University of Nevada
Reno

Automated Digital Image Analysis of Video Ice Crystal Data

A professional paper submitted in partial fulfillment of the
requirements for the degree of Master of Science
with a major in Computer Science.

by

Jane Niehues-Brooks

Dr. Frederick C. Harris, Jr., advisor

December 1997

Abstract[1]

This paper presents a procedure for automating digital image analysis of cloud particle images recorded on video tape using the Cloudscope.[2] The Cloudscope is mounted on the wing of an aircraft and records cloud crystals as they impact the external lens and sublimate. The procedure presented breaks the analysis into several parts: frequency domain filtering, frame to frame movement correction, and new particle detection, counting and sizing. These procedures were automated using LabVIEW and Concept V.i.

[1]This project was funded by a NASA Ames contract NAG 2-1104

[2]This instrument is operated by the Atmospheric Science Center at Desert Research Institute.

Contents

List of Figures

1 Introduction

This paper discusses an analytic tool for Cloudscope ice measurement. We have implemented a technique for automating the digital image analysis of cloud particle data recorded using the Cloudscope instrument. The Cloudscope is a specialized data collection device operated for atmospheric research by the Atmospheric Science Center at Desert Research Institute. This instrument mounts on the wing of an aircraft and is essentially a microscope with an attached video camera. Ice crystals and cloud droplets impact on the small window of the Cloudscope and are recorded by the video system (Hi8 video tape). A line drawing of this instrument is shown in Figure 1 [1]. The goal of this project was to automate the counting and sizing of the cloud crystals recorded and in future work add the ability to compute their the sublimation rates.

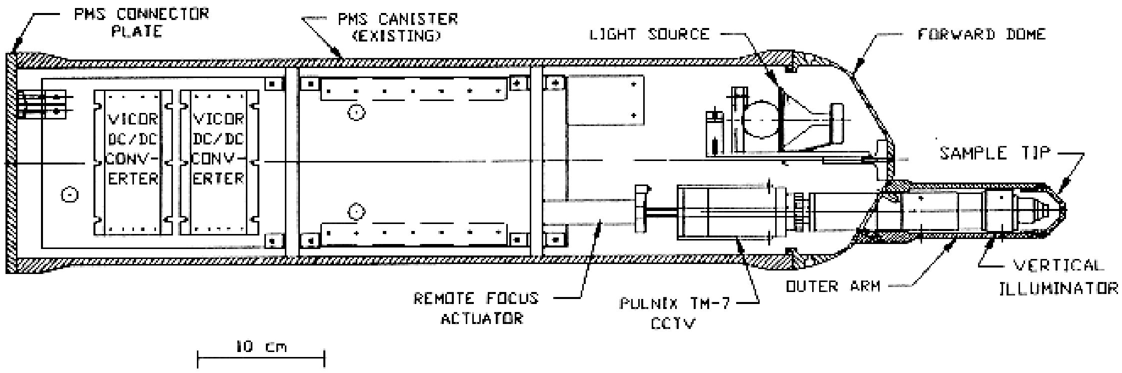

Figure 1: Cloudscope Line Drawing

The automated processing system is made up of a programmable Sony-Visca Hi-8 VCR and an Apple Macintosh Centris 650 with a Scion LG-3 Frame Grabber Card. The frame grabber is loaded with 64 MBytes of RAM and is capable of grabbing up to 128 frames of video, which is equivalent to 4 seconds of flight time at a 30 frames per second capture rate. We created a virtual instrument to perform the automation and image analysis using National Instrument's LabVIEW instrument control software and Graftek's Concept V.i image analysis software. A similar project for the Replicator instrument [5] was used as a guide in implementing the LabVIEW and Concept V.i tools. Virtual instruments will be referred to as Vi's throughout this paper. Many of the Vi's available in LabView and Concept V.i are used as functional building blocks toward the creation of our CloudScope Autoanalysis Vi. The CloudScope Autoanalysis Vi is also made up of several of our own more complex image analysis modules.

Section 2 discusses the actual techniques used to perform the image processing. The Image Processing is broken down into three areas: noise removal, image alignment between frames, and new particle detection, sizing and counting. Section 3 gives an overview of CloudScope Autoanalysis Vi as well as a discussion of future studies and unresolved problems encountered. Concluding comments and insights follow in Section 4.

2 Image Processing Algorithm

To succeed in analyzing the cloudscope video data there are three basic tasks that have to be accomplished. These tasks include: filtering noise, correcting for image jitter, and detecting, counting and sizing new particles. The frame grabber can capture up to 128 frames of video at a time. Once captured, each image goes through a sequence of steps to clean it up. After the image is cleaned up, a threshold is used to change it to a binary image of white particles on a black background. A comparison with the prior image is performed and new particles present are identified and sized. In principle this is a very simple process. Unfortunately, the Cloudscope experiences both electrical interference and physical instability. These issues result in flight video recordings containing excessive movements from frame to frame, color variations due to inconsistent lighting, considerable high frequency signal interference in the images along with false particles caused by tiny scratches and foreign matter on both the CCD lens and the impact lens. The ability to successfully threshold an image into particles and background is the key to automating the counting and sizing of the crystals. Unless preprocessing is performed on each image, any of the above problems will contribute to unsuccessfully thresholding the image, thereby making it impossible to automate the particle count. For our processing purposes we use a 256 by 256 image. When captured, each image is 640 by 480, but we choose to transfer in every other line in both the X and Y direction for two reasons. The most obvious being that a smaller image will process faster, and the second, to handle a problem caused by the interlacing of the image. If a crystal impacts on the lens after the odd rows have been recorded, it will have a striped appearance due to just the even rows being recorded and would not be detected as a single particle. Using every other line of the image alleviates this problem. An adjustment to the image size must also be made. Since our Fast Fourier Transform (FFT) and Inverse Fourier Transform (IFT) routines require dimensions in powers of two, we extract a 256 by 240 portion of the image and expand the image in the Y direction from 240 to 256. This expansion causes an aspect ratio change, which we can correct for by implementing the calibration Vi discussed later in this section.

2.1 Noise Removal Method

The noise on the videos along with the wide color variations result in images that are impossible to threshold properly. Figure 2[3] is an example of a noisy low contrast image. The noise in this image is more apparent when using an enhancement technique called contrast stretching. This is a

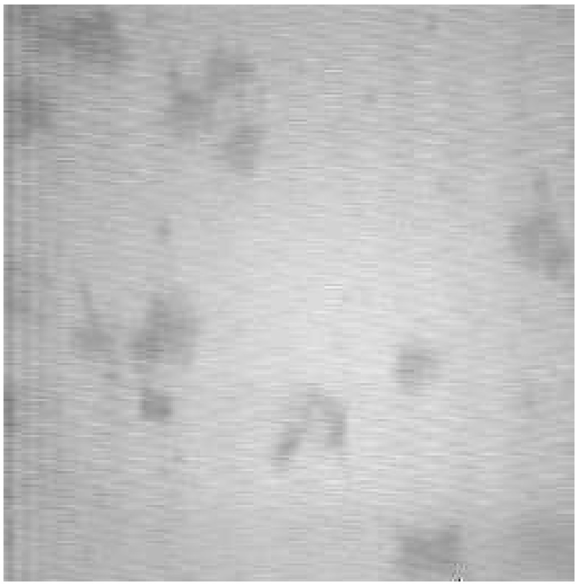

Figure 2: Unprocessed noisy image with poor contrast.

method of increasing the image dynamic range by spreading the images narrow range of gray levels over the available range of 256 colors. This technique will be used on the images throughout this paper for better viewing. Figure 3 is the same image after contrast enhancement. Figure 4 is the

[3] All images are from the NASA SUCCESS Project, May 3, 1996 Flight

resulting image after thresholding. As can be seen on the thresholded image not all particles fall within the same gray level range. Consequently, only a few of the actual crystals present would be detected.

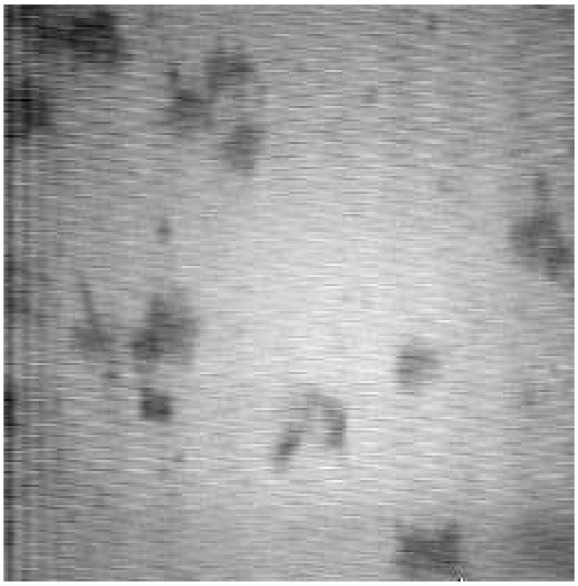

Figure 3: Unprocessed noisy image with improved contrast.

To investigate the noise present, we looked at the Fourier spectrum of the images in the frequency domain. The frequency spectrum of Figure 2 is shown in Figure 5. The vertical lines on either side of the center line in the spectrum show the high frequency components corrupting the image. From this it is apparent that some of the preprocessing must be performed in the frequency domain. An FFT is used to transform the image from the spatial domain to the frequency domain. We have chosen to use a frequency filter mask to remove the undesirable frequency components from the image. Our mask is shown in Figure 6. In order to use the mask as a frequency filter, it is necessary to convert it into a complex image made up of complex zeros and ones for the black and white elements respectively. The lowest frequency color variation is due in a large part to the shape of the impact lens. A property known as vignetting is the absorption of light in the thicker edges of the lens, which causes the image to be darker on the edges than in the center [4], page 11. Other contributors to the light variation are our artificial light source, daylight, flying into the sun, and night flights. This slowly varying color can be corrected by blocking out the lowest

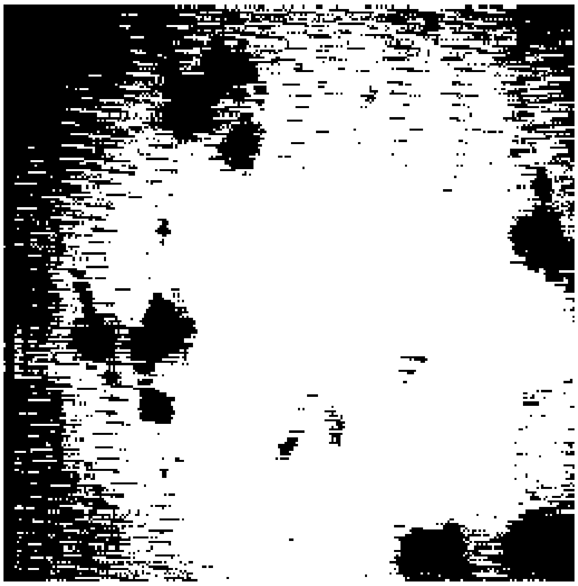

Figure 4: Unprocessed noisy image after thresholding.

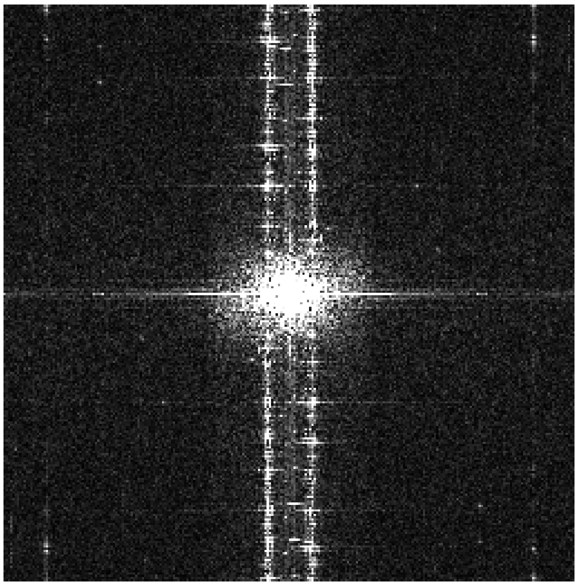

Figure 5: The frequency spectrum of the noisy image.

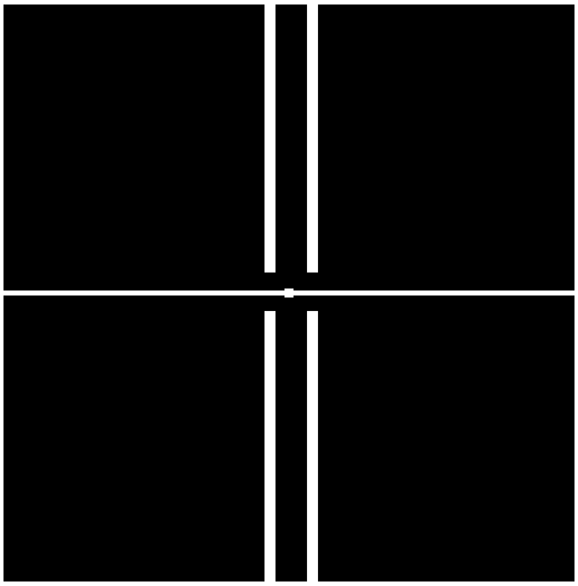

Figure 6: Our frequency filter mask.

frequency, also known as the central–order component, located at the center of our spectrum. This technique is confirmed by discussions in [4] pages 10 to 12 and [3] pages 144 to 145. The IFT of the image information contained in this blocked area is shown in Figure 7. The horizontal center line in the mask removes the vertical wavelike ridges that traverse horizontally from left to right across the image. This interference pattern is best seen by looking at the left edge of the image in Figure 3. This filtering idea was generated from the optical grid example in [3] pages 142–144, where a vertical slit removes the horizontal grid component from view and a horizontal slit removes the vertical components. Figure 8 shows the image data contained in this horizontal line of the spectrum. The last features of our mask are made up of two notch filters. The vertical bars in the upper half of the spectrum notch out the high frequency impulses as do their symmetric pairs in the lower half of the spectrum. Figure 9 shows the image data in one of these filter pairs. We actually retained all of the noise components by multiplying the spectrum of Figure 5 by the mask shown in Figure 6. Figure 10 shows our inverted mask superimposed on the frequency spectrum so that you can see which of the frequency components will be saved. Any frequency component we wish to save is multiplied by the (1+0i) complex mask element and the rest are multiplied by the (0+0i) complex element. We transform the result of this multiplication back into the spatial domain using

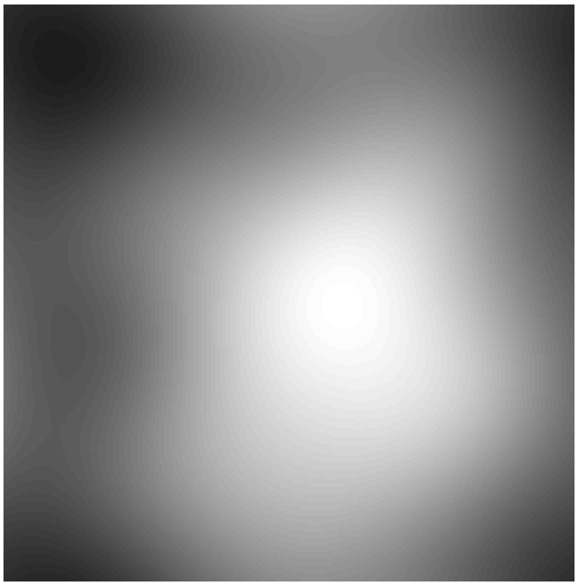

Figure 7: The slow varying low frequency components of our image.

an Inverse Fourier Transform. The noise image, shown in Figure 11, is then shifted down in color and subtracted from Figure 2. With the noise removed, we enhance the image further by increasing its dynamic range. The resulting high contrast image with the corrupting noise removed is given in Figure 12.

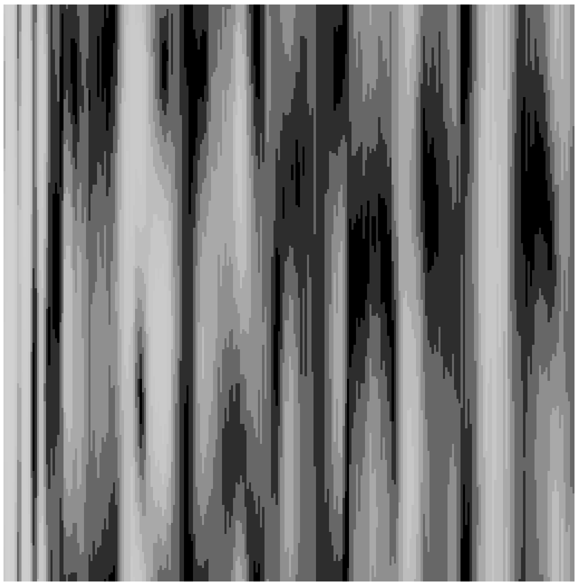

Figure 8: Image data filtered by the horizontal notch.

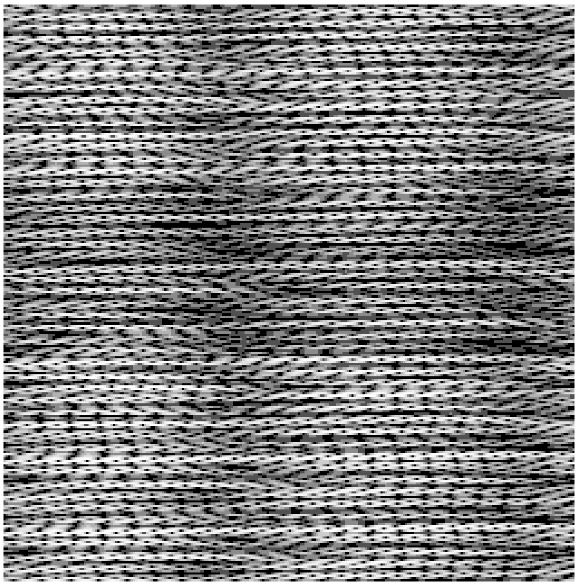

Figure 9: High frequency interference removed by the vertical notch.

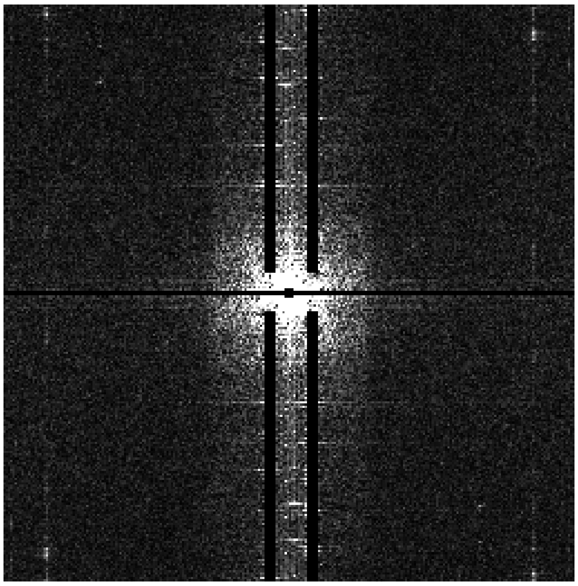

Figure 10: The inverted mask superimposed on the frequency spectrum.

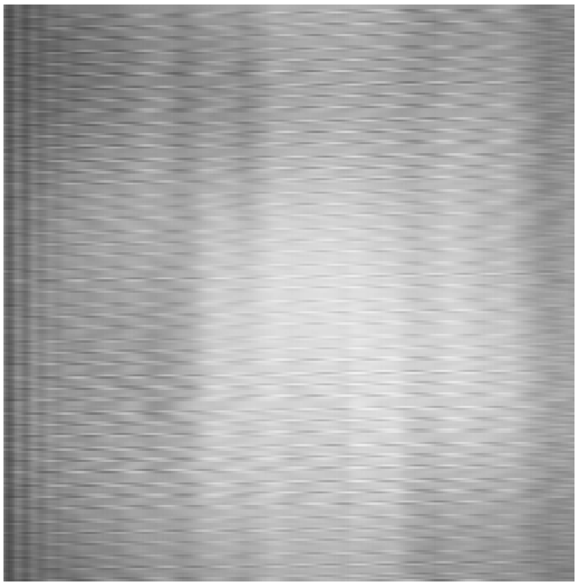

Figure 11: Inverse Fourier Transform of the noise components of the image

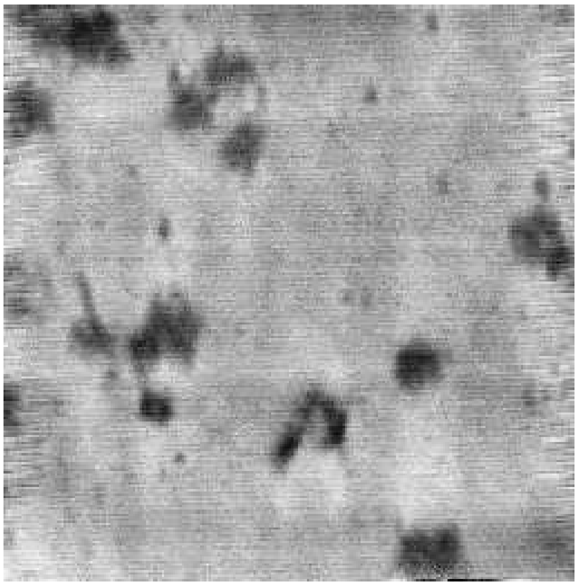

Figure 12: High contrast image after processing (image 0).

2.2 Image Alignment Technique

As we have seen, processing of the cloudscope data is not as straight forward as it would appear. Another hurdle to overcome before any reliable data analysis can be performed is to correct for the image movement from frame to frame. Plane vibration from turbulence and electromechanical problems are more than likely the cause of this jitter. Evidence of this problem is reflected in the two consecutive images of Figure 13 and Figure 14. Notice the change in position of the particle at the bottom of the images. Image movement in this section of our video tape has been measured in excess of +/- 14 pixels. This is considerable since it lies within the size range of the crystals we hope to measure. In order to successfully detect a new particle, the two consecutive images must have the same reference point.

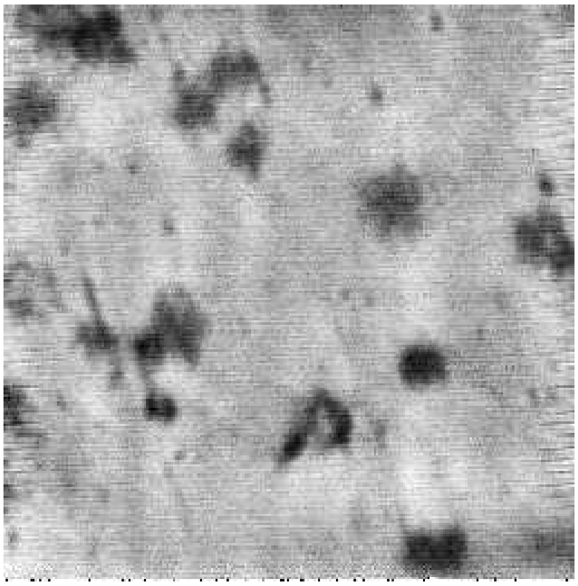

Figure 13: First of two consecutive images after processing (image 1).

Several techniques for aligning the images were tried with various levels of success. Such as: using the first image as a reference and aligning all following images relative to a fixed particle on that first image; selecting a target particle that is present on two consecutive images and aligning them based on the difference in the center of mass coordinates of the two matched particles; and performing a crosscorrelation between the two images. The crosscorrelation required computing the FFT of each image, taking the complex conjugate of one, multiplying it by the other, and then

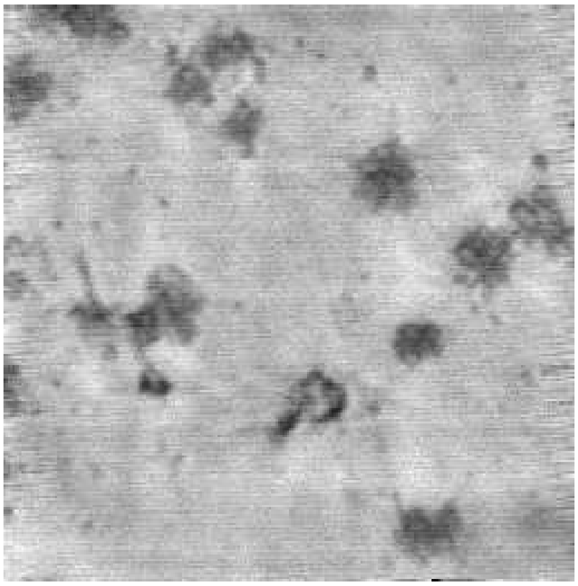

Figure 14: Second of two consecutive images after processing (image 2).

computing the IFT of the product. As a reference, for digital images the discrete version of this computation as presented in [2] is given in equation 1. Note: The subscript e refers to extended arrays described on pgs 107-110 of [2].

$$f_e(x,y) \circ g_e(x,y) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f_e^*(m,n) g_e(x+m, y+n) \tag{1}$$

Since we are already taking the time to compute the FFT of each image we find that the correlation computation can be just as efficiently done in the frequency domain by implementing the following *correlation theorem*:

$$f(x,y) \circ g(x,y) \Leftrightarrow F^*(u,v)G(u,v) \tag{2}$$

Performing an IFT on the right side of theorem 2 results in an image which holds the key to the offset between the two frames. The offset coordinates used in aligning the two images are computed as the difference in both the X and Y directions from the center of the crosscorrelation image to the coordinates of the largest value of this image. The largest value of the crosscorrelation indicates where the best match was found between the two images. Although we believe this last method to be the ideal solution, due to limitation in using the FFT functions we resorted to a form of manual correlation that gave us good results, but not without some error. With our manual method we find all matching particles between two frames and compute their average center of mass difference.

These averages are then used as the alignment offsets. This method is limited by the amount of jitter allowed and the density and similarity of the particles.

In aligning the two images, the prior image is held fixed and the current image is realigned by extracting all or a portion of the image base on the offset values, and then pasting it back onto a white background. A white background is used so that any particle lying on, or touching the boarder, will still be considered a boarder particle when the image is aligned. Figure 15 shows the result of aligning image 2 with image 1. A correction of +3 in the X–direction and +12 in the Y–direction is made. For most particles in the image, this is a fairly accurate correction, but this method is not without exceptions. There are always a few specks or scratches present on the CCD lens that show up in the images. Because of their physical location, they do not jump from frame to frame as do the particles on the impact lens. This introduces an error back into the analysis because these particles have just been misaligned causing them to be counted as new particles. It is possible that as a second stage to this procedure, we could reverse the alignment and subtract out the CCD lens noise and hope that no valid data gets removed. An alternate plan would be to leave it to some post processing number crunching where erroneous data can be identified and deleted from the data set.

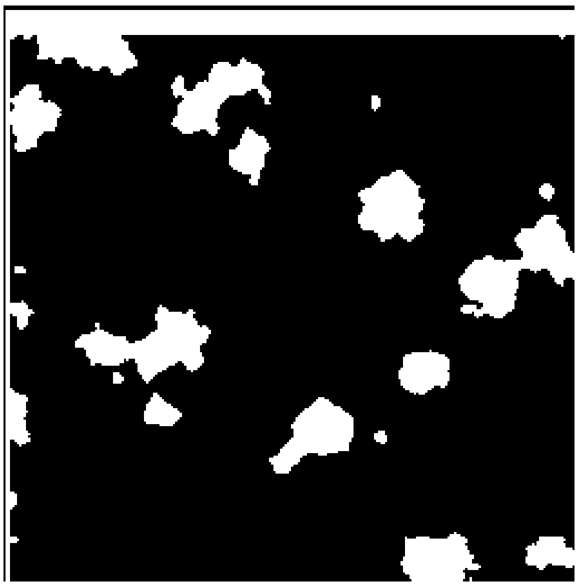

Figure 15: Binary version of image 2 after alignment.

2.3 New Particle Detection and Counting

Given two clean aligned consecutive images, we are then able to count and size the particles. When counting, we are only concerned with the new particles in an image. A new particle is one that was not present on the previous frame. The size of a particle is the area it occupies within the image. Figure 16 is an example of a filtered image after thresholding. This image is still not as clean as we would like so we add a median filtering step. This is a spatial convolution process in which we keep the median value of all neighboring pixels. Figure 17 gives the result of this process in which the furry edges and small background noise have been removed. Figure 17 and Figure 15 are the two consecutive images of Figures 13 and 14 respectively, after thresholding, median filtering, and aligning.

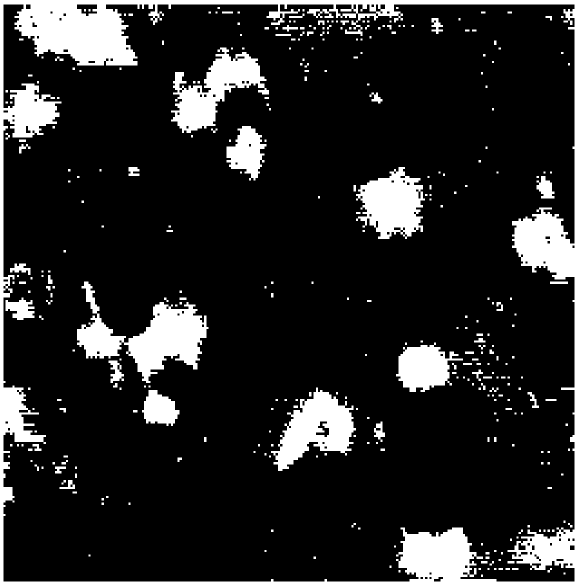

Figure 16: Processed image 1 after thresholding

For a particle to be counted it must be surrounded by black background. If a particle touches or overlaps the image border, it is considered outside the field of view. Since its actual size is not known, it is removed from the image. This removal is handled in the *reject border* processing stage just prior to counting and sizing. Figure 18 shows Figure 15 after border particle rejection has taken place. Notice that both of the particles located at the middle of the right border are removed because they are touching each other and the border. In this case it is debatable

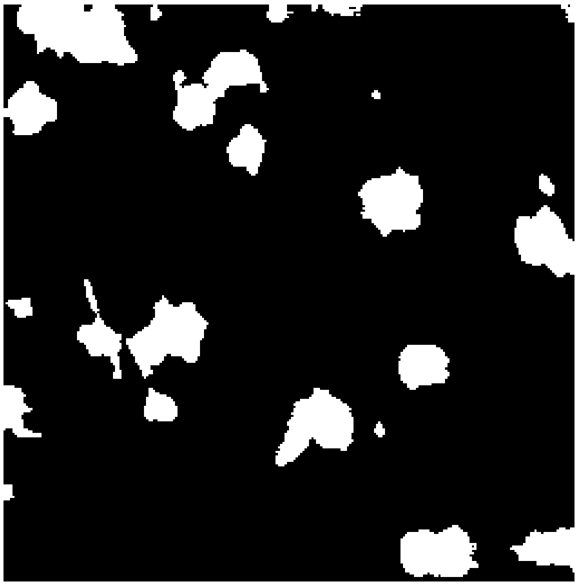

Figure 17: Image 1 after median filtering.

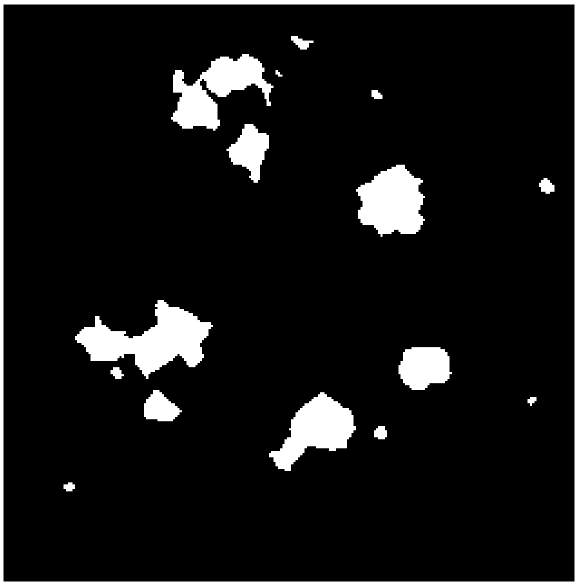

Figure 18: Image 2 after border particle rejection.

whether all border particles should be rejected. Future advancements in processing techniques may minimize the loss of valid data such as this. The method used to determine whether a particle is new or not is to compute the logical difference between two frames. Labview provides a logical difference function that operates on two binary images. If a pixel is white in the current image but not in prior image, then it is retained. Otherwise it is changed to black. The following figures show the result of processing three frames of video. Figure 19 is the image resulting from performing the logical difference between the thresholded images of Figures 12 and 13. and Figure 20 is the same image after eroding by a factor of one. Figure 21 is the image resulting from performing the logical difference between the thresholded images of Figures 13 and 14. and Figure 22 is the same image after erosion. As can be seen, a few residual edges remain, but most are removed through the *erosion* process. Erosion can be specified as having a factor of one or more. If, for example, two is selected and a particle is less than or equal to two in all directions, that particle will be removed from the image. Ideally, the only pixels that are eroded are those that were remnants of particle edges or noise, and not actual tiny ice crystals. After erosion, those particles remaining are considered *new*.

Finally, the *new* particles can be counted and sized. The size of a particle is based on the number of pixels that make up its contiguous white area. LabVIEW provides a special Vi that allows the implementor to select specific particle characteristics to be measured. The parameters selected for our analysis include; area, max intercept length, mean perpendicular intercept, perimeter, center of mass X and Y coordinates, and the particles bounding box coordinates. Once the measurement is complete, the data is recorded in an array that will be written to a file after the set of frames have been analyzed.

The smallest measurable particle is largely dependent on the erosion factor. Using an erosion factor of one a three by three square is the smallest measurable particle and for an erosion factor of two a five by five square is the smallest possible resulting size. The pixels must also be calibrated in the X and Y dimensions. For accuracy it is best if this is implemented in the calibration Vi, then the resulting computed area would be exact. The actual dimensions of the 640 by 480 pixel

Figure 19: The logical difference of consecutive images 0 and 1

image is 420 μm by 314 μm. Correcting for expansion of the original image in the Y dimension and the extraction of only 256 pixels of the 320 in the X dimension our actual per pixel dimension in the X dimension is 1.313 μm and 1.227 μm in the Y dimension. Our smallest particle size of three by three pixels square (3.94μm x 3.68μm) is 14.5 μm^2. Realistically, only those particles with an area of more than fifty pixels (80 μm^2) should be counted as real with any accuracy. For our measurements we have chosen to count all particles remaining after erosion and post filter the data based on size once it's been stored to a file.

3 The AutoAnalysis Vi Overview and Future Work

Prior to starting the Cloudscope AutoAnalysis virtual Instrument, some front panel user inputs are required. In particular the timing method and VCR type used when the video was recorded. The choices for VCR type are programmable Sony Hi8 (default), any Hi8, or any VCR and those for timing method include actual time, time code (default), or manual timing mode. The time at which to start grabbing frames and the number of frames desired up to a maximum of 128 must also be input. In addition, the high and low video setting necessary to get acceptable contrast in the captured image. Finally enter the output file name and any file header information required

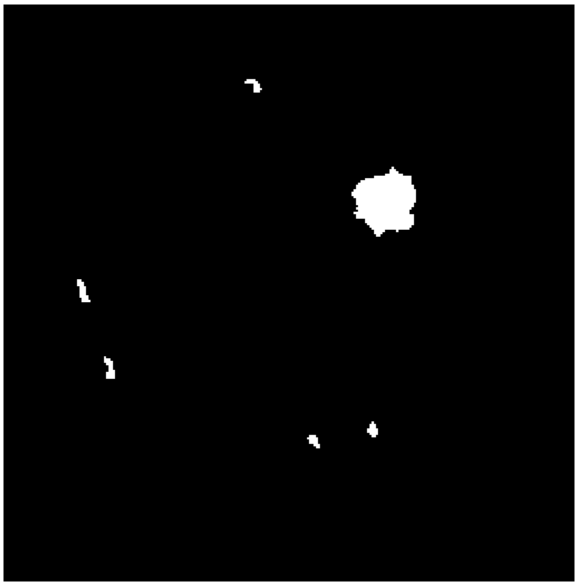

Figure 20: After erosion of the logical difference result.

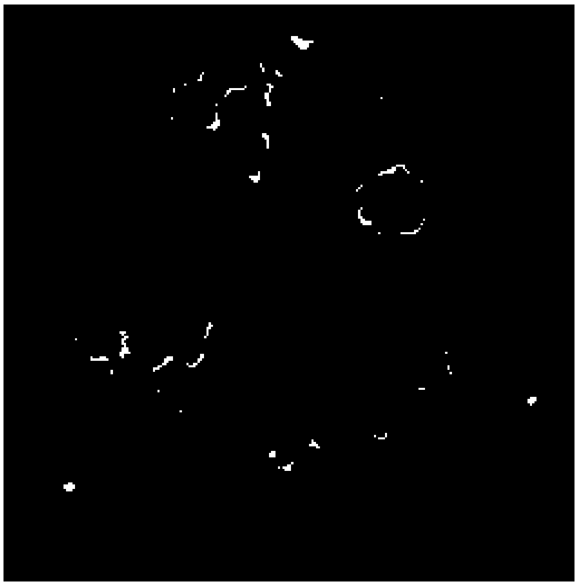

Figure 21: The logical difference of consecutive images 1 and 2

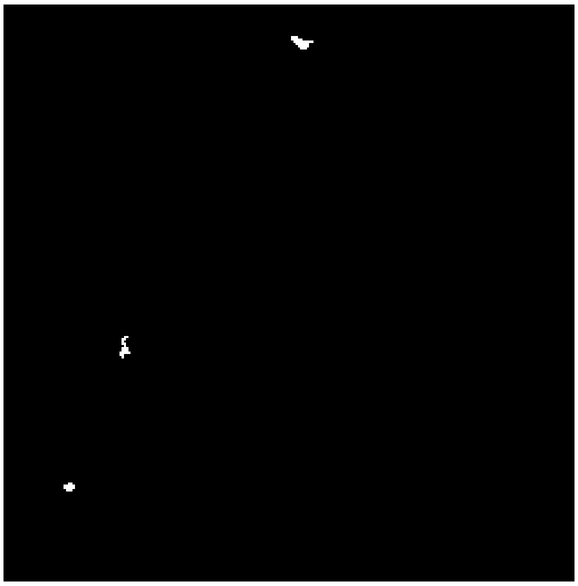

Figure 22: After erosion of the logical difference result.

to identify the data to be stored. The last item is very important for record keeping and should include the flight date, project, and time of the first frame grabbed.

The flow of the Cloudscope autoanalysis instrument starts with the initialization and creation of data structures and objects. These include arrays for storing current and prior image data along with the final array which holds new particle data until it is written to a file. The complex and gray scale image space is also allocated at this time. With the image space available the frequency filter mask is then generated by pasting a sequence of white rectangles onto a black image background. The rectangle locations identify the areas to be masked out of the frequency spectrum. This mask image is then converted to a complex image which is saved for repeated use in the remove noise phase of processing. If a programmable VCR is used, it positions itself at the defined start time and grabs the specified number of frames. At this point the user can view the images and if satisfied begin processing by pressing the start process button. The pause and exit buttons are available should the user wish to pause or terminate the processing. If the processing is allowed to run to completion the new particle results will be saved to the file specified by the user.

In creating this complex virtual instrument several modules were designed and implemented. The primary modules provided a Remove Noise function, which processed the image in the frequency domain using the filter mask to remove the corrupting interference from the original image. This processing step allows the particles to be more accurately detected and counted. Next the Alignment operation which performs a manual two dimensional correlation on two consecutive images by comparing and matching up similar particles within user defined constraints. This function corrects for jitter present from frame to frame, minimizing erroneous particle detection and counting. The last and most important of the processing routines is the Detecting and Counting of new particles. This last operations detects and counts those particles newly occuring from frame to frame. Once a complete set of frames has been processed the results are recorded in the output data file. In order to fully implement these routines several underlying functions were created to aide in their successful operation.

3.1 Computing the Ice Crystal Sublimation Rate

One of the areas of future work is to compute the ice crystal sublimation rate. The sublimation rate is of interest since it can be used in subsequent studies as a means of computing the particle's mass. To compute a sublimation rate, a crystal must be identified and its data parameters recorded for each frame. A reduction in particle size is expected from frame to frame. Once the particle has been tracked for a specified period of time, or once it has lost a certain amount of area, the difference in area from the first frame to the last frame will be recorded along with the number of elapsed frames. With this information we can compute a reasonable sublimation rate.

3.2 Simplification of the Alignment Procedure

Other areas for future improvements to our processing include the following. Implementing a more direct and accurate approach to correlating two images by adding an alignment target to the viewing lens. Increasing the accuracy of particle sizing by designing a means of closing particles. Many particles detected have concave cutouts. This is typically not the nature of an ice crystal. In most cases this cutout is caused by the particle overlapping a previously counted particle, resulting in the common area being removed. This may be a good application for exercising an object recognition algorithm or, in atmospheric science terms, "crystal habit" recognition. The same technique may also be a means of discerning and recreating two partially overlapping particles.

3.3 More Automation and Refinement

Minimizing user interaction by adding more automation can be improved upon in the following areas: automatically adjusting the video High and Low settings of the frame grabber by comparing the color range in the histogram and adjusting the settings accordingly, then repeating the process in an iterative fashion until a satisfactory range is achieved; setting the image threshold ranges based on the histogram peak and its spread; storing new crystal images using their bounding box coordinates; and lastly, allowing the user to create a frequency filter mask by simply drawing on the spectrum image.

4 Results and Conclusion

Here is a sample of the data file recorded after processing ten frames of data:

```
Data file for SUCCESS Flight on Date 05/03/96.  Time Code:  1:56:11:0
The left column is the frame number relative to start time.  To the right
of the frame number is the number of new particles found in that frame.  The
data, listed from left to right, includes: Total Area, Max intercept, Mean
intercept, Perimeter, Center of Mass X coord, Center of Mass Y coord, Bbox
upperleft X, Bbox upper left Y, Bbox lower right X, and Bbox lower right Y.
-----------------------
RelFrame# ParticleCount                 Time Code:  1:56:11:0
-----------------------
  Area   MaxInt  MeanInt  Perim  CofM-X  CofM-Y  Up-X   Left-Y  Low-X   Right-Y
-------------------------------------------------------------------------------
55     0
56     4
  32.00   12.20    2.54   26.66  141.97    2.39  137.00    1.00  149.00    5.00
  19.00    6.00    3.17   13.90  190.84    3.74  189.00    1.00  194.00    7.00
  33.00   13.73    2.40   28.73   36.00  124.24   33.00  119.00   39.00  132.00
  23.00    8.00    2.88   21.57   51.43  151.39   48.00  149.00   56.00  154.00
57     2
  19.00    5.00    3.80   14.24   87.47   31.74   85.00   30.00   90.00   35.00
  12.00    4.00    3.00   10.24   73.08   97.67   72.00   96.00   76.00  100.00
58     3
  16.00    7.52    2.13   15.07  100.88   60.81   98.00   59.00  105.00   63.00
  50.00   21.77    2.30   49.28   35.78  120.98   33.00  111.00   40.00  132.00
  29.00    8.42    3.44   19.90  233.76  164.14  231.00  161.00  238.00  168.00
59     6
  29.00    9.45    3.07   23.06  112.00   35.24  108.00   32.00  116.00   40.00
 606.00   32.00   18.94  117.75  171.31   87.99  156.00   72.00  185.00  104.00
  16.00    6.00    2.67   13.66    4.00  116.94    2.00  115.00    8.00  119.00
  45.00   12.43    3.62   31.31   47.00  161.04   44.00  156.00   50.00  168.00
  18.00    5.00    3.60   13.07  164.89  189.72  163.00  188.00  168.00  193.00
  43.00   10.66    4.03   26.44  138.77  193.44  135.00  189.00  143.00  199.00
60     2
```

17.00	5.00	3.40	13.66	88.24	34.82	86.00	33.00	91.00	38.00
34.00	11.23	3.03	28.14	161.59	95.00	159.00	90.00	167.00	100.00
61	3								
42.00	13.99	3.00	36.36	155.24	71.50	152.00	66.00	160.00	79.00
57.00	21.55	2.64	47.04	35.09	120.84	31.00	112.00	40.00	132.00
20.00	10.42	1.92	21.90	60.60	161.40	59.00	158.00	64.00	168.00
62	1								
27.00	6.00	4.50	21.19	227.52	238.26	225.00	236.00	231.00	242.00
63	5								
27.00	8.34	3.24	19.07	206.33	129.85	203.00	128.00	211.00	133.00
15.00	5.00	3.00	12.24	81.20	130.40	79.00	129.00	84.00	133.00
33.00	11.80	2.80	24.91	5.94	135.15	2.00	131.00	12.00	139.00
38.00	11.00	3.45	24.31	48.26	160.34	46.00	155.00	51.00	166.00
28.00	7.00	4.00	17.72	166.61	186.57	163.00	184.00	170.00	190.00
64	3								
14.00	5.00	2.80	11.66	164.71	29.14	163.00	28.00	168.00	32.00
17.00	6.00	2.83	13.66	5.47	105.00	3.00	103.00	9.00	107.00
11.00	4.00	2.75	9.66	234.82	162.82	233.00	161.00	237.00	165.00

The time to process the ten frames of data is approximately 6.5 minutes, and to process a maximum of 128 frames (4 seconds of flight) is 87 minutes which includes the time to initialize and and grab the frames. This time will increase in future runs when we use the whole 320 by 240 image expanded to 512 by 256 or a 640 by 480 contracted and expanded to 512 by 512. Ideally, we would have preferred a faster process time, but due to the compute time necessary for the FFT and IFT, this was not possible. However, considering the alternative of a manual count, which for a dense particle concentration would be extremely tedious and time consuming, time is not the main issue.

There are some physical properties of the CloudScope data that we may never be able to handle. One example of this would be: a particle lands on its edge in one frame then falls over in the next frame. Even to the human eye this appears to be two particles and as expected our program will count it as two particle. Crystals overlapping either partially or completely is another uncontrollable occurrence. With our current program version this will be counted as one large particle if they both landed at the same time. If on the other hand, as discussed in the previous section, they touched down on two different frames the second particle may be detected as a much smaller particle, as two separate smaller particles, or as a particle with a whole or cutout in it.

Repeatability and accuracy of our program may be determined by that of the programmable VCR. A delay of 4 seconds was built in to handle the slow turn on between the play command and the capture command. We found that given a start time the VCR came within approximately two frames of the selected time. However, every once in awhile, it seemed to ignore the four second delay and started capturing immediately. This may or may not be correctable and is still being investigated. We anticipate that a series of 128-frame sets can be processed one right after the other requiring an overlap of only one to two frames on either end, and would produce minimal duplicate data and missed frames. To check for duplicate data a comparison could be made of the overlapping frames and any repeats could be discarded. For a specified frame set and identical process settings, the automated processing was found to be highly repeatable and I wish, that I could say, completely error free. This, however, is not the case. Some particles are detected that are not crystals, but simply dark areas on the image, others remnants of larger particles are counted as new small particle. The smallest particle size we can detect after eroding by a factor of one is 9 pixels in area however, those particles less than say 50 pixels can not be counted as real with much confidence. Improvement to the image preprocessing, using higher quality video, and applying the enhanced automation techniques discussed in the previous section could reduce this number or at least increase the accuracy of detecting actual crystals versus particle remnants or shaded areas. Being able to detect and size very small ice crystals is of great interest to the atmospheric scientists interested in this analysis tool. We forsee this as an on going project of analysis enhancements and improved capababilities.

References

[1] P. Arnott, Y. Dong, R. Purcell, and J. Hallet. Direct airborne sampling of small ice crystals and the concentration and phase of haze particles. In *Ninth Symposium on Meteorological Oberservations and Instrumentation*, pages 415–420, 1995.

[2] R. Gonzalez and R. Woods. *Digital Image Processing.* Addison-Wesley Publishing Company, 1992.

[3] Joseph W. Goodman. *Introduction to Fourier Optics.* McGraw-Hill, Inc, 1968.

[4] John C. Russ. *The Image Processing Handbook.* CRC Press, 1995.

[5] Mark Turner. Digital image analysis of replicator ice crystal data for detection of small particles in cirrus clouds. Master's thesis, Atmospheric Physics, University of Nevada, Reno, May 1996.