A Rapid Video Frame Correspondence Algorithm for Agricultural Video Field Surveying

Dev Sagar Shrestha
*University of Idaho*

Brian L. Steward
*Iowa State University, bsteward@iastate.edu*

Kelly R. Thorp
*Iowa State University*

Bo Li
*Iowa State University*

Follow this and additional works at: [http://lib.dr.iastate.edu/abe_eng_conf](http://lib.dr.iastate.edu/abe_eng_conf)  
Part of the *Bioresource and Agricultural Engineering Commons*

The complete bibliographic information for this item can be found at [http://lib.dr.iastate.edu/abe_eng_conf/36](http://lib.dr.iastate.edu/abe_eng_conf/36). For information on how to cite this item, please visit [http://lib.dr.iastate.edu/howtocite.html](http://lib.dr.iastate.edu/howtocite.html).
A Rapid Video Frame Correspondence Algorithm for Agricultural Video Field Surveying

Dev S. Shrestha, Assistant Professor
Biological and Agricultural Engineering, University of Idaho, 81 JML, Moscow, ID 83844.
email: devs@uidaho.edu.

Brian L. Steward, Assistant Professor; Kelly R. Thorp and Bo Li, Graduate Research Assistants
Agricultural and Biosystems Engineering, Iowa State University, Ames, IA 50011.

Written for presentation at the
2004 ASAE/CSAE Annual International Meeting
Sponsored by ASAE/CSAE
Fairmont Chateau Laurier, The Westin, Government Centre
Ottawa, Ontario, Canada
1 - 4 August 2004

Abstract. Video frame correspondence is a key operation in video processing for image-based video field surveying for precision agriculture applications. Video frame correspondence takes significant processing time and can be a major constraint in real time video processing applications. A Kalman filter was used to predict future shifts from previously measured shifts, and a gradient ascent method was developed to search for the maximum normalized cross correlation in the vicinity of predicted shifts. Compared with the results from the previous minimum error method developed by authors, the time required to compute the shift for a 30 by 30 pixel image patch with a 90 by 90 pixel search region was approximately ten times less than searching for a match over the entire search region. In a Matlab® script implementation, only 9.5 seconds were required to find the correspondence between 500 video frames of a corn field using the new algorithm whereas with the minimum error method, 114.6 seconds were required. The gradient ascent method with Kalman shift prediction can be used to find the image shift for real time applications. However, the success of the process depends on the closeness of the predicted shift to the actual shift and the characteristics of the correlation surface.

Keywords: real time, machine vision, image processing, video processing, precision agriculture, crop sensing
Introduction

Precision agriculture (PA) is an important technological development in contemporary agriculture for managing the spatial variability that naturally occurs in crop production (Schueller et al., 2002). The National Research Council (1998) refers to PA as a management strategy that uses information technologies to bring data from multiple sources to bear on decisions associated with crop production. The key idea behind PA is to measure and manage spatial variability to optimize the crop production system. Spatial variability can be categorized into yield, field, soil, crop, and management variability. Bottlenecks in successful application of PA include a lack of (1) developed sensing technologies needed to adequately characterize field-scale spatial variability, (2) flexible data acquisition and processing systems that can be deployed in a field to gather and process data, and (3) agronomic knowledge relating crop inputs and plant response to those inputs.

As one solution to these limitations in PA, researchers have developed machine vision systems, which utilize cameras and data acquisition equipment on ground-based vehicles to collect images and video of crop fields. With rapid advances in multimedia computing, it is possible to do video field surveying in which geo-referenced crop parameters can be extracted at a high resolution and mapped for significant portions of fields. A challenge in the development of such a system usually lies in the algorithm to robustly extract the information of interest from the images or video under real-time constraints. In past research, machine vision-based algorithms have been developed to estimate several crop and field parameters such as weed infestations (El-Faki et al., 2000; Tang et al., 2000), plant shape and size (Nishiwaki et al., 2001), plant population (Shrestha and Steward, 2003) and plant height (Shrestha et al., 2002).

Video frame sequencing, the process of determining the amount of overlap in succeeding frames, is an image correspondence problem in which common scene points in two frames are identified and matched. There are many methods available in the literature for image correspondence. One technique uses a matching criterion as a measure of correlation to match a patch in the one frame to patches within a search region in the following frame (Sonka et al., 1999). Feature–based image correspondence, such as the method developed by Dai and Khorram (1999), is another possible approach for matching remotely sensed image pairs. Sanchiz et al. (1995) also developed a feature–based system to sequence the video frames in fields containing small cabbage plants. However, feature–based algorithms for video frame correspondence are computationally expensive.

Image correspondence can be done both in spatial and frequency domains. In the frequency domain, image correspondence can be achieved with sub–pixel accuracy, but the computational cost is higher than spatial correlation–based image matching (Averbuch and Keller, 2002). Correspondence is a key problem in machine vision applications and no generally reliable solution exists (Maciel and Costeira, 2002).

Shrestha and Steward (2003) developed a software architecture that can be used to implement a video field surveying system including data collection, image and video processing, crop parameter extraction and geo-referencing processes. This architecture was designed specifically for machine vision applications such as estimation of plant population and spacing, where composite images of a crop row must be constructed from a series of video frames.
In the method developed by Shrestha and Steward (2003), called the minimum error method, a patch was selected within an image such that the expected match on the next frame does not fall outside of image boundary (Fig 1).

Figure 1. For image sequencing, an image patch X in frame n was searched for the best match within search region in frame n+1 in search. The difference in coordinates of the patch matched to the second frame gives the amount of shift from the current frame to the next.

If the patch was m×n pixels and search region was M×N pixels, the matching error for each position was determined by:

\[
\text{Err}_{p,q} = \sum_{i=1}^{m} \sum_{j=1}^{n} |P_{ij} - S_{i+p-1,j+q-1}|
\]  \hspace{1cm} (1)

where, Err is the M-m × N-n error matrix. The (p, q) term of Err corresponds to the sum of absolute errors when the patch was shifted by (p, q) pixels from the upper left corner of search region. P is the intensity patch from the current frame, and S is the search region from the next frame (Fig. 2). A candidate match was found by finding the minimum valued element in Err. To determine the validity of a match, the minimum value of Err had to be significantly lower than other values. In order to test for a statistically significant minimum, the values in Err were sorted in ascending order, and the difference between successive values was calculated. For a valid match, the difference between the lowest error and the next to the lowest error value was required to be higher than 5 standard deviations (σ) from the mean of the remaining error differences. For example, the error matrix for figure 2 was calculated as:

\[
\text{Err} = \begin{bmatrix}
3.3 & 4.3 & 4.3 \\
2.8 & 0.0 & 3.0 \\
4.1 & 2.6 & 3.8
\end{bmatrix}
\]  \hspace{1cm} (2)

The matrix Err was rearranged in a row of ascending values, and the difference ΔErr was calculated as:

\[
\Delta \text{Err} = \begin{bmatrix}
2.6 & 0.2 & 0.2 & 0.3 & 0.5 & 0.3 & 0.2 & 0.0
\end{bmatrix}
\]  \hspace{1cm} (3)

Since the first value of ΔErr i.e. 2.6 is more than 5 standard deviations from the mean of the rest of the differences, the minimum error 0.0 in the Err matrix was considered to be a true minimum and the match was accepted. If a valid match, based on a 5 σ criteria, could not be found in the specific region, then another random patch was chosen in current frame and searching was repeated.
The main drawback of this image correspondence algorithm was the time it took for computation. For real time applications, video frame correspondence must be performed faster than the frame rate. One method for solving the problem is to predict what the shift will be for current video frames based shifts of previous frames. Such information would help to minimize the size of the search region, and thus the computational effort required to find matching frames may be reduced. Kalman (1960) developed his well known filtering technique which has potential for application to this prediction problem.

![Error Matrix](image)

**Figure 2.** Process of calculating an error matrix, Err. The patch was shift over the search region. For the position shown above Err_{1,1} = |(0.0-0.1)|+|(0.4-0.3)|+…+|(0.2-0.0)| = 3.3.

The objectives of this research were to (1) develop a rapid frame correspondence method for real time video processing in a video field surveying system, and (2) compare the performance of such a method with the previous minimum error method.

## Methodology

Video of corn rows at an early growth stage was acquired with a vehicle mounted camera. The gradient ascent algorithm was developed which consisted primarily of two processes: first and second frame matching where the prediction shift could only a generally estimated and N/N+1 matching where the shift was predicted from previous shifts. The gradient ascent algorithm was compared with the previous minimum error method.

## Equipment

A Sony DCR-TRV900 digital camcorder was mounted on a John Deere Gator utility vehicle at 0.60 m above the ground with a 0.30 m by 0.40 m field of view. Each captured image size was 480 × 720 pixels with 24 bit color resolution. The shutter speed was adjusted to 1/1000 second, frames were captured in progressive scan mode, and other camera settings were set to auto. In the field, the video stream was recorded on a miniDV tape. The vehicle was driven along the row capturing the video of corn rows from above.
**First and Second Frame Correspondence**

For the sequencing of the first two frames, there is no prior estimate of the shift. However, depending on the expected vehicle speed at the time of the start and direction of vehicle movement, maximum possible shift can be estimated using the equation:

\[ s = \frac{vr}{3.6f} \]  

(4)

where, \( s \) is the expected shift in pixel per frame; \( v \) is the speed of the vehicle in km/h; \( r \) is the conversion factor from physical length on the ground surface to pixels (pixels/m) and \( f \) is numbers of video frames per second. From the maximum and the minimum possible shifts, a valid region in the first frame was calculated so that the possible matching scene would still be inside the second frame (Fig 3). For example if the maximum initial velocity is \( \pm 4 \) miles per hour, with NTSC video standard 29.97 frames per second and 1400 pixels per meter of ground surface, from equation 4, \( s = \pm 52 \) pixels. Therefore, for a 480 (height) \( \times \) 720 (width) frame size, the minimum Y coordinate of valid region (Fig 3) should be at least 52 pixels and maximum Y coordinate of valid region should not be greater than 428 (= 480-52) pixels.

If the Y axis of the image frame is parallel to the direction of travel, the shift in the X axis can be assumed to be zero for the first two frames. This assumption greatly reduces the amount of calculation needed for cross correlation coefficients computation to search for the best match in first two images. Instead of randomly choosing a patch (Fig 3) inside the valid region, four corners of the valid region were selected as a candidate patches. The corner areas were chosen because it is less likely that all of the corners will have a plant at the same time since they were the points furthest apart.

![Diagram](https://example.com/diagram.png)

**Figure 3.** Frame correspondence of the first two frames. A small patch from the first frame was searched in the search region of second frame. Frame shift = \((x_1-x_0,y_1-y_0)\)

Among four corners of the valid region, upper left corner was chosen as the first candidate. The selected patch was then checked for contained vegetation or was too high or too low of an intensity. To check for vegetation, the patch was subdivided into 5 by 5 pixel sub-patches which were segmented using the truncated ellipsoidal segmentation method developed by Shrestha and
Steward (2003). If the majority of pixels in a sub-patch were segmented as vegetation pixels, that sub-patch was categorized as a vegetation sub-patch. If more than five percent of the sub-patches in a patch were vegetation sub-patches than that region was classified as a vegetation region. If average intensity of the patch was less than 0.1 or greater than 0.9 in a 0 to 1 range, then the patch was classified as being outside the intensity range. If it was a vegetation region, then the upper right corner patch was reselected. If that patch was a vegetation region, the next corner patch was selected going clockwise until a valid patch was found. If all four corners were found to be covered with plants, it was assumed to be so because of the one of the following reasons:

1. Corn plant leaves are in all four corners of the frame.
2. The entire frame contains vegetation due to the camera being over a grassy region.
3. The corn plants are at a later growth stage and are in the entire camera field of view.
4. High segmentation noise exists.

To check which reason led to all of the corners being classified as vegetative regions, the entire valid region was segmented. The following criteria were used to distinguish between the different cases:

If more than 95% of the valid region consisted of vegetative pixels, then it was assumed that grass or corn plants filled the camera field of view and no attempt was made to count the plants, and the frame was discarded. If less than 50% of the frame was vegetation, than it was assumed that vegetation in the four corners occurred by chance. If the vegetative region was greater than 50% but less than 95%, it was considered to be an area with high weed density or high segmentation noise. In either of the latter two cases, the entire valid region was segmented, and the region with minimum plant density was selected for patch matching.

Once a valid patch was selected, the normalized cross correlation coefficient (Haralick and Shapiro, 1993) was calculated for each position of the patch as it was shifted over the search region and the shift, $z$, which was the best match between the two frames was found using the equation:

$$
z = \max_u \frac{\text{Cov}[g_1(x, -u), g_2(x)]}{\sqrt{V[g_1(x, -u)]V[g_2(x)}]}
$$

where, $g_i(x)$ is the observed intensity of pixel at location $x$. $u$ is an unknown shift, $\text{Cov}$ is covariance and $V$ is the variance. Normalization minimizes the local intensity effects of a frame. Taking the covariance of intensities reduces the average brightness change from one frame to another frame and normalizing with square root of the product of the variances reduces the effect of contrast change in two frames. The unknown shift value $u$ which maximizes the right side of Eqn. 5 is the shift that matches the frames.
**N/N+1 Frame Correspondence**

Once a match is found for the first and the second frames, the shift that matched those two frames was used to predict the next shift. A Kalman filter was used to make a prediction-correction model of the frame shifts from prior patch matching results.

The state vector, $x_k$, for the Kalman filter model was defined to be:

$$ x_k = \begin{bmatrix} s_x & a_x & s_y & a_y \end{bmatrix} $$

where, $s_x$ and $s_y$ are the shifts of the $k$th frame in both X and Y directions (Fig 3) from the previous frame, $s$ is frame shift, and $a$ is measured vehicle acceleration. Subscript $x$ and $y$ indicates measurement in X and Y directions. The state vector for the $k+1$ frame i.e. $x_{k+1}$ can be estimated as:

$$ x_{k+1} = \Phi_k x_k + w_k $$

where $\Phi_k$ is the state transition matrix that relates the state vector at the next time step to the current state and $w_k$ is process noise. The process noise is assumed to be white with a known covariance structure. The effect of vehicle acceleration was captured by matrix $\Phi_k$. If $z_k$ was the measured shift and acceleration of the vehicle with some measurement noise $v_k$ then,

$$ z_k = H_k x_k + v_k $$

where, $H_k$ is the matrix that connects the measurement vector and the state vector, and $v_k$ is the measurement noise which is assumed to be white with known covariance matrix and having zero cross correlation with $w_k$.

**Defining the State Transition and Measurement Matrices**

The vehicle acceleration in either direction is captured by the state transition matrix, $\Phi_k$. If the video rate is $f$ number of frames per second then sampling interval would be $1/f$ seconds. If the frame shift between two frames be $s_k$ pixels, then velocity of the vehicle can be estimated as:

$$ v_k = s_k f $$

Vehicle acceleration can be estimated by the change in velocity between two frame samples and is given by:

$$ a_k = \frac{v_{k+1} - v_k}{1/f^2} = (s_{k+1} - s_k) f^2 $$

or,

$$ s_{k+1} = s_k + a_k f^2 $$

Then the state equation becomes:
During frame correspondence, we measure displacements $s_x$ and $s_y$ directly but we do not measure acceleration directly. Using the last two shift measurements to estimate the acceleration, results in the following measurement equation:

$$
\begin{bmatrix}
    s_{x-1} \\
    s_x \\
    s_{y-1} \\
    s_y
\end{bmatrix} =
\begin{bmatrix}
    1 & -1/f^2 & 0 & 0 \\
    1 & 0 & 0 & 0 \\
    0 & 0 & 1 & -1/f^2 \\
    0 & 0 & 1 & 0
\end{bmatrix}
\begin{bmatrix}
    s_x \\
    a_x \\
    s_y \\
    a_y
\end{bmatrix} +
\begin{bmatrix}
v_k
\end{bmatrix}
$$

(13)

If acceleration is not measured during correspondence, deriving acceleration from measured shifts adds no new information to the system. No matter how many terms are used in estimating acceleration, it is still a linear combination of shift measurements, and it can be shown that in such cases, $E(w_i w_i^T) \neq 0$ for $i \neq k$. Therefore the acceleration term should be dropped from such a model and Eqns. 12 and 13 reduce to:

$$
\begin{align*}
    \mathbf{x}_{k+1} &= \begin{bmatrix} s_x \\ s_y \end{bmatrix} + \mathbf{w}_k \\
    \mathbf{z}_k &= \begin{bmatrix} s_x \\ s_y \end{bmatrix} + \mathbf{v}_k
\end{align*}
$$

(14)

(15)

In this case the acceleration is lumped into the measurement and prediction errors.

**Estimation of Q and R**

The random noise in the prediction process and measurement process were estimated by measuring the shift with the minimum error method and comparing with predicted and measured values. From the error in predicted shifts and measured shifts comparing to manually measured shifts, the covariance matrix of $\mathbf{w}_k$ and $\mathbf{v}_k$ ($\mathbf{Q}$ and $\mathbf{R}$ respectively) were estimated as:

$$
\mathbf{Q} = E[(\mathbf{x}_{k+1} - \Phi_k \mathbf{x}_k)(\mathbf{x}_{k+1} - \Phi_k \mathbf{x}_k)^T]
$$

(16)

$$
\mathbf{R} = E[(\mathbf{z}_k - \mathbf{H}_k \mathbf{z}_k)(\mathbf{z}_k - \mathbf{H}_k \mathbf{z}_k)^T]
$$

(17)

Acceleration of the vehicles was not measured separately, so it was dropped from the model. The covariance matrix $\mathbf{Q}$ was estimated from measurements of shifts using the minimum error method. The measured values were cross checked by sequencing video frames and looking for any visual error in resulting composite image.

$$
\mathbf{Q} = \begin{bmatrix} 10 & 3 \\ 3 & 23 \end{bmatrix}
$$

(18)
The covariance matrix $R$ was estimated from the shifts measurements of the sections that were believed to have a relatively constant acceleration. The average shift was considered to be the true shift and the measurement error covariance was calculated as:

$$
R = \begin{bmatrix} 14 & 2 \\ 2 & 9 \end{bmatrix}
$$

(19)

**A priori and A posteriori Covariance Matrices**

The discrete Kalman filter uses an a priori estimate of $x_k$ written as $\hat{x}_k^-$ with associated error covariance matrix $P_k^-$. When we begin the correspondence of frames, we used the shifts measured with the normalized cross correlation over the entire region to gain the prior knowledge about the process and estimate the covariance of the frame shifts. Also the shift obtained from the correspondence of 1st and 2nd frame should be considered as an estimation of initial $\hat{x}_k^-$. Using the linear Kalman model, the a posteriori estimation of $x_k$ was written as $\hat{x}_k$ with the covariance matrix $P_k$ estimated as:

$$
\hat{x}_k = \hat{x}_k^- + K_k(z_k - H_kx_k^-)
$$

(20)

where $K_k$ is Kalman gain. The value of the Kalman gain that minimizes the sum squared error is given by:

$$
K_k = P_k^-H_k^T(H_kP_k^-H_k^T + R)^{-1}
$$

(21)

For the optimal gain condition, a posteriori covariance matrix is estimated as:

$$
P_k = (I - K_kH_k)P_k^-
$$

(22)

A high value of the a priori error covariance was assumed. It was found that the a priori error covariance quickly converged to a final value regardless of the amount of covariance assumed initially. For the sake of evaluation, $P_k^-$ was assumed:

$$
P_k^- = \begin{bmatrix} 200 & 0 \\ 0 & 200 \end{bmatrix}
$$

(23)

After prior estimation of $Q$, $R$, $\hat{x}_k^-$ and $P_k^-$, the next estimation of $\hat{x}_k$ was calculated using Kalman loop (Fig. 4).
1. Compute Kalman gain:
\[ K_k = P_k H_k^T (H_k P_k H_k^T + R)^{-1} \]
2. Get measurements
3. Update estimate with measured value
\[ \hat{x}_k = \hat{x}_k + K_k (z_k - H_k \hat{x}_k) \]
4. Compute and update the a posteriori error covariance
\[ P_k = (I - K_k H_k) P_k^- \]

Start:
Enter prior estimates \( Q, R, \hat{x}_-^k \) and \( P_-^k \)

Predict ahead:
\[ P_{k+1}^- = \Phi_k P_k^- \Phi_k^T + Q \]
\[ \hat{x}_{k+1}^- = \Phi_k \hat{x}_k \]

Figure 4. Kalman loop. \( Q, R, \hat{x}_-^k \) and \( P_-^k \) are known or estimated parameters from the prior knowledge about the process (Adapted from Brown and Hwang, 1997).

**Gradient Ascent Method**

Once the next shift is predicted, it was assumed that the predicted shift was close to the actual shift and the gradient of the normalized cross correlation surface pointed to the global maximum (Fig. 5). This assumption obviously requires high prediction accuracy. Normalized cross correlation was calculated between the selected patch and a search region that was one pixel wider than the patch in all directions. The search region was shifted in the direction of the maximum normalized cross correlation gradient, and the normalized cross correlation was recalculated for this new search region. This process was repeated until the gradient was less than or equal to zero in all directions meaning the local correlation for that region had been found.

In case of poor prediction, there is a possibility of getting caught in a local correlation maxima and never finding the true shift. If, for example, the search starting point was B in Fig. 5, it would have found the global maximum. This can lead to frame matching errors, and so each shift estimate must be checked for validity. Two checks have been investigated. The first is checking to see if the correlation at the estimated shift is greater than 0.5. Second, it is to check to see if the difference between the estimated shift and the prior shift is not larger than 10 pixels. In the minimum error method, errors were less likely because every possible shift a larger search region was investigate leading to a computationally demanding algorithm.

**Analysis of Performance**

Theoretically, the number of mathematical operations was calculated for both algorithms to determine the potential performance advantage that was available with the gradient ascent algorithm. The algorithm was tested with a sequence 500 frames with both the minimum error and gradient ascent methods. A 30 by 30 pixel size patch was used for evaluation with a 90 by 90 pixel search region. The time to do the sequencing using both algorithms was recorded and compared. In addition, the sequence of detected shifts was recorded and compared.
Results and Discussion

In the algorithm developed by Shrestha and Steward (2003), the number of mathematical operations to calculate minimum error in patch size of $m \times n$ in $M \times N$ search region size would be on the order of $(m \times n)(M-m+1)(N-m+1)$. For the gradient ascent algorithm, the normalized cross correlation was calculated for search region of size $M = m+2$ and $N = n+2$ only. The number of iterations would vary based on how close the global maximum is from the search starting point. If the predicted shift is equal to actual shift, in which case only a minimum number of calculations would be needed, on the order of $9(m \times n)$. In the minimum error method, as used by Shrestha and Steward in their original paper, $M = 90$, $N = 90$, $m = 30$, $n = 30$ the numbers of operations needed in searching for the entire region would be 3,348,900. In the gradient ascent algorithm, it is possible to find the minimum value in as little as 8100 operations. However many iterations may be required to find the nearest maximum. In this particular case, with a 30 by 30 pixel search region and a 90 by 90 pixel search region, maximum number of iterations required is 60 assuming that the global maximum is one of the four corners in search region. Therefore, the maximum number of operations needed to find the peak is limited to 486,000 operations, which is only 14.5 % of the operations needed with the algorithm that searches for a match in the entire region.

The shift estimates in the direction of travel from the gradient ascent method closely followed the those estimated from the minimum error method (Fig 6). However the local variation of shifts was more stable in gradient ascent method. Both the minimum error method and the gradient ascent method had mean shifts of 53 and 3 pixels per frame in the direction of travel and perpendicular to the direction of travel, respectively. However, the time required to sequence this video decrease from 114.6 sec. for the minimum error method to 9.5 sec. for the gradient ascent method.
Figure 6. Comparison of the frame shifts measured using normalized cross correlation with a 30 by 30 pixels patch and 90x90 pixel search region along the direction of travel. The gradient ascent method with Kalman prediction closely followed the shift pattern obtained from normalized cross correlation method.

Figure 7. Comparison of the frame shifts measured using normalized cross correlation with 30 by 30 pixel patch and 90 by 90 pixel search region across the direction of travel. The shift measured from both methods showed that the measurements were mostly random with mean of 3 pixels. The estimated lateral shifts from the gradient ascent algorithm followed the general pattern of shifts estimated by the minimum error algorithm (Fig. 7). The average of three pixels shift per
frame indicates that either the camera was rotated relative to the vehicle centerline or the vehicle was yawing during this video segment. The random variation in shift amount was partially due to vehicle roll from frame to frame and partially due to sequencing uncertainty. For both algorithms, the variance of the shift estimates was similar.

The algorithm was implemented in Matlab®, script, which is an interpreted language and is slower in run time than its compiled counterpart particularly in loops. It is expected that the implementation of this code in C++ or other compiled language will be much faster. Even with the Matlab implementation, the gradient ascent method calculated the shifts in real time. That is, the video frame shifts for 31 seconds of video were calculated in 9.5 seconds.

**Conclusion**

The gradient ascent method with Kalman filter prediction can be used to improve the image correspondence speed for video processing application in ground-based vehicle mounted video field surveying applications. The patterns of estimated frame shifts measured using normalized cross correlation of fixed sized patch and shifts measured using the gradient ascent method with Kalman shift prediction were similar.

The time required for image correspondence decreased over 10 times by using the gradient ascent method. This method requires, however, precise shift prediction close to actual shift value and shift validity must be checked for errors.

**Acknowledgements**

This journal paper of the Iowa Agriculture and Home Economics Experiment Station, Ames, Iowa, Project No. 4003, was supported by Hatch Act and State of Iowa funds. Additional research support was provided by the Iowa State University Center for Advanced Technology Development.

**References**


