Correlation and Local Feature Based Cloud Motion Estimation

Hao Huang
Stony Brook University
100 Nicolls Road
Stony Brook, NY 11794
haohuang@cs.stonybrook.edu

Shinjae Yoo
Brookhaven National Lab.
50 Bell Avenue
Upton, NY 11973
sjyoo@bnl.gov

Dantong Yu
Brookhaven National Lab.
50 Bell Avenue
Upton, NY 11973
dtyu@bnl.gov

Dong Huang
Brookhaven National Lab.
50 Bell Avenue
Upton, NY 11973
dhuang@bnl.gov

Hong Qin
Stony Brook University
100 Nicolls Road
Stony Brook, NY 11794
qin@cs.stonybrook.edu

ABSTRACT

Short-term changes in atmospheric transmissivity caused by clouds can engender more severe fluctuations in photovoltaic (PV) outputs than those from traditional power plants. As PV energy continues to penetrate the U.S. National Energy Grid, such volatility increasingly lowers its reliability, efficiency, and value-added contribution. Therefore a model that can accurately predict the cloud motion and its affect on PV system's production is in a pressing demands. It can be used to mitigate the undesired behavior beforehand. In this paper we explore the use of Total Sky Images and the cloud estimation techniques based on such images. To further improve estimation quality of motion vector, we propose a novel hybrid algorithm taking the advantages of both correlation based and local feature based approaches. Our proposed hybrid approach significantly reduces the cloud motion prediction error rate by 25% on average, which can help to predict short term solar energy frustration in our later work.

1. INTRODUCTION

The rising cost of producing fossil fuel, its severe impact to environment, and contributions to green-house gas prompt continuous efforts globally to invest alternative solutions to the world energy demands. Among them, nuclear energy was a promising source of energy until a recent nuclear accident from Japan raises a concern and suspicion regarding its viability and sustainability. Other alternative or renewable energy solutions have attracted more attentions recently, including photovoltaic also known as solar energy, wind, tides, and so on. Solar energy is one of the most promising energy solution due to these facts: 1) It can deployed in residential and industry areas and virtually has no restriction on installation locations, 2) it is silent, 3) it requires less maintenance, and has a longer lifetime because of no moving parts, and 4) it is economical because of the decreasing solar panel price.

However, variability is the biggest challenge to integrate solar energy into the National Grid. Its intermittency, if left untreated, can seriously interfere with other power sources sharing the same grid [1] and normal services to grid users. Grid operators need to take actions (activating reserve power or scheduling power from remote sources) prior to solar power change, which requires accurate prediction of solar power output. Although Department of Energy (DOE) investigated SEGIS (Solar Energy Grid Integration Systems) [2], an accurate, robust estimation of renewable energy production is still unavailable.Cloud motion and subsequent change in shading on panels have been the primary reason contributing to the volatility in solar power production. Therefore, accurate power prediction essentially involves cloud motion vector detection and subsequent cloud location prediction. To our best knowledge, this type of variability prediction in a short term prediction (from 30 seconds to 10 minutes) has not been studied thoroughly. The prediction information within this time range is critical in smart grid integration because it allows the grid operator to anticipate and compensate for fluctuations in PV output either manually or automatically via smart grid software control.

Wind Forecasting Improvement Project has already showed preliminary benefits of about $1 million in electricity production cost savings for one month in power market simulation studies, which analyzed the implications of improved accuracy in wind forecasting in the Texas (ERCOT) power market. These savings were realized for an improvement of about 2% in the Mean Absolute Error (MAE) for the 6 hour-ahead wind forecast. Using the result from this cloud estimation project we are targeting to predict short term solar energy output with similar preliminary benefits.

We address this variability problem of solar energy by detecting cloud motion vectors using Total Sky Imager (TSI) which takes hemispheric images of the whole sky. The motion vector can be subsequently used to estimate sun radiation level and actual panel power output. The motion vector detection from a series of consecutive images has been widely adopted from particle image velocimetry [3, 4], to

MDMKDD'12, August 12, 2012, Beijing, China.

(a) TSI Imager (b) Original (c) Undistorted

Figure 1: Image Undistortion

cloud coverage prediction from satellite cloud images [5].We present two most popular motion vector detection approaches (cross-correlation and phase correlation) and then customize these algorithms for TSI image processing. There are some limitations and challenges of these two approaches when they are adapted to TSI image processing. In this paper, we will address these challenges including 1) the distortion of the spherical cloud images are not evenly uniform between the center and borders of the image, 2) the shape of cloud changes during its motion, and 3) the resolution of TSI images is low around boundaries. To further improve estimation quality of motion vector, we propose a novel hybrid approach combining the advantages of both conventional area-based approaches and local feature based approaches. Our proposed hybrid approach significantly reduces the cloud coverage prediction error rate by 25% on average.

2. BACKGROUND AND RELATED WORK

The total sky imager (TSI) (Figure 1), which is manufactured by Yankee Environmental Systems (YES) Inc., provides time series of hemispheric full-color sky images. The sky images allow us to accurately compute and predict fractional cloud coverage and solar radiation levels.

When we process TSI images, we need to consider the following effects: First, due to hemispheric reflective lens, the whole sky image can be taken but the sky image is distorted, which requires image recovery from the distortion (Figure 1). Second, the recovered image resolution is not uniform. The resolution of the edge of image is much lower than that of the center. Third, sunlight shadow band and a camera-holding arm block sky image, which may affect cloud motion detection. Fourth, the boundary of image is not rectangle, which does not align well with the standard block-wise computation. To detect cloud motion, we also need to pay special attention to the following cloud image properties: 1) The cloud image may contain arrays of similar partial clouds such as Alto Cumulus. 2) The shape of cloud may continuously change over time. 3) The cloud boundary is not clear frequently. 4) The cloud near Sun tends to show high level of brightness. Many techniques have been developed to detect object motion vectors but due to the unique properties of cloud image properties, a few of them are actually appropriate for cloud motion detection. Arrays of similar partial cloud can generate multiple narrow extrema in the correlation surface, which could mislead SSDA (sequential similarity detection algorithm [14]), for instance. Since cloud does not have a clear boundary, active contour models such as Snakes [15] are difficult to be used. Wavelets and other multi-resolution schemes [16], to some extent, are not suitable to TSI image because of the low resolution especially near the boundary, where ideal features are usually unavailable.

One most recent work which carries similar purpose with our research is [21]. The major drawback of this work is that it only used cross correlation method. Cross correlation method is one of the most popular techniques for rigid movement detection, and it even works on cloud motion detection to certain content. However it may fail on cloud motion detection due to the following reasons 1) it is unable to detect accurate vector when the cloud shape changes, 2) It is quite sensitive to the intensity changes or multi-layer clouds, and 3) it suffers from low resolution around boundary area [18]. Similar problems happen in phase correlation as well.

Many research have been focus on motion capturing based on high-resolution features [24]. These include image registration techniques, which have been used widely in multimodal medical or photo registration problems [23] [25], and some hierarchical methods [22] that align images based on minimizing differences in intensity. Neither includes an applicable way for sky images. Moreover, significant challenges must be overcome in adapting these measures to the shape-changing moving object such as cloud. For example, popular image registration techniques are susceptible to local minima and tend to produce incorrect alignment on the object with obscure shape [23]. Direct methods cannot handle substantial scale and orientation differences between images.

In this paper we introduces a novel hybrid framework to approach the solutions specifically for the above problems, whose idea is in high-level similar to [22] with hierarchical model, but totally different on the chosen techniques and their combination. Our goal is to capture the cloud motion which is non-rigid and with obscure boundary usually.

3. CORRELATION-BASED CLOUD MOTION ESTIMATION

We investigate two most popular motion vector detection approaches: Phase Correlation (PC) and Cross Correlation (CC). Then we discuss how to further improve these approaches. With the detected motion vectors, we explain how to predict upcoming TSI images.

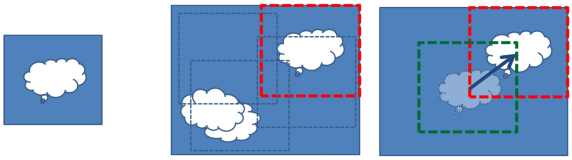
3.1 Phase Correlation (PC)

The first candidate is phase correlation because of its popularity and the estimation speed [3]. Basically, it checks the correlation between two images with the same size and location and compute motion vectors. We divide f and g into overlapped $N \times N$ sub-frame respectively where f is prior frame to g. The following procedure is phase correlation calculation:

1. Compute discrete $2D$ Fourier transform FFT, $F = FFT(f)$ and $G = FFT(g)$.
2. Calculate the cross-power spectrum matrix $R = G \times F^*$ where F^* is the complex conjugate of F.
3. Compute the phase correlation matrix $C = FFT^{-1}(R)$ by performing the inverse FFT of R.
4. Locate the highest correlation point of C, which is our desired motion vector from the center of subframe (Figure 2).

(a) Subframe f (b) Subframe g (c) Motion vector location on highest phase correlation score

Figure 2: Motion vector detection using phase correlation



(a) Subframe f (b) Subframe g (c) Motion vector location on highest cross correlation score

Figure 3: Motion vector detection using cross correlation

However, phase correlation is particularly weak to the noise due to the Fourier transformation. To avoid this, we need to consider bigger block size, N. The second limitation is that it can capture up to half of block size, $N/2$, but this half size is not realistic because the signal-to-noise ratio of correlation will be decreased with big motion vectors [3]. It is also worthy to mention that the Phase correlation assumes no cloud shape changes, which could be unrealistic.

3.2 Cross Correlation

The other popular but more robust approach for motion vector detection is directly computing Pearson Correlation Coefficient between two subframes. Given a prior $N \times N$ sub-frame f, we look for the most correlated current sub-frame g of the same block size from candidate search space $M \times M$ where $M > N$. Specifically we apply the following procedures:

1. Calculate the mean $\overline{f}$ and the standard deviation σ_f of each block f.

2. For each search space (i, j) of $M \times M$, calculate the cross correlation $C(i, j)$, which is

$$C(i,j) = \frac{1}{N^2} \sum \frac{(f - \overline{f})(g - \overline{g})}{\sigma_f \sigma_g} \quad (1)$$

3. Locate the highest correlation point of C, which is our desired motion vector (Figure 3).

Although cross correlation is simple and robust to noise compared to phase correlation, it requires searching motion vectors on $M \times M$ search space, which make it slower. It cannot handle cloud shape changes either.

3.3 Improvements on PC/CC for TSI Images

We apply three additional approaches to improve the above two candidate approaches. The first one is using two different block sizes, bigger block size for the center of images and smaller block size for boundary areas. Although the bigger block size ensures robust motion vector detection, it could not find enough motion vectors and suffers from the boundary effect in that the correlation matrix usually finds its peak along the edge of the circle or the shadowband. With the proposed two block sizes, we could 1) find more motion vectors near edges and 2) relieve the boundary effects.

The second approach to avoid the boundary effect is to fill local mean values into the boundary area, shadowband and camera-holding arm area. It significantly removes spurious motion vector near the holding arm and shadow band.

Third, we also used the algorithm proposed in [4] to remove the spurious motion vector (the vector that is significantly different from the vectors around it). This method was good for cloud shape change but it can not remove multiple spurious motion vectors. Therefore, we have to use all three approaches together.

3.4 Cloud Motion Estimation Using PC/CC

Our core idea of estimating new image frame is that we use the motion vector detected from previous two (I^{t-1} and I^t) frames and project the frame I^t into the new frame I^{t+1}. Our basic assumption is that the cloud shape will not change too much and the cloud maintains the same velocity in a short time scale [13].

As we use image sub-blocks to detect motion vectors, the cloud motion estimation is simplified to block-wise movement where each block starting at (i, j) on time t with block size $m \times m$ ($I^t_{(i,j)(i+m,j+m)}$) has one motion vector $\vec{v}^t_{(i,j)} = \langle u, v \rangle$ where I^t is the image frame at time t. Given frame I^{t-1}, I^t and the frame I^{t+1} which we are going to estimate, we have the following equation:

$$I^t_{(i,j)(i+m,j+m)} \approx I^{t-1}_{(i-u,j-v)(i-u+m,j-v+m)} \quad (2)$$

which means that the texture in $I^t_{(i,j)(i+m,j+m)}$ is approximately the same as the texture in $I^{t-1}_{(i-u,j-v)(i-u+m,j-v+m)}$. Based on the assumption of constant cloud speed in a short time [13], we can estimate the texture in frame I^{t+1}:

$$I^{t+1}_{(i+u,j+v)(i+u+m,j+v+m)} = I^t_{(i,j)(i+m,j+m)} \quad (3)$$

To avoid the mosaic effect in the estimated new frame brought by block-wise movement, three operations are applied. First, we enlarge the moving block size by adding a border around it. Second, for those block-areas in second frame which have the same target area (or overlapping partly) on the new frame, we calculate the mean value of them as the final texture for the overlapped target area. It helps us to smooth the boundary effect of block and get better estimated image. Third, if the moving block contains image boundary (e.g. shadowband, arm), we just move the other area inside the block except boundary area. Finally for those empty area in the estimated third frame, we fill it with the texture in the same area in second frame.

4. LOCAL FEATURE(LF)-BASED CLOUD MOTION ESTIMATION

Cross-correlation and phase correlation mentioned above can be classified as area-based approaches since they compare and match the intensity patterns of block-wise area. However, these methods have three severe drawbacks in motion vector detection on TSI images: 1) Both approaches is unable to detect accurate vector when the cloud shape changes, 2) They are quite sensitive to the intensity changes or multi-layer clouds, and 3) they suffer from low resolution around boundary area [18]. All of these common phenomenon will trigger failure of the current existing approach [21].

On the other hand, local feature can outperform correlation methods on the aspects above because of their robustness about change of cloud shape and illumination. In computer vision, the need for a stable local descriptor that is robust to geometric variations (such as viewpoint, scaling, and affine transformation) and photometric invariance (such as brightness and exposure) has captured the attention of researchers for years. The points with these properties are usually called local features. Through our research in this paper, local features can be used to extract cloud motion vectors based on the point correspondence between two frames with a more stable and robust performance than correlation methods. However, recordings of TSI images show that local features are typically not uniformly distributed, but biased in spatial location and certain sky image texture. Nevertheless, local features can still be a reasonable complementation of correlation methods for cloud motion detection. Then, in this paper, we propose to adopt a local feature based approach, Partially Intensity Invariant Feature Descriptor (PIIFD) [7], to adjust motion vectors detected by correlation methods.

We specifically apply PIIFD on motion vector detection due to its good performance on pattern matching with poor quality textures (especially with the appearance of obscure texture). Generally the process consists of three steps. First, candidate feature points, as sufficient and uniformly distributed as possible, are selected using certain techniques [19][6]. Second, we calculated PIIFD features from the candidate points. Finally, we match PIIFD features between frames, remove the wrong matches and refine those existing matches (image registration).

4.1 Harris Corner Points Detection

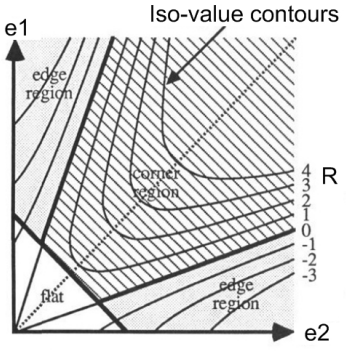


Figure 4: An example of R threshold setting for Harris corner detection [6]. R is a function of first two eigenvalues $e1$ and $e2$ of M. Area with shadow has high enough value of R therefore is corner region.

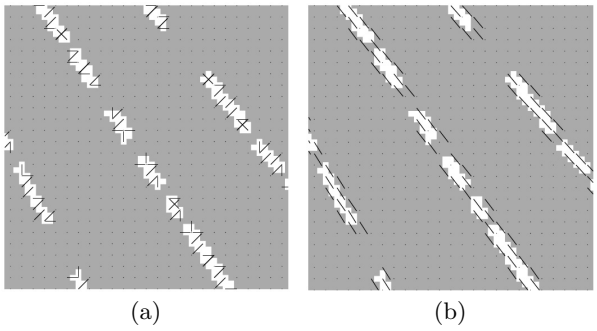


Figure 5: (a) Gradients. (b) Average squared gradients. [12]

To obtain enough corresponding candidate feature pairs among images, we need sufficient and uniform distributed points as feature candidates. For distinctiveness, it is useful to ensure that the neighborhoods of each point have sufficiently rich image texture. However the TSI images boundary area has low resolution and thus it does not provide enough local details there. Many researches on obscure/blurry image registration use corner points to get sufficient quantity of feature candidates, which can be viewed as junctions of contours with large variation in the neighborhood of the point in all direction.

Harris corner point [19] [6] is one of the most popular choices among similar methods. Its basic concept is to measure the changes in all directions by using image gradients. For an image I, the traditional image gradients can be depicted using partial differential along row and column:

$$\begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \partial I/\partial x \\ \partial I/\partial y \end{bmatrix}. \tag{4}$$

With gradient, we can recognize the Harris corner point by looking at intensity value within a small window h. Shifting the window around corner point in any direction should yield a large change in appearance. The value of change, R, is a function of the first two eigenvalues $e1$ and $e2$ (Figure 4) alone. On grounds of rotational invariance. It is attractive to avoid the explicit eigenvalue decomposition of M [6], thus R can be also mathematically expressed as

$$R = det(M) - k \times tr^2(M), \tag{5}$$

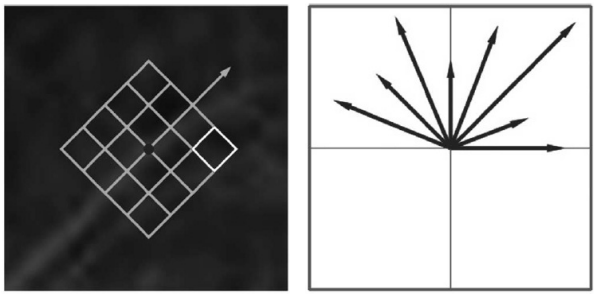
where M is matrix resulted from a convolution as follows:

$$M = \begin{bmatrix} G_x^2 & G_xG_y \\ G_yG_x & G_y^2 \end{bmatrix} * h, \tag{6}$$

In the above formula, h is a Gaussian window with the window size β and k is a relative weight tuning parameter (usually $k \in [0.04, 0.06]$ [19][6]), and det and tr are the determinant and the trace of the matrix M. In one image, only those points with high value of R are assigned as corner points (see Figure 4). Extremal points inside the gaussian window that do not have sufficiently large R are therefore culled. The threshold for R usually depends on the image and the number of corner points needed. In our application, we set it as 0.1 to obtain sufficient local feature candidates.

4.2 Main Orientation Computation

The task of a local feature descriptor is to describe local feature in such a way that it can be distinguished from



(a) Neighborhood of a corner (b) Orientation histogram

Figure 6: Main Orientation Assignment [7]. (a) Neighborhood of each corner point is related to the main orientation. (b) Orientation histogram of the rightmost square in (a)

the other local features and also matched effectively with those that are similar enough. Such distinction is intimately related to the main orientation of each feature. The main orientation is the maximum peak of the gradient histogram in the neighborhood of each feature candidate. To get a rotation invariant feature descriptor, the main orientation according to the local gradient should be assigned to each harris corner point before extracting the feature descriptor. Firstly the image gradient is accumulated across an image window. In order to make those opposite gradient enforcing instead of canceling each other, average square gradient [8] is used before averaging. Moreover, the scale-space theory tells that averaging with a gaussian window minimizes the amount of artifacts that are introduced by subsampling [9], which leads to the reduction of the number of false local feature points.

The average squared gradient $[\overline{G_{s,x}}, \overline{G_{s,y}}]^T$(see Figure 5(b)) around the neighborhood of a candidate point specified by a Gaussian window W (matrix-wise) can be calculated by:

$$\begin{aligned}\begin{bmatrix}\overline{G_{s,x}}\\ \overline{G_{s,y}}\end{bmatrix} &= \begin{bmatrix}\Sigma_W G_{s,x}\\ \Sigma_W G_{s,y}\end{bmatrix}\\ &= \begin{bmatrix}\Sigma_W G_x^2 - G_y^2\\ \Sigma_W 2G_x G_y\end{bmatrix} = \begin{bmatrix}G_{xx} - G_{yy}\\ 2G_{xy}\end{bmatrix},\end{aligned} \tag{7}$$

where

$$\begin{aligned}G_{xx} &= \Sigma_W G_x^2,\\ G_{yy} &= \Sigma_W G_y^2,\\ G_{xy} &= \Sigma_W G_x G_y.\end{aligned} \tag{8}$$

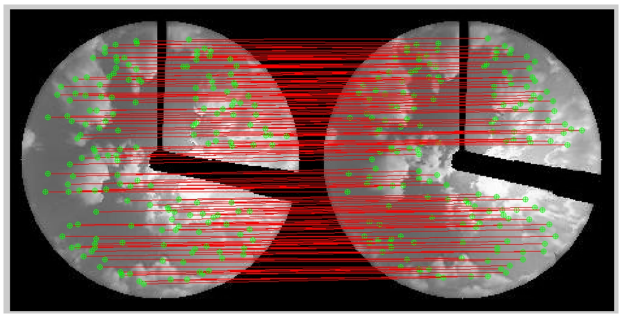
G_{xx}, G_{yy} and G_{xy} are estimates for the variance and cross covariance of G_x and G_y.

For each candidate neighborhood, its main orientation ϕ is given by

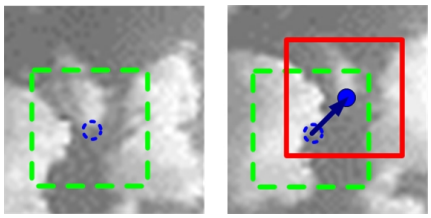
$$\phi = \frac{1}{2}\begin{cases} tan^{-1}(\overline{G_{s,y}}/\overline{G_{s,x}}) + \pi, & \overline{G_{s,x}} \geq 0\\ tan^{-1}(\overline{G_{s,y}}/\overline{G_{s,x}}) + 2\pi, & \overline{G_{s,x}} < 0 \cap \overline{G_{s,y}} \geq 0\\ tan^{-1}(\overline{G_{s,y}}/\overline{G_{s,x}}), & \overline{G_{s,x}} < 0 \cap \overline{G_{s,y}} < 0\end{cases} \tag{9}$$

4.3 Feature Descriptor Extraction

Given the main orientation (arrow in Figure 6(a)) of each feature point candidate, the advantages of achieving partial intensity invariance root from the ability to extract image outlines [10], which is a line marking the multiple contours or boundaries of an object. In PIIFD, image outline is represented by the constrained image gradients, whose orientations are rotated relative to the main orientation ϕ. PIIFD achieve partial intensity invariance by normalizing the gradient magnitudes in order to reduce the adversary effect of gradient magnitude changes. By this way, PIIFD will remain stable even with the intensity of the local neighborhood change dramatically.



(a) Local feature pairs from two frames



(b) Motion vector from local feature pair

Figure 7: Motion vector detection using local features

4.4 Feature Matching and Refinement

The purpose of this step is to build accurate correspondence between the two TSI images. Best-bin-first (BBF) [11] is the key part in this step. It can identify the approximate closest neighbors of high-dimensional points in terms of dot product. Different from [7], we keep all the pair correspondence with similarity (probability) larger than 0.8. The reason is that we want to keep as many candidate pairs as possible but TSI images have lower resolution, so that we have fewer correspondence candidates than other application domain. Since BBF is unilateral instead of bilateral, geometric distribution among candidates is also used to remove incorrect matches. Chen at.el [7] computed a ratio of Euclidean distances between two matches and remove those matches out of range (far away from 1). In our application, we allowed a slightly larger range since some distances may be far away from the other due to the appearance of multilayer cloud with the different motion direction and velocity (Figure 7(a)).

4.5 Cloud Motion Estimation Using Local Feature Correspondence

After investigating how to get stable local features on TSI images, it sets a stage for us to introduce the motion estimation using local features. Here we use the similar strategy as cloud estimation on correlation methods: de-

tect the motion vector first and move texture to estimated frame. We get motion vectors by checking the displacement for each corresponding feature pairs (Figure 7(b)). Denote a local feature point as p with coordinates (p_i, p_j). Suppose p^{t-1} and p^t are a corresponding LF pair from frame at time $t-1$ and t respectively, the motion vector $\vec{v}_p^{t-1}$ for p^{t-1} is $\langle p_i^t - p_i^{t-1}, p_j^t - p_j^{t-1} \rangle$. With assumption for constant cloud velocity in short term [13], the motion vector $\vec{v}_p^t$ for p_t from time t to $t+1$ is the same as $\vec{v}_p^{t-1}$. Once we get all the motion vectors from LF pairs, motion estimation is the next step. For each feature, there is a surrounding block. With these feature-centered blocks, the rest part of motion estimation is quite similar to what we described on section 3.4.

Algorithm 1: MoEst-CC-LF(I^{t-1}, I^t, V_{cc}, P^t, V_P^t, α)

Input: I^{t-1} and I^t are the frame at time $t-1$ and t; $\vec{V}_{cc}$ are the block-wise motion vector matrix from cross correlation motion detection, with each element $\vec{v}_{cc}$ inside depict the motion vector on that block from time $t-1$ to t; P^t are the local feature array in frame t, and $\vec{V}_P$ are the corresponding motion vector array come from local feature pairs of I^{t-1} and I^t; α is a area smoothing parameter

Output: Estimated third frame, I^{t+1}

foreach *block*(i, j) *in* I^t **do**
  Add boundary with α width (pixel) around it, the area of the new block $I^t_{(i-\alpha,j-\alpha)(i+m+\alpha,j+m+\alpha)}$ is denoted as B^t ;
  $\vec{v}_{cc}$ is the motion vector of B^t from V_{cc} ;
  Denote our adjusted motion vector as $\vec{v}_{cc-lf}$ for B^t ;
  $\vec{v}_{cc-lf} = \vec{v}_{cc}$;
  $count = 1$;
  foreach *local feature point* p^t *in* B^t **do**
    $\vec{v}_p$ is the motion vector of p^t from $\vec{V}_P$;
    $\vec{v}_{cc-lf} = \vec{v}_{cc-lf} + \vec{v}_p$;
    $count = count + 1$;
  end
  Get the average motion vector $\vec{v}_{cc-lf}$ by $\vec{v}_{cc-lf} = \sum \vec{v}_{cc-lf}/count = \langle u, v \rangle$;
  We get B^{t+1} as $I^{t+1}_{(i-\alpha+u,j-\alpha+v)(i+m+\alpha+u,j+m+\alpha+v)}$;
  Copy texture of B^t in I^t to B^{t+1} in I^{t+1} ;
end
For the overlapping-copied area, we use the average texture as the final texture ;
Finally, we fill the empty area in I^{t+1} with the same area in I^t ;

Compared to PC and CC, the size of local feature blocks is directly related to the Gaussian window W. As we discussed earlier, in order to represent local gradient, W should not be too large. So the moving block size is usually much smaller than those in correlation methods. The main advantage of motion estimation using local feature (LF) lies in its robustness with respect to non-significant cloud deformation (e.g. change of cloud shape or illuminance) where correlation methods such as [21] will easily fail. However, compared to correlation methods, its disadvantages are also obvious: the insufficiency and nonuniform distribution of motion vectors. These problems will result in much more empty area in the estimated frame, making the result worse than correlation methods overall.

5. CLOUD MOTION ESTIMATION (CC AND LF)

To overcome the problems of both CC and LF, we introduce a novel motion estimation algorithm using the motion detection result from both of them. We first get the motion vectors by combination of motion vectors from CC and those from LF. Depends on the image quality and extent of cloud shape change, we can assign different weights on motion vectors from CC and LF. Here in our implementation, we simple assign the same weight (as 1) to all the related motion vectors and compute the average motion vector. Let $\vec{V}_{cc}$ be a $n \times m \times 2$ matrix (x and y direction) of motion vector from CC, and p^t be the LF in frame t with corresponding motion vector $\vec{v}_p^t$. The combined motion vectors from CC and LF are denoted as $\vec{v}_{cc-lf}$. The new algorithm is detailed in Algorithm 1.

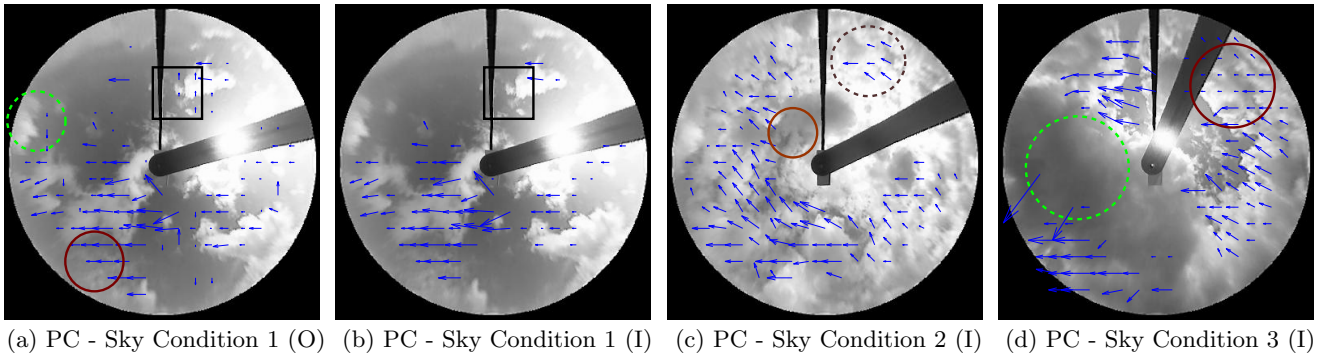
6. EXPERIMENTS

6.1 Experiments Setting

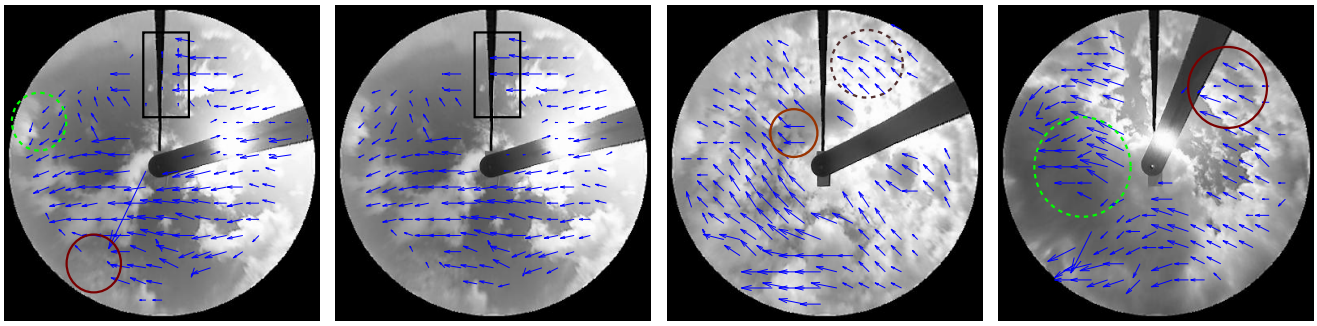
Methods and Evaluation We evaluated four motion vector detection algorithms: phase-correlation (PC), cross correlation (CC), local feature (LF), and the combination of cross correlation and local features (CCLF) on four representative sky conditions. Among these algorithms, CC is still being used by current similar researches [21]. Here we use previous image frame as our baseline approach (Baseline), and "Mean Absolute Error" (MAE) between the estimated image and the true image as our evaluation metric.

Data Selection and Preprocessing The data comes from Manus Island, Papua New Guinea, Tropical Western Pacific, 2010 collected by ARM [20]. Condition 1 shows multi-layer cloud with small sky coverage; Condition 2 shows multi-layer cloud with large sky coverage; Condition 3 shows thick and dark cloud; and Condition 4 shows slow moving cloud. For each condition, we choose four cases. Such diverse data choice is intended to study the robustness of motion vector detection and estimation but we presented only four representative sky conditions due to space limitation. Four preprocessing steps are applied before motion vector detection and estimation: 1) convert the input image format to grayscale; 2) scale the intensities of the input image to the full 8-bit intensity range [0, 255]; 3) undistort the image to a normal format (Figure 1); and 4) remove the shadowband and camera holding arm of TSI image (in our methods we use algorithm in [17] to detect the sun position and set such area as totally black).

Parameter Setting One of the main difficulties of applying motion vector algorithms to TSI images is deciding appropriate parameters. The accuracy of the estimated image depends on how the parameters are chosen. For correlation-based methods, there are three parameters: 1) block size, 2) overlapped area size and 3) search window size. In theory, the block size for both cross-correlation and phase-correlation should be kept large enough to ensure stable results but it should not be too big to make the number of motion vectors large enough. The size of the search window in cross-correlation must be sufficiently larger than the expected displacement but too large search window increas-



(a) PC - Sky Condition 1 (O) (b) PC - Sky Condition 1 (I) (c) PC - Sky Condition 2 (I) (d) PC - Sky Condition 3 (I)



(e) CC - Sky Condition 1 (O) (f) CC - Sky Condition 1 (I) (g) CC - Sky Condition 2 (I) (h) CC - Sky Condition 3 (I)

Figure 8: Cross Correlation (CC) and Phase Correlation (PC) Results. (O) stands for the original results without our proposed improvements and (I) is the improved results.

es the chance to capture spurious motion vector. In case of phase correlation, up to half of block size can be captured, but which is not realistic because the signal-to-noise ratio of correlation will be decreased with bigger motion vectors, resulted in decreasing the number of particle image pairings [3]. There are two parameters to tune for the local feature: 1) β for harris corner point detection and 2) σ for the Gaussian window size of the main orientation computation. For the first parameter β, the purpose is to get as many candidate as possible, and at the same time decrease the possibility of singular features. Determining the Gaussian window size σ of W is also not a trivial job. In accordance with the scale-space theory [9], σ cannot be either too big or too small. For small σ, the average orientation would be very erratic due to the noise or insufficiency of resolution. If σ is too big, it will fail to show the local orientation.

We did leave-one-out cross-validation (LOO-CV) to tune the parameters and evaluate the performances. In cross correlation, the range we used for block size is $[20, 30, 40, 50, 60]$, and search window size is among $[80, 100, 120, 140, 160]$. In phase correlation, the inner-block size is tested in $[5, 10, 15, 20]$ and outer-block size in $[20, 40, 60, 80]$. For local feature, we chose the best β from $[1, 1.5, 2, 2.5]$, and σ from $[3, 4, 5, 6]$. In our combination algorithm we select the best parameters for cross correlation and local feature separately.

6.2 Experimental Result Analysis

Motion Detection Using Correlation Methods.

From Figure 8, we can see that both phase correlation and cross-correlation can extract most of cloud motion vectors, but with differences in certain part. In Figure 8(a) and 8(e), the area of green dash circle and red solid circle has multi-layer cloud, whose moving direction is different from its neighborhood. In Figure 8(a) we can see that phase correlation either ignore the motion or just mark the motion similar to its neighborhood. In other word, phase correlation has inability to discriminate such multi-layer cloud presented in the same block. The reason is Fourier transform only choose the first p highest frequency. So the multi-layer cloud is considered as p-dimensional projections, which could be not enough to extract the right motion vectors. On the other hand, cross-correlation is much stronger on detecting multi-layer cloud motion by searching the blocks from luminance matches. Similar observation can be found inside the red solid circle and brown dash circle in Figure 8(c) and Figure 8(g). Another major weakness of phase correlation is in their inability to track motion of dark cloud (see the area inside green dash circle in Figure 8(d) and Figure 8(h)).

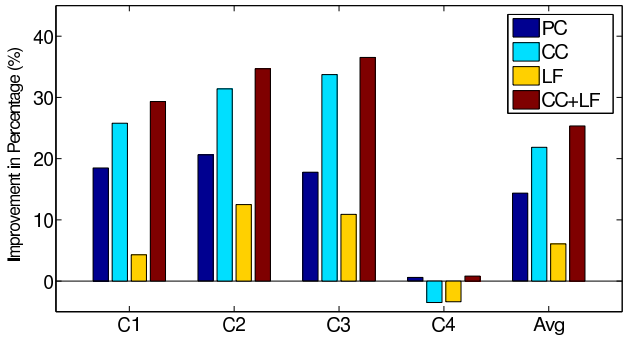
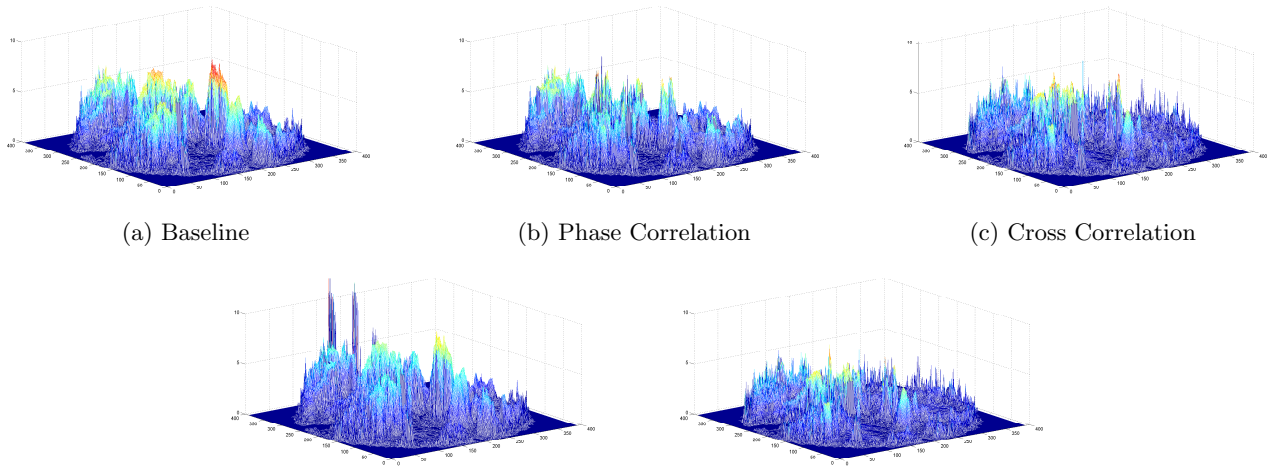


Figure 10: Error rate reduction over baseline



(a) Baseline (b) Phase Correlation (c) Cross Correlation

(d) Local Feature (e) Cross Correlation + Local Feature

Figure 9: Absolute Image Difference visualization of cloud condition 1 (multi-layer and scattered clouds)

The reason is that Fourier transform represent the area with high frequency signal. But the value of these signal is too small when the area gray scale is low enough, which precludes the program to find the right displacement between blocks. However, the complexity of phase correlation is only $O(N^2 logN)$, which is much less than cross-correlation $O(M^2N^2)$ where N is block size and M is search window size. We use the two extra techniques to improve our result. By using mean-filling, we recover the motion vectors near boundary (see comparison between the rectangular area in Figure 8(e) and 8(f)). We also remove those spurious vectors shown in Figure 8(a) compared with Figure 8(b).

Table 1: Mean Absolute Error Results

	Methods	C1	C2	C3	C4
1	Baseline	9.68	10.59	8.78	1.30
2	PC	7.80	8.44	7.05	1.29
3	CC	7.04	7.30	5.75	1.34
4	LF	9.20	9.32	7.80	1.33
5	CC+LF	**6.69**	**6.96**	**5.51**	**1.28**

Motion Estimation Result Analysis.

Overall, PC(phase-correlation) can get 14.4% improvement, local feature 6% and cross-correlation can reach up to 21.8%. The combination of cross-correlation and local feature has the highest average score 25.3% (Figure 10). From Table 1, we can see local feature can slightly decrease the MAE compared to just using previous frame as estimation. Phase correlation and cross correlation can get a much better estimation result. It is because local feature has insufficient area matching information due to the small local area surrounding each feature point. In other word, more uniformly distributed features are required, which could be non-trivial. Cross correlation outperform phase correlation, which confirms our motion vector detection analysis. Cross-correlation is much stronger in detecting multi-layer and dark cloud motion by searching the blocks using luminance pattern matches. The one case cross-correlation fail is in slow moving cloud condition. On the other hand, local feature can always detect such motion regardless of non-dramatical cloud velocity. Figure 9 shows an example Absolute Error plot of cloud condition 1. Baseline shows the biggest error mass and phase correlation and cross correlation significantly reduced it. One interesting point is that local feature error point distribution is different from cross correlation error point distribution. Therefore, when we combined both approaches, we could further reduce cross correlation error rates.

7. CONCLUSION

To provide an accurate prediction of solar radiation, we proposed to predict cloud movements using motion vector detection and estimation of TSI images. We implemented and improved two popular motion vector detection approaches, phase correlation and cross correlation. Due to inherent limitation of area-based approaches, we propose to consider a hybrid approach taking advantages of both correlation based approaches and local feature based approaches. The hybrid approach is robust to intensity and multi-layer clouds, to certain degree of cloud shape changes, and good for low resolution images compared with current similar research [21]. Our systematic experimental results revealed that our proposed novel approach significantly reduced error rate in predicting new cloud motion frames by 25% on average.

8. ACKNOWLEDGEMENTS

We gratefully thank all the anonymous reviewers for constructive suggestions toward paper improvement. This research is supported by a BNL LDRD project 10-001, and Brookhaven Science Associates LLC Under Contract No. DE-AC02-98CH109886.

9. REFERENCES

[1] Sandia National Laboratories, *Solar Energy Grid Integration Systems – Energy Storage (SEGIS-ES)*, DOE Program Concept Paper, 2008

[2] Sandia National Laboratories, *Solar Energy Grid Integration Systems "SEGIS"*, DOE Program Concept Paper, 2007

[3] C. E. Willert and M. Gharib, *Digital particle image velocimetry*, In Experiments in Fluids, 10(4), pages 181–193, 1991.

[4] J. Westerweel and F. Scarano, *Universal outlier detection for PIV data*, In Experiments in Fluids, 39(6), pages 1096–1100, 2005.

[5] C. M. Kishtawal, S.K. Deb, P. K. Pal and P. C. Joshi, *Estimation of Atmospheric Motion Vectors from Kalpana-1 Imagers*, In Journal of Applied Meteorology and Climatology, 48(11), pages 2410–2421, 2009.

[6] C. Harris and M. Stephens, *A combined corner and edge detection*, In Proc. Fourth Alvey Vision Conference, pages 147–151, 1988.

[7] J. Chen, J. Tian, N. Lee, J. Zheng, R. Smith, and A. Laine, *A partial intensity invariant feature descriptor for multimodal retinal image registration*, In IEEE Trans. Biomed. Eng, 2010.

[8] A. M. Bazen and S. H. Gerez, *Systematic methods for the computation of the directional fields and singular points of fingerprints*, In IEEE Trans. Pattern Anal. Mach. Intell., 24(7), pages 905–919, 2002.

[9] T. Lindeberg, *Scale-Space Theory in Computer Vision*, Kluwer Academic Publishers, 1994.

[10] D. G. Lowe, *Distinctive image features from scale-invariant keypoints*, In Int. J. Comput. Vis., 60(2), 2004.

[11] J. Beis and D. G. Lowe, *Shape indexing using approximate nearest neighbor search in high dimensional spaces*, In Proc. Comput. Vis. Pattern Recog., pages 1000–1006, 1997.

[12] R. Verdąäu-Monedero and J. Angulo, *Spatially-variant directional mathematical morphology operators based on a diffused average squared gradient field*, In Advanced Concepts for Intelligent Vision Systems, pages 542–553, 2008.

[13] R. R. Rogers, *A Short Course in Cloud Physics, 2nd Edition.*, Pergamon Press, Oxford, England, 1979.

[14] D. I. Barnea and H. F. Silverman, *A class of algorithms for fast digital image registration*, In IEEE Trans. Computers, 21, pages 179–186, 1972

[15] M. Kass, A. Witkin and D. Terzopoulos, *Snakes: Active contour models*, In Proc. 1st Int. Conf. on Computer Vision, pages 259–268, 1987.

[16] A. R. Lindsey, *The Non-Existence of a Wavelet Function Admitting a Wavelet Transform Convolution Theorem of the Fourier Type*, Rome Laboratory, 1995.

[17] I. Reda and A. Andreas, A, *Solar position algorithm for solar radiation application*, National Renewable Energy Laboratory (NREL) Technical report, 2003.

[18] T. Chanwimaluang, G. Fan, and S. R. Fransen, *Hybrid retinal image registration*, In IEEE Trans. Info. Tech. Biomed., 10(1), pages 129–142, 2006.

[19] P. Glomb, *Detection of interest points on 3D data: Extending the harris operator*, In Computer Recognition Systems, 57, pages 103–111, 2009.

[20] *TSI Images 2010*, http://www.arm.gov/instruments/tsi, Atmospheric Radiation Measurement (ARM) Climate Research Facility, 2010.

[21] C. W. Chowa, B. Urquharta, M. Lavea, A. Domingueza, J. Kleissla, J. Shieldsb and B. Washomc. *Intra-hour forecasting with a total sky imager at the UC San Diego solar energy testbed*, In Solar Energy, 85(11), pages 2881–2893, 2011.

[22] J. R. Bergen, P. Anandan, K. J. Hanna, and R. Hingorani. *Hierarchical model-based motion estimation*, In Proc. Second ECCV, pages 237–252, 1992.

[23] J. P. W. Pluim, J. B. A. Maintz, and M. A. Vierveger. *Mutualinformation-based registration of medical images: a survey*, In IEEE Trans. Med. Imaging., 22(8), pages 986–1004, 2003.

[24] A. Albiol, M. J. Silla, A. Albiol, and J. M. Mossi, *Video analysis using corner motion statistics*, In Proc. IEEE International Workshop on Performance Evaluation of Tracking and Surveillance, 2009.

[25] G. Yang, C. V. Stewart, M. Sofka, and C. L. Tsai *Automatic Robust Image Registration System: Initialization, Estimation and Decision*, In ICVS, 2006.