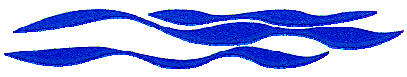
A SYSTEM FOR VIDEO OBSERVATION OF NEARSHORE PROCESSES

BY

TODD J. DEMUNDA AND JAMES T. KIRBY

RESEARCH REPORT NO. CACR-06-07
FALL 2006

This study was supported by the Delaware Sea Grant Program
Project Number NOAA SG0507 R/ETE-5
Award Number NA05OAR4171041



CENTER FOR APPLIED COASTAL RESEARCH

Ocean Engineering Laboratory
University of Delaware
Newark, Delaware 19716

ACKNOWLEDGEMENT

This project was funded by the Delaware Sea Grant Program, Project Number NOAA SG0507 R/ETE-5, Award Number NA05OAR4171041 and by the Delaware Department of Natural Resources and Environmental Control (DNREC) who purchased the cameras. Also, the Sea Colony Hotel, Bethany Beach, DE who allowing us to install cameras on their roof.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

A video camera system has been deployed in Bethany Beach, DE for the purpose of observing nearshore processes. The system captures several types of still images as well as video streams at regular intervals. The imagery can be used in the scientific analysis of coastal dynamics such as the prediction of rip currents or estimation of wave speed.

Multiple video streams are rectified and velocity fields are estimated using MatPIV, a MATLAB implementation of particle image velocimetry. The maximum velocity calculated at each point is assumed to be the local wave phase speed, and this value is used to calculate the local depth according to long wave theory. This bathymetry is compared to tide-corrected survey bathymetry to determine the accuracy of the depth-inversion.

The depth inversion appears to produce the best results when and where the incoming waves are either clearly visible and long-crested, or hardly visible at all. Uniform lighting appears to be ideal. In these regions, the calculated beach slope and depth match up with the survey bathymetry very well. Offshore, the inversion begins to fail possibly due to the diminishing visibility of the waves. Breaking waves and swash zone velocities produce false deep zones in the calculated bathymetry. Glare from the sun washes out the surface and movement cannot be accurately detected. Sunshine or clouds seem to not matter, as long as the sea surface is smooth and uniform.

Chapter 1

INTRODUCTION

1.1 Quantitative Analysis of Coastal Imagery

The utilization of imagery and video can be used for quantitative analysis of multiple coastal processes. Snapshots of the time averaged shoreline, which clearly show the mean border between surf zone and beach face, can be examined over time to track shoreline migration and beach face erosion or accretion. These images can also emphasize the occurence of rip currents and rip channels. Variance images of the area of study highlight zones of wave breaking, and can be used to track sandbar development and movement.

1.2 Bathymetric Surveying

Conventional methods of acquiring bathymetric data on the beach face and in the surf zone are both cost and time-consuming. On land and in wading-depth water, surveying is typically performed using the traditional rod and transit method. Further offshore, a combination of DGPS and a fathometer on a boat is used. The resulting bathymetry data may be sufficiently dense within each cross-shore line, but these lines are usually sparse in the alongshore direction as well as in time, owing to the aforementioned high monetary and time costs. Adverse wave and weather conditions also limit the performance of surveys.

Utilizing wave propagation to determine bathymetry has been studied as early as World War II, when the US military attempted to use wave phase speed

to estimate local bathymetries in order to plan safe landings for military craft (Williams, 1947). Later, water depth was inferred from wave speed calculated from time-lapse aerial photography (Johnson, 1949), (Fuchs, 1953). More recently, a technique for bathymetry estimation was developed using imagery from a fixed onshore video system (Stockdon and Holman, 2000). This method uses EOF analysis and linear wave theory to estimate a cross-shore depth profile from a time stack of pixel intensities in a single cross-shore line.

1.3 Depth Inversion using PIV

Presented in this paper is a technique for depth inversion in two dimensions. Instead of the single cross-shore line analyzed by Stockdon and Holman, a full 2D velocity field is estimated from remotely collected video streams using particle image velocimetry. From this, water depth at each velocity grid point is calculated using long wave theory. Once the method is refined, video streams can be regularly analyzed and a record of morphology changes over time at the video system location can be acquired. This data can be used to test numerical models of nearshore processes.

Chapter 2 provides an overview of the video system installation and setup work involved before the depth inversion analysis can proceed. A detailed description of these steps is found in Appendices A and B. Chapter 3 discusses the work involved for the analysis of each video file, from image rectification to depth inversion. Depth inversion results are displayed and discussed in Chapter 4. Chapter 5 contains a brief summary of the conclusions drawn from this study.

Chapter 2

OVERVIEW

Here, the system installation and setup will be briefly described. For the full details, see Appendices A and B.

2.1 Video Observation System Deployment

The location chosen for study is the coastal zone of Bethany Beach, DE. In order to get a comprehensive view of the shoreline and surf zone, an elevated platform is needed to mount the video cameras. In Bethany Beach, the highest buildings are located in the Sea Colony resort complex. The specific building for this particular deployment is the Brandywine Building, whose roof apex is approximately 90ft above ground and has a clear view offshore, north, and south (Figure 2.1). Adjacent to the rooftop is the control room for the building's elevators, where the system computers are situated.

The video monitoring system was designed by Erdman Video Systems. It consists of five cameras contained in weatherproof housings. The five cameras are arranged to achieve a complete panoramic view from the northern to southern limits of view from the rooftop. Focal lengths for the lenses on each camera vary from 4.5mm for the camera looking straight offshore to 8.5mm looking north and south to the limits of view. The cameras are equipped with windshield wipers and polarizing filters. They are mounted on an aluminium/plywood frame fastened to the roof (Figure **??**. Cameras are numbered from 1 to 5 in order of view from northernmost to southernmost.

Figure 2.1: An aerial view of the Sea Colony resort in Bethany Beach, DE, with northern end of the Brandywine Building encircled. Image courtesy of Connor Jacobsen Realty, Inc.

The wipers and cameras are controlled by two identical wall-mounted computer systems, with management divided three on one computer and two on the other. Erdman's VM95 software controls all aspects of the system. Each camera captures snapshots, time exposures, and variance images every hour during daylight. These are saved in the ARGUS filename format and ftp'ed to a server at the University of Delaware for archiving and web display. In addition to these still images, the camera looking straight down at the shoreline captures video streams every hour as well. These will be later used in the depth inversion analysis. Once configured, the system runs independently and has several features that help avoid accidental shutdowns or loss of automation.

Figure 2.2: The video observation system, post-installation.

2.2 Setup Work

Before deployment, camera lens distortion parameters must be determined to correct for lens imperfections before analyzing the imagery. A routine developed by the ARGUS group facilitates this process. See Appendix B.1 for the details of the lens calibration

For image rectification to be made possible, a survey of ground control points must be taken. These GCPs are markers visible in each camera's view. Makrers were deployed on the beach, and their coordinates were surveyed by the Delaware DNREC survey team. The points were surveyed in Delaware state plane coordinates. At the same time, snapshots of the camera views with visible GCPs are taken and saved. Then, we can find the pixel coordinates of the GCPs in each view and their real world coordinates are known from the survey. More MATLAB routines developed by ARGUS are able to use this pixel to real world correspondence to calculate the azimuth, tilt, and roll of the cameras. The local coordinate system is simply the state plane coordinate system recentered with the origin at point B-1 of the survey. This point is on the beach/dune border and about in the alongshore center of the view of camera 3. With the geometry and distortion parameters in hand, we have all the information we need to begin analysis on the video streams. Table 2.1 shows the distortion parameters for each lens. Table 2.2 shows the surveyed GCPs and other points in state plane coordinates. Table 2.3 shows the system geometries in local coordinates..

	D1	D2	U_0 (pix)	V_0 (pix)
Camera 1	-0.0000003246	0.0233	322.5	-241.12
Camera 2	-0.00000054239	0.0402	319.74	-239.66
Camera 3	-0.000001806	0.1239	238.22	-155.72
Camera 4	-0.00000049568	0.036	320.57	-235.22
Camera 5	-0.00000031187	0.0223	321.8	-238.68

Table 2.1: Lens distortion parameters for the five cameras.

Point Number	Northing (ft)	Easting (ft)	Elevation (ft)	Description
01	193141.368000	759849.238000	015.610000	DECK SCREW 1
02	193524.015000	759789.785000	016.600000	BL S 5+00 TOP AL
03	195610.647993	759881.038692	016.051723	BW-1
04	194787.835725	759877.987104	009.763306	P-1
05	194640.902646	759906.161688	009.099310	P-2
06	194235.196919	759874.013974	009.920817	P-3
07	194041.979396	759862.932781	011.052289	P-4
08	193927.655247	759887.240076	010.873093	P-5
09	193841.938758	759877.686228	011.383488	P-6
10	193772.730333	759836.782192	013.960411	SS-1
11	193756.544915	759835.714057	014.084464	SS-2
12	193766.400263	759889.858591	011.118251	P-7
13	193728.459924	759866.623283	010.759601	P-8
14	193693.103600	759885.901000	011.120000	P-9 ADJUSTED -1’
15	193706.751081	759836.424777	011.762327	F-1
16	193704.864336	759842.438609	010.866143	F-2
17	193644.475704	759881.359188	013.110449	L-1
18	193606.156249	759880.247809	009.947640	P-10
19	193594.095910	759845.015443	011.700209	B-1
20	193568.916998	759845.861811	011.499472	B-2
21	193543.353777	759865.258957	010.868759	P-11
22	193489.478246	759877.044568	011.182099	P-12
23	193490.000482	759901.965850	011.427927	P-13
24	193410.861714	759896.590306	011.385671	P-14
25	193349.455474	759859.552953	014.461116	SS-3
26	193333.752346	759863.582683	013.878686	SS-4
27	193307.849380	759900.972765	013.316676	L-2
28	193216.492415	759888.253333	010.560673	P-15
29	193221.743641	759918.862411	010.796727	P-16
30	193030.843182	759924.777292	010.884751	P-17
31	192958.015979	759933.746936	013.496879	L-3
32	192625.386819	759975.958582	014.468111	L-4
33	192486.538052	759975.778822	009.498442	P-18
34	192237.920400	759984.791000	010.090000	P-19
35	193558.155841	759682.514225	108.588817	C-5
36	193559.143599	759683.316183	108.619242	C-4
37	193566.162179	759688.842239	108.613779	C-3
38	193566.805093	759687.998261	108.752835	C-2
39	193567.508812	759687.211538	108.910373	C-1
40	193125.461000	759844.330000	015.830000	CORNER OF POOL
41	193025.916000	759832.252000	018.540000	BL S10+00 TOP AL
42	192527.647000	759874.560000	016.940000	BL S 15+00 HUB
43	192029.574000	759916.578000	016.780000	BL S20+00 METAL
44	191533.158000	759971.431000	020.150000	BL S25+00 PVC

Table 2.2: Table of surveyed GCPs and other static objects.

	Camera 1	Camera 2	Camera 3	Camera 4	Camera 5
X Pos. (m)	-48.099	-47.859	-47.602	-49.287	-49.531
Y Pos. (m)	-8.1038	-8.3183	-8.5143	-10.654	-10.955
Elevation (m)	33.196	33.148	33.106	33.108	33.098
Azimuth (rad)	0.3306	0.5756	1.3778	2.1447	2.7827
Tilt (rad)	1.3468	1.422	1.3715	1.3332	1.4671
Roll (rad)	-0.0487	-0.0242	-0.0067	0.0033	-0.0091
Field of View (rad)	0.7298	0.9723	1.2931	0.9763	0.7253

Table 2.3: Table of calculated geometry values in local coordinates.

Chapter 3

DEPTH INVERSION THROUGH PIV ANALYSIS

With the hardware in place and operational, and the necessary preparatory steps completed, the incoming video data from the system is ready for analysis. This section details a technique for estimating the bathymetry of the area seen in the camera view using PIV analysis and long wave theory.

3.1 Readying video files for MATLAB

After the distortion and geometry data for each camera is properly prepared, a rectification program is used to transform the camera images into planar images of the beach face and surf zone.

Analyzing a series of snapshots over a long period of time could give indications of shoreline migration and beach erosion. However, they cannot tell us much about fast-scale motion such as rip currents and waves. To examine these processes, video streams from camera 3 will be analyzed. Most of these videos are short, 5-second AVIs captured at 5 frames per second (fps). However, the system initially recorded 47-second, 10fps WMV videos, and it is these videos that will be examined closely in this paper. The analysis is performed on a typical 32-bit Linux machine. In the future, when analysis is performed exclusively on the 64-bit SANDCAM computer, the AVIs will be used since there are no 64-bit WMV codecs in existence.

WMV files are converted into a series of JPEG images using the public domain program Mplayer (www.mplayerhq.hu) The following command issued from the linux prompt performs the conversion:

mplayer -vo jpeg desired_movie.wmv

This results in each frame of the movie being extracted, encoded into a jpeg image, and named in sequential numerical order. For the videos in question, which are 47s long at 10fps, the frame rate is much higher than necessary and results in too many frames to be analyzed. After conversion, unwanted frames can be manually deleted to adjust the time difference between frames. Here, every 7th frame is kept, for a framerate of 0.70s. Since the images are named in sequential order, skipping over unwanted frames won't affect their order. To stay organized, it's a good idea to put sequences of images in separate folders according to the source video. Now we have a sequence of images ready for processing.

3.2 Image Rectification

It's time to transform the sequence of images into rectified grayscale images. In this format, they will be ready for scientific analysis, having been transformed into images with known dimensions and resolution.

The following programs are modified versions of programs originally written by Nat Pearre of the University of Delaware Center for Applied Coastal Research. The program RECTIFY_AND_STACK is the primary routine that oversees the rectification process and calls the various subroutines. The first step is to load the sequence of images into MATLAB. This is handled by the subroutine GETIMAGES2. Here, the user specifies the full directory path to the images that are to be analyzed as well as their filename extension. Then, the program creates a directory listing of all the files in that folder with that extension. If organized correctly, this results in a structure array with information about each image. Its length is the same as the number of frames in the folder, and they are in sequential order thanks to the

automatic naming convention that mplayer uses when converting from video. Then, in sequential order, the program concatenates the directory path with the image name and reads the image into a new structure array. This results in a structure with the length of the number of images, and each entry having the RGB image information.

Next, the main routine calls GETGCP2, GETALLCAMDATA, and GETALL-GEOMETRYDATA. GETGCP2 is the same m-file described in the system geometry section of the appendix. The other two files are almost identical to their similarly-named counterparts in that section, except they load information for all five cameras at once instead of one-at-a-time. In GETALLCAMDATA, the lens centers and correspondence matrices must be scaled to account for the difference in resolution between the still images and the video captures, which are 640x480 pixels and 480x360 pixels, respectively. This is accomplished simply by multiplying the width and height each by 3/4. After loading all the camera and geometry information, an index number is set to determine which of the five cameras is being worked with. Here, again, we are using video from camera 3. A function called GEOMETRY-FUNC4 is then called, which calculates the correct geometry values for the camera. It is exactly the same as the geometry function in the last chapter, except it does not plot the GCPs it calculates.

Now, a grid must be created onto which the video frames will be rectified. The grid bounds and resolution can be adjusted by trial-and-error to get the desired rectified image. The bounds correspond to the coordinate system created in the previous chapter and partially detailed in Table B.1. Remember that the local origin for the new coordinate system is about halfway across the view of camera 3 and right on the dune/beach interface. The rectification grid chosen for this view is 25m to 150m in the cross-shore and -100m to 100m in the alongshore, with a grid spacing of 0.5m. This will yield an image of 251x401 pixels.

After grid definition, several intermediate steps make a mesh of the grid as well as 1D vectors of every point. A 'zero elevation' grid is also made. The new grid and all of the other information loaded is then fed into a program called PLANFORMMAKER2, written by Jack Puleo, also of the UD CACR. It is this program that does the real legwork of the rectification process.

The technique used in PLANFORMMAKER2 is described in detail in 'Practical Use of Video Imagery in Nearshore Oceanographic Field Studies' (Holland et al, 1997). Essentially, the routine takes the z=0 plane previously defined, maps it onto the image plane, then distorts it. The result of these steps is a rectified RGB image, which in MATLAB is stored as an MxNx3 matrix, where M and N are the image dimensions. The three layers are the red, green, and blue intensities. In PIV analysis, only raw intensity is of any interest, so the image must be converted to grayscale. The conversion is performed using the MATLAB-defined technique of calculating the monochrome luminance, a simple equation which combines the RGB values according to the eye's sensitivity to red, green, and blue. Equation 3.1 is used for this conversion, where I is the grayscale intensity image and C is the original MxNx3 RGB image matrix.

$$I = 0.2989 * C(:,:,1) + 0.5870 * C(:,:,2) \ + 0.1140 * C(:,:,3) \tag{3.1}$$

This grayscale image is returned to the main engine RECTIFY_AND_STACK. This entire routine is looped such that the source images are processed in sequential order and the rectified images are stored in a 3D stack. When finished, this image stack is saved as a .mat file in the directory of the PIV program.

Figures 3.1 and 3.2 show an example video frame before and after rectification. The view field of the camera is obstructed by the wiper and polarizing filter, but these do not cover up the area of interest so rectification is unaffected. Note that in the rectified image, north is downward.

Figure 3.1: Example original video frame.

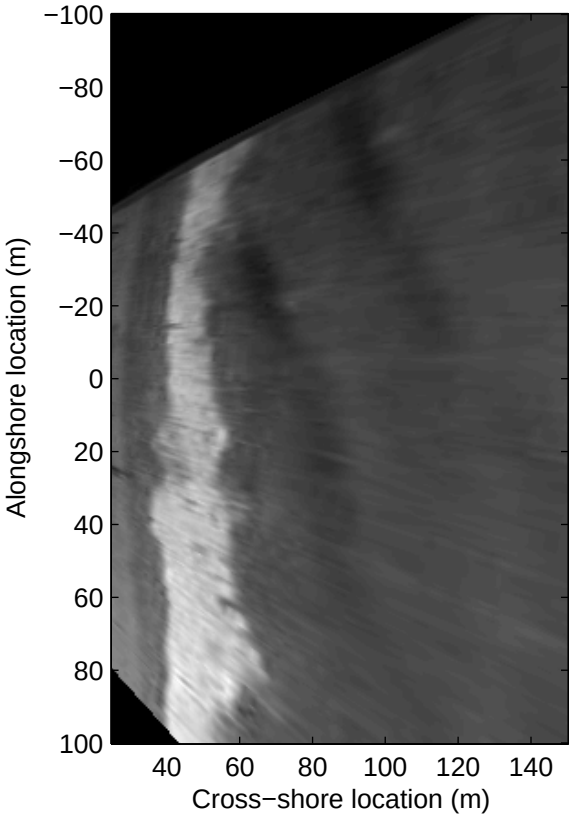


Figure 3.2: The video frame from Figure 3.1, rectified and converted to grayscale.

3.3 PIV analysis with MatPIV

From the previous step, we now have a series of images with known coordinates and time steps. In the videos, waves can be seen propagating towards the shoreline. In the rectified images, the wave crests are clearly seen as darker patches on the lighter ocean surface. Using particle image velocimetry (PIV), an attempt is made to track the wave crests and determine their velocity. If this is successful, the water depth in which they are traveling can also be determined.

PIV analysis is a nonintrusive technique used to estimate two and three-dimensional flow fields by comparing successive images of the field in question. Images are divided into smaller regions called interrogation windows, their size depending on the expected flow velocities. Corresponding successive windows are cross-correlated to determine the correlation between them at every possible overlap. The highest correlation occurs when the overlapping windows are the 'least unlike'; this yields the likely displacement of the window and thus the velocity of that area. Digital particle image velocimetry was first explored by Willert and Gharib (1990).

The package used in this study is called MatPIV, a MATLAB implementation of PIV developed by Sveen (2006). It utilizes MATLAB's cross-correlation functions as well as FFTs.

A program called CALL_MATPIV is used to oversee the PIV processing. Aside from the rectified images, MatPIV needs several other parameters as input. One is the interrogation window size. This is the size of the windows that MatPIV breaks the full image down into, and it is these smaller images that are cross-correlated to estimate velocity. Another parameter is the interrogation window overlap, which is exactly what it sounds like. These numbers can vary and different values will change the results of the analysis. Too small and the window doesn't see the wave crest. Too large and it sees too much. After experimentation and comparing animations of the resulting vector field along with the rectified images,

the best values are determined to be 48 pixels and a 50% window overlap. Also assigned is the image time step of 0.70s and the desired analysis method. For speed and simplicity, the single pass method is used, which simply makes one analysis attempt at the specified window size rather than starting there and then making multiple passes with successively smaller windows.

The images are fed into the MatPIV main routine, which outputs pixel coordinates of the velocity grid and the calculated horizontal and vertical velocities in pixels/sec. Erroneous points are interpolated and a mask is built to zero out the useless black areas in the image. The velocities are then smoothed using filters in MATLAB's image processing toolbox. The final result is a pair of vector fields representing the estimated motion between each pair of successive images. The following images in Figures 3.3 to 3.5 are from analysis of a video captured on June 27, 2006 at 21:36 GMT.

The waves in these views are approaching the shore from the upper left. The instantaneous velocity field appears to capture the motion quite well, with the shoreward propagation of the waves seen up to the breaker line, then a shore-parallel motion in the swash consistent with the oblique wave incidence and the resulting longshore current. The averaged field supports this and shows these motions more clearly. In the lower left, there is a region of vectors that have an alongshore direction opposite that of the rest of the field. This type of feature could results from the three-dimensionality of the wave field.

The maximum velocity is simply the maximum velocity out of all the image pairs at each grid point. It appears to capture the overall direction of wave propagation. This means that MatPIV is most likely seeing the wave crest movement as the dominant motion in the field.

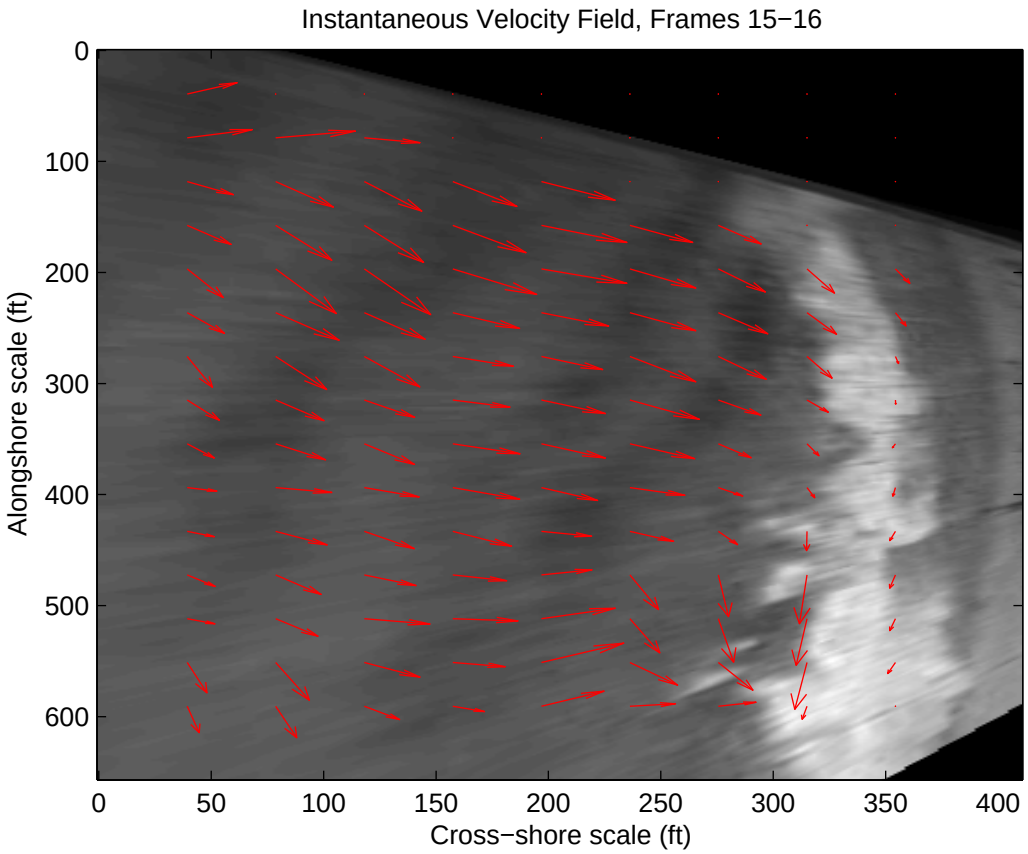


Figure 3.3: Instantaneous velocity field from PIV analysis.

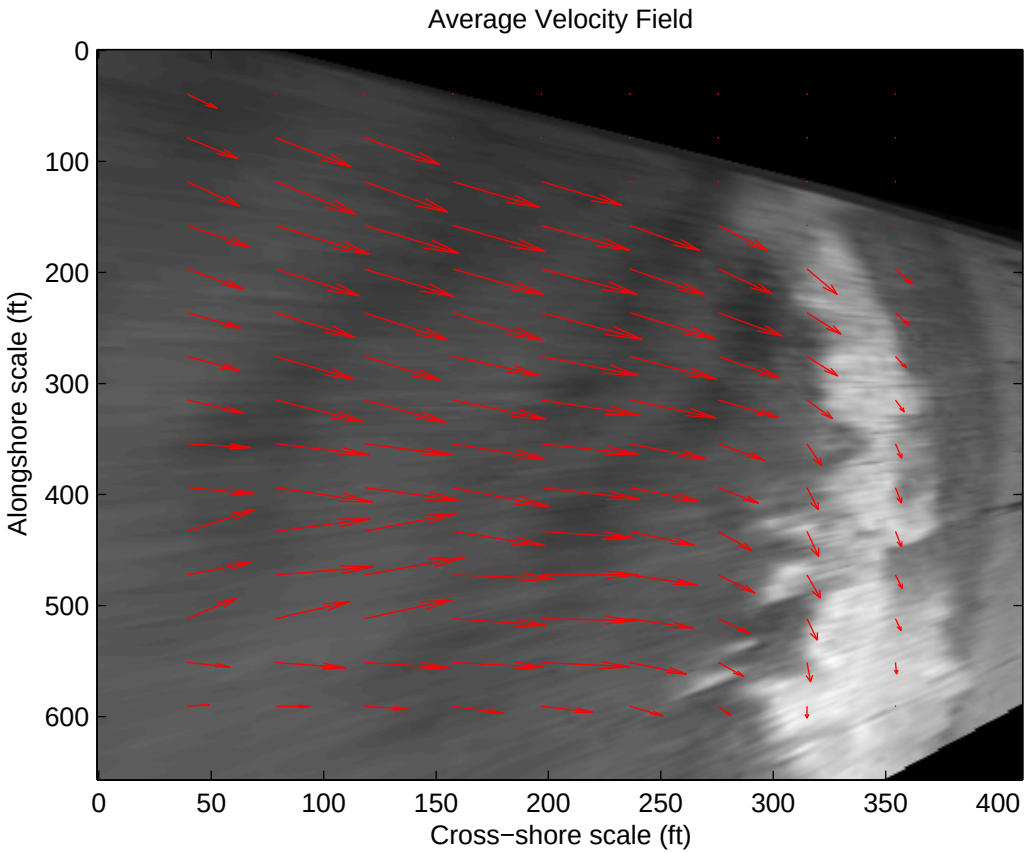


Figure 3.4: Average velocity field from PIV analysis.

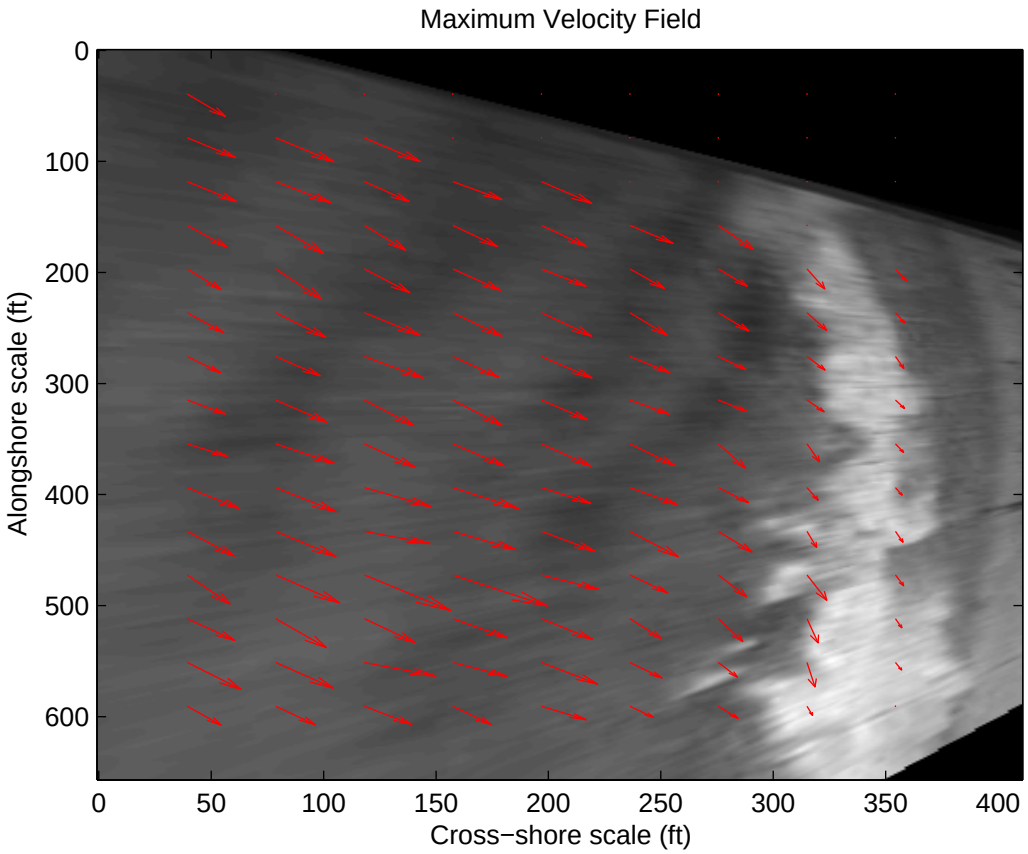


Figure 3.5: Maximum velocity field from PIV analysis.

3.4 Depth Inversion

It has been observed that the velocity vectors of MatPIV appear, from a qualitative perspective, to capture the propagation of wave crests in the imagery quite well. If the region is assumed to be in the 'shallow' domain of linear wave theory, then the wave speed is dependent only on the water depth in which it is traveling and is calculated by Equation 3.2.

$$c = \sqrt{gh} \tag{3.2}$$

Long wave theory is valid when $kh < \pi/10$, with k being the wavenumber and h being the local water depth. In the videos to be analyzed, wave period ranges from 5s to 15s. Surveyed water depth in the study area is as high as 15ft. Table 3.1 shows the wave period / depth combinations where $kh < \pi/10$, labeled as 'valid'. If $kh > \pi/10$, the table shows the value of kh. For the shorter period waves, a large portion of the bathymetry is considered not in the long wave domain. However, kh is still always less than 1, so assuming shallow water is not completely unreasonable.

If the velocity field estimated by MatPIV is in fact the speed of wave propagation in that area, then using Equation 3.2 it is simple to calculate the water depth at each velocity grid point. This results in a 2D bathymetric grid with a grid spacing of 39.4ft (12m), much higher than the alongshore resolution of the typical survey conducted by DNREC in the vicinity.

The velocity at each point varies in time, so first we must choose which velocity to use. In viewing animations of the vector field over the image sequence, as well as examining the above figures, it appears that the maximum velocity at any point occurs when the wave crest is passing through that point. The total velocity is calculated from the maximum u and v components simply by using Equation 3.3.

$$c = \sqrt{u_{max}^2 + v_{max}^2} \tag{3.3}$$

Depth (ft) / Period (s)	5	6	8	9	15
1	valid	valid	valid	valid	valid
2	0.31847	valid	valid	valid	valid
3	0.39333	0.32526	valid	valid	valid
4	0.45803	0.37777	valid	valid	valid
5	0.51649	0.42484	0.31459	valid	valid
6	0.57068	0.46814	0.34575	valid	valid
7	0.62178	0.50866	0.37467	0.33145	valid
8	0.67056	0.54703	0.40186	0.35525	valid
9	0.71756	0.58371	0.42765	0.37777	valid
10	0.76314	0.61901	0.45227	0.39924	valid
11	0.80762	0.65318	0.47593	0.41982	valid
12	0.8512	0.68641	0.49875	0.43964	valid
13	0.89408	0.71884	0.52085	0.45879	valid
14	0.93639	0.7506	0.54233	0.47736	valid
15	0.97827	0.78179	0.56326	0.49541	valid

Table 3.1: Validity of long wave theory for depths and wave periods encountered.

This is assumed to represent wave celerity. Since the image is of known dimensions, the velocities and coordinates can be converted from pixel to real world units by multiplying by the image resolution of 0.5m/pixel. Now that we have real-world velocities, Equation 3.2 is solved for h and the depth is calculated at each grid point. The grid is then converted into Delaware State Plane coordinates by shifting the coordinates according to the planform boundaries set in the rectification process and adding the coordinate of the local zero (GCP point B-1, 193,594ft north, 758,845ft east).

3.5 Preparation of Survey Bathymetry

To test this technique for depth inversion, the depth-inverted bathymetry must be compared to field data for ground-truthing. The field data set used in this study is from a December 2005 beach profile survey performed by the Delaware Department of Natural Resources and Environmental Control. In Bethany Beach,

the region of interest, it consists of 32 cross-shore profile lines with irregularly spaced points along each line starting at the dune system and terminating in roughly 40 feet of water offshore. The vertical datum used is the North American Vertical Datum of 1988 (NAVD 88). The raw data set comes as a series of irregularly spaced points with state plane northing/easting coordinates and their associated depth. In order to be able to make a comparison between this survey information and the depth-inverted bathymetry, these irregular points must be interpolated into a regularly-spaced grid to make a complete bathymetric map of the area of interest. The first step in constructing this bathymetric map is defining the northern, southern, eastern, and western bounds of the regular grid. These bounds are simply the maxima and minima of the northing and easting coordinates found in the data set. This will create a grid that encompasses the whole of the Bethany Beach shoreline. For comparison purposes, this can later be cropped to look at only the area within the camera's view field. Here, the northern and southern limits of the grid in Delaware State Plane coordinates are 200,100ft and 184,000ft, respectively; the eastern and western limits are 762,340ft and 759,130ft, respectively. Now that the grid limits are defined, a grid spacing in each direction must be assigned to determine the bathymetry grid resolution. With 32 grid lines in the alongshore, which encompasses 16,100ft of shoreline, the survey resolution alongshore is 500ft. There are 3093 total points in the survey, and each of the 32 lines have approximately the same number of points, so there are about 97 points in each cross-shore bin which encompasses 3210ft, for survey resolution in the cross-shore of about 33ft. It is immediately evident that this bathymetry is low-resolution and surely misses most of the finer structure of the beach face and bathymetry. At the time, however, it is the only field data set available for our use. In order not to lose resolution in the interpolation, we choose a grid spacing for the regular grid of 25ft in the alongshore and 10ft in the cross-shore. Bad survey points are removed by a simple line of code that looks for

depth values greater than the maximum 'realistic' value and removes that depth and its associated coordinate pair from the set. With the grid bounds and resolution in hand, and with erroneous points cropped out, we are ready to interpolate the survey bathymetry into the regularly-spaced grid. The MATLAB function griddata does this quickly and easily using nearest-neighbor interpolation. The function takes as input the desired regular grid as well as the coordinate pairs of the survey data and their associated depths, and outputs a grid of depth points at each of the regular grid points. Other interpolation schemes were investigated, but were difficult to use successfully without many NaN values appearing. The resulting bathymetry grid for the Sea Colony region is shown in Figure 3.6.

3.6 Comparing Survey Bathymetry to Calculated Bathymetry

Now, we are ready to compare the depth-inverted bathymetry to that of the DNREC survey and discover if MatPIV is accurately capturing the wave celerity and the depth inversion technique subsequently works. Both data sets are already in state plane coordinates, so they can be easily plotted concurrently. The depth-inverted data is cropped to leave out rows that have the black areas from rectification, and the survey bathymetry is also cropped down from all of Sea Colony to this local region. The calculated bathymetry is smoothed using the same filters that were used to smooth the velocity fields. Finally, the survey bathymetry is corrected for the tidal level at the time of the video capture. Tidal data is taken from the NOAA tidal station at Lewes, DE using the same NAVD 88 datum as the survey. Video streams spanning the period from June 11 to June 28, 2006 are analyzed.

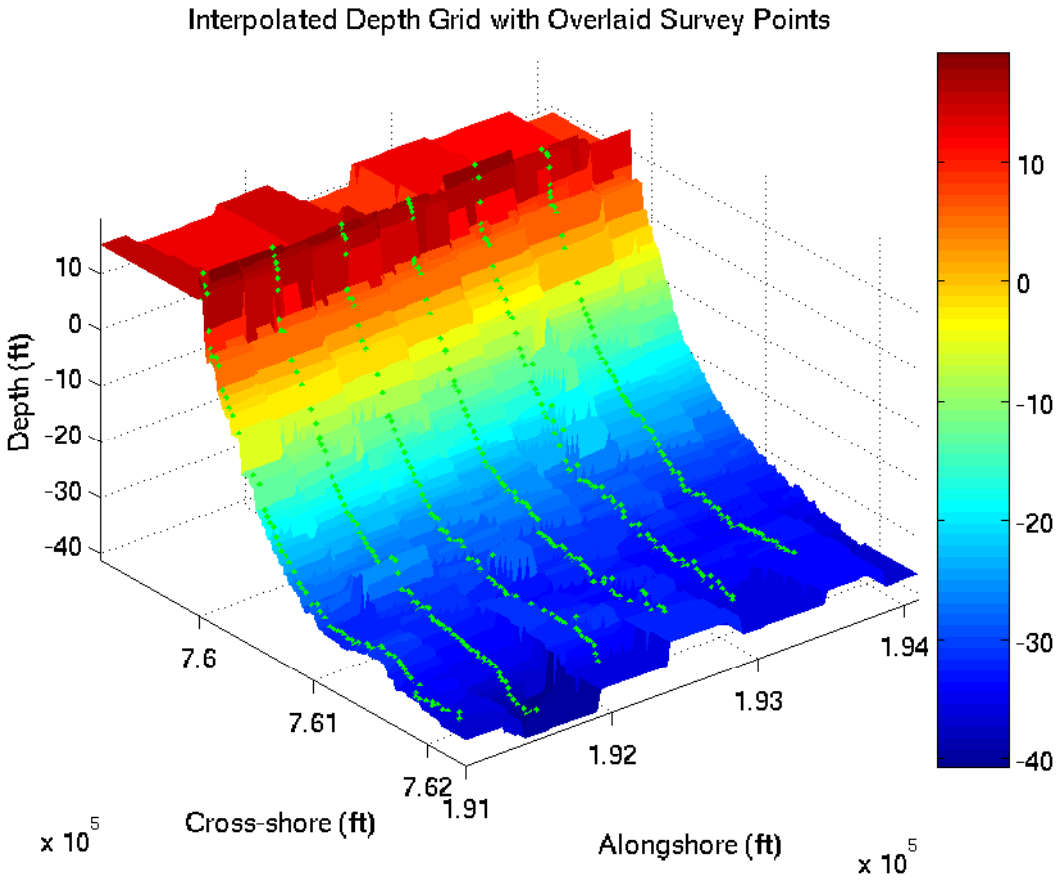


Figure 3.6: Bathymetry interpolated from Delaware state survey, with original survey points overlaid.

Chapter 4

RESULTS AND DISCUSSION

4.1 Preliminary Notes

Before attempting to analyze the results shown below, several issues must be clear. The survey in this area is particularly sparse, so the interpolated bathymetry is basically uniform in the alongshore. Since it is of such low resolution, minor features of the bathymetry may not be captured in the survey but could still show up in the depth inversion. This is not to say these anomalies are right or wrong, simply that there is no verification at this point.

Additionally, the tide correction is not exact, so the depth inversion could be accurately capturing the bathymetry but be off from the survey by a constant value. However, this difference should be very minor. Finally, the survey was performed in December 2005, a full 6 months prior to the video captures. Beaches are dynamic entities, and one shouldn't expect their profiles to remain static for long periods of time. With that being said, the results from each video stream will now be examined.

Four types of plots are shown: a 3d mesh of the two bathymetries, a comparison of the average cross-shore line for each, the average survey cross-shore line versus every depth-inverted line, and finally a single cross-shore line from each at 193,660ft (state plane) in the alongshore. This alongshore location corresponds to about the center of view of camera 3. For reference, the shoreline is located at about 760,000ft in the cross-shore. Also presented are frame grabs from each video to see what the sea surface and lighting conditions were like during the video capture.

4.2 Video from June 11, 2006

On this date, there is no data from the Army Corps directional wave gauge DE002, offshore of Bethany Beach. At NOAA buoy 44009, the waves have a significant height of about 0.75ft with a dominant period of about 5s. Wind direction, the closest approximation of wave direction, is 300°.

4.2.1 June 11, 2006, 11:35 GMT

The conditions for this video can be examined in Figure 4.1. Small, organized short-crested waves are seen propagating from the northeast. Lighting is clear and bright, but there is a large area of sun glare towards the northern end of the view. Local wind appears to be minimal. Overall, the bathymetry is underestimated as seen in Figure 4.2, with the region of sun glare being the shallowest. It seems that the glare maximizes the image intensity and no movement can be detected by PIV analysis.

4.2.2 June 11, 2006, 17:36 GMT

A sample video frame of this time is shown in Figure 4.6. In this video, lighting is bright and clear. There appears to be very little local wind. The sea surface is calm and smooth, and no wave crests can be seen until right at the shorebreak. Despite there being no visible wave propagation, the depth inversion is surprisingly successful. There is some overestimation of depth near the southern offshore boundary as seen in Figure 4.7, but many of the cross-shore lines follow the survey bathymetry very well in Figure 4.8. At the shoreward end, depth is overestimated, but this likely due to the PIV routine capturing the swash velocity, which is not governed by linear wave theory. The calculated depth begins to trend upward at the offshore boundary. This is possibly due to the angle at which the camera is viewing this movement, or it could be that the wave crests are harder to

Figure 4.1: Frame grab from video, June 11, 2006, 11:35 GMT.

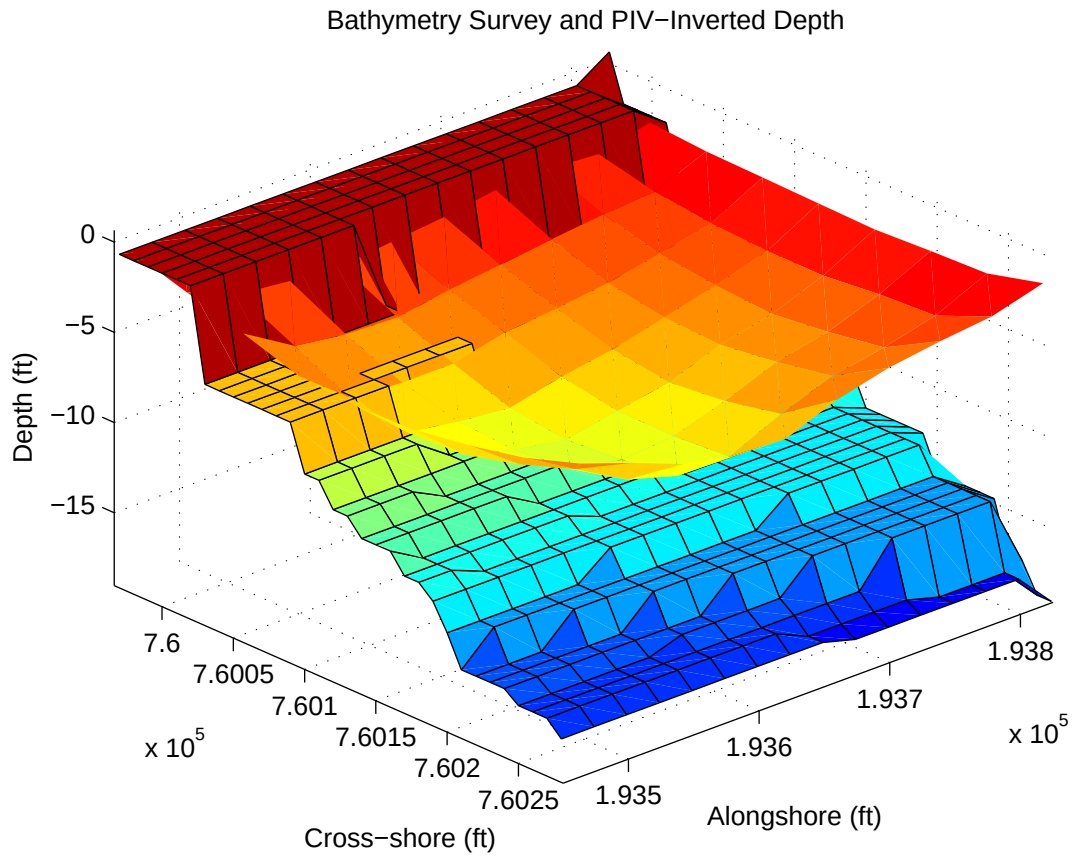


Figure 4.2: 3D calculated bathymetry vs. survey bathymetry, June 11, 2006, 11:35 GMT.

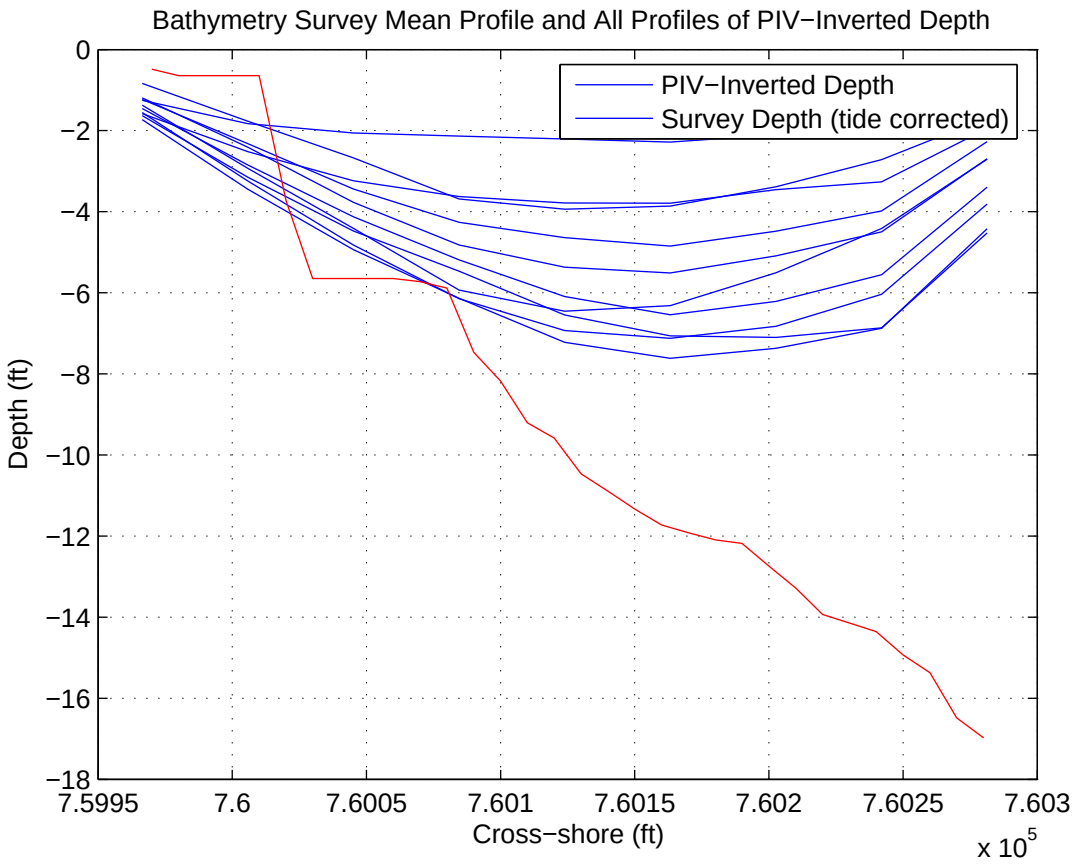


Figure 4.3: All calculated cross-shore lines vs. mean survey line, June 11, 2006, 11:35 GMT.

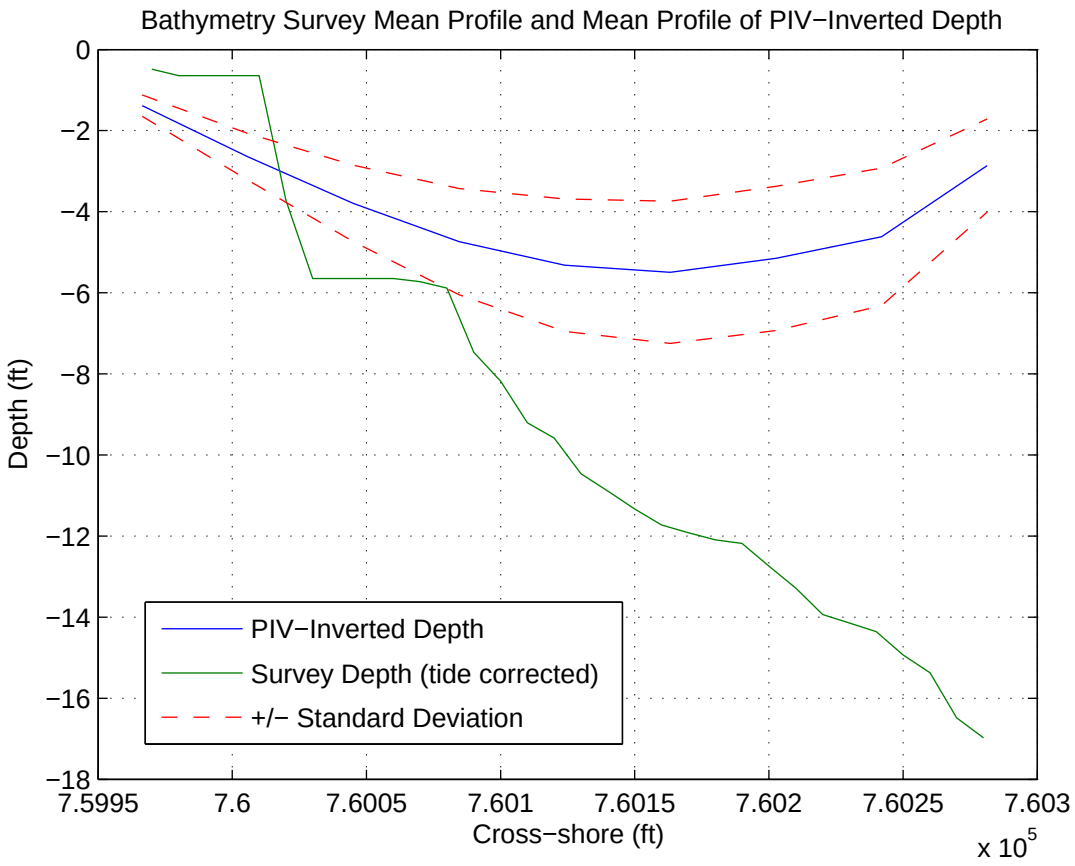


Figure 4.4: Mean calculated cross-shore line vs. mean survey line, June 11, 2006, 11:35 GMT.

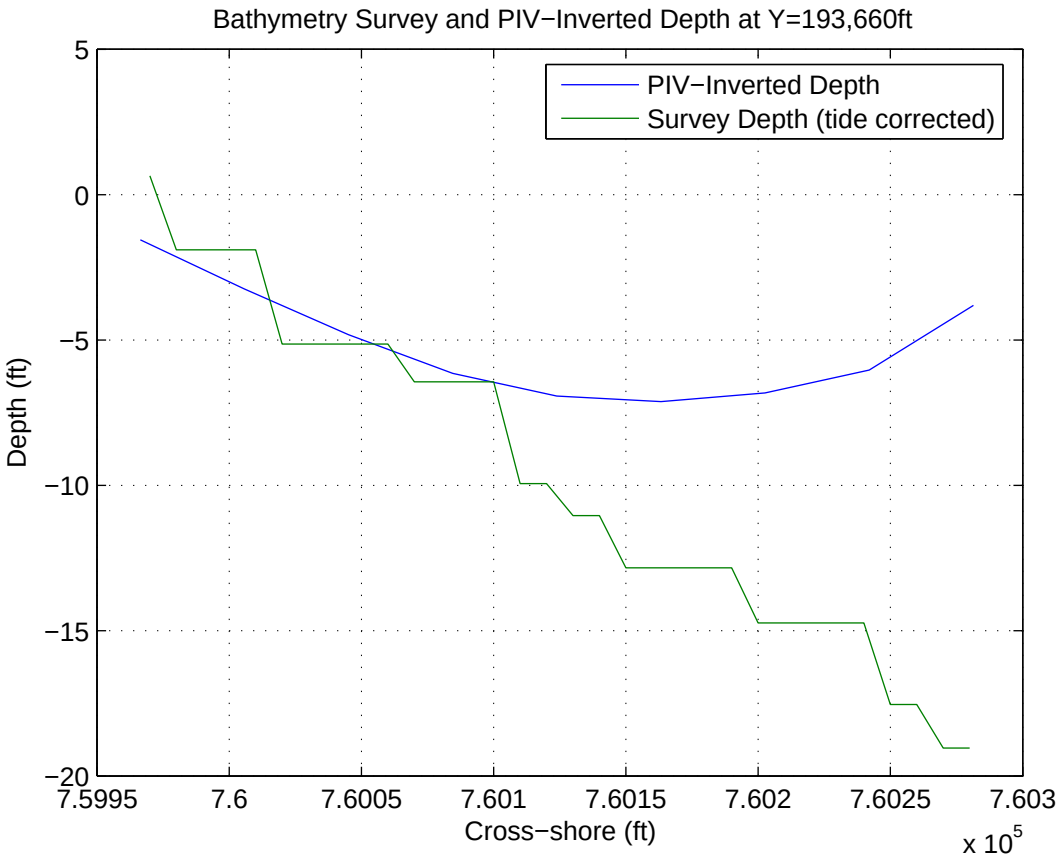


Figure 4.5: Calculated cross-shore line vs. survey line at Y=193,660ft, June 11, 2006, 11:35 GMT.

see because they are in deeper water and have not yet shoaled significantly enough to be visually detected.

Figure 4.6: Frame grab from video, June 11, 2006, 17:35 GMT.

4.3 Video from June 14, 2006

On this date, there is no data from the Army Corps directional wave gauge DE002, offshore of Bethany Beach. At NOAA buoy 44009, the waves have a significant height of about 0.75ft with a dominant period of about 15s. Wind direction, the closest approximation of wave direction, is 180°.

4.3.1 June 14, 2006, 14:35 GMT

Conditions in this video are overcast and blurry, with lighting a bit brighter to the north. A few wave crests can be seen propagating from the east-southeast

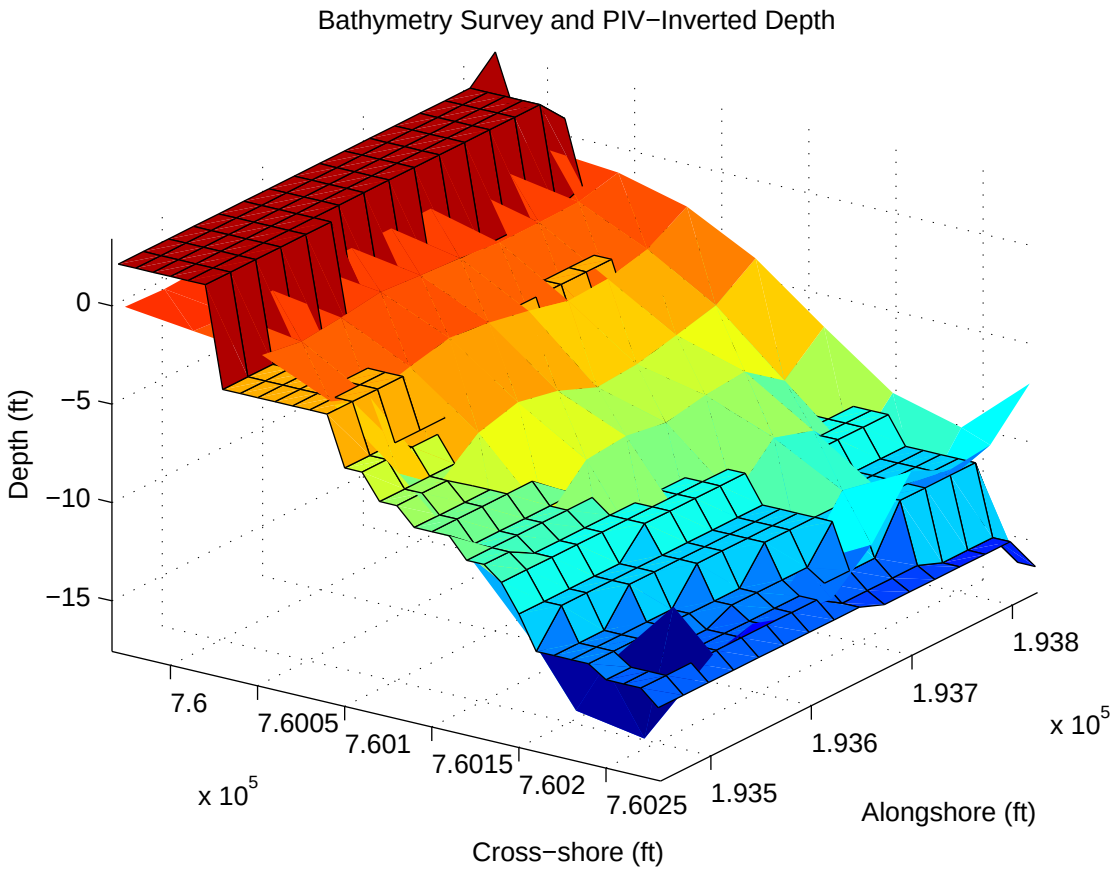


Figure 4.7: 3D calculated bathymetry vs. survey bathymetry, June 11, 2006, 17:36 GMT.

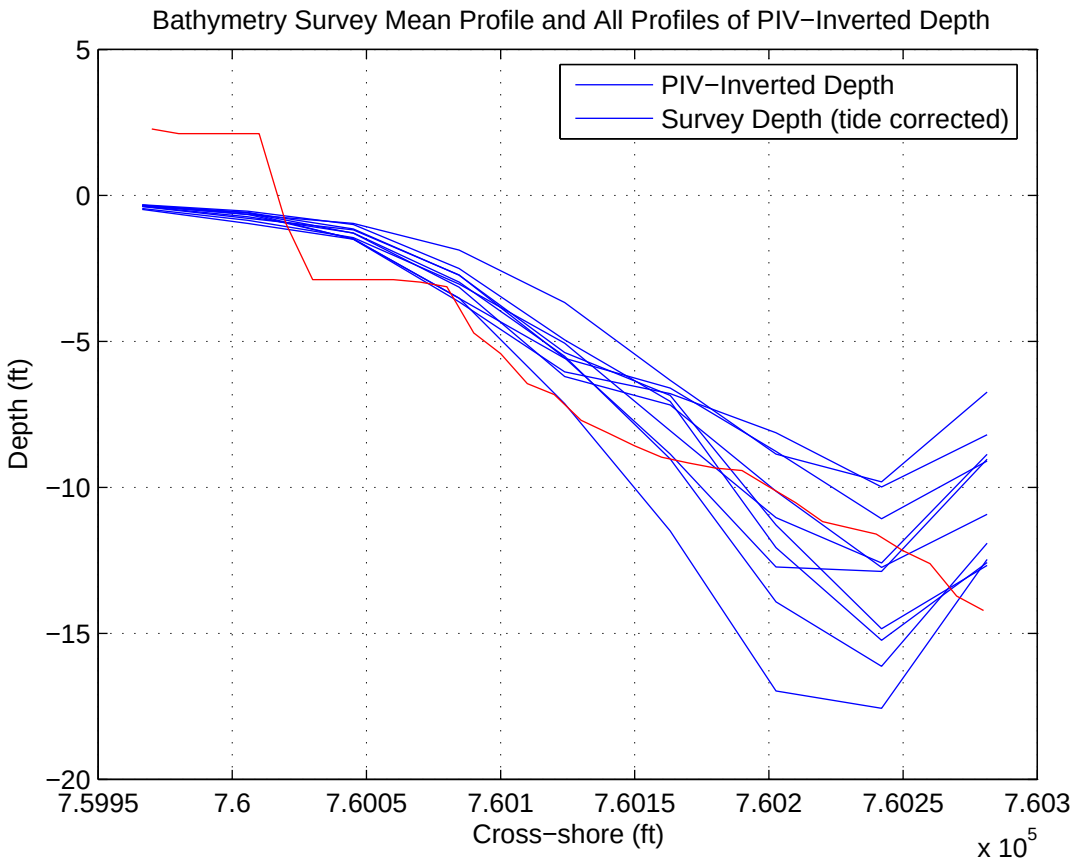


Figure 4.8: All calculated cross-shore lines vs. mean survey line, June 11, 2006, 17:36 GMT.

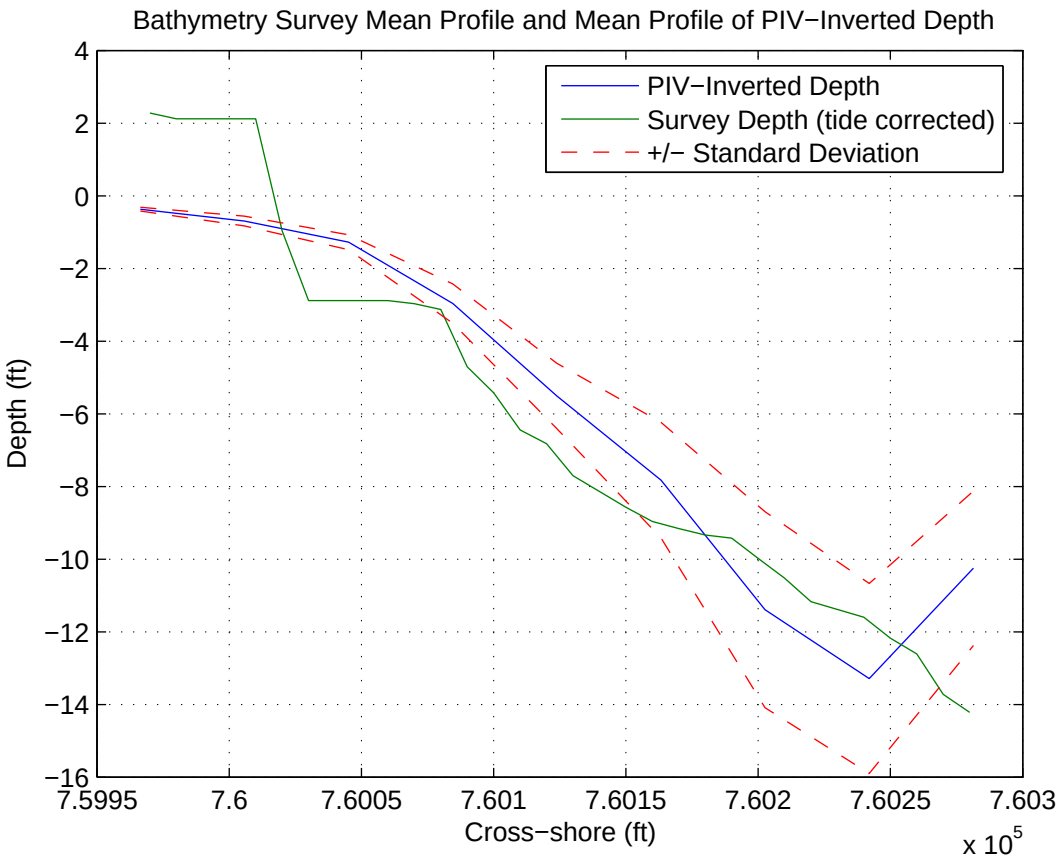


Figure 4.9: Mean calculated cross-shore line vs. mean survey line, June 11, 2006, 17:36 GMT.

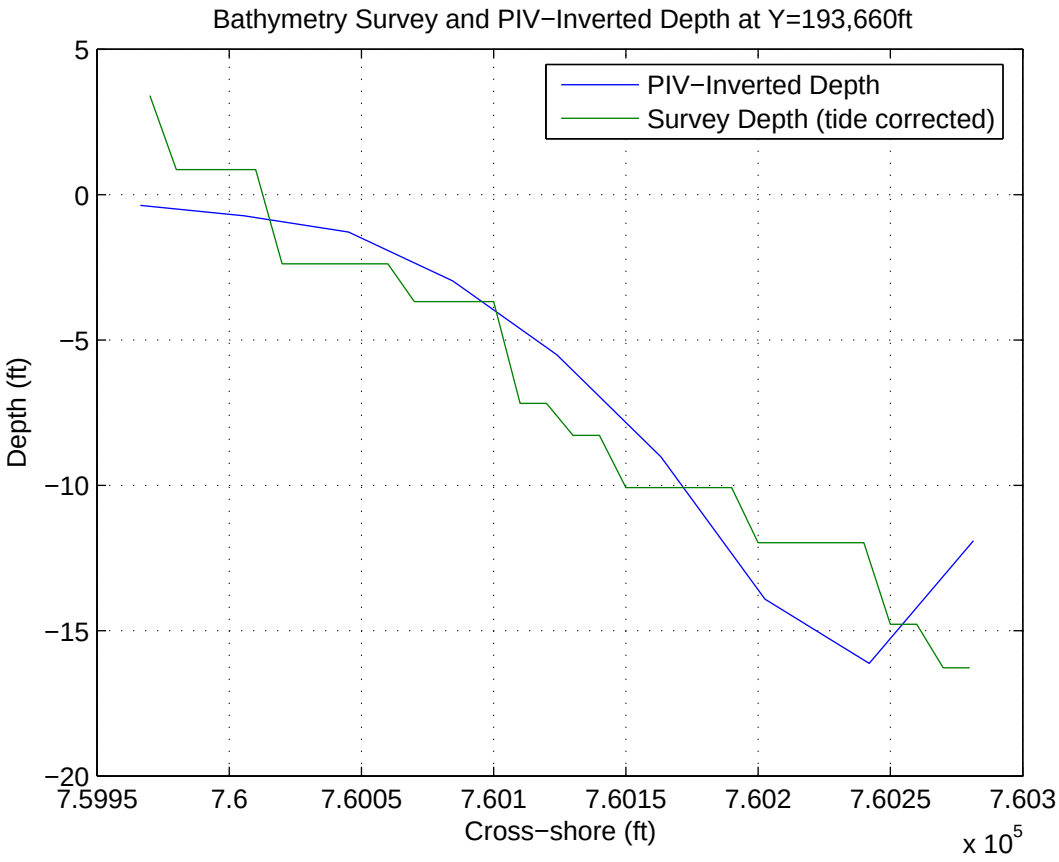


Figure 4.10: Calculated cross-shore line vs. survey line at Y=193,660ft, June 11, 2006, 17:36 GMT.

direction, seen in Figure 4.11. The northern end of the calculated bathymetry is too shallow, but it deepens towards the southern limit, as evidenced in Figure 4.12. This could be related to the fact that the sea surface is brighter at the northern end and waves are not as easly seen under that condition. The mean profile seen in Figure 4.14 correlates somewhat with the survey bathymetry, and there is a fairly large standard deviation in the alongshore. The same upward trend in the offshore previously seen shows up here again.

Figure 4.11: Frame grab from video, June 14, 2006, 14:35 GMT.

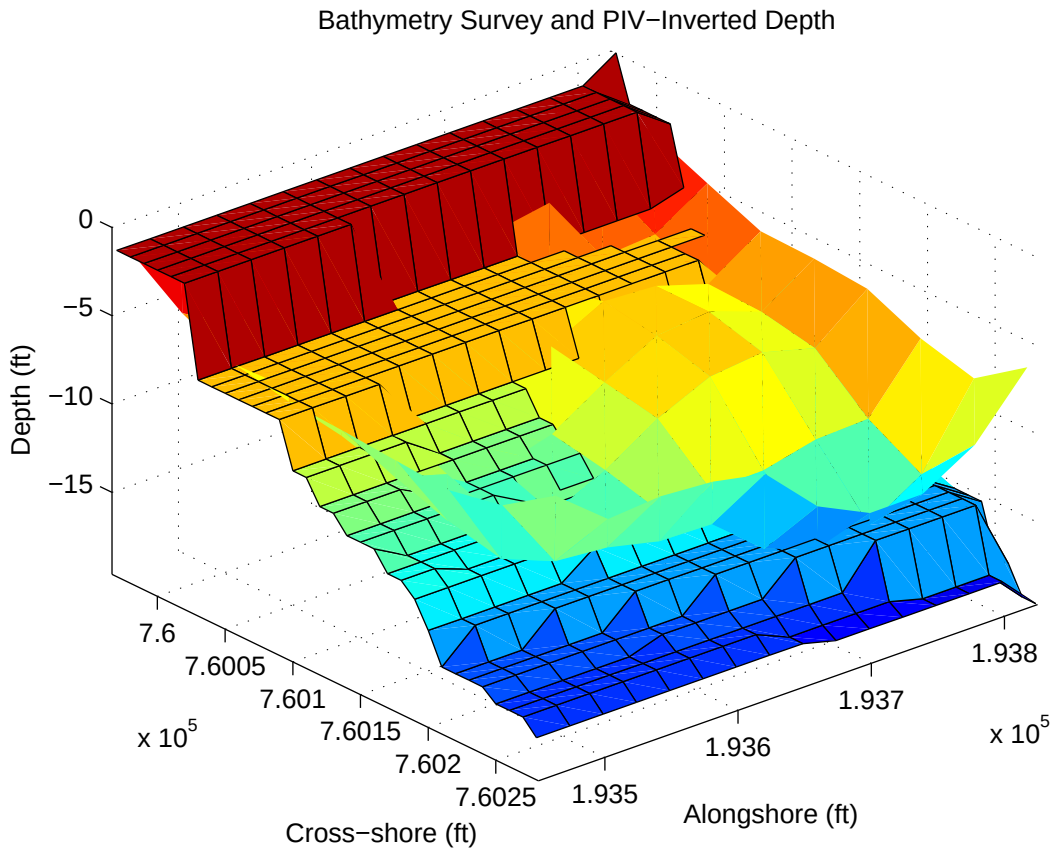


Figure 4.12: 3D calculated bathymetry vs. survey bathymetry, June 14, 2006, 14:35 GMT.

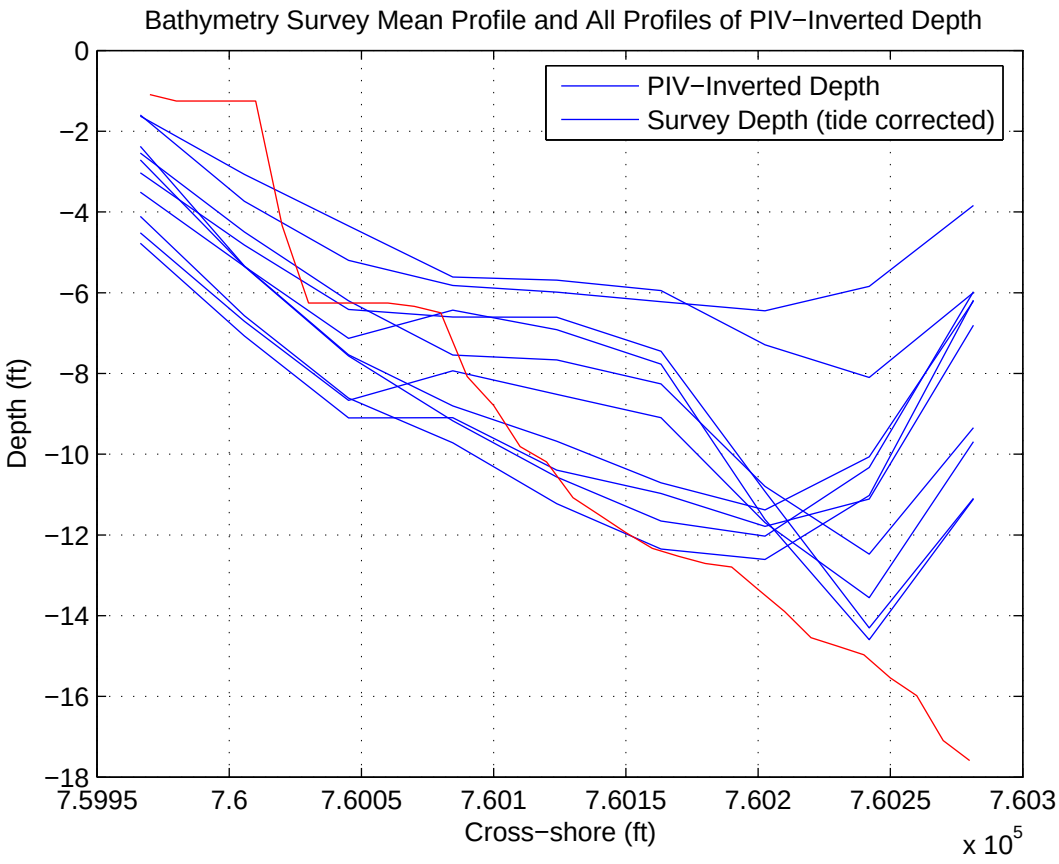


Figure 4.13: All calculated cross-shore lines vs. mean survey line, June 14, 2006, 14:35 GMT.

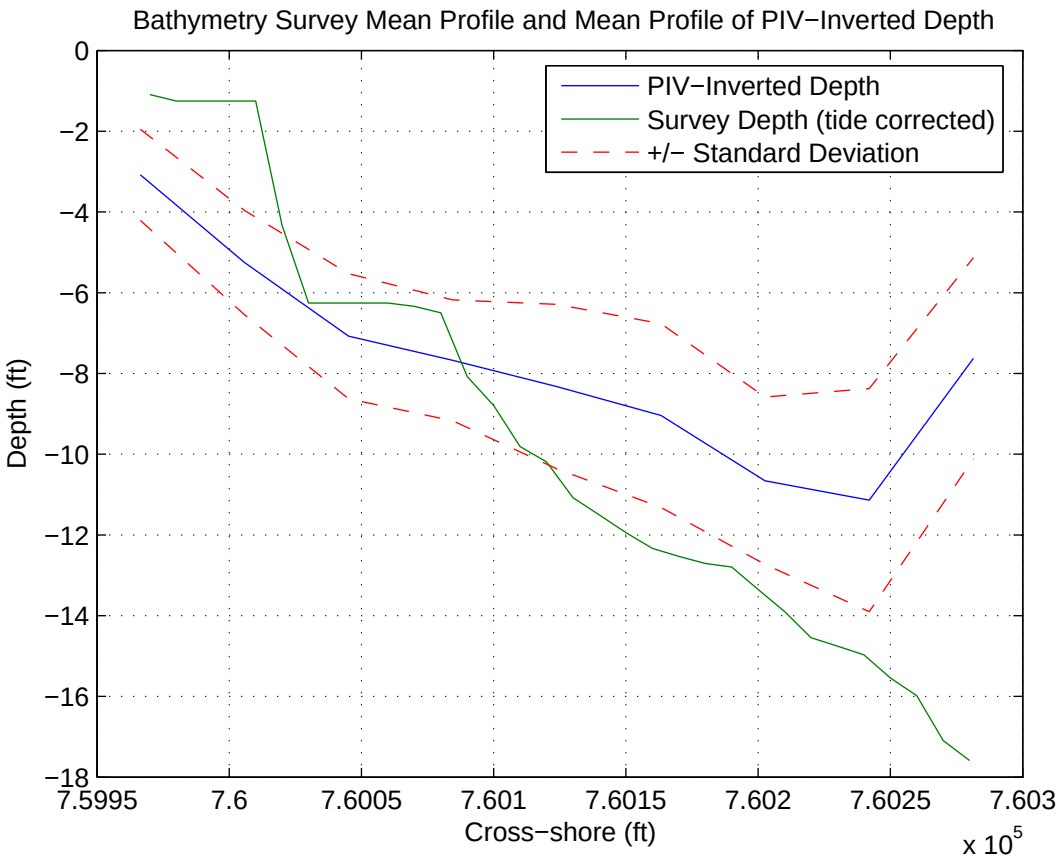


Figure 4.14: Mean calculated cross-shore line vs. mean survey line, June 14, 2006, 14:35 GMT.

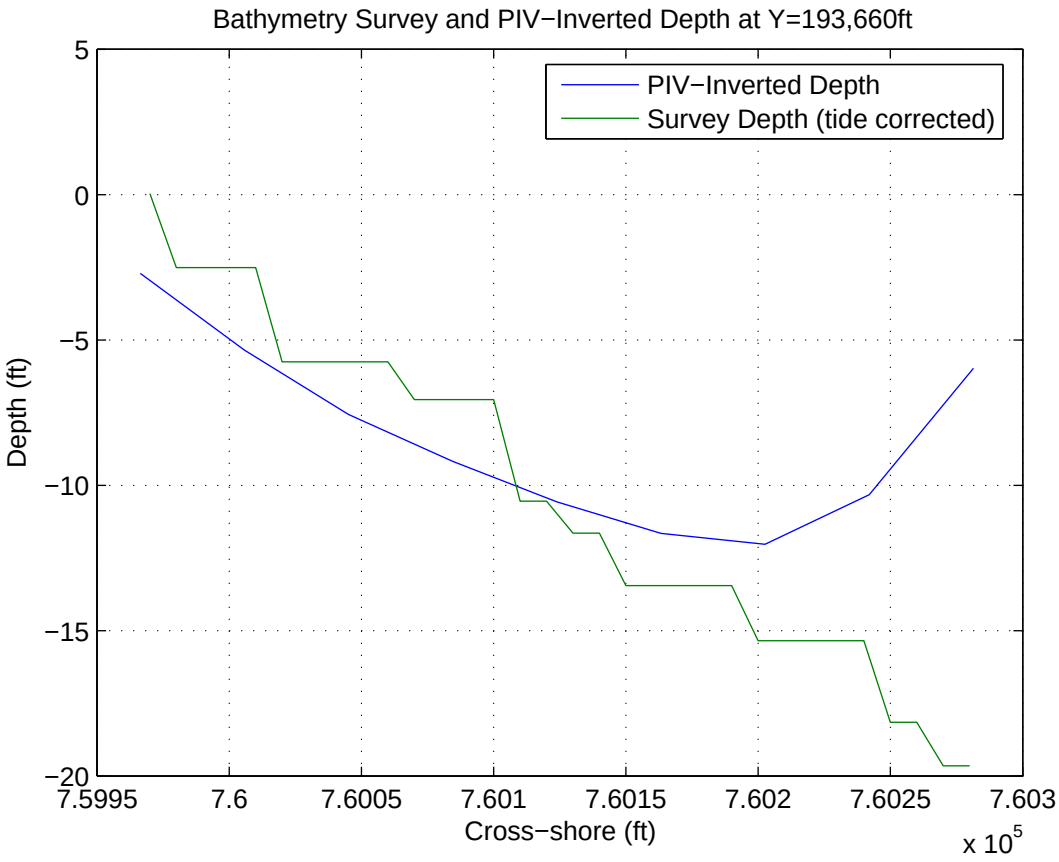


Figure 4.15: Calculated cross-shore line vs. survey line at Y=193,660ft, June 14, 2006, 14:35 GMT.

4.4 Video from June 20, 2006

On this date, there is no data from the Army Corps directional wave gauge DE002, offshore of Bethany Beach. At NOAA buoy 44009, the waves have a significant height of about 1ft with a dominant period of about 6s. Wind direction, the closest approximation of wave direction, is 220°.

4.4.1 June 20, 2006, 11:35 GMT

This video, seen in Figure 4.16, is overcast and blurry, with the northern end again a bit brighter than to the south. The sea surface is smooth, and practically no wave crest movement can be discerned. Bathymetry shown in Figure 4.17 is underestimated to the north, similar to the previous video. This is consistent with the previous example where the northern end was brighter than the rest of the video. The mean bathymetry in Figure 4.19 lines up very well with the survey data, as do the single profile lines in Figure 4.20. The upward offshore trend appears again. Like the June 11, 17:36 example, the lack of clearly visible wave crests does not appear to hinder the PIV analysis.

4.5 Video from June 27, 2006

On this date, the Army Corps directional wave gauge DE002, offshore of Bethany Beach, the waves have a significant height of about 3.25ft with a dominant period of about 9s. Wave direction is 100°. Both videos have an overcast sky.

4.5.1 June 27, 2006, 21:36 GMT

This video is overcast and blurry, but uniformly lit, as seen in Figure 4.21. There is a highly visible train of long-crested waves approaching from the southeast. Bathymetry is in general overestimated, with a deep hole near the northern offshore boundary. It is far too deep to likely be a real part of the bathymetry. This is clearly shown in Figure 4.23. From the mean and single profile comparisons in Figures 4.24

Figure 4.16: Frame grab from video, June 20, 2006, 11:35 GMT.

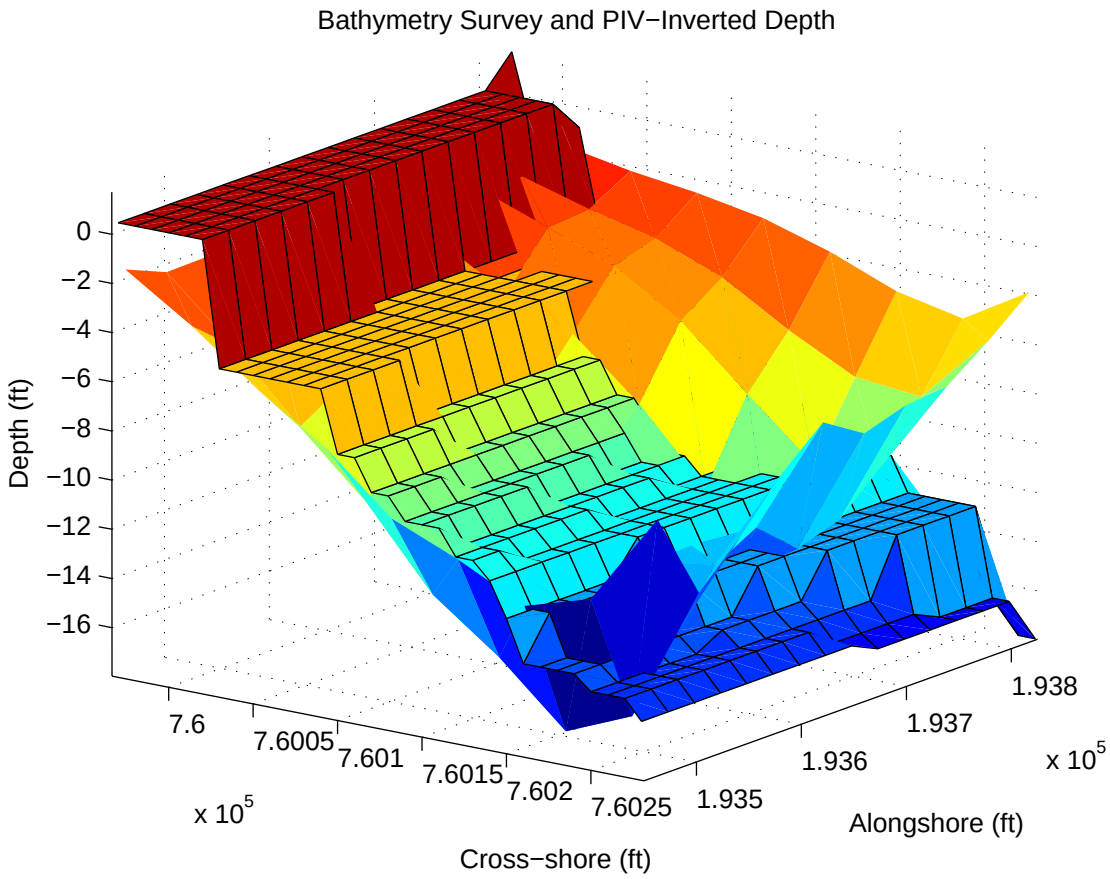


Figure 4.17: 3D calculated bathymetry vs. survey bathymetry, June 20, 2006, 11:35 GMT.

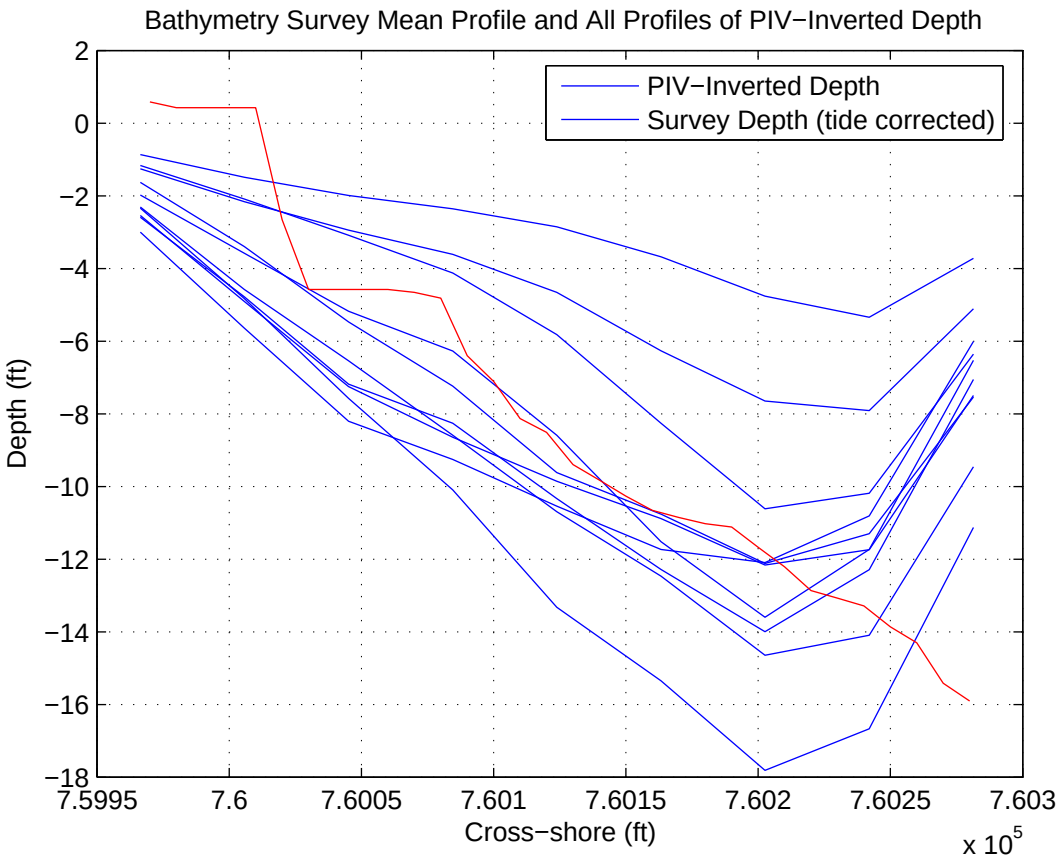


Figure 4.18: All calculated cross-shore lines vs. mean survey line, June 20, 2006, 11:35 GMT.

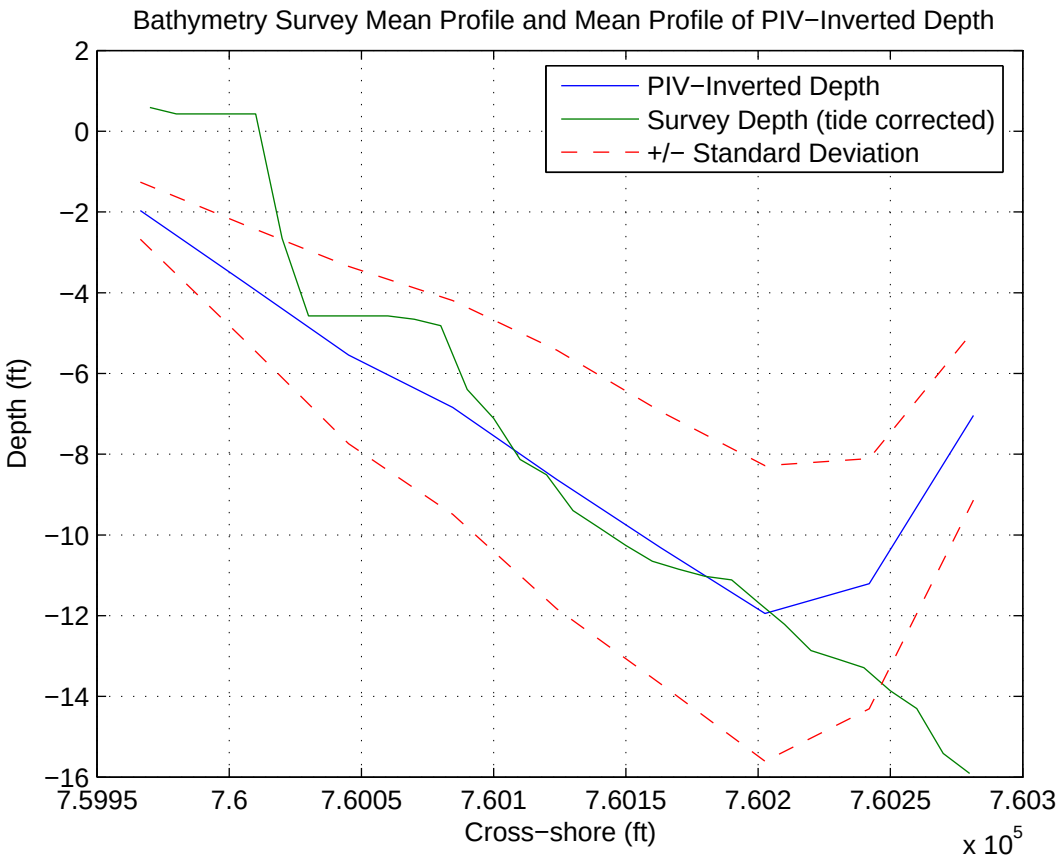


Figure 4.19: Mean calculated cross-shore line vs. mean survey line, June 20, 2006, 11:35 GMT.

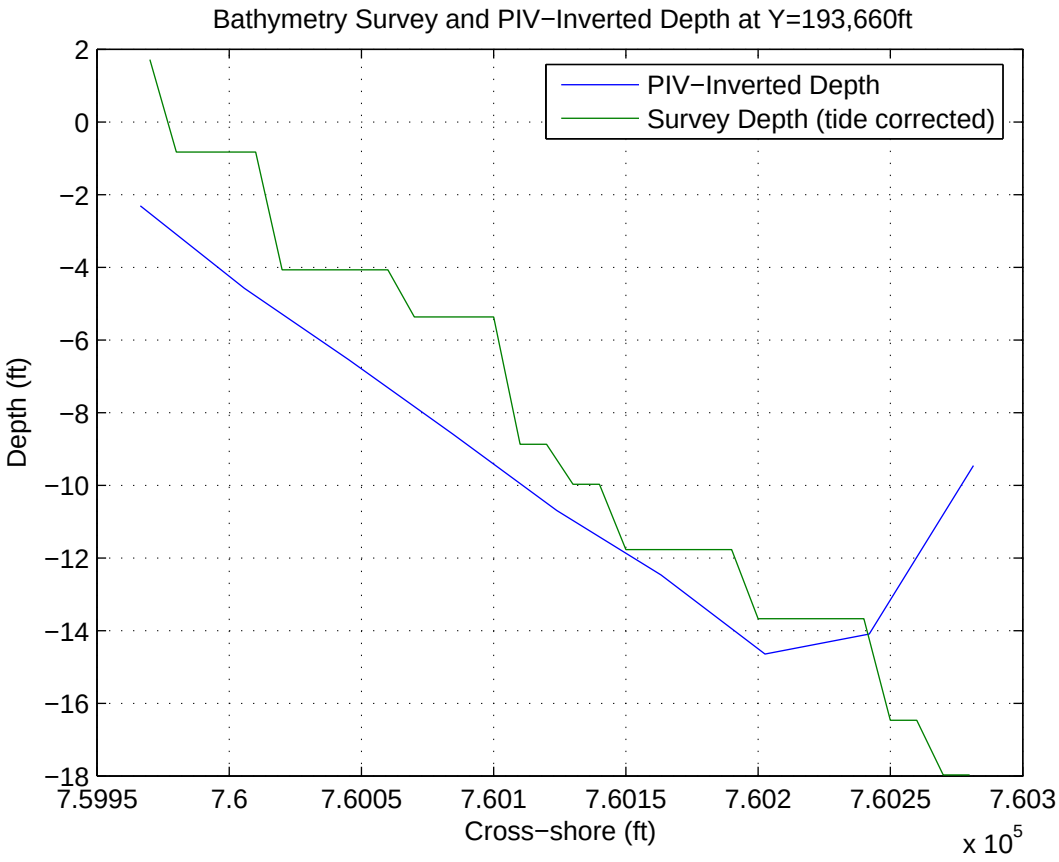


Figure 4.20: Calculated cross-shore line vs. survey line at Y=193,660ft, June 20, 2006, 11:35 GMT.

and 4.25, respectively, the slope of the area seems to be captured, but the depth is incorrect by a constant magnitude. The source of this error is unclear. The upward trend at the offshore persists.

Figure 4.21: Frame grab from video, June 27, 2006, 21:36 GMT.

4.6 Video from June 28, 2006

On this date, the Army Corps directional wave gauge DE002, offshore of Bethany Beach, the waves have a significant height of about 3.25ft with a dominant period of about 9s. Wave direction is 100°.

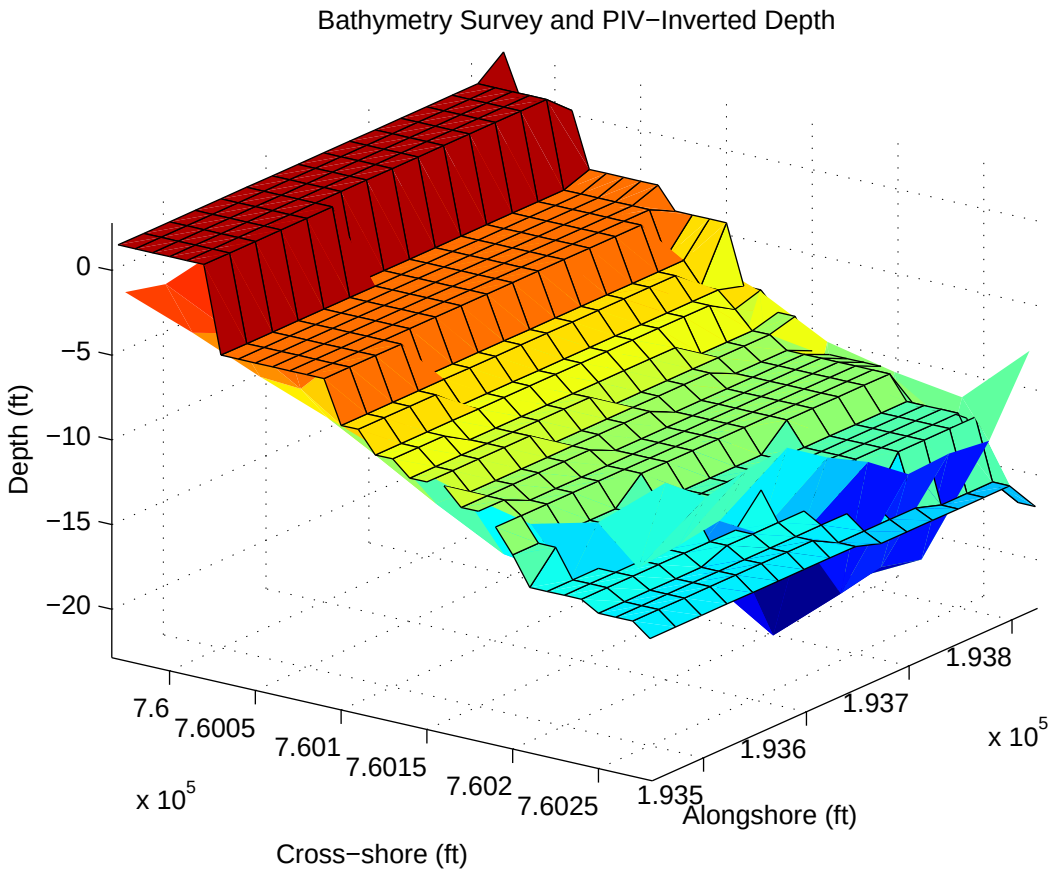


Figure 4.22: 3D calculated bathymetry vs. survey bathymetry, June 27, 2006, 21:36 GMT.

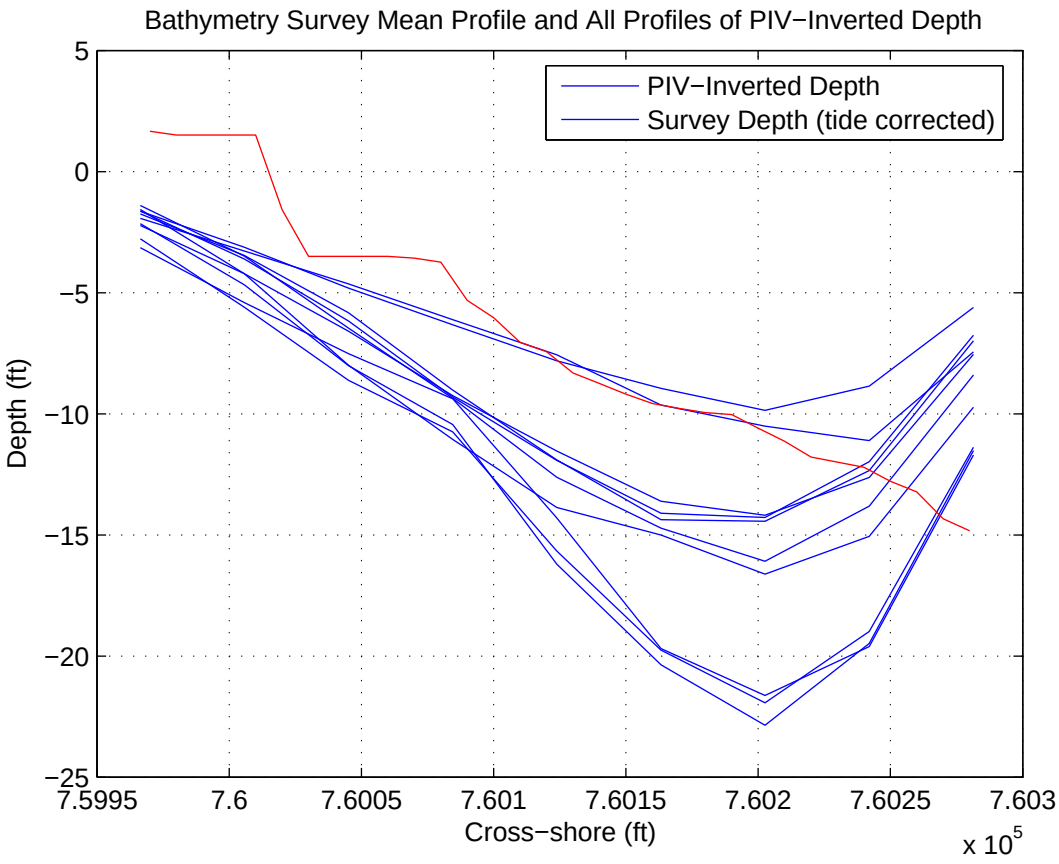


Figure 4.23: All calculated cross-shore lines vs. mean survey line, June 27, 2006, 21:36 GMT.

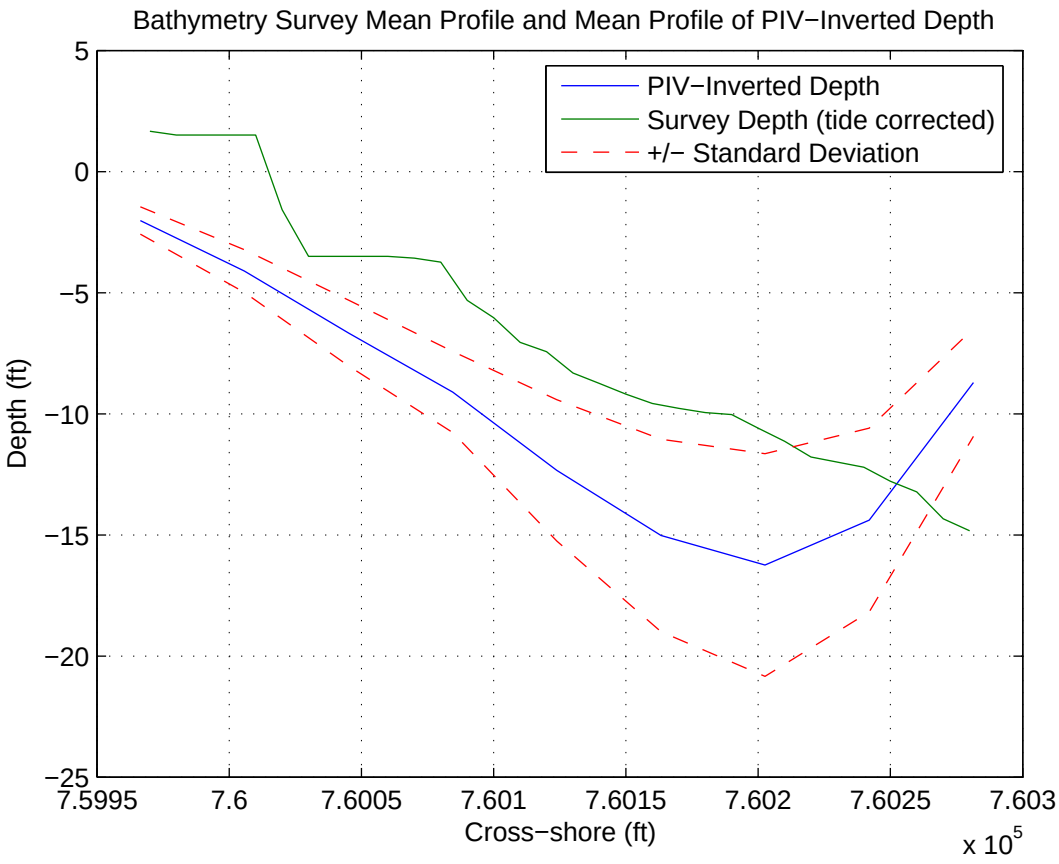


Figure 4.24: Mean calculated cross-shore line vs. mean survey line, June 27, 2006, 21:36 GMT.

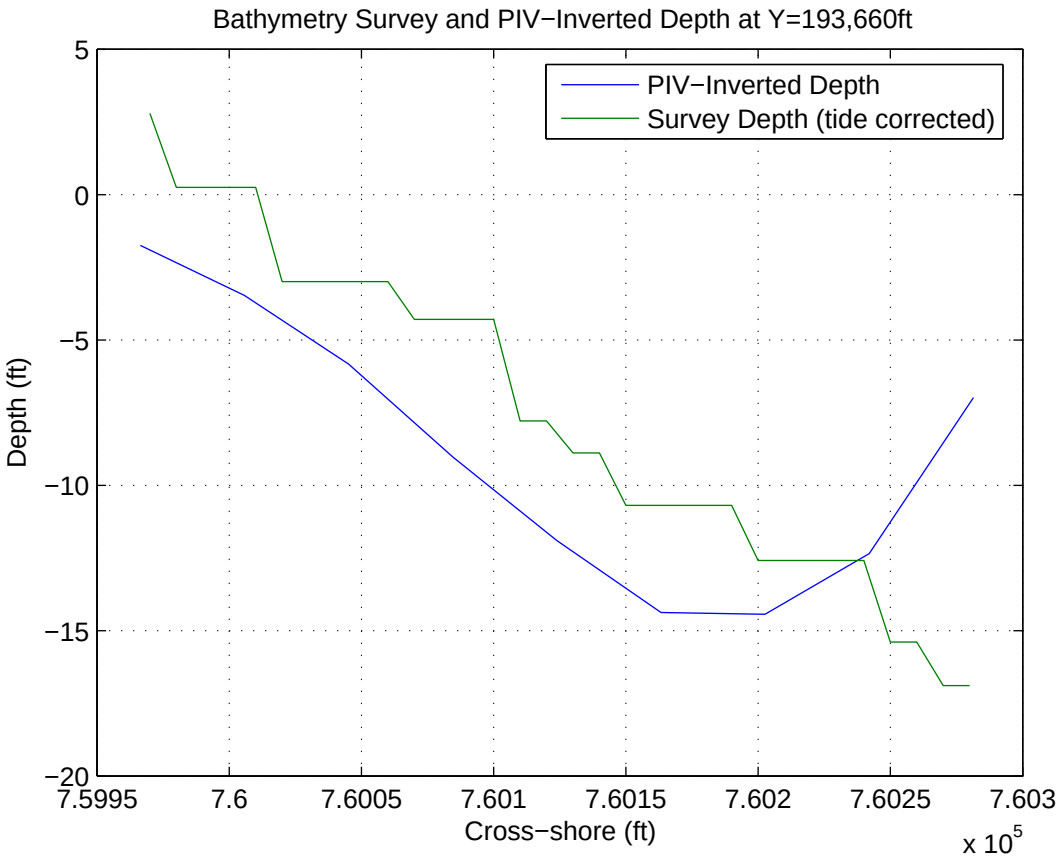


Figure 4.25: Calculated cross-shore line vs. survey line at Y=193,660ft, June 27, 2006, 21:36 GMT.

4.6.1 June 28, 2006, 11:35 GMT

This video shown in Figure 4.26 is blurry and overcast, but a bit brighter than the other overcast days. The sea surface is brighter to the north, and small, short-crested waves are coming in from the southeast. The calculated depth aligns with the survey bathymetry extremely well, evidenced in Figure 4.27. There is consistency in the alongshore profiles in Figure 4.28, which all follow the mean survey depth. Aside from the offshore upward trend, there are no sections exceedingly too deep or shallow. The single profile lines in Figure 4.30 correlate nearly perfectly.

Figure 4.26: Frame grab from video, June 28, 2006, 11:35 GMT.

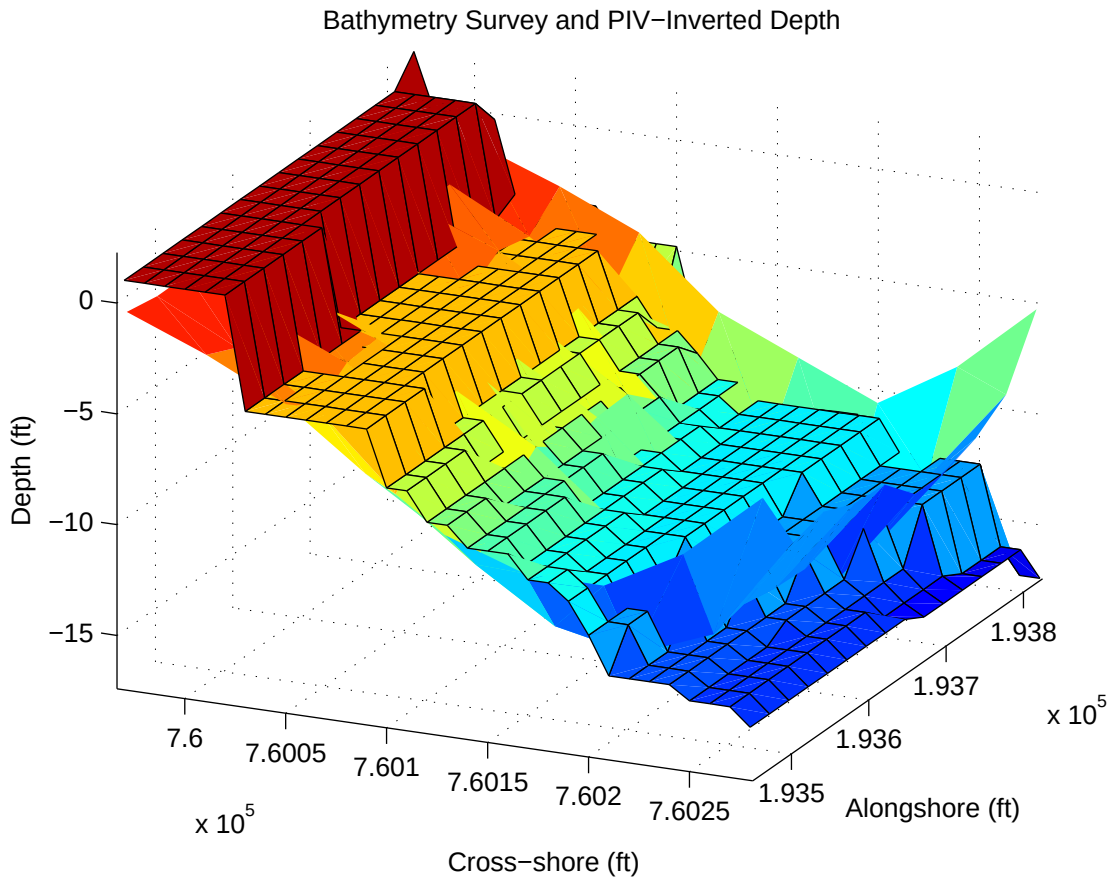


Figure 4.27: 3D calculated bathymetry vs. survey bathymetry, June 28, 2006, 11:35 GMT.

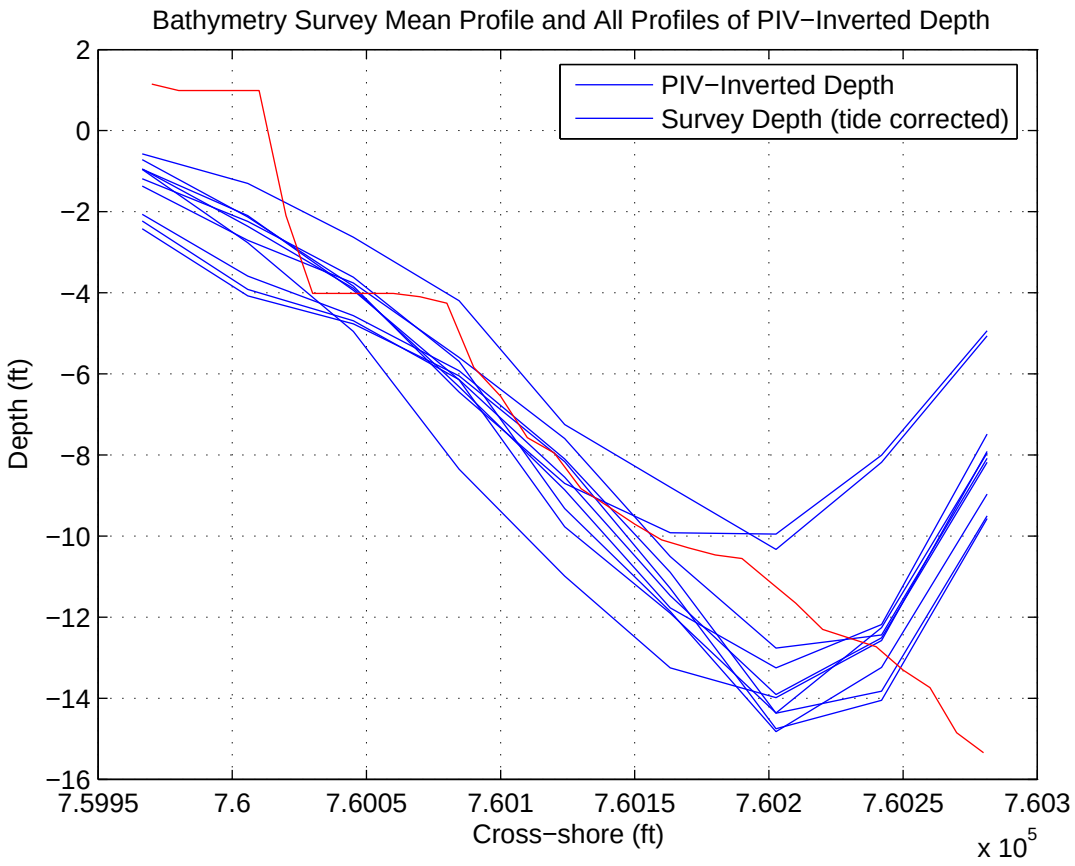


Figure 4.28: All calculated cross-shore lines vs. mean survey line, June 28, 2006, 11:35 GMT.

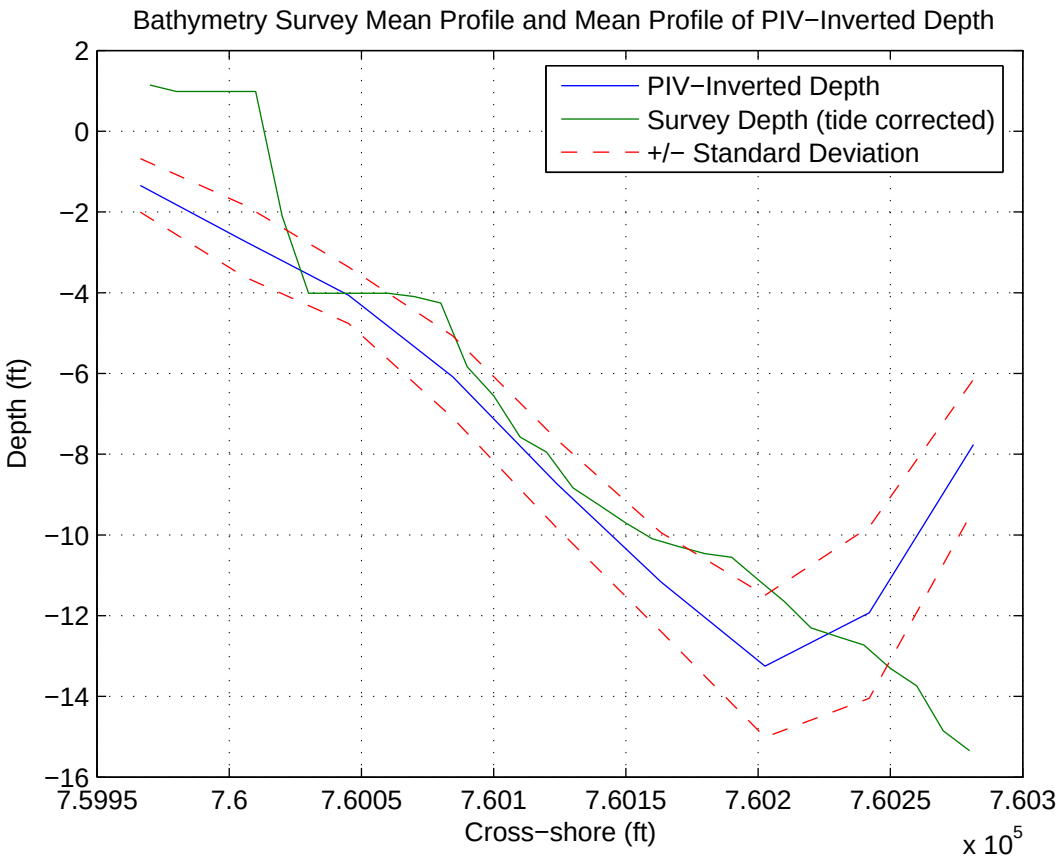


Figure 4.29: Mean calculated cross-shore line vs. mean survey line, June 28, 2006, 11:35 GMT.

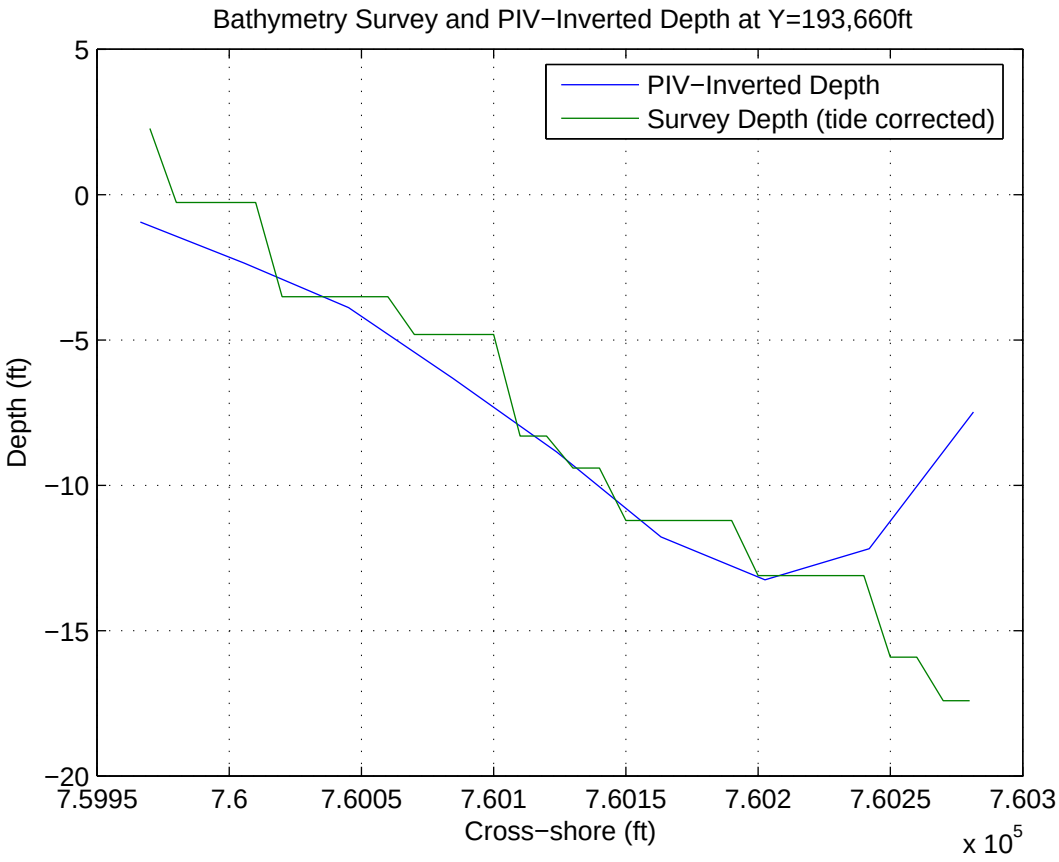


Figure 4.30: Calculated cross-shore line vs. survey line at Y=193,660ft, June 28, 2006, 11:35 GMT.

4.6.2 June 28, 2006, 12:35 GMT

Shown in Figure 4.31, this video is overcast and blurry with minimal local wind on the sea surface. Long-crested waves from the southeast are breaking in the center of the view. The depth-inversion produced multiple depth points of over 1000ft. These were filtered by substituting the 'bad' points for the mean of the remaining 'good' points in the same alongshore bin. This extreme outlier filtering was unnecessary for any of the other results. Even after filtering, there is a deep hole in the shallow area clearly seen in some of the profiles in Figure 4.33 that probably corresponds to the breaking waves. Upon breaking, linear wave theory can not be applied and the depth inversion using this technique shouldn't work. Nonetheless, the single profiles in Figure 4.35 agree somewhat in the shallows but break down everywhere else. The deep hole in the center may be a proxy for the breaking point, where extremely fast fluid motion is seen moving along the broken crest, caused by some backwash interaction with the obliquely breaking wave. Bores can move 30% faster than long waves in the same water depth, so that may account for some of the error.

4.6.3 June 28, 2006, 15:35 GMT

This video shown in Figure 4.36 is overcast but fairly bright, and the lighting is even. There are several highly visible, long-crested waves propagating from the southeast. The depth inversion results are quite good for this video, aside from a deep section in the center offshore region seen in Figure 4.37. Overall, the slope and magnitude of the calculated and survey bathymetry correlate well, as seen in Figure 4.38. The mean calculated profile in Figure 4.39 follows the mean survey data except for being skewed some offshore due to the aforementioned deep spot. The single profile of Figure 4.40 lines match well.

From the analysis of these videos, several recurring themes are evident. Uniform lighting of the sea surface appears to be paramount to successful PIV analysis

Figure 4.31: Frame grab from video, June 28, 2006, 12:35 GMT.

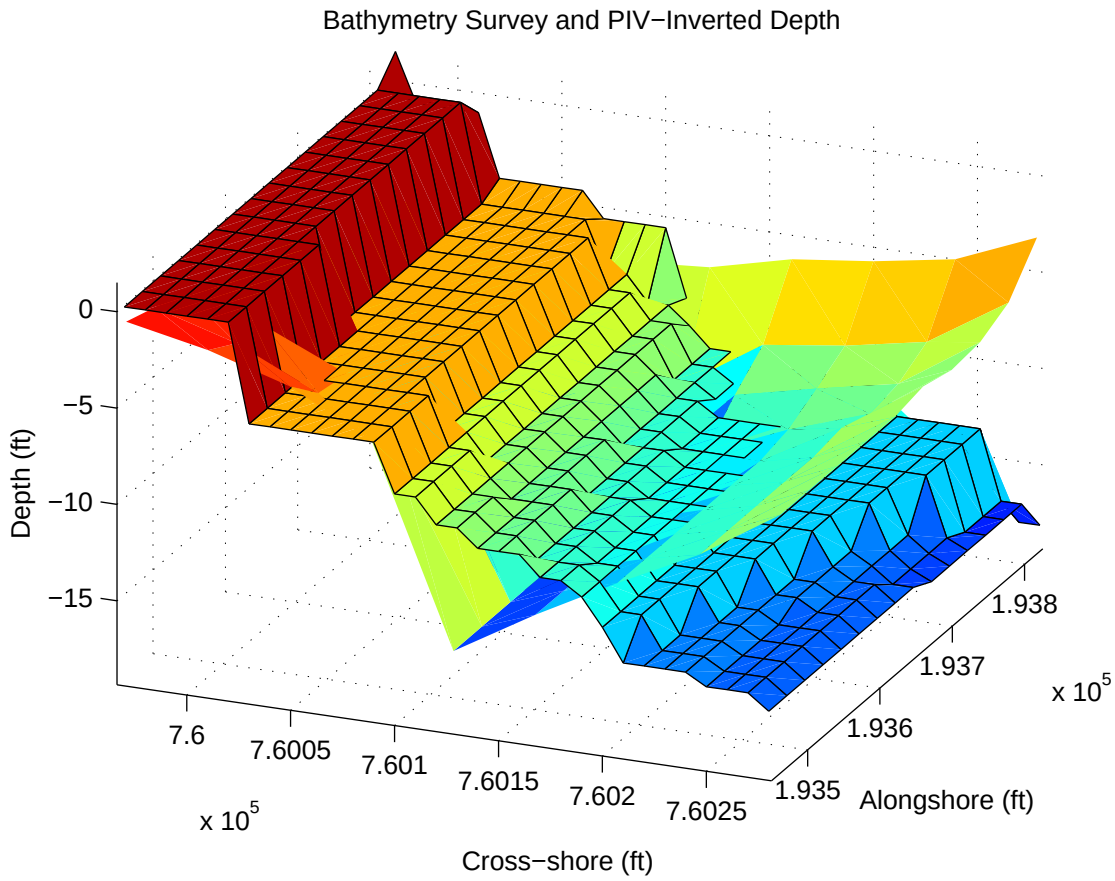


Figure 4.32: 3D calculated bathymetry vs. survey bathymetry, June 28, 2006, 12:35 GMT.

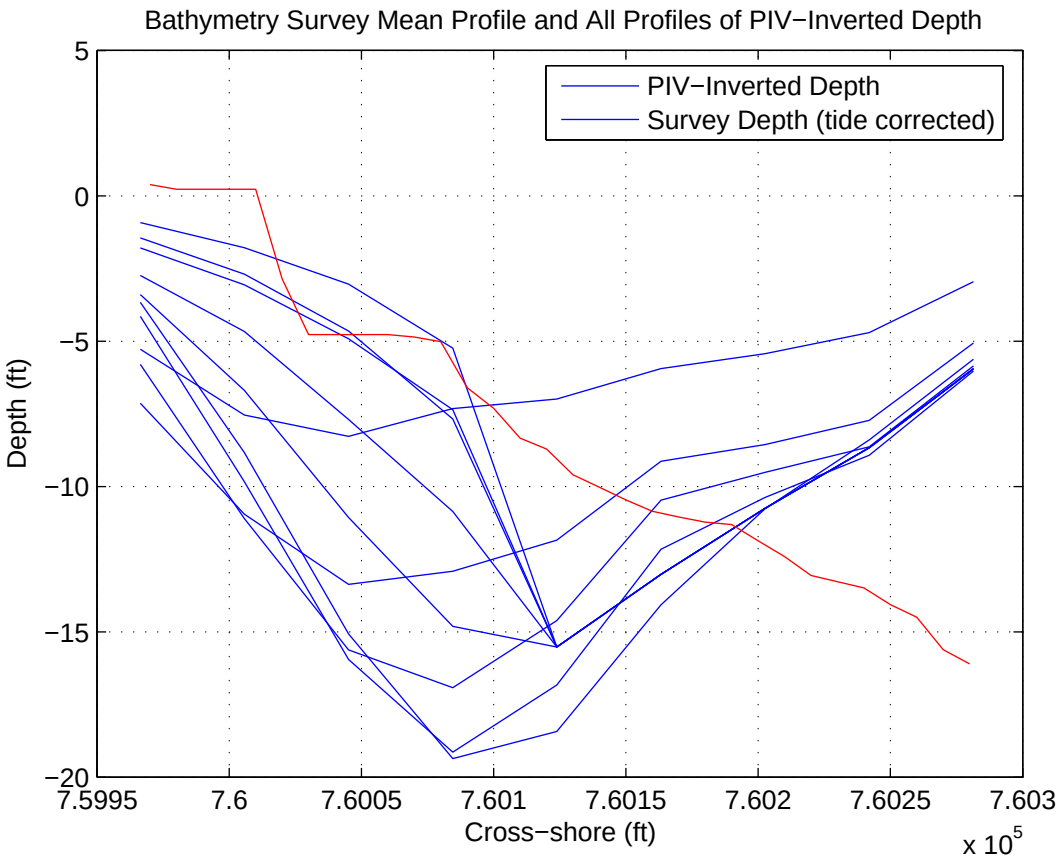


Figure 4.33: All calculated cross-shore lines vs. mean survey line, June 28, 2006, 12:35 GMT.

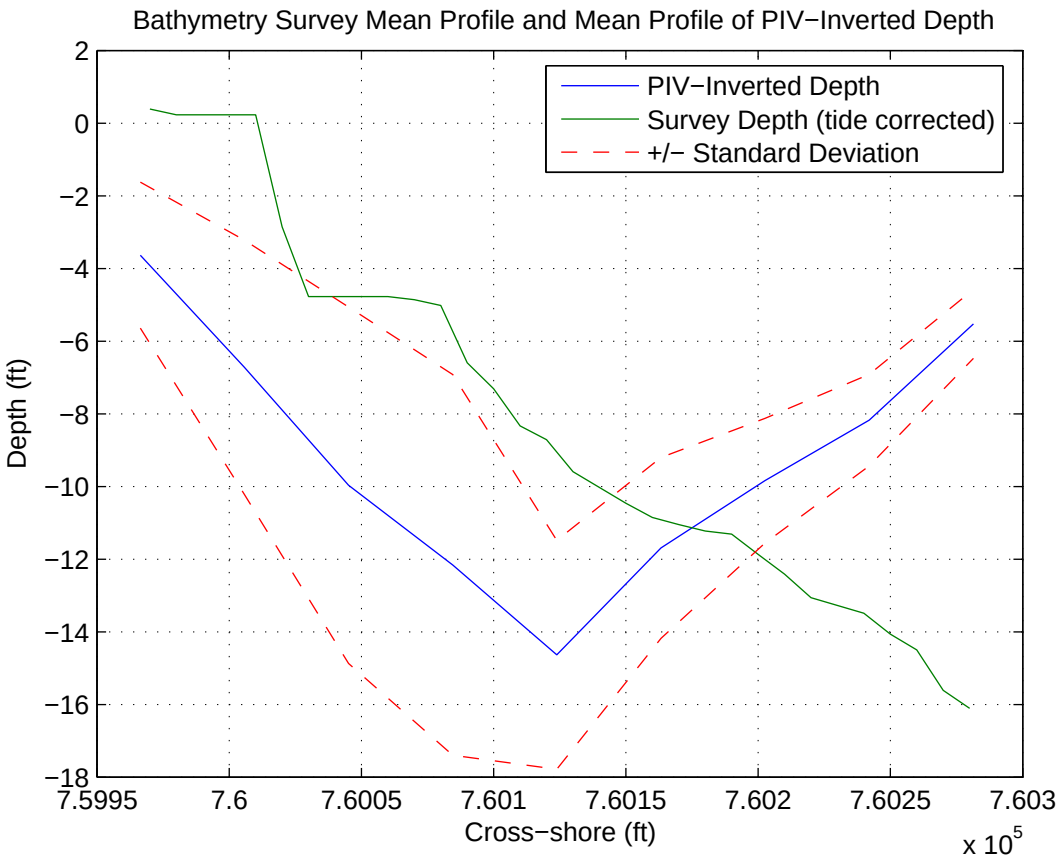


Figure 4.34: Mean calculated cross-shore line vs. mean survey line, June 28, 2006, 12:35 GMT.

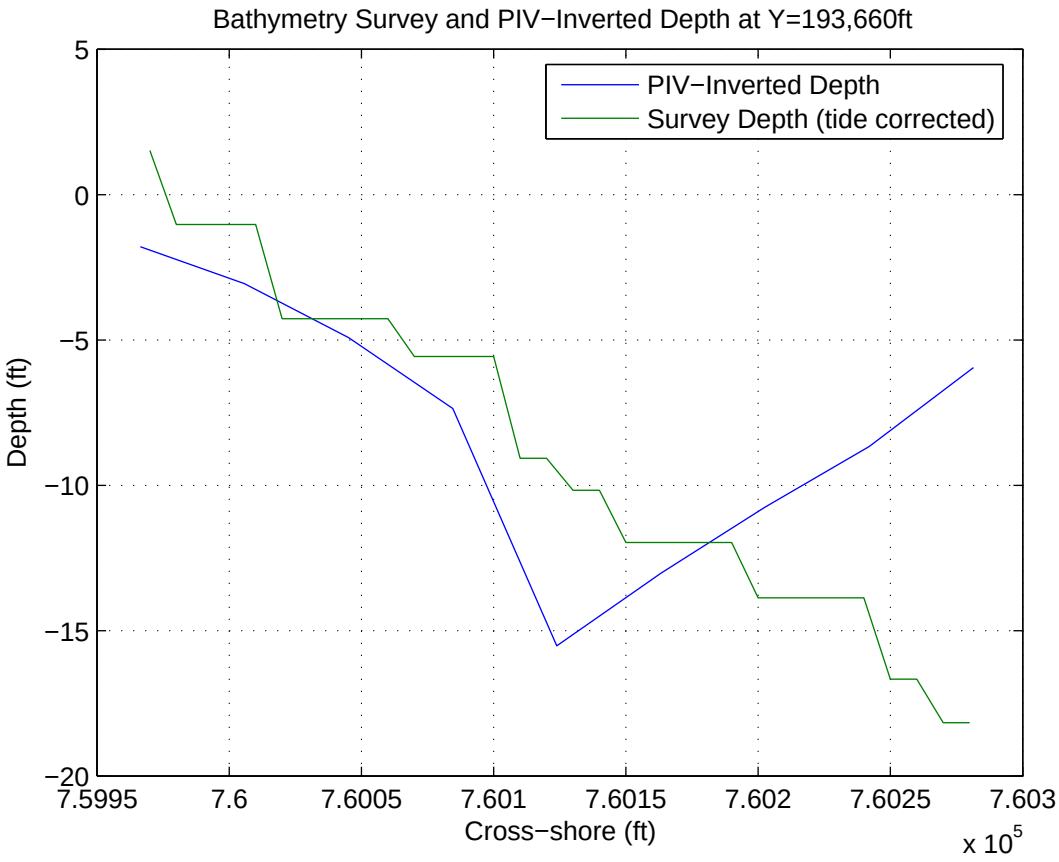


Figure 4.35: Calculated cross-shore line vs. survey line at Y=193,660ft, June 28, 2006, 12:35 GMT.

Figure 4.36: Frame grab from video, June 28, 2006, 15:35 GMT.

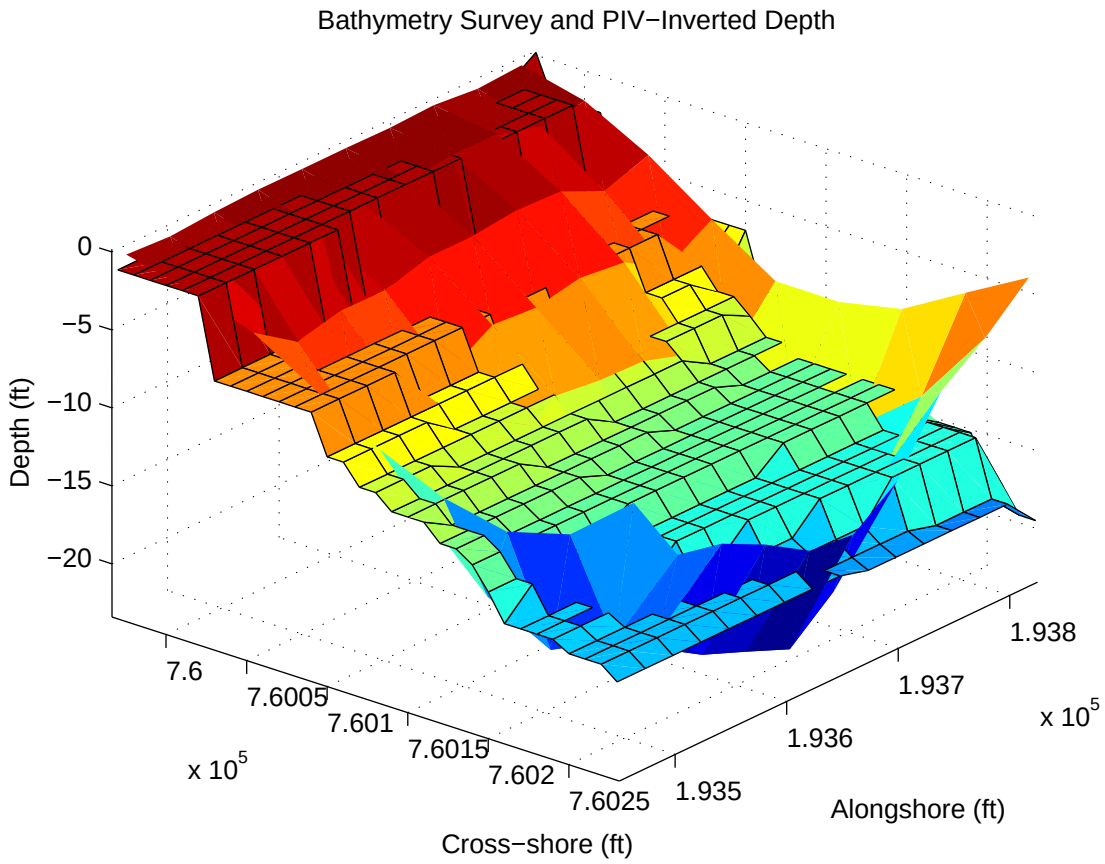


Figure 4.37: 3D calculated bathymetry vs. survey bathymetry, June 28, 2006, 15:35 GMT.

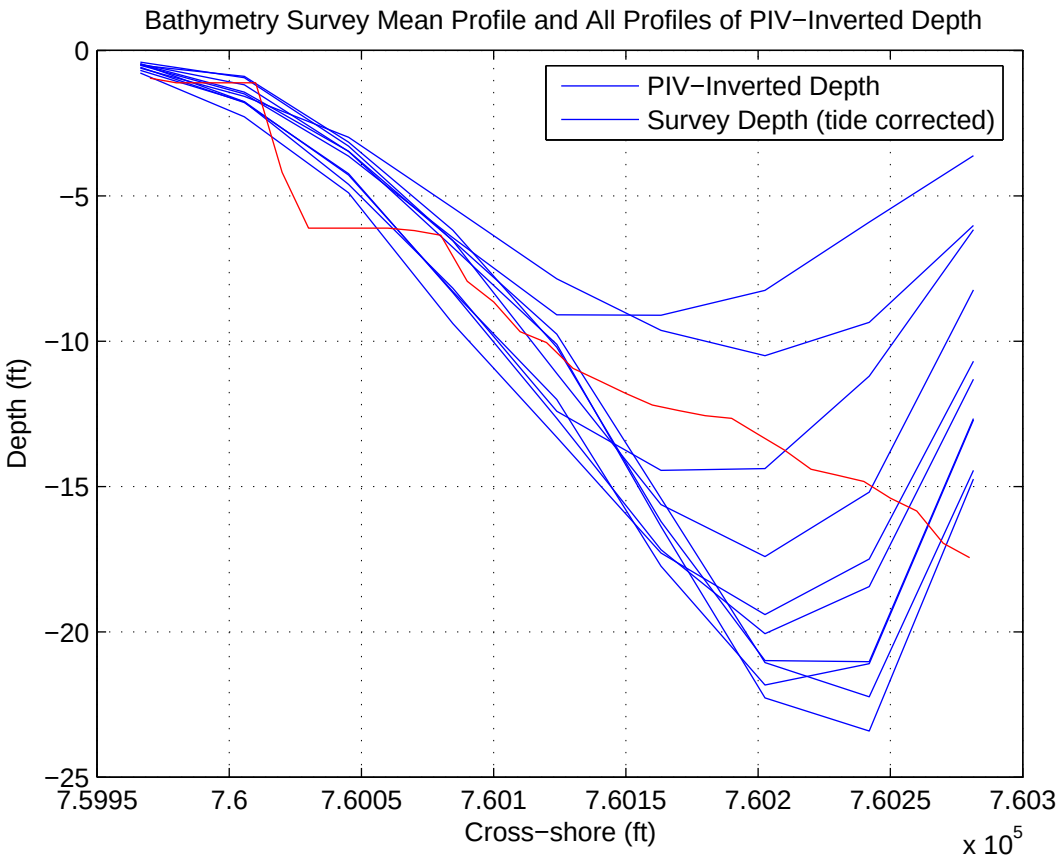


Figure 4.38: All calculated cross-shore lines vs. mean survey line, June 28, 2006, 15:35 GMT.

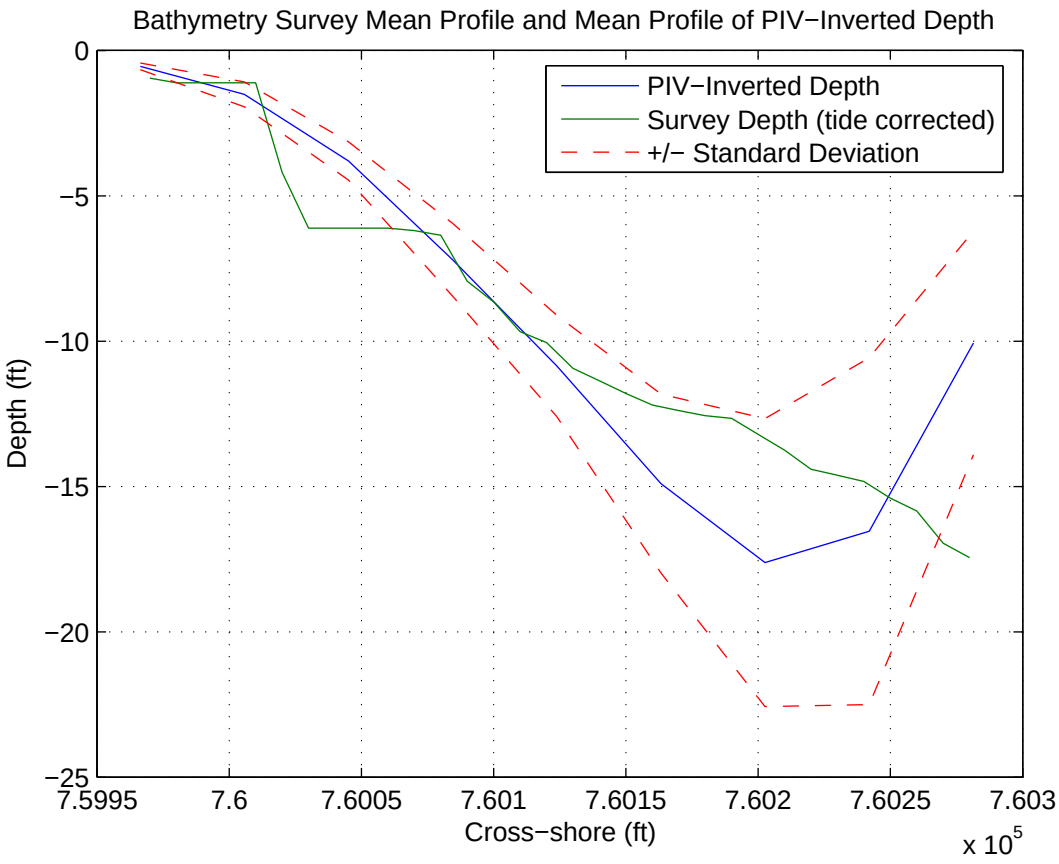


Figure 4.39: Mean calculated cross-shore line vs. mean survey line, June 28, 2006, 15:35 GMT.

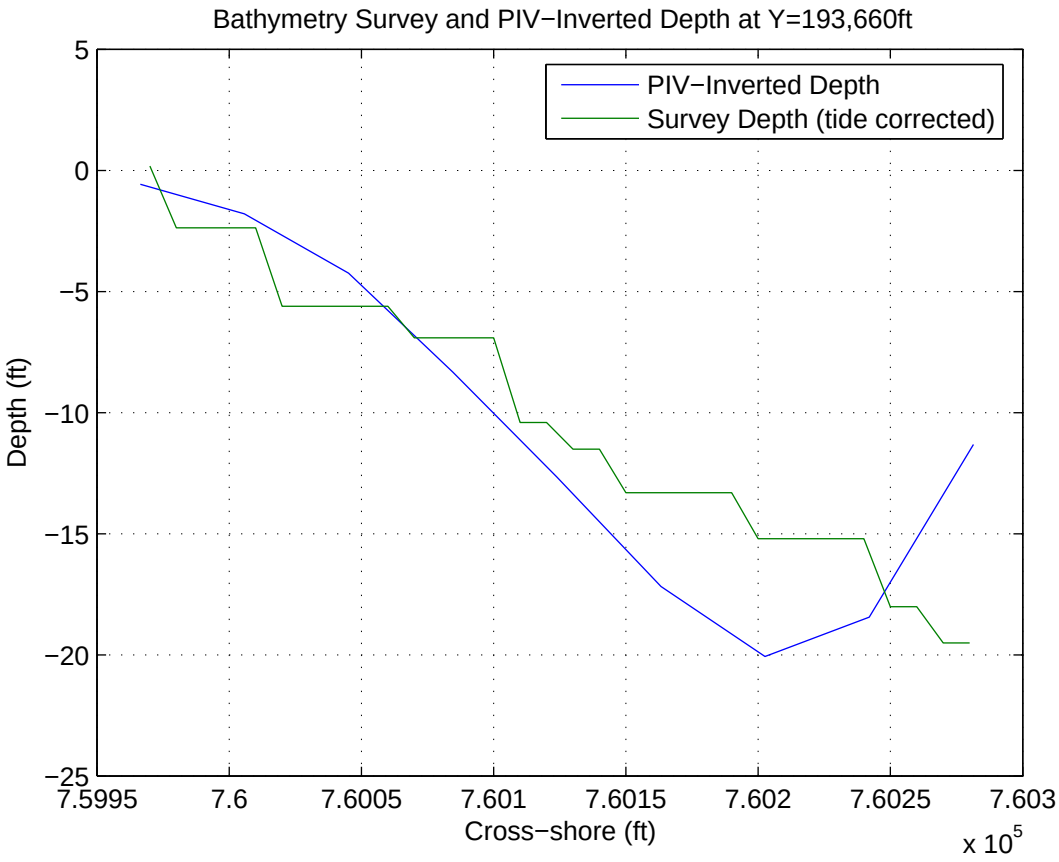


Figure 4.40: Calculated cross-shore line vs. survey line at Y=193,660ft, June 28, 2006, 15:35 GMT.

and depth inversion, whether sunny or overcast. Sun glare washes out the image intensity and PIV analysis underestimates the wave velocity because movement is difficult to detect. A smooth sea surface with weak local winds is favorable. Two extremes of wave conditions lead to successful depth inversion: either the waves are long crested and highly visible, or the sea surface is nearly calm with almost no wave crests seen at all. Both of these situations produce good results if the other aforementioned conditions are met. When the wave field is disorganized, PIV analysis is more likely to fail. The calculated depths consistently tend to decrease at the offshore boundary. This could be due to the waves not having shoaled significantly in the deeper water and are thus harder to visually detect. Aside from small shorebreak, breaking waves cause the bathymetry to be highly overestimated due to the PIV routine detecting nonlinear movements such as bore propagation and splashing in the breaker zone. The depth of the swash zone in general is somewhat overestimated due to the PIV routine detecting the swash bore propagation and this being mischaracterized as wave phase speed. In reality, this is a very shallow layer of water traveling at significant speed. Overall, with the proper lighting and environmental conditions, this depth inversion method produces favorable results.

Chapter 5

CONCLUSIONS

A video camera system has been set up and deployed in Bethany Beach, DE for the purpose of observing nearshore processes. The system captures several types of still images as well as video streams at regular intervals. The imagery is displayed in real time on the internet as well as archived on a computer at the University of Delaware. Lens distortion parameters and the geometry of the system deployment are solved so the imagery can be used in the scientific analysis of coastal dynamics such as the prediction of rip currents or estimation of wave speed.

Multiple video streams are rectified and velocity fields are estimated using MatPIV, a MATLAB implementation of particle image velocimetry. The maximum velocity calculated at each point is assumed to be the local wave phase speed, and this value is used to calculate the local depth according to long wave theory. This bathymetry is compared to survey bathymetry to determine the accuracy of the depth-inversion technique.

These videos covered several different scenarios of lighting and wave conditions. It appears that uniform lighting, whether bright or overcast, is ideal. This is when the PIV routine is able to accurately determine their speed of propagation. Glare from a sunrise appears to 'wash out' the image intensity and prevent MatPIV from seeing much movement on the surface. A sea surface with little wind disturbance is also advantageous. When waves are long-crested and highly visible, the PIV routine is able to capture their velocity. Surprisingly, videos that show a

nearly calm sea surface with little wave activity also produce favorable results. In between these two extremes is when analysis is more likely to break down. The calculated bathymetry consistently trends upward at the offshore boundary, possibly due to the fact that the waves have a milder steepness in deeper water and can not as easily be seen. This effect could also be due to the angle of the sea surface being shallower far offshore in relation to the camera. Wave breaking complicates matters due to nonlinearity of the motion and interaction with backwash and swash zone movements. Large breaking waves appear to make their signature in the bathymetry with a deep 'hole' near the breaker line. Depth in the swash zone is overestimated because MatPIV is seeing swash bore propagation, and this movement is being wrongly characterized as wave phase speed.

Some error could also be due to the inexact tide data used, but the magnitude of the difference is likely higher than what tidal difference there is between Bethany Beach and Lewes. Another source of error could be wave setup, which has not been taken into account here, but the magnitude of wave setup is relatively insignificant compared to the bathymetry deviation.

More recent and higher resolution survey data would go a long way towards validating the usefulness of this technique. But in general, with the proper surf conditions, this method of depth-inversion appears to be effective and yields rather favorable results.

REFERENCES

[1] Dean, R.G., and R.A. Dalrymple, Water Wave Mechanics for Engineers and Scientists, 353pp., World Scientific, River Edge, NJ, 1991.

[2] Fuchs, R.A., Depth determination on beaches by wave velocity methods, Report 74-1, University of California, Berkeley, 1953.

[3] Holland, K.T., R.A. Holman, T.C. Lippmann, J. Stanley and N. Plant, Practical Use of Video Imagery in Nearshore Oceanographic Field Studies, IEEE Journal of Oceanic Engineering, 22(1), 81-92, 1997.

[4] Johnson, J.W., Progress report: Wave-velocity method of depth determination by aerial photographs, Tech Report 155-10, University of California, Berkeley, 1949.

[5] Stockdon, H., and Holman, R. A., Estimation of wave phase speed and nearshore bathymetry from video imagery, Journal of Geophysical Research 105(C9), 22,015-22,033, September 2000.

[6] Sveen, J.K., MatPIV, the PIV toolbox for MATLAB, http://www.math.uio.no/j̃ks/matpiv/, 2006.

[7] Willert, C.E. and M. Gharib, Digital Particle Image Velocimetry, Experiments in Fluids 10(4), 181-193, January 1991.

[8] Williams, W.W., The determination of gradients on enemy-held beaches, The Geographical Journal 109(1/3), 76-90, Jan.-Mar. 1947.

Appendix A

THE OBSERVATION SYSTEM

A.1 Location

To extract useful data from the camera imagery, a clear view of the area of interest with no obstructions or discontinuities is needed. In order to get a comprehensive view of the shoreline and still be able to see smaller details, the vantage point for the cameras should be as high as is necessary to see the entire region unobstructed. On the stretch of coastline of interest, between the Indian River Inlet to the north and Ocean City, MD to the south, there are very few tall buildings. The tallest are the condominiums that comprise the Sea Colony beach resort in Bethany Beach, DE. The Sea Colony resort is a luxury beach resort comprised of nine tall condominium buildings situated essentially right on the beach, just behind the small dunes that separate the beach from inland.

Each building is about 90-110ft tall from ground level to rooftop. From the roof, there is a clear view of the shoreline from approximately 1/2 mile north the the downtown Bethany Beach boardwalk to about the same distance south until the shoreline curves inland a bit and the view is blocked by other Sea Colony condominiums. This is ample view for study.

The specific site within Sea Colony chosen for system deployment is the Brandywine Building, the second tall building from the northern end of the complex. See Figure 2.1 for an aerial view of the Sea Colony resort, with the deployment location circled in red. The apex of the Brandywine rooftop is about 90ft from

ground level and has a wide view of over a mile of coastline north to south, and east-west from the dune-beach interface to the horizon. The corner closest to the shore faces east so that one side of the roof faces northeast and the other faces southeast. At the corner, there is a small exhaust stack that obstructs the view looking down from the building straight towards the beach, but this can be worked around and is not a major issue. The rooftop itself consists of ridged sheet metal on a 25 percent grade, with no guardrails or parapets. It's rather precarious, but after installation of the cameras it isn't necessary to navigate the roof. Figure A.1 is a picture taken from the rooftop, where the roof construction and exhaust stack are visible. Figures A.2 and A.3 show, respectively, the northern southern limits of view from the deployment location.

Figure A.1: The view from the Brandywine Building rooftop at the northeast apex. The exhaust stack obstruction is clearly visible, as is the sheet metal construction of the roof.

Figure A.2: The view to the north from atop the Brandywine Building.

Figure A.3: The view to the south from atop the Brandywine Building.

Weight	440g
Dimensions	74x65x120mm
Lens format	1/2" CCD
Lens mount	CS-mount (C-mount adaptable)
Video resolution	480-line horizontal
Input voltage	24V AC or 12V DC
Power consumption	4.5W

Table A.1: Specifications of Panasonic WV-CL924A camera.

Rooftop access in the building comes from a door and ladder exiting a spacious climate-controlled room that houses the elevator controls for the building. The door from roof to this room is about 50ft from the apex of the roof where the cameras are installed, allowing for reasonable lengths of power and video cabling. The air conditioning in the room makes for a comfortable residence for the computers, transformers, and other hardware that make up the monitoring system.

A.2 Hardware

The video monitoring system was purchased from and designed with the aid of Erdman Video Systems, a company based in Miami, FL. Many of their clients use the systems for simple beach cams and the like, but they are perfectly capable of being used in a scientific study. The system is basically five video cameras in weatherproof housings linked to two computers which control the system.

The five cameras are all identical Panasonic WV-CL924A color CCTV cameras. They have a 1/2" CCD and take CS-mount lenses standard, and C-mount lenses are compatible with a simple adaptor ring. Maximum image resolution in color mode is 640x480 pixels, and the video output takes a coaxial cable and BNC connector. Power to the terminals can be supplied by either a 12V DC or 24V AC power supply. See Table A.1 for full camera specifications.

While all five cameras are the same, the lenses they are equipped with vary depending on the desired field of view. Overall, the goal in lens choice is to have a complete panoramic view from the northern to southern limit of the deployment area, with the view detailed enough to extract meaningful data. Therefore, each camera needs a lens with a focal length best suited to the section of the panorama it will be viewing.

For the center camera, hereafter referred to as Camera 3, which looks down at the coast with the shoreline nearly parallel to the horizontal axis of view, the lens used is a Computar H4514FICS-3. It has a focal length of 4.5mm, F1.4, and is intended for a 1/2" CCD. It is a CS-mount, takes a 30.5mm filter and has a horizontal field of view of 79.6°. The iris is manual, so the amount of light coming into the lens can be fixed to optimize image brightness for the data to be collected. It is fitted with a 30.5mm Sunpak PicturesPlus circular polarizer. This polarizing filter is necessary to minimize glare from the ocean surface. Without one equipped, the glare intensity from the ocean, especially when the sun is on the horizon, would wash out the image and render it useless for analysis. Tables A.2, A.3, and A.4 can be referenced for the full specifications for each of the three lenses.

Cameras 2 and 4 pan outward somewhat from the center view towards the north and south, respectively. Each are equipped with Computar H0612FICS-3 lenses. The focal length for this lens is 6mm, F1.4, and is intended for 1/2" format cameras like the Panasonics used in this study. It is a CS-mount, takes a 30.5mm filter and has a horizontal field of view of 58.3°. The shoreline area that these cameras focus on is further away than that of Camera 3, hence the longer focal length and narrower field of view. Cameras 2 and 4 are also equipped with the same 30.5mm Sunpak polarizing filter as Camera 3.

Cameras 1 and 5 pan north and south, respectively, from approximately where Cameras 2 and 4 lose focus to "infinity", beyond either end of the visible

Model	H4514FICS-3
Focal Length	4.5mm
Max. Aperture Ratio	1 : 1.4
Max. Image Format	6.4x4.8mm (ϕ 8mm)
Iris Operation	F1.4 16C
Focus Operation	0.2m Inf.
Object Dimension at M.O.D.	33.3x22.9cm
Format	1/2”
Angle of View D	100.1°
Angle of View H	79.6°
Angle of View V	59.5°
Effective Lens Aperture Front	ϕ 15.8mm
Effective Lens Aperture Rear	ϕ 9.0mm
Back Focal Length	8.7mm
Flange Back Length	12.5mm
Mount	CS-mount
Filter Size	M30.5x0.5
Dimensions	ϕ 34.5x33mm
Weight	36g

Table A.2: Specifications of Computar 4.5mm lens.

Model	H0614FICS-3
Focal Length	6.0mm
Max. Aperture Ratio	1 : 1.4
Max. Image Format	6.4x4.8mm (ϕ 8mm)
Iris Operation	F1.4 16C
Focus Operation	0.2m Inf.
Object Dimension at M.O.D.	22.3x16.3cm
Format	1/2”
Angle of View D	71.3°
Angle of View H	58.3°
Angle of View V	44.3°
Effective Lens Aperture Front	ϕ 10.8mm
Effective Lens Aperture Rear	ϕ 8.6mm
Back Focal Length	8.2mm
Flange Back Length	12.5mm
Mount	CS-mount
Filter Size	M30.5x0.5
Dimensions	ϕ 34.5x33mm
Weight	35g

Table A.3: Specifications of Computar 6.0mm lens.

Model	M8513
Focal Length	8.5mm
Max. Aperture Ratio	1 : 1.3
Max. Image Format	8.8x6.6mm (ϕ 11mm)
Iris Operation	F1.3 16C
Focus Operation	0.2m Inf.
Object Dimension at M.O.D.	18.6x13.6cm
Format	2/3"
Angle of View D	69.4°
Angle of View H	54.7°
Angle of View V	43.8°
Effective Lens Aperture Front	ϕ 20mm
Effective Lens Aperture Rear	ϕ 12mm
Back Focal Length	12.3mm
Flange Back Length	17.526mm
Mount	C-mount
Filter Size	N/A
Dimensions	ϕ 39x31.6mm
Weight	50g

Table A.4: Specifications of Computar 8.5mm lens.

shoreline. Therefore they need an even larger focal length and narrower field of view than that of Cameras 2 and 4. The lens used on both of these cameras is the Computar M8513. It has a focal length of 8.5mm, F1.3. It is intended for 2/3" format cameras, but a 2/3" lens will fit on a 1/2" camera. In general, the lens size can be larger than the camera size, but not the other way around. The lens is a C-mount, but fits the camera with a simple adaptor ring. The horizontal field of view is 54.7°. Unfortunately, the M8513 lens has no provision for a filter to be threaded on. A 40.5mm Tiffen circular polarizer is attached using a makeshift adapter ring fashioned from the lens cap. This way, the ring can be slid around the lens, placing the polarizing filter in front of it.

For surge protection at each camera's power terminals, a 1.5KESA15 tranzorb is installed between the +12V terminal and the ground terminal.

Each camera is installed in a Videotec HOV weatherproof housing equipped with an adjustable sun shield and a motorized wiper to keep the view glass free from debris and blemishes. The housing is 6.4x4.5x16.2in in size, weighs 4.6lb and is made of epoxy-coated die-cast aluminium. The operational temperature ranges from -4 to 122 °F. It stands atop a 12in column bracket that is bolted to a horizontal mounting surface. The bracket has two degrees of freedom for rotational adjustment and fine tuning of the camera view once mounted.

There is an independent pump and reservoir for each housing's wiper that holds about 1 gallon of cleaning solution. With the automatic wiper scheduling to be discussed later, this amounts to about 6 months of wiper usage between reservoir refills. Power and control lines for the wipers and pumps are routed to a wiper relay box, which controls the timing of pumps and wipers as well as reduces the number of wires coming into the computers. The wiper relay system is contained in a plastic weatherproof housing with the ability to be fastened to a surface. The pump is turned on for a specified amount of time before its associated wiper becomes active, and the wiper continues to operate for some time after the pump is shut off. In addition, the relay system ensures that only one wiper/pump combination is running at any one time. This configuration minimizes the amount of power consumed by the system due to view shield cleaning. The video and power cables from each camera travel directly from the housing to their respective computers without being routed through a relay, since they are operational at all times.

The computers that control this system are PCs contained in a heavy duty wall-mounted, weatherproof, lockable plastic housing. They are stripped down to a motherboard/processor, hard drive, and network card with provisions for a monitor, keyboard and mouse as well as USB devices. There is also a 12V lead-acid battery backup in case of power outages. The system itself is a VIA 667MHz embedded PC with 513Mb of PC133 RAM and a 40Gb hard disk drive. There are no CD

or floppy drives. The operating system is Microsoft Windows XP. On startup, the computer acts like a normal PC, but it has background hardware features that make the system suited for remote deployment. Each computer is equipped with an 8-bit microprocessor board called XPlor32. This functions as a power manager and watchdog for the system. It has a switch on the motherboard so it can be deactivated for system maintenance. When power is turned on, XPlor32 is powered up and it, in turn, powers on the Windows computer. Windows, upon startup, automatically starts Erdman's proprietary video software called VM95, to be discussed later. VM95 keeps XPlor32 aware of it's 'okay' status. If this signal is not received for over an hour, XPlor32 will do a cold restart of the system. There is also a 3-chip circuit that monitors XPlor32's functioning. If XPlor32 stops working, this circuit will restart the entire machine. Upon startup, Windows also launches a remote control program and connects to Erdman's Virtual Private Network, allowing system troubleshooting and maintenance to be performed remotely by either the end user or Erdman's staff. Each computer has four video input jacks. To distribute the workload between the two computers, one will be charged with handling three cameras and the other will control the remaining two.

For power, the computer is connected to a 24V AC transformer that is also enclosed in a durable, weatherproof, wall-mounted housing. For this deployment, the weatherproof housings aren't necessary since the computers and transformers are mounted indoors. Nevertheless, the heavy-duty casing makes for worry-free protection. Finally, the transformers are connected to an APC power supply / battery backup. This is for protection against surges and power outages, and takes the load off the small 12V battery inside the computer. A hardwired internet connection is maintained, with the two computers linked to a router behind a single cable modem. See Figure A.4 for a simplified wiring diagram, showing only one camera and computer.

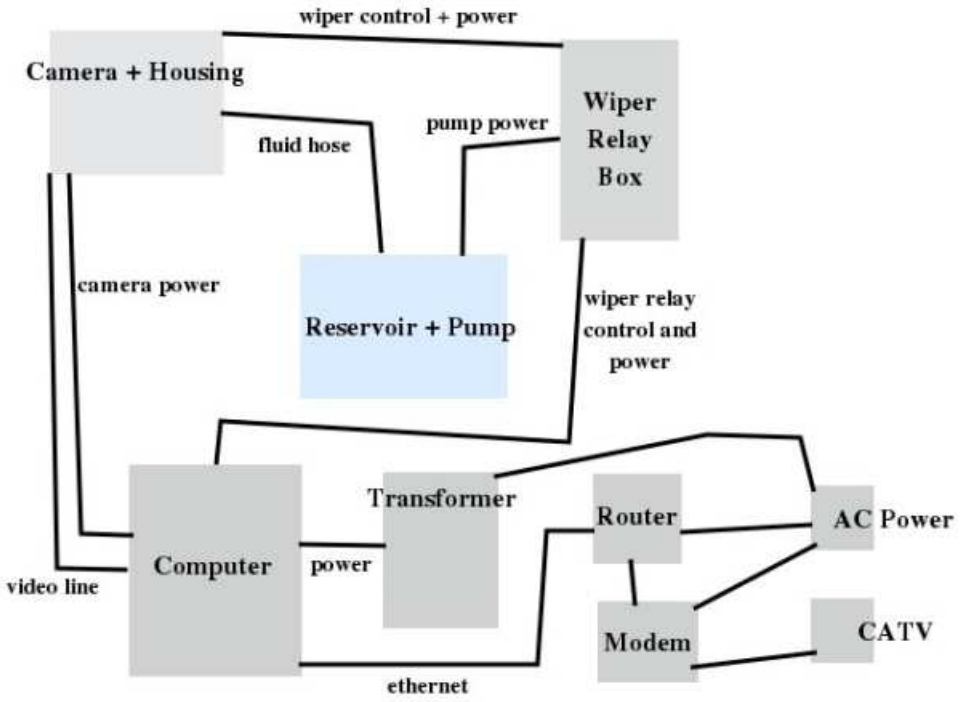


Figure A.4: Connection diagram for the system, using only a single camera and computer for simplicity.

A.3 Deployment Setup

The rooftop of the Brandywine building where the camera system is installed has a 1-on-4 slope. The roof is completely covered in ridged sheet metal with alternating 9" flat sections and 3" ridges. The ridges run parallel to the sloped axis of the roof. In addition, there is an exhaust stack protruding in front of the apex of the roof on the shoreward side, which could obstruct the view of Camera 3. In order to mount the cameras securely and achieve the desired panoramic view, first, a level platform is needed that the camera housing columns can be bolted to. Second, the platform must also have room to mount the wiper relay box and all 5 wiper fluid reservoirs, which are housed in a metal cage. Lastly, there must be a way to get around the exhaust stack and see a full panoramic view with the five cameras.

Two platforms were constructed so that the camera array can be spread out and thus avoid being obstructed by the exhaust stack. They have legs of different lengths so that the platform is level with the horizontal when attached to the sloping roof. The platforms are identical in size and construction; they are metal frames made of 1" aluminium angle iron by A&H Metals in Newark, DE. The main body is rectangular, 24" in width, 36" in length and 12" in height, with the length dimension being parallel to the axis of the slope of the roof. At each corner is a leg, with the legs to be mounted on the "high side" of the rooftop being 3" tall and the legs on the "low side" being 12" so that the top of the frame is level when mounted on the roof. Each leg has an 8x8x.25" aluminium plate "foot" attached at an angle so that it lies flush with the rooftop in the 9" flat sections of the roof. This allows for the frame to be bolted directly onto the roof.

The aluminium frame is lighweight and corrosion-proof, and having so solid sides means it has minimal wind resistance. The top is covered with a 24" by 36" by 1/2" sheet of marine-grade plywood, which is coated with primer and then a waterproof epoxy polymer paint. The paint creates a hard, plastic-like covering

when dried. This combination of wood and paint makes for a superbly weather-resistant platform. It is attached to the frame with stainless steel self-tapping screws. Figures A.5, A.6, and A.7 show multiple views of one of the mountng platforms.

Figure A.5: Side view of mounting frame.

To attach the frames to the rooftop, we enlisted the expertise of Potteiger-Raintree, Inc., the roofing contractor for the Sea Colony resort. They installed footpads underneath the locations of the feet of the two mounting platforms and secured the platforms to the roof with screws. The roof-platform interface is waterproofed with a caulk sealant. The northernmost platform is located close to the corner of the rooftop facing the shore. The other platform is the same distance from the southeast side of the building as the first, but about 8 feet further away from the northeast side. This spreads the camera array such that the exhaust stack at the corner doesn't obstruct the panoramic view.

Just like the division of load between the two computers, the platforms split the load as evenly as possible. The northernmost platform houses Cameras 1, 2,

Figure A.6: Oblique view of mounting frame.

Figure A.7: Close view of the mounting frame footpad.

and 3 as well as their associated wiper reservoirs, while the southern platform holds Cameras 4 and 5, their wiper reservoirs, and the wiper relay box. The camera housing brackets are attached to the plywood surface with four 1/4" stainless steel bolts, nuts and lock washers. Camera 1 is bolted towards the eastern side of its platform so it can face to the north. Camera 2 is attached near the northeast corner of the platform, and Camera 3 is attached close to the southeast corner so it can look directly down at the shoreline. On the southern platform, Cameras 4 and 5 are mounted next to each other on the southeastern edge. The wiper relay box is situated on top of this platform, held in place by four stainless steel self-tapping screws. The wiper reservoir/pump units are mounted to the corners of their respective frames with stainless steel hose clamps. The reservoir sits in a metal cage, and the frames have no solid sides, so it is easy to run a hose clamp through the cage, around the frame and tighten it down. Two clamps are on the vertical member of the corner of the frame, above the bottom-most horizontal members. A third clamp holds the cage to this horizontal member.

With the outside hardware in place, the wiring can be connected and run along the roof to the wall of the elevator control room. The cabling for video and camera power run directly from the cameras to the computers. The lines for the three cameras on the northern platform are bundled together with zip ties and run directly to the roof adjacent to the southern platform, where they are bundled again with the power and video lines of the remaining two cameras and run along the roof straight to the wall of the elevator control room. Along its route, the bundle is periodically fastened to the roof by means of a plastic loop that surrounds the bundle and has a tab that is screwed into the metal surface.

Next, the tubing that supplies the wipers with cleaning solution are led from the fluid reservoir and attached to their respective wipers and the reservoirs are filled with water and cleaner. Now, the wiper and pump wires must be run to

the relay box. For proper function and timing, it is imperative that the wires for each wiper are paired with their respective pump wires. Once each wiper/pump combination is matched up, the wiper/pump wires from Cameras 1, 2 and 3 are led along the roof towards the southern platform in the same manner as the video cabling. From there they are fed into the wiper relay box and each lead is attached to its proper terminal. There is a specific set of terminals for each pump and wiper. The pump requires two leads for 12V power. The wiper requires three; two for 12V power and one extra so that controls the wiper motion and allow it to return to the rest position once its action is finished. The wiper/pump lines from the southern platform are run directly into the relay box without the need to attach to the roof. To maintain a clean installation, all wires are measured and cut so there is as little slack as possible after everything is connected. With the relay box wired, the lines exiting the box toward the computers are led along the roof and bundled with the camera power and video lines already run to the wall of the elevator control room.

Now, the bundle of cabling must have a way to get into the control room. The wall here is plaster with a stone veneer on the outside. Inside the control room, there is a tiled drop ceiling. To run the cables through the outside wall and up to the drop ceiling, a 5" diameter hole is drilled through the wall and met on the inside by 4" metal conduit that runs from the hole up into the ceiling. The hole in the wall is sealed with silicone caulk. With a string attached, the bundle of cables is pulled through the hole and up the conduit to the ceiling. The bundle can then be dragged along the frame of the drop ceiling by alternately removing tiles until it reaches the opposite wall where the computers are to be located. Then it is run through another conduit back down into the room, waiting to be connected to the computers.

To take up the least amount of space as possible, wall-mounting the computers and transformers is the best technique. This wall is plaster with metal studs,

and the spacing of the mounting holes for the computers and transformers is not standard 16" centers. To make mounting easier, a 4' by 3' sheet of 1/2" plywood is fastened to the studs on the wall. Then the computers and tranformers can be mounted solidly onto this without having to worry about securing to wall studs. The are attached at all four corners with wood screws and washers. The computers are mounted above their respective transformers on the plywood board. The APC power supply sits on the ground, while the cable modem and ethernet router rest on top of the computers. Electrical and cable TV outlets were run from an electrical control room one floor below the elevator control room and terminate in four electrical plugs and a cable attachment on the wall just above the plywood sheet. With all the hardware in place, the cables from outside can be attached to their respective computers, which have a slot for the plug coming from the wiper relay box. With this attached, the 12V power lines from the cameras are fastened in their terminals on this plug. The video lines are attached via RCA jacks on the computer board. Ethernet cables connect the computers to the router and router to modem. Power cables from the two transformers, the modem and the router are run through the power supply and then the wall outlet. The system is now ready and software setup and camera alignment can begin. See Figure 2.2 for a picture of the complete outdoor portion of the video observation system, post-installation.

A.4 Software

Before starting up the computers, the watchdog switch is turned off so setup can be peformed without the computer automatically rebooting. Upon startup, VM95 is automatically launched and the user switches it to manual mode. The first order of business is to set up the video lines and assign a line to each camera. Each computer has a possible 4 lines, and there are five cameras split 3 and 3 between the two computers. On computer #1, line 1 is assigned to Camera 1, line 2 to Camera 2, and line 3 to Camera 3. On computer #2, line 1 is set for Camera 4 and line

2 to Camera 5. This is also the arrangement the cameras respective lines must be plugged into, from left to right, in the row of RCA video jacks on the motherboard. All lines are set to record at the maximum resolution of 640x480 pixels. Figure A.8 is an example screenshot of VM95 in action.

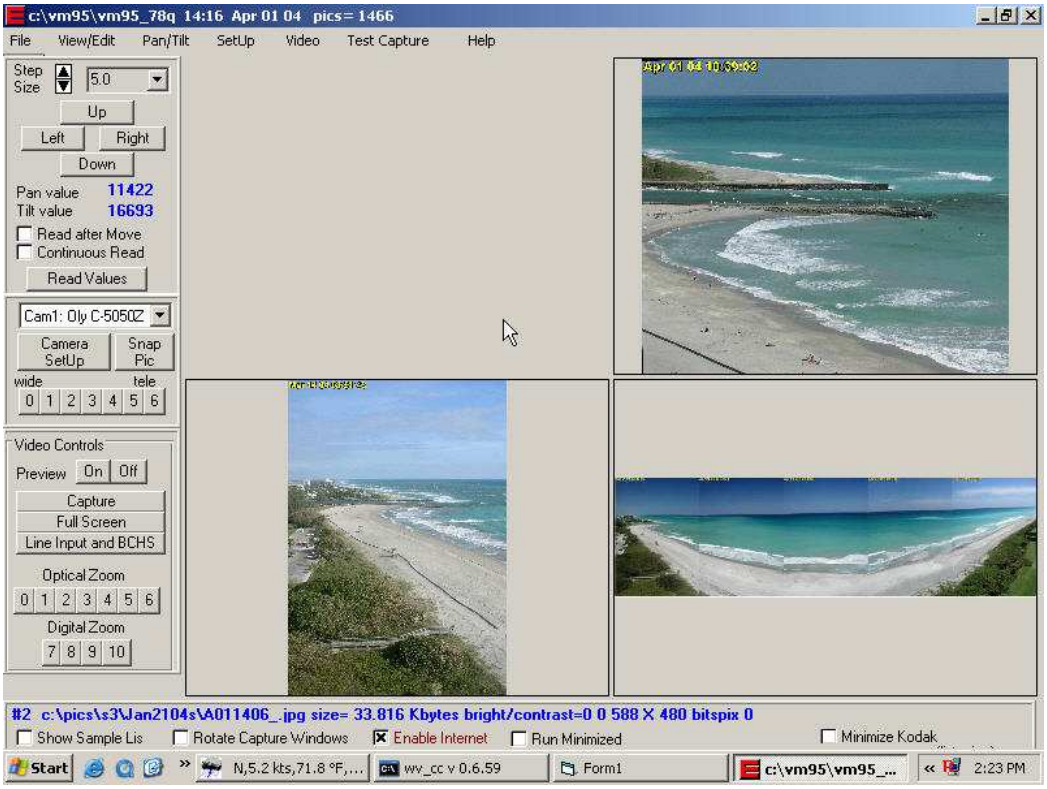


Figure A.8: Example screenshot of VM95. Image courtesy of Erdman Video Systems.

Now that the video lines are assigned, VM95 can view the video stream from any of the cameras. This allows us to rotate the cameras into their final position before the column bracket bolts are secured and the camera view is fixed. Camera 3 looks down at the beach and views from landward of the dune to the horizon. Cameras 2 and 4 look more northward and southward, respectively, of this view, but have some overlap with the view of Camera 3. This will become useful later when it is time to survey the beach, and it also makes a panoramic view easier to

compile since it assures that no part of the beach is left out. In the same manner, Cameras 1 and 5 are pointed from the outside edges of the view of Cameras 2 and 4 to infinity. As each view is established, the rotation bolts on the housing bracket are secured so the view does not change.

With the fixed camera views in place, the image sampling schedule can be set up. The VM95 scheduler operates using a series of instances called scenes. In automatic mode, the program runs through these scenes and performs the capture or action that each one specifies. In each scene, the video line to use and the type of capture can be specified, as well as the daily start and stop times to perform the scene and the interval on which it is repeated. Also, the file size and naming scheme can be specified. Aside from capturing images and video, scenes can be assigned to perform such tasks as uploading the latest captures and rebooting the computer. For all five cameras, three scenes each are set up for three types of image captures. All are set to run once every hour during approximate daylight hours, from 11:30GMT to 00:30GMT (6:30AM EST to 7:30PM EST). File size is set at 80 percent quality for a jpeg capture. All image captures are named in the standard ARGUS filename format. This format looks like EEEEEEEEEE.WWW.MMM.DD_HH_MM_SS.gmt.YYYY.SSSS.CC.TTTT.jpg, where E is the 10-digit epoch time, W is the day of the week, M is the month, D is the day number, HH_MM_SS is the time in GMT, Y is the year, S is the site name ('beth' for this study), C is the camera number (ex. c1 for camera 1), and T is the image type ('snap' for snapshot, 'var' for variance image, 'timex' for time exposure). The ARGUS filename formatting makes the task of identifying and archiving images very simple by allowing the filename to be parsed for whatever attribute is needed.

The first type of capture is a snapshot, which is simply a capture of the raw image from the camera. The next type is the time exposure, where the system takes multiple snapshots at 5 frames per second for one minute, and averages the

intensities. In this type of image, oftentimes features such as the mean shoreline and the edge of the offshore sandbar can be identified. Finally there is the variance image, which is similar to the time exposure but where regions of high intensity indicate high variability. For example, the swash zone would have a high intensity since it is in constant motion, whereas a sunbather on the beach would have low intensity because they do not move much. Figure A.9 is an example snapshot, taken from camera 4. Figure A.10 is a time-exposure image from camera 4. Note the smoothing of the ocean surface and surf zone turbulence compoared to the previous snapshot. The edge of the surf zone is clearly visible. Figure A.11 is an example variance image. Here, the surf zone is even more clearly defined by the high variance bright region compared to the much darker ocean and beach face. Also note the areas of high intensity in the lower left region of the image where people are presumably moving on the beach.

In addition to these image captures, the system is set up to capture video streams from Camera 3. The timing start, stop and frequency are the same as for the still captures. Capture resolution is 480x360 pixels. The video is saved as a Microsoft AVI, and is 5 seconds long with a framerate of 5fps. This format is less than ideal, being that AVIs are very large files compared to something like a WMV. Unfortunately, video processing will be performed on a 64-bit Linux machine, and there are no WMV codecs available for this architecture. Before this was realized, the system was capturing 47 second WMVs at 10fps, and these videos have been saved and will be used in the later analysis on a 32-bit computer.

Several more scenes are also implemented. One is an FTP scene, where the system transfers the latest image captures to an FTP site on campus. This is the SANDCAM machine that handles image archiving and data analysis for this study and its sister site in Rehoboth Beach. The FTP information can be set up in a separate options menu within VM95. Finally a scene is created that reboots the

Figure A.9: Example snapshot image, from camera 4.

Figure A.10: Example time exposure image, from camera 4.

Figure A.11: Example variance image, from camera 4.

computer once per evening when the captures are done for the day. Scenes are numbered and subsequently executed by VM95 in sequential order, so that images and video are captured before the FTP process is run.

Once scheduling is complete, FTP site parameters are input and the pump to wiper delay and schedule can be assigned. After this, setup at the site is complete. The watchdog switch is turned back on and the system is rebooted into VM95's automatic mode, and image capturing begins. Now focus can be set on what happens after the images are transferred to the University of Delaware campus.

Appendix B

SETUP WORK

In order to turn the captured imagery from simply pictures of the beach into useful data, several preparatory analyses must be completed.

B.1 Lens Calibration

Before system deployment, each camera lens must be calibrated to determine its distortion parameters in order to be able to correct the image before analysis can take place. Lenses are never perfectly formed and have inherent imperfections that necessitate correction. There is also distortion around the edges of the images, especially in the smaller focal length lenses, that leads to the 'fish-eye' effect. To perform a meaningful analysis with the images, these distortions must be accounted for. The calibration need only be performed once for each lens to determine its distortion characteristics. The process for calibration is simple. The camera and lens pair is connected to a computer with video capture software running. The type of software doesn't matter; here we use the VM95 program.

With the camera on a tripod, the image is focused on a paper printout of an array of white dots on a black background that is taped to the wall. The setup here is critical to produce accurate distortion numbers. The dot paper must be square with the floor, as well as flat as possible against the wall. Ripples and bubbles in the paper can affect the distortion calibration. With the paper set up properly, the tripod must also be aligned carefully. It needs to be close enough to the paper so that as little of the wall is visible in the image while still seeing every bit of

each dot. Clearly, the distance between the lens and wall varies depending on the focal length of the camera. For the 4.5mm lens, the fish-eye effect is prominent and it is impossible to not have any wall area showing in the image. This problem will be addressed later. The image must be focused as clearly as possible to get proper dot definition. Just as the dot picture is square with the floor, it must be perfectly square within the image view as well. Lighting is also extremely important. Typical flourescent bulbs flicker at a frequency that makes for poor image lighting when the program takes a snapshot. To remedy this, all the flourescent lights in the vicinity are turned off and a tripod array of halogen bulbs are used. They are arranged so lighting is clear and even, with no bright glare spots as well as no dark areas on the dot array. Since 'good' lighting and focus can be sometimes difficult to discern with the naked eye, a multitude of snapshots are taken and saved with different lighting and focus scenarios to be certain that the best possible snapshot is taken. All snapshots are taken at the best possible resolution of 640x480 pixels. Figures B.1, B.2, and B.3 show example dot images from lenses of all three focal lengths used. In Figure 2.1, which shows the 4.5mm lens, note the significant fish-eye effect and the extra white space around the perimeter. These problems are much less evident in the next figures of the larger focal length lenses. In all three, one can see the sharp focus and attempt to have even lighting across the image.

Once a satisfactory image is obtained, a MATLAB routine developed by the ARGUS group at Oregon State University is used for analysis. Out of the many dot pictures previously taken, one is chosen that looks well lit and focused. Its name is supplied to the initial routine called GETDOTS along with the total number of dots in the image. This program, as its name suggests, looks for the areas in the image with high intensity (the white dots) and separates them from the low intensity black areas. It does this by searching for intensity values above a certain threshold value, usually taken to be 125 on a scale of 0 to 255. It then transforms the image into one

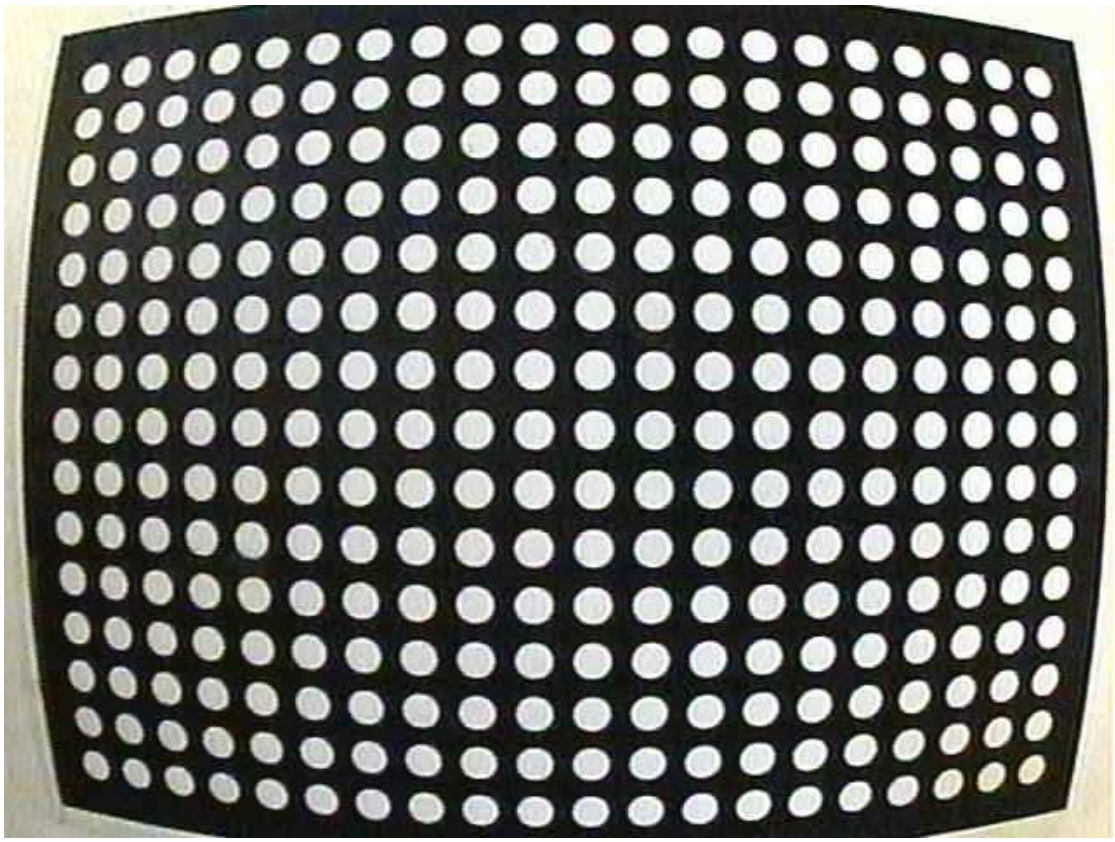


Figure B.1: Example dot image from the 4.5mm lens.

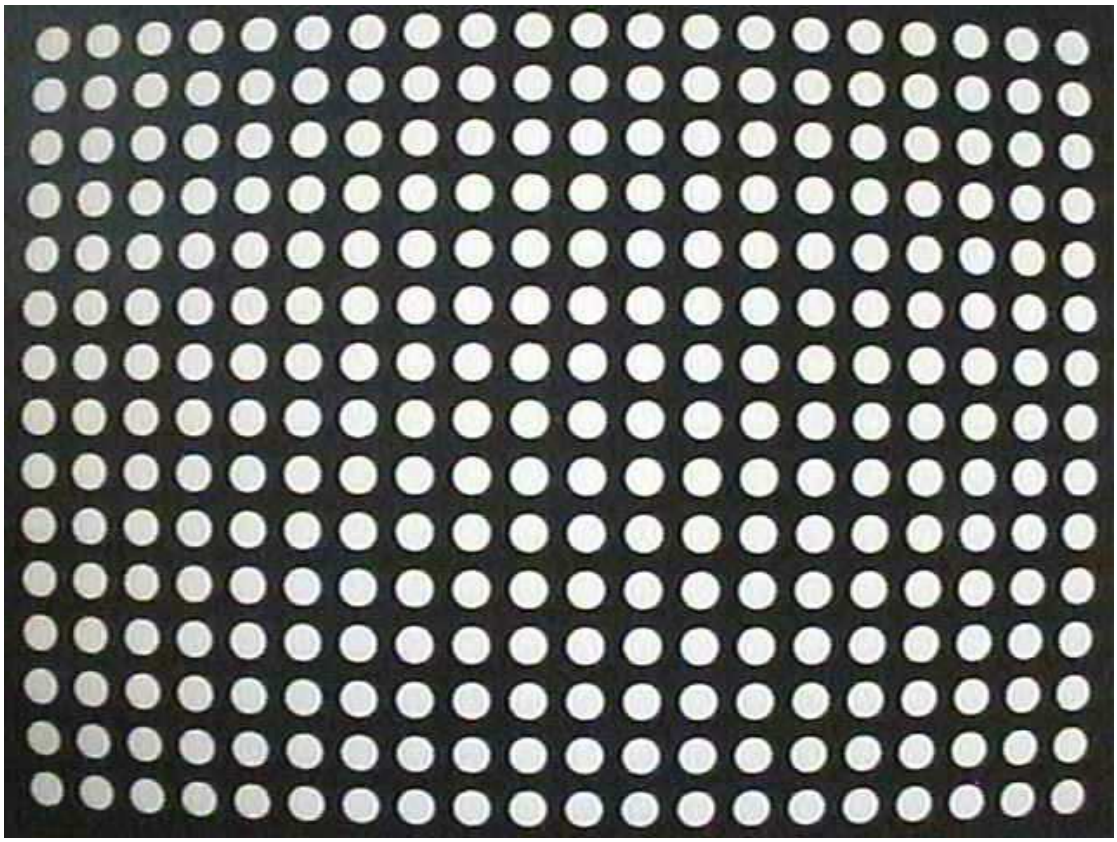


Figure B.2: Example dot image from a 6mm lens.

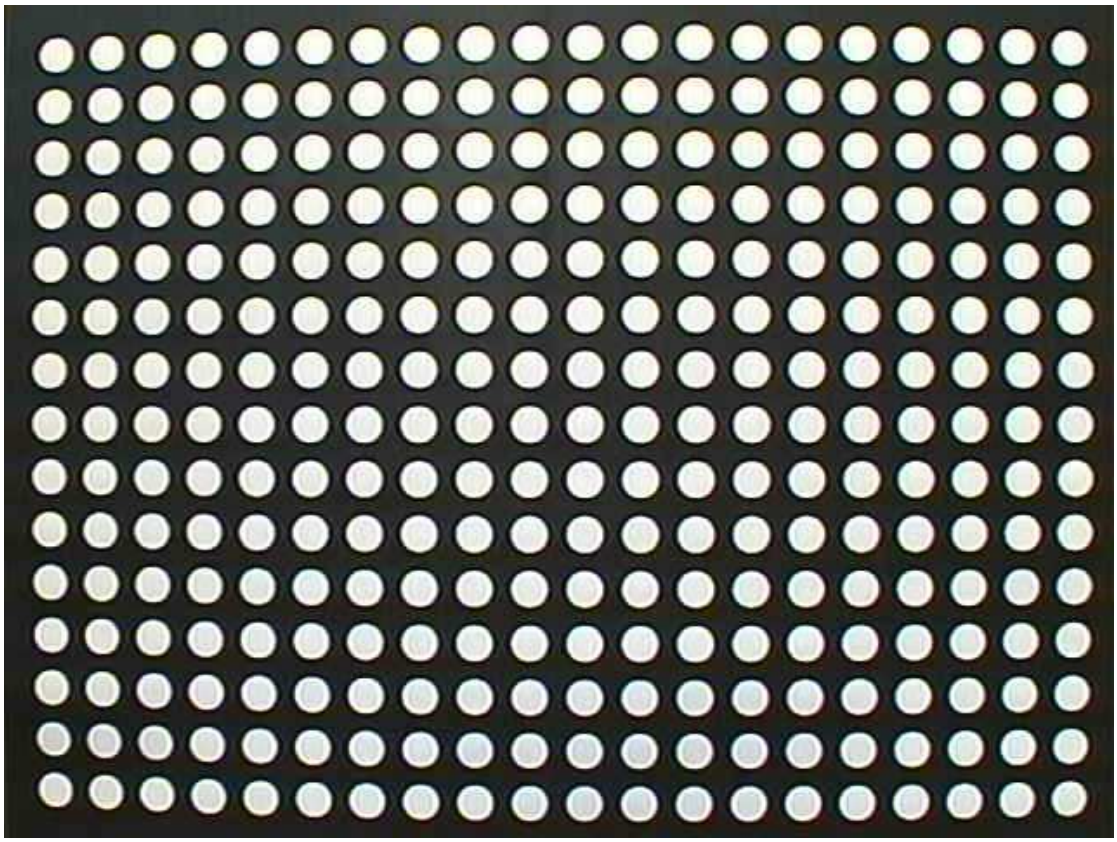


Figure B.3: Example dot image from an 8.5mm lens.

with values of 255 at the locations of the dots and zero elsewhere, leaving a clearly defined dot image of 'all or nothing' intensity. It also crops the extra black space, if any, that surrounds the box which contains the dots. Finally, the coordinates of the centers of each dot is calculated and returned in a matrix. This is where the fish-eye problem can rear its head. If there is significant white space around the dot paper, a situation which is unavoidable with the 4.5mm lens, these areas can be confused as being another 'dot'. Thus, the coordinate matrix contains more dot centers than there are dots in the image. Before using these coordinates for calibration, one must manually examine the coordinates and remove those that correspond to the wall areas and not dots.

Once cropped, this coordinate matrix is supplied to the program INTCAL, which performs the actual calibration algorithms. This program also needs the number of rows and columns of dots in the image and the image width in pixels. It first calculates the roll angle of the dot matrix, which is a check to see how well the dot array is squared within the image. If the magnitude of this value is more than 0.1 radians, the image should not be used for calibration. If it is less, the array is sufficiently aligned. INTCAL first displays an graph of the calculated dot positions from GETDOTS compared to where the dot centers 'should' be in a properly undistorted array. Then, a distortion solution is run using the distorted dots using a Walton solution and errors are examined. The analysis is run again using undistorted dots and the results are shown. If the calibration is 'good', then the difference vector plot should have vectors radiating outwards from center. The pixel error points should follow the correlation curve closely. The plot of error after correction may look messy, but this is due to the errors likely being small so the vector scale looks odd in the plot. There are six important parameters that need to be saved in some manner here for later use. The easiest way is to store them in a separate m-file, each parameter for each lens in its own array. This way, multiple

calibration runs are taken for each lens, and the values from each run can be averaged together to get a single value for each parameter for each lens.

The first parameters are the two distortion coefficients D1 and D2. Next are U_0 and V_0, which are, respectively, the actual horizontal and vertical pixel centers of the lens view. Finally are the values lx and ly, which are fixed at 1 and -1 for most analyses including this one. After these values are recorded, the routine continues and histograms are shown of the pixel difference between the distorted and undistorted dot patterns before and after correction. The errors after correction should be all close to zero pixels, although there can be a high count in that range. The distortion correction, of course, cannot be absolutely perfect, but should take care of all major errors. Finally, the corrected dot centers are plotted with the perfect array of dots. This time, they should line up nearly perfectly with each other, unlike the large errors seen before correction. In highly distorted lenses such as the 4.5mm, dot centers near the four corners of the image probably will still be visibly misaligned, but this is practically unavoidable. The vast majority of the pixels, however, will be properly corrected for.

After performing this analysis using multiple runs for each lens, we have a data set of distortion parameters and lens centers that are averaged together to get characteristic values unique to each lens. These values are saved and will be used later to process the images.

Figures B.4 to B.21 show examples of the output of GETDOTS and INTCAL for one run of each size lens. In Figure B.4, note the white space around the edge due to distortion and the false centers the routine predicted for them. These centers were, of course, removed before passing to INTCAL. These false centers don't appear in Figures B.10 and B.16. In Figure B.5, the high distortion at the lens corners is clear to see where the actual dot positions are quite far from where a 'correct' dot pattern should be. The distortion is much greater than that seen in Figures B.11

and B.17 of the larger focal length lenses.

In Figures B.7, B.13, and B.19, note the smooth, outwardly radiating difference vectors and the tight fit with the correlation line indicating a good image capture and successful distortion correction. For the 4.5mm lens, the actual lens center is several pixels away from where it ought to be, but is still close. The high distortion of smaller focal length lenses causes these errors.

The before/after error histograms in Figures B.8, B.14, and B.20 show that the major distortion errors are accounted for by INTCAL. For the 4.5mm lens, there are still some multi-pixel errors, but these are located at the highly distorted corners of the view. For the other lenses, errors remain clustered close to zero pixels. These errors are better spatially visualized in Figures B.9, B.15, and B.21, which show the corrected dot centers.

To reiterate, these are just example runs for each of the lenses. Many runs were made with many images and the results, although not too variant, were averaged together. The final results are displayed in the analysis section.

B.2 Surveying Ground Control Points

After correcting for lens distortion, the next step towards extracting useful data from the imagery is to rectify the images. This is a process of transforming the image into new coordinates. As it stands, the cameras shoot snapshots of the shoreline from an oblique angle. Therefore, movement on the beach face and in the surf zone is difficult to capture because particles are moving into and out of the image as well as up and down across it. So there are 3 dimensions of movement, but only a 2D image. In order to measure movement, we need to reduce the image to 2 dimensions so the degrees of freedom for a moving particle is the same as the number of dimensions in the image. The image must be transformed so that its horizontal and vertical coordinates are coplanar with the beach face and ocean surface.

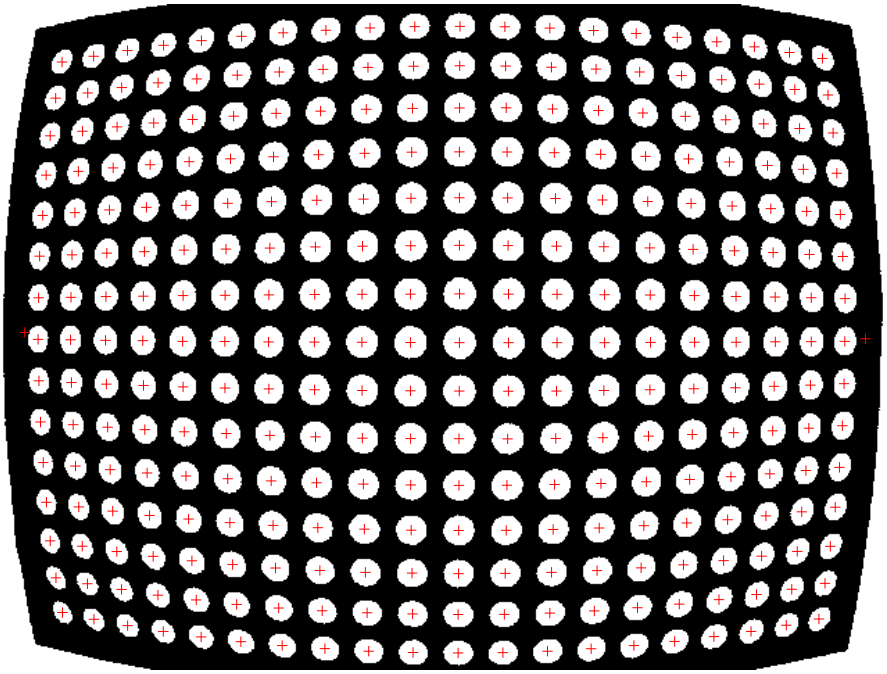


Figure B.4: Dot centers calculated by GETDOTS for the 4.5mm lens, run 4.

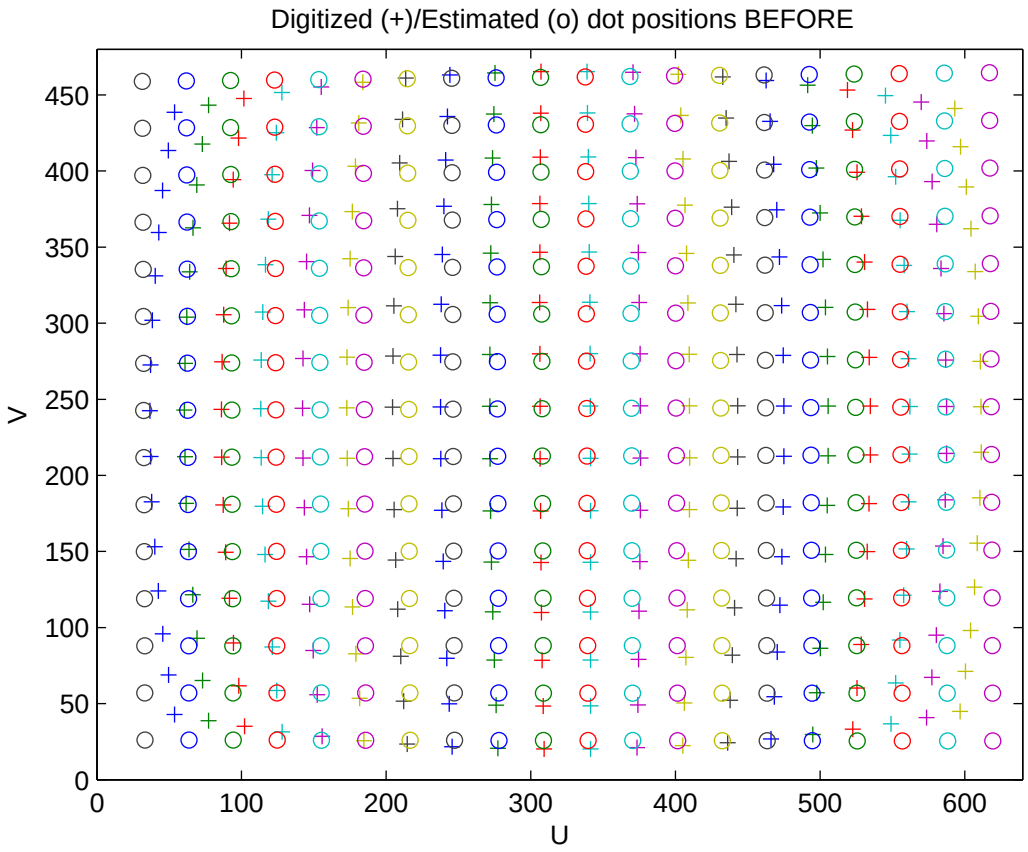


Figure B.5: Comparison from INTCAL between actual and theoretical dots before distortion correction for the 4.5mm lens, run 4.

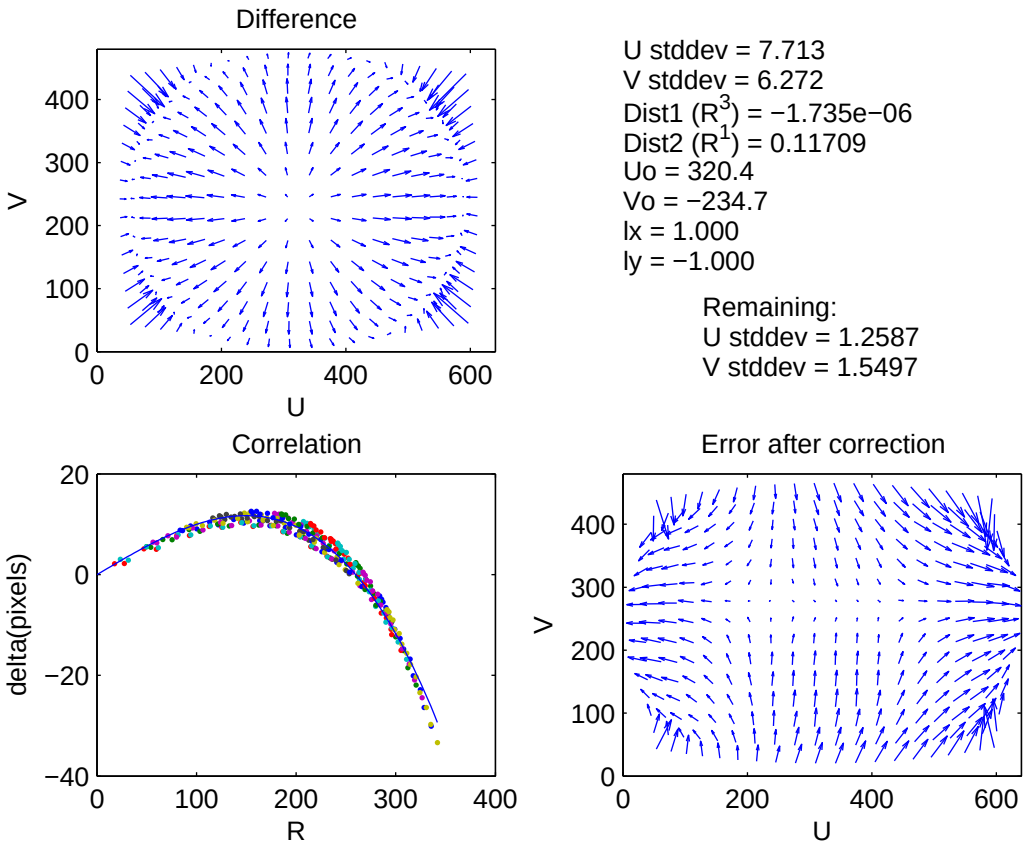


Figure B.6: Walton solution parameters using distorted image for the 4.5mm lens, run 4.

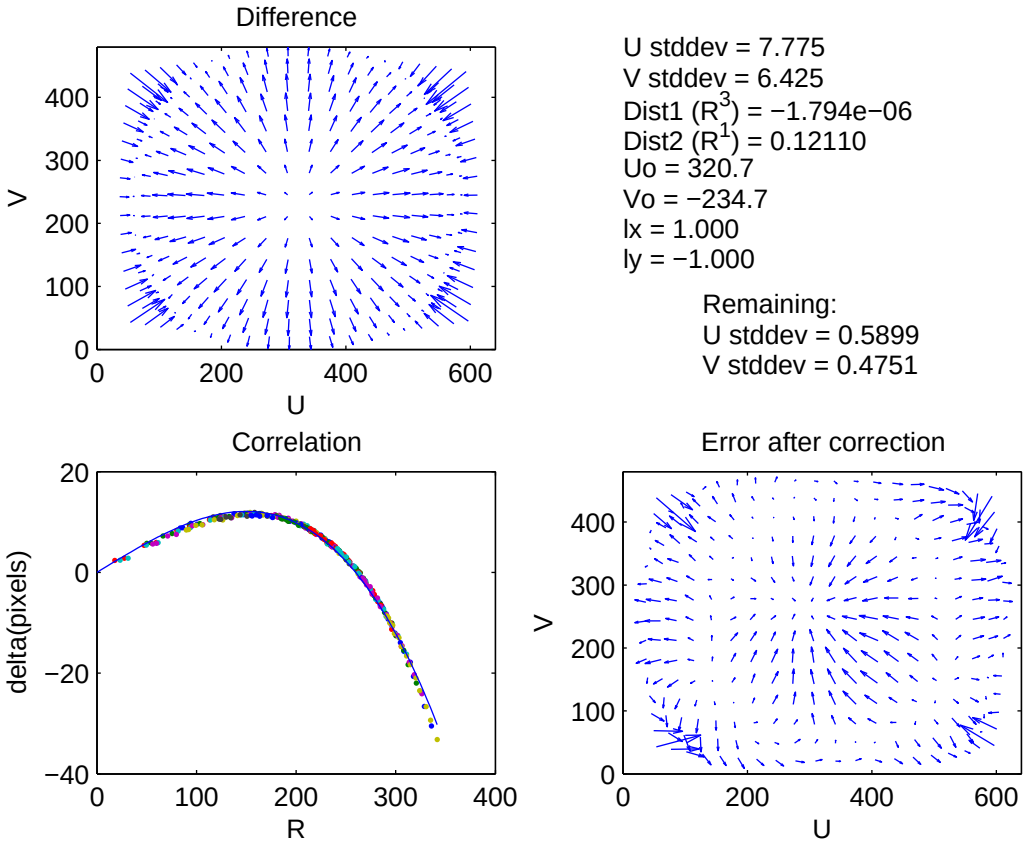


Figure B.7: Walton solution parameters using undistorted image for the 4.5mm lens, run 4. These are the numbers that are saved for use in rectification.

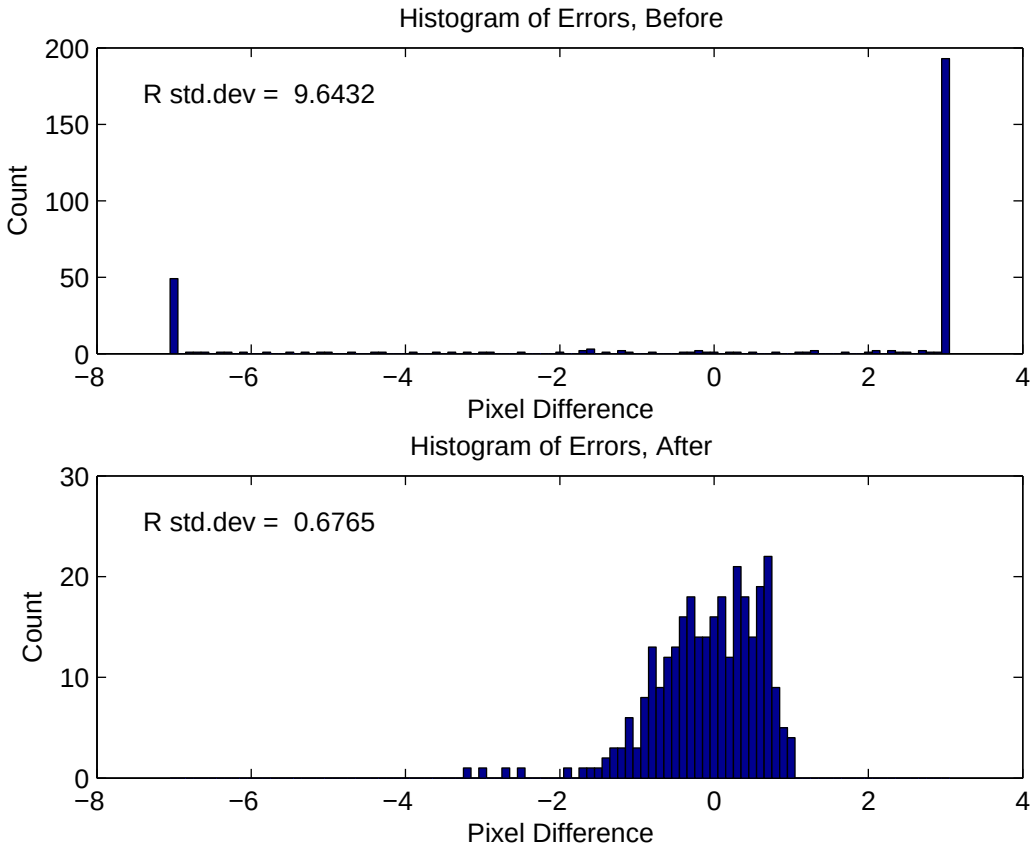


Figure B.8: Histogram of errors before and after correction for the 4.5mm lens, run 4.

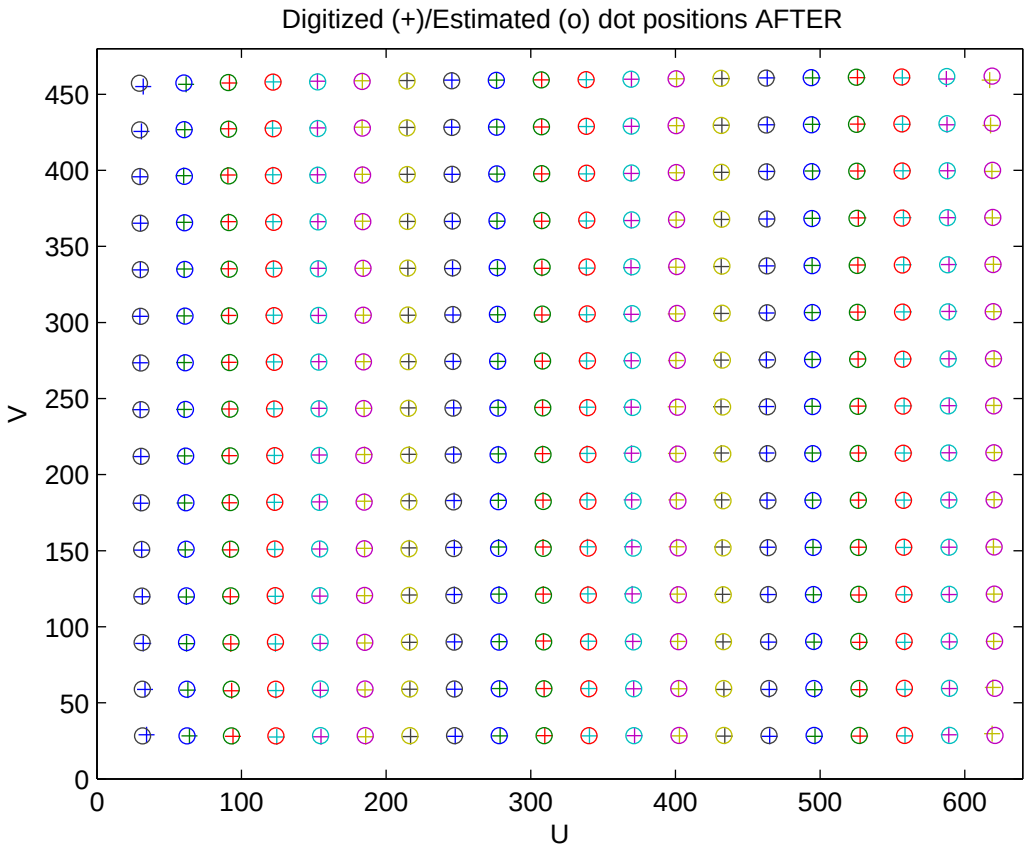


Figure B.9: Comparison between actual and theoretical dots after distortion correction for the 4.5mm lens, run 4.

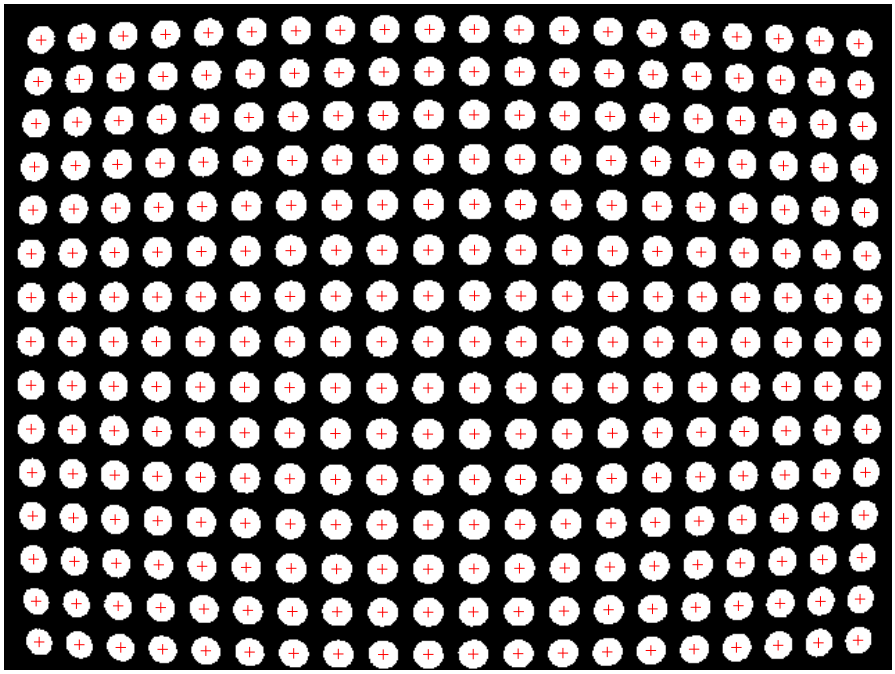


Figure B.10: Dot centers calculated by GETDOTS for 6mm lens #1, run 15.

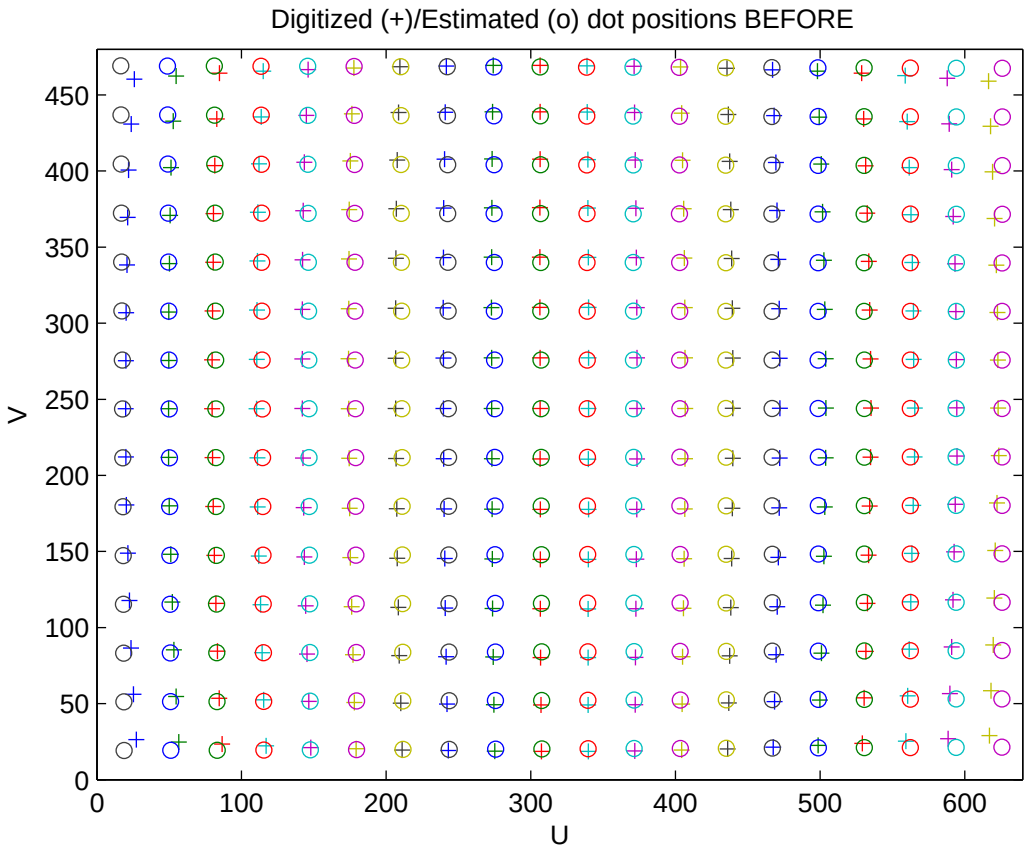


Figure B.11: Comparison from INTCAL between actual and theoretical dots before distortion correction for the 6mm lens #1, run 15.

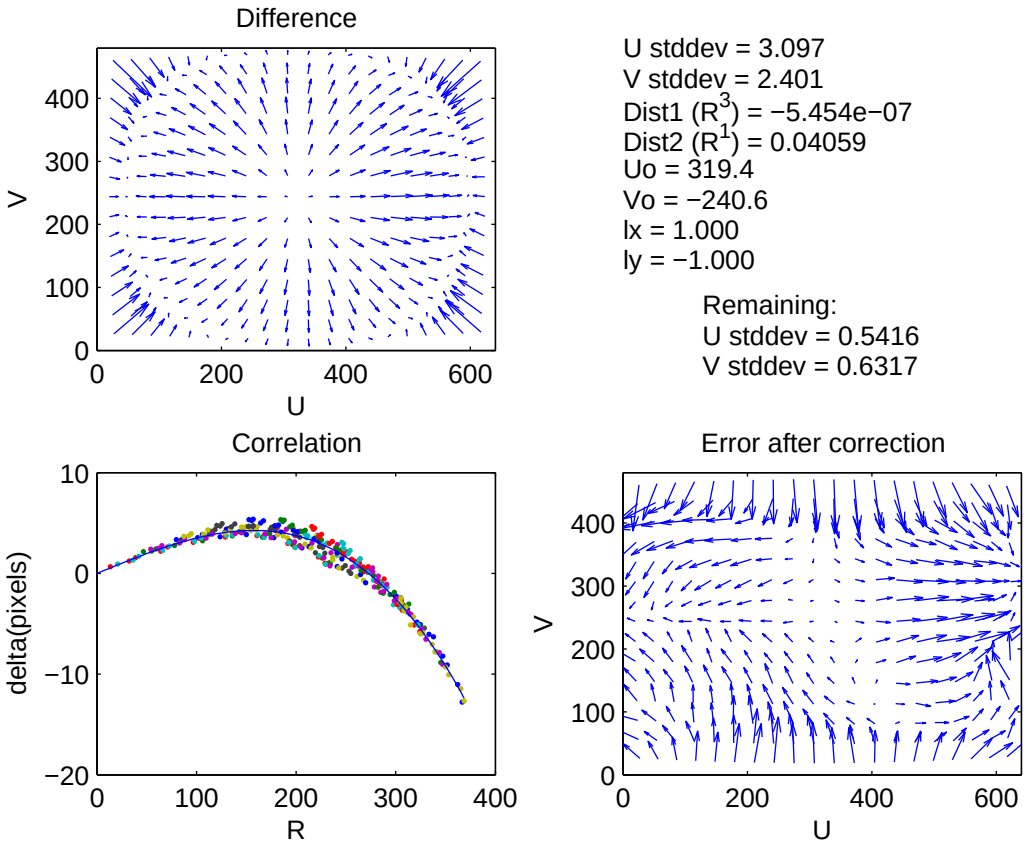


Figure B.12: Walton solution parameters using distorted image for the 6mm lens #1, run 15.

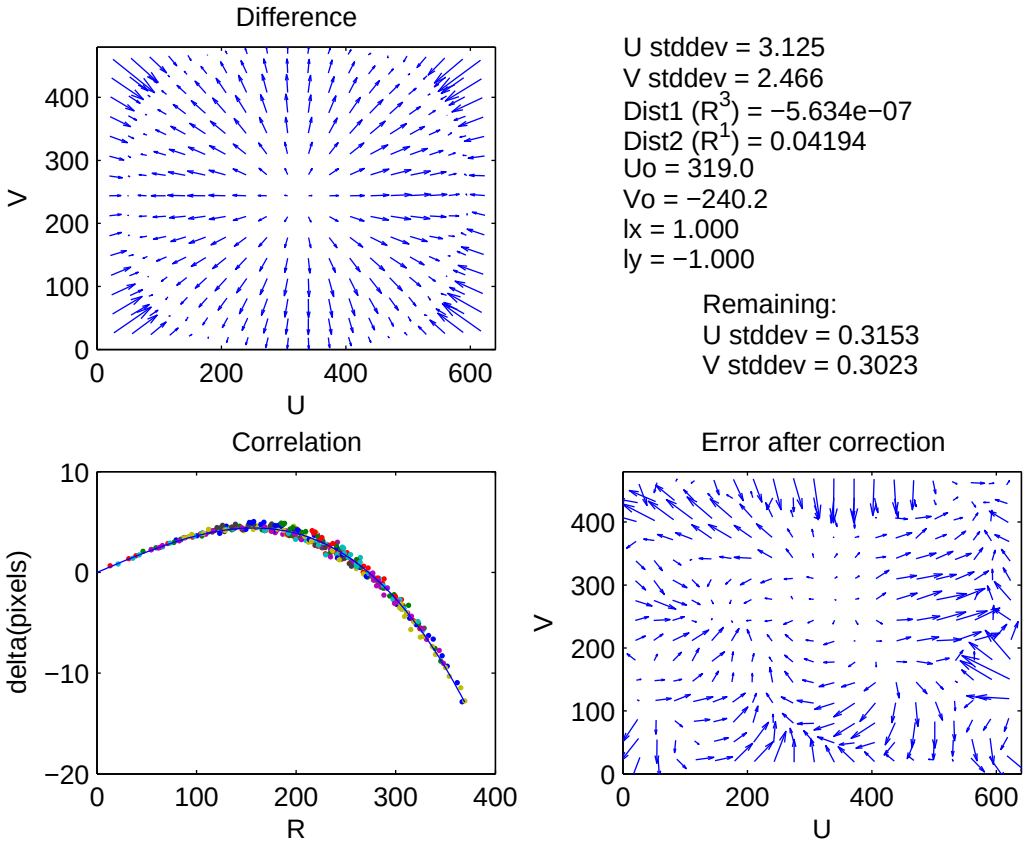


Figure B.13: Walton solution parameters using undistorted image for the 6mm lens #1, run 15. These are the numbers that are saved for use in rectification.

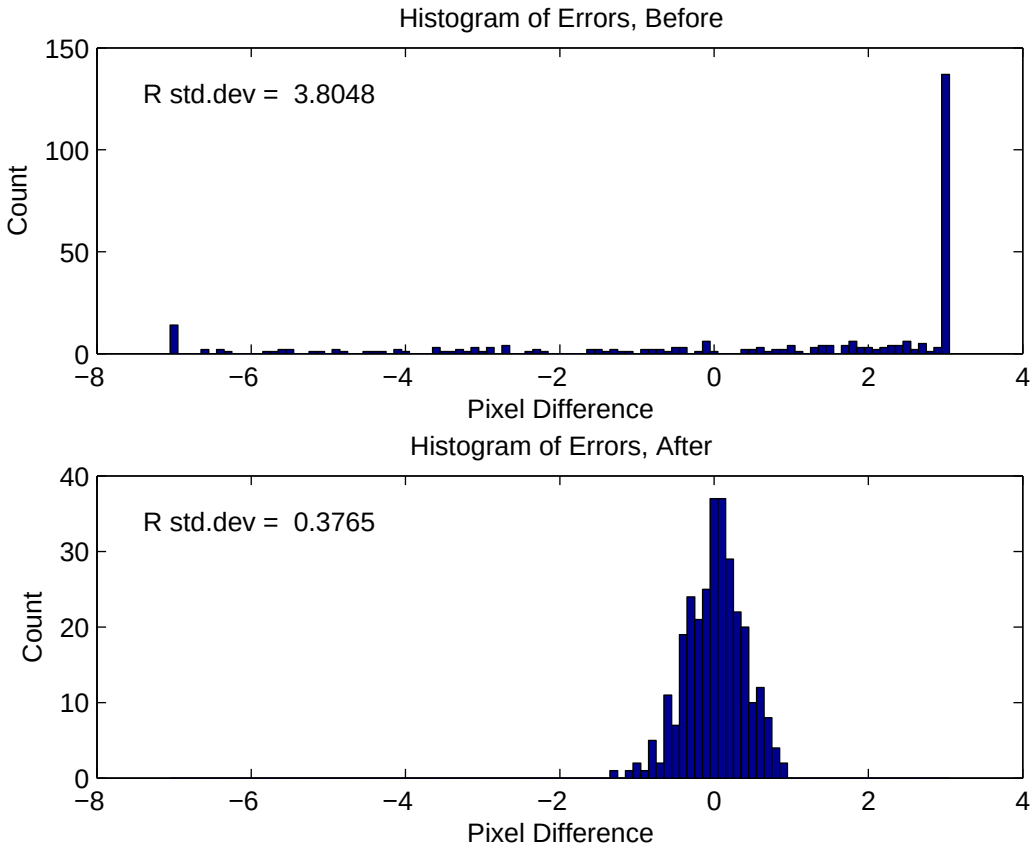


Figure B.14: Histogram of errors before and after correction for the 6mm lens #1, run 15.

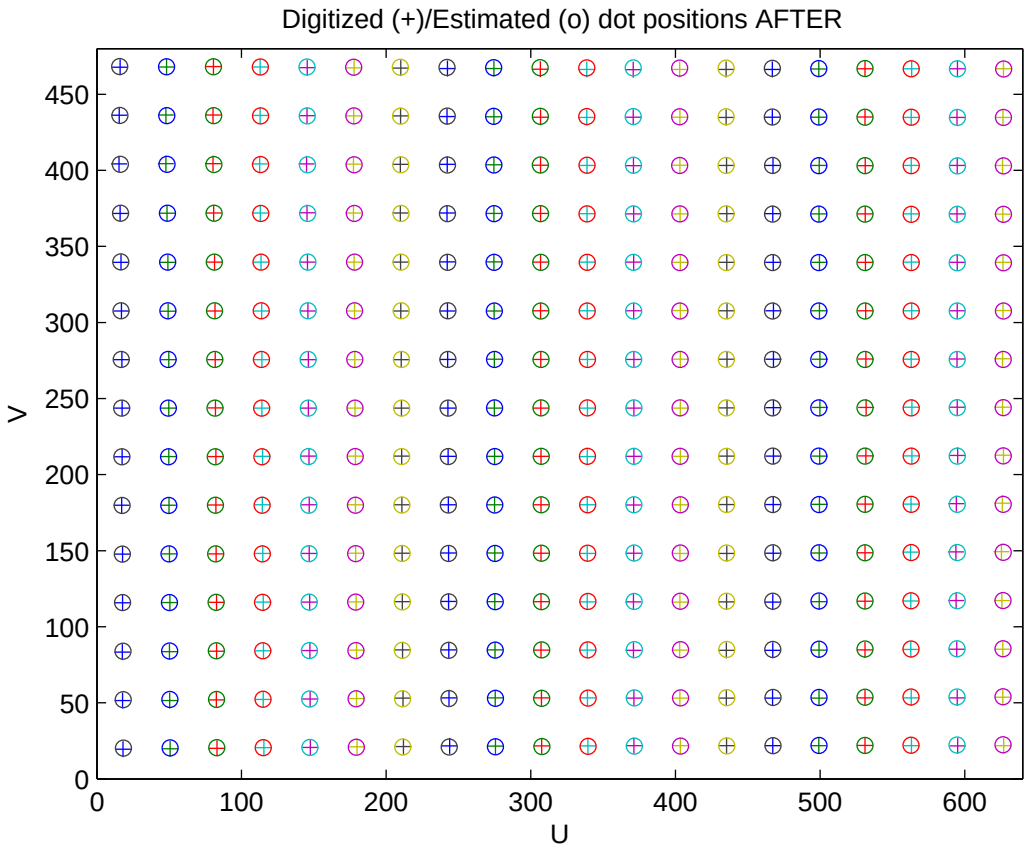


Figure B.15: Comparison between actual and theoretical dots after distortion correction for the 6mm lens #1, run 15.

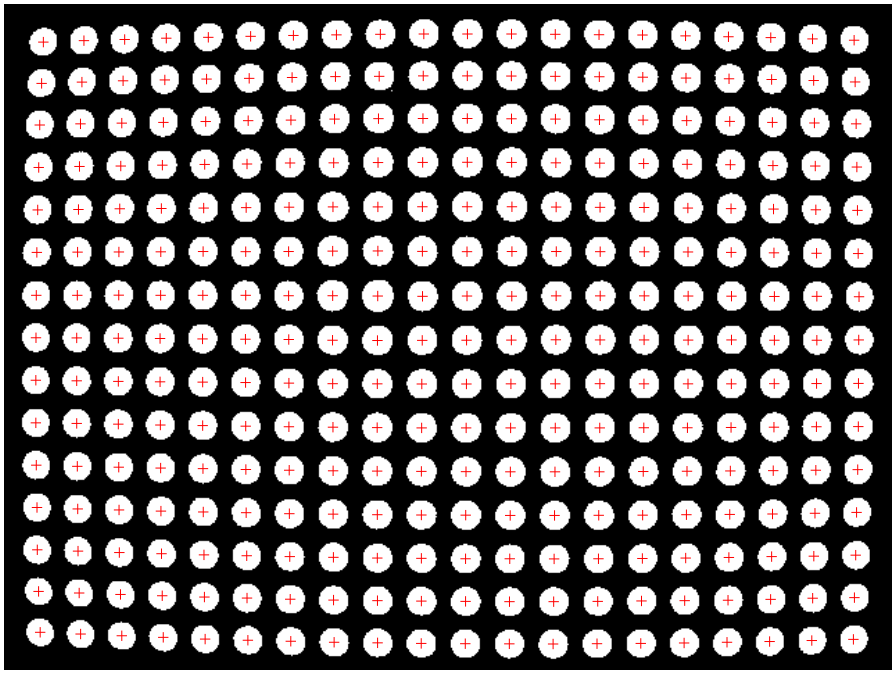


Figure B.16: Dot centers calculated by GETDOTS for 8.5mm lens #1, run 3.

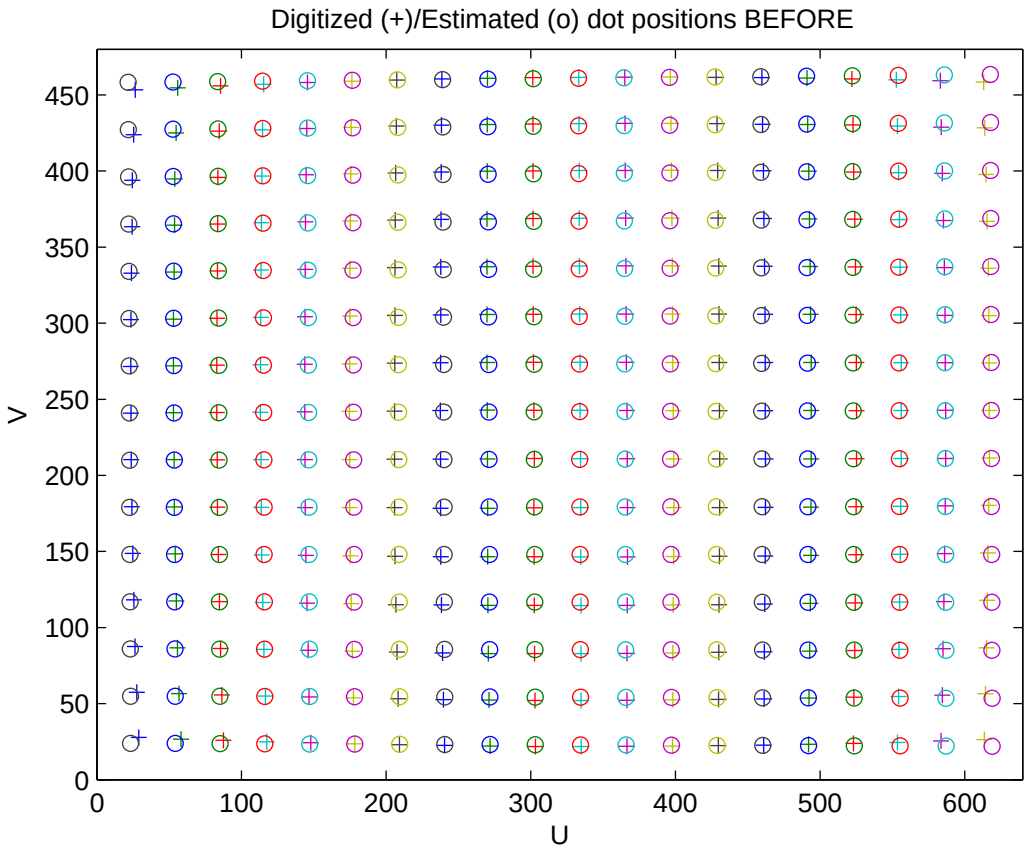


Figure B.17: Comparison from INTCAL between actual and theoretical dots before distortion correction for the 8.5mm lens #1, run 3.

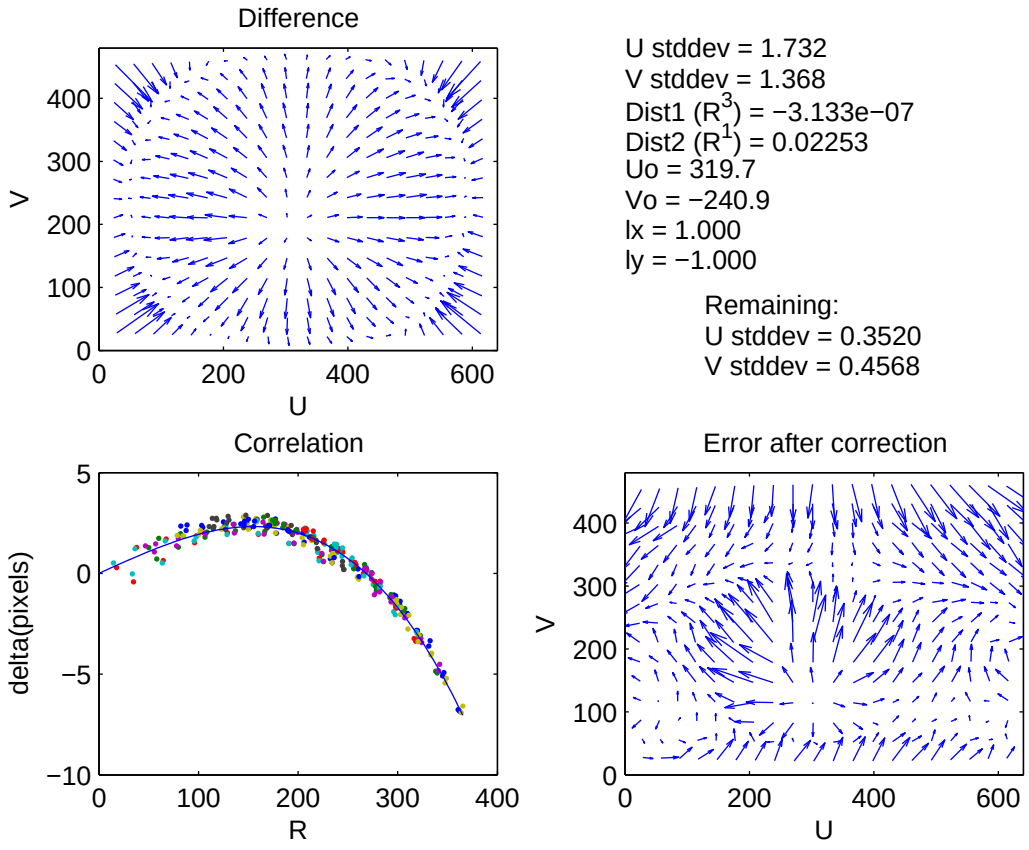


Figure B.18: Walton solution parameters using distorted image for the 8.5mm lens #1, run 3.

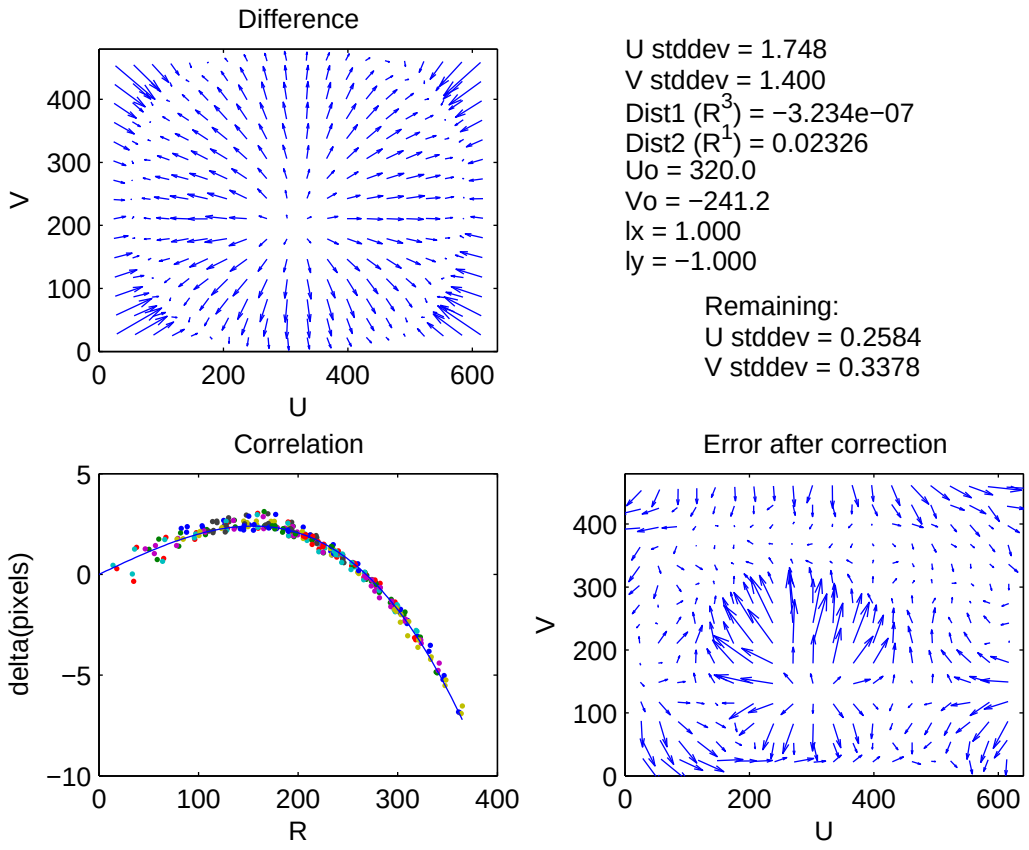


Figure B.19: Walton solution parameters using undistorted image for the 8.5mm lens #1, run 3. These are the numbers that are saved for use in rectification.

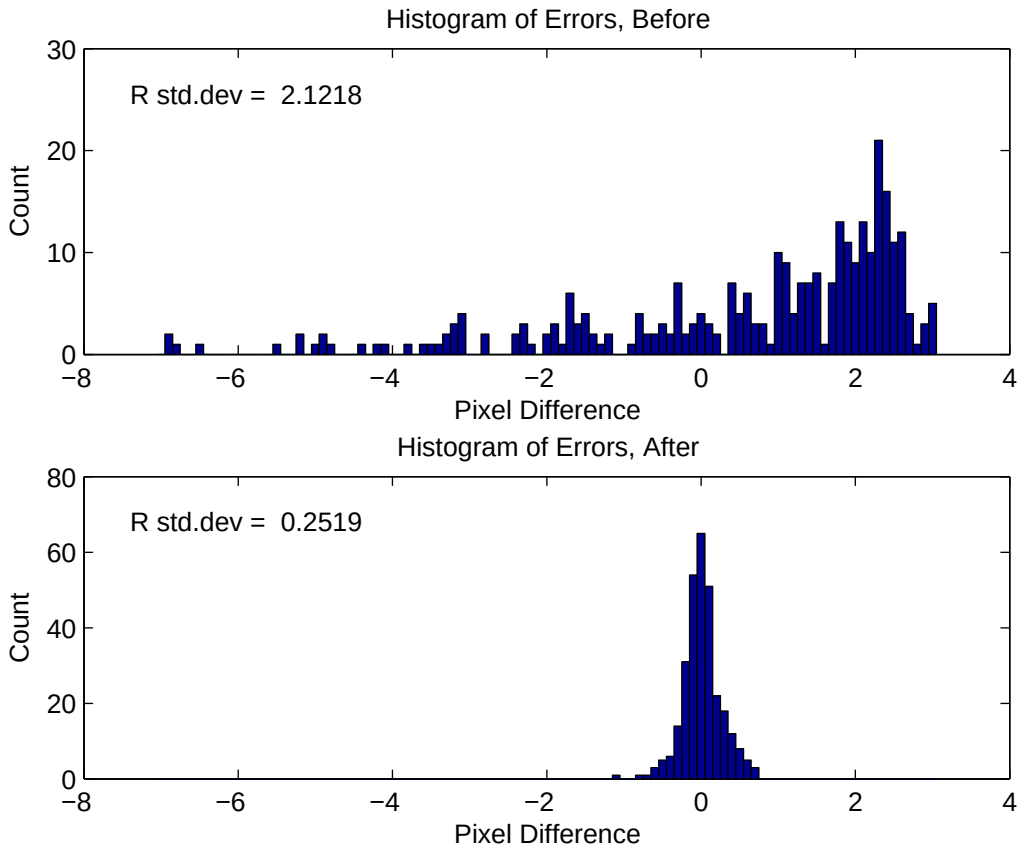


Figure B.20: Histogram of errors before and after correction for the 8.5mm lens #1, run 3.

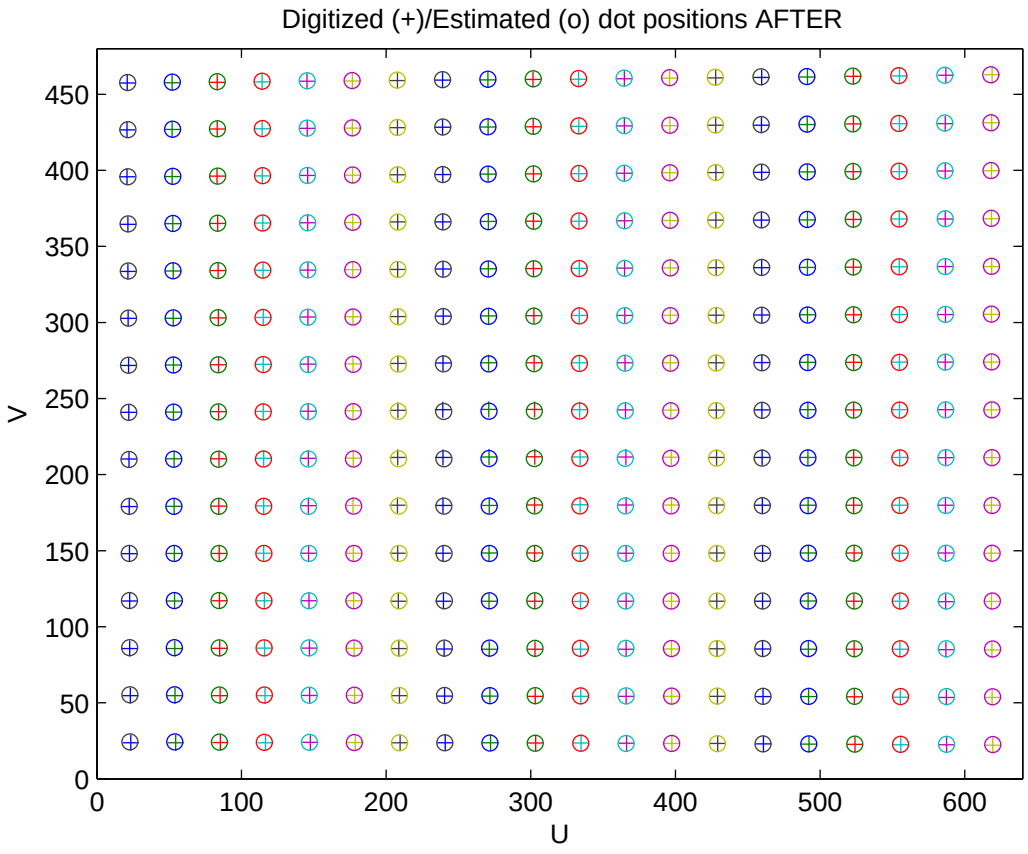


Figure B.21: Comparison between actual and theoretical dots after distortion correction for the 8.5mm lens #1, run 3.

For this manipulation to be possible, we need to know how the pixel coordinates in the original image correspond to 'real world' coordinates on the beach. After the camera system is set up at the site and camera views are locked down, a group of ground control points, or GCPs, are distributed along the beach in the view of each camera. There are several types of varying size used, depending on how far away from the cameras they will be deployed. Obviously, the further away, the larger they need to be. The smaller GCPs consist of a length of 2x4 with a 2' diameter black circular disc made of masonite screwed on at one end. The 2x4 is buried in the sand so the disc stands up and faces the cameras. The larger GCP is similar, but with a 4' diameter disc.

These GCPs are distributed along the entire beach within view of the cameras. The points they represent will be used in fitting a plane to the beach, so there need to be at least three GCPs visible in each camera view. In addition, they should be varied in their distance to the dune or shore so that they are staggered along the beach and don't simply lie in a straight line. It is useful that the cameras have overlapping views, so GCPs can be placed in the overlap and be used for the rectification of both camera's images. Once the GCPs are set in place, their coordinates need to be determined. In this study, the GCPs are surveyed by the Delaware DNREC survey team. The method used is a prism and transit, where the surveying station is set up at a known location and a laser is shot at a reflective prism to determine the locations of each GCP relative to the known location of the survey station. These values are double-checked with a differential GPS unit. In addition to the GCPs being shot, also surveyed are fixed environmental locations such as lifeguard stands, fenceposts, and storage boxes on the beach. Of course, the cameras themselves are surveyed as well.

While surveying is taking place, someone is at the camera control computers taking snapshots of the shoreline with all of the GCPs in place. Several are taken

from each camera and saved. The survey yields 44 GCPs, including the cameras, and five snapshots from each camera to be used in determining the camera geometries. All survey coordinates are in Delaware State Plane northing, easting and elevation (feet).

Figures B.22 to B.26 are snapshots from each camera view with the GCPs deployed and labeled. Table 2.2 lists the GCPs and their state plane coordinates.

Figure B.22: Camera 1 snapshot with GCPs labeled.

B.3 System Geometry

With the snapshots of the beach filled with GCPs, and with their real world coordinates being known, it is now possible to rectify the raw 3D images into a birds-eye view, 2D planar image of the beach face and surf zone. It is these images

Figure B.23: Camera 2 snapshot with GCPs labeled.

Figure B.24: Camera 3 snapshot with GCPs labeled.

Figure B.25: Camera 4 snapshot with GCPs labeled.

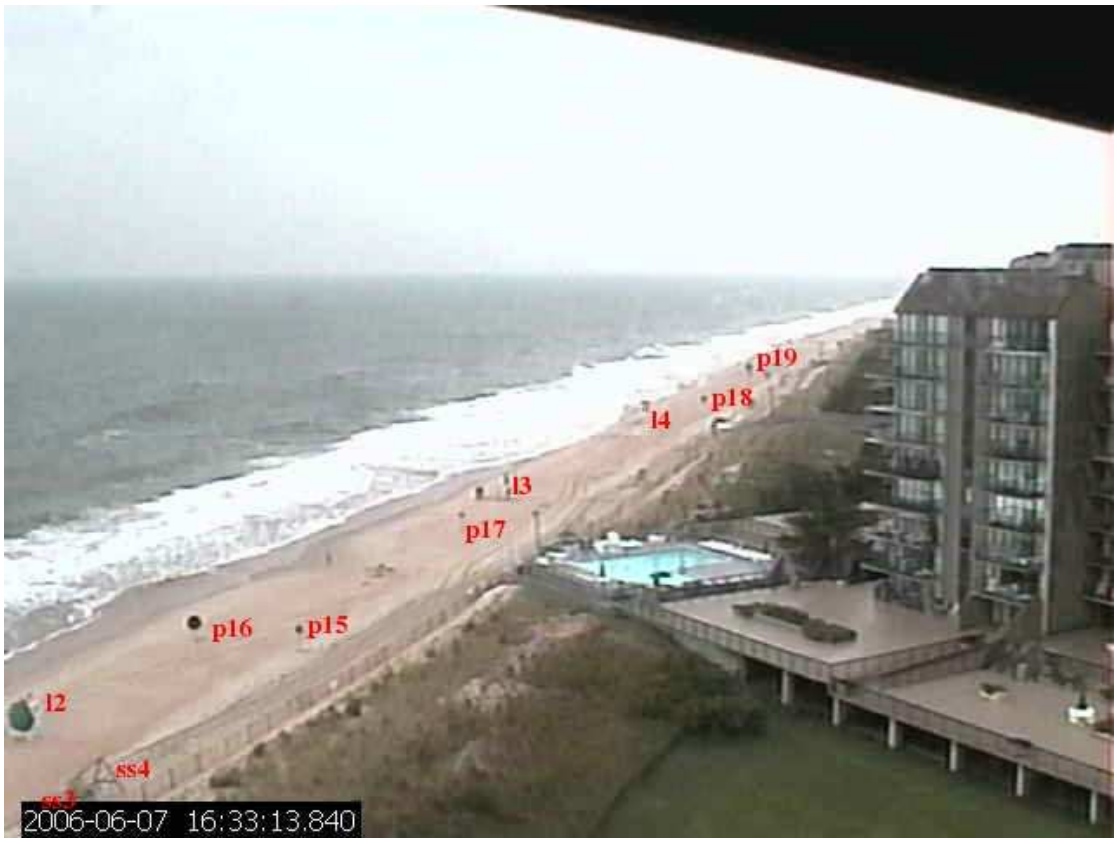


Figure B.26: Camera 5 snapshot with GCPs labeled.

that can be analyzed meaningfully. Before this manipulation can take place, the camera and geometry data must be properly prepared. MATLAB will be used for the remainder of the data analysis in this study.

The list of GCPs is contained in a comma-delimited text file, with fields for the point number, northing feet, easting feet, elevation, and a short description of the point. This information must initially be parsed into its respective fields for use in MATLAB. The first step in facilitating this is to manually zero-pad some of the numbers so each entry has the same amount of digits. The function GETGCP2 loads this text file and uses the comma delimiters to separate each field into cells, which are subsequently converted into an array of strings, one string for each survey point. A local coordinate 'zero' origin is defined, here being point B1, a box on the beach located at 193,594' northing, 759,845' easting. It is approximately in the middle of the view field of camera 3 at the shoreward edge of the dune.

Now, a MATLAB structure called GCP is built to store the GCP information. It has 44 entries, one for each GCP, and fields corresponding to each parameter of the GCP. Directly from the survey data, each string can be parsed into the point number, state plane north feet, state plane east feet, elevation feet, and description according to the digit index. For example, if beth_gcp is the survey string array, the point number is the first two digits in the string, so the field PointNumber in the structure GCP is assigned by $GCP(i).PointNumber = str2num(beth\_gcp(i, 1:2))$.

Similar commands assign the other fields. Several other fields are calculated directly from these. Northing feet and easting feet are simply the state plane north and east minus the origin previously defined. These coordinates are converted into meters as northing and easting meters. Finally, the local coordinates are defined as X and Y position. In this study, no rotation is used so the local coordinates are simply the state plane coordinates with the origin shifted. Positive Y is northward, and positive X is eastward. Table B.1 lists the local coordinates of the surveyed

GCPs.

A short m-file called GETIMAGES is used to store the directory paths to the GCP snapshots.

Another m-file called GETCAMDATA is used to store information about each camera. A new structure called Cam is defined, with fields for D1, D2, U_0, V_0, lx and ly. These, as you recall, are the distortion characteristics for each lens. These values are fixed. The final field in the Cam structure is called the correspondence matrix. This is an array consisting of the point numbers of the GCPs visible in each camera view along with their pixel coordinates. This will allow the rectification routine to determine how the real world survey coordinates on the beach translate into pixel coordinates in the images. These coordinates are determined by hand. First, the GCP snapshot is loaded and opened for view. Then it is possible to manually zoom in to each individual GCP. Using the GINPUT command, the user clicks on the center of the GCP and MATLAB returns its pixel coordinates within the image. This is repeated for each GCP until the correspondence matrices are complete. As an aside, one cannot forget to take into account image resolution when entering values in pixel units in the Cam structure. For example, the video files are at a resolution of 480x360, whereas snapshots are 640x480, so pixel coordinates must be scaled accordingly to keep the correspondence matrices correct.

Table 2.1 shows the distortion characteristics for the five cameras, determined from the previous lens calibration procedure. Tables B.2 and B.3 show the corresponding pixel coordinates of the GCPs visible in each camera view.

Finally, an m-file called GETGEOMETRYDATA is used to store each camera's geometry information in a structure appropriately called Geometry. The fields X, Y, and El are the local coordinates, in meters, of each camera. So, X and Y are the survey coordinates minus the local origin coordinates previously defined in GETGCP2, converted to meters. El is the elevation from the survey converted to

Point Number	Y Position (m)	X Position (m)	Description
1	-137.99	1.2871	DECK SCREW 1
2	-21.361	-16.834	BL S 5+00 TOP AL
3	614.65	10.98	BW-1
4	363.86	10.05	P-1
5	319.07	18.638	P-2
6	195.41	8.8389	P-3
7	136.52	5.4613	P-4
8	101.67	12.87	P-5
9	75.543	9.9582	P-6
10	54.448	-2.5095	SS-1
11	49.515	-2.8351	SS-2
12	52.519	13.668	P-7
13	40.955	6.5861	P-8
14	30.178	12.462	P-9 ADJUSTED -1'
15	34.338	-2.6185	F-1
16	33.763	-0.78543	F-2
17	15.356	11.078	L-1
18	3.676	10.739	P-10
19	0	0	B-1
20	-7.6746	0.25798	B-2
21	-15.466	6.1703	P-11
22	-31.888	9.7626	P-12
23	-31.729	17.359	P-13
24	-55.85	15.72	P-14
25	-74.567	4.4311	SS-3
26	-79.354	5.6594	SS-4
27	-87.249	17.056	L-2
28	-115.09	13.179	P-15
29	-113.49	22.509	P-16
30	-171.68	24.312	P-17
31	-193.88	27.046	L-3
32	-295.27	39.912	L-4
33	-337.59	39.857	P-18
34	-413.37	42.604	P-19
35	-10.955	-49.531	C-5
36	-10.654	-49.287	C-4
37	-8.5143	-47.602	C-3
38	-8.3183	-47.859	C-2
39	-8.1038	-48.099	C-1
40	-142.84	-0.20893	CORNER OF POOL
41	-173.18	-3.8903	BL S10+00 TOP AL
42	-325.06	9.0053	BL S 15+00 HUB
43	-476.87	21.813	BL S20+00 METAL
44	-628.18	38.532	BL S25+00 PVC

Table B.1: Local coordinates of the surveyed GCPs.

	Point Number	Horiz. (pix)	Vert. (pix)
Camera 1	3	110.74	99.96
	4	165.54	127.5
	5	205.1	133.03
	6	269.91	173.49
	7	341.98	212.61
	8	470.49	245.76
	9	554.44	290.34
	10	572.53	355.47
	11	599.22	369.94
Camera 2	6	127.99	244.04
	7	181.48	271.98
	8	277.07	294.49
	9	338.97	324.18
	10	353.84	371.3
	11	374.17	380.91
	12	453.03	355.61
	13	478.48	391.88
	14	581.48	404.59
	15	464.87	431.79
	16	480.05	428.61
Camera 3	12	40.883	211.04
	13	58.557	226.83
	14	113.61	223.67
	15	56.887	248.08
	16	64.004	244.92
	17	177.51	231.04
	18	235.11	241.1
	19	243.23	266.89
	20	293.13	269.27
	21	340.67	255.8
	22	424.8	246.37
	23	413.66	231.28

Table B.2: Table of correspondence coordinates for cameras 1-3.

	Point Number	Horiz. (pix)	Vert. (pix)
Camera 4	21	40.474	420.91
	22	188.09	368.05
	23	163.23	344.58
	24	340.34	311.01
	25	501.09	302.02
	26	514.88	294.47
	27	487.77	269.08
	28	606.48	253.51
	29	561.26	244.36
Camera 5	25	37.09	449.56
	26	56.5	433.76
	27	13.295	406.13
	28	170.96	360.2
	29	110.17	356.55
	30	264.52	294.17
	31	287.56	273.15
	32	368.11	232.96
	33	404.11	227.55
	34	430.07	209.28
	40	326.55	322.76

Table B.3: Table of correspondence coordinates for cameras 4-5.

meters. The lens field of view, FoV, is known from the lens specifications but may not be exactly correct. The unknown parameters, which need to be determined before rectification can take place, are azimuth, tilt, and roll. Azimuth is the clockwise rotation of the camera from the positive Y-axis, here being state plane north. Tilt is the angle rotating upward from straight down. Roll is the camera's counterclockwise angle from horizontal. All are entered in radians. Finally, the Flags field allows the azimuth, tilt, roll, and field of view to be fixed or free numbers. This is a crucial aspect to determining their correct values.

The geometry software uses an iterative solution method to calculate the unknown angular geometry parameters from above. However, the iteration can blow up rather easily, so the initial 'guessed' value must be reasonably close to the actual value in order for the iteration to close and rectification to succeed. This is where the fixing flags come in handy. The field of view is entered according to the specification sheet and left free. For azimuth, tilt, and roll, values are inserted according to the user's best guess of their value judging by the known camera locations and the image view relative to the desired coordinate system. These best guesses are set to fixed for the time being. Only one image can be worked on at a time.

Once set, a program called GEOMETRYFUNC3 is run, which in turn uses the program GEOMETRYSOLVER written by the ARGUS Group. This routine loads the camera, correspondence and geometry parameters as well as the GCP coordinates. For the desired image, the program will use the input (estimated) geometry data and plot the calculated coordinates of the visible GCPs along with the GCP image. It also plots a calculated horizon line. Now, it is possible to see how close the estimated angles are to their actual unknown values by examining how close the calculated points line up with their corresponding actual GCPs in the image. The horizon should line up as well.

It takes a bit of manipulation and trial-and-error to get things lined up. By

using the flags to free values that appear close to correct, and adjusting the fixed values, eventually the calculated GCPs will be reasonably close the their actual locations in the image. At this point, they may all be freed, and when the program is run again, everything should line up perfectly. The camera geometry has now been determined. This process is repeated for each camera in the array to determine its geometry. Now that the geometries for each camera view have been solved, image rectification can proceed.

Figure B.27: Output of GEOMETRYFUNC3 after iterative solution closes, camera 1.

Figures B.27 to B.31 show the final output of GEOMETRYFUNC3, when the guessed values of the unknown parameters are close to correct, the iterative solution closes, and the calculated points line up with their respective GCPs in the snapshot. Table 2.3 shows the calculated geometry values (coordinates and angular

Figure B.28: Output of GEOMETRYFUNC3 after iterative solution closes, camera 2.

Figure B.29: Output of GEOMETRYFUNC3 after iterative solution closes, camera 3.

Figure B.30: Output of GEOMETRYFUNC3 after iterative solution closes, camera 4.

Figure B.31: Output of GEOMETRYFUNC3 after iterative solution closes, camera 5.

displacements) for the five cameras after GEOMETRYFUNC3 succesfully iterates to a solution.