Edge detection methods applied to the analysis of spherical raindrop images

J. R. Saylor* and N. A. Sivasubramanian

Department of Mechanical Engineering, Clemson University, Clemson, South Carolina 29634, USA

*Corresponding author: jrsaylor@ces.clemson.edu

Received 8 December 2006; revised 25 April 2007; accepted 30 April 2007;
posted 14 May 2007 (Doc. ID 77911); published 23 July 2007

Optical imaging of raindrops provides important information on the statistical distribution of raindrop size and raindrop shape. These distributions are critical for extracting rainfall rates from both dual- and single-polarization radar signals. A large number of raindrop images are required to obtain these statistics, necessitating automatic processing of the imagery. The accuracy of the measured drop size depends critically on the characteristics of the digital image processing algorithm used to identify and size the drop. Additionally, the algorithm partially determines the effective depth of field of the camera/image processing system. Because a large number of drop images are required to obtain accurate statistics, a large depth of field is needed, which tends to increase errors in drop size measurement. This trade-off between accuracy and depth of field (*dof*) is also affected by the algorithm used to identify the drop outline. In this paper, eight edge detection algorithms are investigated and compared to determine which is best suited for accurately extracting the drop outline and measuring the diameter of an imaged raindrop while maintaining a relatively large depth of field. The algorithm which overall gave the largest *dof* along with the most accurate estimate of the size of the drop was the Hueckel algorithm [J. Assoc. Comput. Mach. **20,** 634 (1973)]. © 2007 Optical Society of America

OCIS codes: 010.0010, 010.3920, 100.0100, 100.2000, 100.2960, 100.2980, 100.5010.

1. Introduction

There are several methods for measuring rainfall, however only radars provide the capability of providing detailed maps of rainfall over large areas. Precipitation radars fall primarily into two classes: single-polarization radars and dual-polarization radars [1]. The majority of precipitation radars are of the single-polarization type since these are used in operational weather stations; dual polarization radars are currently used primarily in research. While precipitation radars provide large spatial coverage, the accuracy of the measurements obtained is questionable due, among other things, to imperfect knowledge of the size and shape of raindrops, as is demonstrated below [2,3].

A single-polarization radar measures a reflectivity factor Z over its measurement domain. The measured reflectivity can be related to the distribution of raindrop sizes in the measurement volume. Following the treatment presented in Doviak and Zrnić [1], the drop size distribution (DSD) is related to Z via the equation:

$$Z = \int_0^\infty D^6 N(D) dD, \tag{1}$$

where D is the diameter of the drop and $N(D)$ is the DSD. The rainfall rate R is related to the DSD by the following equation:

$$R = \frac{\pi}{6} \int_0^\infty D^3 N(D) w_t(D) dD, \tag{2}$$

where $w_t(D)$ is the terminal velocity of the raindrop, which is itself a function of the diameter. Since both the reflectivity factor Z and the rainfall rate R depend on the DSD, a relation between R and Z can be es-

0003-6935/07/220001-16$15.00/0

tablished once the DSD is known. The DSD is typically modeled as an exponential having the form:

$$N(D) = N_0 e^{-\Lambda D}, \tag{3}$$

where N_0 and Λ are parameters used to obtain the best fit of the actual DSD to this exponential function. Substituting $N(D)$ from Eq. (3) into Eq. (1) and integrating, the following expression for Z is obtained:

$$Z = N_0(6!)\Lambda^{-7}. \tag{4}$$

Because single-polarization radars provide only one measurement, Z, and according to Eq. (3), the DSD requires a minimum of two parameters (N_0, Λ), the task of extracting R from Z using Eq. (2) is inherently underconstrained. Hence, for the single-polarization measurement, one needs to have *a priori* knowledge of the DSD. Since the DSD is known to vary with storm type, geographical location, and season [1], researchers continue to measure and study the DSD, and there is a need for improved DSD measuring methods.

Dual-polarization radars provide an improvement on the single-polarization measurement in that a two-parameter measurement is performed. In this case, the problem of needing to have *a priori* knowledge of the DSD is obviated. However, one does need to know the average shape of each drop size in the radar measurement volume in order to accurately obtain R from the two radar measurements Z_h and Z_v, the horizontal and vertical radar reflectivity, respectively. The reader is referred to [1], which shows that the relationship between (Z_h, Z_v) and R requires the eccentricity e versus D relationship for the drops in the radar measurement volume. The eccentricity is defined as:

$$e = (1 - \alpha^2)^{1/2}, \tag{5}$$

where α is

$$\alpha = v/h, \tag{6}$$

and v and h are the vertical and horizontal extent of the drop, respectively. Eccentricity quantifies the raindrop shape, and measurement of raindrop shape is complicated by the tendency of raindrops to oscillate. Theoretical models have been developed to predict the shape of raindrops [4,5], and laboratory measurements have been carried out with drops of different sizes to understand their behavior [6–8]. In addition, field measurements have also been carried out to extend the laboratory measurements and theoretical models [9]. However, inaccuracies still exist in the e versus D relationship, and a need for measurements of raindrop shape remains.

Summarizing, single-polarization radars require field measurements of the DSD for the particular storm type, geographical location and season for which that radar is being operated. Without this, extraction of R from Z results in large errors. Dual-polarization radars require the e versus D distribution which necessitates field measurements of raindrop shapes. Several methods exist for measuring the DSD and the e versus D distribution as are described below. First, however it should be noted that even if one were to measure the DSD perfectly, other issues exist which would complicate rain measurement via radar. Among these include the fact that the radar volume is large and therefore integrates over heterogeneities within the radar volume. Hence variations in rain rate due to differences in the DSD within the measurement volume will not be seen and will result in errors in R. Also, measurements of DSDs on the ground may not agree with those in the radar measurement volume. Furthermore, the presence of wind can cause rain observed at a certain location in a radar volume to fall in a different spatial location on the ground. Finally, the radar measurement volume may be at sufficient elevation that some or all of the signal is due to ice, which has a very different radar cross section, potentially causing large errors in the value of R obtained from Z. Although not investigated here, these and other issues can have a significant impact on measurements of rain using radar.

Several methods have been developed and utilized for measuring individual droplet sizes for computation of the DSD. The Joss-Waldvogel impact disdrometer (JWD) [10,11] is perhaps the most commonly used method for measuring the DSD. This disdrometer measures the size of a drop based on the impact of the drop on a Styrofoam cone. The limitations of this disdrometer are that high noise environments can result in erroneous drop counts. The JWD also underestimates the amount of small drops in heavy rain because of simultaneous drop impacts [12]. Because the JWD is an impact type device, it does not have the capacity to determine drop shape, and therefore cannot provide measurements of e.

Several optical methods have been developed for DSD measurement. Hauser *et al.* [13] developed a disdrometer that created a rectangular sheet of light by collimating light emitted from an IR LED. Drops falling past this sheet of light occlude light striking a photodiode, permitting measurement of drop diameter. Löffler-Mang and Joss [14] used a sheet of laser light and an optical sensor to detect drop sizes via extinction of the light by the drops. A single optical sensor was used in the work of both Hauser *et al.* [13] and Löffler-Mang and Joss [14] (and other methods of this general type [15,16]), providing a time trace which must then be processing to extract drop diameter and fall speed via some type of calibration. Because a single detector is employed in these methods, the presence of more than one drop in the laser sheet results in coincidence issues. A similar approach was employed by Grossklaus *et al.* [17] where a cylinder of laser light instead of a sheet is used. This device was optimized for high wind speeds and shipboard use. Delahaye *et al.* [18] extended the method of using a sheet or laser light to two sheets, permitting accurate measurement of droplet velocity. Borrmann and Jaenicke [19] developed a holographic method for measuring droplet size and velocity in clouds. This holographic droplet and aerosol recording (HODAR) system was successfully used in obtaining measurements for droplets ranging in diameter from 3 μm to several hundred microns. The two-dimensional video disdrometer (2DVD) [20–22] is capable of measuring both the drop diameter and *e*. The 2DVD consists of two line scan cameras that face two light sources along axes that are orthogonal to each other. Drops falling between the light source and the line scan camera cast shadows on the line scan camera, which are recorded. The region which both line scan cameras record is the measurement volume of the system. The shadowed area recorded by successive scans of each camera, are stacked one upon the other to construct an image of the raindrop. The size and shape of the raindrop are obtained from this reconstructed image. One issue with this method is that in the presence of horizontal winds, the drop travels laterally through the measurement area, resulting in a distorted image of the raindrop. While corrections for this exist, errors in the drop shape measurement remain [12,21]. The shape of the enclosure of the 2DVD also causes errors in the counting of small drops in the presence of wind [23]. A somewhat similar device was developed by Knollenberg [24]. This device consists of a HeNe laser focused on a linear diode array [24–26]. As droplets pass between the laser and the array, an occluded line is observed by the array, in a fashion similar to that for the 2DVD. As the drop passes, sequential scans of the array are used to construct a pseudodrop image [27].

Certain aspects of the above optical methods can be improved by utilizing direct optical imaging as opposed to a single sensor or a line scan camera. When a raindrop is imaged by a video or CCD camera the entire drop image is recorded at (essentially) one instant in time. This provides the drop shape information and avoids problems inherent in line scan cameras where the drop is imaged as it moves through the measurement region. Such a direct optical imaging approach has been used in, for example, the pluviospectrometer, which was developed by Frank *et al.* [28]. This method is very similar to that which is used here (see below) except that Frank *et al.* [28] focus the illumination source using condenser lenses placed around the light source and a pair of Fresnel lenses placed in front of the light source.

Direct optical imaging has been employed by NASA in the form of the Rain Imaging System (RIS) which obtains images of raindrops, illuminated from behind [29]. Figure 1 shows the general optical setup used in this approach, where the camera records images of drops that are backlit by the lamp. This results in a drop image that is a silhouette of the drop. A sample gray scale image obtained using this backlit configuration is presented in Fig. 2(a).

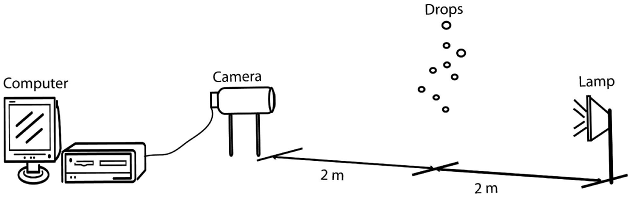


Fig. 1. Raindrop imaging setup.

Imaging drops that are illuminated from behind is a particularly useful technique in field measurements of raindrops [30,31]. Because the light source is located behind the drop, an image of the light source can be seen through the center of the drop itself, appearing as a bright spot on the dark background of the drop. Here the drop acts as a lens, focusing the image of the light source at a location just in front of the drop. Hence, when the camera is focused on the drop itself, the image of the light source in the drop is also in focus. Because of this, bright spots can be used to determine if the drop is in focus or not. This is particularly useful when attempting to automate the identification of in-focus and out-of-focus drop images. An example of an out-of-focus drop image is presented in Fig. 2(b) which shows the lack of a hole in the image. The depth of field of the camera can be determined by identifying the region along the optical axis of the camera, *z* where the drop exhibits a bright spot. This approach is used in the present work, and we define the depth of field, *dof* as:

$$dof = z_e - z_s, \tag{7}$$

where z_s and z_e are the locations along the optical axis where the bright spot "starts" and "ends", respectively, as the drop location moves from the camera to the light source. Because the shape of falling raindrops oscillate, the images obtained will be of ellipses having a range of α [Eq. (6)]. While this oscillation in α does change the shape of the bright spot, Saylor *et al.* [30] showed that variations in α do not have a significant effect on the *dof* or on the ratio of the size of the bright spot to the size of the drop. Hence, the work presented herein where spheres are imaged, is relevant to actual rain.

To improve the accuracy with which the raindrop size distribution is obtained, the rate at which rain-

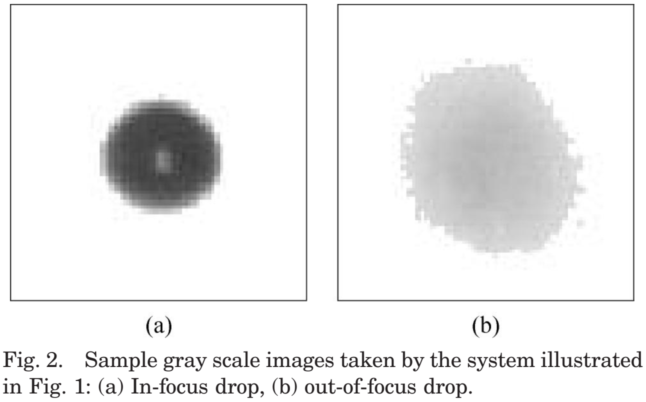


Fig. 2. Sample gray scale images taken by the system illustrated in Fig. 1: (a) In-focus drop, (b) out-of-focus drop.

drops are measured should be high. For optical systems like those presented in Fig. 1, this translates into having a large measurement volume. This is because, for a given rainrate, the only way to acquire more raindrop measurements in a unit period of time is to image a greater spatial volume. Of course it is possible to increase the imaging rate, but this is not desirable since at a sufficiently high rate of image acquisition, the smaller drops, which have small terminal velocities, will be imaged more than once, thereby distorting the measured drop size statistics. However, by increasing the measurement volume by increasing the depth of field, more drops are imaged in each acquired frame. As noted above, computing the depth of field is done by identifying the z locations, where a hole appears and disappears. Depending on the image processing method used, a drop image obtained at a particular location may or may not have a hole. Therefore, the *dof* depends not just on the optical characteristics of the camera and light source, but also on the algorithm used to process the resulting images. Hence, by identifying an algorithm that detects holes over a wider range of z, a larger *dof* and a larger measurement volume can be obtained.

In addition to a large measurement volume, a precise measurement of the DSD requires accurate measurements of drop diameter. Within the *dof*, drops closer to the camera will result in drop images that are larger than those that are located farther from the camera. This is an inherent characteristic of geometric optics and results in an increasing sizing error with increasing *dof*. However, the image processing algorithm that is used to identify the drop outline can increase or decrease this error. Hence, the trade-off between *dof* and measurement accuracy is partially dependent upon the image processing algorithm used.

As the above discussion shows, the quality of the DSD measurements obtained using the setup shown in Fig. 1 depends critically on the image processing algorithm used. The goal of this image processing algorithm is segmentation followed by sizing; the drop must be identified and the existence of a hole must be determined. If there is a hole, then the drop diameter is measured. Image segmentation processes are broadly classified into the following three groups [32,33]: a) thresholding or clustering techniques; b) region growing and c) edge detection techniques. Thresholding methods have been investigated in our earlier work and will not be discussed here [34,35]. Region growing techniques have a high degree of complexity associated with them and were not considered to be practical for this application. Accordingly, the present work focuses on edge detection methods.

In addition to the choice of an image processing method, other factors affect the ability to measure accurately the drop size using a CCD camera. Among these are problems related to smearing of drops traveling at terminal velocity. This is particularly true with large raindrops which have large terminal velocities. With most digital cameras, a range of exposure times are available and can be reduced to very small durations. In this situation, the problem of avoiding image smearing ultimately reduces to one of providing sufficient lighting intensity. In the work of Saxena and Saylor [34], the setup used in the present work was used to obtain DSDs using imagery acquired during an hour of rain, and using a 300 W halogen lamp. Smearing was not observed for the raindrop images acquired in that work. Should blurring exist for different imaging configurations, higher power lamps and/or more sensitive cameras are commercially available.

2. Edge Detection Algorithms

Edge detection algorithms identify the boundaries of an object based on abrupt gray level changes. For analyzing the raindrop images considered here, the inner (hole boundary) and outer boundary of the drop must be identified. The identification of the inner hole is used to determine if the drop lies within the depth of field, and the identification of the outer boundary provides the drop diameter.

Two steps are involved in the present application. First, the edge operator is applied to the original image producing an image where the edges are enhanced. Second, a thresholding operation is applied where edge pixels are identified. Figures 3(a) and 3(b) show the images obtained after edge enhancement and thresholding, respectively of the sample gray scale raindrop image presented in Fig. 2(a). Note that while the hole in the center of the drop in Fig. 2(a) is white, the hole in Fig. 3(a) is black.

A brief summary of existing edge detection algorithms is now presented. Several studies have compared the performance of edge detectors. Peli and Malah [36] compared the performance of edge detection methods not requiring *a priori* knowledge of the image. The algorithms considered were Roberts's algorithm [37], Hale's operators [38] and Rosenfeld's algorithms [39–41] all of which are nonlinear algorithms. These algorithms were evaluated based on two quantitative and three qualitative measures. The

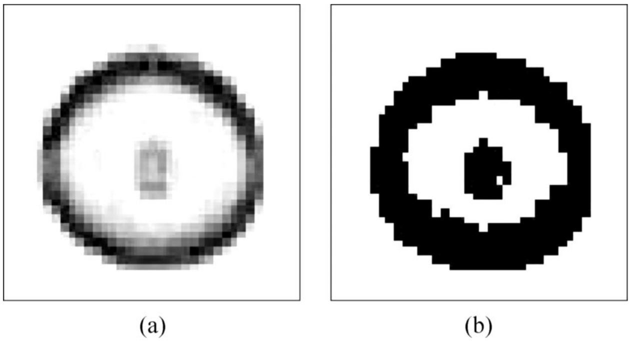


Fig. 3. (a) Edge-enhanced version of the sample gray scale image presented in Fig. 2(a). The Sobel operator was used in this case. (b) Binary image obtained after thresholding the edge enhanced image in (a). The slightly oval shape of the white portion in the center of the image is due to a difference in the x and y-direction pixel resolutions, discussed in Section 4.

authors found that the Roberts's algorithm performed best on low noise images and the Rosenfeld 1–4 algorithm (1–4 represents the size of the neighborhood used) was best suited for noisy edges. Both of these algorithms were found to detect curved edges. Shaw [42] compared the accuracy and reliability of local and regional edge operators. The local methods considered were the Sobel operator [43], Mero and Vassy's approximate Hueckel operator [44] and Hummel's operator [45], while the regional operators were Hueckel's algorithm [46,47] and a discrete Hueckel-like operator [42]. The regional operators performed better than the local. Although the Hueckel and the discrete Hueckel operator gave similar results, the latter was found to be more stable. Shin *et al.* [48] compared the performance of eight edge detectors, evaluating them by measuring the accuracy with which they recovered the structure and motion of a sequence of images. The edge detectors compared were Anisotropic [49], Bergholm [50], Canny [51], Rosenthaler [52], Rothwell [53], Sarkar [54], Sobel [43] and SUSAN [55]. The Canny edge detector gave the best results for this particular application.

Heath *et al.* [56] quantitatively compared the Canny [51], Nalwa–Binford [57], Sarkar–Boyer [54], and Sobel edge detectors based on the subjective evaluation of human subjects. These authors found that there was no one best detector for all images considered. However, they did conclude that if the images to be analyzed were similar, then the Canny edge detector gave good results. On the other hand, the Nalwa-Binford detector was found to be suited for the analysis of a broader range of image types. Fram and Deutsch [58] compared the performance of human subjects with the performance of three edge detection schemes in the presence of noise. The algorithms considered were Hueckel [46,47], Macleod [59,60] and Rosenfeld [40,41,61]. The Macleod algorithm with a large mask gave the best results. Abdou and Pratt [62] evaluated the performance of several algorithms based on the amplitude of the edge gradient magnitude, the probabilities of correct and false edge detection, and a figure of merit computation. They considered the Roberts operator [37], the Sobel operator [43], the Prewitt operator [63], the Compass gradient operator, the Kirsch operator [64], and the 3-level and the 5-level operators. They found that the 3 × 3 pixel 3-level template matching edge detector, the 3 × 3 pixel Sobel and the Prewitt differential edge detectors performed well.

The comparison studies described above do not provide a consensus on a single 'best' edge detector. However, the operators that tended to give the best results were:

1. The Roberts algorithm
2. The Rosenfeld (1–4) algorithm
3. The Hueckel operator
4. The Canny edge detector
5. The Macleod algorithm with large mask
6. 3 × 3 pixel 3-level template matching operator
7. 3 × 3 pixel Sobel operator
8. 3 × 3 pixel Prewitt operator

These were chosen as candidate edge detection algorithms in the present work. As noted above, the discrete Hueckel operator was found to be more stable than the Hueckel operator. However, the latter was chosen for study in this work because of its wider acceptance as a good edge detector [65]. Each of these candidate algorithms is now described. Space considerations preclude a detailed discussion of each method, and the reader is referred to the original citations for more information.

A. Roberts Operator

The Roberts operator [37,66] uses a 2 × 2 neighborhood to estimate the derivative using the following convolution masks:

$$H_1 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \qquad H_2 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}. \tag{8}$$

The convolutions obtained from these two masks are used to compute the gradient magnitude using Eq. (9):

$$g(i, j) = [g_1{}^2(i, j) + g_2{}^2(i, j)]^{1/2}, \tag{9}$$

where g_1 and g_2 and the results of the two convolutions.

B. Rosenfeld (1–4) Algorithm

This algorithm is based on computing differences of the average of two adjacent, nonoverlapping neighborhoods [36,39–41]. In this algorithm, four differences are computed, one along the horizontal, the vertical, at 45° and at 135°. Here, the implementation of this algorithm as described by Peli and Malah [36] is presented. The neighborhood over which the algorithm is applied is $(2r + 1) \times (2r + 1)$, where r is a parameter.

Step 1:

For $r = r_1 + 1, \ldots, r_2$
    For each point (i, j) in the image

1) The average of the neighborhood centered at the point (i, j) is computed:

$$f^{(r)}(i, j) = \frac{\left[\sum_{m=i-r}^{i+r} \sum_{n=j-r}^{j+r} f(m, n)\right]}{(2r + 1)^2}. \tag{10}$$

2) Four average neighborhood differences are computed: $L^{(r,H)}, L^{(r,V)}, L^{(r,45)}, L^{(r,135)}$, which are the horizontal, vertical, 45° diagonal and 135° diagonal differences of $f^{(r)}$, respectively.

Step 2:

For each point (i, j) and
  For each direction d $\in [H, V, 45^\circ, 135^\circ]$

$$L^{(d)}(i, j) = L^{(r_1,d)}(i, j) \times L^{(r_1+1,d)}(i, j) \times \ldots \times L^{(r_2,d)}. \tag{11}$$

Step 3:

For each point (i, j)

$$L(i, j) = \max[L^{(d)}(i, j)]. \tag{12}$$

The image $L(i, j)$ is then thresholded to obtain the binary image. This algorithm was implemented for $r = 1$ to 4 in this work, and hence it is referred to as the Rosenfeld (1–4) algorithm.

C. Hueckel Algorithm

The Hueckel [44,46,47] edge operator is a two-dimensional edge fitting operator. The ideal step edge $S(x, y, b, h, \rho, \theta)$ in two dimensions, is given by [67]:

$$S(x, y, b, h, \rho, \theta) = \begin{cases} b & (x \cos\theta + y \sin\theta) < \rho \\ b + h & (x \cos\theta + y \sin\theta) \geq \rho' \end{cases} \tag{13}$$

where h is the step height, b is the base intensity, and (ρ, θ) define the position and orientation of the edge.

This operator uses a small circular subset of the image called the input-disk, as an input to the algorithm. Based on experiments, an input-disk consisting of 69 pixels was found to be most practical by Hueckel and the same was used here. The values for the input-disk pixels are assigned from the image matrix. The operator works by fitting the gray levels in this disk to the ideal edge model given by Eq. (13). The values b, h, ρ and θ are chosen to minimize the error between the ideal edge and the image. This minimization is performed by expanding the ideal edge and the image intensity as a truncated Fourier series. The expansion is carried out using a set of nine basis functions. The constants defining these functions are computed and then analyzed to determine if there is an edge fit. If there is, then the location and orientation of the edge in relation to the circular disk is obtained along with the intensity of the edge. The location and orientation is then matched to the location on the image, and the intensity obtained is assigned to that location. This process is repeated by moving the input-disk over the entire image creating a new image of edge intensities. The resulting image is then thresholded to obtain the binary image.

D. Canny Algorithm

The Canny edge detector [51,66,68] consists of the following six steps.

Step 1:

The image is first smoothed to remove noise using a two dimensional Gaussian filter:

$$G(i, j) = e^{\left(-\frac{i^2+j^2}{2\sigma^2}\right)}, \tag{14}$$

where σ is the standard deviation of the Gaussian which was set to $\sigma = w/5$, where w is the mask width, which was set to $w = 7$. For these settings, the mask comprises 98.76% of the total area under the Gaussian.

Step 2:

The x- and y-direction gradients are computed at each (i, j) location of the smoothed image using a fourth-order central difference approximation.

Step 3:

The magnitude of the gradient at each point is calculated using Eq. (9), where the values of g_1 and g_2 correspond to the x- and y-direction gradients, respectively.

Step 4:

The orientation of the edge normal is calculated at each point using the formula

$$\theta(i, j) = \tan^{-1}\left(\frac{g_j(i, j)}{g_i(i, j)}\right), \tag{15}$$

where g_i and g_j are the gradient magnitudes in the i and j directions.

Step 5:

Four directions, d_1 through d_4 are defined (0°, 45°, 90° and 135°, respectively) with respect to the horizontal axis. For each (i, j) point in the image, the orientation of the edge normal [Eq. (15)] is classified as belonging to one of these four directions, depending on which it is closest to. Hence, each $\theta(i, j)$ is reassigned to $\hat{\theta}(i, j)$, where $\hat{\theta}(i, j) = 0^\circ, 45^\circ, 90^\circ$ or 135°. The gradient magnitude for each point is compared with the gradient magnitude of two of its neighbors along the direction $\hat{\theta}$ of that point. If the gradient of the point is smaller than its neighbors, then the value of the gradient at that point is set to zero.

Step 6:

The resulting image is thresholded using hysteresis thresholding, a process requiring two thresholds. The high threshold (t_h) is selected by trial and error and the low threshold (t_l) is set to $t_l = t_h/3$. All pixels having an edge magnitude greater than t_h are classified as edge pixels. For all pixels having edge magnitudes between t_l and t_h, the pixel under consideration is classified as an edge pixel, if any four of its neighbors are edge pixels. Once a pixel is classified as an edge pixel, all those pixels connected to it are examined for the presence of edge pixels. This step is repeated until all pixels have been visited.

E. Macleod Algorithm

This algorithm is a correlation-based edge detector [58–60] using a mask defined as:

$$H(i, j) = \exp\left(-\frac{i^2+j^2}{d_r^2}\right)\left[\exp\left(-\left(\frac{d_{ij}+d_{pk}}{d_{pk}}\right)^2\right) - \exp\left(-\left(\frac{d_{ij}-d_{pk}}{d_{pk}}\right)^2\right)\right], \tag{16}$$

where $d_{ij} = i \sin\theta - j\cos\theta$, d_{pk} and d_r^2 are constants which determine the rate of decay of the exponentials perpendicular to and along the edge, respectively, and θ is the relative direction of the edge. A 13 × 13 mask is constructed from the above expression. The indices (i, j) are defined relative to the center of the mask. The values of θ, d_{pk} and d_r^2 are specified by the user. The mask $H(i, j)$ is then correlated with the image to create the edge-enhanced image.

F. 3 × 3 Pixel 3-Level Template Matching Operator

The 3 × 3 3-level template matching operator is also known as the Robinson 3-level operator, and uses eight convolution masks defined as:

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}\begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -1 \end{bmatrix}\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}\begin{bmatrix} 0 & -1 & -1 \\ 1 & 0 & -1 \\ 1 & 1 & 0 \end{bmatrix}$$

$$\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}\begin{bmatrix} -1 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}\begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix}.$$

Each mask is convolved with the image to provide eight gradients. The maximum of these is taken as the gradient magnitude for that pixel, and the resulting image thresholded to identify the edge locations. The 3 × 3 pixel 3-level template matching operator will be referred to as the Template 3 operator hereinafter.

G. Sobel Operator

The Sobel operator is a differential gradient operator used to detect horizontal and vertical edges. The masks for this operator are:

$$H_1 = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}, \qquad H_2 = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}. \tag{17}$$

A binary image is obtained by thresholding the image corresponding to the edge gradient magnitude, obtained by summing the results of the two convolutions obtained from Eq. (17) at each point.

H. Prewitt Operator

Similar to the Sobel operator, the Prewitt operator approximates the first derivative. The masks for the Prewitt operator are:

$$H_1 = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}, \qquad H_2 = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}, \tag{18}$$

which compute the gradient in the x and y directions. The gradient magnitude is then computed, and the resulting image thresholded.

3. Procedure

To ascertain the accuracy of the selected image processing algorithms, it was necessary to obtain images of drops having known diameters and then to compare the diameter obtained from each algorithm to the known diameter. Because of the problems associated with consistently producing water drops of the same size, spheres made of magnesium fluoride were used in place of water drops. Magnesium fluoride (MgF_2) was chosen because it has a refractive index ($n = 1.38$) very close to that of water ($n = 1.33$) [69]. Figures 4(a) and 4(b) show in-focus and out-of-focus gray scale images of an MgF_2 sphere having a diameter of 8 mm.

A. Experimental Setup

The experimental setup used to obtain the required images is shown in Fig. 5. This setup consists of a CCD camera and a halogen lamp separated by a distance of 4 m along the optical, or z-axis. This camera is connected to a PC which records and stores the images using a LabVIEW code. The setup also has a stand, the position of which can be varied along the optical axis. It can be located ±15 cm on either side of the focal point, which is 200 cm from the camera. The design is such that the smallest possible distance separating any two positions at which the stand can be located on the optical axis is 1 mm. This stand consists of a horizontal extension mounted with a plate containing a hole through which the MgF_2 spheres are dropped. It also consists of a horizontal slide used to adjust the position of the dropping plate

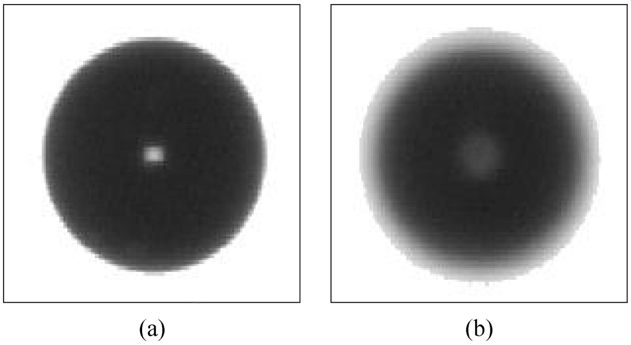


Fig. 4. Sample gray scale images of an MgF_2 sphere. (a) In-focus and (b) out-of-focus.

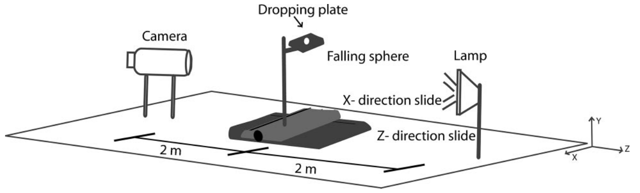


Fig. 5. Laboratory setup used to collect images of MgF_2 spheres of different diameters and at different z-locations.

along the x-axis. For the research reported here, the height of the stand was adjusted so that it was just above the image frame. The spheres were dropped through the stand, and their images were recorded as they fell. Because the spheres were manually dropped (viz., without any synchronization to the video clock), the y-locations of the recorded images were random. The size of the image frame is 640 × 240 pixels, with 640 pixels in the x-direction and 240 in the y-direction. The magnification of the camera was adjusted to obtain a pixel resolution of 0.05 mm/pixel in the x-direction and 0.1 mm/pixel in the y-direction. The f-number of the camera was set to 4. This setup is similar to that used by Saylor *et al.* [31] except that here the spheres were dropped instead of being mounted on a vertical post. Mounting of the spheres introduces the image of the post into the image frame which affects the outcome of the algorithms considered here. It is noted that because the spheres are dropped from a relatively low height, they are not traveling at terminal velocity when imaged. Hence, this present work does not demonstrate the ability of this system to image drops at terminal velocity. However, as noted earlier in this paper, Saxena and Saylor [34] employed the setup used in the present work to obtain raindrop images without smearing.

The setup shown in Fig. 5 allows the *dof* to be measured by determining the z-locations where the drop image just begins to exhibit a hole and stops exhibiting a hole. Images were recorded at intervals in z of 1 mm near z_s and z_e. Approximately 30 images were recorded at each location. A large number of images were collected because not all of the images contained a sphere located completely within the frame due to the fact that the y-location was not controlled. For the purpose of testing the image processing algorithms, only those images having the sphere lying completely within the frame were used. These images were then processed using each image processing algorithm and analyzed to determine the exact z_s and z_e values for each algorithm. Images were obtained for six sphere diameters: $D = 3, 4, 5, 6, 7,$ and 8 mm.

The selected edge detection algorithms were implemented in MATLAB. The images obtained after the application of the edge detection algorithms were thresholded to create binary images. A different threshold was used for each algorithm and was selected by trial-and-error to give the best results. Once selected, the threshold was used consistently for all images for that given algorithm.

After the images were thresholded, they were manually examined to determine the presence or absence of a hole. Identification of z_s and z_e was based on the following conditions. Starting with the image recorded at the smallest z and moving to images obtained at locations progressively farther from the camera, z_s was defined to be the first location having a hole, where the two successive positions also had holes. Moving further from the camera, z_e was defined as the first location to not have a hole where three successive positions also did not have holes in the center of the drop image. These criteria used to identify the *dof* will be referred to as the 3-hole criteria. The 3-hole criteria was generalized to an n-hole criteria to see if n had a significant role on algorithmic performance for the range $n = 3, 4, 5, 6, 7$. For this range of n, *dof* changed negligibly with n for all algorithms except for the Roberts algorithm where *dof* increased by ~40% when n was increased from 3 to 4. Upon further increase in n, *dof* changed little for the Roberts algorithm. Hence, the 3-hole criteria is used in computing *dof* for all of the results presented herein.

Because the method for measuring depth of field relies on the existence of a bright spot in the drop image, there is a minimum drop size below which such a spot will not be observable. It should be noted that this minimum drop size is not an absolute number, but rather depends on the magnification of the lens being used. Inherently, the minimum detectable size of the bright spot is a single pixel. Hence, the physical size of the minimum detectable bright spot is determined by the magnification ratio of the lens used, which for the system used here is 0.05 mm by 0.1 mm. However, by increasing the magnification of the lens, smaller hole sizes could be resolved. Similarly, it is noted that in the work presented here, the minimum sphere diameter used was 3 mm, while raindrops can have diameters considerably less than 3 mm. However, by using a higher magnification ratio lens, much smaller droplet diameters can be imaged, and hence the only lower limit is really the diffraction limit for the wavelength of light used, which is much smaller than the minimum size typically tabulated when measuring DSDs in precipitation science.

For each image within the *dof*, the measured drop diameter (D_m) was obtained. This was done by counting the number of pixels that fell within the outer drop boundary n_p, and then using the equation:

$$D_m = 2 \times \sqrt{\frac{A_d}{\pi}}, \tag{19}$$

where A_d is the area of the drop in mm^2 and is given by:

$$A_d = n_p \times 0.05 \times 0.1, \tag{20}$$

where 0.05 and 0.1 are the pixel resolutions (mm/pixel) in the x- and y-directions, respectively.

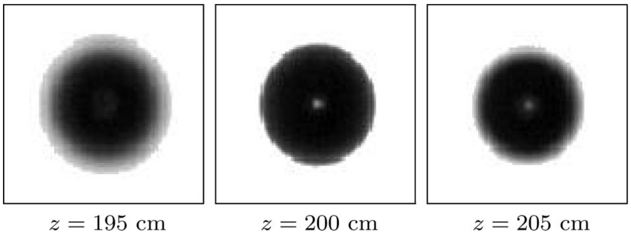


Fig. 6. Sample gray scale images of a 5 mm sphere at three different z-locations.

After the measured diameter was obtained for each z-location, the average diameter $\langle D\rangle$ was computed by integrating over all measured diameters in the *dof*:

$$\langle D \rangle = \frac{1}{dof}\int_{z_s}^{z_e} P(z)dz, \tag{21}$$

where $P(z)$ is a polynomial fit to the D_m versus z data.

The individual edge detection algorithms were compared using *dof* and the error in $\langle D\rangle$ as figures of merit.

4. Results

As noted in Section 2, eight different edge detection algorithms were considered here. Figure 6 shows the gray scale images of a 5 mm sphere obtained at three different z-locations, and Fig. 7 shows the binary images obtained after application of each of the eight edge detection algorithms to these images. Figure 7 shows that the images obtained using the Macleod algorithm are directionally sensitive i.e., the image is skewed in the direction of θ, the relative direction of the edge given as an input. The specification of a single direction as an input to the Macleod algorithm causes one direction to be weighted higher than the others, resulting in directionally sensitive outputs. This is not useful in this work where the circularly symmetric spheres have edges in all directions, and hence the Macleod algorithm is not considered further. Figure 7 also shows that some algorithms result in an oval shape instead of the expected circular shape. The ultimate cause of this is a difference in pixel resolution in the x- and y- directions. This problem did not prevent a comparison of the algorithms evaluated here as will be shown in Section 5, where an explanation of this effect is also presented. Finally, Fig. 7 shows that all the algorithms detect

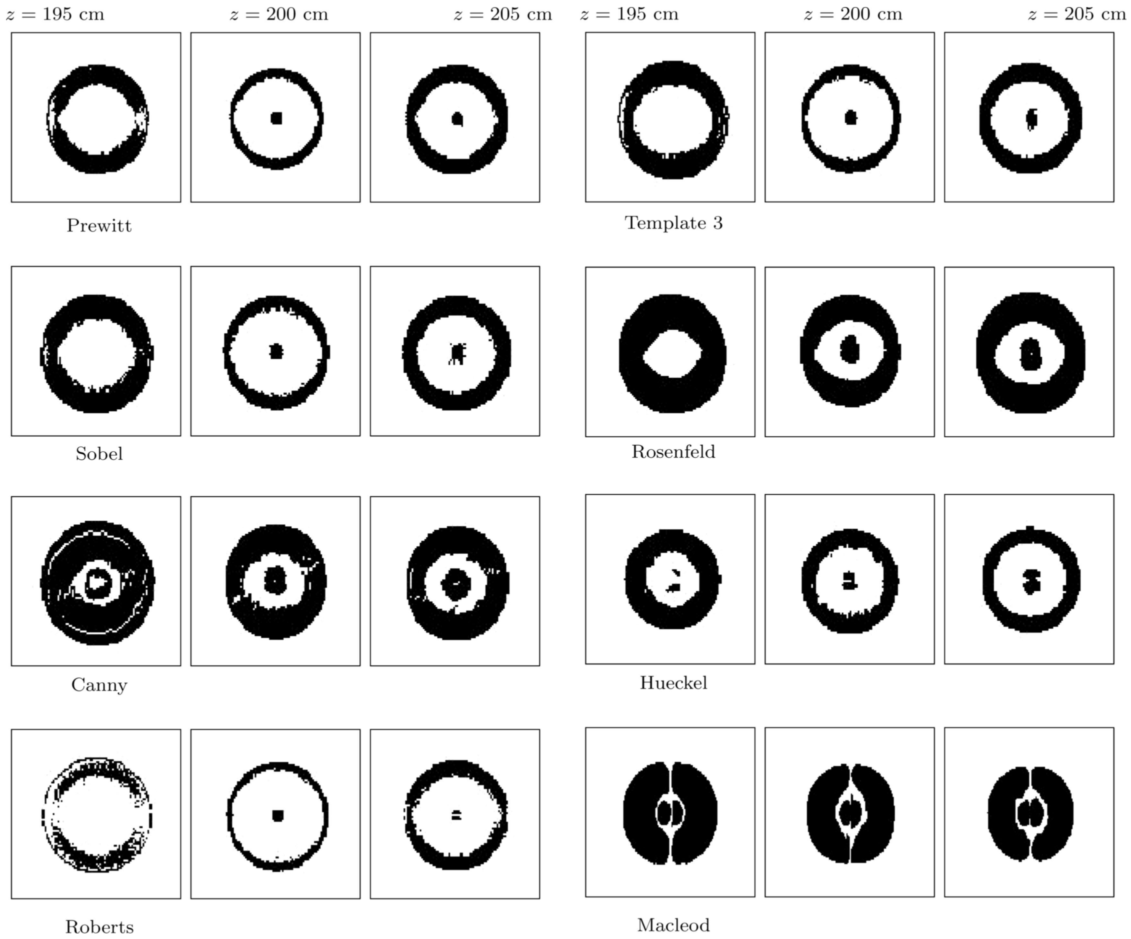


Fig. 7. Sample binary images of a 5 mm sphere after the application of the edge detection algorithms.

holes in the images at $z = 200$ cm, the focal plane, as expected.

For each algorithm, the diameter was measured for all the images and a plot of D_m versus z was obtained. Figure 8 shows the variation of D_m with z obtained by applying the Canny algorithm to the images. A single plot is presented for each of the six diameters tested. The pair of vertical dashed lines in each of the plots identifies the hole start and hole end positions, z_s and z_e, respectively. The distance between these two lines is the *dof*. The vertical solid line is the location of the focal point of the camera at $z = 200$ cm. In this figure, for all D, the value of D_m is high at the beginning of the *dof*, then drops down in the region around the

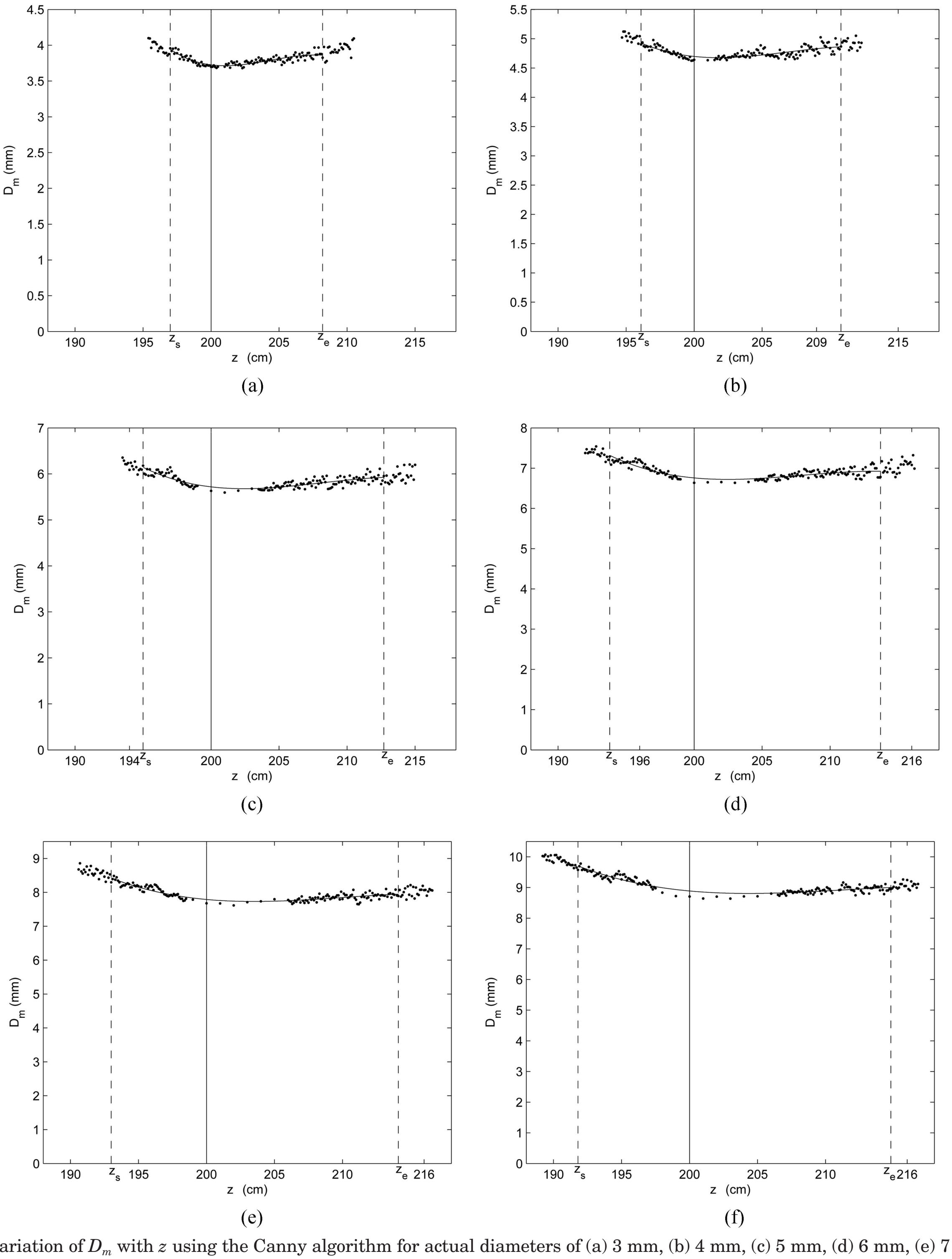
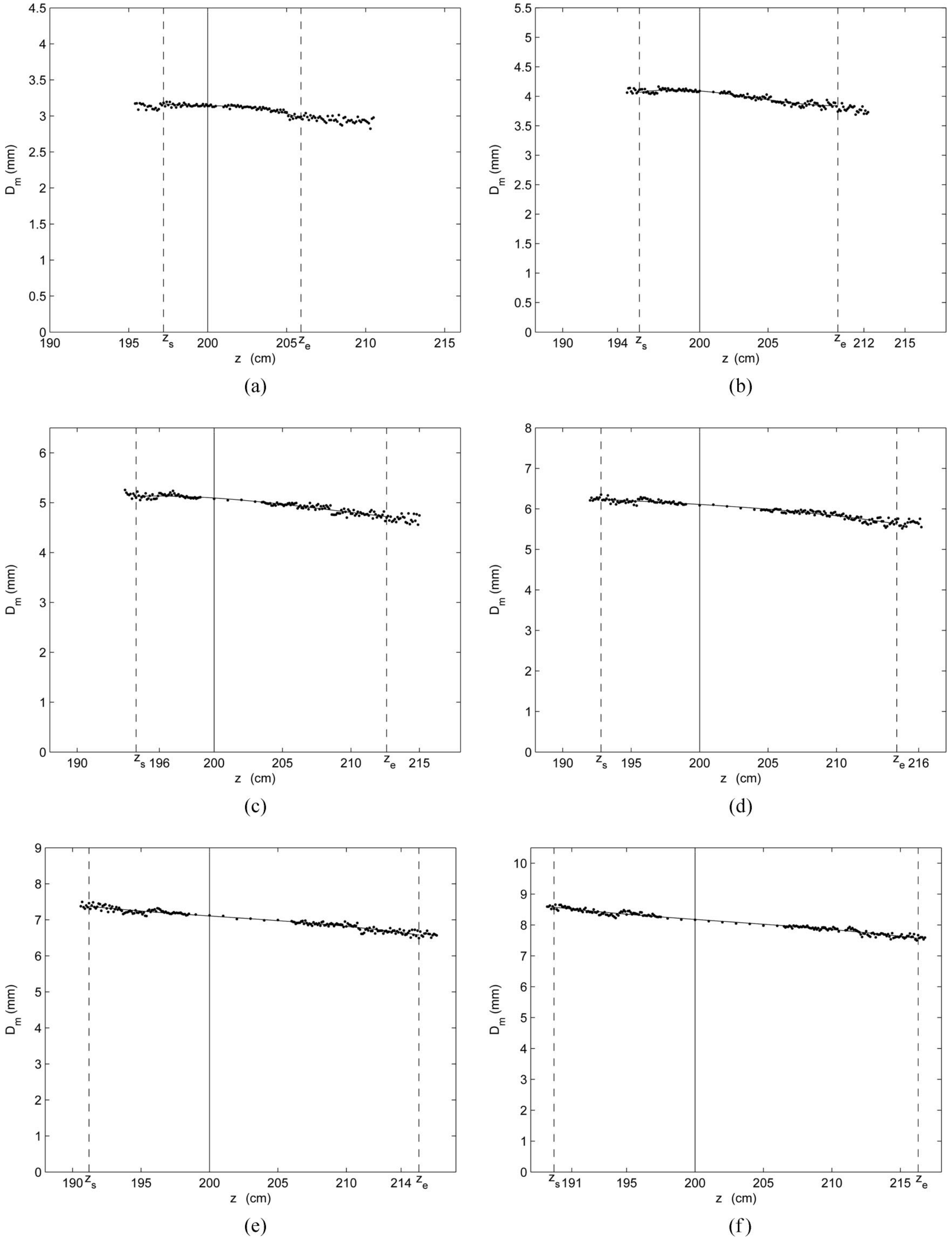


Fig. 8. Variation of D_m with z using the Canny algorithm for actual diameters of (a) 3 mm, (b) 4 mm, (c) 5 mm, (d) 6 mm, (e) 7 mm, and (f) 8 mm.

focal point and increases again at the end of the *dof*. The D_m versus z behavior of the other edge detection algorithms were all similar to Fig. 8, except for the Hueckel algorithm which is shown in Fig. 9. For the Hueckel algorithm the D_m versus z plot is close to a straight line with a shallow slope. Also, the deviation of D_m from D is small.

Figure 10 is a plot of *dof* versus D for each of the edge detection algorithms. The figure shows that the depth of field increases monotonically with D for all of the algorithms except the Roberts algorithm which shows some oscillatory behavior. The data presented in this plot are tabulated in Table 1.



(a)

(b)

(c)

(d)

(e)

(f)

Fig. 9. Variation of D_m with z using the Hueckel algorithm for sphere diameters of (a) 3 mm, (b) 4 mm, (c) 5 mm, (d) 6 mm, (e) 7 mm, and (f) 8 mm.

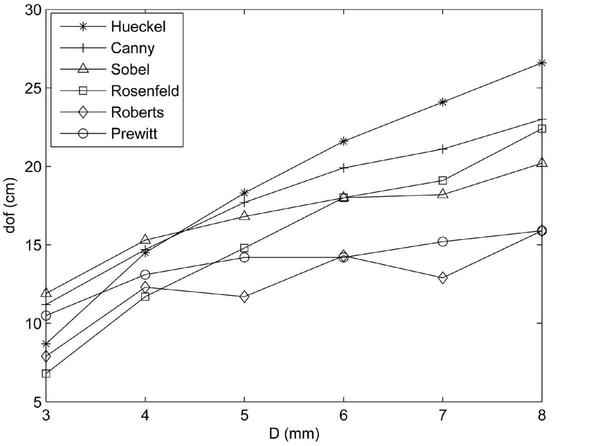


Fig. 10. Plot of *dof* versus D for each edge detection algorithm. The line through each data is a simple linear interpolation between each point.

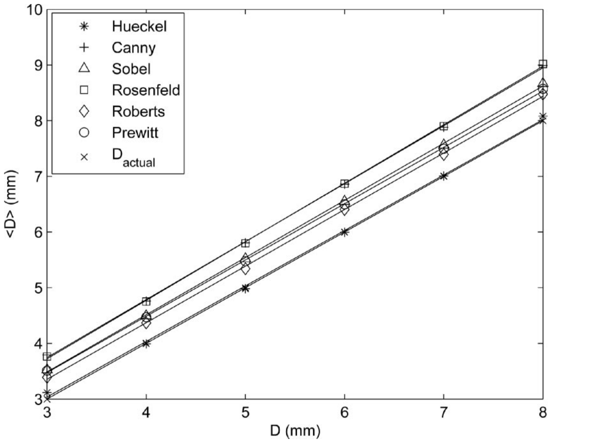


Fig. 11. Plot of $\langle D\rangle$ versus D (the actual diameter) for each of the edge detection algorithms.

The average diameter $\langle D\rangle$ was calculated using Eq. (21) for each sphere diameter for all of the algorithms. A cubic polynomial $P(z)$ was found to accurately fit the D_m versus z data in all cases. This cubic polynomial is shown in the D_m versus z plots in Figs. 8 and 9. A plot of $\langle D\rangle$ versus the actual sphere diameter, D, is presented in Fig. 11, and the data appearing in these plots are presented in Table 2. Linear fits to the data are also presented in Fig. 11, and the corresponding slopes and intercepts are presented in Table 3. For each of the edge detection algorithms, the slope of the linear fit is very close to one. Also, as Table 2 and Fig. 11 show, the average measured diameters obtained are always greater than the actual diameter for all of the algorithms except for the Hueckel algorithm.

5. Discussion

All of the images obtained in this study were of spheres, and hence the output images obtained from each algorithm investigated should be circular in shape. This was not the case for some of the edge detection algorithms whose binary images were slightly oval in shape (see Fig. 7). The distortion from a circular shape was due to the different pixel resolutions: 0.05 mm/pixel in the x-direction, and 0.1 mm/pixel in the y-direction. Edge detection algorithms use gradients and approximations to gradients to detect edges. These gradients were computed on a pixel basis. That is to say, the intensity change was computed over a certain number of pixels, *not* over a certain physical length. Because the pixel resolutions were unequal in the x- and y-directions, some of these algorithms generated images that were oval.

To determine the degree of distortion in these binary images, the ratio α was calculated according to Eq. (6) where h and v are the horizontal and vertical extent of the image, respectively. For a perfectly circular image $\alpha = 1$. For each of the edge detection algorithms, α was computed at the beginning of the *dof*, at the focal point and at the end of the *dof*. The deviation of α from unity was maximum for the Canny and Rosenfeld algorithms where it was 5%; for all other algorithms the deviation was less than 2%. This flaw can be easily corrected by either using a camera with equal x and y-direction pixel resolutions, or by computing gradients with respect to distance as opposed to pixels. This would be critical for computing the e versus D relationship needed for dual-polarization radars. Since such computations were not done in the present work, the gradients were computed with respect to pixels for the sake of simplicity.

In this work, the *dof* was obtained by manually inspecting the binary images obtained from each algorithm to determine the presence or absence of a hole. This inspection technique worked for the current study because the number of images processed was relatively small, since the purpose of the study was only to evaluate algorithms. This visual inspection procedure becomes untenable in an actual field application where the number of images to be processed becomes very large. To avoid these problems,

Table 1. Depth of Field *dof* (cm) for the Different Edge Detection Algorithms for Each Sphere Diameter

D	Prewitt	Sobel	Canny	Roberts	Template 3	Rosenfeld	Hueckel
8 mm	15.9	20.2	23	15.9	19.3	22.4	26.6
7 mm	15.2	18.2	21.1	12.9	17.6	19.1	24.1
6 mm	14.2	18.0	19.9	14.3	16.4	18.0	21.6
5 mm	14.2	16.8	17.7	11.7	16.3	14.8	18.3
4 mm	13.1	15.3	14.7	12.3	14.3	11.7	14.5
3 mm	10.5	11.9	11.2	7.9	11.7	6.8	8.7

Table 2. Average Measured Diameter $\langle D\rangle$ (mm) for the Different Edge Detection Algorithms for Each Sphere Diameter

D	Prewitt	Sobel	Canny	Roberts	Template 3	Rosenfeld	Hueckel
8 mm	8.5721	8.6683	8.9894	8.4775	8.5987	9.0171	8.0807
7 mm	7.4926	7.5624	7.8814	7.3923	7.5138	7.9009	7.0075
6 mm	6.4974	6.5516	6.8587	6.4043	6.4725	6.8680	5.9959
5 mm	5.4756	5.5202	5.8042	5.3405	5.4707	5.7984	4.9831
4 mm	4.4710	4.4954	4.7587	4.3683	4.4396	4.7532	3.9938
3 mm	3.5104	3.5254	3.7915	3.3923	3.4842	3.7639	3.1160

an algorithm can be developed which identifies the presence or absence of a hole. Saxena and Saylor [34] utilized a boundary counting algorithm for this purpose. Because the characteristics of the holes observed in the images generated by edge detection algorithms are different than those obtained using the thresholding methods presented in Saxena and Saylor [34], some modifications of their boundary counting algorithm would be needed. Additionally, that algorithm did not work perfectly, occasionally classifying out-of-focus drops as in-focus. Hence, some attention is needed in this area, and this is left as future work.

A. D_m versus z

As noted in Section 4, the D_m versus z behavior did not vary significantly among algorithms. Figure 8 plots D_m versus z for the Canny algorithm for each of the sphere diameters. The plots show that the value of the measured diameter is high near z_s, then decreases in the region near the focal point, and increases again in the region near z_e. All of the edge detection algorithms considered here showed similar behavior. The only exception to this is the Hueckel algorithm, which is plotted in Fig. 9 and shows an almost flat straight line, indicating that the measured diameter does not vary significantly with position within the *dof*. When compared to the results obtained from the other edge detection algorithms, the Hueckel algorithm has the smallest variation of the measured diameter with z. One of the goals of this work was to determine which edge detection algorithm results in a minimum variation of the measured diameter over the *dof*, and the Hueckel algorithm clearly gives the best performance in this regard.

It should also be noted that all of the edge detection algorithms investigated here performed better as a group, when compared to other methods of identifying and measuring raindrops images. Figure 12 presents D_m versus z plots for two of the edge detection algorithms presented here, and another obtained using a histogram modification technique due to Peleg [70]. As the figure shows, the degree of variation in D_m versus z behavior is small among the two edge detection algorithms, when compared to that for the Peleg algorithm.

All of the D_m versus z data obtained in this work show that D_m is closest in value to D in the center of the *dof*, near the focal point $z = 200$ cm, which is expected. The plots of D_m versus z also show that the *dof* extends farther to the right side of the focal point than to the left. This is because the spheres to the right are closer to the light source than the spheres to the left of the focal point. When the spheres are closer to the light source, a larger number of rays fall on the sphere, causing the image of the lamp in the center of the sphere image to be brighter, increasing the likelihood that the algorithm will result in an image with a hole. Another feature observed in the plots of D_m versus z is that the D_m values are larger in the region between the camera and the focal point than in the region between the focal point and the lamp. This result is expected, since objects naturally appear larger when closer to the camera and smaller when far from it. This creates finite error in D_m. However this error is relatively small when compared to nonedge-detection methods, as shown in Fig. 12.

B. Depth of Field

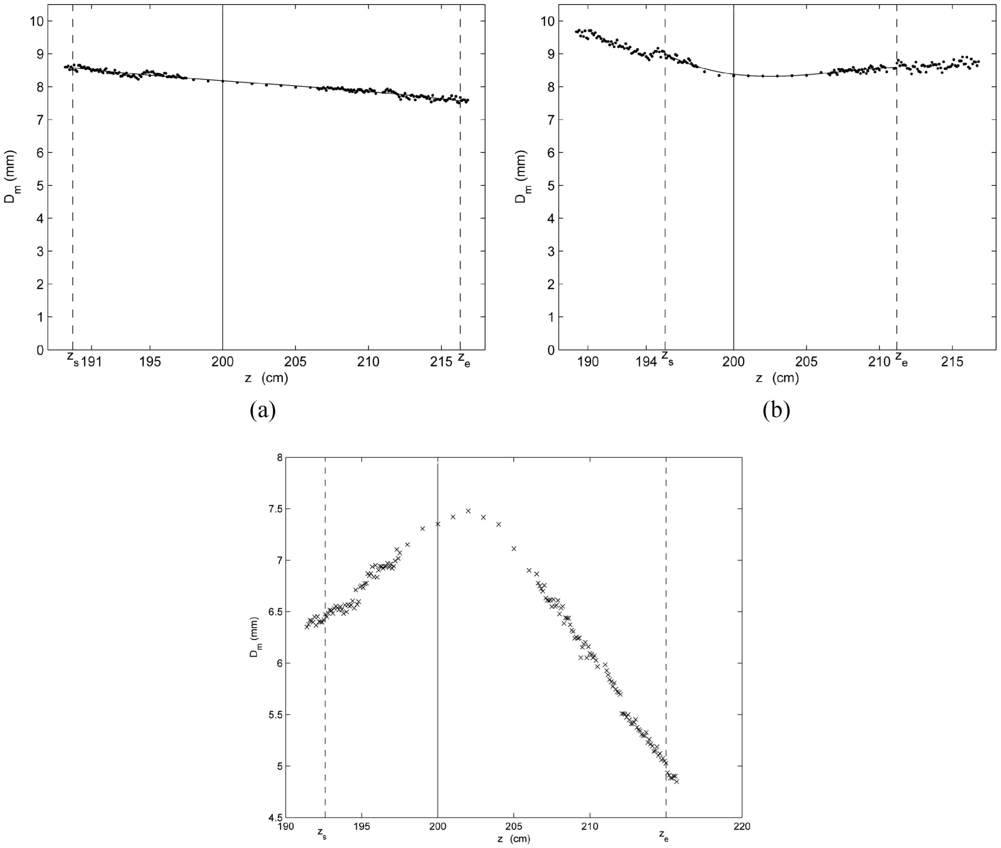
The depth of field increases with D for all of the edge detection algorithms, as can be seen from Fig. 10 and Table 1. The Hueckel algorithm gives the maximum *dof* for all diameters except the 3 and 4 mm spheres where the Sobel edge operator gave a larger *dof*. Figure 10 shows that the *dof* versus D relationship is erratic for all of the algorithms except the Hueckel algorithm where the variation is smooth. Indeed, the *dof* versus D data for the Hueckel algorithm can be approximated using a quadratic fit, as shown in Fig. 13. This is useful since, in a field application, it provides a means for determining the *dof* for each drop size category measured.

C. Average Diameter

As Fig. 11 shows, the average measured diameter $\langle D\rangle$ varies linearly with the actual diameter D for all the edge detection algorithms considered here. A line of unity slope and zero intercept is included in this fig-

Table 3. Slope and Intercept of the Linear Fits to the $\langle D\rangle$ versus D Data Presented in Fig. 11

Algorithm	Slope	Intercept
Hueckel	0.99	0.04
Canny	1.04	0.62
Sobel	1.02	0.40
Rosenfeld	1.05	0.57
Roberts	1.01	0.30
Prewitt	1.01	0.44



(a)

(b)

(c)

Fig. 12. Plots of D_m versus z for two of the edge detection methods presented here: (a) Hueckel and (b) Roberts. Also included is (c) a histogram modification method due to Peleg [70]. While there are differences in the D_m versus z behavior for the edge detection algorithms presented in this work, the behavior among these algorithms is quite similar when compared to different algorithmic approaches. (plot (c) obtained from Sivasubramanian and Saylor [35]).

ure to help in comparing the deviation of the average diameter obtained from the different algorithms from the actual diameter. This figure shows that the Hueckel algorithm provides the least deviation from the actual diameter. In fact, this deviation is so small that the unity slope line and the Hueckel line are almost identical. The Rosenfeld algorithm shows the worst behavior, having the maximum deviation from the actual diameter. A linear fit to the $\langle D \rangle$ versus D data are presented for each of the edge detection algorithms in Fig. 11. These fits can be used to obtain the actual diameter D from the measured diameter for each algorithm. Hence, even for the algorithms that show a large deviation of $\langle D \rangle$ from D, D can still be extracted. The caveat to this statement is that $\langle D \rangle$ is the average of D_m obtained over the *dof*. Hence it masks any variations of D_m over the *dof*. Nevertheless, as was shown above, these variations were relatively small for all of the algorithms considered here. This notwithstanding, the deviation of D_m from D over the whole *dof* is quantified below.

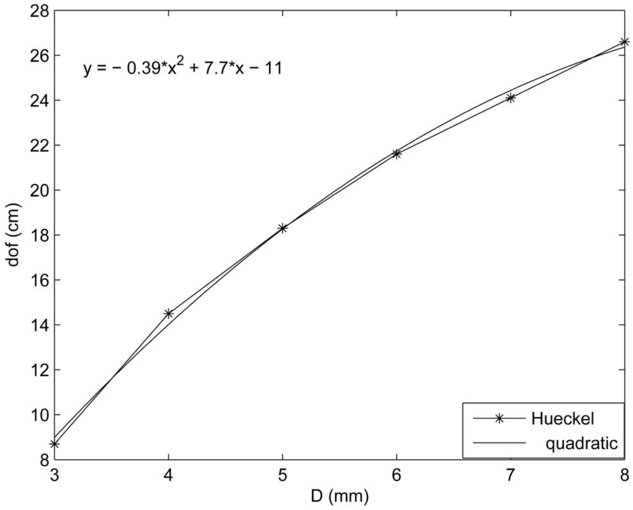


Fig. 13. Plot of *dof* versus D for the Hueckel algorithm with quadratic fit.

D. Measurement Error

To compare the algorithms considered here, their deviation from the actual diameter over the *dof* is quantified by an average error computed for each diameter and for each algorithm. For each diameter D, for each z-location within the *dof*, the error E_m was calculated as:

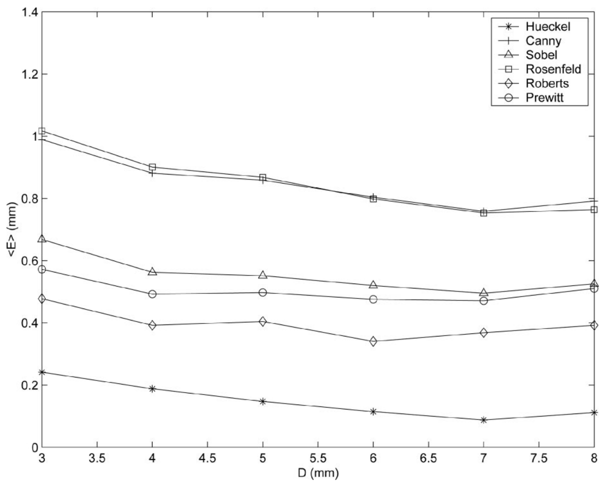


Fig. 14. Plot of $\langle E \rangle$ versus D, both in mm.

$$E_m = |D_m - D|. \tag{22}$$

An average of E_m over the *dof* was computed by fitting a curve $Q(z)$ to the E_m versus z data and computing $\langle E \rangle$:

$$\langle E \rangle = \frac{1}{dof}\int_{z_s}^{z_e} Q(z)dz. \tag{23}$$

A plot of $\langle E \rangle$ versus D is presented for each of the edge detection algorithms in Fig. 14. The figure shows that the Hueckel algorithm has the least error followed by the Roberts operator, while the Rosenfeld operator has the largest error.

The flatness of the D_m versus z plots presented here shows that, in general, the average measured diameter is not significantly influenced by the size of the depth of field. Hence, there is no trade-off between *dof* and the accuracy of measuring the diameter, which is usually the case in these applications [34,35].

6. Conclusion

A set of edge detection algorithms was evaluated to determine which algorithm would provide the most accurate measurement of raindrop diameter, when applied to raindrop imagery, thereby increasing the accuracy of the drop size distributions (DSD) obtained from this imagery. The selected algorithms were applied to images of MgF_2 spheres obtained from a laboratory setup. Images of six different sphere sizes were obtained using the lab setup and the *dof* and diameter were obtained using each of the algorithms considered. The Hueckel algorithm was found to give results that were best suited to the goals of raindrop sizing. The *dof* obtained using the Hueckel algorithm was the largest for all sphere diameters greater than 4 mm. The average diameter obtained using the Hueckel algorithm was found to be the closest to the actual diameter when compared to the other algorithms. For all of the edge detection algorithms considered, the measured diameter was not found to vary significantly over the *dof*. The smallest variation of measured diameter over the *dof* occurred for the Hueckel algorithm.

This work was supported by the National Science Foundation.

References

1. R. J. Doviak and D. S. Zrnić, *Doppler Radar and Weather Observations* (Academic, 1984).
2. V. N. Bringi and V. Chandrasekar, *Polarimetric Doppler Weather Radar* (Cambridge U. Press, 2001).
3. P. Meischner, *Weather Radar* (Springer, 2004).
4. K. V. Beard, "Oscillation models for predicting raindrop axis and backscatter ratios," Radio Sci. **19,** 67–74 (1984).
5. K. V. Beard and C. Chuang, "A new model for the equilibrium shape of raindrops," J. Atmos. Sci. **44,** 1509–1524 (1987).
6. K. V. Beard, R. J. Kubesh, and H. T. Ochs, "Laboratory measurements of small raindrop distortion, Pt. 1: axis ratios and fall behavior," J. Atmos. Sci. **48,** 698–710 (1991).
7. K. V. Beard and R. J. Kubesh, "Laboratory measurements of small raindrop distortion. Pt. 2: oscillation frequencies and modes," J. Atmos. Sci. **48,** 2245–2264 (1991).
8. K. Andsager, K. V. Beard, and N. F. Laird, "Laboratory measurements of axis ratios for large raindrops," J. Atmos. Sci. **56,** 2673–2683 (1999).
9. K. V. Beard and A. Tokay, "A field study of raindrop oscillations: observations of size spectra and evaluation of oscillation causes," Geophys. Res. Lett. **18,** 2257–2260 (1991).
10. J. Joss and A. Waldvogel, "A raindrop spectrograph with automatic analysis," Pure Appl. Geophys. **68,** 240–246 (1967).
11. J. Joss and A. Waldvogel, "Comments on 'Some observations on the Joss-Waldvogel rainfall disdrometer'," J. Appl. Meteorol. **16,** 112–113 (1977).
12. A. Tokay, A. Kruger, and W. F. Krajewski, "Comparison of drop size distribution measurements by impact and optical disdrometers," J. Appl. Meteorol. **40,** 2083–2097 (2001).
13. D. Hauser, P. Amayenc, and B. Nutten, "A new optical instrument for simultaneous measurement of raindrop diameter and fall speed distributions," J. Atmos. Ocean. Technol. **1,** 256–269 (1984).
14. M. Löffler-Mang and J. Joss, "An optical disdrometer for measuring size and velocity of hydrometeors," J. Atmos. Ocean. Technol. **17,** 130–139 (2000).
15. G. Donnadieu, "Mesure de la vitesse terminale des gouttes du pluie au sol à l'aide du spectropluviomètre VIDIAZ," J. Rech. Atmos. **12,** 245–259 (1978).
16. G. Donnadieu, "Comparison of results obtained with the VIDIAZ spectropluviometer and the Joss-Waldvogel rainfall disdrometer in a 'rain of a thundery type'," J. Appl. Meteorol. **19,** 593–597 (1980).
17. M. Grossklaus, K. Uhlig, and L. Hasse, "An optical disdrometer for use in high wind speeds," J. Atmos. Ocean. Technol. **15,** 1051–1059 (1998).
18. J.-Y. Delahaye, L. Barthès, P. Golé, J. Lavergnat, and J. P. Vinson, "A dual-beam spectropluviometer concept," J. Hydrol. **328,** 110–120 (2006).
19. S. Borrmann and R. Jaenicke, "Application of microholography for ground-based in situ measurements in stratus cloud layers: a case study," J. Atmos. Ocean. Technol. **10,** 277–293 (1993).
20. M. Schönhuber, H. E. Urban, J. P. V. Poiares-Baptista, W. L. Randeu, and W. Riedler, "Measurements of precipitation characteristics by a new distrometer," in *Proceedings of Atmospheric Physics and Dynamics in the Analysis and Prognosis of Precipitation Fields* (SIMA, 1994).
21. T. J. Schuur and A. V. Rhyzhkov, "Drop size distributions measured by a 2D video disdrometer: comparison with dual-

polarization radar data," J. App. Meteorol. **40,** 1019–1034 (2001).

22. A. Kruger and W. F. Krajewski, "Two-dimensional video disdrometer: a description," J. Atmos. Ocean. Technol. **19,** 602–617 (2002).
23. V. Nĕspor, W. F. Krajewski, and A. Kruger, "Wind-induced error of raindrop size distribution measurement using a two-dimensional video disdrometer," J. Atmos. Ocean. Technol. **17,** 1483–1492 (2000).
24. R. G. Knollenberg, "The optical array: an alternative to scattering or extinction for airborne particle size determination," J. Appl. Meteorol. **9,** 86–103 (1970).
25. R. G. Knollenberg, *Clouds: Their Formation, Optical Properties, and Effects* (Academic, 1981), pp. 15–89.
26. T. W. Cannon, "Imaging devices," Atmos. Technol. **8,** 32–37 (1976).
27. R. A. Black and J. Hallett, "Observations of the distribution of ice in hurricanes," J. Atmos. Sci. **43,** 802–822 (1986).
28. G. Frank, T. Härtl, and J. Tschiersch, "The pluviospectrometer: classification of falling hydrometeors via digital image processing," Atmos. Res. **34,** 367–378 (1994).
29. L. Bliven (personal communication, 2001).
30. J. R. Saylor, B. K. Jones, and L. F. Bliven, "Single-camera method to determine the optical axis position of ellipsoidal drops," Appl. Opt. **42,** 972–978 (2003).
31. J. R. Saylor, B. K. Jones, and L. F. Bliven, "A method for increasing depth of field during droplet imaging," Rev. Sci. Instrum. **73,** 2422–2427 (2002).
32. K. S. Fu and J. K. Mui, "A survey on image segmentation," Pattern Recogn. **13,** 3–16 (1981).
33. K. R. Castleman, *Digital Image Processing* (Prentice Hall, 1996).
34. D. D. Saxena and J. R. Saylor, "Use of thresholding algorithms in the processing of raindrop imagery," Appl. Opt. **45,** 2672–2688 (2006).
35. N. A. Sivasubramanian and J. R. Saylor, "Application of a histogram modification algorithm to thresholding of raindrop images," submitted to Opt. Eng.
36. T. Peli and D. Malah, "A study of edge detection algorithms," Computer Graph. Image Process. **20,** 1–21 (1982).
37. L. G. Roberts, "Machine perception of three-dimensional solids," in *Optical and Electro-optical Information Processing,* J. T. Tippet, D. A. Berkowitz, L. C. Clapp, C. J. Koester and A. Vanderburgh, eds. (MIT Press, 1965), pp. 159–197.
38. J. H. G. Hale, "Dectection of elementary features in a picture by non-linear local numerical processing," in *Proceedings of the 3rd International Joint Conference on Pattern Recognition* (1976), pp. 764–768.
39. A. Rosenfeld, "A nonlinear edge detection technique," Proc. IEEE **58,** 814–816 (1970).
40. A. Rosenfeld and M. Thurston, "Edge and curve detection for visual scene analysis," IEEE Trans. Comput. **C-20,** 562–569 (1971).
41. A. Rosenfeld, M. Thurston, and Y.-H. Lee, "Edge and curve detection: further experiments," IEEE Trans. Comput. **C-21,** 677–715 (1972).
42. G. B. Shaw, "Local and regional edge detectors: some comparisons," Computer Graph. Image Process. **9,** 135–149 (1979).
43. I. E. Sobel, *Camera Models and Machine Perception* (Stanford U. Press, 1970).
44. L. Mero and Z. Vassy, "A simplified and fast version of the Hueckel operator for finding optimal edges in pictures," in *Proceedings of the 4th International Joint Conference on Artificial Intelligence* (ASCE, 1975), pp. 650–655.
45. R. A. Hummel, "Edge detection using basis functions," Technical Report TR-569 (University of Maryland, 1977).
46. M. H. Hueckel, "An operator which locates edges in digitized pictures," J. Assoc. Comput. Mach. **18,** 113–125 (1971).
47. M. H. Hueckel, "A local visual operator which recognizes edges and lines," J. Assoc. Comput. Mach. **20,** 634–647 (1973).
48. M. C. Shin, D. B. Goldgof, K. W. Bowyer, and S. Nikiforou, "Comparison of edge detection algorithms using a structure from motion task," IEEE Trans. Syst. Man Cybern. **31,** 589–601 (2001).
49. M. J. Black, G. Sapiro, D. H. Marimont, and D. Heeger, "Robust anisotropic diffusion," IEEE Trans. Image Process. **7,** 421–432 (1998).
50. F. Bergholm, "Edge focussing," IEEE Trans. Pattern Anal. Mach. Intell. **9,** 726–741 (1987).
51. J. Canny, "A. computational approach to edge detection," IEEE Trans. Pattern Anal. Mach. Intell. **8,** 679–698 (1986).
52. L. Rosenthaler, F. Heitger, O. Kübler, and R. von der Heydt, "Detection of general edges and keypoints," in *Proceedings of the European Conference on Computer Vision,* (1992), pp. 78–86.
53. C. A. Rothwell, J. L. Mundy, W. Hoffman, and V.-D. Nguyen, "Driving vision by topology," in *IEEE International Symposium on Computer Vision* (IEEE, 1995), pp. 395–400.
54. S. Y. Sarkar and K. L. Boyer, "Optimal infinite impulse response zero crossing based edge detectors," Comput. Vision Graph. Image Process. **54,** 224–243 (1991).
55. S. M. Smith and J. M. Brady, "SUSAN—A new approach to low level image processing," Int. J. Comput. Vision **23,** 45–78 (1997).
56. M. D. Heath, S. Sarkar, T. Sanocki, and K. W. Bowyer, "Comparison of edge detectors, a methodology and initial study," Comput. Vison Image Understand. **69,** 38–54 (1998).
57. V. S. Nalwa and T. O. Binford, "On detecting edges," IEEE Trans. Pattern Anal. Mach. Intell. **8,** 699–714 (1986).
58. J. R. Fram and E. S. Deutsch, "On the quantitative evaluation of edge detection schemes and their comparison with human performance," IEEE Trans. Comput. **24,** 616–628 (1975).
59. I. D. G. Macleod, "On finding structure in pictures," in *Picture Language Machines,* S. Kaneff, ed. (Academic, 1970), pp. 231–256.
60. I. D. G. Macleod, "Comments on techniques of edge detection," Proc. IEEE **60,** 344 (1972).
61. K. C. Hayes and A. Rosenfeld, "Efficient edge detectors and applications," Technical Report TR-207 (University of Maryland, 1972).
62. I. E. Abdou and W. Pratt, "Quantitative design and evaluation of enhancement/thresholding edge detectors," Proc. IEEE **67,** 753–763 (1979).
63. J. M. S. Prewitt, "Object enhancement and extraction," in *Picture Processing and Psychopictorics,* B. S. Lipkin and A. Rosenfeld, eds. (Academic, 1970).
64. R. Kirsch, "Computer determination of the consitituent structure of biological images," Comput. Biomed. Res. **4,** 315–328 (1971).
65. R. Nevatia, "Evaluation of a simplified Hueckel edge-line detector," Comput. Graph. Image Process. **6,** 582–588 (1977).
66. M. Sonka, V. Hlavac, and R. Boyle, *Image Processing Analysis and Machine Vision* (Thomson-Engineering, 1999).
67. D. Vernon, *Machine Vision* (Prentice Hall, 1991).
68. E. Trucco and A. Verri, *Introductory Techniques for 3-D Computer Vision* (Prentice Hall, 1998).
69. J. Lucas, F. Smektala, and J. L. Adam, "Fluorine in optics," J. Fluorine Chem. **114,** 113–118 (2002).
70. S. Peleg, "Iterative histogram modification, 2," IEEE Trans. Syst. Man Cybern. **8,** 555–556 (1978).