The Computation of Cloud-Base Height from Paired Whole-Sky Imaging Cameras

MARK C. ALLMEN* AND W. PHILIP KEGELMEYER JR.

Sandia National Laboratories, Livermore, California

(Manuscript received 30 September 1994, in final form 31 July 1995)

ABSTRACT

The authors have developed a novel approach to the extraction of cloud-base height (CBH) from pairs of whole-sky images. The core problem is to spatially register cloud fields from widely separated whole-sky imaging (WSI) cameras; this complete triangulation then provides the CBH measurements. The wide camera separation (necessary to cover the desired observation area), occluded regions, and the self-similarity of clouds defeats standard matching algorithms when applied to static views of the sky. The authors address this with an approach that is based on optical flow methods, exploiting the fact that modern WSIs provide time-ordered sequences of images. The authors will describe the algorithm for CBH determination, a confidence metric, as well as a method to correct for the severe projective effects on cloud shape induced by the WSI camera. Finally, the authors present the performance as evaluated both on real data validated by ceilometer measurements and on a variety of simulated cases.

1. Introduction

Cloud-base height (CBH) is a dominant factor in determining the infrared radiative properties of clouds. However, cloud-base heights are not well known for large areas, as they are difficult to measure. To address this fact, this paper presents a novel approach for the extraction of cloud-base height from pairs of whole-sky imaging (WSI) cameras. Given two images from widely separated WSI cameras, the core problem of computing cloud-base height is to find corresponding points between the two images. Correlation of intensity values is a standard technique for finding the corresponding point along the epipolar line. However, the wide camera separation (necessary to cover the observation area required by the field measurements site), occluded regions, and the self-similarity of clouds may defeat this approach when applied to static views of the sky.

To compute CBH we exploit the fact that modern WSIs provide time-ordered sequences of images. With these sequences, the optical flow field—that is, a field of vectors that represent the image motion of points, can be recovered and used to aid in finding corresponding points. Specifically, we augment the correlation ap-

approach to include this motion information along with the intensity information at each pixel.

The following sections will motivate this problem and the use of WSI cameras, describe the central principle of the height extraction algorithm, discuss in detail how to overcome the effects of the WSI camera projection on cloud shape and motion, and present quantitative measures of the CBH algorithm performance as evaluated on real and simulated data.

a. Background

1) GENERAL CIRCULATION MODELS AND CLOUDS

A major goal for the Department of Energy’s (DOE’s) global change efforts is to improve the accuracy of general circulation models (GCMs) capable of predicting the timing and magnitude of greenhouse-gas-induced global warming. Research has shown cloud radiative feedback is the single most important feedback effect determining the magnitude of possible climate responses to human activity. Yet, as pointed out by Cess (Cess et al. 1989), clouds are not well parameterized in GCMs and are, in fact, currently the greatest factor limiting the accuracy of atmospheric GCMs. Thus, clouds exert the largest influence while at the same time present the largest uncertainties in predicting global climate change. As a result, cloud studies are critical to understanding global climate change and improving the predictive accuracy of GCMs. In recognition of this problem (e.g., Rossow et al. 1985), a number of important national and international programs have recently been initiated to characterize cloud–radiation interactions, including DOE’s Atmo-
pheric Radiation Measurement Program (ARM), and the International Satellite Cloud Climatology Project.

In the ARM program a key to this cloud-radiation characterization is the effective treatment of cloud formation and cloud properties in GCMs as supported by a field measurements program—\"an important feature of the ARM Program Plan is to establish a surface-based cloud imaging system at the research sites that will provide appropriate information for parameterizing solar flux over an entire grid cell.\" The first such Cloud and Radiation Testbed (CART) site makes measurements, including cloud measurements, over a 30-km-diameter region.

2) MACROSCOPIC CLOUD PROPERTIES

A well-recognized approach to reducing the uncertainties associated with cloud-radiation interactions involves measuring the macroscopic properties of clouds (shape, size, extent, cloud cover fraction, radiance, altitude, etc.) on the mesoscale (20–200 km). Of these measurements, cloud base height is particularly important because it is a dominant factor in determining the infrared radiation from clouds to the lower atmosphere and the earth’s surface. Furthermore, as shown by Rossow (Rossow et al. 1985), base heights are essential to measuring the cloud cover fraction at low, medium, and high altitude, and these in turn are needed to establish a cloud-radiation climatology.

3) WHOLE-SKY IMAGERS AND DIGITAL IMAGERY

Digital whole-sky imagers have a number of desirable features for making cloud studies. They are passive and therefore relatively inexpensive and reliable, can be used in unattended operation, and obtain images of the entire whole-sky dome rapidly. The particular passive camera that has generated our existing data is the Whole-Sky Imager developed by the Marine Physical Laboratory (MPL) at the Scripps Institution of Oceanography, as described in Shields (Shields et al. 1990a). These imagers are rugged and have demonstrated many years of high-reliability field service. Full-resolution (1/3° angular resolution) digital images can be acquired at one per minute. This is rapid enough to capture most of the cloud dynamics of interest and fully utilize the image motion of the clouds in our optical flow field approach.

b. Method

1) AN OVERVIEW

Unfortunately, the state of the art in cloud imagery processing is not yet capable of extracting measures as central and important as cloud-bottom heights. The main problem is to spatially match up cloud fields from widely separated WSI cameras; once registered against each other, computation of cloud-bottom heights proceeds in a straightforward fashion from triangulation and knowledge of the camera locations. Our solution to this problem utilizes temporal flow fields from each camera separately. In the following subsections we will review the prior history of this problem, illustrate its difficulty, and suggest why flow fields provide the necessary additional constraints.

2) PRIOR WORK

Extracting cloud-bottom heights via triangulation of registered points has been in the cloud stereoscopy literature for 20 years and is well understood. The registration itself, however, has not been well addressed. In the earlier literature (Bradbury and Fujita 1968; Lyons 1971), the problem was side-stepped though human intervention, registering the images by hand before triangulation. More recently (Rocks 1987), the automatic registration problem has been successfully handled, but only for nearly parallel views of the sky (i.e., cameras spaced closed together). In that case the stereoscopic nature of the views permitted simple limited-displacement correlation to suffice as a registration algorithm.

The registration problem facing ARM, however, is considerably harder in that a camera spacing on the order of 5 km is required to achieve adequate coverage at the required resolution with a small number of cameras. With this baseline spacing, the three-dimensional nature of clouds generates occlusion and perspective effects that will cause them to image differently at the various cameras. Because of this, correlation-based registration using intensity only is insufficient. Further, the visual self-similarity of clouds defeats token matching (the detection and matching of a small number of visually distinctive regions), the only common alternative approach.

As an example, Fig. 1 contains a pair of simultaneous frames. The center of each image is the point directly overhead of the WSI and the edge of the image shows points near the horizon (80° from vertical). Careful examination will show that corresponding points appear shifted down and to the left in the right image. This can be difficult to determine by eye, primarily due to the difference in perspective experienced by the widely separated cameras. However, additional information provided by the motion in a sequence can be used to help find corresponding points.

3) FLOW FIELD CORRELATION

The use of image sequences to identify the correspondence suggests how one can automate the registration process. WSI images are acquired at a rate of one per minute, which provides a comparatively dynamic view of the sky. This provides a means to overcome the registration obstacles mentioned before, which apply only to the attempt to register two static views of the sky.
An optical flow field is a two-dimensional vector field where the vector at each pixel indicates the motion of that point between frames of an image sequence. The temporal sampling rate of the WSI camera is high enough that optical flow fields can be computed using hierarchical correlation methods (Burt 1984). With the optical flow fields computed for the images from both WSI cameras, each image pixel becomes associated with a vector indicating the image motion of that point. Combining this with the intensity value at the pixel, a multidimensional quantity is associated with each image pixel. It is the core of our approach that these flow and intensity fields from the separated WSIIs are jointly registered against each other. In this way, the additional constraints provided by the flow field are exploited to make the matching unique.

Vectors in optical flow fields are typically two-dimensional, $\mathbf{V} = (\Delta x, \Delta y)$, with $\Delta x$ and $\Delta y$ indicating the direction of motion and the length, $\|\mathbf{V}\|$, indicating the speed. However, when using flow fields to find matching points between corresponding images, the correlation needs to be normalized so that areas with large motion are not unjustly favored. But by normalizing, the speed, or length, of two-dimensional vectors becomes meaningless. An alternative to using optical flow vector length to represent speed is to convert each two-dimensional flow vector into a three-dimensional unit vector $\mathbf{V}_3$:

$$\mathbf{V}_3 = \frac{(\Delta x, \Delta y, t)}{[(\Delta x)^2 + (\Delta y)^2 + t^2]^{1/2}},$$

where $t$ is the time between image frames. The third component of $\mathbf{V}_3$ represents the speed that varies from $t$ to 0 as $\|\mathbf{V}\|$ varies from 0 to infinity. By using the three-dimensional flow vectors, the speed information is retained in the third component even with normalized correlation.

2. Extracting cloud-base heights

In this section the equations describing the geometry between WSI sites will be shown. This includes the equation for computing the height of a point given the corresponding points in two images and the equations to generate the epipolar line. The uncertainty and sensitivity of these equations and a confidence metric will also be examined.
Fig. 2. Illustration of reverse mapping as a confidence measure. (a) Let \( p \) be a point in the left WSI image. The epipolar line for point \( p \) is computed in the right WSI image and the best matching point, \( p' \), along this epipolar line is found. (b) The epipolar line for point \( p' \) is computed in the left WSI image. The best matching point, \( p''\), along this epipolar line is found. In the ideal case, \( p \) and \( p''\) are the same point. The distance between points \( p \) and \( p''\) is the confidence value.

\( a. \) Height and epipolar line equations

The position of points in the sky are defined by two variables: zenith angle (the angle between the vertical and a point) and azimuth angle (the angle between north and a point). Given a point in an image, the relationship between the point’s location \((x, y)\) in the image and its azimuth \(\phi\), and zenith \(\theta\) is given by the following camera calibration equations (Koehler and Shields 1990):

\[
z = \left[ 1.25(x - 255) \right]^2 + (y - 240)^2 \right]^{1/2} \tag{1}
\]

\[
\theta = 585 \left[ 1 - \left( 1 - 0.001128z \right)^{1/2} \right] \tag{2}
\]

\[
\phi = \tan^{-1} \left[ \frac{1.25(x - 255)}{y - 240} \right] \tag{3}
\]

where \( z \) is the distance from the center of the image to the projected point, and \( x \) and \( y \) specify the location of the point in the image.

Given a point in one image, Eqs. (1) – (3) are used to compute the azimuth \(\phi_1\) and zenith \(\theta_1\). The epipolar line in a corresponding image, the line along which the matching point will be located is then given by the following two equations:

\[
\phi_2 = \arctan \left[ \frac{\tan(\phi_1) - \frac{d}{r \sin(\theta_1) \cos(\phi_1)}}{r \sin(\theta_1) \cos(\phi_1)} \right] \tag{4}
\]

\[
\theta_2 = \arctan \left[ \frac{r \sin(\theta_1) \sin(\phi_2) - d}{r \cos(\theta_1) \sin(\phi_2)} \right] \tag{5}
\]

where \( r \) is the distance from the first camera to the point in the sky and \( d \) is the distance between the two cameras. As \( r \) varies from zero to infinity, the epipolar line in the image from the other camera is swept out.

Once corresponding points have been found, Eqs. (1) – (3) are used to compute the azimuth \(\phi_2\) and zenith \(\theta_2\) of the matching point in second camera. The height \( h \) of the point is then given by

\[
h = \frac{d}{\tan(\theta_1) \sin(\phi_1) - \tan(\theta_2) \sin(\phi_2)} \tag{6}
\]

The following sections will describe a confidence metric for the computed base heights and examine the sensitivity and uncertainty in the equations.

\( b. \) A confidence metric

Given that the WSI cameras are widely separated, it is possible that many cloud points will be visible in one camera but not the other. When this happens, the algorithm for finding corresponding points between images will nonetheless report a matching point (whichever point best matches), but the resulting computed height will most likely be incorrect. In cases where the corresponding point does not exist, a height should not be computed and the algorithm should report this. To detect cases where a cloud point is only visible in a single image, we describe a confidence metric that indicates the confidence of the algorithm that it has found two truly corresponding points.

To compute the confidence, we effectively find corresponding points twice. First, corresponding points for the points in an image from the left WSI are found. Second, corresponding points for the points in an image from the right WSI are found. Let \( p_1 \) be a point in the left image. The epipolar line for point \( p_1 \) is computed in the right image. The best matching point along this epipolar line is found. Call this point \( p'_1 \). This is shown in Fig. 2a. Next, the epipolar line for point \( p'_1 \) is computed in the left image. The best matching point along this epipolar line is found. Call this point \( p''_1 \). In the ideal case, \( p_1 \) and \( p''_1 \) are the same point. The distance between points \( p_1 \) and \( p''_1 \) is the confidence value. So the smaller the value, the higher the confidence of the algorithm. This is shown in Fig. 2b. Confidence metrics very similar to ours have been developed by others (Fua 1993; Hannah 1989).

If there is no correct corresponding point to \( p_1 \) in the right image, then the point labeled as \( p'_1 \) is clearly not the correct corresponding point. When the corresponding point for \( p'_1 \) is found in the left image, it is more likely that its correct corresponding point will be found rather than \( p_1 \) and the confidence of the computed height of \( p_1 \) will be low.
In practice the confidence metric is used as a confidence threshold. As described in section 5 the only points that are shown in the results are those where the confidence metric is less than 2.1. If the confidence metric is greater than 2.1, then it is assumed that the correct matching point was not found. So a point with a confidence metric value of, say, 5.0 is no more confident than a point with a confidence metric value of, say, 20.0.

Now the overall cloud base height algorithm, with the confidence computation, is as follows:

- Compute the optical flows from WSI 1 and WSI 2 [section 1b(4)].
  - For every point \( p_i \) in image 1, the image from WSI 1.
  - Compute the epipolar line for \( p_i \) in image 2, the image from WSI 2 [Eqs. (4) and (5)].
  - For each point \( p_j \) along the epipolar line, match \( p_i \) and \( p_j \) (section 3e).
  - Find the point of maximum correlation value along the epipolar line. Call this point \( p_i' \).
  - Compute the epipolar line for \( p_i' \) in image 1 [Eqs. (4) and (5)].
  - For each point \( p_j \) along the epipolar line in image 1, match \( p_j \) and \( p_i' \) (section 3e).
  - Find the point of maximum correlation value along the epipolar line. Call this point \( p_i'' \).
  - Compute the cloud height using \( p_i \) and \( p_i'' \) as corresponding points [Eq. (6)].
  - Compute the distance between \( p_i \) and \( p_i'' \). This is the confidence value of point \( p_i \).

Given two points from corresponding images, how to determine how well they match will be discussed in section 3.

c. Sensitivity and uncertainty

There are five sources of uncertainty in the system: 1) the relative elevation of the WSIs, 2) the positioning of the WSIs, 3) the alignment of the WSIs relative to north, 4) the plumbness of the WSIs, and 5) the projective properties of the WSIs. These five sources of error are analyzed at length in (Allmen and Kegelmeyer 1994); the results are shown here.

Typical values for the sources of uncertainty and cloud base height were chosen and the total potential displacement of pixels in an image was computed. The following values were used:

- cloud height: 10 000 m
- uncertainty in \( x \) position: 100 m
- uncertainty in \( y \) position: 10 m
- uncertainty in elevation: 1°
- alignment uncertainty: 1°
- uncertainty in plumbness: 0 pixels if zenith is less than 65°
- uncertainty in calibration equation: 0.5 pixels if zenith is 65°–75°
- 1 pixel if zenith is greater than 75°
While in practice one source of error could counteract another source of error, in this example all errors were cumulative. Figure 3a shows displacement error, with bright indicating more error. The brightest section near the center has pixel displacement of 8, whereas the dark area also near the center of the image has pixel displacement of 3. Only the displacement due to a difference in elevation between the cameras is not significant. Roughly, all other sources contribute equally to the displacement shown in Fig. 3a.

When searching along the epipolar line, the size of the correlation window is $19 \times 19$ (see section 5), so it is likely that the correct match can be found at a displacement of 3, but a displacement of 8 will make finding the correct match difficult.

The sensitivity, the amount that a point could vertically move without being detected, is shown in Fig. 3b. The brightest area has error up to 5% (500 m for this scenario), while the darkest area has error around 1.5% (150 m for this scenario).

3. Determining the common field of reference

Because of the projective effects of the WSI camera, images from separate WSIs must be transformed so that shape and velocity of the same cloud appear the same in both WSIs. Consider the case where a cloud moves horizontally over a WSI with constant velocity. In the fish-eye WSI view, radial distance from the center of the image is proportional to the zenith angle. Therefore, as the cloud enters the field of view of the WSI at high zenith angle, a unit displacement of the cloud in the scene produces a smaller displacement in the image than if the cloud were at small zenith angle. So when the cloud of uniform velocity enters the field of view it moves slowly, accelerates as it passes overhead, then decelerates as it approaches the horizon. Similarly, the shapes of clouds are compressed as the zenith angle increases.

For movement and shape in the image to reflect the movement and shape in the scene regardless of zenith angle, one transforms the WSI image into pseudo-Cartesian coordinates (Koehler and Shields 1990). In this section we show that the pseudo-Cartesian transformation (PCT) results in an image sequence where a horizontal motion in the scene results in an image motion (or flow vector) that is independent of where that point projects into the image; that is, it is independent of zenith angle.

a. The pseudo-Cartesian transformation (PCT)

To counteract the projective effects of the WSI, the PCT rescales distance from the center of the image to a dependence that varies linearly with the tangent of the zenith angle. Let $R_{\text{wsi}}$ and $R_{\text{pc}}$ be the distance from the center of the image to a point in WSI and PCT coordinates, respectively. Let $\theta$ be the zenith angle. The coordinate transformations are defined as follows (Koehler and Shields 1990):

\[
\text{WSI: } R_{\text{wsi}} = \theta (3.032 + 0.00259\theta) \tag{7}
\]

\[
\text{pseudo-Cartesian: } \theta = \arctan \left( \frac{R_{\text{pc}} \tan(65^\circ)}{235} \right) \tag{8}
\]
Note that Eq. (7) is the WSI calibration equation, Eq. (1), rewritten. The choice of \("\tan(65^\circ)\)" and \("235^\circ\) in Eq. (8) is arbitrary and will be discussed in section 3d.

The procedure for constructing the PCT from the fish-eye image begins with identifying a pixel \(p_{\text{wsi}}\) at coordinates \((x_{\text{wsi}}, y_{\text{wsi}})\) in the PCT image. The zenith of the point is found using Eq. (8). The computed zenith is then substituted into Eq. (7) to find the distance \(R_{\text{wsi}}\) that the point is from the center of the image. That pixel is then copied to \((x_{\text{pct}}, y_{\text{pct}})\) in the PCT image. The azimuth is not changed by the PCT. This procedure is performed for all pixels in the PCT image.

Figure 4a shows an image before PCT. Figure 4b shows the same image after PCT. The "stretching" of the image away from the center is clearly visible. This stretching counteracts the compression of the fish-eye camera.

When finding corresponding points between pairs of images, we need to perform the PCT on small neighborhoods so that the neighborhoods are in the same frame of reference. Let \(p_{\text{wsi}}\) be a point, around which a neighborhood is to be PCT-ed. The zenith of \(p_{\text{wsi}}\) is found using Eq. (7). The computed zenith is then substituted into Eq. (8) to find the distance \(R_{\text{wsi}}\) giving \(p_{\text{pct}}\) (see Fig. 5a). Each point in the neighborhood around \(p_{\text{wsi}}\) has its distance from the center of the image computed \(R_{\text{wsi}}\), and substituted into Eq. (8) to find the zenith of that point. Equation (7) is used then to find the point in the WSI that maps to it. So the mapping is first from WSI to pseudo-Cartesian then from pseudo-Cartesian to WSI. This is shown by the arrows in Fig. 5a. One cannot simply PCT each point around \(p_{\text{wsi}}\) to obtain the neighborhood around \(p_{\text{pct}}\), because not every point in the neighborhood around \(p_{\text{wsi}}\) will necessarily have a value mapped to it.

\[ x' - x_1 = x_2' - x_2 \]  
\[ y' - y_1 = y_2' - y_2. \]

For a horizontal shape in the sky we need to show how to make the resulting shape in two WSI images identical. Without the pseudo-Cartesian transformation, the resulting shapes are clearly not equal. Let the position of a cloud point at two times be \(p\) and \(p'\). Let the azimuth and zenith of \(p\) with respect to site 1 be \(\phi_1\) and \(\theta_1\), respectively. Let the azimuth and zenith of \(p'\) with respect to site 2 be \(\phi_2\) and \(\theta_2\), respectively.

If it can be shown that the displacement between any two points is the same in PCT-ed images from two WSI sites, then the shape of clouds will appear the same in both images. Therefore, it needs to be shown that

\[ x' = 255 + 0.8R \sin(\phi) \]  
\[ y = 240 + R \cos(\phi), \]

where

\[ R = 235 \frac{\tan(\theta)}{\tan(65^\circ)}. \]

Substituting Eqs. (11) and (13) into Eq. (9) gives

\[ x'_1 - x_1 = \left[ 255 + 0.8R \sin(\phi'_1) \right] \]
\[ - \left[ 255 + 0.8R \sin(\phi_1) \right] \]
\[ = \frac{(0.8)(235)}{\tan(65^\circ)} \left[ \tan(\theta'_1) \sin(\phi'_1) \right] \]
\[ - \tan(\theta_1) \sin(\phi_1). \]

Fig. 5. (a) Illustration of WSI–PCT mapping. (b) Illustration of PCT expansion and compression. (a) If two neighborhoods are to be compared they must be PCT-ed first. The neighborhood around \(p_{\text{wsi}}\) is one such neighborhood. To PCT it, the center point, \(p_{\text{wsi}}\) is PCT-ed giving \(p_{\text{pct}}\). For each point around \(p_{\text{wsi}}\), the point that maps to it from around \(p_{\text{wsi}}\) is then found. The solid arrow shows the mapping from WSI to PCT. The dashed arrow shows the mapping from PCT to WSI. (b) Two points, \(p_1\) and \(p_2\), from different WSI images, are to be PCT-ed and then correlated. The neighborhoods after PCT are shown in gray. The two resulting gray neighborhoods are the same size so can be correlated, but in the left image the WSI neighborhood expanded, whereas in the right image the WSI neighborhood contracted.
Similarly,

\[
x'_2 - x_2 = \frac{(0.8)(235)}{\tan(65^\circ)} \left[ \tan(\theta'_2) \sin(\phi'_2) \right.
\]
\[
- \tan(\theta_2) \sin(\phi_2) \left] \right. \tag{15}
\]

The equations describing the relationship between sites 1 and 2 [Eqs. (4) and (5) rewritten] are given by

\[\phi'_2 = \arctan \left[ \frac{d}{h \tan(\theta'_2) \cos(\phi'_2)} \right] \tag{16}\]

\[\theta'_2 = \arctan \left[ \frac{h \tan(\theta'_2) \sin(\phi'_2) - d}{h \sin(\phi'_2)} \right] \tag{17}\]

Substituting Eq. (17) into Eq. (15) gives

\[
x'_2 - x_2 = \frac{(0.8)(235)}{\tan(65^\circ)} \times \left[ \frac{h' \tan(\theta'_1) \sin(\phi'_1) - d}{h' \sin(\phi'_1)} \right.
\]
\[
- \frac{h \tan(\theta'_1) \sin(\phi'_1) - d}{h \sin(\phi'_1)} \left] \right. \tag{18}\]

Assuming that \(h' = h\)—that is, assuming that the shape in the scene is horizontal—

\[
x'_2 - x_2 = \frac{(0.8)(235)}{\tan(65^\circ)} \times \left[ \frac{h \tan(\theta'_1) \sin(\phi'_1) - h \tan(\theta_1) \sin(\phi_1)}{h} \right.
\]
\[
- \tan(\theta_1) \sin(\phi_1) \left] \right. \tag{14}\]

which equals Eq. (14).

A similar procedure is performed for the y component. By showing that the displacement between any two points is the same in PCT-ed images from two WSI sites, we have shown that the shape of horizontal features in the sky will appear the same in both images, and thus that PCT will permit cloud shapes to be properly matched.

c. PCT-ing flow vectors

In this section it is shown that if WSI images are PCT-ed and flow fields computed, then corresponding vectors will be identical. However, optical flow fields are computed on un-PCT-ed images; therefore, the resulting flow fields must be PCT-ed so that corresponding flow vectors are equivalent. By PCT-ing the flow fields, corresponding vectors will be equal.

There are two ways that flow vectors can be altered: they can be repositioned and their length and/or orientation can be changed. In the next subsection it is shown that PCT-ing flow vectors results in corresponding vectors being equal. The following subsection then discusses how to reposition flow vectors.

1) Changing length and orientation of vectors

Given a horizontal movement of a point in the sky we need to show how to make the resulting flow vector in two WSI images equal. Without the pseudo-Cartesian transformation, the resulting flow vectors are clearly not equal. To PCT a flow vector, the positions of the base and the end of the vector are PCT-ed. The difference between the resulting two points gives the new PCT-ed flow vector.

The procedure for showing that two corresponding vectors are equal after PCT is identical to the method used to show that horizontal shapes in two WSI images are equal after PCT-ing. In Allmen and Kegelmeyer (1994) it was shown that the relative spacing between two points at the same altitude will be identical in two WSI images if the images are PCT-ed. If those two points are thought of as two separate points but rather the same point at two different times, then the same derivation can be applied.

2) Repositioning flow vectors

Once a flow vector is altered, it must be repositioned. If repositioning is not performed, then while corresponding vectors will be equal, the relative placement between corresponding vectors will not be correct. After a vector is PCT-ed, the base of the vector—that is, the pixel that the vector is flowing from—is PCT-ed. The new position is the new location of the flow vector. The resulting vector and its position is now identical to the vector that would have resulted from computing the flow field on a sequence of PCT-ed images.

d. Setting the parameters of the PCT

The choice of \("\tan(65^\circ)"\) and \("235"\) as standard parameters for the PCT is arbitrary. After explaining why the parameters should change continuously, depending upon the location of points being compared, we will show how to choose the parameters.

When an image is PCT-ed the outer portion of the image is expanded and the inner portion of the image is compressed. The standard parameters of the PCT were chosen so as to provide a balance between the expansion and the compression. Two images and their flow fields could be PCT-ed once with any suitable parameters and the resulting images and flow fields
could be correlated to find corresponding points. However, in general, when correlating two neighborhoods from the two images and flow fields, no single set of overall PCT parameters will be optimal. It is possible that both neighborhoods will be compressed, throwing away useful pixels, or that both neighborhoods will be expanded, with pixel copying resulting in an effectively smaller correlation neighborhood.

One way to achieve a balance is to choose the parameters so that the amount of compression and expansion of the two neighborhoods is minimized. When a neighborhood is PCT-ed, it can expand or compress in the radial dimension (along the zenith) and/or in the circular dimension (along the azimuth). Given two neighborhoods, the expansion and compression along these two dimensions will be formulated. Then the parameters of the PCT can be chosen such that the amount of expansion and compression is minimized.

Figure 5b shows two points, $p_1$ and $p_2$, from different WSI images, neighborhoods around which are to be PCT-ed then correlated. Clearly, since the distance from the center of the image of the two points differ, the outer neighborhood will have fewer pixels than the inner neighborhood when doing the correlation. Figure 5b also shows in gray the neighborhoods of the two points after PCT. Now the neighborhoods are the same size but the inner neighborhood has skipped over some pixels and the outer neighborhood has repeatedly copied pixels. The skipping of pixels resulted because the PCT can move pixels toward the center along the zenith direction. Since this compresses pixels together, some pixels must be skipped. Pixels are copied more than once because the pixels can be moved away from the center of the image by the PCT; therefore, some pixels must be copied multiple times to fill in gaps. This is inevitable, but now we will quantify these effects and show how to minimize them.

The camera calibration equation (discussed in section 2) relating the distance from the center of the image to a point at zenith $\theta$ before PCT is
\[ R_{\text{wsi}}(\theta) = \theta (173.72 - 8.5 \theta), \]
where $\theta$ is in radians. After PCT we have
\[ R_{\text{pct}}(\theta) = \frac{235}{\tan(65^\circ)} \tan(\theta), \]
where $\theta$ is in radians. Combining $235/\tan(\theta)$ into one term, $C$, to be determined below, we have
\[ R_{\text{pct}}(\theta) = C \tan(\theta). \]

To find the amount of compression or expansion along the radial direction we consider a small neighborhood around a point that is to be PCT-ed. The radial extent of the neighborhood corresponds to some small change in zenith angle, $\Delta \theta$. For there to be no compression or expansion, the radial extent of the neighborhood must be the same before and after PCT for the same $\Delta \theta$. That is, we want to minimize the following function $g_1(C)$, which computes the difference in radial extent before and after PCT (the subscript on $g$ indicates the point being considered):
\[ g_1(C) = [R_{\text{pct}}(\theta_i + \Delta \theta_i) - R_{\text{wsi}}(\theta_i)] - [R_{\text{wsi}}(\theta_i + \Delta \theta_i) - R_{\text{wsi}}(\theta_i)]. \]

Equivalently, the square of the function can be minimized. Simple substitution and manipulation gives
\[ g_1(C) = \{(173.72 \Delta \theta_i - 17.0 \theta_i \Delta \theta_i - 8.5 \Delta \theta_i^2) - C[\tan(\theta_i + \Delta \theta_i) - \tan(\theta_i)]\}^2. \tag{19} \]

The angular distance of a neighborhood before PCT is given by
\[ S_{\text{wsi}}(\theta) = \alpha R_{\text{wsi}} = \alpha \theta (173.72 - 8.5 \theta), \tag{20} \]
where $\alpha$ is the angle that subtends the neighborhood. Here $\theta$ and $\alpha$ are in radians. The angular distance of a neighborhood after PCT is given by
\[ S_{\text{pct}}(\theta) = \alpha R_{\text{pct}} = \alpha C \tan(\theta), \tag{21} \]
where again $\alpha$ is the angle that subtends the neighborhood. Since the PCT only moves points radially, $\alpha$ is the same before and after PCT.

To minimize the amount of circular compression or expansion due to the PCT, we want the difference between Eqs. (20) and (21) to be minimized. Equivalently, the square of the difference of Eqs. (20) and (21) can be minimized, giving
\[ h_1(C) = [S_{\text{pct}}(\theta_i) - S_{\text{wsi}}(\theta_i)]^2 = [\alpha_i \theta_i (173.72 - 8.5 \theta_i) - C \alpha_i \tan(\theta_i)]^2. \tag{22} \]

The subscript on $h$ indicates the point being considered.

Equations (19) and (22) give the amount of compression or expansion for one neighborhood when PCT-ed. However, we need to PCT two neighborhoods that will then be correlated. To minimize the compression and expansion for both neighborhoods, we sum Eqs. (19) and (22) for both neighborhoods and minimize. Substitution gives
\[ g_1(C) + g_2(C) + h_1(C) + h_2(C) = \{(173.72 \Delta \theta_i - 17.0 \theta_i \Delta \theta_i - 8.5 \Delta \theta_i^2) - C[\tan(\theta_i + \Delta \theta_i) - \tan(\theta_i)]\}^2 + \{(173.72 \Delta \theta_i - 17.0 \theta_i \Delta \theta_i - 8.5 \Delta \theta_i^2) - C[\tan(\theta_i + \Delta \theta_i) - \tan(\theta_i)]\}^2 + [\alpha_i \theta_i (173.72 - 8.5 \theta_i) - C \alpha_i \tan(\theta_i)]^2 + [\alpha_i \theta_i (173.72 - 8.5 \theta_i) - C \alpha_i \tan(\theta_i)]^2. \]

We want the value of $C$ that minimizes this. This is just a parabola, so the minimum is found by computing the
appropriate values for \( \alpha \) and \( \Delta \theta \) still need to be found. In the discussion below, subscripts on \( \Delta \theta \) and \( \alpha \) will be dropped, but reference to \( \Delta \theta \) and \( \alpha \) will refer to all \( \Delta \theta \)'s and \( \alpha \)'s. One approach to PCT-ing a neighborhood would be to PCT every point in the neighborhood, effectively taking a square patch in WSI coordinates and putting it in PCT coordinates. If done in this way, \( \Delta \theta \) and \( \alpha \) are easily determined from the size of the neighborhood in WSI coordinates. However, as described in section 3a, using this method may result in pixels in the PCT neighborhood that have no value mapped to them since the PCT does not map pixels one to one. To prevent this, the mapping is done from PCT coordinates to WSI coordinates as described in section 3a. In this case \( \Delta \theta \) and \( \alpha \) would have to be determined from the neighborhood in PCT coordinates. But this cannot be done without first PCT-ing the neighborhood. But this cannot be done until \( \Delta \theta \) and \( \alpha \) are determined. In other words, there is a circular dependency. This dependency is eliminated by assuming that the values of \( \Delta \theta \) and \( \alpha \) for square WSI neighborhood coordinates will result in the same minimizing value of \( C \) as would result using a square PCT neighborhood around the two points to be PCT-ed. So, \( \Delta \theta \) and \( \alpha \) are determined from square neighborhoods in WSI coordinates. Recall that we are trying only to find the optimal parameters of the PCT in order to minimize the number of pixels copied and/or skipped. If this assumption fails, we still have a PCT that results in corresponding points having equal shape and flow vectors.

e. The algorithm

At this point everything necessary to compute cloud-base heights has been described. The complete algorithm is as follows:

1. Compute the optical flow from WSI 1 and WSI 2 [section 1b(4)].
   - For every point \( p_1 \) in image 1
     1) Compute the epipolar line for \( p_1 \) in image 2 [section 2a, Eqs. (4) and (5)].
     2) For each point \( p_2 \) along the epipolar line
        a) perform PCT on the pixel values in neighborhoods around \( p_1 \) and \( p_2 \) (section 3a),

---

**Fig. 6.** One frame of a synthetic altocumulus cloud scene as viewed from two WSI cameras. WSI 2 was south of WSI 1, so corresponding pixels in the right image are shifted up relative to the left image.

**Fig. 7.** Relative geometry of the cameras and ceilometer at White Sands. There is a WSI camera at site 1 and site 2. The distance and bearing from site 2 to site 1 is 5.54 km and 125°, respectively. The distance and bearing from site 2 to the ceilometer is 2.16 km and 133°, respectively.
(b) perform PCT on the flow fields in neighborhoods around \( p_i \) and \( p_{i+2} \) (section 3c), and
(c) perform a 4D correlation of the PCT-ed pixel and flow neighborhoods.

3) Find the maximum correlation value along the epipolar line. Call this point \( p'_{i} \).

4) Compute the epipolar line for \( p'_{i} \) in image 1 [section 2a, Eqs. (4) and (5)].

5) For each point \( p_i \) along the epipolar line
   (a) perform PCT on the pixel values in neighborhoods around \( p'_{i} \) and \( p_{i} \) (section 3a),
   (b) perform PCT on the flow fields in neighborhoods around \( p'_{i} \) and \( p_{i} \) (section 3c), and
   (c) perform a 4D correlation of the PCT-ed pixel and flow neighborhoods.

6) Find the point of maximum correlation value along the epipolar line. Call this point \( p'_{i} \).

7) Compute the cloud height using \( p_i \) and \( p'_{i} \) as corresponding points [Eq. (6)].

8) Compute the distance between \( p_i \) and \( p'_{i} \). This is the confidence of point \( p_i \).

4. Real and simulated test data

Cloud-base heights have been computed for various synthetic cloud image sequences and for real WSI data collected under conditions designed to be similar to those that will actually be experienced in field studies. The generation of the synthetic data and the collection of the real data is described in the subsections to follow.

4a. Simulated data

The simulated sequences were created with a cloud scene simulation model, developed by Ciuciol (Ciucioi and Rasmussen 1992), which uses stochastic field generation techniques and knowledge of atmospheric structure and physics to model four-dimensional (three spatial and one temporal) cloud scenes, represented by liquid water content (LWC) values. To this we attached a cloud density model to derive radiance fields from the LWC volumes. Synthetic images are then generated by projecting idealized sunlight through a cloud scene using a WSI camera model that is identical to an actual WSI. Since the cloud scene has a temporal component, it can be projected at a sequence of times, creating an image sequence that captures cloud evolution and motion.

Figure 6 shows one example of images produced from a synthetic cloud field. Two images are shown, one from each WSI camera viewing the cloud field.

Although the model and projection are much simpler than the real cloud projection in the atmosphere, the dataset can be used as an initial optimized case for the cloud-base height calculation. Also, this allows calculation and, most importantly, verification of cloud base heights on realistic data. Moreover, the simulation can produce a wider range of cloud types than exist in our real imagery.

4b. Whole-Sky Imager data

The WSI camera that generated our existing data is the Whole-Sky Imager developed by the Marine Phys-

**Fig. 8.** Accuracy of two synthetic cases. White indicates that the computed height was too high. Black indicates too low. Dark gray indicates there was no cloud or that the computed height was not confident. Light gray indicates that the height was computed within 5% of the true height.
Fig. 9. Normalized error histograms for eight different synthetic cases.
Fig. 10. Normalized error histograms for different correlation neighborhood sizes.
ical Laboratory (MPL) at the Scripps Institution of Oceanography, as described in Shields et al. (1990b). Our real WSI data were taken in May 1992, in White Sands, New Mexico. In an attempt to simulate the eventual CART data, we separated two WSIs by 5.54 km, with a ceilometer located close to the midpoint between them. The intent of the ceilometer was to provide fiduciary points with which to check our algorithm. It made measurements once per minute, in time with both WSIs. As a result, when the ceilometer reports the presence of clouds, simple geometry and knowledge of the camera location suffices to compute which pixel on the WSI images corresponds to that ceilometer report. The cloud-base height computed at that point can then be compared to the ceilometer measurement. Figure 7 shows the geometry between the sites and the ceilometer.

5. Performance results

In this section results will be presented for both synthetic and real imagery. A number of issues are investigated, including the following.

- What size correlation window should be used?
- How far off the epipolar line should you look for corresponding points?
- How much does using the optical flow field help in locating corresponding points?

The geometry in the synthetic case is known exactly so it does not make sense to look off the epipolar line for matches. But all the other issues are relevant to synthetic and real images. After showing results for the synthetic cloud data, each of these issues will be addressed.

In all cases, only points where the computed height was confident are considered when presenting results. Points with confidence values below 2.1 are considered confident. If the confidence threshold is set to a lower value (so only more confident points are considered), the number of confident points decreases, but the distribution of errors remains approximately the same.

For all histograms, the standard deviation of the data shown in the histogram is presented. However, the usefulness of standard deviation as a metric depends on the application. If large errors are no worse than small errors, then the standard deviation is not a good indicator of performance since large errors have more impact on the standard deviation than do small errors. However, if large errors are in fact worse than small errors, then the standard deviation is an appropriate measure of performance.

Figure 8 shows the points where cloud height was computed correctly and incorrectly for two synthetic cases. White indicates that the computed height was too high. Black indicates too low. Dark gray indicates there...
was no cloud or that the computed height was not confident. Light gray indicates that the height was computed within 5% of the true height.

The error histograms of these two cases and six other synthetic ones are shown in Fig. 9. For each confidently computed height, it is subtracted from the true height. The resulting differences are shown in the normalized histogram. As expected, the peaks are centered at zero.

To test the accuracy of our computed heights with real images, a ceilometer was used to measure the cloud height for a point on a cloud. Cloud heights were computed for all the points in a $5 \times 5$ neighborhood around the point where the ceilometer was pointing. This is done for all the ceilometer hits in one day, and results for the entire day are shown together. Unfortunately, the range of the ceilometer was only 3800 m, so clouds above 3800 m were not detectable by the ceilometer. Since only points around ceilometer hits are processed for these results, all clouds in these experiments were below 3800 m. The lower the cloud, the greater its shape varies between two WSI cameras. Therefore, these results show the performance of the algorithm on the hardest of cases—that is, low clouds. It is reasonable to ex-

Fig. 12. Normalized error histograms using optical flow (left) and not using optical flow (right). Each row shows corresponding histograms. The 4 May data were used in every run shown here, but the parameters of the runs were different for each row.
pect that results will improve if clouds at all heights are considered.

The size of the neighborhood used to do the correlation when trying to find matching points can vary anywhere from one to the size of the image. Smaller neighborhood sizes require less computation but often do not perform as well as larger sizes. To determine the optimal value, results are computed for various correlation sizes. This is shown in Fig. 10. After a size of about $19 \times 19$ little is gained by using a larger window.

Since there is uncertainty in the relative positioning of the WSI cameras, it is possible that the matching point is not on the epipolar line. Therefore it may be beneficial to look some small distance away from the epipolar line for matching points. Figure 11 shows the results when additional points near the epipolar line are considered as possible matching points. Clearly, nothing is gained by examining points off the epipolar line. So, if the correct matching point is in fact off the line, the point on the epipolar line that best matches must be very close to the true matching point. Since wider epipolar lines require more computation, an epipolar line width of 1 is used.

Finally, the importance of using optical flow was tested. Figure 12 shows results with and without using flow to help find corresponding points. The benefits of using flow, while present, are not as significant as one would initially expect. This tells us that, while the human visual systems needs the additional help of flow to find corresponding points, the computer gets most of the matching information from the texture of the images and not from the variations in the flow field.

The results for two days using the optimal parameter settings are shown in Fig. 13. In this section we have shown the following:

- A $19 \times 19$ correlation window should be used.
- Looking for matching points off the epipolar lines does not increase performance.
- The usefulness of optical flow for finding corresponding points is minimal.

6. Concluding remarks

We have demonstrated how paired data from widely separated whole-sky imagers can be fused to extract cloud-base heights, an important cloud property, and one that could not be recovered from either imager alone. An important feature of our approach, one that will help it to generalize the incorporation of other data sources, is its ability to measure its own confidence in the determined base heights.

A motivating factor for this research was to exploit the ability of the WSI imager to provide temporally dense sequences of images. This would allow us to use cloud dynamics to help find corresponding points between images. However, the usefulness of using optical flow was not as great as was expected. But this only says there is sufficient texture in the images to find corresponding points, and that the additional information of optical flow is not strictly necessary. We also found that looking off the epipolar line does not improve performance and that a $19 \times 19$ correlation window is a reasonable choice.

We presented results on both synthetic data and particularly challenging real data. Recall that due to the limitations of the ceilometer, only the hardest case clouds were used in the results. With this data, cloud-base heights are computed to within 5% of the correct height approximately 50% of the time.

Future efforts would most usefully devoted to improving the confidence metric. The confidence metric described in this work is a measure of how likely it is that a cloud point is visible in both cameras. Other metrics can be sensitive to how large and/or unique the best correlation value is along the epipolar line. If the best match is weak or there are other points with correlation values of similar strength, then the confidence that the correct matching point has been found should be low. This could be formulated using a Bayesian model where corresponding points along the epipolar line are found using the maximum a posteriori (MAP) probability criterion. In this case, the a posterior prob-

![Fig. 13. Normalized error histograms using the best setting of all the parameters.](image)
ability can be obtained for each possible match along the epipolar line. A low MAP probability indicates that the posterior probability distribution has a broad, low peak (as would occur for sky points), or that the there are many strong matches along the epipolar line. With this approach, the confidence would be low in these cases, as desired.

Other future and concurrent work investigates the extraction of other properties of interest (particularly fractional cloud cover and aspect ratio) from paired WSI images, and fusing of WSI images with satellite imagery in order to determine cloud top structure as well.

Acknowledgments. This research was supported by the U.S. Department of Energy through the Atmospheric Radiation Measurement Program, Sandia National Laboratories, Livermore, Contract DE-AC04-76DO00789. The Khoros\(^1\) system, developed by Rasure (Rasure and Williams 1991), was used for code development and digital image visualization. Code for the generation of synthetic cloud scenes was provided by TASC (The Analytic Sciences Corporation), and the code that simulates radiance maps and projects 3D cloud volumes into simulated WSI imagery was written by Chen-Hui Sun of Sandia National Laboratories. We also gratefully acknowledge Janet Shields and Richard Johnson of the Scripps Marine Physical Laboratory and Bob Endlich of White Sands Missile Range for providing whole-sky images for our analysis.

REFERENCES


---

\(^1\) A publicly available integrated software development environment for information processing and visualization. Send e-mail to khoros@chama.ece.unm.edu for further information.

---

---


Lyons, R. D., 1971: Computation of height and velocity of clouds over Barbados from a whole-sky camera network. SMRP Research Report 95, Department of Geophysical Sciences, University of Chicago.


---