Machine Vision and Applications (2005) 16(4): 203–216
DOI 10.1007/s00138-005-0173-x

ORIGINAL PAPER

Zhigang Zhu · Edward M. Riseman ·
Allen R. Hanson · Howard Schultz

An efficient method for geo-referenced video mosaicing for environmental monitoring

Received: 25 May 2004 / Accepted: 16 October 2004 / Published online: 10 June 2005

Abstract Environmental monitoring applications require seamless registration of optical data into large area mosaics that are geographically referenced to the world frame. Using frame-by-frame image registration alone, we can obtain seamless mosaics, but it will not exhibit geographical accuracy due to frame-to-frame error accumulation. On the other hand, the 3D geo-data from GPS, a laser profiler, an INS system provides a globally correct track of the motion without error propagation. However, the inherent (absolute) errors in the instrumentation are large for seamless mosaicing. The paper describes an effective two-track method for combining two different sources of data to achieve a seamless and geo-referenced mosaic, without 3D reconstruction or complex global registration. Experiments with real airborne video images show that the proposed algorithms are practical in important environmental applications.

Keywords Image registration · Video mosaicing · Motion analysis · Geo-reference image · Environmental modeling

1 Introduction

A critical issue among nations in the coming decades will be how to manage the use of land and natural resources. Unfortunately, the use of satellite data has not enabled general and automatic ecosystem modeling because many of the dynamic changes of interest in ecosystems take place at a finer level of resolution than is available. Thus, using high-resolution low-altitude video sequences is highly required for interpreting the lower-resolution data. Our interdisciplinary NSF environmental monitoring project aims at developing a methodology for estimating the standing biomass of forests. The instrumentation package mounted on an airplane consists of two bore-sighted video cameras (one telephoto and one wide-angle), a Global Position System (GPS), an Inertial Navigation System (INS), and a profiling pulse laser. The previous manual approach used by our forestry experts (e.g. [1]) utilized only a fraction of the available data due to the labor involved in hand interpreting the large amount of video data. A more compact representation and more flexible interactive visualization interface are clearly demanded.

Our long-term goal is to develop automated tools that can correlate video mosaics from high-resolution low-altitude video sequences (both wide-angle video and telephoto video) with lower-resolution high-altitude aerial image data or satellite image data that are of lower spatial resolution, as a tool for interpreting the lower-resolution data. However, directly matching video streams with satellite images will be very difficult, since they have significantly different spatial resolution, color and perspective views. For this reason, generating video mosaics that tie to the same geographical reference will be very useful, not only for the long-term goal but also as an intermediate representation for environmental study, and this is the specific goal of this paper. We have demonstrated the significant potential of the geo-referenced stereo mosaics through a set of initial collaborative projects with environmental science partners, in regions of New England, Bolivia, Brazil, and Madagascar.

Creating panoramic images and high-quality mosaic images from video sequences (or a collection of images) has attracted significant attention in the research community, industry, and government [2–20]. Applications span a variety of fields, including panoramic photography, video compression, surveillance, and virtual environments. The existing mosaic methods can be roughly divided into three classes: cylindrical/spherical mosaics [2–9], manifold mosaics [10–14], and geo-referenced mosaics [15–20]. We will discuss each of them briefly.

Z. Zhu (✉)
Department of Computer Science, City College of New York/CUNY, Convent Avenue and 138th Street, New York, NY 10031, USA
E-mail: zhu@cs.ccny.cuny.edu
URL: http://www-cs.engr.ccny.cuny.edu/∼zhu/

E. M. Riseman · A. R. Hanson · H. Schultz
Department of Computer Science, University of Massachusetts at Amherst, MA 01003, USA

1.1 Cylindrical/spherical mosaics

In generating full-view (360°) cylindrical mosaics, a camera pans around a scene to obtain a complete description of the surrounding environment. The basic motion model is camera rotation so there is no (significant) motion parallax involved. For example, Apple's QuickTime VR [2] captures a 360° panoramic image of a scene with a camera rotating horizontally from a fixed position. The overlap in images is registered first by the user and then "stitched" together by the software at the best match. Kang and Weiss [3] analyzed the error in constructing panoramic images and proposed a technique that has the advantage of not having to know the camera focal length *a priori*. In order to create a panorama, they first had to ensure that the camera is rotating about an axis passing through the nodal point. The correct focal length is determined by iterating the process of projecting original video images onto cylindrical surfaces given an estimate of the focal length. In other work, in order to generate panoramic mosaics from video on a hand-held camcorder, Sawhney et al. [4] provided a method for automatic detection of a loop closure to warp the conic mosaic into a cylindrical mosaic. Zhu et al. [5] proposed a similar methodology independently to deal with more complex camera motion—3 degree of freedom (DOF) rotation, zoom and small translation. Due to scale change and accumulating error, this required warping from a deformed conic mosaic to a cylindrical panorama. Generally speaking, in this class the loop closure constraint is used to connect first and last matched frames of a 360° full-view video sequence.

Since most of the mosaic methods operate on video images in a sequential, pairwise manner, small errors in registration accumulate from one pair of images to the next. These errors are unavoidable if no other constraints are provided. Full-view panoramic mosaic generation tries to solve this problem by matching the last frame with the first frame and forcing the original mosaic to warp to a cylinder [4, 5]. Some researchers use more general global constraints to ensure that the final mosaic (composed of all the images) is globally registered [6–8]. Shum and Szeliski [6] proposed a global registration strategy for a full-view panorama, which establishes point correspondences in a set of images. Minimizing the projected difference of these points results in global alignment; however, the search could be quite slow to determine many point correspondences. Sawhney et al. [7] developed a local-to-global algorithm that uses constraints between non-consecutive but spatially neighboring frames. A global consistency estimation of alignment parameters is iteratively performed in order to match each frame to a consistent mosaic coordinate system. The large number of parameters makes computation prohibitive for more than a few frames. Practical application of this algorithm requires efficient optimization strategies. Davis [8] provided an efficient method for finding a globally consistent registration of all images by solving a sparse linear system of equations. However, the sparse linear system is valid only if any image can register only with a few other images. A full-sphere or hemisphere mosaic can also be generated by rotating a camera, for example as in [9].

1.2 Manifold mosaics

In this class, the motion of the camera is not constrained only to pure rotation; however, usually a planar scene assumption needs to be made under a more general motion model in order to use a parameter-based transformation. Mann and Picard [10] discussed different transformation models—affine, bilinear, projective and pseudo-projective–to register and reduce the set of images into a single, larger composite frame. The final image mosaic is not a full 360° view, nor is 3D geometrical correctness guaranteed. Peleg and Herman [11] use manifold projection to enable the fast creation of low-distortion panoramic mosaics under a more general motion than exact panning. The basic principle is the alignment of the strips that contribute to the mosaic, rather than the alignment of the entire overlap between frames. The aim of the mosaic is for rendering in applications related to virtual walk-throughs rather than applications requiring geometric precision. Manifold mosaicing is based on an early work on panoramic view image generation for robot navigation [12]. Recently, this approach was extended to mosaics with adaptive manifold [13] and crossed-slits projection [14] for image-based rendering.

1.3 Geo-referenced mosaics

Recently, there have been a few reports on geographically corrected ("geo-referenced") mosaics [15–20]. Kumar et al. [15] presented a geo-registration method that consists of (1) video-to-video frame alignment and local mosaic every second or so, (2) coarse indexing of the video mosaics in a high-altitude reference image using the geo-data, and (3) the fine geo-registration between the local video mosaics and the reference image. The time taken in the first step ranged from 30 s to 2 min for a triple of frames, each of size 320 × 240, on a Pentium 200 MHz machine [15, 16]. Steps 2 and 3 rely on the matches between the video and the reference imagery that have a large time gap, and hence have quite different appearances. The fine geo-registration requires knowledge of a reference image (geo-referenced aerial image with broader coverage) and accompanying co-registered digital elevation map (DEM). Twelve parameters are estimated by a nonlinear optimization performed in an iterative manner, requiring significant computational overhead. Bethel's research group [17] reported geo-registration results on modeling of airborne pushbroom imagery—photography with a 1D scan system [18]. The orthorectified imagery is produced by exploiting control points and linear features (semi-automatically), and exploiting GPS/INS data wherever possible. In making geo-referenced aerial mosaics, VTT's EnsoMOSAIC aerial digital imaging system [19, 20] reported 1–5 m accuracy with control points and terrain model, and 5–10 m accuracy with aerial GPS/INS observations only.

Our work falls in the category of geo-referenced mosaics, and we state our problem as follows: *Given the geo-data from our instrumentation package defining the 3D global track of the camera and range to the terrain at the center of each frame, what is a computationally efficient and fully automatic methodology for generating a seamless geo-referenced mosaic from a video sequence, in the absence of a high-altitude aerial image and an accompanying co-registered DEM*?

The solution to this quite challenging problem is enabled by a sensor package and a geo-mosaic algorithm. First, a sophisticated aerial instrumentation package the augments the video data with 3D motion, location and range data. The geographical data ("geo-data") from our aerial instrumentation package—GPS, Laser, and INS—provides information that constrains (without accumulating error) the track of the global sensor motion, and also determines the distance to the often irregular terrain surface (e.g. tree canopy). However, there are still complex problems because each of the 3D aerial sensors has its own inherent noise and error characteristics, and each sensor collects data at varying temporal rates, which leads to temporal errors, as the data must be synchronized through interpolation.

Second, we propose an efficient algorithm for using the geo-data to obtain seamless and geo-referenced strip mosaics—a two-track geo-mosaic composition method that achieves the required mosaic fidelity even if the geo-data is not accurate. The methodology also allows re-correction of the geo-mosaic when given further geo-referenced information from other sources, such as a match with geo-referenced aerial images. Moreover, no complex global optimization is used, and the algorithm is robust and fast.

A system diagram in Fig. 1 shows the overall procedures of the proposed geo-referenced mosaicing approach, which consists of the following three main steps: (1) initial image registration, (2) global transformation generation and (3) geo-referenced mosaicing. After the discussion of geo-data acquisition and the mosaic model in Sect. 2, the three steps of our geo-mosaicing approach will be described in Sects. 3–5. Experimental results with time analysis and system performance using real video data are given in Sect. 6. Discussions and conclusions are given in the final section.

2 Geo-data and mosaic models

2.1 Geographical data and geometry

The set of 3D geographical aerial data coming with the image sequences are captured with a "labtime", a common computer clock time in milliseconds that is used to synchronize the data. The set of sensor data and their recording rates are as follows: (1) *Video Image Sequences* are captured at a 30 Hz frame rate for both wide-angle video and telephoto video. (2) *A laser range profiler* gives the distance D in meters of a point laser beam from aircraft to ground (down the approximate centers of the video frames) at 238 Hz. (3) *An Inertial Navigation System (INS)*—the Watson box gyroscope provides rotation angles in degrees at 11 Hz with: *tip*—the angle between gravity and the z-axis of the aircraft in the direction of flight; *tilt*—the angle between gravity and the z-axis of the aircraft perpendicular to flight; and *heading*—the clockwise direction-of-flight angle from north. The INS provides us orientation information to an accuracy of 0.1° about the two horizontal axes, and 0.2° about the vertical axes. We use (A, B, Γ) to represent the heading, tip and tilt, and they form the rotation matrix $\mathbf{R}_w$. (4) *GPS*—a differential GPS measuring the position of the camera at 1 Hz with *altitude*—the altitude of the aircraft from sea level in meters, *A/C northing and A/C easting*—Universal Transverse Mercator (UTM) coordinates. As configured, the GPS has an absolute accuracy of approximately 1 m horizontally and 2 m vertically. We use $\mathbf{T}_w = (T_e, T_n, T_a)^t$ to denote the 3D coordinates (east, north, altitude) of the camera center in a ground coordinate system.

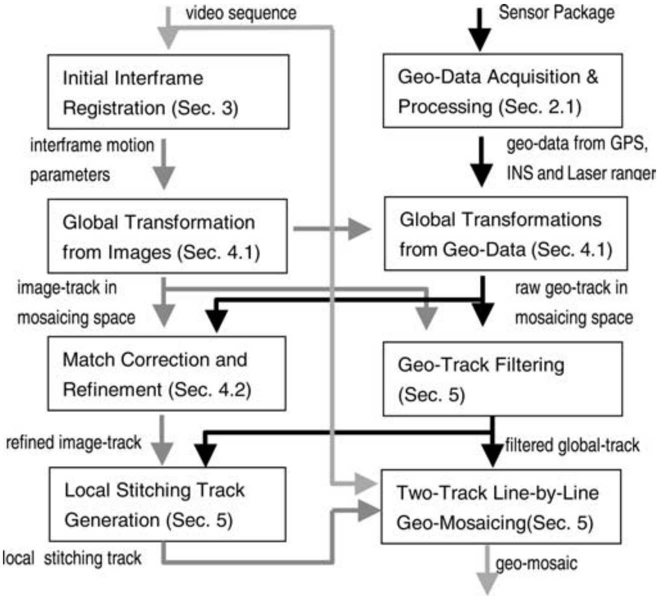

video sequence
Sensor Package
Initial Interframe Registration (Sec. 3)
Geo-Data Acquisition & Processing (Sec. 2.1)
interframe motion parameters
geo-data from GPS, INS and Laser ranger
Global Transformation from Images (Sec. 4.1)
Global Transformations from Geo-Data (Sec. 4.1)
image-track in mosaicing space
raw geo-track in mosaicing space
Match Correction and Refinement (Sec. 4.2)
Geo-Track Filtering (Sec. 5)
refined image-track
filtered global-track
Local Stitching Track Generation (Sec. 5)
Two-Track Line-by-Line Geo-Mosaicing(Sec. 5)
local stitching track
geo-mosaic

Fig. 1 Geo-mosaic system diagram. The input of each processing module is placed above the processing box and the output below the box. The data flows in the image side are represented by gray arrows and in the geo-data side by black arrows

It should be noted that the different devices, because they operating at varying rates, require us to employ linear interpolation to synchronize timing information from GPS running at 1 Hz to put all the temporal data in a common coordinate system. The relationship between camera coordinates $\mathbf{X} = (X_t, Y_t, Z_t)^t$ at time t and ground coordinates $\mathbf{X}_w = (X_w, Y_w, Z_w)^t$ can be expressed as (Fig. 2)

$$\mathbf{X}_w = \mathbf{R}_w \mathbf{X}_t + \mathbf{T}_w \tag{1}$$

where $\mathbf{R}_w$ and $\mathbf{T}_w$ have been defined in the earlier sensor description. The ground point coordinates (X_g, Y_g, Z_g) is the UTM location of a point on the ground that the laser beam has hit, and its altitude is relative to sea level. They are

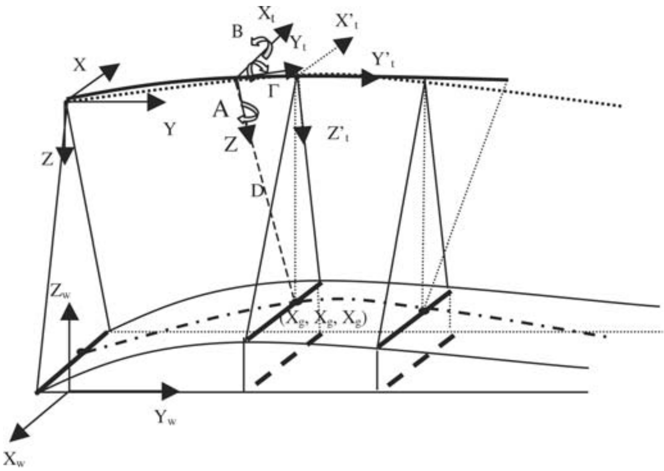


Fig. 2 Flight geometry

the assumed coordinates of the center of the video image, so we have $(X_t, Y_t, Z_t) = (0, 0, D_t)$ for this point. Therefore,

$$\begin{pmatrix} X_g^{(t)} \\ Y_g^{(t)} \\ Z_g^{(t)} \end{pmatrix} = \begin{pmatrix} -D_t \cos A \sin\Gamma + D_t \sin A \sin B \cos\Gamma + T_e \\ D_t \sin A \sin\Gamma + D_t \cos A \sin B \cos\Gamma + T_n \\ -D_t \cos B \cos\Gamma + T_a \end{pmatrix} \tag{2}$$

where super index (t) on the left-hand side of the equation and sub index t on the right-hand side mean time t (or frame t).

2.2 Inter-frame motion model

A 3D point $X_t = (X_t, Y_t, Z_t)^t$ with image coordinates (u_t, v_t) at current time t will have moved from 3D point $\mathbf{X}_{t-1} = (X_{t-1}, Y_{t-1}, Z_{t-1})^t$ with the image point (u_{t-1}, v_{t-1}) at reference time $t-1$. The relation between the 3D coordinates is

$$\mathbf{X}_{t-1} = \mathbf{R}\mathbf{X}_t + \mathbf{T}$$

where

$$\mathbf{R} = \mathbf{R}_{w,t}^{-1}\mathbf{R}_{w,t-1}, \mathbf{T} = \mathbf{R}_{w,t}^{-1}(\mathbf{T}_{\mathbf{w},t-1} - \mathbf{T}_{\mathbf{w},t}) \overset{\Delta}{=} (t_x, t_y, t_z)^t.$$

The inter-frame rotation matrix $\mathbf{R}$ has the same form as $\mathbf{R}_w$ except that (A, B, Γ) is replaced by the inter-frame rotation angles (α, β, γ). If the rotation angles are small between the successive frames, e.g., less than 5°, we can use a much simpler inter-frame motion model. Suppose the camera focal length f does not change during the motion. Using homogenous coordinates $\mathbf{u} = (u, v, 1)^t$ for an image point, under a pinhole camera model, we have

$$\mathbf{u}_{t-1} \approx \mathbf{M}_t \mathbf{u}_t \tag{3}$$

where

$$M_t = \frac{1}{s}\begin{pmatrix} \cos\alpha & -\sin\alpha & t_u \\ \sin\alpha & \cos\alpha & t_v \\ 0 & 0 & 1 \end{pmatrix} \tag{4}$$

and

$$\begin{aligned} s &= (-u_t\gamma - v_t\beta + f + f t_z/Z_t)/f \\ t_u &= f(Z_t\gamma + t_x)/Z_t \\ t_v &= f(Z_t\beta + t_y)/Z_t \end{aligned} \tag{5}$$

For the vertical tracking movement of the airborne camera (Fig. 2), involving tip (β), tilt (γ), heading (α) and range changes, we have very small β and γ. If the change in range (for the part of an image under consideration) is small relative to the range, then Eq. (3) can be treated as a 2D rigid inter-frame motion model, where $s \approx Z_{t-1}/Z_t$ could be approximated as a scale factor associated with range changes, (t_u, t_v) is the translation vector representing (tilt/X-translation, tip/Y-translation), and α is the heading change. When the inter-frame heading angle difference α is also very small, Eq. (3) can be further simplified as [21, 22]

$$\begin{cases} s \cdot u_{t-1} = u_t - v_t\alpha + t_u \\ s \cdot v_{t-1} = v_t + u_t\alpha + t_v \end{cases} \tag{6}$$

In Sect. 3, given more than two pairs of corresponding points between two frames, we can obtain the least squares solution for the motion parameters, s, t_u, t_v and α, in Eq. (6). The errors in approximation are especially small for the narrow horizontal strip (the center scan lines) in the center of each image that will be used in our image mosaic algorithm (Fig. 3). In cases where the consecutive rotation angles are large, a projective transformation model could be applied. However, the estimation of the eight parameters of the projective transform is not as robust as that of the affine transformation, hence, larger perspective distortion will be introduced with noisy data. Alternatively, we use the INS rotation measurements to pre-rectify the original video images [29]. Note that the INS measurements contain small errors (0.1–0.2°) so the heading angle is still included in our inter-frame motion model even after pre-rectification.

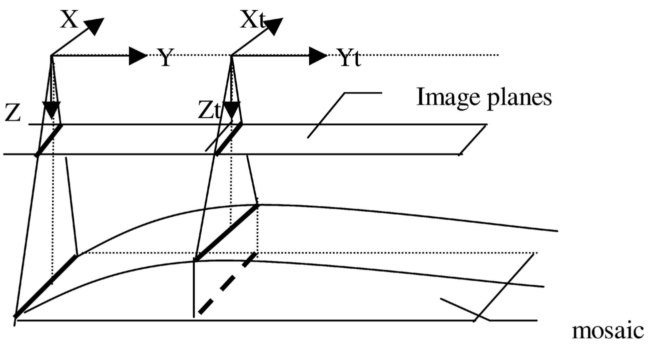


Fig. 3 Pseudo parallel-projection mosaic representation

2.3 Parallel-projection mosaic representation

A full-parallel-projection mosaic is very different from a perspective projection in the sense that distant objects do not appear smaller than nearby objects, which is ideal for our geo-based mosaic. However, we need the full 3D range map of the scene in order to construct a full-parallel-projection mosaic, and most of all, we need to match every image with the 3D model. So our question is: can we obtain the 2D geo-referenced mosaic in a much more efficient manner? Without loss of generality, we use the first frame coordinates (*X*,*Y*,*Z*) as the mosaic coordinate system (time $t = 0$) and assume that $(T_e, T_n, T_a)|_{t=0} = (0, 0, H)$, $(A, B, \Gamma)|_{t=0} = (0, 0, 0)$, and the range from the camera nodal point at time $t = 0$ to the ground is D_0. With a full-parallel-projection model, a mosaic can be represented as

$$\begin{pmatrix} wu_p \\ wv_p \\ w \end{pmatrix} = \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ D_0 \end{pmatrix} \tag{7}$$

If we assume that at all times t, we have 1D translational motion, i.e., $(T_e, T_n, T_a) = (0, T_n, H)$ and $(A, B, \Gamma) = (0, 0, 0)$, and the scene has constant depth in the *X*-direction in the camera's field of view, which is approximately true for the wide-angle video, then a *pseudo* parallel-projection mosaic point (u_p, v_p) can be constructed from perspective image point (u_t, v_t) at time t as

$$(u_p, v_p) = \frac{D(v_t)}{D_0}(u_t, v_t) + (0, v_{\mathrm{p}t}) \tag{8}$$

where $(0, v_{\mathrm{p}t})$ is the projection of center of the perspective image at time t in the mosaic image, and $D(v_t)$ is the range in the track connecting the image centers of frame $t-1$, t and $t+1$. Recall that range information is available along the motion track of the camera center by our instrumentation. The *pseudo* parallel-projection mosaic is an approximation of a full-parallel-projection mosaic. However, it is much easier to construct. In cases where the depths vary significantly in the *X*-direction (e.g. with the telephoto video), a ray interpolation approach we have proposed for stereo mosaicing [29] can be applied here to create parallel-perspective mosaics. The parallel-perspective mosaic representation is similar to the linear pushbroom imaging representation in [18].

We also need to generalize the earlier mosaic representation to the real motion model of the airborne-mounted camera when the motion has 6 degrees of freedom (Fig. 2, Eq. (1)). During forward motion, we assume that the camera's tip and tilt do not change very much during a long flight, i.e. the plane does not "accumulate" large tip and tilt, B and Γ. However, the heading angle A can change significantly over a long flight. A 2D rigid motion model can be derived from Eqs. (1) and (2) in a manner similar to Eq. (3). Let $\mathbf{u} = (u, v, 1)^t$ be the coordinates in the mosaic coordinate system (i.e. frame 0), and $\mathbf{u}_t = (u_t, v_t, 1)^t$ in the current frame t, we have

$$\mathbf{u} = \mathbf{P}_t \mathbf{u}_t \tag{9}$$

where

$$P_t = \begin{pmatrix} S\cos A & -S\sin A & T_u \\ S\sin A & S\cos A & T_v \\ 0 & 0 & 1 \end{pmatrix} \tag{10}$$

and

$$S \approx D_t/D_0, \quad T_u \approx fT_x/D_0, \quad T_v \approx fT_y/D_0 \tag{11}$$

In Eq. (11), (T_x, T_y) is the X and Y coordinates in the reference camera coordinate system of the ground point (X_g, Y_g) in Eq. (2), and we have $(T_x, T_y) = (-X_g, Y_g)$ in our coordinate system definition. Equations 9–11 imply that the mosaiced image obeys *parallel* projection in Eq. (7) and the original camera images can be approximately modeled by a *weak-perspective* projection, i.e., $(u_t, v_t) = (fX/D_t, fY/D_t)$, where D_t is the average depth of the portion of an image in time t that will be used in the mosaic. In other words, the original image is approximated by a weak-perspective projection of a "virtual camera" $X'_g Y'_g Z'_g$ with nodal point at $(X_g^{(t)}, Y_g^{(t)}, Z_g^{(t)} + D_t)$ (see Fig. 2).

Note that neither the 3D meta-data nor the registration process alone is sufficient to generate a seamless and geographically corrected mosaic. Using frame-by-frame image registration alone, we can achieve a seamless mosaic, but it will not exhibit geographical accuracy due to frame-to-frame error accumulation, even if the errors between two successive frames are very small. These errors stem from model approximation, scene complexity, and image registration errors. On the other hand, the 3D geographical data (from GPS, laser ranges and INS) provides a globally correct track of the motion without error propagation. However, the inherent (absolute) errors in the instrumentation are large, and how to match the 3D data with the 2D image is still a problem. The following sections describe an effective method to combine two different sources of data to achieve the seamless and geo-referenced mosaic, without 3D reconstruction or complex nonlinear global registration.

3 Initial registration

The inter-frame image displacements are estimated by using a pyramid-based matching algorithm [21, 22]. The hierarchical algorithm consists of four steps:

Step 1 Generate the pyramids for the current and the reference (preceding) images. For computational efficiency, the final image displacements are only given for non-overlapping image blocks of a given size, say 16×16, in the finest layer (i.e. original image) of the reference frame. The matching process is carried out from coarse to fine resolution layers. The list of the blocks is represented by their center coordinates (u_i, v_i), $i = 0, \ldots, B-1$ in the reference frame.

Step 2 Determine the image displacements. For each block in a layer of the reference frame, the absolute difference operation (a simple version of correlation) is carried out. After the first layer, the search window is reduced to only nine searches (i.e., ±1 pixel shifts in both the x- and the y-directions) in the finer layers, thanks to the use of pyramids. The motion vectors for these blocks are presented by $(\Delta u_i, \Delta v_i), i = 0, \ldots, B-1$.

Step 3 Evaluate each match by combining a texture measure with the correlation measurement. This step is important because the confidence values will serve as weights in the parameter estimation. The evaluation of the matching (as the "correlation" measurement) is calculated from the normalized absolute difference of each block as

$$d_i = 1.0 - \frac{1}{255 N_w} \sum_{(u,v)\in W(u_i,v_i)} \times \left| I(u,v) - I'(u+\Delta u_i, v+\Delta v_i) \right|$$

where $W(u_i, v_i)$ is the block centered at (u_i, v_i), N_w is the pixel number in the block, $I(\cdot)$ and $I'(\cdot)$ are the intensity values (0–255) in the reference and current frames, respectively. The texture is measured as the normalized average magnitude of the gradient image of the reference frame inside a given block i

$$g_i = \frac{1}{g_{\max} N_w} \sum_{(u,v)\in W(u_i,v_i)} \left| \left(\frac{\partial I(u,v)}{\partial u}, \frac{\partial I(u,v)}{\partial v} \right) \right|$$

where $g_{\max}$ is the maximum value of average magnitudes of all the blocks. The initial weight for the ith match is computed as

$$w_i^{(0)} = \frac{1 - \mathrm{e}^{\kappa d_i g_i}}{1 - \mathrm{e}^{-\kappa}} \tag{12}$$

where $\kappa = 8.0$ is chosen by experiments. Note that $w_i^{(0)} = 1$ iff $d_i = g_i = 1$, and $w_i^{(0)} = 0$ if $d_i = 0$ or $g_i = 0$.

Step 4 Estimate inter-frame motion parameters. We use a weighted least mean square method to iteratively estimate the inter-frame motion parameters $\theta = (t_u, t_v, \alpha, s)$ in Eq. (6). The objective function is

$$J = \min \sum_i w_i^{(k)} \left(r_i^{(k)}\right)^2, \left(r_i^{(k)}\right)^2 = \left| \mathbf{u}_i - \theta^{(k)}(\mathbf{u}'_i) \right|^2 \tag{13}$$

where $u_i = (u_i, v_i)^t$, $u'{}_i = (u_i + \Delta u_i, v_i + \Delta v_i)^t$, $i = 0, \ldots, B-1$, and the weight updating function is

$$w_i^{(k+1)} = \frac{w_i^{(0)}}{1 + \left(r_i^{(k)}/\rho\right)^2}$$

where the scale factor ρ is estimated as $\rho = \mathrm{median}_i(|r_i^{(k)}|) \times 1.4826$, assuming that the residuals can be modeled as a noisy Gaussian distribution [23, 24]. It has been pointed out in [24] that a median-based estimate has excellent resistance to outliers.

4 Registration correction and refinement

4.1 Global tracks from image registration and geo-data

Before we can create geo-mosaics, we need to generate "global tracks" from both image registration and geo-data. Here we define a track as a sequence of 2D rigid motion parameters $\Theta = (\Theta^{(0)}, \Theta^{(1)}, \ldots, \Theta^{(F)})$, where parameters in $\Theta^{(t)} = (T_u^{(t)}, T_v^{(t)}, A^{(t)}, S^{(t)})$ are defined in Eqs. (9)–(11), and F is the frame number. As in Sect. 2, select the first frame as the reference frame where the mosaic coordinate system will be generated. We can find the geometric transformation between the current frame and the first frame recursively from Eqs. (3) and (9), hence, the *image track*—the estimated global transformation track from the registration of image sequence is

$$\Theta_I^{(t)} : \mathbf{P}_t = \prod_{j=0}^{t} \mathbf{M}_j = \mathbf{P}_{t-1}\mathbf{M}_t, \quad t = 1, \ldots, F; \quad \mathbf{P}_0 = \mathbf{I} \tag{14}$$

where F is the frame number. The image track length (measured in pixels) can be calculated as

$$L_I = \sum_{t=1}^{F} \left| \left(T_u^{(t)}, T_v^{(t)}\right) - \left(T_u^{(t-1)}, T_v^{(t-1)}\right) \right|$$

We can also find the "raw" *geo-track*—the geo-referenced global transformation track on the ground (measured in meters) from the geo-data, as

$$\Theta_{GR}^{(t)} = \left(-X_g^{(t)}, Y_g^{(t)}, A^{(t)}, D^{(t)}/D^{(0)} \right), \quad t = 1, \ldots, F \tag{15}$$

where $(X_g^{(t)}, Y_g^{(t)})$ is the corresponding point of the image center at frame t (Eq. (2)), $A^{(t)}$ is the heading angle, and $D^{(t)}$ is the average range of the ground points between $(X_g^{(t)}, Y_g^{(t)})$ and $(X_g^{(t-1)}, Y_g^{(t-1)})$ (see Fig. 2). The track length (in meters) on the ground can be calculated as

$$L_G = \sum_{t=1}^{F} \left| \left(X_g^{(t)}, Y_g^{(t)}\right) - \left(X_g^{(t-1)}, Y_g^{(t-1)}\right) \right| \tag{16}$$

From Eq. (11), the effective focal length can be estimated as $f = D_0 \frac{L_I}{L_G}$. Then the raw geo-track measured in pixels in the mosaicing coordinate system is

$$\Theta_D^{(t)} = \left(-f\frac{X_g^{(t)}}{D^{(0)}}, f\frac{Y_g^{(t)}}{D^{(0)}}, A^{(t)}, \frac{D^{(t)}}{D^{(0)}} \right), \quad t = 1, \ldots, F \tag{17}$$

where $D^{(t)} = D_t$. (Both of these two notations are used in the paper for convenience.) Notice the distinctly different ways that the tracks are derived. The estimated track from the image (image-track) is the composition of inter-frame transformations with accumulating error, while the geo-track is captured directly from absolute 3D geo-data, with associated inherent absolute error, but is free from error propagation.

4.2 Match correction and refinement

If the initial estimation of inter-frame motion parameters are significantly different from the results of the geo-data, and/or the *weighted* frame difference (which will be defined later) is very large, the geo-data are used to estimate the initial values of the "expected" motion parameters, and then the corresponding frames are re-matched. Note that the geographical data is used *only* for correcting the possible mis-registration between successive frames; it is also possible to use motion smoothness constraints if the geo-data are not available. Given that our goal for image registration is to create an image mosaic, the *weight* function employed for the image difference is a 1D Gaussian

$$h(u, v) = \frac{1}{\sqrt{2\pi\alpha}} \mathrm{e}^{\frac{v^2}{2\pi\sigma^2}} \tag{18}$$

which favors those points near the center scan lines of the frames used in the mosaic images (refer to Fig. 4). With the initial motion vectors of each block from the given initial inter-frame motion parameters, the match process starts from a suitable intermediate layer (in the pyramid) in which the initial displacements are detectable.

Even if no mismatch occurs, the refinement process is needed when the rotation angle α is large, since α instead of sine of α is used in motion estimation. The refinement can be performed by iteratively warping the current image and re-matching the warped image with the reference image. We emphasize that $\mathbf{M}_t$ in Eq. (3) is used to warp the image, even if we still use linear Eq. (6) to estimate the motion parameters $\theta^{(m)} = (t_u, t_v, \alpha, s)|_m$, where (m) denotes the iteration count, so that errors will be reduced with decreasing residual rotating angles.

5 Two-track geo-mosaic composition

We propose a two-track method to build a geo-mosaic—a mosaic that is both seamless and geo-referenced. We already have two "tracks" of motion transformation parameters: the *refined image-track* Θ_I and the raw *geo-track* from geo-data, Θ_D. They are used to calculate two final tracks that will be used to generate geo-reference mosaics: a "filtered" *global-track* ("global-corrected track") Θ_G that matches the global geographical track, and an "updated" *local track* ("local-stitching track") Θ_L that guarantees precise local stitching of the mosaic.

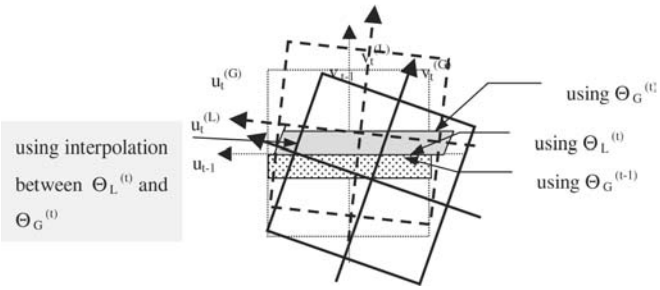


Fig. 4 Two-track line-by-line geo-mosaic geometry

The filtered global-track Θ_G is generated from the raw geo-track Θ_D by using a simplified version of the extended Kalman filtering approach [25, 26]. The basic idea is to use each inter-frame image matching result as a prediction to the inter-frame pose estimation in order to reduce the relatively large absolute errors in the geo-track. However, we will show in our experiments that even with data filtering, directly using the filtered geo-track for video mosaicing usually brings geometric seams (mis-alignments) in the mosaicing results, since the filtered global track does not account for the local perspective distortions between two successive frames for 3D scenes. Therefore, we also calculate a second track for video mosaicing—the updated "local track". The local-track Θ_L is calculated as follows:

$$\Theta_L^{(t)} : \mathbf{P}_L^{(t)} = \mathbf{P}_G^{(t-1)} \mathbf{M}_I^{(t)}, \quad t = 1, \ldots \tag{19}$$

where $\Theta_L^{(t)}$ and $\mathbf{P}_L^{(t)}$ are the parameters and the matrix of the stitching transformation at frame t; $\mathbf{M}_{\mathrm{I}}^{(t)}$ is the inter-frame transformation matrix from image registration of frame t and $t-1$; and $\mathbf{P}_G^{(t-1)}$ (from $\Theta_G^{(t-1)}$) is the geo-transformation matrix of frame $t-1$. Notice that we combine the inter-frame image transformation $M_I^{(t)}$ with the previous geo-referenced global transformation $P_G^{(t-1)}$ in time $t-1$ (instead of the previous local-stitching transformation $P_{\mathrm{L}}^{(t-1)}$ in time $t-1$). This leads to the two-track line-by-line geo-mosaic algorithm (Fig. 4):

Step 1. From frame t, suppose we warp $N+1$ scan lines into the mosaic, where N is determined by the geo-data. These scan lines are expressed in the mosaic image, i.e., we use the inverse transform that maps from the mosaic to the original image frames.

Step 2. The transformation for the ith scan line is estimated as the linear interpolation of each parameters between $\Theta_L^{(t)}$ and $\Theta_G^{(t)}$

$$\Theta_i^{(t)} = \frac{(i-N)\Theta_L^{(t)} + i\Theta_G^{(t)}}{N}, \quad i = 0, 1, \ldots, N, \tag{20}$$

Step 3. The mosaic process is to transform *a line* in frame t and paste it to the ith scan line in the mosaic ($i = 0, 1, \ldots, N$). The first scan line from frame t will be precisely stitched to the last scan line from frame $t-1$, since the transformation between them is just the inter-frame image transformation $\mathbf{M}_I^{(t)}$; while the last scan line from frame t satisfies the geometrical transformation from the global track constraint, $\mathbf{P}_G^{(t)}$.

Figure 4 shows the geometry of line-by-line mosaic. More suitable interpolation methods can be used by using the ranges data from the range profiler across image rows in the center column (along the line $v = 0$). For the real geographical image mosaic, the difference between the inter-frame transformations from image registration and the filtered geo-data is small, so the line-by-line transformations compensate for the original distortion due to 3D geometry and 2D perspective projection of 3D scenes, and bias due to error propagation, rather than bringing additional distortion to the geo-mosaic. The two-track algorithm is computationally fast, since only linear transformations and image warping operations are needed in the mosaicing process.

6 Experimental analysis

We will discuss two aspects of our experimental system for geo-reference mosaics. First, we will provide some implementation details and do a time analysis of the system. Second, we will show the accuracy of the geo-mosaics through some real examples.

6.1 Time analysis

There are three main components in the system (Fig. 1): video registration for generating the local track, geo-data processing for generating the geo-track, and two-track video mosaicing. Since the geo-data processing only deals with a few motion parameters, the processing time needed is trivial. Most of the processing time is spent on video registration step, and some on video mosaicing step.

First, we will do a time complexity analysis of the video registration and the video mosaicing algorithms, respectively. The time complexity of the video registration algorithm will be measured between two frames, and the time performance of the video registration will be measured as frame rate (frames per second—fps). The time complexity of the video mosaicing algorithm will be measured against the size of the final mosaic, or the coverage of the mosaic.

The basic video registration algorithm (Sect. 3) uses a pyramid-based correlation method. Since we only need to obtain inter-frame motion model, we only perform matches for non-overlapping image blocks. In addition, we use an addressing look-up table (LUT) for indexing the square window of the correlation, and the correlation is calculated as sum of absolute differences, so there are no multiplications in the main body of the correlation. Let us assume the image size is $M \times N$, and the matching block size is $b \times b$. Therefore, the number of blocks is $B = MN/b^2$. One correlation calculation needs to do b^2 operations, which are mainly additions and subtraction. We only need to perform nine searches for each matching block due to the nature of the pyramid searches. As such, the total numbers of operations is $9Bb^2 = 9\ (MN/b^2)\ b^2 = 9MN$. For the entire match process using pyramid-based approach, the total operation is $9MN \log_2(9MN)$, which in Big-O form is $O(MN \log MN)$ for the registration algorithm.

The two-track geo-mosaicing algorithm is based on a line-to-line linear transformation. Essentially, for each scan line in the mosaic (perpendicular to the mosaicing direction v, see Fig. 4), we perform an affine transformation defined in Eq. (3). The following treatments have made this algorithm linear in the size of the mosaic. (1) We use an *inverse transformation technique* to make the iterative dense warping possible. That is, given mosaicing coordinates, we calculate the image coordinates in an original video frame using Eq. (3), with parameters calculated in Eq. (20). Since only one coordinate (i.e., u) of the mosaic changes along the scan line, we have a 1D iterative procedure when u increases in each step by 1. (2) We use *LUT techniques* to simplify Eq. (3) into a few operations of additions and subtractions. (3) We use a *scaling technique* to make integer operations possible in both coordinate transformations and bilinear interpolation. Before coordinate transformation, we scale up the mosaic coordinates by 1024 (via a left-shift operation), which means that an integer can represent a coordinate to one of 1024 accuracy. Then we scale the frame coordinates down by 1024 (via a right-shift operation), after the coordinate transformation. By using inverse transformation, integer scaling and LUT techniques, the coordinate transformations (with LUTs) and the bilinear interpolations are all in integers, with mainly indexing, addition and subtraction operations, which greatly simplify the computation. Assume that the mosaiced image size is $P \times Q$, then time complexity of the two-track mosaicing algorithm is $O(P \times Q)$, with mainly additions and subtraction.

For time estimation in real conditions, we processed a long 955-frame forest image sequence with 3-band color images, each with 360 × 240 pixels. The average inter-frame translation is about 2.5 pixels in the motion direction. Note that the magnitudes of the inter-frame motion does not affect the time spent on inter-frame matching, since our pyramid-based video registration algorithm starts searching matches with displacements up to half of the image size. We have tested our experimental system in two PCs. The first machine is an IBM T30 notebook, with a Pentium IV 2.0 GHz CPU and 512 MB RAM. The second machine is a Dell Precision workstation, with 2.4 GHz Dual Xeon CPUs, 512 K Cache and 1 GB RAM at 266 MHz rate. Table 1 lists the time estimates in the registration step (in seconds, for all 955 frames) and the frame rate for registration (*frames per second* – fps), the time estimates in the mosaicing step (in seconds, for a 2448 × 336 mosaic), and the mosaicing rate (*scan line per second* – lps), in IBM and Dell machines respectively. The time estimation includes video display and

Table 1 Time analysis for registration and mosaicing (results for creating a 2448 × 336 mosaic from a 955-frame video of 360 × 240 color images)

Measures/ machines	Registration (s)	Frame rate (fps)	Mosaicing (s)	Line rate (lps)
IBM notebook	83.75	**11.4**	5	490
Dell workstation	42.5	**22.5**	3	816

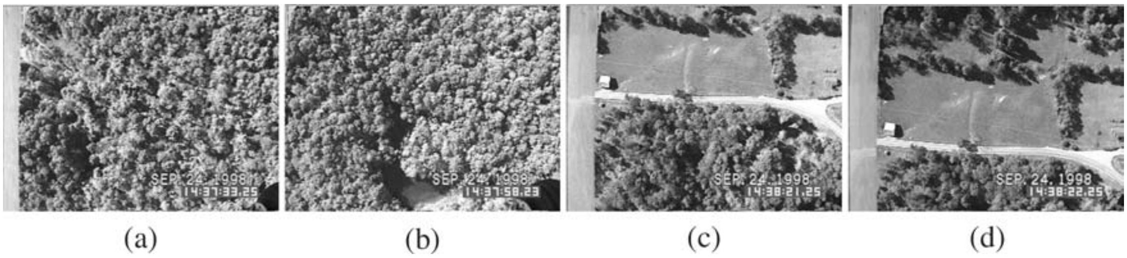


Fig. 5 Sampled frames from the forestry video sequence

image loading. The registration frame rate achieves 11.4 fps in the IBM notebook and 22.5 fps in the dual-CPUs Dell workstation, close to real-time performance.

6.2 Mosaicing result analysis

We give experimental results with real forestry video images over The Nature Conservancy (TNC) test site in Ohio, and the Amazon rain forest in Brazil. Figure 5 shows four frames of a 53-frame sampled video sequence taken by the wide-angle camera over TNC test site for which a full set of geo-data are available. The original image sequence is sub-sampled to 1 fps with image size 360 × 240 in our experiment (see online video at our web site [27]). Recalling the data rates for GPS/INS/Laser data, every digitized frame is linearly interpolated to correspond to a GPS location and INS rotation angles for the camera. Between the centers of consecutive frames there are about 238 range samples. It should be noted that there are obvious illumination changes due to auto iris effects (see Fig. 5c and d). The inter-frame translation along the Y-axis is about 60–70 pixels, which is more than 1/4 of the image height.

By using the two-track method described in Sect. 5, a seamless, geo-referenced mosaic is created (Fig. 6b; see online image at [27]). The (translation components of) the geo-track and the image-track are superimposed in the geo-referenced mosaic in red and white respectively. The translation components of the two tracks are found to be very close to each other except for certain locations. Figure 7a and b show the comparisons of the headings and scales of the two tracks, respectively. The global trends of the headings are similar, but the scales are quite different, which is obvious by comparing the mosaics in Fig. 6a and b. The expected scales are calculated from the absolute geo-data, but the estimated scales are accumulated from inter-frame motion parameters. Although the estimated inter-frame scales are within 0.998–1.012 (Fig. 8), which is quite close to the real situation, the accumulating errors are as large as 30% by the end of the 53-frame sequence. Note that we will be doing sequences of many minutes to hours in the future.

The two-track method corrects this accumulating error frame-by-frame while maintaining precise stitching of successive frames. To show the role of the local-track for seamless stitching, we compare our geo-referenced mosaic to a *geo-only* mosaic where only the global transformation (from the geo-data only) is applied. Figure 9 is a comparison of a sub-image of the same portion of the geo-referenced mosaics and geo-only mosaic. It is obvious that the geo-only mosaic is not seamless even though the global track is

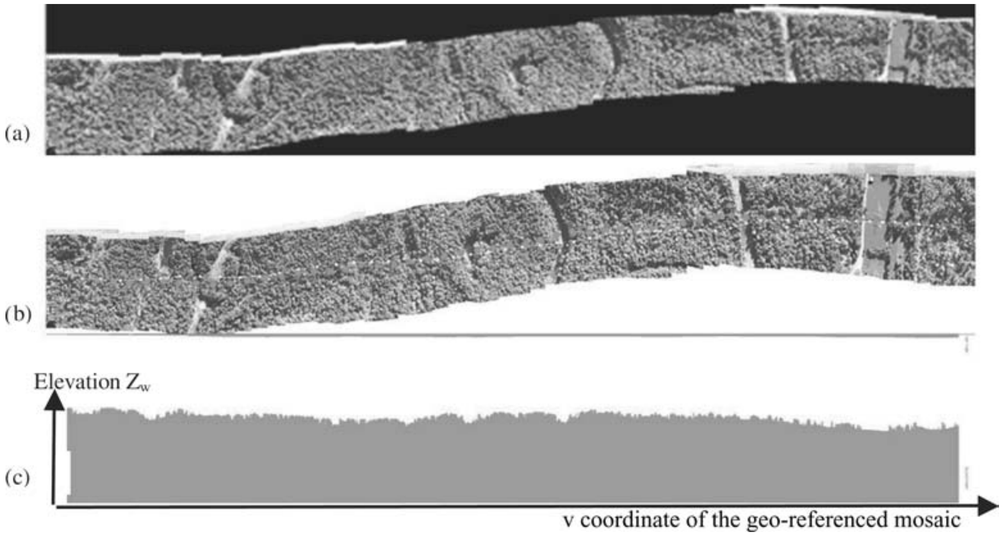


Fig. 6 Mosaicing comparison, **a** free mosaic, **b** geo-referenced mosaic, and **c** the range histogram from the range profiler

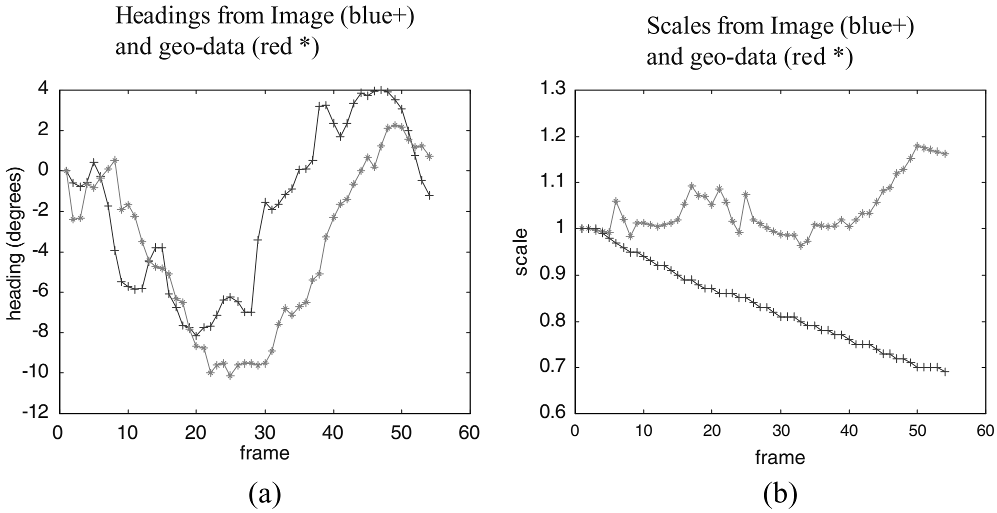


Fig. 7 Headings and scales in the two tracks

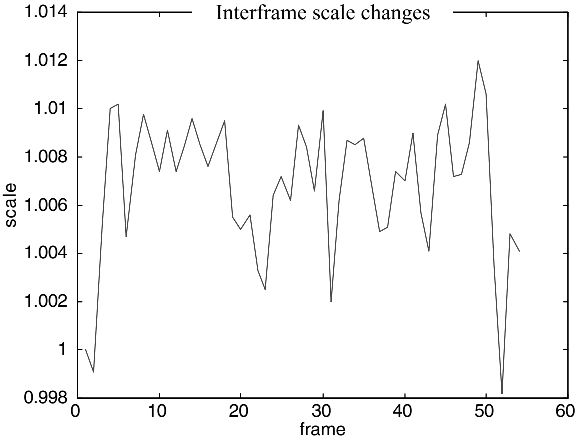


Fig. 8 Inter-frame scale changes

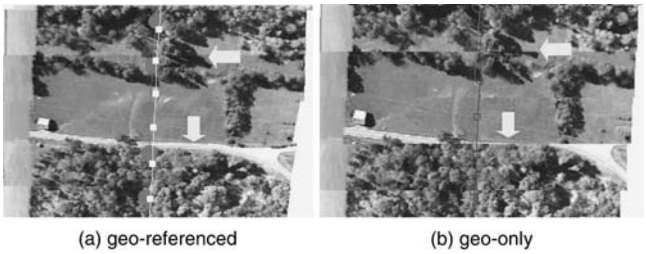


Fig. 9 Zoom-in comparison of geo-referenced and geo-only mosaic. Please compare two places with arrows (a road and a tree)

faithful to the geographical data, which is not accurate enough for a seamless mosaic. The geo-referenced mosaic satisfies both of the requirements.

For comparison, we also generated a free mosaic using the evaluation version of a commercial software VideoBrush 2.0. From their published papers [4, 15] related to this system and the mosaic results, we find that no scaling is applied to the mosaic and the track is smoothed. While the mosaic is seamless, and color/luminance blending is handled, it is not geo-referenced; for example the scale is not changed as a function of ranges of ground points. Figure 10 is a comparison of a sub-image of the same portion of the geo-referenced mosaics and VideoBrush mosaic superimposed on the high-altitude image that was taken at 5000 ft (1524 m) above the ground at about the same time as the video sequence. Image points along the center track in our geo-mosaic register precisely with the high-altitude image, and at the border of the mosaic there are only small errors (due to the assumption of the constant range in the *X*-direction—a discussion and extension is given in the next subsection). As expected, the VideoBrush mosaic cannot register with the high-altitude image (Fig. 10b). It should be noted here that two feature points are selected in the head and tail of both of the mosaics; one of them is under the white circle (O) in each of the image in Fig. 10. Notice the obvious different location errors of a white building (pointed by an arrow) below the road in the right of each image. The reason for large offset in VideoBrush's mosaic is that it does not change the scales with the change of the terrain ranges, which is obvious in this part of the scene.

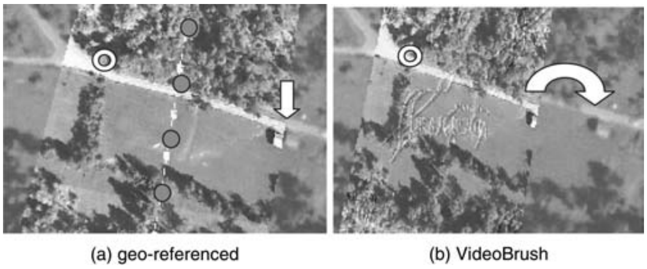


Fig. 10 Zoom-in comparison of geo-referenced and VideoBrush mosaic overlaid on the high-altitude ortho-image

In the geo-referenced mosaic of Figs. 6b and 10a, matching of the 2D image with the 3D geo-data is also shown.

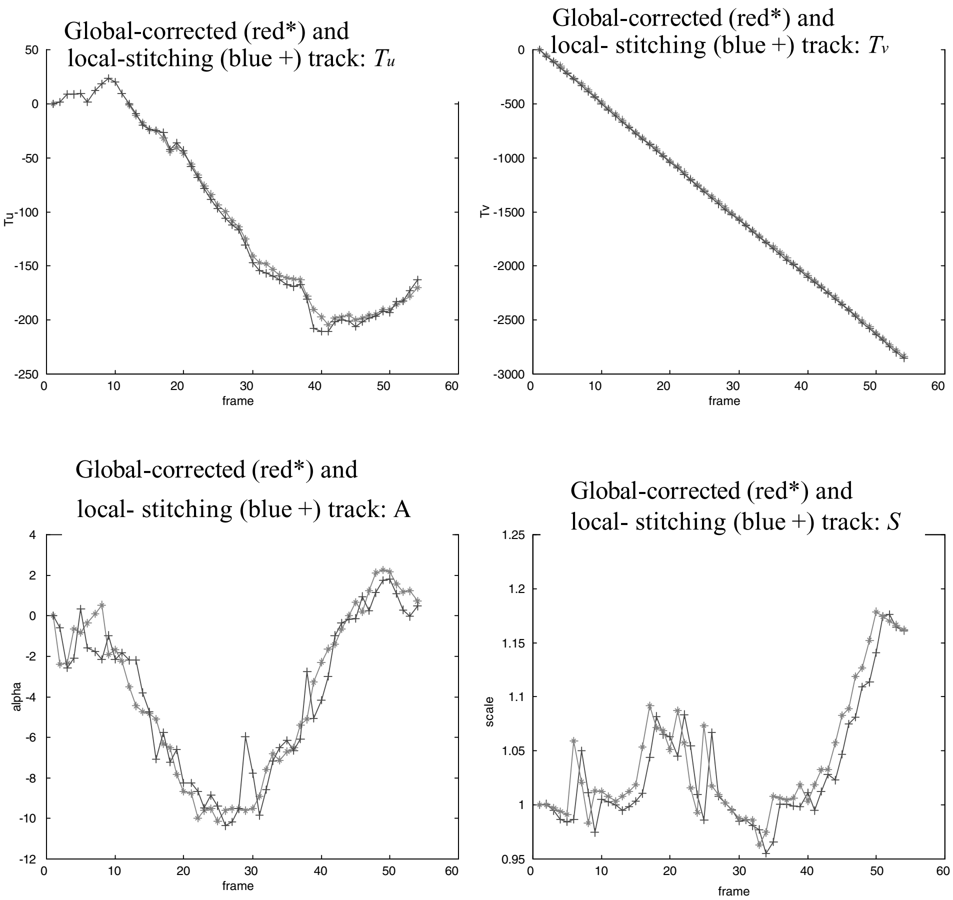


Fig. 11 Two tracks: global corrected track and local stitching track

Each red circle and the attached number indicate the center of each frame and the ground altitude in meters of that point. The red circle of 5-m radius represents the error in pixels corresponding to a 5 m location error on the ground. The recovered altitudes of the flight and the altitudes of the ground along the track are shown beside the geo-mosaic as histograms in Fig. 6c. The geo-referenced mosaic image matches quite well with the geo-data, for example, the roads and the grassland in the mosaic image. By comparing the size of the red circles of 5-m radius along the center of the mosaic, and the alignment of a building on the right edge of the mosaic (pointed by an arrow), we found that the registration error in this example is less than 5 m in the vertical direction (the mosaicing direction) and 5–10 m in the horizontal direction. Figure 11 shows the four components of the two tracks: the global-corrected tracks and local-stitching tracks. They lay *close* to each other, but *differences* exist. That is why we use both of the tracks in order to generate geo-referenced and seamless mosaics.

Another noteworthy recent success was the collection of 130 h of digital video consisting of 10 TB of uncompressed imagery over the Amazon rain forest. This was carried out in collaboration with the Smithsonian Institute, the University of California at Santa Barbara (UCSB) and I.N.P.E. (an arm of the Brazilian Space Agency). It was the largest ever aerial video data collection over the Amazon Basin. In the data collection, two cameras with about 1:10 ratio focal lengths were mounted side by side in a small airplane to capture both the high resolution and the wide FOV video sequences of the forest scene in an economic way.

Figure 12 shows full base map of the Smithsonian site made from eight overlapping geo-referenced mosaic strips from the wide-angle camera. The map covers an area of approximately 50×10 km ($25{,}113 \times 9040$ pixels, with a ground sampling distance (GSD) of 2 m. The eight separate mosaics from eight video sequences are aligned nicely side by side, indicating that the pseudo-parallel-projection mosaicing model does a reasonably good job. The close-up window of a small area shows the alignments between mosaic strips. Three strips pass this window, which can be found by the unbalanced colors between strips. The curved edges between two strips were generated interactively using an image tool.

It is challenge for us to evaluate the accuracy of the geo-mosaicing results because this area does not have available ground control points. However, we have tried to obtain some quantitative estimation using our available sensor package. By measuring the absolute coordinates of some ground landmarks (a few trees and towers) using our GPS on the ground, we compare the geo-locations of the same

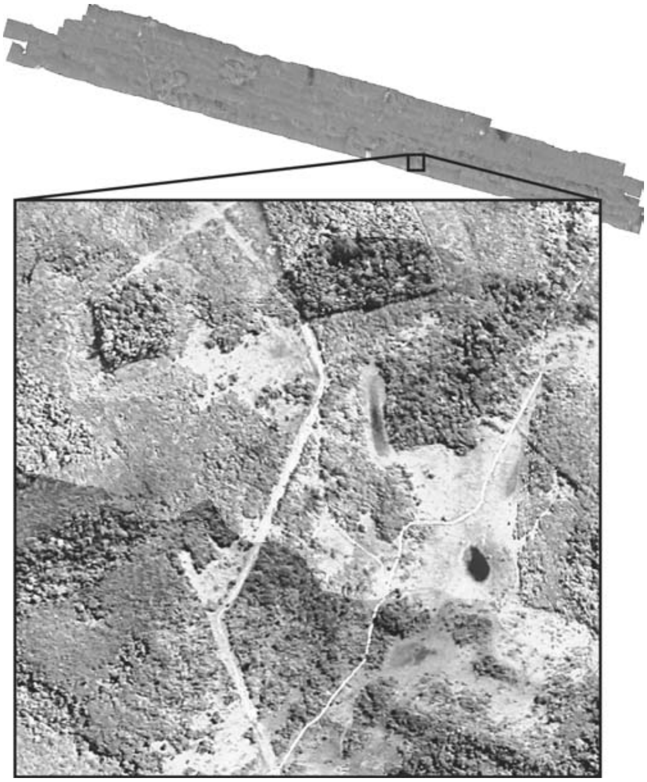


Fig. 12 The full base map of the Smithsonian site made from eight overlapping geo-registered mosaic strips, covering an area of approximately 50 × 10 km (25,113 × 9040 pixels, with a ground sampling distance (GSD) of 2 m. A close-up window with three mosaic strips running through is also shown

landmarks on the geo-referenced mosaics. We found that the geometric accuracy of the mosaics is 5–10 m, which is comparable to the 2-m ground resolution of the wide-angle camera. This accuracy can also be visually verified from the previous experiment (Fig. 10a). The errors are mainly from two sources—sensor package and video registration/mosaicing process. First data sources of our geo-mosaics have errors: GPS 1–2 m, INS 0.1°–0.2° and video images 2 m/pixel. Second, the errors of processing come from the approximate modeling for image registration, and the assumption of uniform depths along the X-direction. More quantitative analysis needs to be done for the accuracy of the mosaiced results, but we will leave this as future work, since currently we do not have sufficient "ground truth" data.

7 Conclusions and discussions

A new method of creating a seamless and geo-referenced video mosaic has been presented. By analyzing the motion model of the flight, a pseudo-parallel-projection mosaic representation is developed to represent the geo-referenced mosaic given the available geographical data. A complete geo-mosaic prototype system, including local registration, track generation, matching refinement and two-track-based mosaic composition are provided. The advantages of this approach are that sensor motion information is effectively employed in a simple model to produce effective results, and a fast, robust and practical implementation is achieved. The accuracy of our mosaiced results is 5–10 m with video images of 2-m ground resolution. The video alignment step, which takes more than 90% of the total processing time of our experimental geo-mosaic system, works at 11.4 fps for 360 × 240 color images in a Pentium IV 2.0 GHz IBM notebook, and 22.5 fps for in a Xeon 2.4 GHz dual-CPU Dell workstation. We have demonstrated the significant potential of the emerging digital video technology (with geo-referenced stereo mosaics as an important component) through a set of initial collaborative projects with environmental science partners, in regions of New England, Bolivia, Brazil, and Madagascar.

Comparing with other geo-registration methods [15, 17, 19, 20], our approach has three distinctive features. (1) Large geo-referenced video mosaic from a long image sequence can be generated before the match of video and reference imagery. This makes it easy to further register overlapped video frames (or a video mosaic) with the reference image. (2) Only geo-data from GPS/INS/Laser are used to generate a geo-mosaic, without the need of a geo-referenced image and the accompanying DEM. Notice that in addition to the computational burden and difficulties in matching two different kinds of images in their methods, the error in 3D DEM may distort the video mosaic such that seamless-ness may not be guaranteed. (3) Our method is fast and efficient. This is due to the different models, methodologies and goals. We did not try to recover the full 3D motion parameters of the moving cameras, for example, by using the computationally expensive bundle adjustments [28]; neither did we try to reconstruct the point-wise 3D structure of the scene frame-by-frame. Our goal is to make 2D geo-referenced and seamless mosaics, without 3D reconstruction, and in near real-time.

However, we should point out that there are limitations in our current implementation. One potential weakness of our current work is the assumption of constant range along the x-axis, even if this is reasonable when dealing with the wide FOV videos (e.g., in Fig. 12, eight mosaic strips were successfully aligned). The simplified model we use is due to the availability of range along the optical axis and hence along the center line of the flight path. As a result, image points along the center track in our geo-mosaic register precisely with the high-altitude image, but at the border of the mosaic there are errors especially for the telephoto mosaics. This can be improved by the following extensions.

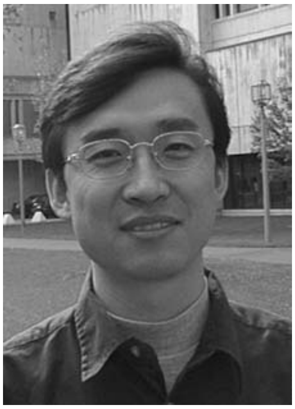
- *Generalization of the geo-referenced mosaic method.* If a DEM is available or motion parallax can be reliably applied, more complicated model, e.g. projective transformation model can be utilized between two video frames. Note that the way a geo-mosaic is created—one strip from each frame, and one transform per scan line, so a projective transformation would model the depth change along the x-axis well without dramatically increasing the computational burden if pointwise 3D data is a applied. A geo-referenced DEM can be used to generate the corresponding "projective track", and the same two-track method can be applied except that the transformations between scan lines are changed to projective transformation. In the absence of an available DEM, motion parallax can be explored – though with some difficulties. Recently, we have proposed a ray interpolation approach following this idea in making parallel-perspective stereo mosaics [29].

- *Registration of aerial images and video mosaics*. With a geo-reference aerial imagery available, the registration can be carried out to reduce the error from the model simplification, without the need of a accompanying DEM. After a few reliable matches along the boundary of the video mosaic are established, the same technique of line-by-line transformations can be used before or after the generation of the video mosaic. The geo-registration between the aerial image and a few video images provides a more accurate global track, while a seamless and high-quality mosaic can be guaranteed by our approach.

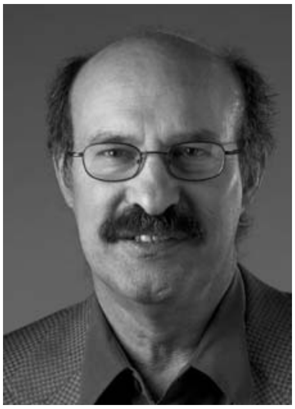
Acknowledgements This work was supported by NSF EIA-9726401, NSF CNPq EIA9970046, TNC in conjunction with the Winrock Corporation and DURIP equipment grant (DAAD19-99-1-0016). The first author is also supported by AFRL (Grant No. F33615-03-1-63-83), New York Institute for Advanced Studies, and PSC-CUNY. The authors thank Kris Denio, Chris Holmes, Chris Hayward and Dana Slaymaker at UMass-Amherst for collecting the video, processing the geographical data, generating some of the geo-referenced mosaics and evaluating the geo-referenced mosaics.

References

1. Slaymaker, D.M., Jones, K.M.L., Griffin, C.R., Finn, J.T.: Mapping deciduous forests in Southern New England using aerial videography and hyerclustered multi-temporal Landsat TM imagery. ASPRS Gap Analysis, Bethesda, Maryland (1996)
2. Chen, S.E.: QuickTime VR—An image-based approach to virtual environment navigation. Proc. SIGGRAPH **95**, 29–38 (1995)
3. Kang, S.B., Weiss, R.: Characteristics of errors in compositing panoramic images. Proc. CVPR'97, 103–109 (1997)
4. Sawhney, H.S., Kumar, R., Gendel, G., Bergen, J., Dixon, D., Paragano, V.: VideoBrush^{TM}: Experiences with consumer video mosaicing. Proc. IEEE WACV'98, 56–62 (1998)
5. Zhu, Z., Xu, G., Riseman, E.M., Hanson, A.R.: Fast generation of dynamic and multi-resolution 360° panorama from video sequences. In: Proceedings of the IEEE International Conference on Multimedia Computing and Systems. June 7–11, vol. 1, pp. 400–406 Florence, Italy, (1999)
6. Shum, H.-Y., Szeliski, R.: Construction of panoramic image mosaics with global and local alignment. Int. J. Comput. Vision. **36**(2): 101–130 (2000)
7. Sawhney, H.S., Hsu, S., Kumar, R.: Robust video mosaicing through topology inference and local to global alignment. Proc. ECCV'98 **2**, 103–119 (1998)
8. Davis, J.: Mosaics of scenes with moving objects. Proc. CVPR'98, 354–360 (1998)
9. Coorg, S.R., Teller, S.J.: Spherical mosaics with quaternions and dense correlation. Int. J. Comput. Vision. **37**(3), 259–273 (2000)
10. Mann, S., Picard, R.W.: Video orbits of the projective group; A simple approach to featureless estimation of parameters. IEEE Trans. Image. Proc. **6**(9) (1997)
11. Peleg, S., Herman, J.: Panoramic mosaics by manifold projection. Proc. CVPR'97, 338–343 (1997)
12. Zheng, J.Y., Tsuji, S.: Panoramic representation for route recognition by a mobile robot. Int. J. Comput. Vision. **9**(1), 55–76 (1992)
13. Peleg, S., Rousso, B., Rav-Akha, A., Zomet, A.: Mosaicing on adaptive manifolds. IEEE Trans. PAMI **22**(10), 1144–1154 (2000)
14. Zomet, A., Feldman, D., Peleg, S., Weinshall, D.: Mosaicing new views: The crossed-slits projection. IEEE Trans PAMI **25**(6) (2003)
15. Kumar, R., Sawhney, H., Asmuth, J., Pope, J., Hsu, S.: Registration of video to geo-referenced imagery. Proc. ICPR'98 **2**, 1393–1400 (1998)
16. Sawhney, H., Kumar, R.: True multi-image alignment and its application to mosaicing and lens distortion correction. IEEE Trans. PAMI **21**(3), 235–243 (1999)
17. Lee, C., Theiss, H., Bethel, J., Mikhail, E.: Rigorous mathematical modeling of airborne pushbroom imaging systems. Photogram. Eng. Remote Sens. **66**(4) (2000)
18. Gupta, R., Hartley, R.: Linear pushbroom cameras. IEEE Trans. PAMI **19**(9), 963–975 (1997)
19. Holm, M., Rautakorpi, S.: Experiences of automatic creation of image mosaics and digital surface models using airborne digital camera data. IS&T/SPIE Conference on Videometrics, VI, San Jose, California pp. 139–150 (1999)
20. EnsoMOSAIC Aerial Digital Imaging, http://www.storaenso.com/CDAvgn/main/0,,1_-1362–2381-,00.html
21. Zhu, Z., Xu, G., Yang, Y., Jin, J.S.: Camera stabilization based on 2.5D motion estimation and inertial motion filtering. IEEE International Conference on Intelligent Vehicles, Stuttgart, Germany, 28–30 (1998)
22. Jin, J.S., Zhu, Z., Xu, G.: A stable vision system for moving vehicles. IEEE Trans. Intell. Transport. Syst. **1**(1), 32–39 (2000)
23. Rousseeuw, P.J., Leroy, A.M.: Robust Regression and Outlier Detection. Wiley, New York (1987)
24. Sawhney, H.S., Ayer, S.: Compact representation of videos through dominant and multiple motion estimation. IEEE Trans. PAMI **18**(8), 814–830 (1996)
25. Gelb, A.: Applied Optimal Estimation. MIT Press, Cambridge, MA (1974)
26. Broida, T.J., Chellappa, R.: Estimating the kinematics and structure of a rigid object from a sequence of monocular images. IEEE Trans. PAMI **13**(6), 497–513 (1991)
27. Zhu, Z.: Geo-Mosaic for environmental monitoring, http://www.cs.umass.edu/∼zhu/ geomosaic.html or http://www-cs.engr.ccny.cuny.edu/∼zhu/geomosaic.html (2004)
28. Slama, C.C. (ed.): Manual of Photogrammetry, 4th edn. American Society of Photogrammetry (1980)
29. Zhu, Z., Riseman, E.M., Hanson, A.R.: Generalized parallel-perspective stereo mosaics from airborne videos. IEEE Trans. PAMI **26**(2), 226–237 (2004)



Zhigang Zhu received his B.E., M.E. and Ph.D. degrees, all in computer science from Tsinghua University, Beijing, in 1988, 1991 and 1997, respectively. He is currently an associate professor in the Department of Computer Science, the City College of the City University of New York. Previously, he was an associate professor at Tsinghua University, and a senior research fellow at the University of Massachusetts, Amherst. His research interests include 3D computer vision, HCI, virtual/augmented reality, video representation, and various applications in education, environment, robotics, surveillance and transportation. He has published over 90 technical papers in the related fields. He is a member of IEEE and ACM.



Edward M. Riseman received his B.S. degree from Clarkson College of Technology in 1964 and his M.S. and Ph.D. degrees in electrical engineering from Cornell University in 1966 and 1969, respectively. He joined the Computer Science Department at UMass-Amherst as assistant professor in 1969, has been a professor since 1978, and served as chairman of the department from 1981 to 1985. Professor Riseman has conducted research in computer vision, artificial intelligence, learning, and pattern recognition, and has more than 200 publications. He has co-directed the Computer Vision Laboratory since its inception in 1975. Professor Riseman has been on the editorial boards of Computer Vision and Image Understanding (CVIU) from 1992 to 1997 and of the *International Journal of Computer Vision* (*IJCV*) from 1987 to the present. He is a senior member of IEEE, and a fellow of AAAI.

Allen R. Hanson received his B.S. degree from Clarkson College of Technology in 1964 and his M.S. and Ph.D. degrees in electrical engineering from Cornell University in 1966 and 1969, respectively. He joined the Computer Science Department at UMass-Amherst as an associate professor in 1981, and has been a professor there since 1989. Professor Hanson has conducted research in computer vision, artificial intelligence, learning, and pattern recognition, and has more than 150 publications. He is co-director of the Computer Vision Laboratory at UMass-Amherst, and has been on the editorial boards of the following journals: *Computer Vision*, *Graphics and Image Processing* 1983–1990, *Computer Vision*, *Graphics, and Image Processing—Image Understanding* 1991–1994, and *Computer Vision and Image Understanding* 1995–present.

Howard Schultz received a M.S. degree in physics from UCLA in 1974 and a Ph.D. in physical oceanography from the University of Michigan in 1982. Currently, he is a senior research fellow with the Computer Science Department at the University of Massachusetts, Amherst. His research interests include quantitative methods for image understanding and remote sensing. The current focus of his research activities are on developing automatic techniques for generating complex, 3D models from sequences of images. This research has found application in a variety of programs including real-time terrain modeling and video aided navigation. He is a member of the IEEE, the American Geophysical Union, and the American Society of Photogrammetry and Remote Sensing.