Tomographic imaging of foam


Beckman Institute, University of Illinois at Urbana-Champaign, Urbana, IL 61801
fetterma@uiuc.edu
http://www.phs.uiuc.edu

Abstract: The morphology of three-dimensional foams is of interest to physicists, engineers, and mathematicians. It is desired to image the 3-dimensional structure of the foam. Many different techniques have been used to image the foam, including magnetic resonance imaging, and short-focal length lenses. We use a camera and apply tomographic algorithms to accurately image a set of bubbles. We correct for the distortion of a curved plexiglas container using ray-tracing.

© 2000 Optical Society of America
OCIS codes: (100.6960) Tomography; (100.6950) Tomographic image processing

References and links

1 Introduction

Light’s interaction with soap bubbles creates colorful patterns that vividly illustrate the rudimentary principle of wave interference. But there is a lot more to learn about soap bubbles. In a cluster they serve as a model for many cellular systems occurring in nature. At low liquid content, they are organized into an intricate network of polyhedral foam adhering to certain geometric rules discovered by Plateau more than a century ago
Fig. 1. A CCD video image of polyhedral aqueous foam showing the network of vertices and edges. The camera is set at a large depth of field to reveal the interior features.

[1] Scientists are interested in how energy and entropy extremum principles determine the partition of space by soap bubbles. This motivates a technique capable of resolving the coordinates of vertices and edges of bubbles in foam.

Polyhedral aqueous foam made with soap solution looks like an open face structure because of soap film's transparency. The internal features are revealed to the extent that light rays can maintain straight paths before they are scattered and absorbed by the liquid borders. Fig. 1 shows the polyhedral network of vertices and edges captured by a video camera with a large depth of field.

Durian and his colleagues have developed a multiple light scattering technique to study foam [2]. It works by approximating light propagation through foam as a diffusion process. Light transmitted through a sample is measured and correlated with average bubble size. Durian's method has been demonstrated to work suitably with foam densely packed with small spherical bubbles of radius less than 50µm. However, it cannot provide information about the geometry of foam with polyhedral cells. To this end, attempts to measure the vertices' position and their connectivity would succeed by scanning a focal plane of a CCD camera, adjusted to small depth of field, through the layers of bubbles; internal features concealed in one direction are usually observable in other directions. Monnerau and Vignes-Adler have used this technique to reconstruct a cluster of up to 50 bubbles [3, 4]. The main disadvantage of this method is that image processing is needed to pick out the vertices from a set of noisy two-dimensional data slices. Vertices outside the focal plane are superimposed on data slices and thus make the resolution process difficult. Another approach to the foam imaging problem is to create a three-dimensional data volume by tomography. Magnetic resonance imaging (with tomographic algorithms) has been employed to examine the interior of foam with various degrees of success [5].

In this paper, we use Feldkamp's conebeam tomographic algorithms [6, 7, 8, 9] to reconstruct a three-dimensional foam. In the optical domain, we take many pictures of the object from different angles. These pictures are then processed by the conebeam algorithm to reconstruct the three-dimensional volume.

One advantage of this technique is that both the conebeam algorithm and the image processing both are designed to require no human intervention. This will speed up our rate of taking data, making it possible to analyze larger and more complex data sets. The time scale of this technique is also important. Currently we are taking 1 scan in a time of approximately 5 minutes, a time which could be reduced by taking images at video rate instead of still pictures. Since the time scale of the bubble development is on...
Fig. 2. Top. Experimental setup. The bubbles are in a cylindrical plexiglas container, that is placed on a rotation stage. The plexiglas container is held in a mount such that it is centered on the center of the rotation stage. A lightbox (a flat box with fluorescent lights that provides a uniform white light) is placed behind the cylinder, so that the camera sees the silhouetted image. The computer controls the rotation stage, and the computer also acquires images from the digital camera. Bottom. As shown above, the image rotates around on a stage, and the camera remains stationary. However, it is equivalent to view the object as stationary, while the camera rotates. The positions that the camera acquires an image at are referred to as the vertex points. The x, y, and z axes travel with the camera.

the order of hours, we will be able to take several images as the bubbles evolve.

An experimental problem that we encountered was that the container, a plexiglas cylinder, that held the bubbles distorted the ray path. Using ray-tracing we were able to compensate for this distortion. This distortion compensation may have applications to a wide range of tomography problems.

Our experimental results show a reconstruction of a test object, using the distortion correction algorithm. In future work, we will use a matched filter to extract the three-dimensional positions of the vertices.

2 Optical System Design and Algorithm

The experimental setup consists of an object mounted on a rotating stage. A schematic of the setup is shown in Fig. 2 (top). The bubbles are in a cylindrical plexiglas container, that is placed on a rotation stage. The cylinder had an index of refraction of $n = 1.49$, an inner radius of $r_{\text{inner}} = 2.54cm$, and an outer radius of $r_{\text{outer}} = 3.17cm$. The plexiglas container is held in a mount such that it is centered on the center of the rotation stage. A lightbox (used in photography, the lightbox is a flat box with fluorescent lights that provides a diffuse white light) is placed behind the object, so that the camera records the silhouette of the object. The computer controls the rotation stage, and the computer also acquires images from the digital camera.

Our goal is to image the edges of the bubbles, which should show as lines on a silhouette image. Under optimal lighting conditions, the edges will appear as lines, while the faces will appear transparent. However, we do observe some scatter from the
Fig. 3. Top. The angular notation used in this paper. Consider a particular voxel and vertex point. We write \( \alpha \) for the angle in the vertex plane, \( \beta \) for the angle normal to the vertex plane, and \( r \) for the vector that connects the center of the vertex path to the voxel. Bottom. Angular notation, continued. For a given image, recorded by a camera, \( \Psi_y \) and \( \Psi_z \) refer to the coordinates on the camera. It is necessary to find a mapping function between the camera coordinates \( \Psi_y \) and \( \Psi_z \), and the points in the reconstructed voxel space, which are denoted by the angles \( \alpha \), \( \beta \), and the vector \( r \).

faces of the bubbles.

As the object rotates through \( N \) steps, an image is taken at each step. Although the object is rotated, one may picture the object as stationary, and the camera as rotating about it Fig. 2 (bottom). Each camera position is referred to as a vertex point \( P_\phi \), where \( \phi \) describes the angle of the vertex point from the center of the vertex path. The vertex points are all in a circle. The algorithms described here do not require a circular vertex path, but we choose to use one for experimental simplicity. This circle, referred to as the vertex path, lies on the vertex plane. The point \( V \) is an arbitrary vertex point, which will be referred to later. The axes are defined such that the \( x, y, \) and \( z \) axes travel with the camera. The \( x \) axis points towards the center of the vertex path [8]. The \( y \) axis is in the vertex plane but normal to the vertex path, and the \( z \) axis is normal to the vertex plane.

The angular coordinate notation used in this paper is shown in Fig. 3. Consider a certain voxel, and the angles that it makes with respect to a vertex point such as \( V \). We refer to \( \alpha \) as the angle in the vertex plane, and \( \beta \) as the angle normal to the vertex plane. The coordinate \( r \) connects the center of the vertex path to the voxel. Note that for a given voxel, the values of \( \alpha \) and \( \beta \) change, depending on which vertex point we are considering, but the value of \( r \) remains constant.

For a given image, recorded by a camera, \( \Psi_y \) and \( \Psi_z \) refer to the coordinates on the camera. These coordinates are not angles, although each value of \( \Psi \) does correspond to an angle projecting out from the camera. It is necessary to find a mapping function between the camera coordinates \( \Psi_y \) and \( \Psi_z \), and the points in the reconstructed voxel space, which are denoted by the angles \( \alpha \), \( \beta \), and a distance coordinate.

Typical images are shown in Fig. 4. Fig. 4 (left) shows an image of a test object,
which is chosen to be the plexiglas cylinder with needles inside. We chose needles because we were interested in reconstructing fine objects. Each image has dimension 256x256 and we took \( N = 256 \) images at equal rotation angles. The image is backlit, so that our dataset is a silhouette of the object. Fig 4 (right) shows a picture of the bubbles (the dimensions of the cylinder used for the test object and for the bubbles were the same). The polygonal structure of the bubbles can be seen. The fine features of the bubbles makes it important to accurately reconstruct the image; otherwise it will be impossible to locate the edges and vertices.

The Feldkamp conebeam tomographic algorithm [6, 7, 8, 9] is used to reconstruct the volume. The original algorithm consists of two steps: filtering and backprojection. In the filtering step, we apply a high pass filter to each image \( P_\phi \). In the equation below, \( g \) is this filter function and \( \omega_{y0} \) is a windowing parameter defined by the user. The specific form of the function below [9] is slightly different than the original filter function [7].

\[
g_y(\Psi_y) = \int_{-\omega_{y0}}^{\omega_{y0}} d\omega |\omega| \exp(i\omega\Psi_y - 2|\omega|/\omega_{y0}) \tag{1}
\]

A similar equation is written for \( g_z \). We call \( P_\phi \) the function \( P_\phi \) after convolution with \( g \):

\[
P_\phi(\Psi_y, \Psi_z) = \int_{-\infty}^{\infty} d\Psi_y' \int_{-\infty}^{\infty} d\Psi_z' g_y(\Psi_y - \Psi_y')g_z(\Psi_z - \Psi_z') P_\phi(\Psi_y', \Psi_z') \frac{1}{\sqrt{1 + \Psi_y'^2 + \Psi_z'^2}} \tag{2}
\]

The source is then backprojected from this function.

\[
f(\tau) = \frac{1}{4\pi^2} \int_V d\phi \frac{d^2}{(d + \tau \cdot \hat{x}')} P_\phi(\tan \alpha, \tan \beta) \tag{3}
\]

In this equation, \( f(\tau) \) is the reconstructed voxel intensity, and \( \tau \) is a vector in real space that connects the center point to a given voxel. The length \( d \) is the vertex path radius and the unit vectors of the coordinate associated with the camera are \( \hat{x}' \), \( \hat{y}' \) and \( \hat{z}' \). The integral is taken over all of the vertex points.

The variables \( \alpha \) and \( \beta \) refer to the position of the voxel, as defined in Fig. 3, and they are given by:
\[
\alpha = \arctan \frac{\vec{y}' \cdot \hat{y}}{d + \vec{r} \cdot \hat{x}}
\] (4)

\[
\beta = \arctan \frac{\vec{z}' \cdot \hat{z}}{d + \vec{r} \cdot \hat{x}}
\] (5)

We refer to Eq.4 and Eq.5 as the mapping equations, because they define how the three-dimensional voxel space is to be mapped into the two-dimensional plane of the image. Writing Eq.3 as the tan of the arctan of the angle in Eq.4 may seem somewhat circular, but it is convenient to work with the angle \(\alpha\).

Eq.3 may be evaluated in two ways: the voxel-oriented method or the pixel-oriented method. In the voxel oriented method, which we use in this paper, every voxel is considered. Then, for each voxel, we sum over all pixels. In the pixel oriented method, the rays originating from each pixel are projected through space, and their intersection with the voxel space is calculated. The difference between the voxel-oriented method and the pixel-oriented method is one of computational preference and convenience only, and does not appear in the Feldkamp equations.

3 Distortion Compensation

Our goal was to image the bubbles, which have very fine features. The bubbles must be contained in a cylinder, and this cylinder distorts the light rays. Using ray tracing techniques, we compensate for this distortion, and recover the correct three-dimensional reconstruction. First we discuss the approach to compensating for a generalized distortion, and then we discuss the specific case of the plexiglas cylinder.

3.1 Distortion compensation for an arbitrary refractive index profile

In Fig. 5, we illustrate the case of a generalized (2-D) distortion. Assume that we have a known index profile \(n(x, y)\) that is completely contained within a circle \(C\) with radius \(R_c\). Outside this circle, the index of refraction is \(n = 1.0\). In Fig. 5 (top), we show the case when \(n(x, y) = 1\). Then we can connect the vertex point \(V\) and a voxel \(P\) with a straight line, and the equations described in the previous section apply.

Next we consider the situation when \(n(x, y)\) is a known function, as shown in Fig. 5 (bottom). Here, the blue circle represents a region where \(n(x, y)\) is modified (e.g. \(n(x, y) = 2\) in this region). We cannot compensate for this distortion by simply stretching the image. The reason is that distorted rays do not follow the same path as any of the undistorted rays. That is, the rays in the undistorted space contain a different set of voxel points than the rays in the distorted space. Therefore, no matter how the image is mapped, these rays will not coincide. We note that, for the special case of the cylinder, it may be possible to correct the reconstruction by shifting the apparent position of the vertex point and stretching the images, since the cylinder acts as a lens. However, we chose the approach described here because it applies to a more general class of arbitrary distortions \(n(x, y)\), and it is also more exact.

Under the distortion, the equations Eq.1, Eq.2, and Eq.3 will still be valid. However, we must change the mapping functions defined in Eq.4 and Eq.5. First, we will find the ray that connects \(P\) and \(V\). The point \(P'\) is defined as the intersection of this ray with the circle \(C\). The angle that is necessary for the mapping functions (in Eq.4 and Eq.5) is the angle between \(P'\) and \(V\).

The problem of connecting 2 points by ray-tracing, given an arbitrary refractive index profile, is solved by Fermat’s principle of least time [12]. To solve it numerically, we require that Snell’s Law [12] is satisfied at all interfaces, and that the ray intersects \(P\) and \(V\). This inverse problem can be approached with a simple searching algorithm.
Top. This illustrates the case of a generalized (2-D) distortion. Assume that we have a known index profile \( n(x, y) \) that is completely contained within a circle \( C \) with radius \( R_c \). Outside this circle, the index of refraction is \( n = 1.0 \). Here, we show the case when \( n(x, y) = 1 \). Then we can connect the vertex point \( V \) and a voxel \( P \) with a straight line, and the equations described in section 2 apply.

Bottom. Consider the situation when \( n(x, y) \) is a known function. In this case, the blue circle could represent a region where \( n(x, y) = 2 \); everywhere else, \( n(x, y) = 1 \). Using Snell’s Law and numerical iteration, we find the ray that connects \( P \) to \( V \). Starting at a voxel \( P \), and given a direction vector, we find the intersection of the ray with the circle \( C \) at \( P' \). Then the angles (for Eq.4 and Eq.5) can be found from the points \( M \) and \( P' \).
We start from the voxel, $P$, and assume a direction vector. We shoot a trial ray into the volume, and measure the distance between the ray and the vertex point $V$. Then we iterate the initial direction vector until the ray intersects the vertex point.

### 3.2 Distortion compensation for the specific case of the plexiglas cylinder

The plexiglas cylinder is easy to model because it is an exact circle in the $x-y$ plane. Thus, we can solve for the exact angles, rather than modeling the refractive index profile on a grid. Assume that the cylinder has inner radius $R_1$, outer radius $R_2$, and index of refraction $n$. Because the cylinder has no curvature in the plane normal to the vertex plane, Eq.5 remains unchanged. Eq.4 must be changed because the distortion will alter the angle $\alpha$. This is now a two-dimensional problem.

Fig. 6 is a diagram in which we show one step in the construction used to solve for the ray path. This diagram represents the general case in which we have a voxel with position $\text{\rightarrow} r$ inside a circle, and a direction vector. We wish to find the point $N'$ at which this ray will intersect the circle, as well as the output direction vector. The point $N'$ is found by following the initial direction vector until the circle is intersected. The radius connecting $N'$ with the center of the circle $C$ makes a $90^\circ$ angle with the tangent to the circle. Thus, we can find the angle $\gamma$, and then using Snell’s Law, find the angle $\chi$. In this case we take the inner index of refraction as $n_1$ and the outer index of refraction as $n_2$. Note that we are considering a cylinder with a given thickness, so that we will repeat this calculation twice. The first time, we will assume that the cylinder has $n_1 = 1.0$ (air) inside, and $n_2 = 1.49$ (plexiglas) outside. The second time, after finding the point $N'$, we will then assume that the cylinder has plexiglas inside and air outside. The point at which this ray intersects the x-axis is then found.

In Fig. 7, we show a ray tracing diagram with several rays plotted. This represents the result of the calculation described above, as well as the searching algorithm to find
the correct direction vector that will intersect the vertex point.

4 Results and Analysis

In Fig. 8 (left), we show the three-dimensional reconstruction of the test object from Fig. 4 (left). This reconstruction is done without correcting for the distortion. It can be seen that the features are somewhat blurry. Fig. 8 (right) shows the substantial improvement when the correction algorithm is applied. Fig. 9 (left) is a slice of the dataset (without distortion correction) that is normal to the $\hat{z}$ axis, and Fig. 9 (right) is a similar slice of the dataset (with distortion correction). The distortion will be more significant for points that are closer to the edge of the cylinder. As shown in Fig. 7, the rays from such edge points are modified more than points that are closer to the center. This effect can be seen in Fig. 9 (left), where points farther away from the center appear as blurred crosses, but points closer to the center of the cylinder appear as sharper points.

We then applied this algorithm to the case of the bubbles, using the input data shown in Fig. 4 (right). Fig. 10 (left: uncorrected and right: corrected) shows the result with and without the distortion correction algorithm. Clearly, after applying the correction algorithm, the image is substantially improved. Slices of the bubble dataset are shown
Fig. 9. Left. This is a slice through the dataset of Fig. 8, which is a reconstruction of the needles test object without applying the distortion correction. The needles, which should appear as points, in this case appear as blobs. Some of the features appear as crosses. Right. The distortion correction algorithm is applied. The cross section of the needles now appear as points.

Fig. 10. Left. Reconstructing the bubbles without applying the distortion algorithm. Only a slice of the image from Fig. 4 is reconstructed. Right. The image is improved through application of the distortion algorithm.
Fig. 11. Top. This is a slice through the dataset of Fig. 10, which is a reconstruction of the bubbles without applying the distortion correction. There is some blurring in this image. Bottom Left. The distortion correction algorithm is applied, improving the bubbles images. Bottom Right. An erosion algorithm is applied to reduce each blob in the image at left (the corrected bubbles images) to a single point.
in Fig. 11 (top: uncorrected and bottom left: corrected). As in the example of the needles, the closer the points are to the center of the cylinder (in the uncorrected image), the less they are distorted. For studying the foam, it will be important to have a complete set of data that includes the points closest to the cylinder edge, so that this correction is necessary. The improvement with this correction algorithm seems quite clear, although the corrected image still shows some blurring for points close to the edge of the cylinder. The image may be further improved with image processing. In Fig. 11 (bottom right), we show an erosion [13] algorithm applied to the data of Fig. 11 (bottom left).

5 Conclusions

In this paper we show reconstruction of a three-dimensional foam. The foam is reconstructed using tomography algorithms. Using an algorithm that incorporates ray tracing, we are able to compensate for the distortion induced by a plexiglas cylinder. It is shown that this algorithm improves the images for the case of a test object, as well as for the bubbles. This distortion correction algorithm may be useful in various areas of tomography.

One area of future work will include improving the initial images. This could include redesign of the container, with thinner walls, to reduce or eliminate the distortion problem. Although this would improve our foam imaging, it is noted that part of our goal was to study the computational correction of optical distortion. The illumination of the Plateau borders could also be improved. In [14], the authors dissolve fluorescein salt in their foaming solution and illuminate with ultra-violet light. With this technique, the Plateau borders fluoresce and there is no stray light.

We are currently analyzing the 3-dimensional data set to reveal the exact polyhedral configuration and its evolution. This work includes signal processing techniques such as matched filtering.

We thank the reviewers for helpful comments. E. Tan, S.T. Thoroddsen, and J.M. Sullivan are supported by NASA Grant NAG3-2122 under the Microgravity Fluid Physics Program.