Project Report
ATC-453

Visibility Estimation Through Image Analytics

M.P. Matthews

20 June 2023

Lincoln Laboratory
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LEXINGTON, MASSACHUSETTS

This report is the result of studies performed at Lincoln Laboratory, a federally funded research and development center operated by Massachusetts Institute of Technology. This material is based upon work supported by the Federal Aviation Administration under Air Force Contract No. FA8702-15-D-0001. Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Federal Aviation Administration.

Massachusetts Institute of Technology
Lincoln Laboratory

Visibility Estimation Through Image Analytics

M.P. Matthews
Group 43

Project Report ATC-453

20 June 2023

This material is based upon work supported by the Federal Aviation Administration under Air Force Contract No. FA8702-15-D-0001.

Lexington Massachusetts

This page intentionally left blank.

ABSTRACT

MIT Lincoln Laboratory (MIT LL) has developed an algorithm, known as the Visibility Estimation through Image Analytics Algorithm (VEIA), that ingests camera imagery collected by the FAA Weather Cameras Program Office (WeatherCams) and estimates the meteorological visibility in statute miles. The algorithm uses the presence of edges in the imagery and the strength of those edges to provide an estimation of the meteorological visibility within the scene. The algorithm also combines the estimates from multiple camera images into one estimate for a site or location using information about the agreement between camera estimates and the position of the Sun relative to each camera's view. The final output for a site is a prevailing visibility estimate in statute miles that can be easily compared to existing automated surface observation systems (ASOS) and/or human-observed visibility. This report includes thorough discussion of the VEIA background, development methodology, and transition process to the WeatherCams office operational platform (Sections 2–4). A detailed software description with flow diagrams is also provided in Section 5. Section 6 provides a brief overview of future research and development related to the VEIA algorithm.

This page intentionally left blank.

TABLE OF CONTENTS

This page intentionally left blank.

LIST OF ILLUSTRATIONS

Figure No. **Page**

This page intentionally left blank.

1. INTRODUCTION

For members of the general aviation (GA) community, low ceiling and visibility (C&V) conditions significantly influence flight planning and can have dire consequences when encountered unexpectedly. In remote regions, such as Alaska, it can be difficult for the GA community to obtain accurate current conditions of the C&V. Traditional weather observations are widely dispersed and inadequately forewarn of hazardous weather potentially encountered at the remote airstrips, along the desired routes between destinations, or through hazardous mountain passes with localized conditions. In response, the Federal Aviation Administration (FAA) has deployed a large number of high-quality camera installations, beginning in Alaska and expanding outward, to provide pilots with a live view of the current conditions via an FAA-maintained website. In 2016, the FAA tasked the Massachusetts Institute of Technology Lincoln Laboratory (MIT LL) to develop an algorithm that automatically estimates the visibility from the cameras.

The algorithm, known as Visibility Estimation through Image Analytics (VEIA), uses the presence and strength of edges in the imagery in comparison to a clear day reference to compute a visibility estimate. Since most of the FAA weather camera locations have 3–4 cameras pointed in different directions, VEIA combines the camera specific estimates into a single prevailing estimate using logic that considers the agreement between the cameras and the position of the Sun. The final output for a site is a prevailing visibility estimate in statute miles that can be easily compared to existing Automated Surface Observation System (ASOS) readings and/or human-observed visibility.

This report consists of four main sections that document the VEIA algorithm. The background section provides the history of the VEIA development from initial concept to present day. The methodology section provides a technical discussion on the algorithm and the performance analysis. The operational implementation section describes the implementation of the algorithm on the FAA Weather Cameras Program Office (WeatherCams) real-time system. Finally, the section on future research activities discusses potential algorithm improvements and additional functionality that could be possible through additional research.

This page intentionally left blank.

2. BACKGROUND

MIT Lincoln Laboratory began working with the FAA to develop camera derived visibility estimation algorithms in 2016. The initial focus was the implementation of the algorithm concepts on an archived dataset provided to MIT LL by the National Center for Atmospheric Research (NCAR). This dataset consisted of images collected between 2013 and 2015 gathered from the WeatherCams office through the public-facing website and application program interface (API) at a 20-minute update rate from the cameras available at that time. MIT LL also began archiving all available imagery from WeatherCams in the summer of 2017 at a 10-minute update rate, matching the image collection rate performed by WeatherCams. This included the latest camera installations in critical mountain pass regions and locations in Canada from an agreement between the FAA WeatherCams office and NAV CANADA.

Prior to 2016, MIT Lincoln Laboratory had developed some of the initial VEIA concepts while also working with other government organizations such as the Federal Highway Administration in the lower 48 states (Hallowell 2005, 2007). The primary focus of the initial effort with the FAA was to evaluate and adapt the techniques to the unique Alaska environment with limited daylight hours during the winter and the very low Sun angles that are present across the state during the entire year. Throughout 2017, 2018, and 2019, MIT LL evaluated the translation functions required to convert the edge strength information into visibility and developed techniques to combine the multiple cameras available at each site into one estimate. This was critical to alleviate issues with the low Sun angle negatively impacting cameras looking directly into the Sun during times when clouds or fog do not block the Sun.

By 2019, evaluations conducted on VEIA performance by MIT LL on a large-scale, archived dataset had shown a skill level believed to be acceptable for the user community. Therefore, planning began for an operational demonstration to be conducted during the summer of 2020, which would include an evaluation by the Forecast Impact and Quality Assessment Branch at National Oceanic and Atmospheric Administration (NOAA)/Global Systems Laboratory (GSL). The operational demonstration was conducted to provide independent evaluation and focus on two main objectives; a real-time interface for review of the operational system and a large-scale data processing effort to provide retrospective runs for statistical analysis.

To accomplish the first objective, MIT LL began working closely with the WeatherCams office to transition the VEIA technology into their real-time operational system over the winter of 2020. To implement the developmental software on the WeatherCams system, modifications were required to operate on the WeatherCams real-time platform due to limitations on environments that could be supported by WeatherCams. The developmental software was implemented in the MATLAB environment, so the operational environment chosen was an open-source platform known as Octave, which is mostly compatible with MATLAB. Modifications were required to adapt the software to accommodate a large-scale dataset, to meet the required update rates that were consistent with the image gathering frequency, and to add additional system monitoring processes.

The second objective was accomplished by performing retrospective runs on all imagery gathered between 1 September 2019 and 30 September 2020. This required gathering and archiving all imagery beginning mid-August 2019 and performing a large-scale reprocessing effort upon completion of data gathering in the early days of October 2020. The MIT Lincoln Laboratory Supercomputing Center (LLSC) was utilized for this effort, and processing was completed with distribution of the results to NOAA/GSL on 15 October. It is important to note that the reprocessing effort was conducted on LLSC with the development software in MATLAB, whereas the real-time results produced on WeatherCams were with the Octave version.

Results from the operational demonstration and evaluation conducted by NOAA/GSL confirmed the performance of VEIA was similar to the evaluations conducted by MIT LL. The conclusions also highlighted the strengths and weaknesses of VEIA. Overall, the results showed VEIA was performing well compared to alternative technologies and should move on to the next phase for operational acceptance. (Fenton, 2021)

During the spring of 2021, the FAA Aviation Weather Demonstration and Evaluation (AWDE) Services Team conducted a user assessment to determine the suitability and usability of VEIA. MIT LL worked with the FAA AWDE to provide training to a range of users who would be provided with access to VEIA on the WeatherCams website for a period of six weeks. Upon completion of the six-week period, the FAA AWDE team conducted interviews and provided questionnaires to 32 operational users ranging from pilots (Part 135 and GA), to dispatchers (Part 121 and Part 135), to meteorologists (Alaskan Aviation Weather Unit [AAWU], Center Weather Service Unit [CWSU], and NWS Weather Forecast Offices [WFO]). The user assessment concluded that VEIA was both suitable and usable by the aviation community with a number of recommendations for ways to improve the product display (currently the responsibility of the WeatherCams office). (Miller, 2021)

Throughout this time, MIT LL also worked with the FAA to explore crowd-sourcing techniques to estimate the visibility from the cameras. The crowd-sourcing concept involves asking human workers (both trained and untrained) to evaluate the images and provide a visibility estimate. The crowd-sourcing algorithms would then combine the set of estimates into a final visibility value based on historical worker performance and statistical techniques. The most challenging aspect of crowd sourcing is the sheer volume of images from thousands of cameras at a 10-minute update rate. If the worker pool is limited and/or taking too long to estimate the visibility, the crowd is unable to meet the demand from the image gathering. MIT LL worked with the FAA to provide 'triggers' that would flag a site or image to be analyzed by the workers. The trigger methodology has evolved throughout the research effort and has settled on monitoring for changes in the edge strength ratio and triggering when specific thresholds are crossed. The thresholds chosen are a close approximation of the thresholds for the flight categories. Triggers are also generated when the crowd should revisit the imagery for a site if after a certain length of time there hasn't been any change detected.

Finally, MIT LL continues to evaluate VEIA and develop concepts that will improve its performance as well as add additional functionality. One such example is functionality to monitor the quality of the imagery and report any issues to the system operators. The ability to identify cameras blocked by ice or moved by human intervention or the wind was added to VEIA in the summer of 2021. Possible near-term performance improvements include the potential to cluster imagery based upon the scene 'type' and tune the algorithm to optimize performance for each cluster. Additionally, the focus of VEIA has been on estimating visibility, but several components of the algorithm are potentially useful in providing estimates on cloud cover and/or cloud height.

3. METHODOLOGY

The VEIA algorithm was designed to take advantage of the concepts of edge strength and its relationship with visibility. Using this concept, the VEIA algorithm first computes a visibility estimate for each camera and then combines them into prevailing estimates for each site. VEIA also computes a confidence value for the final site estimate and provides a flag, or “trigger,” when conditions may need evaluation by a human observer. Additional details for each step of the VEIA process are discussed in this section.

3.1 THE RELATIONSHIP BETWEEN VISIBILITY AND EDGE STRENGTH

The main component of the visibility estimation algorithm is the application of an edge detection technique to the imagery. An example of a clear day and low visibility day for Chandler Shelf, AK, is shown in Figure 1. The human eye has the ability to quickly identify a significant number of missing edges in the low visibility example compared to the clear day image. The algorithm does the same by looking for the edges that persist over a long period of time within an image and comparing them to a clear day reference. Persistent edges consist of the horizon, tree lines, roadways, and permanent buildings.

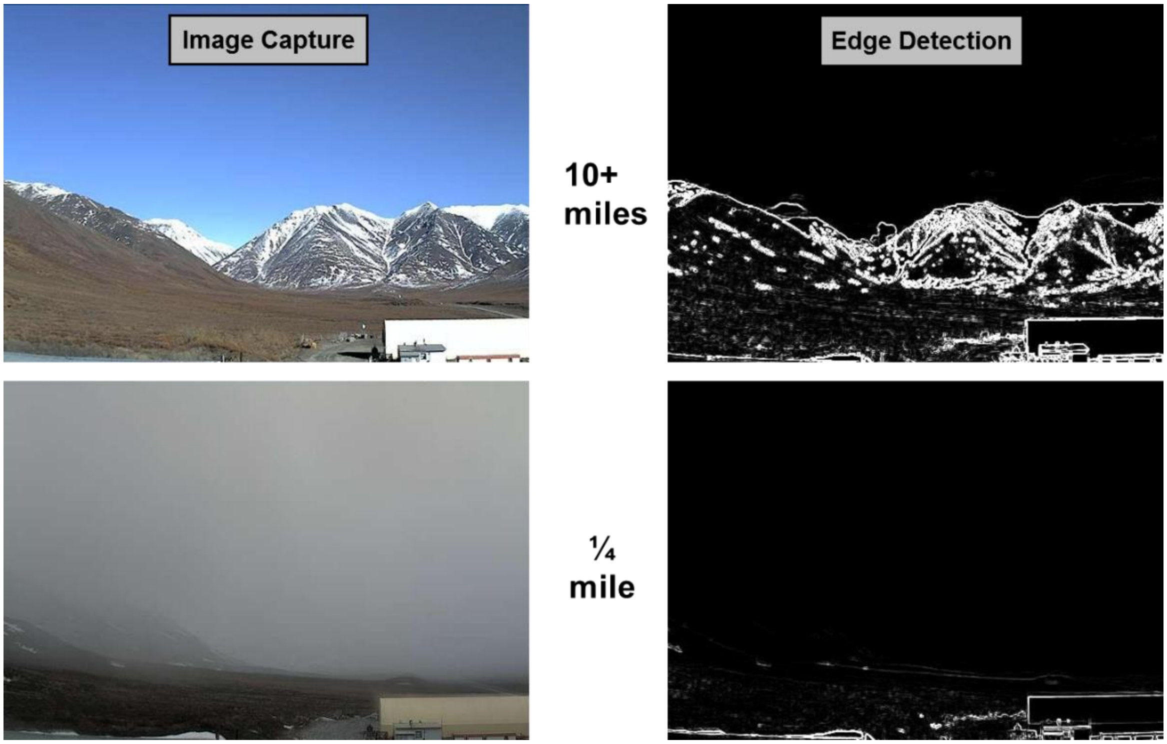


Figure 1. Clear day and low visibility images from Chandler Shelf, AK, and the resultant edges from a Sobel edge detection algorithm. The algorithm uses the number and strength of the edges to estimate the visibility from the imagery.

The VEIA algorithm is built upon the concept that there is a strong correlation between an edge observed in an image (or a marker in a scene) and visibility. This concept is the basis for how National Weather Service observers manually estimate the visibility for their hourly observations. Traditionally, observers are taught to note the presence of a marker (persistent edge) and the known distance to the marker and compare that with many markers in the full 360-degree view. Once a set of markers is identified as visible or not, the observers are taught to use the clarity of the marker to estimate the visibility in between marker distances.

Figure 2 shows an example of three such markers (or persistent edges) in an image from McGrath, AK. The graph shows a strong correlation between edge strength (i.e., clarity of the marker) and surface visibility. The strength of three identified markers are plotted vs. the visibility as measured by a nearby ASOS. The lines represent the median edge strength over a two-year period that correlated to a given ASOS reading. The edge located approximately three miles from the camera shows a roughly linear relationship between edge strength and visibility until the visibility reaches five miles, at which point it levels off. The edge located 0.50 mile from the camera shows a similar relationship, but is more difficult to categorize because it is so close to the camera. The edge located 20 miles from the camera is never visible when the visibility is less than 8 miles, which is expected. This is clear evidence that it is possible for the algorithm to not only note the presence of an edge, and thus a visibility equal to or greater than the marker distance, but also of the usefulness of the edge strength in identifying the visibility between available marker distances. The VEIA algorithm uses this concept, but applies it to the entire scene of persistent edges, not individual hand-selected markers, as shown in Figure 2.

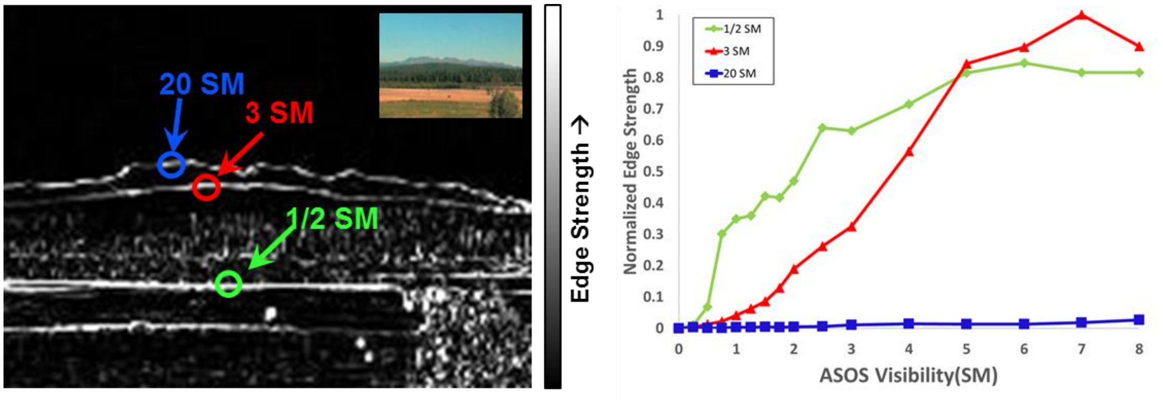


Figure 2. The edge strength is plotted vs. the observed ASOS visibility for three markers (persistent edges) from the northwest-facing camera in McGrath, AK (aggregated for two years of data). The green line is from an edge located on the far side of the runway at approximately 0.5 statute miles (SM) from the camera. The red line is from an edge located along a tree line located at approximately 3 SM from the camera. Finally, the blue line is for an edge located along the horizon on a distant mountain at over 20 SM. The plot shows that the green line is first visible at a half mile ASOS observation and increases in intensity as the visibility improves to around a maximum of 5 SM. The red line has some signal when the visibility is at one mile and increases to maximum intensity at approximately 7 SM.

3.2 COMPUTING INDIVIDUAL CAMERA VISIBILITY ESTIMATES

The individual camera estimates in VEIA are computed using a three-phase process: composite image generation, edges of interest extraction, and translation to visibility, as shown in Figure 3.

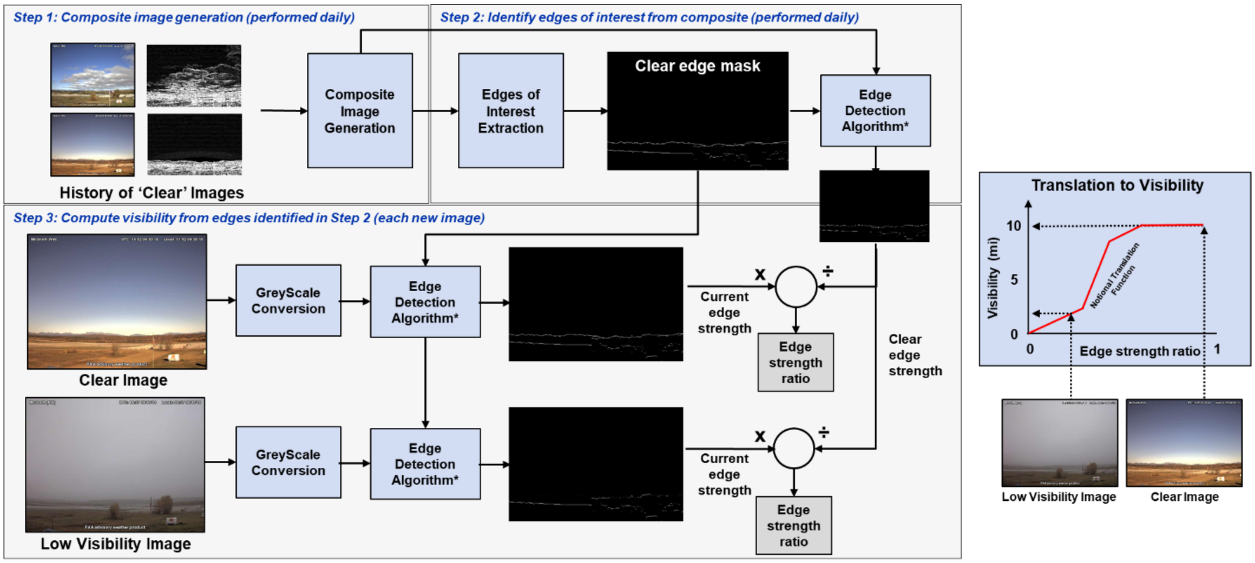

Step 1: Composite image generation (performed daily)
History of 'Clear' Images
Composite Image Generation
Step 2: Identify edges of interest from composite (performed daily)
Edges of Interest Extraction
Clear edge mask
Edge Detection Algorithm*
Step 3: Compute visibility from edges identified in Step 2 (each new image)
Clear Image
GreyScale Conversion
Edge Detection Algorithm*
Current edge strength
x
÷
Edge strength ratio
Clear edge strength
Low Visibility Image
GreyScale Conversion
Edge Detection Algorithm*
Current edge strength
x
÷
Edge strength ratio
Translation to Visibility
Visibility (mi)
10
5
0
0
Edge strength ratio
1
Notional Translation Function
Low Visibility Image
Clear Image

Figure 3. Overview of how individual camera estimates are computed in the VEIA algorithm. The algorithm uses a history of clear images to produce a composite of the persistent edges (i.e., horizon, roadways, buildings, etc.) that are expected during times of unlimited visibility. Then, the algorithm compares the overall edge strength of the current image to that of the clear day composite image to generate an edge strength ratio. The ratio is then converted to visibility in statute miles using a translation function.

3.2.1 Composite Image Generation

The first component, *composite image generation*, involves saving a history of the last several days of images and processing a reference composite image. This composite image represents the best estimate for what a clear day image would look like given recent conditions. The composite image generation filters out images without enough available sunlight (threshold on Sun angle) and those potentially influenced by low visibility conditions. The composite is generated using a 10-day sliding window of images. The 10-day period was selected because it is long enough to include enough clear days and short enough to account for seasonal variation in the scene.

3.2.2 Edges of Interest Extraction

The next component is *edges of interest extraction*. During this process, the Sobel edge detection algorithm is applied to both the current image and the reference composite image. Then a technique is applied on the composite image edges to identify the edges that are persistent (such as the horizon, fixed buildings, etc.). The technique scans each vertical column of the image independently to detect the horizon first, followed by one or more strong persistent edges in each column. This technique removes transient edges such as people, cars, and clouds. This technique also reduces the impact of strong prominent persistent edges from objects in the foreground that can negatively impact the estimation process by overwhelming the translation process.

The proper selection of the edges used within the estimation of the visibility is a critical aspect of the VEIA algorithm. First, it is important to eliminate all transient edges such as vehicles, aircraft, people, clouds, etc. If any of these objects break through the processing, they can corrupt the results by overestimating the visibility when they are present in the current image or underestimating the visibility if they are in the clear day reference image. However, it is also important for the edge selection process to select a set of edges that are equally distributed across the entire visibility range of the image. If all the edges selected are located at the same distance from the camera, then it will be difficult for the algorithm to estimate visibilities through the entire visibility spectrum. For instance, if all edges chosen are 1 SM from the camera, then accurately estimating a visibility of 7 SM compared to 5 SM may be impossible.

As discussed previously, the primary function of creating the composite image is to provide a representation of the clear day image for comparison. This allows the generation of an edge strength ratio between the composite and current image for visibility estimation. The secondary function of the composite image is to identify the non-transient objects. By blending all images over the previous several days, the transient edges are washed out of the image and thus eliminated when the edge detection is applied to the composite image. This meets the first need of the edge selection process. However, the most challenging need in the edge selection process remains, the equal distribution of edges across the visibility spectrum.

To discuss the method chosen to perform the best edge selection, two example images of different scene scenarios are shown in Figure 4. The first image, from Homer, AK, is one typical scene type that is observed in the state of Alaska. In this image, the camera is looking out across an open bay to a set of distant mountains. The camera is mounted high on a pedestal with the added advantage of being on a hillside. The view contains both foreground objects, such as the buildings that are only a short distance from the camera, as well as the distant mountains located more than 10 SM from the camera. This image also contains mid-range features such as the spit (extended stretch of beach that projects out into the sea) that is labeled at 5 SM and roads and a lake at a range of between 1 and 5 SM. There are limited objects between 5 and 12 SM due to the open water. The second image, from McGrath, AK, has different characteristics than the Homer image. First, the farthest distance to a feature is 2.5 SM to a nearby hillside. Therefore, all visible objects are closer than 2.5 SM. The most prominent feature is a tower in very close proximity that is less than 20 feet from the camera.

To evaluate the performance of these two cameras, we can use a receiver operating characteristic (ROC) curve. The ROC curve provides a methodology to compare the performance of the algorithm from multiple input sources while also varying an algorithmic parameter. For these two cameras, the accompanying ROC curve is shown in Figure 4. In this ROC curve, the true positive and false positive for detecting the instrument meteorological conditions (IMC) as compared to a co-located ASOS is shown. For visibility-only measurements, the threshold between IMC and visual meteorological conditions (VMC is 3 SM). The closer the curve is to the top-left corner of the ROC curve, the better the performance.

A comparison of the two ROC curves is shown in Figure 4 using a very basic edge selection process that will be discussed later. The ROC curve shows that the performance of VEIA on the image from Homer, AK, is quite good, even with a basic edge selection method. The curve is very near to the top-left corner and, depending upon the parameters chosen, the true positive rate can be greater than 90% with a false positive rate under 10%. However, for the McGrath camera, the performance is notably lower. For McGrath to obtain a high true positive rate, the false positive rate would be unacceptable. To reduce the high false positive rate to an acceptable level, the true positive rate would be unacceptable. This poor performance is due to the large quantity of edges that are generated from the antenna using a basic edge selection method. The edges from the antenna represent a high percentage of the total number of strong edges. As stated previously, if the large majority of edges are at one distance, the algorithm will be unable to estimate visibility across the entire visibility spectrum. Also, the proximity of the antenna results in the edges associated with the antenna always being present and demonstrating a high edge strength. This results in an edge strength ratio that is always near or above 1.0, and thus a high visibility estimate even when the visibility is reduced to below IMC values.

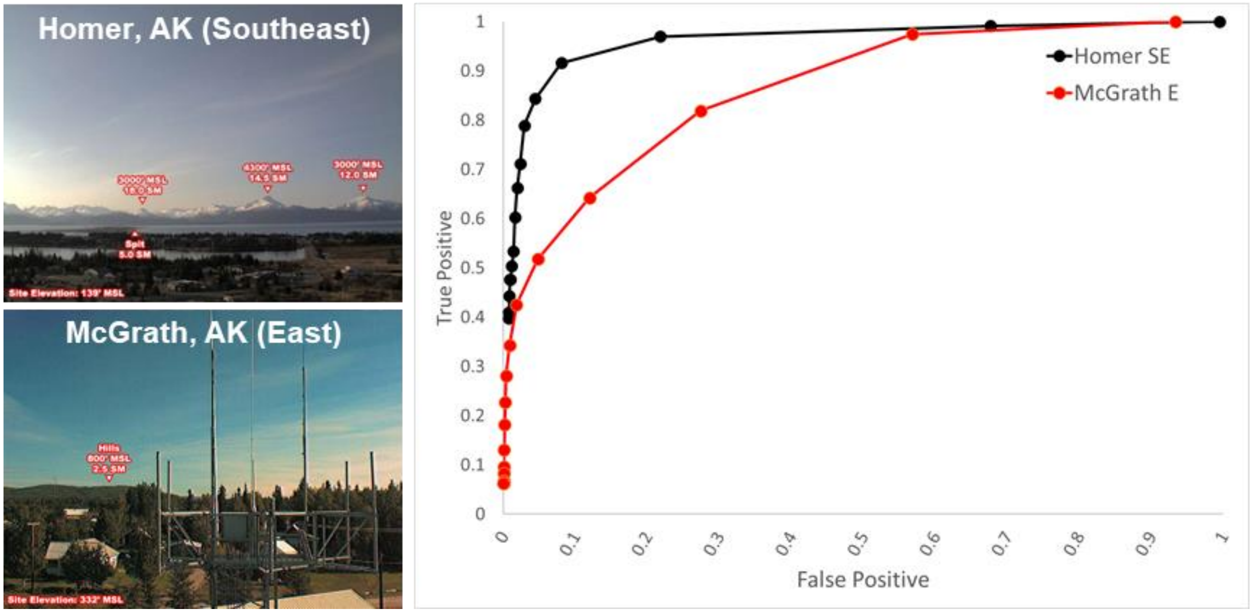


Figure 4. Clear weather annotated images and ROC curves from Homer, AK, and McGrath, AK. ROC curves are comparing algorithm performance for a southeast-facing camera from Homer, AK, and an east-facing camera from McGrath, AK, varying the multiplier applied to a basic edge selection process. The performance is notably poorer from the McGrath camera with a large antenna in close proximity to the camera.

To improve the performance of images with close-proximity large objects, different edge selection techniques were investigated. Ultimately, three methods were fully evaluated over a large set of cameras to identify the best performing method. One basic method computes the median edge strength across all the pixels in the imagery, applies a multiplier to the mean, then performs thresholding on all pixels to identify the strongest edges. This method, using a multiplier of five, is depicted in Figure 5 for the northwest-facing camera from McGrath, AK, in the lower left of the figure. The selected pixels for edge strength computation are depicted in white. The second method consisted of a sliding scale of percentile calculations. In this method, the pixels above a percentile threshold were identified as the pixels for selection in the processing. In Figure 5, the pixels that are stronger than the 99th percentile are shown in the bottom middle image. Therefore, of the 307,200 pixels in the image, the top 1% by strength, or 3,072 pixels, would be used in the edge selection. The final method evaluated was the called the ‘Horizon Plus’ method. The Horizon Plus method begins by finding the horizon using a peak finding process searching vertically (top to bottom) through each of the columns of pixels in an image. During this process, it is assumed that the sky does not contain any peaks. This is generally true when using a composite image because clouds are removed. The first peak is typically a distant mountain range, hillside, or some natural feature along the horizon. In some instances, the first peak may be an object that is above the horizon in the image due to the camera viewing angle, such as a nearby tree or tall building. The algorithm then finds the next strongest peak in the vertical column. In most instances, this may be a nearby non-natural object such as the roofline of a building, a ‘shoulder’ between a roadway and the adjacent grass area, or a nearby ridgeline that is against a background of a more distant mountain range. The algorithm then finds the next strongest peak in a column and continues this process until a set number of peaks is reached. The number of peaks to use is defined as a parameter in the code. In Figure 5, this parameter is set to horizon plus one additional peak.

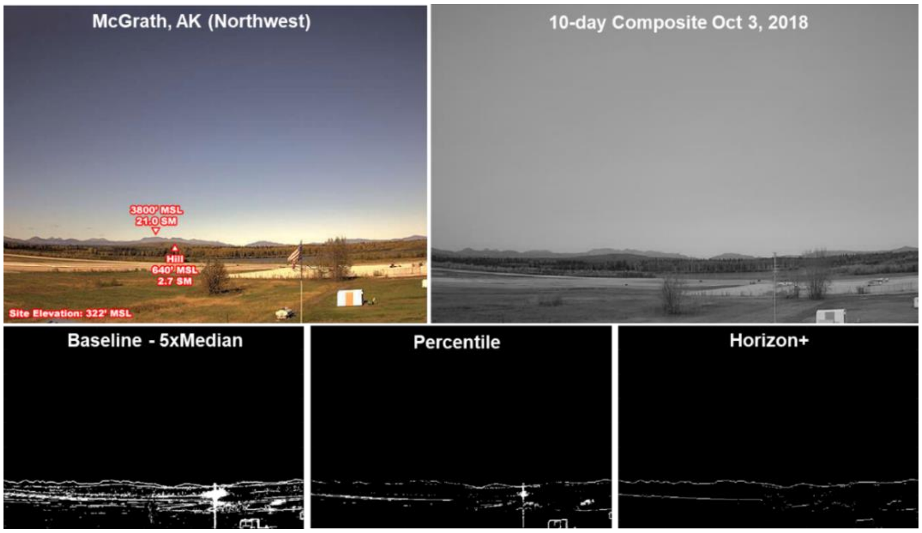


Figure 5. Clear weather annotated image, 10-day composite image, and three edge selection techniques for the northwest facing McGrath, AK, image. The three edge selection techniques consist of the baseline method (five times the median value), a percentile method (99th), and the horizon plus method (horizon plus one).

The difference in the selection of edges is easily observable in the figure. Many edges are chosen in the baseline method. The edges are distributed throughout the image (minus the sky), but are often associated with edges that are difficult for the human eye to distinguish. The performance of VEIA using the baseline method for this image is quite good, which may be due to the equal distribution of these edges through the visibility spectrum. It is also important to note that the flag pole (approximately 100 feet from the camera) is selected along with the flag, albeit the movement of the flag creates many edges in a blur near the top of the pole, which likely negatively impacts performance. The small close-proximity communications building is also easily identifiable in the edges.

The second method, using a threshold from the edge strength percentile, significantly reduces the number of edges selected. Using the 99th percentile as a threshold, of the 307,200 pixels in the image, the top 1% by strength or 3,072 pixels would be used in the edge selection. The blur from the flag is less obvious, whereas the edge of the small communications building is retained. It is important to note that in this method, significant edges are lost as the percentile threshold is increased. A review of the horizon shows that many of the pixels are not selected which may impact performance.

The last method, horizon plus, once again significantly reduces the number of edges selected. Although the percentile method uses over 3,000 pixels, the horizon plus method uses a minimum of 640 pixels from the horizon and, with a parameter setting of one (shown in figure), it uses 1,280 total pixels. For this image, the edge selection clearly picks up all the edges along the distant horizon and the roofline of the small communications building as well as a clearly defined runway edge. The distribution of the edges across the visibility spectrum appears to be quite good while not overwhelming the processing with a large number of nonconsequential edges.

Figure 6 shows the edge selection for the three different methods for the east-facing camera in McGrath, AK. As discussed previously, the antenna presents a challenge for the edge selection process due to the large number of close proximity edges. As can be seen in the baseline method, there are many edges selected from the antenna, with many of them being above the horizon line. These edges are also very strong and have been previously demonstrated to reduce the VEIA performance.

For the second method, using a threshold from the edge strength percentile, there is a significant reduction in the number of edges chosen. However, the selected edges are dominated by the antenna and many of the desired edges, such as the horizon, are lost. Visually, it appears that this method did not result in the desired effect of removing the large, close-proximity antenna. It should be noted (not shown) that by decreasing the percentile parameter various desired edges are included; however, that is done at the expense of increasing the number of edges associated with the large proximity antenna.

For the last method, horizon plus again reduces the number of edges selected based upon the desired number of strong edges in the vertical column (horizon plus one is shown). For this image, the edge selection clearly eliminates most of the edges associated with large close proximity antenna. Meanwhile, the method has selected the horizon and the tops of the evergreen trees along with a collection of the building rooflines. The distribution of the edges across the visibility spectrum appears to be quite good while not overwhelming the processing with a large number of close proximity edges.

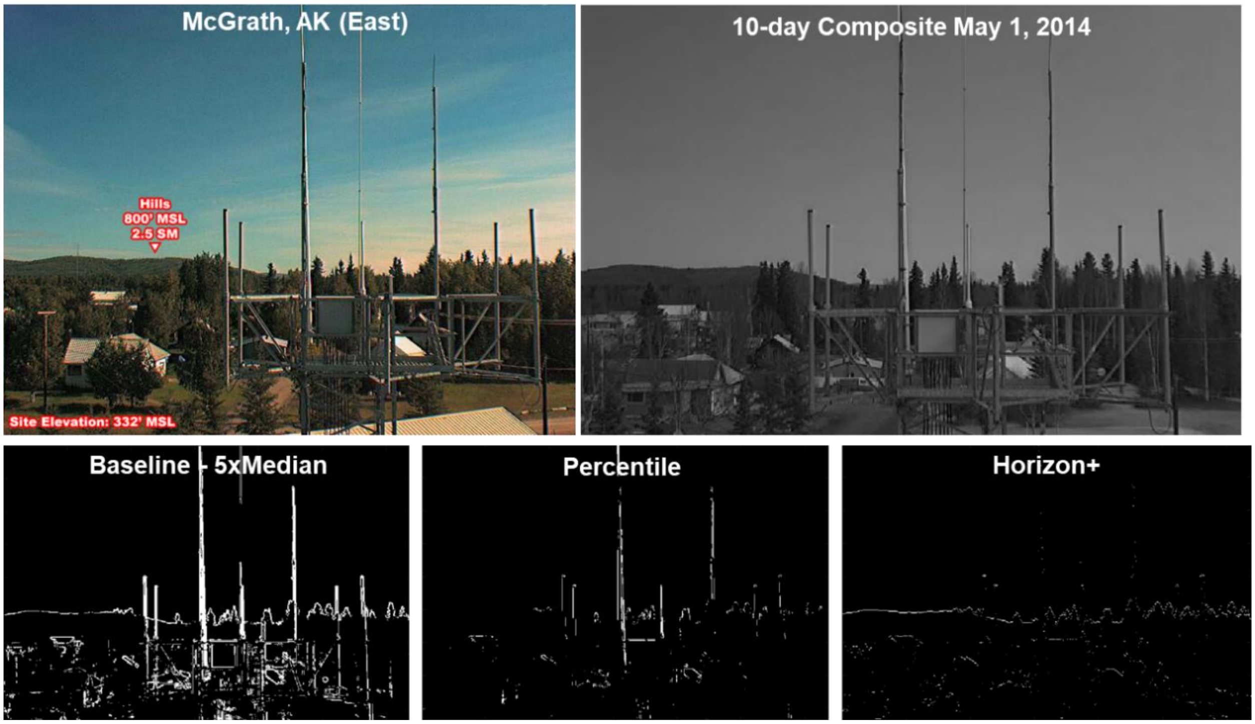


Figure 6. Clear weather annotated image, 10-day composite image, and three edge selection techniques for the east-facing McGrath, AK image. The three edge selection techniques consist of the baseline method (five times the median value), a percentile method (99th), and the horizon method (horizon plus one).

A subjective visual evaluation of the edge selection suggests that the horizon plus one method is choosing the best edges for processing. The goal, however, is to improve the performance of the algorithm for the images with large close-proximity objects while not reducing the performance of images without this observed problem. To evaluate the performance, the algorithm was run with multiple parameter settings using both the percentile method and the horizon plus method along with the baseline method. For the percentile method, the threshold ranged from 50 to 99.9, whereas for the horizon plus method, the parameter ranged from 0 (horizon only) to 10. The results for the east- and northwest-facing cameras are shown in Figure 7.

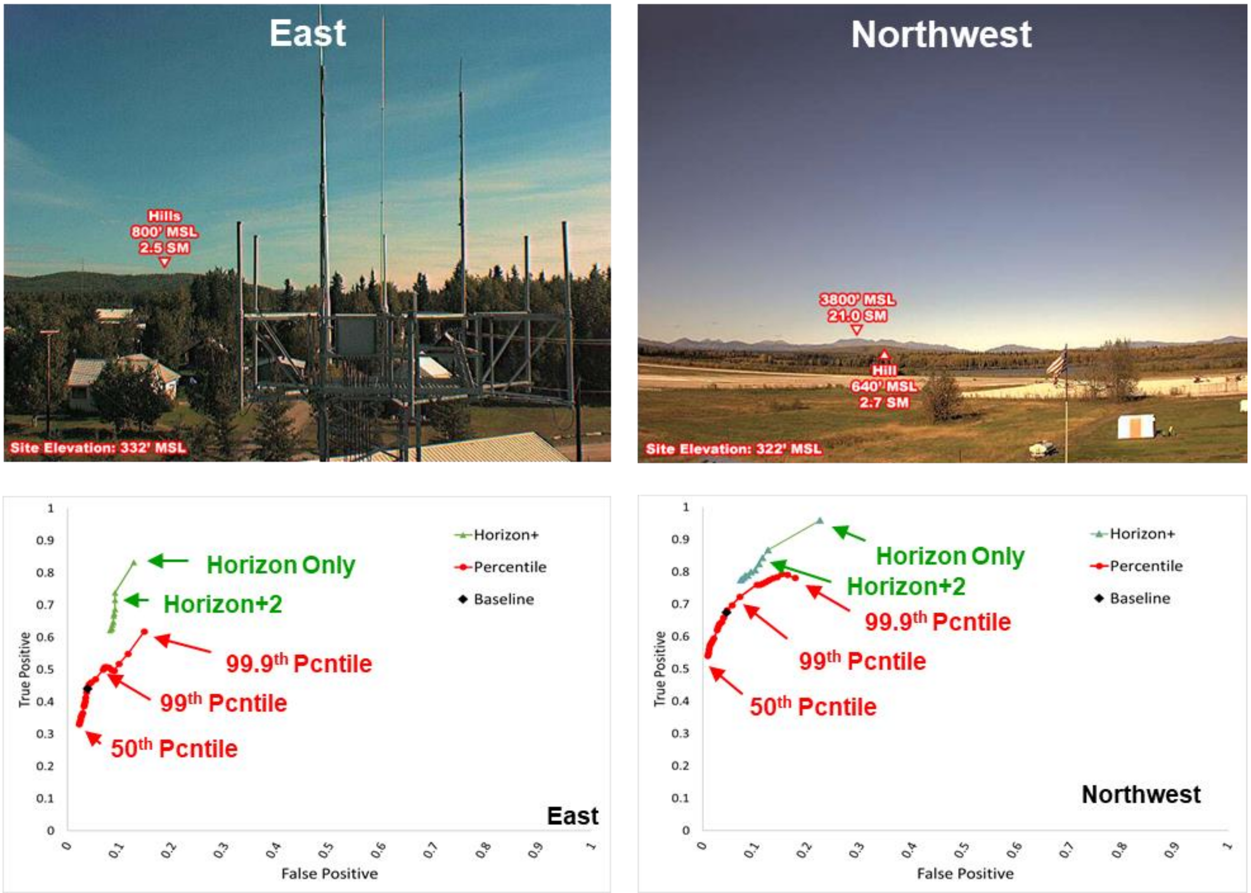


Figure 7. Clear weather annotated image and ROC curves evaluating the performance of the algorithm to identify IMC conditions from the baseline method, percentile method (parameter range is 50 to 99.9), and the horizon plus method (parameter range is 0 to 10).

It can be observed in Figure 7 that the baseline method aligns well with the percentile method when the percentile is roughly 90%. Therefore, the edges selected are very similar in quantity and distribution. With the increased percentile setting (99th), the performance increases regarding the true positive rate while compromising the false positive rate. It can also be observed in Figure 7 that the horizon plus method outperforms the percentile method for the true positive rate with little to no impact on the false positive rate for the east-facing camera. However, there appears to be a reduced performance as measured by the false positive rate for the northwest-facing camera.

The scoring of multiple cameras was conducted comparing the performance of the baseline method and the horizon plus method. The general conclusion by viewing multiple ROC curves is that the horizon plus method performs the best overall. To highlight this observation, Figure 8 plots the camera's true positive vs. false positive rate. The horizon plus method (horizon plus one), shown in green, shows a shift towards the top-left corner of the plot, indicating improved performance. This is encouraging both from the perspective that the visibility estimates for the cameras with large proximity antennas are performing better, and the new method has not negatively impacted the results from the other cameras.

Figure 8 also contains a plot of each camera's true positive rate (red) with the baseline method on the x-axis and the horizon plus method (horizon plus one) on the y-axis. Similarly, the false positive rate (blue) is plotted. For the true positive rate, an upward shift (increasing) shows an improvement, whereas for the false positive rate, a downward shift (decreasing) shows improvement. Therefore, the best performance of the algorithm is obtained with the horizon method using the horizon plus the next strongest peak (horizon plus one).

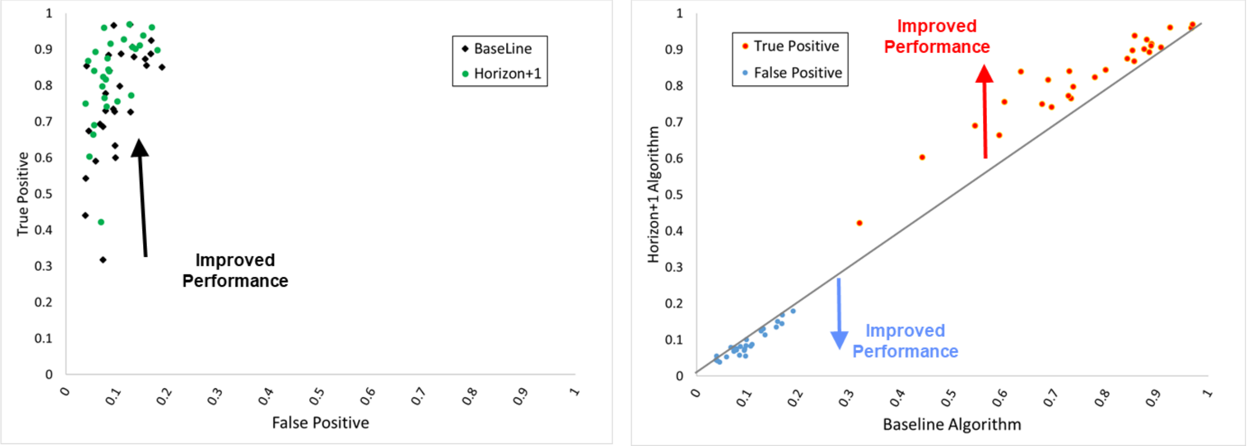


Figure 8. True positive vs. false positive rates comparing the baseline method with the horizon plus method (horizon plus one) for 28 cameras in Alaska.

3.2.3 Translation to Visibility

Finally, the VEIA algorithm estimates the visibility on a per-camera basis by computing the edge strength ratio, which is the sum of the selected edges from the current image divided by the sum of the selected edges from the reference composite image. The edge strength ratio is then converted to visibility in statute miles using a calibrated translation function. The translation function was developed using aggregate statistics that compare the observed true visibility to the edge strength ratio. This was done over multiple years and hundreds of cameras. The same translation function is used for all cameras and is not individually calibrated to the scene. In the future, VEIA may adopt a variable translation function that is tailored by scene type (ocean view, mountain view, city view, etc.).

The translation function is intended to be a single function that is applied to all camera images regardless of the scene or furthest distance in the image. Throughout the development of VEIA, the translation function has undergone several significant updates to reflect the latest evaluation of the relationship between edge strength ratio and visibility. In the initial stages of the development, the translation function was simply a linear function that anchored a ratio of zero to a visibility of zero and a ratio of one to the maximum viewable distance in the image. After more study, this was modified to use a universal constant for the slope of the line, which made an edge strength ratio of one equal to a visibility of six statue miles for all cameras. Eventually, as the archives grew to include a large set of low visibility events and the processing capabilities expanded to a much larger camera set, the translation function was modified to be a set of contiguous linear functions that were ratio dependent. After the completion of the operational demonstration in 2020, the translation function underwent additional changes due to the evaluation revealing accuracy issues in the 3- and 5-mile ranges. It was also noted that the algorithm required re-calibration in the 5- to 10-mile range.

The translation function was created by performing an analysis at 20 sites with closely co-located ASOS observations provided at a five-minute update rate. This included 80 cameras with a wide range of camera views, such as images with distant mountain ranges, to cameras with nearby hill sides or tree lines, to cameras with some proximity blockage.

A notable problem with defining the translation function to an ASOS-like visibility estimate is the very different measurement techniques of the VEIA and ASOS instrumentation. The ASOS is looking across a 1-meter distance and relating the refraction of light to visibility, whereas VEIA is measuring visibility across a wide distance. If the ASOS and cameras are mounted side-by-side, and the true visibility is extremely low at the sensors (<1 mile), then agreement between the two methods would be expected. However, as the true visibility at the sensor location increases, the difference between VEIA and ASOS may become quite large. One example would be a camera looking out a short distance to a mountain range that is completely blocked due to low clouds sinking down the side of the mountains. VEIA may be reporting a low visibility for the area, whereas the ASOS, sitting in the valley, would be reporting visibility at 10+ SM at the sensor location.

A second notable problem is the ASOS limitation of only reporting visibility out to 10 SM. A human observer may document the visibility to be 20+ SM if a distant mountain range is clearly visible at 20 SM; however, the ASOS in this scenario would report 10 SM. Since VEIA assesses the scene out to the same distances as the human observer, it could potentially go beyond 10 SM. However, since VEIA is calibrated to the ASOS, the performance beyond 10 SM would be questionable.

Figure 9 plots the relationship between the ASOS visibility measurements and the edge strength ratio. The filled circles represent the median edge strength ratio for each ASOS visibility increment for each of the 80 cameras used in the analysis. The black diamonds represent a median across all cameras. A red line depicts the initial translation function, which was a simple linear relationship fixing the zero-edge strength ratio to zero statute miles and an edge strength ratio of one to a visibility of six statute miles (the red line maintains this linear function above one to infinity). The black line depicts the updated translation function that was in use during the 2020 operational demonstration. This function closely mirrors the initial translation function to an edge strength ratio of 0.45, and then adjusts the slope of the function for ratios greater than 0.45. Therefore, between 0.45 and infinity, the visibility increases at a more significant rate as the edge strength ratio increases. This is likely due to a small fraction of the edges that were selected being in the range between 3 and 9 SM.

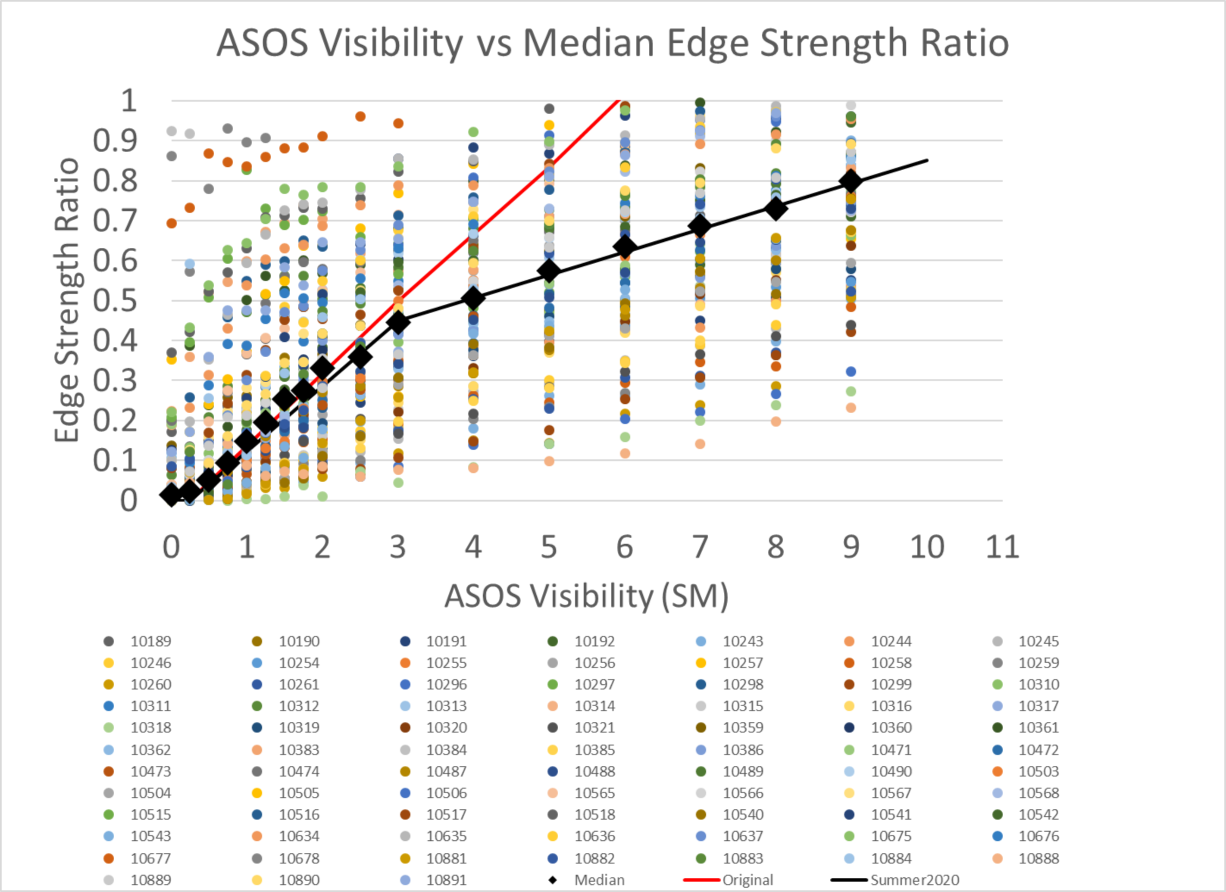


Figure 9. Relationship between the edge strength ratio and visibility for 80 cameras across 20 Alaska sites with a closely co-located ASOS. The median edge strength ratio for all observations at the specific ASOS visibility is plotted for all cameras in the circles. The median edge strength ratio for all cameras is plotted in the large black diamond. The original (linear from zero to six statute miles) translation function is plotted as a red line. The updated 2020 translation function is plotted as a black line.

The translation function was updated again after the 2020 operational demonstration. From subjective analysis during the evaluation, it was believed that the VEIA estimations were too low when ASOS was reporting in the 3 SM to 10 SM range. One commonly observed condition in the camera imagery was a backlighting effect when the Sun would be in the imagery and thus washing out the edges, resulting in a low visibility estimate and thus a low edge strength ratio during times with a high visibility. Another commonly observed condition was very sharp edges when the perfect lighting would occur. This would create a very high edge strength ratio associated with very high visibilities (much greater than the ASOS 10 SM limitation). In order to eliminate issues with the lighting, the cases used in the analysis were limited to times when the Sun was behind the camera and there was a cloud cover present with cloud bottoms at or below 5,000 feet. By doing this, the most optimal lighting conditions are chosen, much like when a photographer uses a ‘flash’ to illuminate the scene to highlight the features. Also, by choosing observations with a low cloud cover, this eliminated the perfectly blue-sky conditions where the visibility is most likely close to infinite.

Figure 10 plots the relationship between the ASOS visibility measurements and the edge strength ratio after filtering the cases. This analysis was performed on 28 cameras from 7 sites. The filled circles represent the median edge strength ratio for each ASOS visibility increment for each of the 28 cameras used in the analysis. The black diamonds represent a median across all cameras. The black line depicts the translation function used during the operational demonstration in 2020. The blue line depicts the updated translation function used during the 2021 user assessment. This function closely mirrors the previous translation functions to an edge strength ratio of 0.45, and then adjusts the slope of the line for values between 0.45 and 0.60. The slope is adjusted again above 0.60. Once again, the visibility increases at a more significant rate as the edge strength ratio increases. For example, an edge strength ratio of 0.6 would have translated to a visibility of 4 SM in the original version, to 6 SM in the summer 2020 version, and to 9 SM in the final version. It is also important to note that an edge strength ratio of 0.3 would remain virtually unchanged in all three versions.

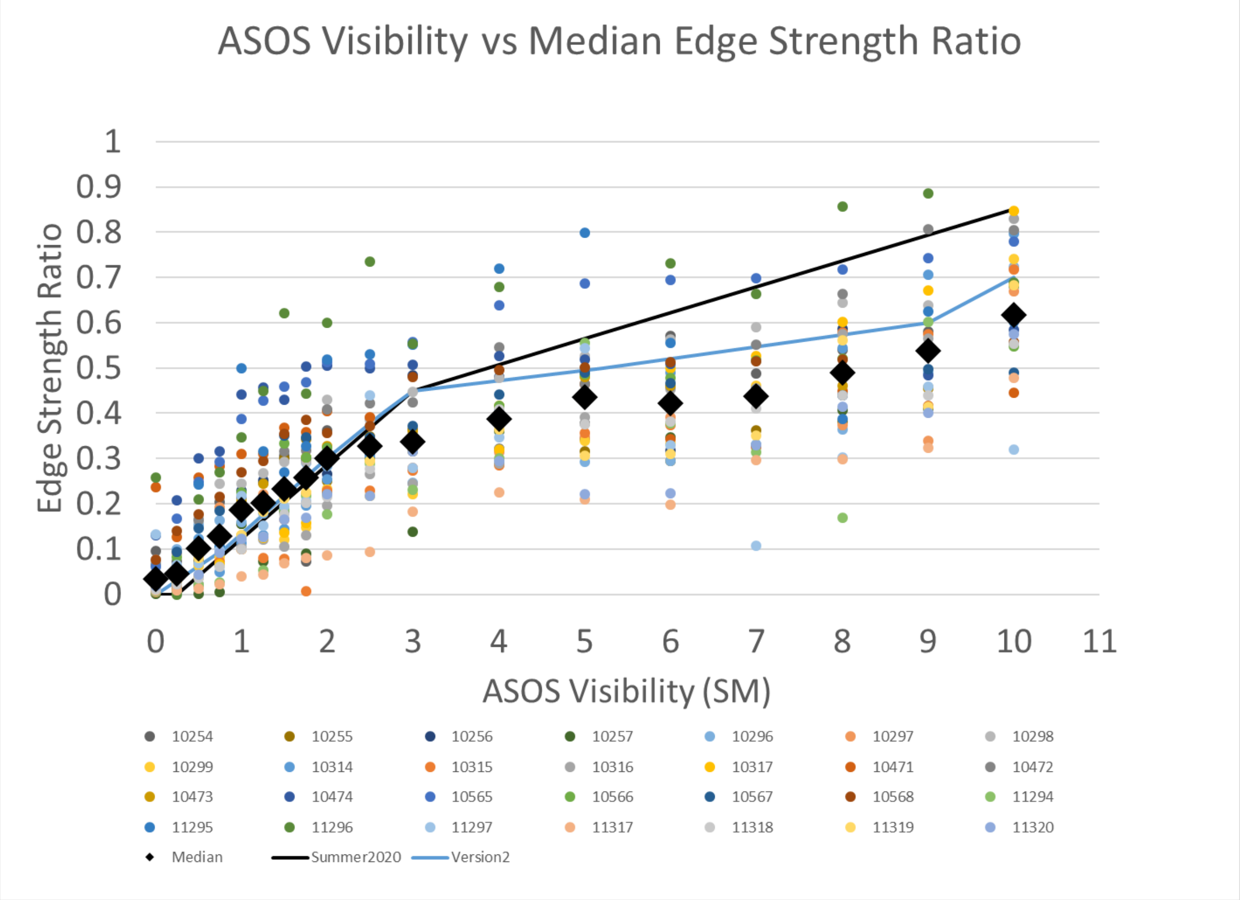


Figure 10. Relationship between the edge strength ratio and visibility for 28 cameras across 7 Alaska sites with a closely co-located ASOS. The cases are filtered to only include observations were the Sun is behind the camera and there is a cloud cover present with cloud bottoms at or below 5,000 ft. The median edge strength ratio for all observations at the specific ASOS visibility is plotted for all cameras in the circle. The median edge strength ratio for all cameras is plotted in the large black diamond. The updated 2020 translation function is plotted as a black line. The final 2021 translation function is plotted as a blue line.

3.3 CREATING A SITE ESTIMATE

The current installation of the FAA Aviation Weather Cameras consists of between two and four cameras at each site. Most sites have the full complement of four cameras. The main purpose of this is to allow users to view the weather conditions in multiple directions. Although VEIA is capable of estimating the visibility on a per-camera basis, failure mode analysis has shown that solar glare causes false reports of low visibility. Furthermore, when solar glare is present in one camera, the accompanying cameras at the same site are generally performing quite well. Therefore, the development team designed an algorithm to combine the input from multiple cameras at a single site into one visibility estimate, which reduces the impact of any cameras experiencing the solar glare issue. Figure 11 is a high-level flow chart for the process of combining multiple cameras at a single site into one site estimate.

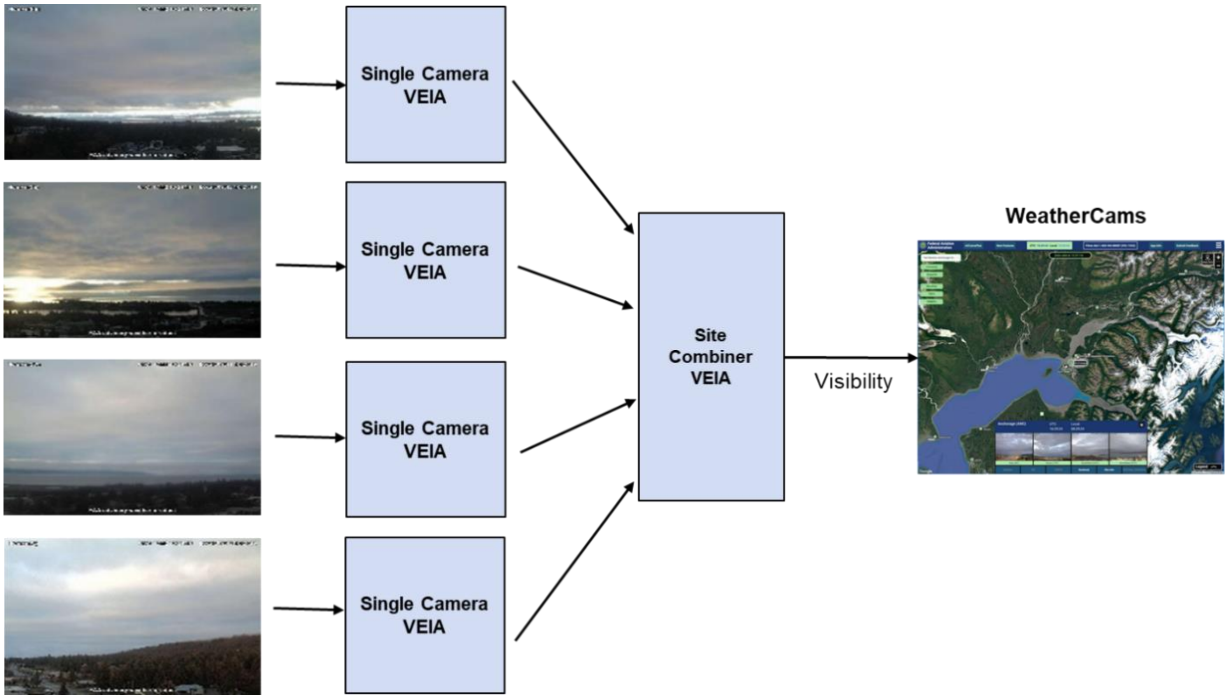


Figure 11. High-level flow chart showing the processing camera imagery from the same site into one site estimate for display on the WeatherCams system.

The primary functionality of the site combiner is to weight each camera based upon known limitations in the performance due to the position of the Sun relative to the camera lens. Figure 12 depicts the weighting scheme for considering the solar elevation and azimuth. A camera facing directly into the Sun will be highly impacted by solar glare and its weighting would be zero, whereas a camera with the Sun located behind it will have the best lighting and would receive the most weighting. The weighting would be reduced proportionally as the Sun angle moved to the left or right side of the camera view. The solar elevation is also considered, but only as a threshold at which point to turn the camera on or off (0 or 1) when the Sun is too low in the sky to illuminate the scene. The solar elevation threshold is set to 15 degrees above the horizon. Since we are combining the estimates for four cameras located at the same site, the solar elevation impacts all cameras equally and is only evident during sunrise and sunset, when the Sun is passing through the solar elevation threshold.

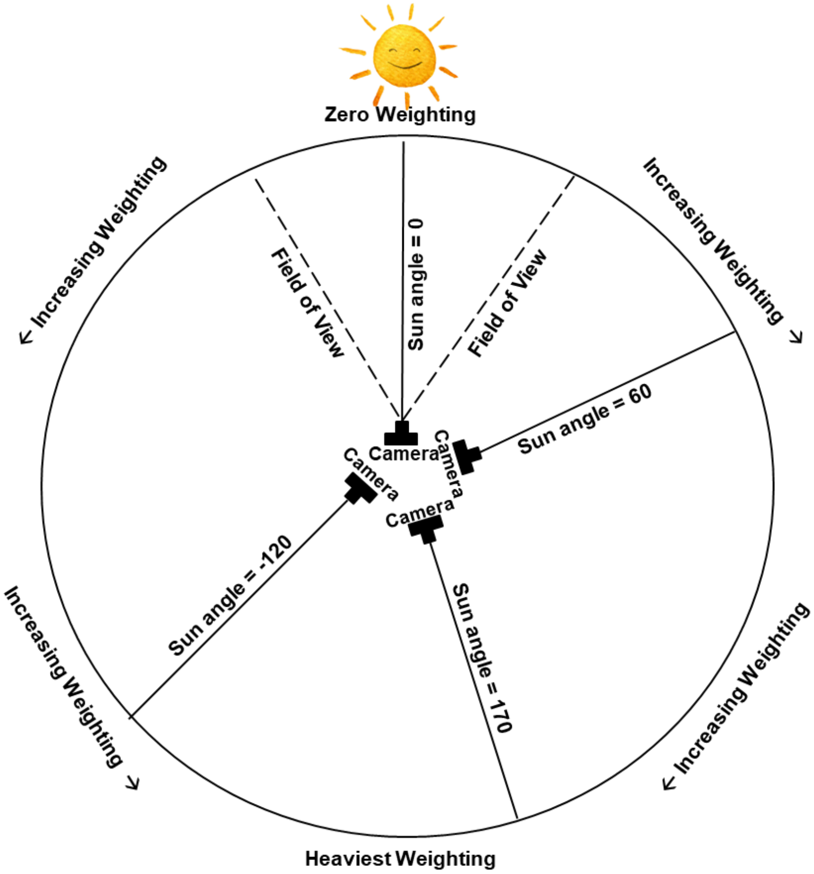


Figure 12. Weighting scheme for a camera when combing multiple cameras into one site estimate. Cameras facing toward the Sun are weighted less than cameras with the Sun located behind them.

Another factor used in the weighting scheme is the edge strength ratios themselves. When all four cameras have an edge strength ratio below a certain threshold, they are all equally weighted. This approach is chosen because during times of heavy fog, the solar glare is no longer a concern because the Sun is being filtered out by the fog and/or low clouds. As suggested by the translation function analysis, with a sharp shift when the edge strength ration is greater than 0.45, the threshold chosen to shift to equal weighting is 0.4.

To evaluate the combined site estimates, a ROC curve is used to evaluate the results of the four camera estimates compared to the site estimate. Figure 13 is a comparison for Homer, AK. Homer is a small city on Kachemak Bay on Alaska’s Kenai Peninsula. The camera installation in Homer sits up on the hill in the city looking down upon the airport and the bay. A large spit, or long strip of land, extends out into the bay and provides excellent data for the visibility estimation. Three of the cameras have distance mountain ranges past the bay and one camera looks towards the nearby hillside. The four cameras provide excellent high-quality imagery for visibility estimation with some limitations due to the extent of the water and the proximity of the nearby hillside. The ASOS in Homer, AK, is offset by more than a mile from the camera site with a significant orographic effect. In Figure 13, the site estimate is shown as a black line, and each camera estimate is shown in red (northeast), blue (southeast), green (southwest), or yellow (west). It is important to note that the plot is zoomed in to a range of 0.5 to 1 for the true positive rate and 0 to 0.5 for the false positive rate. If any of the results fall outside of this range, the data are not plotted. It can be observed in the figure that by combining all four cameras into one estimate using a weighting scheme based upon the Sun angle and a threshold on the edge detection ratios that the performance is improved.

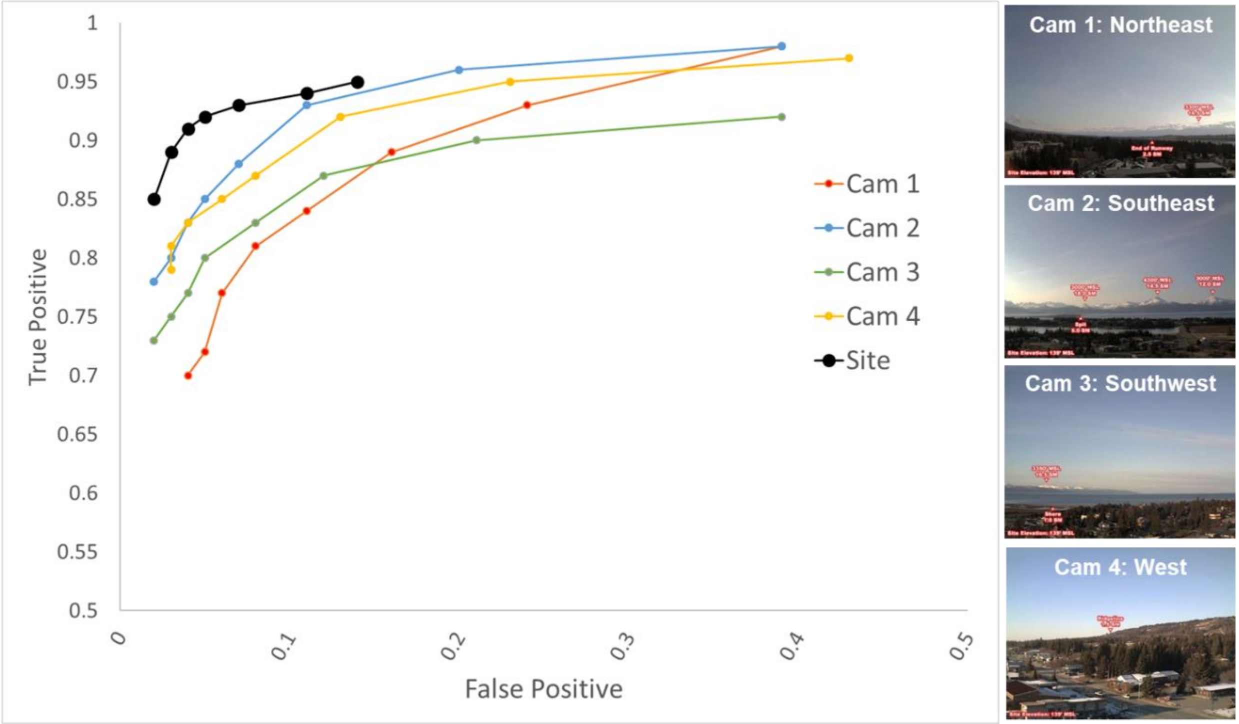


Figure 13. ROC curve comparing performance of algorithm to identify IMC conditions for four cameras located in Homer, AK, and the combined site estimate (shown in black) along with annotated clear day images. The percentile edge selection method is shown varying the percentile for comparison. The four camera-only estimates are below the site estimate, indicating that the performance is improved when all four cameras are combined using a weighting scheme based upon the Sun's location relative to the view of each camera.

Figures 14 is like Figure 13, but for Cordova, AK. Cordova is a small city at the mouth of the Copper River, at the head of Orca Inlet in the east side of Prince William Sound. The camera installation in Cordova sites at the local airport looking down the runway and at the apron. Three of the cameras have obstructions from nearby structures. The ASOS in Cordova, AK, is located close to the camera installation. It is also observed that the combined site estimate out performs any of the camera images individually. Of special note are the impacts in the east and northwest cameras. The east and northwest cameras are both underperforming because their views contain very few mid-range edges. They are dominated by close-proximity ground clutter in the foreground and the distant mountains that make up the horizon. A simple average may allow the east-facing and the northwest-facing camera to negatively impact the site scoring.

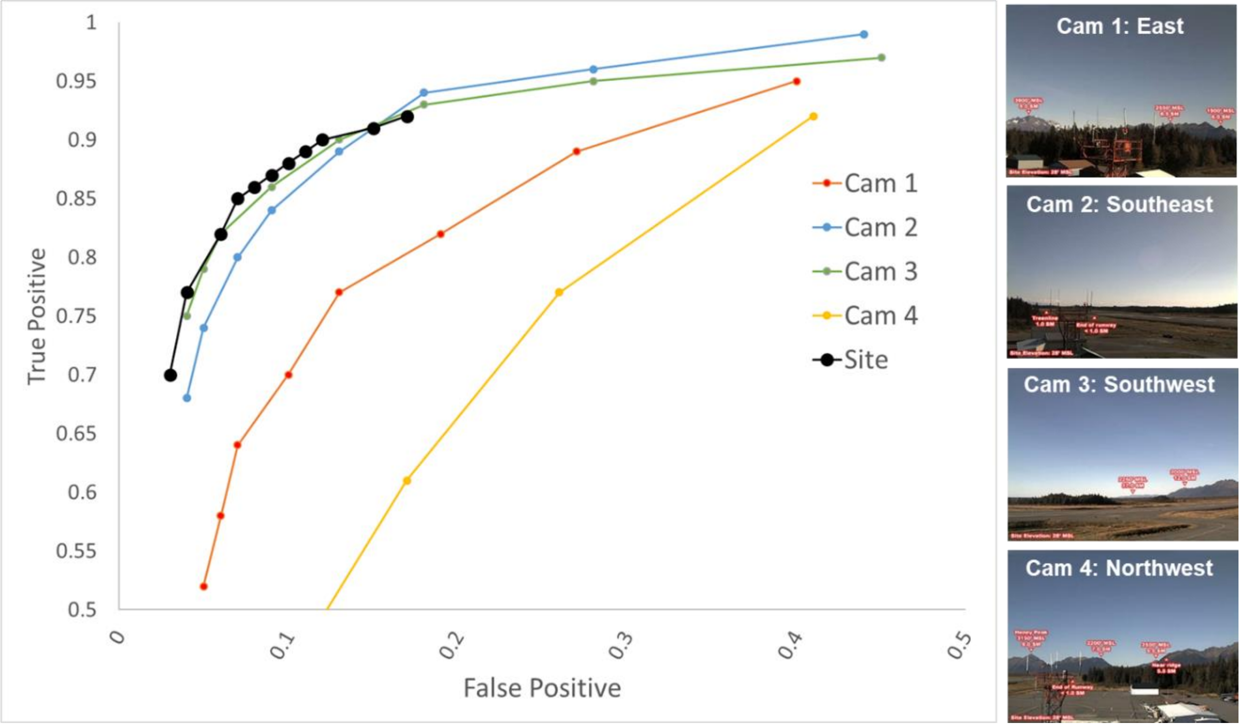


Figure 14. ROC curve comparing performance of algorithm to identify IMC conditions for four cameras located in Cordova, AK, and the combined site estimate (shown in black) along with annotated clear day images. The percentile edge selection method is shown varying the percentile for comparison. The four camera-only estimates are below the site estimate indicating that the performance is improved when all four cameras are combined using a weighting scheme based upon the Sun's location relative to the view of each camera.

Figure 15 compares the performance of the combined site estimate for eight sites in Alaska. In all instances the overall performance is within an acceptable range. The best performing site is Palmer, AK, where the ASOS and camera are near each other and the camera images provide a view that is free of large close-proximity antennas and buildings while still having nearby objects to use in edge selection. This is combined with views containing distant mountain tops and ridgelines that are excellent for estimating higher visibilities. The worst performing sites are Ketchikan, Cordova, and Barrow. Each of these sites has challenges that have been previously noted, from large distances between the ASOS and the camera site to the overabundance of close proximity objects to the opposite effect of too few objects in the foreground.

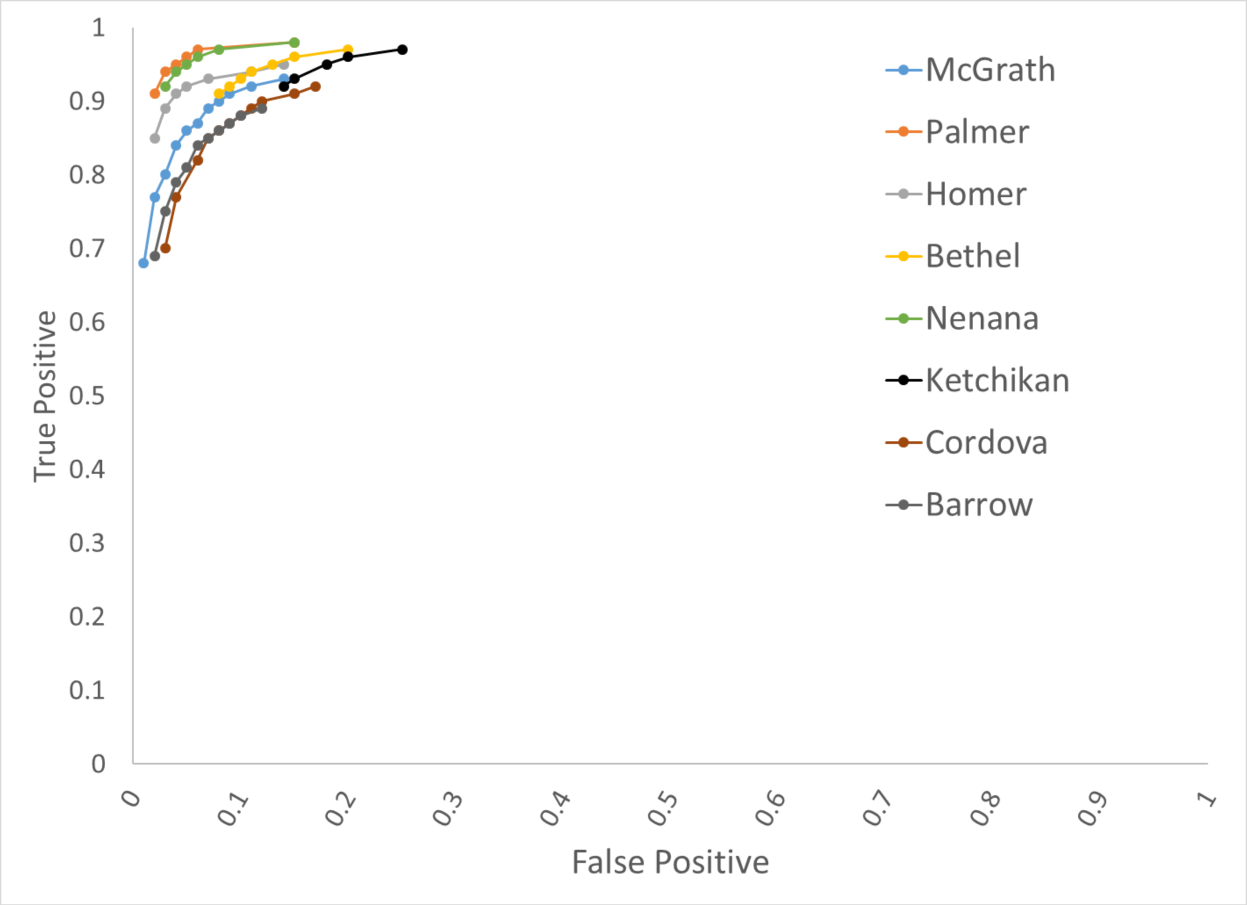


Figure 15. ROC curve comparing performance of algorithm to identify IMC conditions for the eight Alaska sites. The percentile edge selection method is shown varying the percentile for comparison.

As a last demonstration of the algorithm abilities using a combined site estimate, two different types of events are shown in Figure 16 and 17. The first event was on 17 April 2019, when a snow squall approached Homer, AK, and reduced the visibility. Prior to the event, each of the cameras are producing very different estimates, albeit all above IMC thresholds. The algorithm correctly merges the estimates into a site estimate matching the ASOS of 10 SM (the maximum visibility allowed). As the event approaches, the camera estimates begin decreasing prior to the ASOS indicating decreasing visibility. This is consistent with the algorithm capturing the prevailing visibility while the ASOS is measuring the visibility over a very small domain (1 meter) at the site location. As the squall line moves over the sensor site and begins to impact all cameras, the visibility drops below 3 SM.

The second event was on 4 March 2018 in McGrath, AK. This event began in the morning with a low layer of fog sitting over the site. As the heating of the day occurred, the fog burned off and revealed excellent visibility with clear skies. As the day progressed, a low cloud deck moved back over the site. Once again, the algorithm correctly estimated the visibility matching the ASOS as the Sun rose revealing the layer of fog, and it correctly detected the burn off and subsequent impacts on visibility of distant mountain ranges.

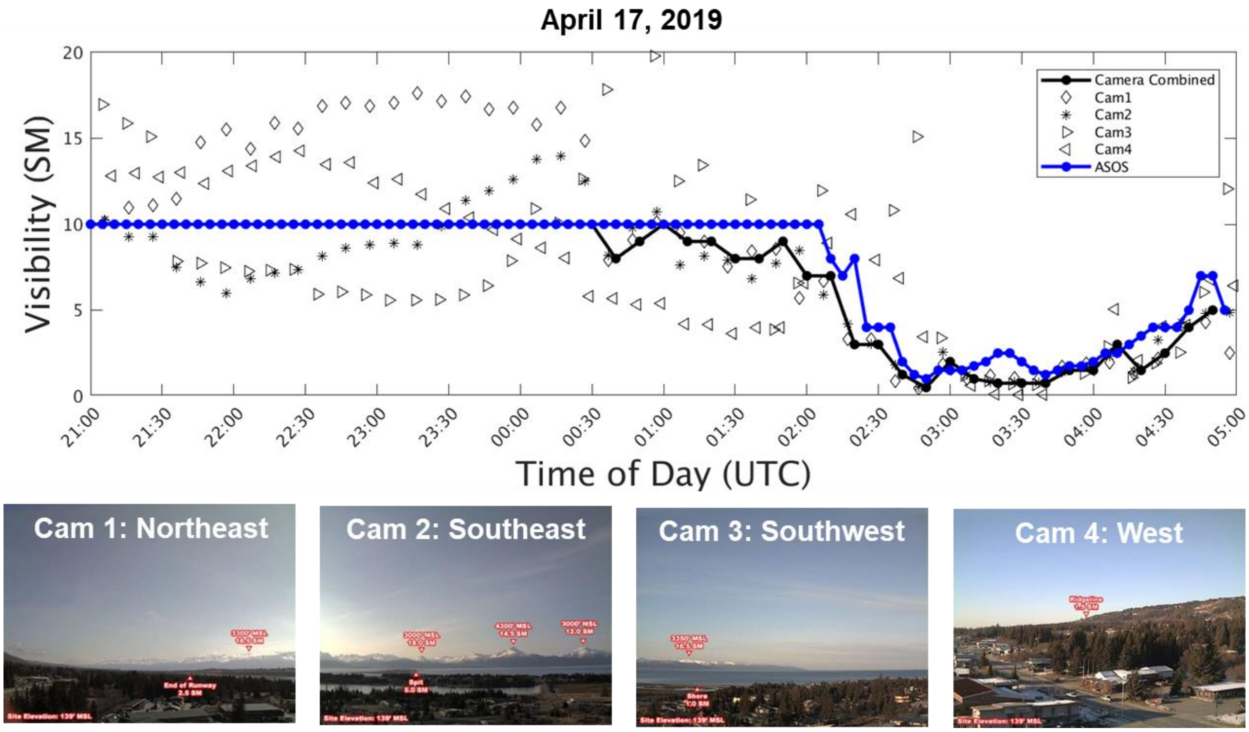


Figure 16. Comparison of visibility estimates over an eight-hour period from Homer, AK, on 17 April 2019 along with the annotated clear day images. The reduced visibility event began in the late afternoon around 01:00 UTC as a snow squall moved in from the northeast. The event was captured in the imagery, observed by the ASOS, and correctly detected in the combined site estimate.

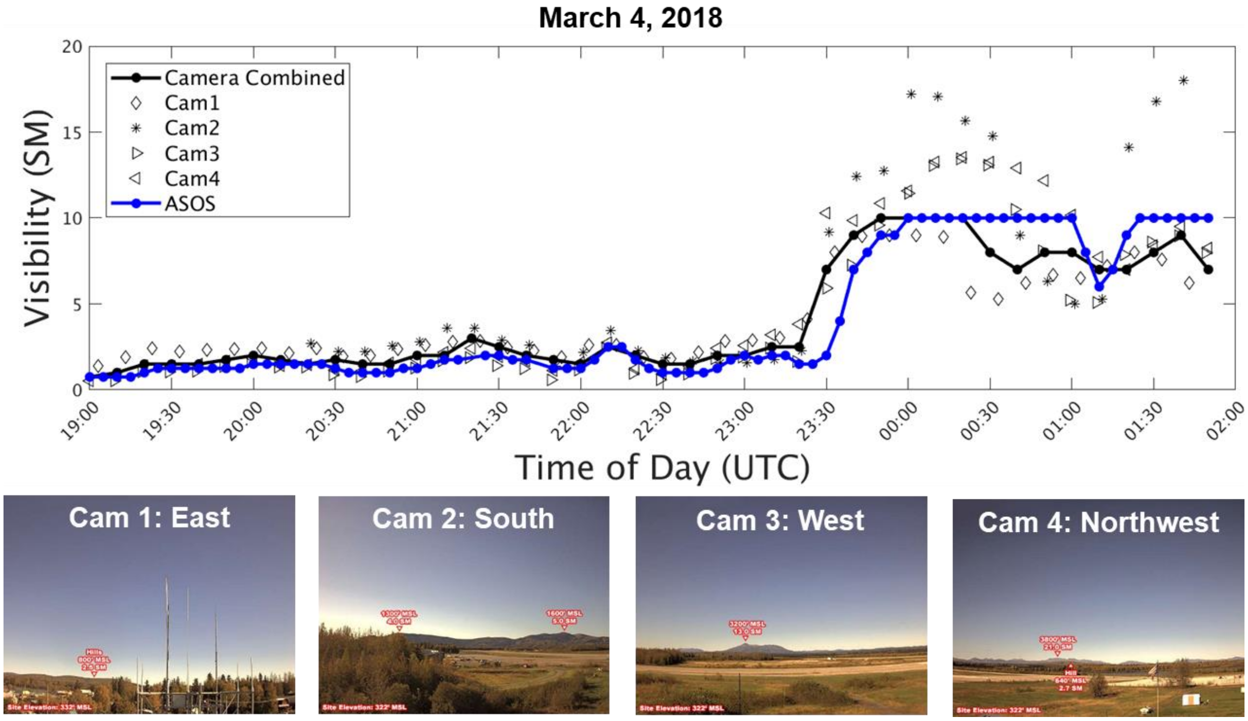


Figure 17. Comparison of visibility estimates over a seven-hour period from McGrath, AK, on 4 March 2018 along with the annotated clear day images. The day began with low fog and clouds over the region which burned off around 23:30 UTC. The event was captured in imagery and observed in ASOS measurements. The visibility estimation algorithm correctly detected the burn off in each of the camera estimates and in the combined site estimate.

3.4 CONFIDENCE MEASURES

As described above, combining the co-located cameras into one estimate is accomplished by using information on the location of the Sun relative to the direction the camera is facing. This information can also be used to develop a confidence measure. The VEIA confidence index is intended to provide a user a sense of the quality of the site-based visibility estimate when combining multiple cameras. The VEIA confidence index also attempts to account for known limitations in the VEIA performance due to the number of cameras, lighting from the Sun, and agreement between the available single-camera estimates.

The confidence index is assigned a value between 0 and 100 and is computed from three separate measures of performance. The first measure is based upon the Sun angle—the higher the Sun is in the sky, the more reliable the estimate is expected to be and the higher confidence value assigned. For this first component, when the Sun is more than 15 degrees above the horizon, the confidence is assigned a value of 100. When the Sun is at the horizon the confidence is assigned a value of 50. A linear relationship is assigned in the range between the two thresholds. The second measure is based upon the relative agreement between cameras using the standard deviation. A value of 100 is assigned when the standard deviation is zero and decreased as the standard deviation increases. Currently, for each standard deviation the value is decreased by 10 points. The final component is the number of cameras available. If all four cameras are available the confidence is set to 100; with only three cameras the value is 75; and with only two cameras the value is 50. Combined co-located estimates of visibility are not provided if only one camera is available. The final step is to combine the three confidence measures into one value. For the VEIA site confidence metric, the minimum of all three components is assigned. If the confidence index is below 26, then a visibility estimate is not provided for the site.

It is important to note that these concepts and values were developed and assigned through a subjective analysis.

3.5 TRIGGER INPUT FROM A HUMAN OBSERVER/CROWD

Another component of the VEIA algorithm is the creation of a ‘trigger’ to support a crowd-sourcing effort funded by the Weather Technology In the Cockpit (WTIC) program. The crowd-sourcing concept involves asking human workers (both trained and untrained) to evaluate the images and provide a visibility estimate. The crowd-sourcing algorithm then combines the set of estimates into a final visibility value based upon historical worker performance and statistical techniques. MIT Lincoln Laboratory has contributed to this effort by participating in the development of a ‘hybrid’ system between the VEIA algorithm and the crowd sourcing. This has included providing VEIA visibility estimates for inclusion in the crowd sourcing, performing analysis on the crowd estimates and crowd-sourced results, as well as creating the software and implementing a trigger to determine the most appropriate time to send images to the human observers in the crowd.

The collaboration with WTIC and WeatherCams has led to a hybrid concept consisting of VEIA working in conjunction with crowd sourcing to provide the best possible visibility estimates. The backbone of the system is VEIA, performing continuous automated visibility assessments using image processing techniques on the entire sensor suite operated by WeatherCams. VEIA also monitors for times when the conditions are changing rapidly. At those times, VEIA would ‘trigger’ the crowd to perform an independent assessment of the visibility conditions. Without VEIA, the crowd sourcing would become overwhelmed with the number of images being collected at WeatherCams in a short time period.

The main component of the trigger is the identification of changes in the imagery that may be indicative of changing conditions. As previously discussed, built into the VEIA is a component to compare the current image with a clear day proxy generated from several days of images combined into a composite. This comparison results in an edge strength ratio that is then converted to the visibility. Changes to the edge strength ratio represent changes to the weather conditions in the scene. Therefore, the first component of the trigger algorithm is to monitor for changes in the visibility estimates and trigger when specific thresholds are crossed. The thresholds chosen are based on the flight categories. The second component of the trigger is a timer that solicits crowd input if there hasn’t been any change detected after a certain length of time. The timer is currently set to trigger every 24 hours if no change is detected.

A final contribution to the crowd sourcing effort was participation in the creation and evaluation of supplemental information to be collected from the crowd. It was envisioned that the human observers in the crowd could inform the VEIA algorithm of conditions that would impact the performance of VEIA, such as the Sun directly in the image, thick cloud cover, snow covering the ground surface, and problems with the imagery.

Evaluation of the trigger mechanism for a nine-day period in September 2021 is shown in Figure 18. This analysis has shown that the median number of triggers across all sites at each update (every 10 minutes) is between five and eight depending upon the dominant weather conditions on any given day. An early morning spike occurs that is associated with the improving performance of VEIA as the lighting from the Sun fully illuminates the view and reveals a significant difference between the visibility conditions at sunrise as opposed to sunset several hours earlier. This early morning spike is typically between 10 and 20 triggers per update period. Whether these numbers are acceptable or not is yet to be determined, as results from an evaluation conducted by Rockwell Collins were unavailable at time of publication. However, it is expected that further reduction may be required. Additional methods to reduce the trigger may require developing techniques that better determine when VEIA is underperforming and requires crowd source assistance.

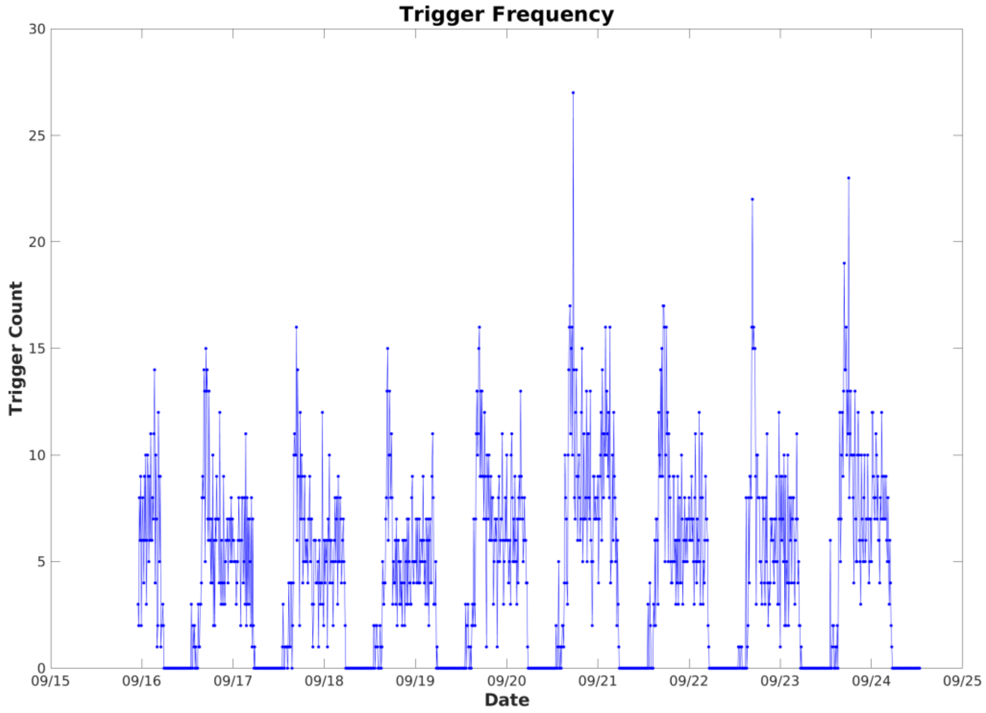


Figure 18. Number of triggers occurring at each VEIA ten-minute update for all available sites. Nominally, there are between five and eight triggers at each update depending upon the predominant weather conditions. An early morning spike can be observed as the Sun rises and the morning visibility conditions differ from the conditions at sunset.

A potential method to further reduce trigger frequency may be to prioritize the triggers based upon the VEIA visibility estimate at the time of the trigger. It may be more important to prioritize low visibility triggers vs. high visibility triggers. Figure 19 shows the current visibility for triggers that occurred over the nine-day period. Of the 5,183 triggers during this time period, 1,628 are when VEIA is estimating a 10+ SM visibility. Since VEIA is very reliable in the highest visibility estimates, these triggers may be of a lower priority. These triggers are also when VEIA is transitioning to a higher visibility value, which may be less important than when VEIA is transitioning to lower visibilities. Further analysis on the specific circumstances in each of these visibility thresholds may also be required.

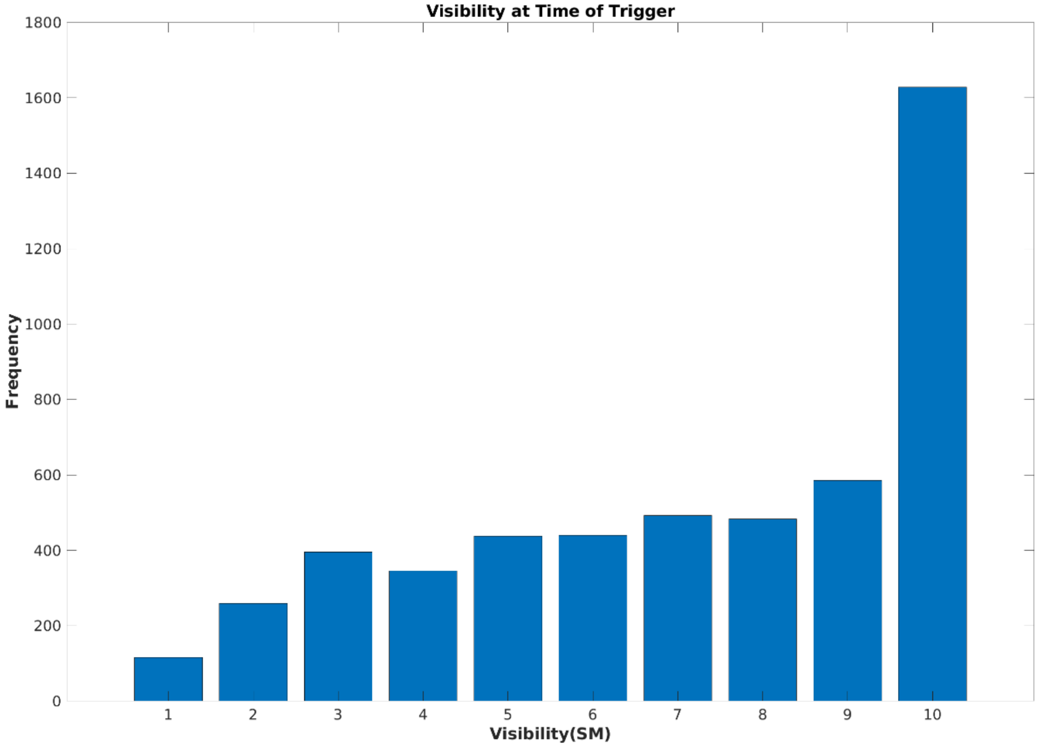


Figure 19. Trigger counts by visibility over the nine-day period between 15 September and 24 September. Of the 5,183 triggers, almost one third are when the visibility estimate from VEIA is 10 SM or greater. These triggers during high visibility conditions may take a lower priority than triggers associated with lower visibility conditions.

This page intentionally left blank.

4. OPERATIONAL TRANSITION AND DEMONSTRATION

In September 2019, MIT Lincoln Laboratory began coordinating with the FAA sponsor as well as WeatherCams to implement a version of the VEIA algorithm on the WeatherCams system. At that time, it was determined that the best approach would be to grant MIT LL access to the WeatherCams system for software development and evaluation. The main obstacle in this approach was allowing MIT LL access to the WeatherCams real-time system, which is hosted on the Google Cloud. MIT LL documented the approach and obtained the required Data Security Plan (DSP), allowing approval for MIT Lincoln Laboratory to access other organizations' systems. In the DSP, it was critical that MIT LL only have access to the VEIA components and not the operational components of the WeatherCams system. WeatherCams was able to meet this requirement and began preparing their system in January 2020.

Another aspect of the implementation was determining the software platform for the algorithm. The VEIA concepts and development occurred in MATLAB, an off-the-shelf development platform that allows for flexibility in the testing of analytical concepts. Unfortunately, MATLAB is an expensive tool and would require purchase, installation, and maintenance in support of VEIA on WeatherCams. To reduce cost, and for ease of installation, an open-source alternative was identified that is compatible with MATLAB called GNU Octave. Octave appeared to be capable of running MATLAB-derived software with minimal changes, but there were some differences between MATLAB and Octave that required algorithm modifications.

In March 2020, all approvals were finalized and the process of releasing the MATLAB-based software on the WeatherCams system using Octave as the platform commenced. The initial stages required setting up a version of VEIA on the WeatherCams Github repository. MIT Lincoln Laboratory obtained an account on Github, and access to the VEIA repository was granted by WeatherCams. The initial software release did not cycle on the WeatherCams system, but through a series of modifications to the software to align with the various function calls specific to Octave, an algorithm was created that cycled on the WeatherCams system.

With the software cycling on the WeatherCams system in April 2020, the focus shifted toward validating the results of the Octave version with the MATLAB version. The initial debugging and evaluation period consisted of a very small subset of the sites and cameras. This allowed for rapid debugging of issues that were specific to the Octave platform. One of the major identified issues was the differences in the output of the edge detection calls between Octave and MATLAB. Analysis was required to understand the difference in the edge detection functions, perform software modifications to account for these differences, and validate that the resulting output was identical. Eventually, all issues were resolved and visibility estimates between the two systems were nearly identical.

In May 2020, with the software cycling and producing acceptable results in the real-time system, an additional 120 cameras were added to the processing list. The goal was for a system that would provide updates at a 10-minute rate. This would match the nominal update rate of the cameras themselves, albeit with a data collection and processing delay. As the number of sites and cameras was expanded, it became obvious that the current system was not going to meet the required update rate. One issue with the design was that it had one instance of the software running for all sites and cameras, and the software was performing the data collection and visibility estimation along with the composite generation. The original software design on the development system envisioned one instance running per camera at the camera site. The design on the WeatherCams system gathered all the cameras into one location, then performed the processing for all the cameras before providing the results. This caused system latency of greater than 10 minutes between the start of data collection and the production of results.

In an attempt to increase the update rate of the current software design, debugging and timing tests were conducted, and the lag in the performance was isolated to two functions in the Octave platform: the sort and 2D cross-correlation functions. Software modifications were made to reduce the calls to the sort functions, which provided a significant increase in performance. However, the latency in the call to the 2D cross-correlation function meant that it could not be used in this version of the software, and, as a result, the call was disabled. The main usage of this call was to align the current image with the composite image and 'adjust' the image in cases where the camera is being moved or shaking due to strong winds, etc. This algorithm module was a necessary component in the development system due to the poor mounting structures used by data providers prior to working with the FAA WeatherCams imagery. After a review of the frequency at which the 2D cross-correlation function adjusted the images from WeatherCams, it was determined that this module could be disabled with very minimal impact.

By May 2020, the software was cycling and providing a 10-minute update on nearly the full set of camera imagery obtained from WeatherCams. The final problem identified in the collection of camera imagery was a set of sites known as 'remote' sites. Within the WeatherCams system, remote sites are defined as sites where there is a limited source of electrical power. For a limited power source, the WeatherCams hardware developers had created a configuration where the cameras would not collect or transfer imagery overnight. This was intended to conserve power during the hours when the imagery would be completely black. As designed, the VEIA software required the overnight imagery in order to trigger the composite generation module, so these remote camera sites could not be supported.

Throughout the summer of 2020, the VEIA software remained frozen and cycling uninterrupted on the WeatherCams system. The daily monitoring of the visibility estimates as compared to a visual review of the imagery itself also showed a consistent level of quality performance as anticipated. The few noted issues were either related to known limitations in the algorithm or extenuating circumstances. Once such example was due to construction on a nearby structure using heavy equipment: the equipment was parking directly in front of the camera, resulting in a blocked image. It was also observed that estimating visibility in the ASOS ranges between 6 SM and 9 SM was very challenging. The strength of the algorithm was estimating in the lower visibility ranges where there are a large set of edges. As a general observation, the algorithm over-reported visibilities in the 6 to 9 SM range when the true visibility was greater than 10 SM, which led to re-evaluation of the translation function, as noted in Section 3.2.3.

The main failure in the system was the limited ability to provide rapid updates (faster than 10 minutes) during daylight hours due to the single process running all components of the algorithm in series. Increasing the number of cameras or sites during the demonstration would have delayed the processing beyond 10 minutes. The other main finding was limitations in the system design when generating composites. The software was designed to create the composite images during the overnight hours when darkness prevents the algorithm from providing useful visibility estimates. In theory, this would be acceptable on a per-camera basis. However, the software was required to create the composites for all the cameras during the night. The data access and composite generation was very time consuming and would often take multiple hours.

During the height of summer in Alaska, the Sun does not set at many of the locations or the night hours are very limited. This inevitably created time periods when the Sun was illuminating the scene, but the VEIA algorithm was still creating composites for all the sites and did not produce visibility estimates during the desired time of day.

In the fall of 2020, the real-time Octave software was re-designed to improve several of the deficiencies that were observed during the initial six months of operations on the WeatherCams system. The primary change was to convert VEIA to a collection of processes, or threads, that would process independently. This required developing a method for these processes to share intermediary results through a file sharing process. The algorithm was broken into three separate components: one for data collection, one for image processing. and one for composite generation. With the re-design, the algorithm was able to meet the expected 10-minute update rate and produce the composites throughout the day without delaying the distribution of results to the WeatherCams displays. The redesign also allowed composites to be generated for sites that do not provide imagery during the night period. The re-designed software was distributed to the WeatherCams system in October 2020 and monitoring of system and algorithm performance commenced. In November, it was determined that the CPU resources were being maxed out throughout the day, which was still limiting the capabilities of the system. WeatherCams was informed and agreed to double the CPU power.

The last major modifications to the real-time Octave software occurred in July 2021. At that time, the trigger functionality to support the crowd-sourcing effort was added to the real-time system and two additional components to support the real-time operations and maintenance of the WeatherCams system were added. The first component is a daily health check of the cameras to provide rapid identification to the system operators of any issues with the cameras, such as blockage from ice or man-made objects (i.e., vehicles, heavy equipment, etc.). The second component identifies the ‘best’ image that may represent a clear day image for the location allowing the system operators to perform less manual review when they wish to update the annotated clear day image accompanying the last image on the website. This was accomplished by adding two additional threads to the VEIA Octave software.

4.1 DAILY CAMERA HEALTH REPORT

During the deployment of the VEIA software on to the WeatherCams system, the personnel responsible for operation and maintenance of the system noted that VEIA was performing several functions that had the potential to assist the management in identifying when cameras are out of service or experiencing some type of malfunction. When comparing the camera imagery and edge strengths to a composite representation of a clear day image, the instances that are considered failure modes of VEIA (i.e., estimating visibilities lower than truth) are often associated with when the camera has been moved or blocked. Example images of three different scenarios that have been observed in the WeatherCams imagery are shown in Figure 20. The first (ice blockage) is very common in the winter time and occurs when the heating elements in the camera housing are either malfunctioning or unable to keep up with the icing conditions. A similar example would be moisture condensation on the lens. The second example shows that the camera has moved due to a mounting failure. A similar example would be the camera being displaced due to high wind conditions. The last example is when human activity blocks or interferes with the camera equipment. This is much less common, but does occur at sites located near human activity.

To assist the WeatherCams system managers, VEIA performs a daily check of all the camera imagery to determine whether any of the cameras are experiencing the failures previously described. This is accomplished by using the composite image as a reference to compare each of the images against. First, in the absence of any reduced visibility conditions, the edge strength ratio used in the estimation of visibility should be close to or greater than one for the entire day. For weather impacted periods, it is expected that visibility will vary throughout the day. Although it is possible that the visibility will remain near zero for an entire day in extreme conditions, it should not persist over multiple days. Therefore, the edge strength ratio should not remain at or near zero for extended periods of time. To examine for these conditions, VEIA identifies the maximum edge strength ratio observed over a calendar day and saves that information. Next, VEIA uses a 2D cross-correlation to measure any offset between the composite image and the image with the highest edge strength ratio for the day. In most instances, it is expected that the cross-correlation will return an offset of zero, indicating that the composite image and the comparison image are perfectly aligned. In these cases, VEIA will flag this camera as having no issues. For instances where the 2D cross-correlation is identifying an offset between the two images, VEIA will threshold the offset and label the camera as having moved if this value is small. Small offsets may occur when the wind is vibrating the camera or the system managers have decided to reposition the camera view. If the offset is outside of acceptable bounds, then the algorithm will flag the camera as needing review by the system managers. VEIA will also provide information on the absence of new imagery, which may be an indication of communication or power failure. Finally, VEIA will flag instances where the composite image is unavailable indicating this camera or site may have been offline for an extended period of time.

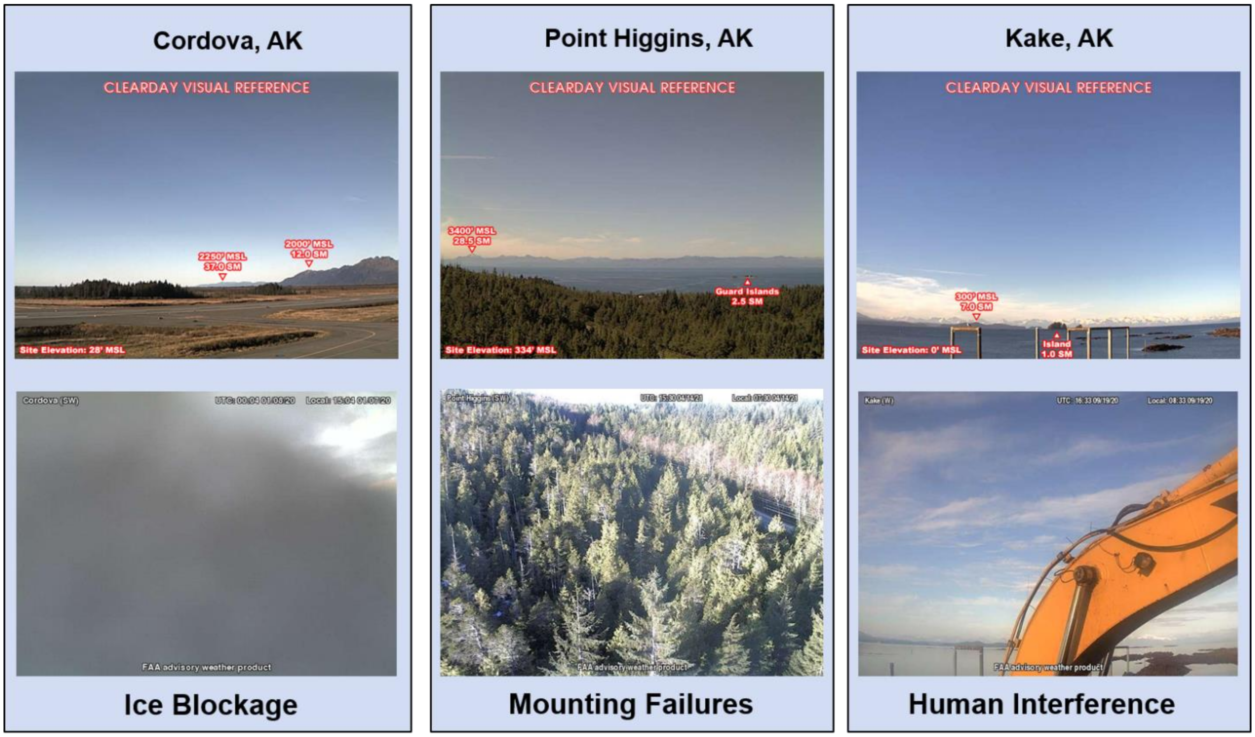
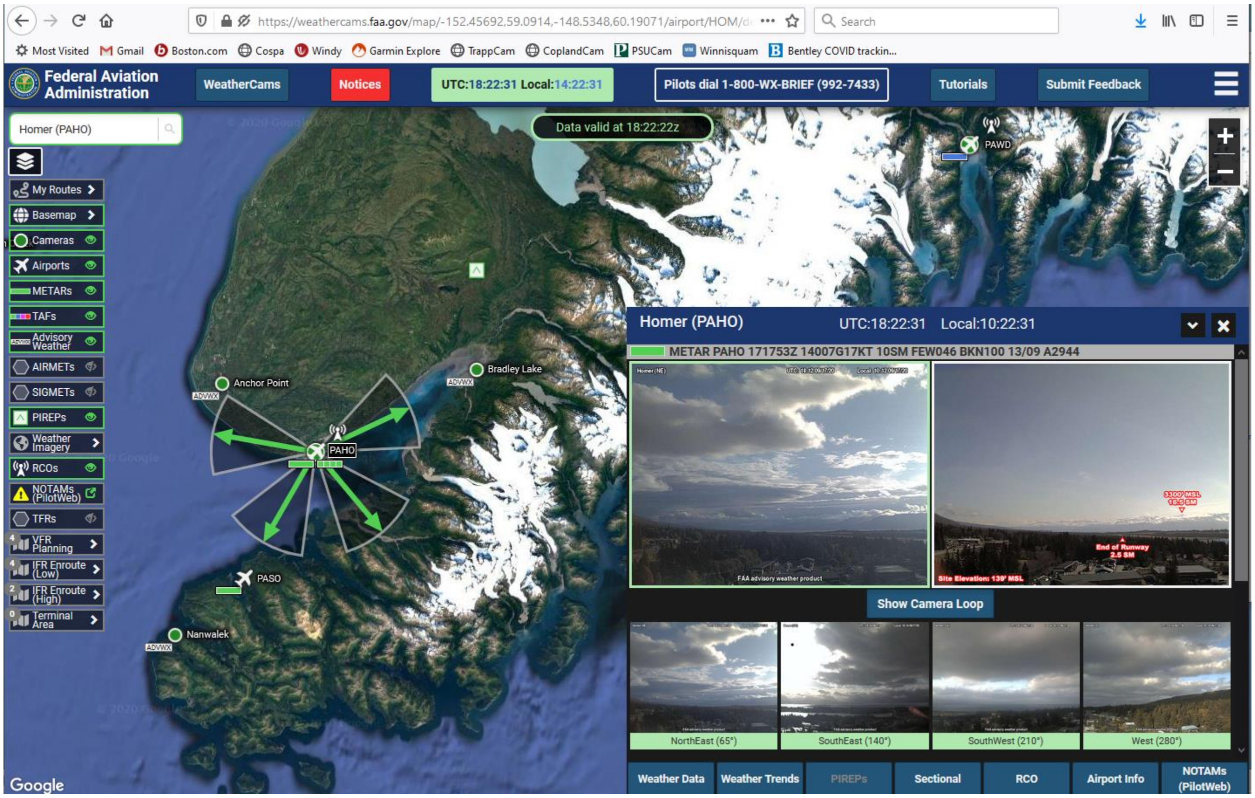


Figure 20. Three examples of issues with camera imagery that the VEIA algorithm can detect and provide an alert to the WeatherCams system managers. The first is very common in the winter time and occurs when the heating elements in the camera housing are either malfunctioning or unable to keep up with the icing conditions. A similar example (not shown) would be moisture on the lens. The second is when the camera has moved due to a mounting failure. A similar example (not shown) would be the camera being displaced due to high wind conditions. The last, human interference, is less common, but a possibility at sites located near human activity.

4.2 CLEAR DAY SELECTION

Another challenge identified by the WeatherCams system managers with the camera maintenance procedures is the labor hours needed to identify and update the clear day images. These images accompany the current image on the WeatherCams website to provide a reference for pilots and other users of the system. The managers noted that when the cameras are repositioned due to requests for different viewing angles, or after routine maintenance, the process of identifying the best clear day image could benefit from the VEIA algorithms. An example of the clear day image accompanying the current image from Homer, AK, on the WeatherCams website is shown in Figure 21. Many of these clear day images also contain annotations to identify the distance to various markers in the image.



METAR PAHO 171753Z 14007G17KT 10SM FEW046 BKN100 13/09 A2944

Figure 21. WeatherCams web display showing the latest camera imagery alongside an annotated clear day example. Users are able to use the clear day image as a reference along with the annotations to estimate the visibility from the current images.

The VEIA software assists the system managers by identifying two images for each camera during the daily camera health check. These two images are the image with the strongest edge strength ratio and the image most closely matching the composite image. These two images are stored in a separate directory to allow the system managers to quickly look at the 'best guess' images from the last few days.

5. SOFTWARE DESCRIPTION

VEIA consists of five processes, or threads, that work in parallel to produce the visibility estimates. Logic in each thread prevents more than one thread from processing a particular camera or site at the same time. Figure 22 depicts these processes as they are called from a shell script within the docker container used by WeatherCams for configuration and deployment on the Google Cloud. The first process (startSystem) sets up the system at startup and then does the daily check of the system health and cleans up old files. The second process (startRetrieve) gathers the camera imagery from the WeatherCams API and sends the new images to the third process (startProcess) which processes all the new imagery. The fourth process (startOutput) gathers all the camera-based visibility estimates and combines them into site-based estimates (combining all cameras at one site) and creates an output file for the graphical user interface (GUI) to read. The final process (startComposite) creates a composite of the last several days of images that is used by the image processing thread as a representation of a clear day.

The processes are started with the following calls from a shell script within a docker file.

- startSystem
- startRetrieve
- startProcess
- startOutput
- startComposite

All processes lead to an infinite loop that performs the desired actions on either a set strobe rate or at a specific time each day. Currently, the goal is to provide an output file for the GUI every 10 minutes. However, due to the required processing power, some of the cycles occur less frequently. All processes also open a log file for that process and produce statements in the log file for monitoring and/or debugging issues. For all processes, a new log file is opened each day to enable smaller log files for review and a bi-monthly deletion of the older log files.

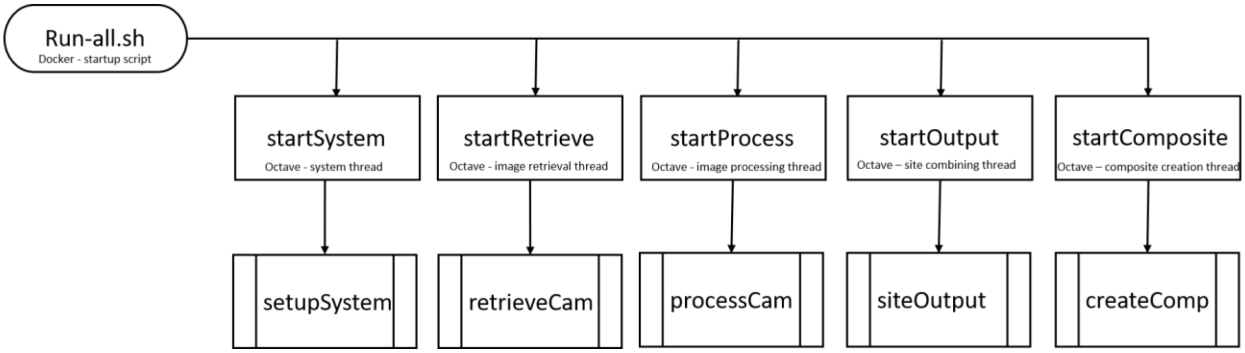


Figure 22. Top level flow chart of the VEIA processes. VEIA consists of five processes that execute simultaneously and are started from a shell script (Run-all.sh) within the docker file. All processes lead to an infinite loop that performs the desired actions on either a set strobe rate or at a specific time of day.

5.1 DESCRIPTION OF PROCESSES

The first process, setupSystem, begins by making sure all the directories that are required for intermediary files to be saved are available, and, if not, it creates the directories. The option is also available to flag a fresh start that cleans out all these directories. Otherwise, the old files are available for the system to continue to use after a system restart. This allows for a seamless restart without interruption and use of the previously generated composite images. Then the process begins an infinite loop. The loop is set to execute processing each day at 12 UTC. The loop often takes about 10–12 hours to complete. The infinite loop executes an inner loop through all the cameras calling the systemStatus function for each camera. Figure 23 depicts the flow chart for the setupSystem process.

The systemStatus function evaluates all the images for a single camera from the previous day relative to the established composite image for that camera. The function will determine the clearest image from the day and compare the images to determine whether the camera has moved or is experiencing some type of blockage. A text string is generated with statistics about the camera and a flag is created to inform the system managers of any potential issues with the imagery. This text string is returned to the setupSystem process. The function will also store the clearest image to the cam_clear directory. Figure 24 depicts the flow chart for the systemStatus function.

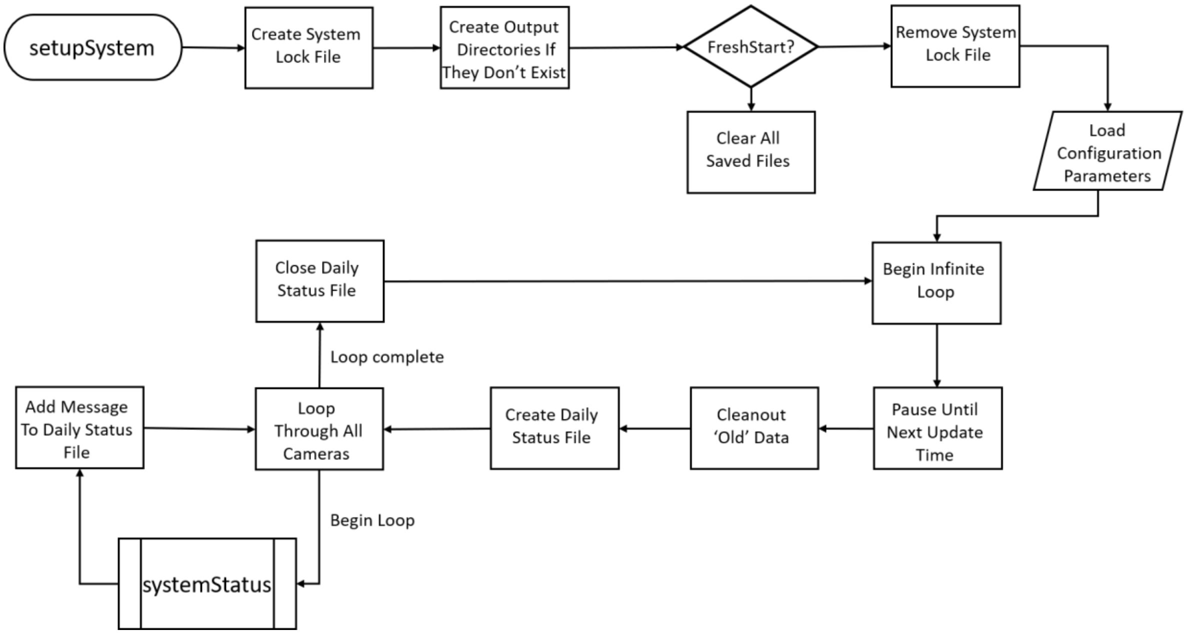


Figure 23. Flow chart for the setupSystem process that begins by setting up the system and then executing an infinite loop that performs a daily check of the camera and provides potential clear day images.

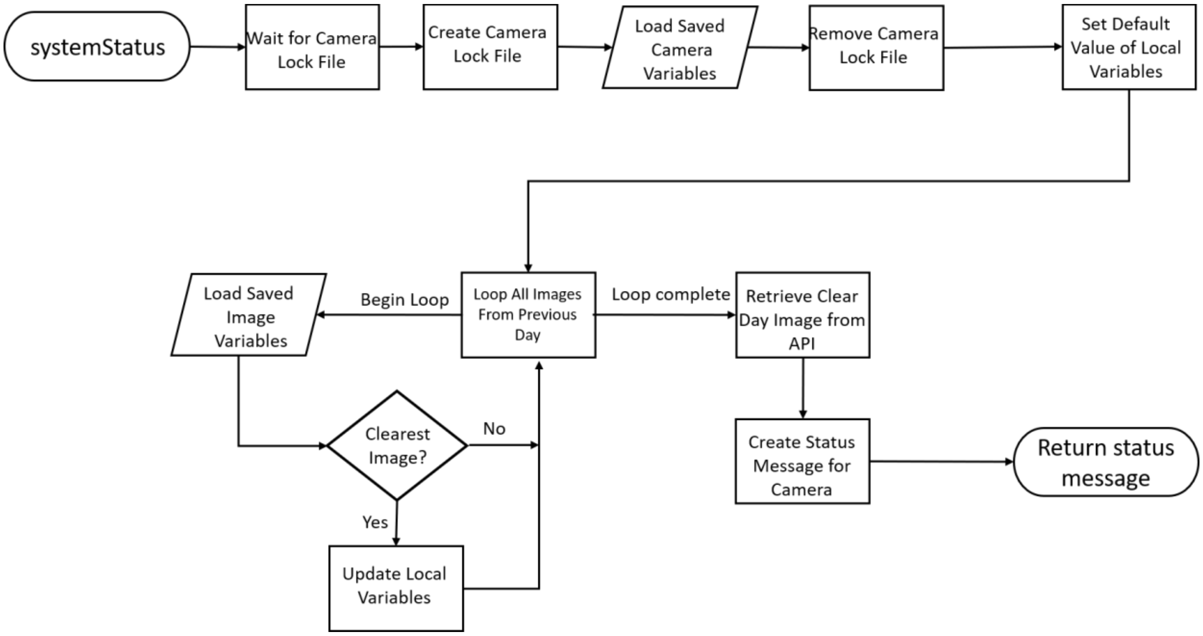

systemStatus
Wait for Camera Lock File
Create Camera Lock File
Load Saved Camera Variables
Remove Camera Lock File
Set Default Value of Local Variables
Loop All Images From Previous Day
Begin Loop
Load Saved Image Variables
Clearest Image?
No
Yes
Update Local Variables
Loop complete
Retrieve Clear Day Image from API
Create Status Message for Camera
Return status message

Figure 24. Flow chart for the systemStatus function called by the setupSystem process. This function performs the daily check on the status of an individual camera and provides both a status message for the daily log and determines the clearest images from the day.

The second process, retrieveCam, gathers the camera imagery from the WeatherCams API. The process cycles once per minute and saves the index to all the available cameras into a binary file that is shared with the image processing thread. The process first asks the API for the newest camera image and compares that file with the previous file that was gathered from the API. If it is the same file, the process continues to the next camera. If the files are different, then the process goes to WeatherCams and gathers the image, stores it locally, and adds it to the list of new files in the newFileStatus.mat file. The process uses several intermediary files in the data directory. These files are stored in the cam_url, cam_images, and cam_file directories where the uniform resource locator (URL) of the image is stored in a text file, and then the local name of the file is saved in the cam_file directory. The images themselves are stored in the cam_images directory. A lock file is used to prevent other processes from working on the camera if the camera image is being retrieved. The process also checks for the image time to be properly read. There are also several components of the process to make sure the web retrieval (wget) process is working properly by checking for error messages, allowing wget to fail and catching the failed process, as well as allowing a read timeout and a specified number of attempts be made before giving up on the camera. Figure 25 depicts the flow chart for the retrieveCam process.

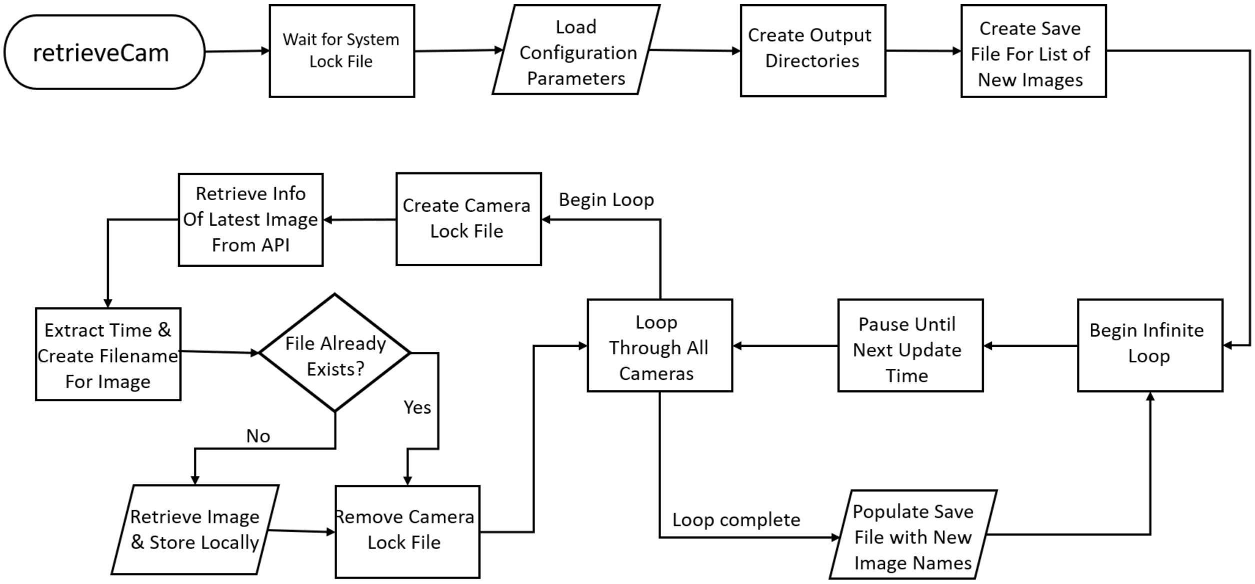

retrieveCam
Wait for System Lock File
Load Configuration Parameters
Create Output Directories
Create Save File For List of New Images
Retrieve Info Of Latest Image From API
Create Camera Lock File
Begin Loop
Extract Time & Create Filename For Image
File Already Exists?
Loop Through All Cameras
Pause Until Next Update Time
Begin Infinite Loop
Yes
No
Retrieve Image & Store Locally
Remove Camera Lock File
Loop complete
Populate Save File with New Image Names

Figure 25. Flow chart for the retrieveCam process that collects the imagery from the WeatherCams API. The process begins reading in the camera list and setting up the appropriate files and directories. The process then executes an infinite loop that loops through the complete list of cameras, checking for a new image, storing the image, and informing downstream processes of the available imagery.

The third process, processCam, is where the bulk of the VEIA processing occurs. processCam begins with an initial setup of log files and reading of the configuration files. It then begins an infinite loop on a 60-minute strobe. The process gathers the names of all new camera files from the retrieve process and calls the computeCam function for each new camera image. A lock file is used to prevent other processes from accessing or saving new data related to a camera during the camera processing. Figure 26 depicts the flow chart for processCam.

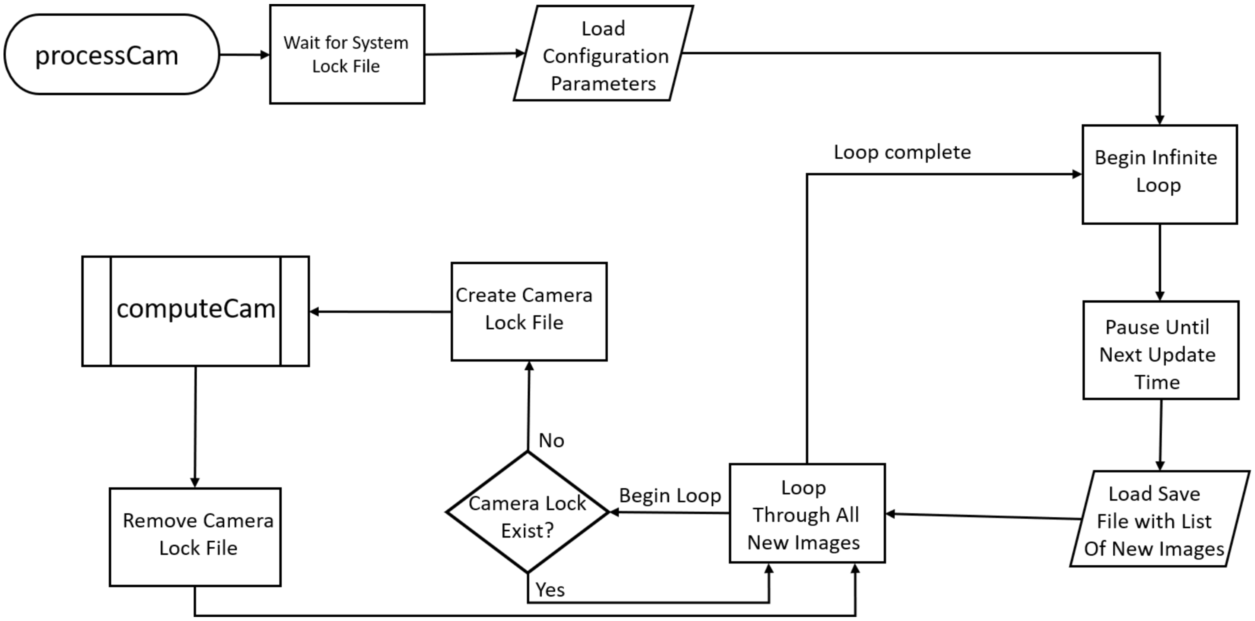

processCam
Wait for System Lock File
Load Configuration Parameters
Begin Infinite Loop
Loop complete
Pause Until Next Update Time
Load Save File with List Of New Images
Loop Through All New Images
Begin Loop
Camera Lock Exist?
No
Yes
Create Camera Lock File
computeCam
Remove Camera Lock File

Figure 26. Flow chart for processCam where the image processing occurs. The process begins by reading in the camera list. The process then executes an infinite loop that loops through the complete list of new camera images, sending the image to the computeCam function.

The computeCam function performs the visibility estimation for each camera image. This is accomplished using a composite image created in the startComposite process. The computeCam function uses a 'savFile' for each camera to save binary variables in each process loop and share camera variables with the computeComp process. Sun angle processing limits and image filtering parameters are set up inside this function. The computeCam function loads information on the image from the urlName file and the actual image from the cam_images directory and then checks that the information meets specified criteria. Once image processing begins, there are checks to verify that the image has enough sunlight and whether the image should be saved for composite generation. The process also cleans out old images that are no longer needed for composite generation. Once all this data storage is sorted out, the visibility estimation is completed by performing the edge detection and edge strength computation for comparison with the composite image. The final step is storing the output of the image processing in a text file in the cam_results directory. Figure 27 depicts the flow chart for the computeCam function.

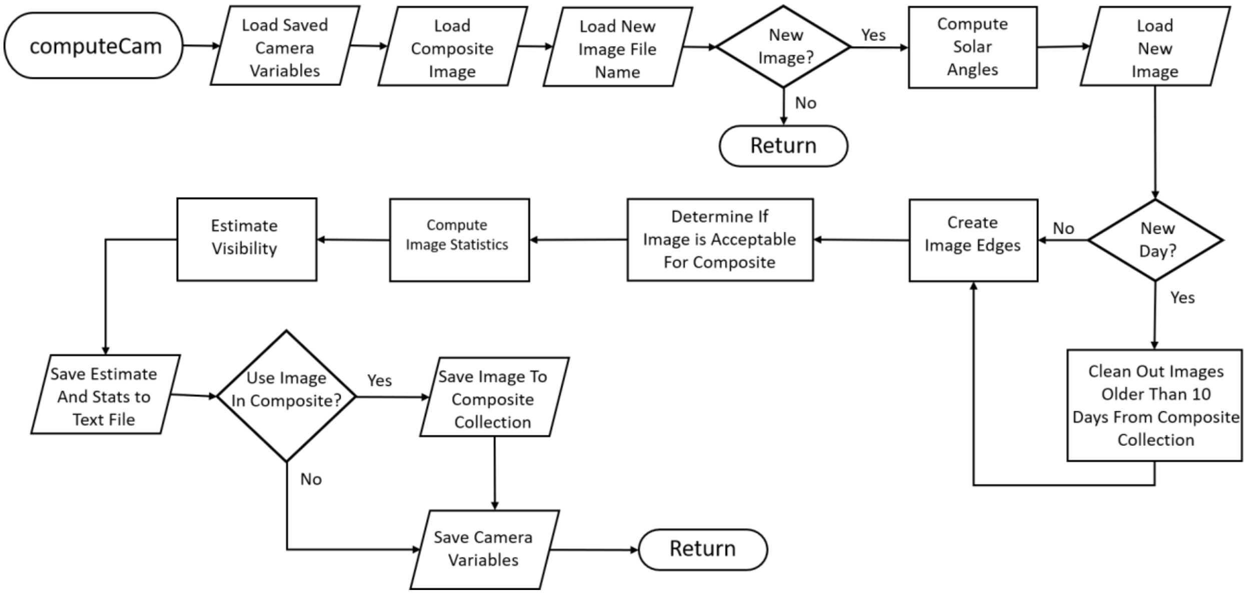


Figure 27. Flow chart for the computeCam function, where the bulk of the image processing occurs. The function begins by loading in the saved camera variables and the composite image. It also computes information on the solar angle and performs the edge detection. The function also maintains the history of images for the composite generation and performs the crucial function of estimating the visibility from the imagery.

The fourth process, siteOutput, creates the prevailing visibility for a site by combining multiple cameras into one estimate and generating an output file for the GUI. The output file contains all the sites on a 10-minute strobe. The siteOutput function begins an infinite loop after reading in the configuration parameters, making log files, etc. Then, during each loop iteration, it loads the site save file and then gets all the per-camera visibility estimates. Next, it applies the weighting scheme to each of the camera visibility estimates that is a function of the Sun angle along with the observed edge strength ratios. As each of the sites are processed, the results are added to the output file that will be shared with the GUI and any other downstream processes. Figure 28 depicts the flow chart for siteOutput process.

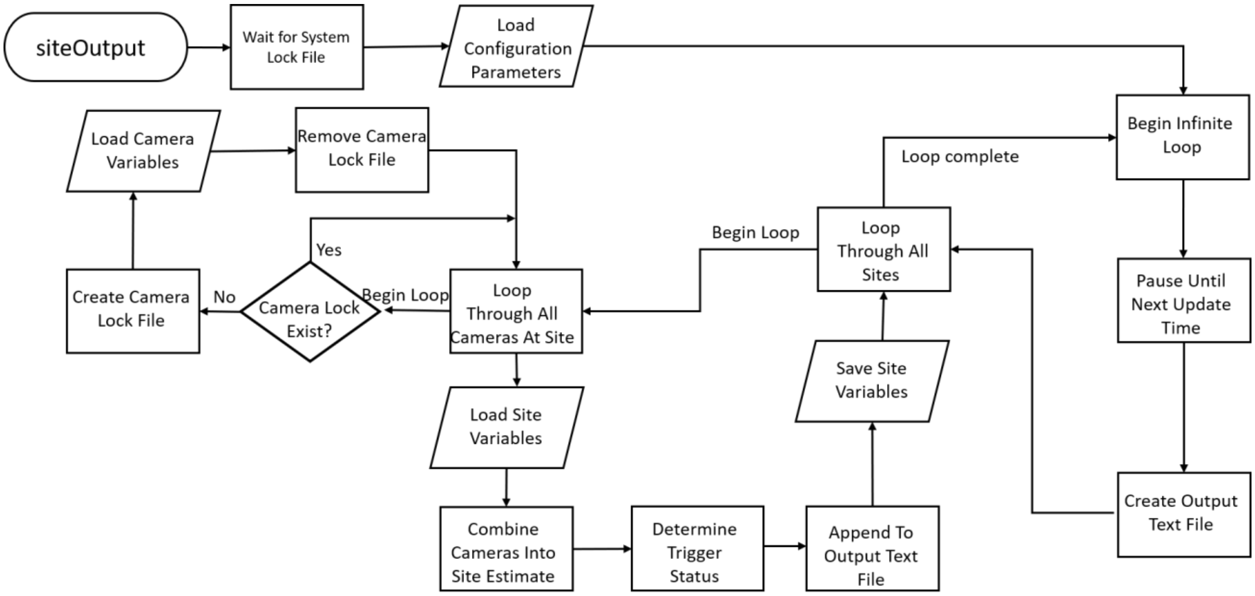


Figure 28. Flow chart for the siteOutput process, where processing for a site's imagery occurs. The process begins by reading in the site and camera lists. The process then executes an infinite loop that loops through the complete list of sites, then loops through each of the cameras for the site and combines all the camera imagery results into one site estimate. The process also creates the trigger and maintains the site variables between each process loop.

The final process, createComp, creates composites for each of the cameras that reflect the closest resemblance to a clear day image without the transitory objects that can pollute an image like cars, people, planes, clouds, etc. createComp begins with an initial setup of log files and reading of the configuration files. Then it begins an infinite loop that is triggered each day at 12 UTC. The process then calls the computeComp function for each camera. A lock file is used to prevent other processes from accessing or saving new data related to this camera during the camera processing. Figure 29 depicts the flow chart for processCam.

The computeComp function performs the composite generation for each camera. This is accomplished by loading a historical collection of images over the previous several days and merging the images into one final composite. The function validates the availability of historical images and saves the variables associated with the composite image for the image processing functions. Figure 30 depicts the flow chart for the computeComp function.

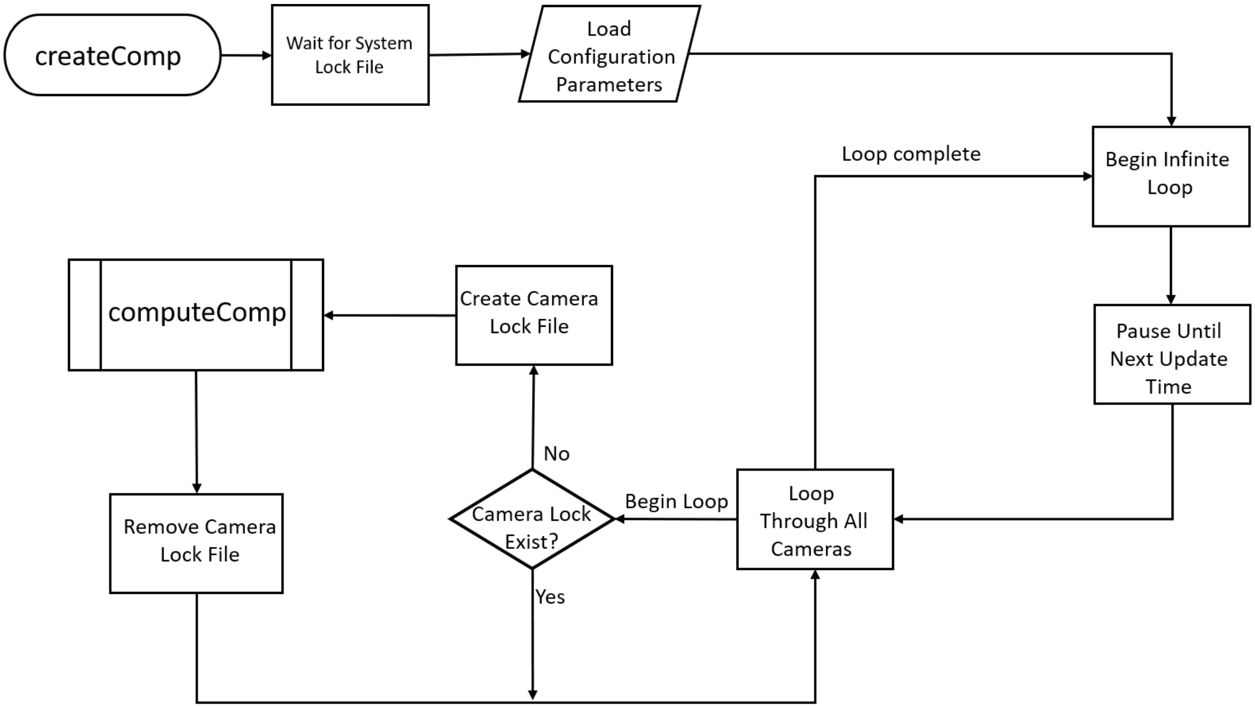


Figure 29. Flow chart for the createComp process, where the composite generation occurs. The process begins by reading in the site and camera lists. The process then executes an infinite loop that loops through the complete list of cameras and executes the computeComp function.

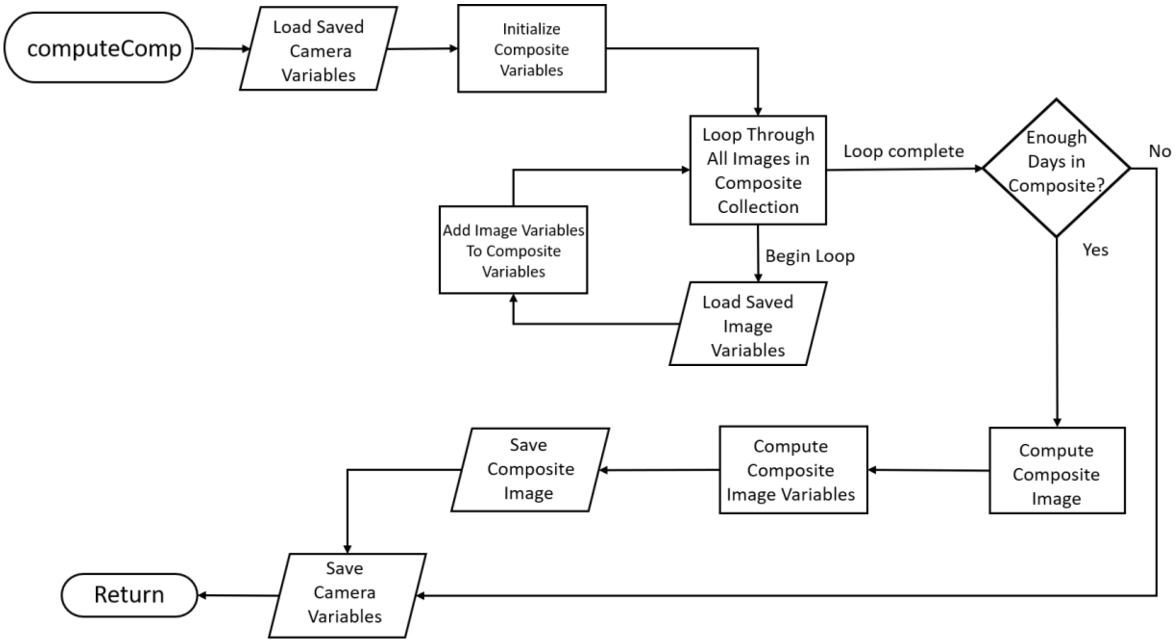

computeComp
Load Saved Camera Variables
Initialize Composite Variables
Loop Through All Images in Composite Collection
Loop complete
Enough Days in Composite?
No
Yes
Add Image Variables To Composite Variables
Begin Loop
Load Saved Image Variables
Save Composite Image
Compute Composite Image Variables
Compute Composite Image
Return
Save Camera Variables

Figure 30. Flow chart for the computeComp function called by the createComp process. This function performs the composite generation of an individual camera by loading a historical collection of images over the previous several days and merging the images into one final composite. The function checks to be sure enough historical images are available and saves the variables associated with the composite image for the image processing functions.

5.2 CONFIGURATION FILES

VEIA uses a set of configuration files for information on the location of the cameras, the direction the cameras are pointed, the maximum viewable distance in the image, and information about whether VEIA should process the camera image and with what method. These files exist in the config directory and are as follows.

AKCV_Cameras.txt contains information on each of the sites and cameras.

SiteID ID ICAO Name Lat Lon Elev CamID Direction Bearing

SiteID	Numeric site id from WEATHERCAMS id list
ID	Three letter site id from WEATHERCAMS id list
Name	Name of the city, town, island, etc., where camera is installed
Lat	Latitude of camera
Lon	Longitude of camera
Elev	MSL elevation of the camera in feet
CamID	Numeric camera id from WEATHERCAMS id list
Direction	Direction camera is pointed (i.e., northeast)
Bearing	Numeric direction camera is pointed (true north)

AKCV_GetList.txt contains information on which cameras to process.

SiteID CamID EdgeMethod MaxCamRange CameraFlag

SiteID	Numeric site id from WEATHERCAMS id list
CamID	Numeric camera id from WEATHERCAMS id list
EdgeMethod	Number of edges to use in each column beyond the horizon
MaxCamRange	Maximum visibility annotation in imagery. Clear day images are available on WEATHERCAMS and this value will represent the max annotation. MaxCamRange is currently not used in the algorithm. NOTE: The site estimate is limited to 10S M.
CameraFlag	Flag indicating whether to process camera imagery or not (Negative values indicate camera imagery should not be processed, positive values indicate camera imagery should be processed)

5.3 OUTPUT FILES

VEIA produces a number of output files for the end users. These files are either ingested by the user interface or monitored by the system managers and any downstream process with a need for the VEIA end products. The output file naming convention and formatting are as follows.

live/output/cam_vis_estimate_YYYYMMDDTHHmmss is the VEIA results file created every 10 minutes with the visibility estimate for all available cameras.

Where:

YYYYMMDD	Year, month, and day
HHmmss	Hour, minute, and second

Header string: YYYYMMDDTHHmmss Cnt

YYYMMDD	Month, day, and year
HHmmss	Hour, minute, and second
Cnt	Number of lines to follow, one per site

Data string: SiteID,Vis,Conf,Num,Trigger

SiteID	Numeric site id from WEATHERCAMS id list
Vis	Visibility estimate in SM in METAR bins
Conf	Confidence value between 0 and 100
Num	Number of new cameras in this estimate
Trigger	Flag providing a trigger to crowd sourcing (0 or 1)

live/cam_clear/SID_CID_YYYYMMDD_MaxRatio.jpg is a JPEG image representing the highest edge strength ratio for this day.

Where:

SID	Numeric site id from WEATHERCAMS id list
CID	Numeric camera id from WEATHERCAMS id list
YYYYMMDD	Year, month, and day

live/cam_clear/SID_CID_YYYYMMDD_NormRatio.jpg is a JPEG image with the edge strength ratio closest to the composite image for this day.

Where:

SID	Numeric site id from WEATHERCAMS id list
CID	Numeric camera id from WEATHERCAMS id list
YYYYMMDD	Year, month, and day

live/status/systemStatus_YYYYMMDD is a daily status file with information on the health of each camera.

Where:

YYYYMMDD Year, month and day

Data string: SiteID CamID Cnt LC MC HC MR MRDiff MROffset Status

SiteID	Numeric site id from WEATHERCAMS id list
CamID	Numeric cam id from WEATHERCAMS id list
Cnt	Number of images evaluated for the day
LC	Number of images with a ratio <= 0.4
MC	Number of images with a ratio > 0.4 & <= 1.0
HC	Number of images with a ratio > 1.0

MR	Maximum observed ratio for this day	
MRDiff	Grayscale difference between MR image and composite	
MROffset	Pixel offset between MR image and composite	
Status	Status of image	
		NO_IMAGES
		FAILED_TO_REVIEW
		CAMERA_UNAVAIL
		CAMERA_OK
		CAMERA_MOVE
		CAMERA_REVIEW

5.4 INTERMEDIARY FILES

VEIA relies on a number of intermediary files to share data between the five different processes that perform the algorithm functions. These files are saved in local directories that are named based on the type of files contained within them. Some of the files are empty and serve as lock files, some contain a single line with the name of a file or a URL, some contain binary data that are formatted specifically for the Octave application, and some are formatted text containing human readable text strings. In each directory, there is typically one file per camera containing the images or variables or one file per site.

live/cam_lock/camera_lock_CID is an empty file present when processing is occurring.

Where:

CID	Numeric camera id from WEATHERCAMS id list

live/cam_url/camera_url_CID contains a text string of the URL location for an image.

Where:

CID	Numeric camera id from WEATHERCAMS id list

live/cam_file/camera_file_CID contains a filename where the JPEG image is stored.

Where:

CID	Numeric camera id from WEATHERCAMS id list

live/cam_var/camera_var_CID is a binary Octave file containing camera variables.

Where:

CID	Numeric camera id from WEATHERCAMS id list

live/site_var/site_var_CID is a binary Octave file containing site variables.

Where:

SID	Numeric site id from WEATHERCAMS id list

live/cam_images/YYYYMMDDHHmm.SID.CID.jpg is the camera image file.

Where:

YYYYMMDD	Year, month and day
SID	Numeric site id from WEATHERCAMS id list
CID	Numeric camera id from WEATHERCAMS id list

live/cam_results/cam_results_CID is a text string of the camera-based visibility estimate.

Where:

CID	Numeric camera id from WEATHERCAMS id list

Data string: Date SiteID CamID MV Vis SZ SA ES NES Ratio IDiff CE UrlName

SiteID	Numeric site id from WEATHERCAMS id list
CamID	Numeric camera id from WEATHERCAMS id list
MV	Maximum allowed visibility (SM) output for camera
Vis	Estimated visibility in statute miles
SZ	Solar angle of Sun in zenith (0 above, 180 below)
SA	Solar angle of Sun relative to lens (0 in front, 180 behind)
ES	Mean edge strength of selected edges
NES	Mean edge strength of selected edges from composite
Ratio	Edge strength ratio (translated to visibility in algorithm)
IDiff	Grayscale difference of image and composite
CE	Flag indicating if a composite is available
UrlName	Text string of URL from WeatherCams

6. FUTURE RESEARCH ACTIVITIES

The FAA Weather Cameras platform is an excellent resource for the aviation community to view real-time weather conditions and continue the development of the VEIA concepts as well as future potential weather products. The current concept of providing automated observations from the camera image has focused solely on visibility and using a single translation function that does not account for variations in the scenes that may be captured by the imagery. This section will briefly describe potential areas for enhancing VEIA through algorithm modifications to improve performance and expand the weather features that are important to the aviation community.

6.1 SCENE CLUSTERING AND UPDATED TRANSLATION FUNCTION

Throughout the development of VEIA, several performance issues have been observed that are associated with the specific characteristics of the scene that is captured by the camera. One such characteristic, previously discussed in Section 3.2.2, was the presence of large antenna or tower structure that is close to the camera. This issue was largely mitigated through the development of an edge selection process that reduces the number of edges selected that are associated with the tower or structure. Other scene characteristics of concern are the lack of edges due to open water sources over a large range of the visibility spectrum and scenes with a very limited view range.

To illustrate how siting conditions can affect VEIA performance, new camera installations in Colorado were studied and compared to the performance of VEIA across Alaska. Beginning in 2020, a set of cameras was installed in the mountains of Colorado by the FAA WeatherCams office. A total of 12 sites were available in Colorado beginning in mid-July 2020, and MIT Lincoln Laboratory began archiving data from these sites to allow a review of any potential issues that may be identified in this environment. By the end of August 2021, enough data had been collected, including a sufficient number of low visibility events, to facilitate further analysis.

Overall, the performance of VEIA across the Colorado sites was very similar to the Alaska network. Figure 31 plots the median VEIA visibility for each Meteorological Terminal Aviation Routine Weather Report (METAR) visibility bin for all the Colorado sites. There are 18 visibility bins in METAR, beginning at 0 SM and ranging up to 10+ SM. In the figure, the median values are plotted with circles and a median of all sites is also calculated and shown with a black diamond. The median of all sites compares very well with the METAR visibility as shown in the black line. However, there is significant spread across all the sites. This suggested that although the median across a large set of sites produces a very good comparison, individual sites may require a closer look to determine a cause as well as a potential solution to these differences.

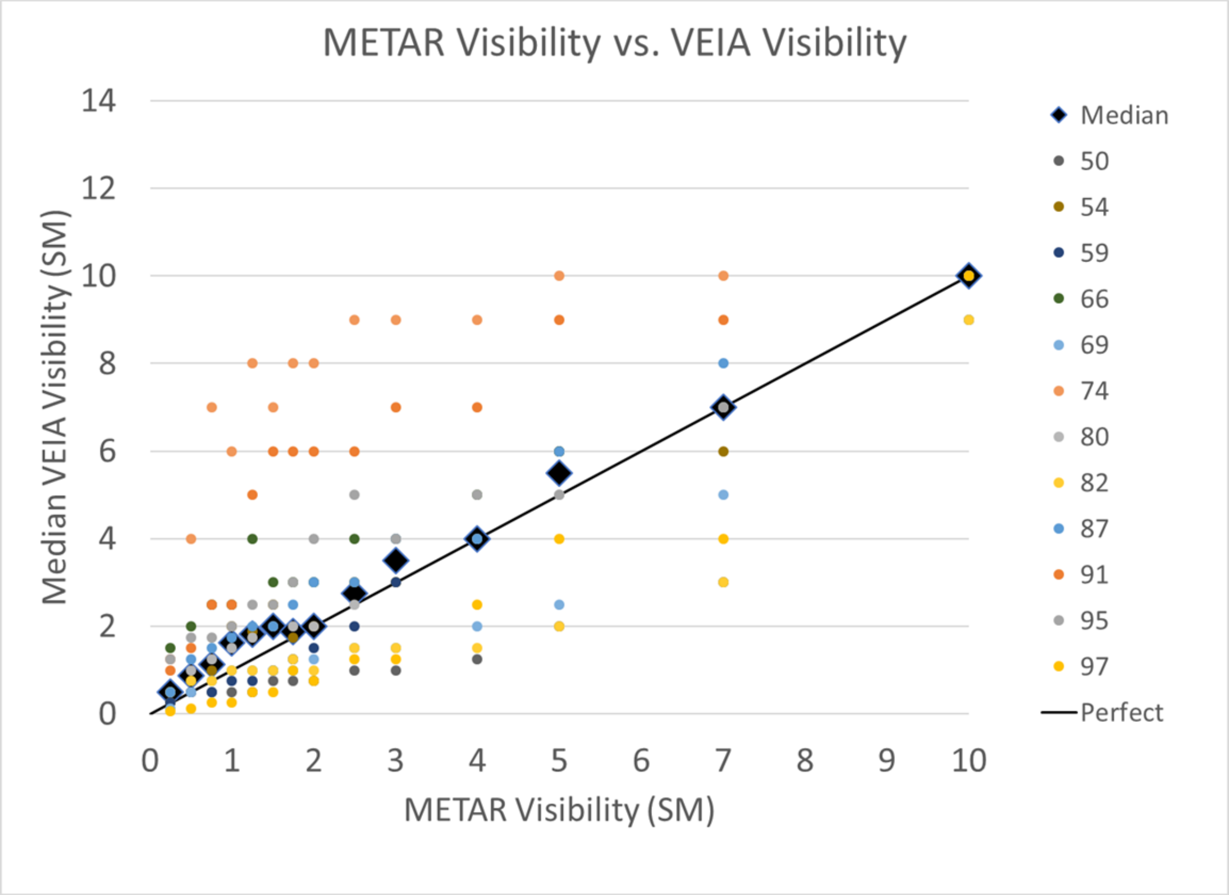


Figure 31. Comparison of the METAR visibility measurements and the VEIA visibility estimates for all Colorado sites. The median of all the VEIA visibility estimates at each METAR visibility bin is shown for each site (colored dots) and an overall median (black triangle) is computed using all the sites. The median of all sites compares very well with the METAR truth. However, individual sites show significant variability.

A similar analysis can be conducted at the camera level. This analysis has been the historical method used both to score and calibrate the translation function used within the VEIA algorithm. Figure 32 is a plot of all the cameras (colored dots) in the Colorado domain as compared to the co-located METAR sensors. At the camera level, the median computed across all cameras, as shown in the black diamond, compares very well with the METAR truth at visibility values less than 3 SM. However, in the range from 3 SM to 7 SM, the camera-based median is showing an underestimation of the visibility. Although this may look concerning, it may also be suggesting that the weighting function applied in VEIA that is dependent upon the Sun angle and agreement between cameras is doing a very good job of generating the VEIA site estimates.

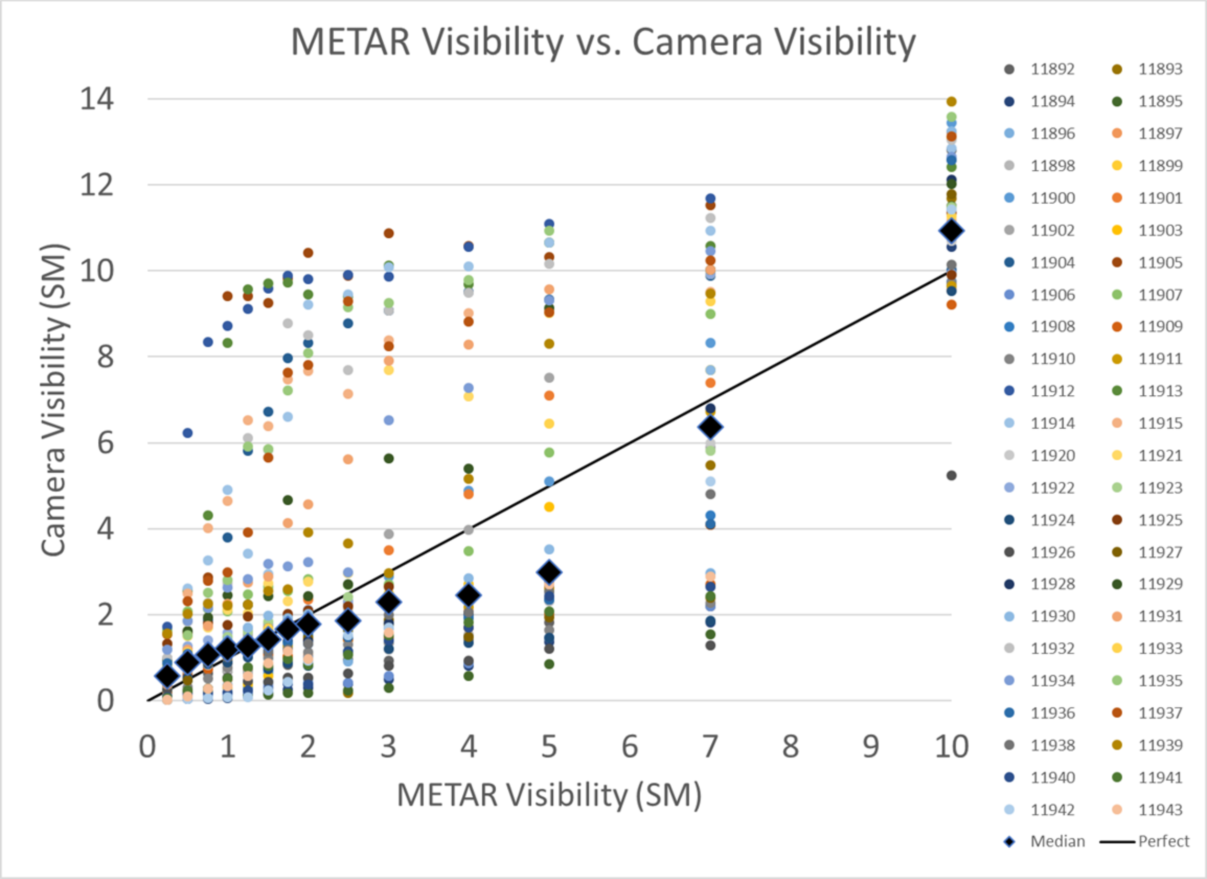


Figure 32. Comparison of the METAR visibility measurements and the camera-based visibility estimates for all Colorado cameras. The median of all the camera-based visibility estimates at each METAR visibility bin is shown for each camera (colored dots identified by cameraId) and an overall median (black triangle) is computed using all the cameras. The median of all cameras compares very well with the METAR reading when the METAR is below 3 SM, whereas the estimate is low in the range of 3 SM to 7 SM. The results show an excellent comparison at 10 SM and above. However, individual cameras show significant variability.

Further analysis on the large variability across cameras leads to a comparison of the site estimate with the individual camera estimates for the sites and/or cameras with the highest error. In Figures 33 through 35, the median VEIA site estimate is shown in the black diamond and compared to the four median camera estimates for three sites with the highest error. Each of the four cameras is plotted with a colored circle and the view angle of the camera is noted in the legend. A perfect match with the METAR is shown in the black line. Accompanying the plot are the ‘clear day’ images for each of the four cameras.

Figure 33 shows a site with relatively good performance, at least in the range between 0 SM and 2 SM. At ranges greater than 2 SM, the visibility is underestimated, but there is a general agreement between the cameras. A review of the clear day images for Kremmling, CO, shows four images with similar camera views. All four cameras have clear views of the horizon. However, for this site, the horizon is generally at a distance of 1 SM. Also, the foreground of these images is lacking in available markers, which may have an impact on the VEIA performance. It should also be noted that this camera site is close to the surface, which may also impact the viewing angle of the images when looking across landscapes. The results of these images and the observed performance and associated errors at ranges greater than 1 SM may suggest this site could benefit from a modified translation function.

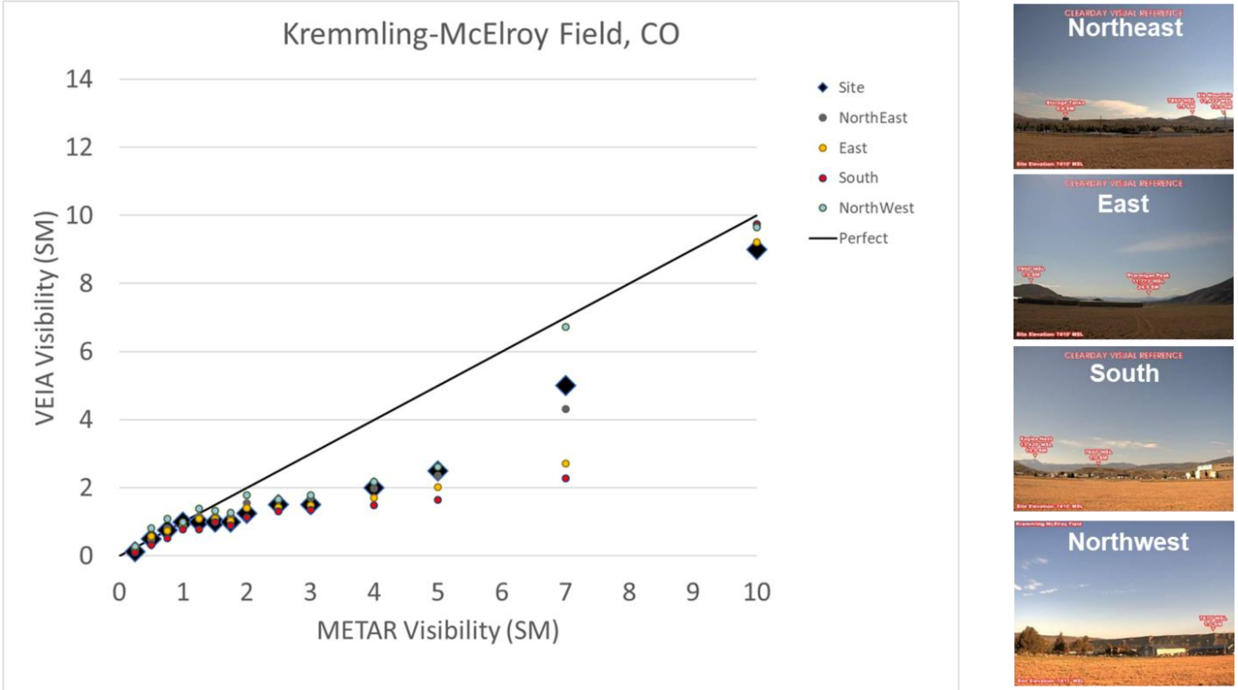


Figure 33. Comparison of the METAR visibility measurements and the VEIA visibility estimates from Kremmling, CO, for both the site estimate and each of the individual camera estimates. The median of the site and all cameras compares very well below 2 SM and is underestimated between 2 SM and 10 SM. Also shown is the 'clear day' reference image for each of the cameras.

In contrast to the performance at Kremmling, the performance of VEIA in Figure 34 is notably much poorer. For this site, Walton Peak, there is a significant overestimation of the visibility from VEIA. Two of the cameras, the west-looking camera and the northeast-looking camera, are contributing to this overestimation in all visibility ranges, whereas the south- and southeast-looking cameras are contributing to this in visibility ranges greater than 3 SM. A review of the clear day images for the west and northwest cameras clearly shows that the horizon is blocked by a nearby tree line where the tops of the trees extend above the distant horizon in the image. In these instances, the camera-based estimation algorithm is selecting the tops of the trees as the horizon and, as long as the trees are detectable, is assuming a high visibility. Of course, these trees are visible in the imagery through all visibility ranges down to less than 0.25 SM.

Finally, in Figure 35, for La Veta Pass, the problem of all four cameras overestimating the visibility due to nearby tree line and other objects is obvious. At this site, the north camera has been turned off, as it was identified as a camera image with high tension wires and nearby towers that would have negative effects on VEIA. Despite that, the other three cameras are also impacted by the nearby tree line. In the clear day images, the trees have lost their foliage due to a forest fire; however, the impact is still present as the trees have recovered.

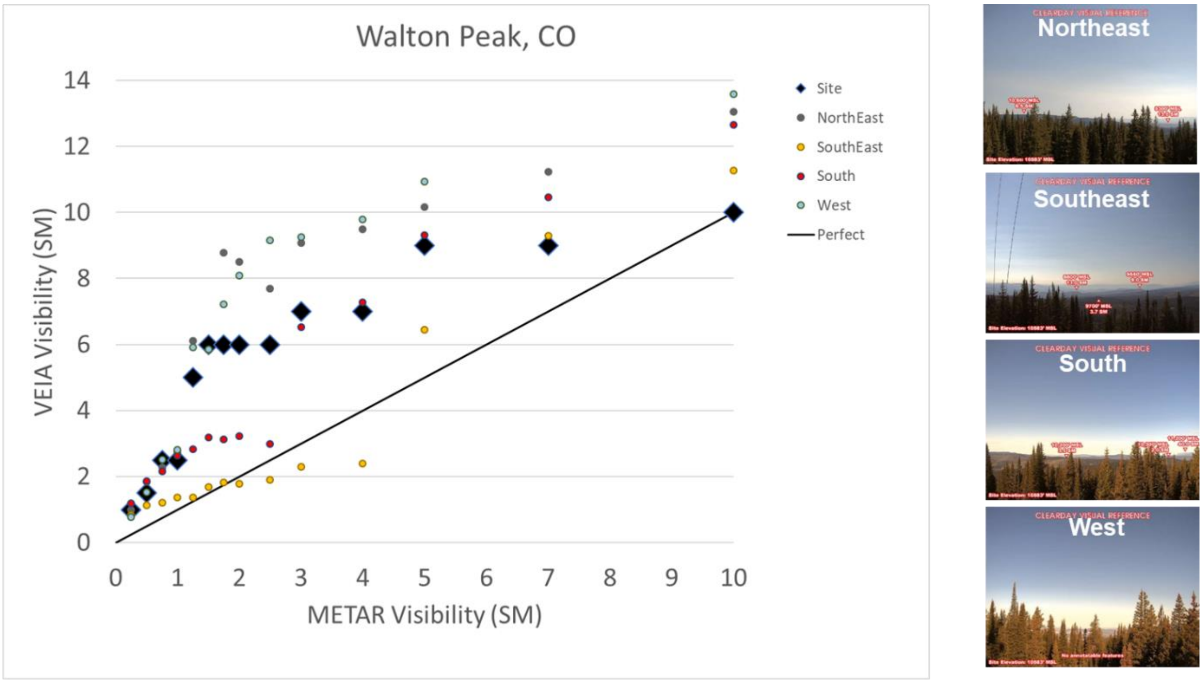


Figure 34. Comparison of the METAR visibility measurements and the VEIA visibility estimates from Walton Peak, CO, for both the site estimate and each of the individual camera estimates. The median of the site and all cameras is overestimated for all visibility ranges. Also shown is the 'clear day' reference image for each of the cameras.

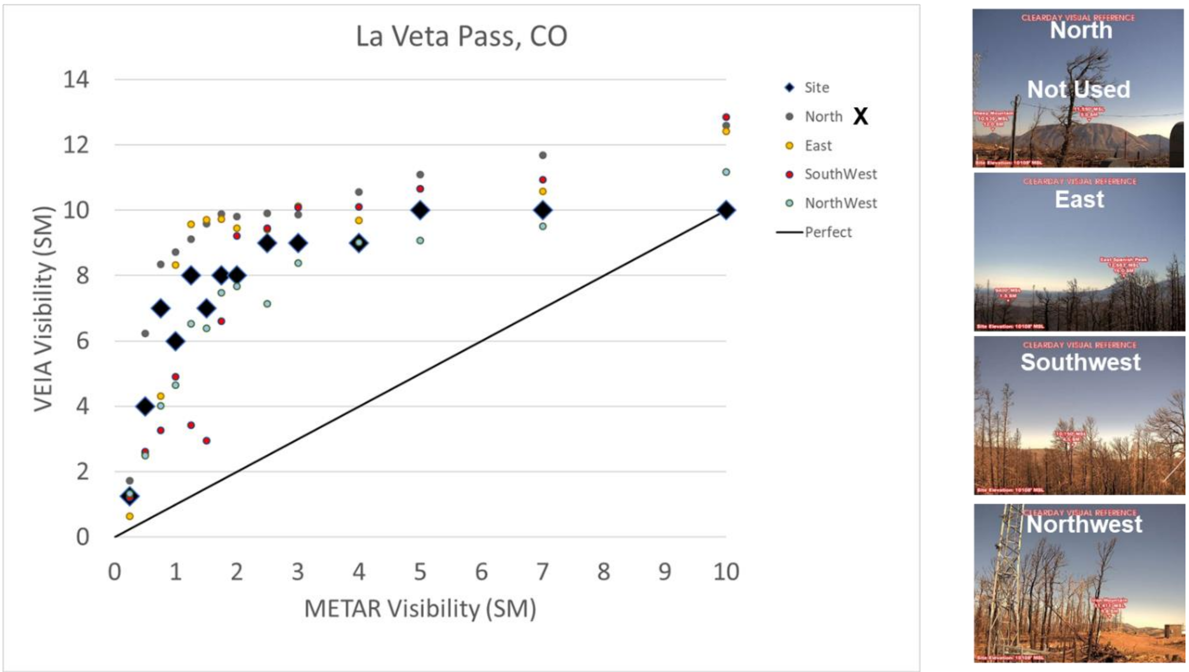


Figure 35. Comparison of the METAR visibility measurements and the VEIA visibility estimates from La Veta Pass, CO, for both the site estimate and each of the individual camera estimates. The median of the site and the cameras is overestimated for all visibility ranges. Also shown is the 'clear day' reference image for each of the cameras.

To improve the performance of VEIA, future research should focus on clustering scenes with similar characteristics together and then applying a modified translation function to the cluster. For instance, the cameras in Walton Peak and La Veta Pass that have a close proximity tree line could be clustered together and then a translation function that recognized the rapid increase in visibility as the edge strength ratio increases would be applied. Another approach could be to develop a different edge selection process for a cluster without a distance horizon due to a close proximity tree line. In this instance, perhaps only using the horizon (eliminating the next strongest edge) would provide the best selection of edges for a modified translation function. If this concept proved successful, then additional research could focus on developing a machine learning algorithm to perform the clustering. This would be critical as the network of cameras expands and it becomes too time consuming to manually associate a new image with a pre-defined cluster.

6.2 CLOUD ESTIMATION THROUGH IMAGE ANALYTICS

The image processing techniques applied to date have focused on the estimation of visibility with an emphasis on accurately detecting low visibility events to allow awareness of trends towards lower visibility during the critical transition time from VMC to IMC. The camera imagery also contains valuable information on the cloud characteristics that could be extracted. Important cloud features are the cloud coverage, cloud height, and even the cloud type. The image processing in VEIA lends itself well to performing cloud estimation in conjunction with visibility estimation. Additional research could lead to a cloud estimation through image analytics (CEIA) algorithm.

One of the techniques applied within VEIA is a horizon detection process used in defining markers in the image. Using this technique, a future CEIA algorithm could easily identify the sky and begin performing feature extraction of cloud features. Edge extraction could potentially determine the texture in the sky. Another potential feature that could be extracted from the sky is its natural hues. Hues that are in a range that is common to blue sky could be used to measure the overall ‘blueness’ of the sky potentially being indicative of a clear sky. Various machine-learning techniques could also be investigated using a high-quality truth set and the horizon detection to provide a training and test set for the development of a cloud cover estimation.

6.3 SELECTIVE WEIGHTING FOR OBSTRUCTED IMAGES

One final, potentially near-term research effort could be selectively changing the weighting function between the four images based on the determination of image obstruction. The functionality built into VEIA to monitor the cameras for the system managers is identifying cameras that are potentially blocked with ice or moisture on the lens as well as human interference. Modifications might be possible to automatically eliminate cameras from the weighting process during these times of questionable camera performance.

This page intentionally left blank.

7. SUMMARY

This document provides a detailed documentation and history of development for the VEIA algorithm developed by MIT Lincoln Laboratory, funded by the FAA AWRP, and implemented on the FAA Aviation WeatherCams system.

VEIA provides an inexpensive and robust way to extract meteorological visibility from cameras—thus transforming weather cameras into weather sensors. With the proliferation of web-based camera imagery for monitoring conditions near airports and other remote locations, there is an opportunity to significantly expand the density of visibility observations, especially in areas where low visibility can have dire consequences.

The VEIA algorithm uses the presence and strength of edges in an image to provide an estimation of the meteorological visibility within the scene. The algorithm compares the overall edge strength of the current image to a clear day representation to generate an edge strength ratio. The ratio is then converted to visibility in statute miles using a translation function derived from a historical relationship between edge strength and visibility. VEIA uses a multiple-day composite of clear day images for the representation of a clear day to ensure that only permanent edges are considered (i.e., the horizon, roadways, buildings, etc.).

The algorithm also combines the estimates from multiple camera images into one estimate for a site or location using a weighted average. The weighting applied for each camera is based on the agreement between camera estimates and the position of the Sun relative to each camera's view. The final output for a site is a prevailing visibility estimate in statute miles that can be easily compared to existing ASOS and/or human-observed visibility.

MIT Lincoln Laboratory began working with the FAA in 2016 and implemented VEIA on the FAA Weather Cameras cloud during the spring of 2020. To implement the developmental software on the WeatherCams system, modifications were required in order to operate on the WeatherCams real-time platform. This included converting VEIA to the appropriate software environment that is supported by WeatherCams. GNU Octave was chosen as an environment that is compatible with MATLAB, the original development environment.

This page intentionally left blank.

GLOSSARY

AAWU	Alaskan Aviation Weather Unit
API	Application Program Interface
ASOS	Automated Surface Observation System
C&V	Ceiling and Visibility
CEIA	Cloud Estimation through Image Analytics
CWSU	Center Weather Service Unit
DSP	Data Security Plan
FAA	Federal Aviation Administration
GA	General Aviation
GSL	Global Systems Laboratory
GUI	Graphical User Interface
IMC	Instrument Meteorological Conditions
LLSC	Lincoln Laboratory Supercomputing Center
METAR	Meteorological Terminal Aviation Routine Weather Report
MIT LL	Massachusetts Institute of Technology/Lincoln Laboratory
NCAR	National Center of Atmospheric Research
NOAA	National Oceanic and Atmospheric Administration
ROC	Receiver Operating Characteristic
URL	Uniform Resource Locator
VEIA	Visibility Estimation through Image Analytics
VMC	Visual Meteorological Conditions
WeatherCams	FAA Weather Cameras Program Office
WFO	Weather Forecast Offices
WTIC	Weather Technology In the Cockpit

This page intentionally left blank.

REFERENCES

[1.] Hallowell, R. G. (2005) *Automated Extraction of Weather Variables from Camera Imagery*, 2005 Mid-Continent Transportation Research Symposium, Ames, IA.

[2.] Hallowell, R. G. (2007) *An Automated Visibility Detection Algorithm Utilizing Camera Imagery*, 23rd Conference on Interactive Information and Processing Systems for Meteorology, Oceanography, and Hydrology (IIPS), San Antonio, TX, *Amer. Meteor. Soc.*

[3.] Fenton, K. R. (2021). *Visibility Estimation through Image Analytics (VEIA) Assessment.* Global Systems Laboratory, Forecast Impact and Quality Assessment Services Branch. Boulder, Colorado: National Oceanic and Atmospheric Administration.

[4.] Miller, J. (2021). *Visibility Estimation through Image Analytics (VEIA) User Assessment Final Report.* Office of NextGen, Aviation Weather Division, William J. Hughes Technical Center, Aviation Weather Demonstration and Evaluation (AWDE) Services. Atlantic City, NJ: Federal Aviation Administration.

This page intentionally left blank.