
Journal of Physics—Condensed Matter, 2:SA433-SA436, 1990.

- [21] RA Leonard and R Lemlich.
A study of interstitial liquid flow in foam.
Journal of the American Institute of Chemical Engineers, 11(1):18–29,
January 1965.
- [22] M Barigou and JF Davidson.
Drainage of a soap film.
In *The 1993 IChemE Research Event*, pages 717–719, 1993.
- [23] AD Barber and S Hartland.
The effects of surface viscosity on the axisymmetric drainage of planar
liquid films.
Canadian Journal of Chemical Engineering, 54:279–284, August 1976.
- [24] S Hartland and AD Barber.
A model for a cellular foam.
Transactions of the Institution of Chemical Engineers, 52:43–52, 1974.
- [25] D Weaire, N Pittet, S Hutzler, and D Pardal.
Steady-state drainage of an aqueous foam.
Physical Review Letters, 71(16):2670–2673, October 1993.
- [26] G Verbist and D Weaire.
A soluble model for foam drainage.
Europhysics Letters, 26(8):613–634, June 1994.
- [27] S Hutzler, G Verbist, D Weaire, and JA van der Steen.
Measurement of foam density profiles using ac capacitance.
Europhysics Letters, 31(8):497–502, 1995.
- [28] G Verbist, D Weaire, and AM Kraynik.
The foam drainage equation.
Journal of Physics—Condensed Matter, 8:3715–3731, 1996.
- [29] AD Barber and S Hartland.
The collapse of cellular foams.
Transactions of the Institution of Chemical Engineers, 53:106–111, 1975.
- [30] D Weaire and JP Kermode.
Computer simulation of a two-dimensional soap froth I. method and
motivation.
Philosophical Magazine B, 48(3):245–259, 1983.
- [31] T Herdtle and H Aref.
Numerical experiments on 2-dimensional foam.
Journal of Fluid Mechanics, 241:233–260, 1992.
- [32] F Bolton and D Weaire.

- Rigidity loss transition in a disordered 2d froth.
Physical Review Letters, 65(27):3449–3451, December 1990.
- [33] M Tanemura, T Ogawa, and N Ogita.
A new algorithm for three-dimensional voronoi tessellation.
Journal of Computational Physics, 51:191–207, 1983.
- [34] D Bideau and JP Troadec.
Compacity and mean coordination number of dense packings of hard discs.
Journal of Physics C: Solid State Physics, 17:L731–L735, 1984.
- [35] W Thomson (Lord Kelvin).
On the division of space with minimum partitional area.
Philosophical Magazine Series 5, 24(151):503–513, December 1887.
- [36] HM Princen and P Levinson.
The surface area of Kelvin’s minimal tetrakaidecahedron: The ideal foam cell (?).
Journal of Colloid and Interface Science, 120(1):172–175, November 1987.
- [37] DA Reinelt and AM Kraynik.
Simple shearing flow of a dry kelvin soap foam.
Journal of Fluid Mechanics, 311:327–342, 1996.
- [38] DA Reinelt.
Simple shearing flow of three-dimensional foams and highly concentrated emulsions with planar films.
Journal of Rheology, 37(6):1117–1139, 1993.
- [39] DA Reinelt and AM Kraynik.
Large elastic deformations of three-dimensional foams and highly concentrated emulsions.
Journal of Colloid and Interface Science, 159(2):460–470, 1993.
- [40] D Weaire and R Phelan.
A counter-example to Kelvin’s conjecture on minimal surfaces.
Philosophical Magazine Letters, 69(2):107–110, 1994.
- [41] K Brakke.
The surface evolver.
Experimental Mathematics, 1:141–165, 1992.
- [42] KA Brakke.
The surface evolver and the stability of liquid surfaces.

- Philosophical Transactions of the Royal Society of London, Series A: Mathematical, Physical and Engineering Sciences*, 354(1715):2143–2157, 1996.
- [43] AM Kraynik and DA Reinelt.
The linear elastic behaviour of dry soap foams.
Journal of Colloid and Interface Science, 181(2):511–520, 1996.
- [44] AM Kraynik and DA Reinelt.
The linear elastic behaviour of a bidisperse Weaire-Phelan soap foam.
Submitted to the Professor Howard Brenner *Festschrift* volume of Chemical Engineering Communications.
- [45] MP Wernet and RV Edwards.
New space domain processing technique for pulsed laser velocimetry.
Applied Optics, 29(23):3399–3417, August 1990.
- [46] RC Chen and L-S Fan.
Particle image velocimetry for characterizing the flow structure in three-dimensional gas-liquid-solid fluidized beds.
Chemical Engineering Science, 47(13/14):3615–3622, 1992.
- [47] N MacCuaig, JPK Seville, WB Gilboy, and R Clift.
Application of gamma-ray tomography to gas fluidized beds.
Applied Optics, 24(23):4083–4085, December 1985.
- [48] PH Calderbank and J Rennie.
The physical properties of foams and froths formed on sieve-plates.
Transactions of the Institution of Chemical Engineers, 40:3–12, 1962.
- [49] RWH Sargent, WP Macmillan, PH Calderbank, and J Rennie.
The physical properties of froths and foams (comments).
Transactions of the Institution of Chemical Engineers, 40:191–193, 1962.
- [50] MB Rhodes and B Khaykin.
Foam characterization and quantitative stereology.
Langmuir, 2(5):643–649, 1986.
- [51] EB Matzke.
The three-dimensional shape of bubbles in foam—an analysis of the rôle of surface forces in three-dimensional cell shape determination.
American Journal of Botany, 33:58–80, January 1946.
- [52] RJ Foot, E Kocianova, CF Forster, and AJ Wilson.
An examination into the structure of stable foams formed on activated sludge plants.
Journal of Chemical Technology and Biotechnology, 56:21–25, 1993.

- [53] RM Morris and A Morris.
The measurement of interfacial areas of foams and froths by a radiographic technique.
Chemistry and Industry, pages 1902–1903, November 1965.
- [54] CP Gonatas, JS Leigh, AG Yodh, JA Glazier, and B Prause.
Magnetic resonance images of coarsening inside a foam.
Physical Review Letters, 75(3):573–576, 1995.
- [55] BA Prause, JA Glazier, S Gravina, and CD Montemagno.
Three-dimensional magnetic resonance imaging of a liquid foam.
Journal of Physics—Condensed Matter, 7:L511–L516, 1995.
- [56] JM Spivey.
The Z Notation: A Reference Manual, Second Edition.
Prentice Hall International (UK), 1992.
- [57] J Canny.
A computational approach to edge detection.
IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-8(6):679–697, November 1986.
- [58] C Arcelli, L Cordella, and S Levialdi.
Parallel thinning of binary pictures.
Electronics Letters, 11(7):148–149, April 1975.
- [59] T Toffoli and N Margolus.
Cellular Automata Machines.
MIT Press, 1987.
- [60] J Radon.
Über die Bestimmung von Funktionen durch ihre Integralwerte längs gewisser Mannigfaltigkeiten.
Berichte Saechsische Akademie der Wissenschaften, 69:262–277, 1917.
- [61] W Swindell and HH Barrett.
Computerized tomography: Taking sectional x-rays.
Physics Today, pages 32–41, December 1977.
- [62] AC Kak.
Computerized tomography with x-ray, emission and ultrasound sources.
Proceedings of the IEEE, 67(9):1245–1272, September 1979.
- [63] DG Grant.
Tomosynthesis: A three-dimensional radiographic imaging technique.
IEEE Transactions on Biomedical Engineering, BME-19(1):20–28, January 1972.

- [64] R Snyder and L Hesselink.
High speed optical tomography for flow visualisation.
Applied Optics, 24(23):4046–4051, December 1985.
- [65] GW Faris and RL Byer.
Three-dimensional beam-deflection optical tomography of a supersonic jet.
Applied Optics, 27(24):5202–5212, December 1988.
- [66] R Snyder and L Hesselink.
Measurement of mixing fluid flows with optical tomography.
Optics Letters, 13(2):87–89, February 1988.
- [67] GW Faris and HM Hertz.
Tunable differential interferometer for optical tomography.
Applied Optics, 28(21):4662–4667, November 1989.
- [68] EJ Beiting.
Fast optical absorption tomography.
Optics Letters, 16(16):1280–1282, August 1991.
- [69] W Fischer and H Burkhardt.
Three-dimensional temperature measurement in flames by multispectral tomographic image analysis.
SPIE Applications of Digital Image Processing XIII, 1349:96–105, 1990.
- [70] A Schwarz.
Tomographic temperature measurement of flames with the schlieren effect.
Technisches Messen, 60(7–8):289–295, 1993.
- [71] H Burkhardt.
Image analysis and control of combustion processes.
In S Sideman and K Hijikata, editors, *Proceedings of the International Seminar on Imaging in Transport Processes, Athens May 1992*. Begel House Inc., New York, 1993.
- [72] A Rougee, KM Hanson, and D Saint-Félix.
Comparison of 3-d tomographic algorithms for vascular reconstruction.
SPIE Medical Imaging II, 914:397–405, 1988.
- [73] Y Trouset, A Rougée, C Picard, and D Saint-Félix.
3-d reconstruction of high contrast objects from limited conic x-ray projections.
SPIE Science and Engineering of Medical Imaging, 1137:54–59, 1989.
- [74] Y Trouset, D Saint-Félix, A Rougée, and C Chardenon.

- Multiscale cone-beam x-ray reconstruction.
SPIE Medical imaging IV: Image Formation, 1231:229–238, 1990.
- [75] D Saint-Félix, C Picard, C Ponchut, R Roméas, A Rougée, and Y Trouset.
Three-dimensional x-ray angiography: First in vivo results with a new system.
SPIE Image Capture, Formatting and Display, 1897:90–98, 1993.
- [76] D Saint-Félix, Y Trouset, C Picard, C Ponchut, R Roméas, and A Rougée.
In Vivo evaluation of a new system for 3D computerized angiography.
Physics in Medicine and Biology, 39:583–595, 1994.
- [77] H Haneishi and N Ohyama.
3D blood vessel reconstruction by simulated annealing.
SPIE Medical Imaging IV: Image Processing, 1233:97–102, 1990.
- [78] S Kirkpatrick, CD Gelatt, Jr., and MP Vecchi.
Optimization by simulated annealing.
Science, 220(4598):671–680, May 1983.
- [79] GO Brown, ML Stone, and JE Gazin.
Accuracy of gamma ray computerized tomography in porous media.
Water Resources Research, 29(2):479–486, February 1993.
- [80] A Sasov.
Software and hardware for microtomography based on ibm-compatible computers.
SPIE Analytical Methods for Optical Tomography, 1843:272–277, 1991.
- [81] S Kawata, O Nakamura, and S Minami.
Optical microscope tomography. I. support constraint.
Journal of the Optical Society of America A, 4(1):292–297, January 1987.
- [82] O Nakamura, S Kawata, and S Minami.
Optical microscope tomography. II. nonnegative constraint by a gradient-projection method.
Journal of the Optical Society of America A, 5(4):554–561, April 1988.
- [83] S Kawata, O Nakamura, H Ooki, K Ogino, Y Kuroiwa, and S Minami.
Laser computed-tomography microscope.
Applied Optics, 29(26):3805–3809, September 1990.
- [84] CS Brown, DH Burns, FA Spelman, and AC Nelson.
Computed tomography from optical projections for three-dimensional reconstruction of thick objects.

- Applied Optics*, 31(29):6247–6254, October 1992.
- [85] MS Beck, BS Hoyle, MA Morris, RC Waterfall, and RA Williams, editors.
Process Tomography - 95 Implementation for Industrial Processes. University of Manchester Institute of Science and Technology, 1995.
- [86] BD Smith.
Cone beam convolution formula.
Computers in Biology and Medicine, 13(2):81–87, 1983.
- [87] BD Smith.
Image reconstruction from cone-beam projections: Necessary and sufficient conditions and reconstruction methods.
IEEE Transactions on Medical Imaging, MI-4(1):14–25, March 1985.
- [88] BD Smith.
Cone-beam tomography: Recent advances and a tutorial review.
Optical Engineering, 29(5):524–534, May 1990.
- [89] F Natterer.
Sampling in fan beam tomography.
SIAM Journal of Applied Mathematics, 53(2):358–380, April 1993.
- [90] TM Peters.
Algorithms for fast back- and re-projection in computer tomography.
IEEE Transactions on Nuclear Science, NS-28(4):3641–3647, August 1981.
- [91] LA Feldkamp, LC Davis, and JW Kress.
Practical cone-beam algorithm.
Journal of the Optical Society of America A, 1(6):612–619, June 1984.
- [92] D Weaire, S Hutzler, and N Pittet.
Cylindrical packings of foam cells.
Forma, 7:259–263, 1992.
- [93] PD Thomas, RC Darton, and PB Whalley.
Foam characterisation using a vision system.
In *The 1994 IChemE Research Event*, pages 713–715, 1994.
- [94] PD Thomas, RC Darton, and PB Whalley.
Non-invasive liquid foam structure analysis.
In *The 1995 IChemE Research Event*, pages 751–753, 1995.
- [95] RC Darton, PD Thomas, and PB Whalley.
Application of optical tomography in process engineering.

- In MS Beck, BS Hoyle, MA Morris, RC Waterfall, and RA Williams, editors, *Process Tomography—1995 Implementation for Industrial Processes*, pages 427–439, 1995.
- [96] PD Thomas, RC Darton, and PB Whalley.
Liquid foam structure analysis by visible light tomography.
The Chemical Engineering Journal, 56:187–192, 1995.
- [97] RC Darton, PD Thomas, and PB Whalley.
Optical tomography.
In DM Scott and RA Williams, editors, *Frontiers in Industrial Tomography*, chapter 7, pages 73–84. Engineering Foundation, New York, 1995.

