Video Recognition of Breaking Waves

Curtis Rusch*, J. Thomson, S. Zippel, and M. Schwendeman
Applied Physics Lab
University of Washington
Seattle, Washington

***Abstract*— An algorithm is presented to automate the identification of breaking waves in images collected with a camera on a drifting buoy. Each image is given a score from four separate analysis techniques: brightness detection, pixel histogram, entropy (texture) analysis, and glare identification. By combining these in a composite score, potential breaking wave images are detected and the number of images requiring manual review is a small fraction of the original set. Most of the images with false breaking wave signals due to sun glare are identified and removed. The final output is the wave-breaking rate over the length of the video capture.**

Keywords*— *Wave-breaking, image processing

I. Background

Breaking waves dissipate energy in the form of turbulence and bubbles and are important to fundamental processes at the air-sea interface. Breaking waves are also a hazard to safe operations at sea and to the survivability of coastal structures. Quantification of breaking waves is made difficult by the intermittent and complex nature of breaking. Here, we describe a method to identify breaking waves in video images and thereby compliment the *in situ* wave and turbulence data collected on SWIFT buoys [1].

SWIFT (Surface Wave Instrument Float with Tracking) buoys are Lagrangian drifters designed for *in situ* observations at the air-sea interface (Fig. 1). They measure currents, winds, waves, turbulence, and take images of the sea surface. SWIFTs have been used to study river mouths at the New River Inlet in North Carolina, USA, and the Columbia River between Washington and Oregon, USA [2]. They have also been used to study whitecaps in fetch-limited seas [3] and the open ocean [4]. Additional deployments are planned in the Arctic as part of a sea-ice study in the summer of 2014 and 2015.

GoPro cameras are mounted on the mast to observe breaking waves. The camera is located just less than 1 meter above the water level. The cameras are oriented to point straight down at the water surface, parallel to the mast. The cameras are set to record one image per second to maximize battery life. Cameras capture approximately 12,000 images on a typical 3-4 hour deployment per drifter. In previous field studies, up to 6 drifters were deployed multiple times each day for a month. To date, these images have been manually reviewed to identify particular events of turbulent mixing by breaking waves. The large number of images makes manual review impractical for the scale of a full experiment. The purpose of this work is to turn these images into quantitative data, which will be valuable in the continued analysis of the other SWIFT measurements.

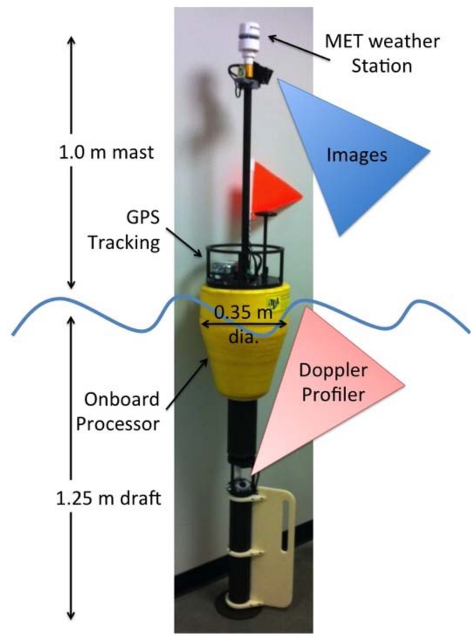

Fig. 1. SWIFT drifter

II. Methods

The following methods are implemented in a MATLAB algorithm to detect breaking waves in images. The algorithm consists of three steps: masking, breaking wave identification, and glare detection. The basis of this algorithm is a scoring system, where images are granted one point for each breaking wave detection algorithm that they pass, and the images have two points deducted from their score if glare is detected in the image:

$$\text{Score} = \text{Contrast} + \text{Histogram} + \text{Entropy} - \text{Glare}$$

After the algorithm is complete, images are reported with scores of one, two or three points. The higher an image's score, the higher the probability that it contains a breaking wave. Manual review is then performed on the highly scored images, which are only a small fraction of the total set.

Funding provided by the Office of Naval Research and the National Science Foundation.

* Corresponding author: Curtis Rusch, curusch@apl.uw.edu

A. *Masking*

As shown in Fig. 2, images taken on board SWIFT buoys contain the base, mast, and flag of the buoy. In addition, many of the images contain the horizon and sky, especially when conditions are rough and the buoys are tilted by large swells. These features add brightness and texture to the image which confound the identification of breakers. Therefore, each image is masked before processing. The pixels located in a rectuangular region around the mast are assigned pixel intensities of zero. The image is then converted into polar coordinates and a circular region of zero pixel intensity is placed over the buoy base. A second, outer pixel radius is then specified to eliminate noise from the horizon and distant wave fields. An example of a masked image is shown in Fig. 3.

B. *Contrast Filtering*

The next step of image processing is contrast filtering. All images are run through the MATLAB function "imadjust." This function screens pixels below a prescribed pixel intensity (in this case, 70%), and then adjusts the intensities of the remaining pixels to conform to a 0-100% intensity scale. The remaining pixels are then counted for each image. Images not containing breaking waves generally have few remaining, nonzero intensity pixels, if any.

Before an image passes to the four scoring algorithms, it must first pass through a dynamic threshold. When the algorithm begins, the first six images taken in a deployment are used to calibrate the dynamic threshold. A set of six images was chosen because the SWIFT buoys rarely encounter breaking waves more often than once every six seconds. The first six images are run through the masking and bright pixel counting algorithms, and the number of bright pixels for each image is recorded in a six-column vector. When the algorithm proceeds past the calibration section, the algorithm updates the vector each time a new image completes analysis. The most recent pixel count is added as a temporary seventh vector entry, and the first entry is deleted. In this way the vector retains only the past six pixel counts. For each image, the average pixel count of the previous six images is divided by the number of bright pixels in the current image. The ratio of average bright pixel count to bright pixel count in the current image is expected to be small for breaking waves. If the ratio is below an empirically determined threshold of 0.17, the image is allowed to enter the scoring algorithms. A low ratio represents a spike in the bright pixel count, meaning the image currently being analyzed has more bright pixels than normal. If the ratio is larger than this threshold, then the algorithm proceeds to the next image, since this means that there is no large deviation from the mean bright pixel count.

Fig. 2. Example of a SWIFT breaking wave image

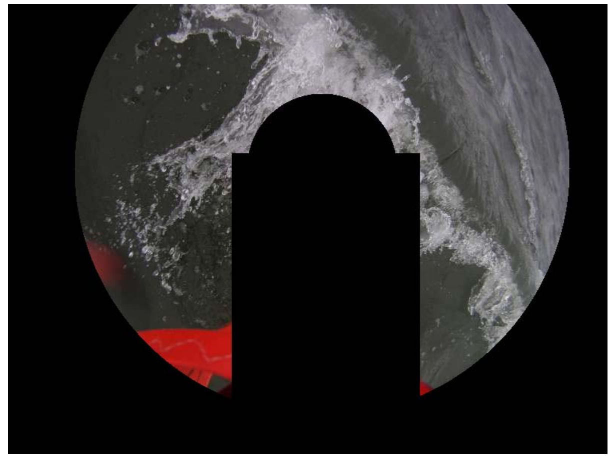

Fig. 3. Example of a masked image

If a spike has been detected, then the bright pixel count for the image is checked. If this number is greater than 3000 (a threshold determined empirically for this data set), then the image has one point added to its score.

Using the initial dynamic threshold decreases the number of false positives recognized as breaking waves. For example, sea foam, which is common in breaking regions, often passes the static brightness threshold. Since the foam often persists for consecutive images, the algorithm will account for the increase in average background pixel intensity. As a result, even though the sea foam produces a high bright pixel count, the image will not be scored. In addition to reducing false positives, this allows most breaking waves to still be detected in the presence of foam.

C. *Texture Analysis via Range-Filtered Histogram*

The next technique employed is a texture analysis. Using the MATLAB "rangefilt" function, the images are analyzed for identifiable variations in surface texture. Each image is analyzed as neighborhoods of pixels. Each output pixel is assigned a new intensity value based on its range value. The range value is the difference between the maximum and minimum intensity values of the 3x3 neighborhood of pixels surrounding the output pixel. Brighter output pixels are the result of a greater difference in pixel intensity across the neighborhood. The output is similar to a contour map of the image, highlighting the regions of high texture (see Fig. 4). Next, the range filtered image is run through the MATLAB "imhist" function, which produces a histogram of the pixel

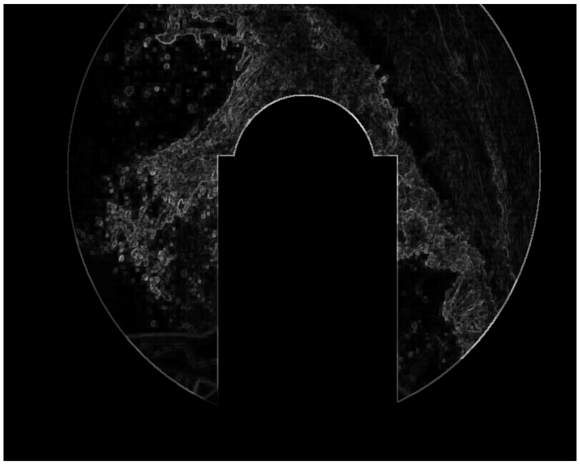

Fig. 4. Example of a range-filtered image

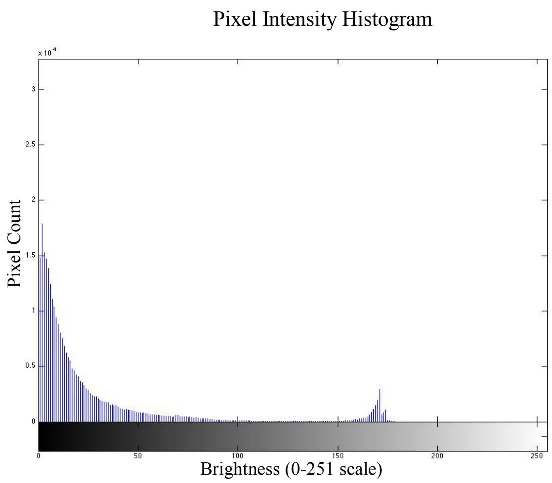

Fig. 5. Pixel intensity histogram for the image in figure 4. Small peak in bright region indicative of breaking wave.

intensities (Fig. 5). Secondary peaks in the histogram, in the region of higher pixel brightness, indicates a breaking wave.

This approach is similar to one used for infrared detection of breaking waves [5]. The median of the histogram is then calculated for use as a threshold. When the secondary peak is present, the median pixel intensity value is increased. Images with a median greater than a threshold of 30 (again an empirically determined value) are scored, adding one point to the image score.

D. Image Entropy (Texture) via Fast Fourier Transform

The next step combines the tools of Fast Fourier Transforms and Gaussian filters, similar to many facial recognition algorithms discussed on MATLAB central, an online MATLAB forum. The images pass through the MATLAB “fft2” function, which performs a two dimensional Fourier Transform on the image. The important features of the image are then distinguished using Gaussian filters. This particular algorithm uses a band-pass filter, which blurs the image, emphasizing the larger regions of texture. The resulting filtered images then pass through the “ifft2” function, taking the inverse of the two dimensional Fast Fourier Transform (an example of the output can be seen in Fig. 6). Lastly the entropy of the new image is found using the MATLAB “entropy” function. This function essentially measures the randomness of the image, a statistical value that is typically larger for images containing breaking waves. An empirically determined minimum entropy value of 2.8 is then used to identify breaking waves (and add one point to the image score).

E. Glare Detection

Glare was a large source of false positives in earlier versions of the breaking wave recognition algorithm. Glare was observed across a variety of sea-state and weather conditions. While it is simple for a human to distinguish glare from a breaking wave, images with glare are often scored by the image brightness detection and texture analysis. The glare detection algorithm described below was developed to reduce the number of glare-induced false positives.

The algorithm is based on the observation that glare is strongest close to the horizon (i.e., at grazing angles). When glare is present, pixel intensity generally increases away from the center of the image to the horizon. Thus, this algorithm sorts each pixel according to its radial distance from the center of the buoy. Since the buoy is not in the exact center of the image, the location of the buoy center was determined and input for this particular SWIFT camera orientation. The maximum pixel intensities are recorded at each integer radial value and plotted against their respective radial values. Using the MATLAB function “polyfit”, the maximum pixel intensities are regressed against radial distance. An example of one of these plots with the linear regression shown for an image containing glare can be seen in Fig. 7. Fits with positive slopes greater than 0.25 are assumed to correspond with a glare-induced false positive and two points are subtracted from the image score. Even if an image tests positive for glare, it may still have a positive score, and thus will be considered a potential breaking wave. Oftentimes a breaking wave will cause the buoy to tip drastically, increasing the camera angle. This near grazing angle leads to

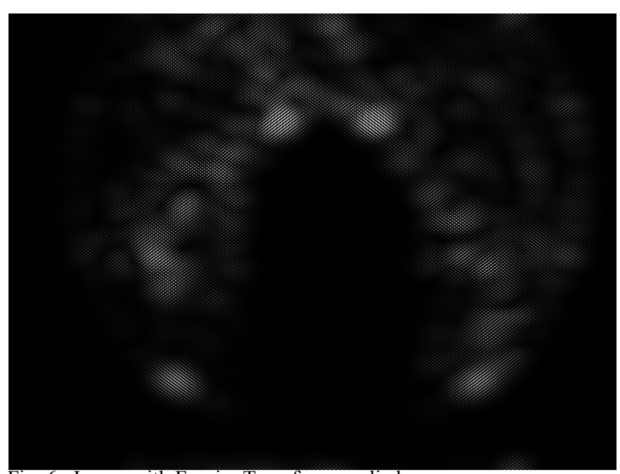

Fig. 6. Image with Fourier Transform applied

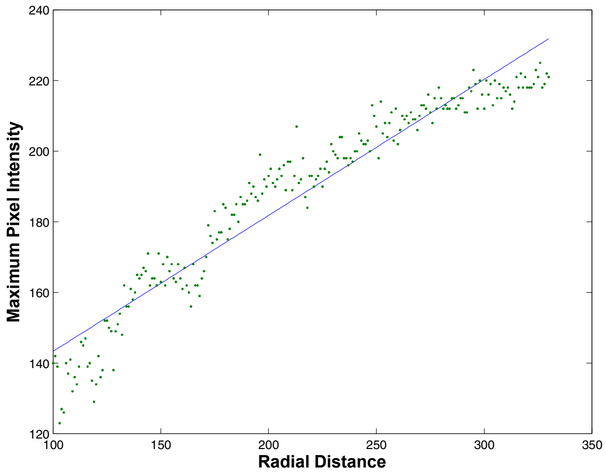

Fig. 7. Glare detection plot for glary image. Plotted are maximum pixel intensity at each integer radial value, and the line of fit for these points. The positive slope is characteristic of glare.

increased glare. Thus, images that pass the three detection algorithms will still be considered if they test positive for glare. This allows most breaking waves to pass on to the user for manual review, while eliminating most of the glare-induced false positives.

F. Algorithm Completion and Manual Review

Even if an image has passed through the scoring algorithms with a positive score, it has two more steps it must pass through before it is counted as a breaking wave. First, dynamic processing is used to avoid double-counting breaking waves. When large waves break, there is often a large amount of sea foam present for just one or two images. Since the large breakers have such a large number of bright pixels, the ratios from the bright pixel averages mentioned above may still be low enough for the image to be counted. As a result, the algorithm also keeps track of which images have been counted. At the beginning of processing, a “check” vector is initialized containing a zero for every image that will be analyzed. As the algorithm proceeds, the score of each image is checked after it has passed through all of the scoring algorithms. If the image has been given a score greater than zero, a 1 is added to the “check” vector in the position corresponding to the image being analyzed. Then, the last six entries in the vector are checked. If a 1 is present in any of the previous six entries, the image is discounted as a double count, and removed from the final scored images list.

After the algorithm runs, there are two initial outputs. There is a vector named “ScoredFiles”, which contains the name of each image file that passed through the algorithm with a score greater than zero. Another vector, named “Score”, shows the corresponding score that each image file was given. Not all of these files contain breaking waves – in fact, many contain no breaking waves. As a result, an additional, manual review follows the completion of the MATLAB program.

The last step that the images must complete is manual review. For best accuracy, the user reviews all images with a score above zero, even though the majority of breaking waves are found in images with a score of three. The images listed in the “ScoredFiles” vector are displayed one at a time on the screen. If the user sees a breaking wave in the image, the space bar is pressed, and if there is no breaking wave, the arrow keys are used to advance to the next image. This allows a brief check that eliminates remaining false positives. A final vector list of all breaking waves identified in the deployment is then saved. When this step is complete, the algorithm then outputs a few final statistics useful for data analysis. The major output of this version is breaking rates over the course of a SWIFT deployment. The number of breakers is tallied every five minutes over the span of the deployment (since other data recorded and processed by the SWIFT buoys are recorded in five minute intervals).

III. Results

The focus of this algorithm is to reduce the number of images requiring manual review to a feasable amount. As a result, the algorithm values accuracy over speed and over full automation. The algorithm identifies extra false positives since it must be sensitive enough to identify breaking waves across a range of conditions. Even with this added sensitivity, the algorithm does well in reducing the number of images to manually review. Thus far, the algorithm has been used to analyze over 142,000 images from SWIFT deployments at the Mouth of the Columbia River, and it has flagged just over 1% of those for manual review. The algorithm is able to analyze images at an average rate of 3.5 images per second (i.e., 3.5 times faster than real-time playback of the image data). Of the images that score above zero, about 18.5% show actual breaking waves (as determined by manual review).

The overall accuracy of the algorithim is 78%, as determined by manual review of *all* images (rather than just those with a score greater than zero). The algorithm performs best in cloudy conditions. Cloudy conditions yield not only fewer missed breakers, but also fewer false positives. This is mostly due to the more consistent lighting, and decreased number of bright pixels not associated with breaking waves. Mid-day sunny conditions create problems for the algorithm, since bright sunshine at a high angle often reflects from the middle of the image, allowing the bright spot to pass through the glare detection algorithm unnoticed. For the dataset collected at the Mouth of the Columbia River in 2013, there are few sunny days, and sunny days typically do not have very high breaking rates. Four deployments with bright sun were analyzed. Of the 33,742 images taken during these sunny deployments, just 20 were breaking waves. Unfortunately, due to the sun glare, 12 of these images were missed by the algorithm. Comparitively, there were 13 deployments analyzed with cloudy conditions. Of the 98,717 images during the cloudy deployments, 380 were breaking waves, and the algorithm correctly identified 80% of these images. This percentage was diminished by a few days with sun-breaks. Completely overcast deployments were 96% accurate, with only 2 missed breakers out of 45 manually counted breakers. See Table I, on the last page, for more detailed statistics.

Two main causes of missed breaker detection were sun glare and the dynamic thresholding algorithm. Many of the breaking waves were missed because high sun angle created large numbers of bright pixels in the image, often in the same places that breaking waves occur. This means that there is little to no spike in bright pixel numbers when a breaking wave occurs, and the images would be passed over completely by the algorithm. The second main cause of missed detection is when breaking waves occur within six seconds of a false positive. These images often still show the signatures of a breaking wave, but since they occur within six seconds of an image that was already flagged, they are discounted.

There were also a few small breakers that the algorithm missed. These were hidden by the masking procedure. Thus, the masking knowingly reduces the field of view of the camera, and since this is consistent for the entire batch of images, the breaking rates will be slightly reduced, evenly throughout the deployment. The breaking rate plots that have been output from the algorithm have been compared with other measurements collected onboard the SWIFTs, in particular the measurements of turbulence and wave dissipation rates, and used to confirm wave breaking as the driving mechanism.

IV. Discussion

Although this algorithm was developed specifically for images collected with SWIFT buoys, video recognition of breaking waves could be applied to many other studies. The availibility and ease of depoyment of GoPro cameras makes video observations a viable source of quantitative data. Simple modifications to this algorithm could be used in different applications. The simplest would be the adjustment of the initial masking regions. A different platform with even a slightly different camera angle would have different regions of the image that would show breaking waves. The most difficult modification would be adjusting the thresholds, which would require a training data set, because the thresholds are empirical.

Thresholds were determined by trial and error on a data set that had been manually reviewed for breaking waves. The images were run through the vaious functions in batches, and values for each function were recorded and analyzed by hand. By comparing the different values for breaking and nonbreaking waves, suitable thresholds were determined. A new generation of the SWIFT buoys has a different camera angle, so the algorithm will need to be adapted. Using methods similar to facial recognition software, the algorithm could be trained to set its own thresholds much more accurately than can be done by hand. This would require a thorough set of images containing breaking waves in a variety of conditions, as well as a number of images that could have a high likelihood of being counted as a false positive. When applied, this should further increase the accuracy algorithm, and allow easy adaptation to any number of different configurations.

V. Conclusion

An algorithm to detect breaking ocean waves in images collected onboard a drifting platform was developed. The algorithm combines three separate detection methods; brightness detection, pixel histogram, and entropy (texture) analysis, as well as a glare detection method to score each image. High scoring images are flagged for manual review.

This algorthim flagged 1% of images on average and thereby reduced manual review by 99%. About 18.5% of the flagged images were actual breakers. The algorthim missed 22% of breakers on average. Algorithm performace was found to relate to weather conditions, with poor performance on sunny days and excellent performance on cloudy days.

Continued work will attempt to improve perscription of thresholds and test the robustness of the algorithm at different field sites. With objective threshold determination, and an ever-increasing batch of images to analyze, it is expected that the algorithm will increase in accuracy.

Acknowledgments

We thank the Applied Physics Laboratory field engineers, Joe Talbert and Alex De Klerk, as well as the countless people who assisted with SWIFT deployment on the *R/V Oceanus, R/V Point Sur* and the *F/V Westwind* on the Columbia River. We thank Adam Keen for the image masking method and initial progress on automated detection.

References

[1] J. Thomson, “Wave Breaking Dissipation Observed with ‘SWIFT’ Drifters”, *J. Atmos. Oceanic Technol.,* vol. 29, no. 12, pp. 1866-1882, Dec. 2012.

[2] Zippel, S., and J. Thomson, “Wave breaking due to depth and currents”, poster at *Ocean Sciences Meeting*, 2014.

[3] Schwendeman, M., J. Thomson, and J. Gemmrich, “Wave breaking dissipation in a fetch limited sea,” *J. Phys. Oceanog*., 44 (2014).

[4] Thomson, J., E. D’Asaro, M. Cronin, E. Rogers, R. Harcourt, and A. Shcherbina, “Waves and the equilibrium range at Ocean Weather Station P,” *J. Geophys. Res*., 118 (2013).

[5] Carini, R., C.C. Chickadel, and A. T. Jessup, “Estimating energy dissipation due to wave breaking in the surf zone using infrared imagery”, presentation at *Ocean Sciences Meeting*, 2014.

TABLE I. ALGORITHM PERFORMANCE STATISTICS FOR SELECT 2013 MOUTH OF COLUMBIA RIVER DEPLOYMENTS

	Batch	**Conditions**	**Reviewed Images**	**Flagged Images**	**Flagged Breakers**	**False Positives**	**Missed Breakers**	**Accuracy**
Training Batches[a]	SWIFT05_24May2013	Cloudy	9777	2	0	2	0	N/A
	SWIFT06_24May2013	Cloudy	13941	64	7	57	0	1.00
	SWIFT04_25May2013	Cloudy/Rain	11826	92	12	80	1	0.92
	SWIFT05_25May2013	Cloudy/Rain	4509	1	1	0	2	0.33
	SWIFT05_28May2013	Cloudy/Rain	8144	129	62	67	16	0.79
	SWIFT06_28May2013	Cloudy/Rain	9000	231	87	144	30	0.74
	SWIFT05_03Jun2013	Cloudy	6688	14	5	9	1	0.83
	SWIFT06_03Jun2013	Cloudy	9760	31	12	19	0	1.00
	Subtotals		73645	564	186	378	50	0.79
Independently Tested Batches	SWIFT01_21May2013	Cloudy/Rain	6431	96	34	62	7	0.83
	SWIFT04_21May2013	Cloudy/Rain	7081	187	24	163	7	0.77
	SWIFT05_21May2013	Cloudy/Rain	6540	192	21	171	4	0.84
	SWIFT07_21May2013	Cloudy/Rain	8877	353	31	322	9	0.78
	SWIFT05_04Jun2013	Bright Sun	6434	56	4	52	3	0.57
	SWIFT06_04Jun2013	Bright Sun	7121	8	1	7	6	0.14
	SWIFT04_08Jun2013	Bright Sun	10989	93	1	92	1	0.50
	SWIFT05_08Jun2013	Bright Sun	9198	112	2	110	2	0.50
	SWIFT04_24Jul2013	Cloudy	5930	21	7	14	0	1.00
	Subtotals		68601	1118	125	993	39	0.76
	Totals		142246	1682	311	1371	89	0.78

a. Training batches refers to images used during threshold determination.