Real-time construction and visualisation of drift-free video mosaics from unconstrained camera motion

Mateusz Brzeszcz[1,2], *Toby P. Breckon*[3]

[1]*School of Engineering, Cranfield University, Cranfield, UK*
[2]*Automatic Control, Electronics and Computer Science Department, Silesian University of Technology, Gliwice, Poland*
[3]*School of Engineering and Computing Sciences, Durham University, Durham, UK*
E-mail: toby.breckon@durham.ac.uk

Published in *The Journal of Engineering*; Received on 2nd February 2015; Accepted on 18th June 2015

Abstract: This work proposes a novel approach for real-time video mosaicking facilitating drift-free mosaic construction and visualisation, with integrated frame blending and redundancy management, that is shown to be flexible to a range of varying mosaic scenarios. The approach supports unconstrained camera motion with in-sequence loop closing, variation in camera focal distance (zoom) and recovery from video sequence breaks. Real-time performance, over extended duration sequences, is realised via novel aspects of frame management within the mosaic representation and thus avoiding the high data redundancy associated with temporally dense, spatially overlapping video frame inputs. This managed set of image frames is visualised in real time using a dynamic mosaic representation of overlapping textured graphics primitives in place of the traditional globally constructed, and hence frequently reconstructed, mosaic image. Within this formulation, subsequent optimisation occurring during online construction can thus efficiency adjust relative frame positions via simple primitive position transforms. Effective visualisation is similarly facilitated by online inter-frame blending to overcome the illumination and colour variance associated with modern camera hardware. The evaluation illustrates overall robustness in video mosaic construction under a diverse range of conditions including indoor and outdoor environments, varying illumination and presence of in-scene motion on varying computational platforms.

1 Introduction

The problem of effective visualisation of multi-view imagery is present in most camera surveillance systems. With the development and increased deployment of pan–tilt–zoom (PTZ) capable surveillance cameras, the problem of limited situational awareness has arisen with respect to any given (current) camera viewpoint. The camera operator has to effectively make a constant compromise between viewing a wide angle picture of the overall environment under surveillance, or a limited, narrow angle view focusing on particular object of interest. In this paper, a video mosaic is constructed from the incoming video imagery providing the operator with the contextually aware ability to zoom in on a specific object of interest within the scene while having this detailed information presented in a panoramic (mosaicked) visualisation of the wider environment (i.e. situational awareness).

A range of prior work in this topic area exists, not only dealing with video mosaicking [1–3] but also in the very closely related problem of image-based panoramic stitching [4]. The input to such a technique is a set of overlapping images (video frames), and the goal is to align them spatially and produce a larger output panoramic image (mosaic). However, when we examine these techniques in detail, they are generally not the same. In the case of panoramic stitching the input is a set of unordered, high-resolution still images that overlap slightly. In the case of video mosaicking, the input video frames are temporally dense (i.e. multiple frames per second (fps)) and have a large spatial overlap. This is caused by the fact that the camera movement between two consecutive video frames, within the environment, is usually relatively small and constrained. Although it may appear that this secondary case presents a somewhat easier mosaicking problem, it consequently gives rise to issues of (i) frame (data) management for constructing large mosaicking sequences (because of the high data redundancy associated with temporally dense input video frames with significant spatial overlap) and (ii) the accumulated error associated with long-term sequential image registration [3, 5, 6] (i.e. long-term drift). Furthermore, in most cases video mosaicking algorithms have a real-time requirement and the problem is therefore most generally studied with a live video source – in our case a mobile PTZ camera is used for this purpose. By contrast, the panorama stitching problem involving still images does not have a real-time constraint and thus regular approaches to this related problem focus mainly on the quality of the output composite (panoramic) image rather than real-time performance and visualisation issues. This facilitates the use of a one-time optimisation approach in such image-based panorama problems [4].

In contrast to earlier recent work [3, 5], we present a pipeline for real-time video mosaicking through the use of constrained online bundle adjustment [7–9] supported by a novel online approach to both real-time processing and data (image frame) redundancy management. Extending prior work in the field [3, 5, 6], we explicitly resolve both camera rotation (i.e. pan–tilt) and focal changes (i.e. zoom (Z)) to facilitate the emplacement of high-resolution (quality) 'zoomed-in' image detail within the context of a lower-resolution mosaic of the environment (e.g. Fig. 1). Furthermore, we introduce the frame sieve concept to handle the large data redundancy which is associated with temporally and spatially dense input video frames. This is supported by graphics accelerated visualisation, using a dynamic representation of our mosaic as a set of overlapping graphics primitives, with adapted visual enhancements suitable for consistent mosaic visualisation within a real-time context. Overall this facilitates the construction of real-time video mosaics from a live video source, in the presence of in-sequence breaks (i.e. breaks in the 'video feed'), presented as a visually consistent mosaic rendered for real-time interactive visualisation. We illustrate the flexibility of this technique to both the rotational + zoom (i.e. PTZ) camera scenario (Fig. 11) in addition to translational camera motion (Fig. 8).

This paper makes several key contributions that both extend the mosaiking capability of prior work [3, 5, 10] and additionally address the practical issues of (i) efficiently managing image frame (data) redundancy [3, 10] and (ii) multi-image compositing [4] within a real-time context.

Our use of dual steps of *pairwise alignment* and *global bundle adjustment* decouples the online problem of '*next frame matching*'

Fig. 1 *Video mosaic of varying focal length (resolution) imagery from a PTZ camera*
a Individual input video frames – without camera Z applied
b Individual input video frames – with camera Z applied
c The resulting constructed mosaic from input images in (*a*) and (*b*)

within the mosaic map from that of building a globally consistent mosaic of the scene. This provides both the drift-free capability of [3, 5] but also additionally facilitates variation in camera focal length (Zoom, Z) within the mosaic itself (see Fig. 1). Following the work of [3, 10], a key-frame approach is introduced to manage frame redundancy because of overlap and maintain the complexity of the required global optimisation (*global bundle adjustment*) to a minimum. Loop-closing and in-sequence break recovery are both further supported via robust feature matching [3] over general camera motion in $\mathbb{R}^3$. Prior work in this area either does not address this complete set of issues within a real-time context (e.g. [3, 5, 10, 11]) or does so within the limitations of a pure rotational camera context targeting mobile device usage [6, 12]. By contrast we present a complete and flexible pipeline that facilitates the relative placement of mosaicked image frames as graphics primitives in $\mathbb{R}^3$, independent of the spherical (rotational) or planar (translational) projection models associated with prior work that uses a global mosaic image representation [3, 5, 10, 11]. An example of a video mosaic constructed within this context is illustrated in Fig. 1 where we see 'close-up' imagery, via camera Z presented within the global scene context of the scene.

This paper is outlined as follows: first, we detail the prior contextual work in this domain (Section 2) before detailing our base pipeline for video mosaicking (Section 3) that supports our generation of this example (Fig. 1). Real-time performance is in turn supported by dual image alignment and frame redundancy management within this context (Section 4). Final mosaic visualisation is further supported by consistent inter-frame rendering of frame primitives within a real-time context (Section 5). Experimental results are presented over a range of environmental contexts (Section 6) with conclusions summarised in Section 7.

J Eng 2015
doi: 10.1049/joe.2015.0016

2 Prior work

Prior work on panoramic imaging is well established with respect to the panoramic stitching of static images [2, 13, 14]. Work centres around an offline pipeline of inter-image alignment, global registration and final compositing to produce a given panoramic image [2]. Alignment can either be carried out using direct-pixel-based methods [1, 2] or, as in more recent work, based on feature-based matching [4]. From this initial alignment global registration is thus performed, commonly via bundle adjustment driven optimisation [15], with subsequent compositing consistent mutually of inter-image seam selection and blending [2, 13]. Within current abilities, initial alignment poses the greatest computational challenge and work in this area on feature-based correspondence has given rise to the concept of '*panoramic recognition*' [2, 4, 16].

By contrast a review of prior work on real-time video mosaicking presents a more potted history. Early work from [1, 17] presented impressive results but failed to address the in-sequence loop-closing problem of re-visited scene areas. Works by Robinson [1] and Sawhney *et al.* [18], such as [17], rely on a direct matching approach for optimisation which is prone to accumulated error causing alignment drift. Super-resolution mosaicking from video was achieved by Capel and Zisserman [19] using a feature driven framework that is not dissimilar to the later work of Steedly *et al.* [10] which explicitly considered the computational efficiency of mosaic construction. Although Steedly *et al.* [10] and Capel and Zisserman [19] did not achieve real-time performance; this was achievable using the contemporary direct matching, yet drift-prone approaches of [1].

Indeed numerous authors [1, 20–22] have shown real-time performance using a simple frame-to-frame image matching but these approaches inherently suffer from the accumulation of small alignment errors. These cause inconsistency problems within the mosaic when scene areas are re-visited (i.e. loop-closing) or for co-registration against secondary source imagery. Several approaches have been proposed to address this issue by either performing global optimisation [23] or explicit loop closing detection for each new mosaic frame [24]. More recently, Civera *et al.* [5] has considered this problem within the context of a self-localisation and mapping (SLAM) approach whereby an extended Kalman filter (EKF) is used to jointly estimate both the current sensor position and that of the scene features observed. Civera *et al.* [5] was able to demonstrate drift-free mosaicking in real time at frame-rate using this technique but suffered because of the scalability of the EKF technique to large numbers of image features. In reality, Civera *et al.* [5] used only about 3% of detected image features which limited the quality of the resulting mosaic. Following from [3, 5] developed an approach using a key-frame subset of the mosaic over which optimisation is performed using efficient second-order minimisation. As is common in SLAM approaches [5], the work of Lovegrove and Davidson [3] decouples the problem of '*next frame matching*' within the mosaic from that of building a global consistent mosaic. Both tasks are performed independently in separate threads following the paradigm of the parallel tracking and mapping (PTAM) [25] approach whereby the estimation of the current frame is only required to the nearest key-frame in the mosaic with global optimisation performed as a background task informing the main mosaic visualisation. Notably, Lovegrove and Davidson [3] uses a whole image alignment approach for frame-to-frame alignment in order to leverage all of the image texture and overcome the quality limitations of [5]. A key limitation of both Lovegrove and Davidson [3] and Civera *et al*. [5], for separate reasons, is the limited degrees of freedom over which they operate. The EKF approach [5], based on estimation of sensor position, is not robust to camera focal length changes (i.e. lens Z in the PTZ case) as this would translate as unintended camera motion resulting in potentially erroneous mosaicking. Similarly the whole image alignment approach of Lovegrove and Davidson [3] is not scale-invariant thus prohibiting mosaic construction under variable focal length (i.e. camera Z). Both works concentrate on image alignment [3, 5] to achieve drift-free mosaicking, ignoring blending issues of aesthetic appearance for mosaic presentation [4] but yet appear unable to cope with the case presented in Fig. 1.

By contrast to this prior work on drift-free mosaicking, we target an approach based on combined pairwise alignment and global bundle adjustment, following the geometry driven approach of [19] but within a similar PTAM [25] inspired approach to Lovegrove and Davidson [3]. We adopt this methodology (*pairwise alignment* and *global bundle adjustment*) to robustly estimate frame-to-frame correspondences of new frames in real time while global optimisation is similarly performed as a parallel task providing periodic global alignment updates. This facilitates the drift-free capability of [3, 5] while similarly allowing for an additional degree of freedom, image Z, within the mosaic construction (e.g. Fig. 1). Contemporary approaches performing real-time mosaiking via either feature-based or direct-pixel-based methods in certain application spaces are generally limited to camera rotation [6, 12] or lack loop-closing [11] – our approach inherently performs both as per Lovegrove and Davidson [3].

Furthermore, we introduce a novel variation on the key-frame concept of Lovegrove and Davidson [3] and Steedly *et al.* [10] to derive a frame sieving methodology to manage the growth complexity of this parallel optimisation task to the maximally required set of images for mosaic visualisation. Both Civera *et al.* [5] and Lovegrove and Davidson [3] noted the complexity issue in their respective approaches. Finally, we address the issues of effective mosaic presentation in the presence of mosaic artefacts caused by automatic gain control (AGC) present on modern camera hardware following a real-time substitute of the approaches proposed in [4]. Overall, we present a complete pipeline for video mosaic construction incorporating both novel aspects of parallel match optimisation and in-sequence scale changes (Z) realised within the practical context of frame (data) management for scalability and inter-frame blending for global mosaic visualisation as an efficient and flexible graphics primitive representation. Such a complete 'end-to-end' recipe for real-time video mosaicking is not presented in the prior literature [3, 5, 6, 10, 11].

3 From video frames to a mosaic

First, we outline a base-line approach to real-time video mosaicking, inclusive of data redundancy management via frame overlap detection, before proceeding to explicit aspects of maintaining real-time performance (Section 4) and mosaic presentation (Section 5).

3.1 Outline

Our video mosaicking approach is driven by initial feature point correspondences (*speeded up robust features (SURFs)* feature points [26]) between consecutive video frames. Subsequently, a RANdom SAmpling and Consensus (*RANSAC)*-based[27] methodology is applied for the dual purpose of outlier rejection and rapid identification of a maximally consistent set of detected inter-frame feature correspondences. On the basis of on these match correspondences *pairwise alignment* is used to robustly estimate relative frame-to-frame image transformation (Section 4.1). As the mosaic increases in size, a novel *key-frame*-based approach is used to identify frame overlap and facilitate redundant frame removal limiting global mosaic complexity (Section 3.4). *Global bundle adjustment* is performed in parallel, over this redundancy-filtered frame set, to eliminate accumulated error (i.e. drift) within this pairwise local registration process (Section 4.1). This provides a periodic update, in the form of *globally optimised* drift-free image registration, over all frames present in the mosaic. Additional *gain compensation* is used, again on a pairwise and global basis, to compensate for artefacts caused by the commonplace *AGC* present on most

modern cameras (Section 5). Break-in-sequence occurrences or occurrences where consecutive frame-to-frame matching fails are handled by invocation of specific *global frame search* overall all frames within the mosaic (Section 4.2) within the complexity limitation provided by prior redundant frame removal.

3.2 Feature detection and matching

The primary stage in our approach is the extraction of feature points from the image following the invariant SURF approach of Bay *et al.* [26]. The subsequent SURF feature descriptor characterises a given feature point as a vector in $\mathbb{R}^{64}$. SURF descriptor matching is performed using a simple L2 distance comparison embodied in an efficient *k–d* tree look-up structure using the *nearest neighbour ratio matching strategy* [26, 28]. Following this approach, a feature f_a from the first frame is considered a match to the feature f_b from the second frame if the descriptor distance $\mathrm{d}(f_1,\ f_2)$ between these features fulfils the following relationship:

$$\frac{\mathrm{d}(f_a,\ f_b)}{\min_i d(f_a,\ f_i)} < t \tag{1}$$

such that the ratio of this distance to the next closest match for a separate feature in the second frame, $f_i\ i \neq b$, is greater than a given threshold value $t \in (0,\ 1)$ (empirically set as $t = 0.65$ [28]). When considering the matches extracted in this previous filtering step, we must consider that a given amount of statistical outliers remain in the filtered correspondences. This is especially true in scenes containing moving objects or significant image noise. RANSAC fitting [27] is thus employed for outlier rejection to cope with this occurrence. In general, RANSAC determines which measurements are statistical inliers or outliers against an estimated model fit. Here our chosen model, for RANSAC fitting, is a frame-to-frame *projective transform* (i.e. a 3×3 *homography* matrix, $\boldsymbol{H}$) forming pairwise image alignment in the first instance [7]. This is obtained from multiple point correspondences by solving a set of linear equations using a *direct linear transformation (DLT)* as follows

$$\boldsymbol{x}_i' = s\boldsymbol{H}x_i \tag{2}$$

where $\boldsymbol{x}_i' \leftrightarrow x_i$ is a frame-wise feature point correspondence, in the set $i = \{0 \ldots n\}$ and $\boldsymbol{H}$ is the homography. Since the $\boldsymbol{x}_i'$ and $\boldsymbol{H}x_i$ are homogeneous vectors, they may differ in magnitude by a non-zero scale factor s. The equation can be expressed in the cross-product form as follows

$$(\boldsymbol{x}_i')^{\top} \boldsymbol{H}\boldsymbol{x}_i = 0 \tag{3}$$

allowing the derivation of a simple linear solution for $\boldsymbol{H}$ matrix following the DLT algorithm of [7] within our RANSAC framework. However, it has to be noted that as $\boldsymbol{H}$ is determined up to scale, only eight unknowns are present in the linear system of equations. As each point correspondence gives two linearly independent equations, we thus need a minimum of four correspondences to calculate the $\boldsymbol{H}$ matrix projection [7]. Empirically, we appear to obtain average number of point-wise matches between image pairs significantly above this threshold over which RANSAC is used to identify the maximally consistent model, $\boldsymbol{H}$. In this paper, this estimated projective transform model, $\boldsymbol{H}$, is only used to eliminate the statistical outliers from the set of identified matches over which *pairwise alignment* (Section 3.5) is subsequently performed.

In terms of feature matching for video mosaicking, as opposed to still image panoramic stitching, we can assume without loss of generality that consecutive video frames overlap to a given degree. There can be special cases when this assumption is broken but we handle these explicitly (see Section 4.2). This assumption simplifies the matching step, as we need to match the current video frame only with the previous one – that is, pairwise matching. For subsequent global bundle adjustment, we already know the prior video frame overlap relationships from this pairwise case and previous global estimations (providing a good initial set of frame co-registration estimates). This distinction between the pairwise and global image alignment will be further detailed in Section 4.1.

3.3 Camera geometry

Our feature matching and homography estimation is performed over an assumed pinhole camera model that may both rotate around its optical centre and Z in or out of the scene view – *in practical terms a stationary PTZ camera.* These movements result in a special group of homographies of the received video frames. In our case, each video frame is parametrised by an axis-angle representation of camera rotation and its associated focal length. The axis-angle representation is a four parameter model used to describe an arbitrary rotation in the three-dimensional (3D) space. It consists of a normalised vector which describes the axis around which the rotation will occur (the rotation axis is parallel to this normalised vector) and the fourth parameter is the amount of applied rotation – an angle of rotation. The focal length parameter is used to parametrise camera zooming and is essentially the video frame scaling factor.

In general, the problem can be formulated as a placement of video frames in 3D space around an origin, that is, every plane containing a video frame is perpendicular to the ray going through the centre of that video frame with the starting point in that 3D space origin. Since the assumed geometry is a constrained case of the general perspective homography, the 3×3 *homography* matrix representation can be computed from this representation

$$\boldsymbol{H} = \boldsymbol{K}\boldsymbol{R} \tag{4}$$

where $\boldsymbol{K}$ is the scaling matrix based on the focal length f and $\boldsymbol{R}$ is the rotation matrix derived from Rodrigues' rotation formula [29]. Note that the opposite transformation, that is, computation of the parameters of our assumed geometry from the 3×3 homography matrix is not possible in the general case because of the assumed constraints, that is, the homography matrix can represent transformations that cannot be represented by the scale and rotations parameters only. Here our consideration of camera PTZ extends prior work in the field [3, 5, 6, 12]. Furthermore, we will illustrate that by extending the bounds on this space we can additionally cope with camera translation in $\mathbb{R}^3$, in combination with f, by assuming planar mosaic projection as illustrated in Fig. 8.

3.4 Computing video frame redundancy

A further key issue is that of data redundancy management (i.e. frame redundancy management) within the context of continuous dense environment sampling from video. Given the reasonable bounds on the speed of camera motion within the environment, a large number of video frames will contribute largely duplicate information to the resulting mosaic. This growth in the overall mosaic complexity poses key scalability issues for our parallelised global bundle adjustment (discussed in Section 3.5) where we otherwise experience an unchecked quadratic growth in feature matching complexity (Section 3.2 [3]). Determining potential frame redundancy by determining if any given two video frames overlap, as well as the estimation of the extent by which the frames overlap, is thus key to realisation of this paper for generalised large-scale environments. This remains unaddressed in prior work [3, 5, 6, 12].

Hereby, let us consider two images i and j with their associated camera parameters (i.e. homography, $\boldsymbol{H}$). Let the image i be considered the reference (i.e. the ith image coordinate frame). We assume the centred, normalised image coordinates, so that the image

bounding box extends from -1 to 1 horizontally and from $-\frac{1}{ar}$ to $\frac{1}{ar}$ vertically, where ar is the aspect ratio of the image. Subsequently, the jth image bounding box needs to be *warped* from the jth image coordinate frame to the assumed, the ith image coordinate frame. This can be done by simply transforming the jth image bounding box using the associated camera parameters. The theory behind these transformations is described in further detail in [7].

Actual transformation of the jth image bounding box coordinates from the jth image coordinate frame to the reference (i.e. the ith image coordinate frame) is carried out as follows. First, the coordinates $(x_j,\ y_j)$ are transformed to projective geometry as follows

$$\boldsymbol{u}_j = \begin{bmatrix} x_j \\ y_j \\ 1 \end{bmatrix} \tag{5}$$

Subsequently, we use the homography matrices of images i and j to *warp* the coordinates as follows:

$$\boldsymbol{u}_i = H_i H_j^{-1} \boldsymbol{u}_j \tag{6}$$

The coordinates $(x_i,\ y_i)$ in the reference coordinate frame can be calculated by transforming them from the projective geometry:

$$x_i = \frac{\boldsymbol{u}_{ix}}{\boldsymbol{u}_{iz}} \quad y_i = \frac{\boldsymbol{u}_{iy}}{\boldsymbol{u}_{iz}} \tag{7}$$

After the jth image bounding box has been *warped* to the reference frame, we are looking at a simple 2D geometry problem. The ith image is represented by a rectangle and the jth image bounding box is an arbitrary quadrilateral because of the perspective transform that it underwent. To calculate the common area of the two image frames, the intersection points of these co-located bounding boxes are calculated from which the common overlap area for both quadrilaterals is obtained.

Furthermore, we obtain the percentage of inter-frame overlap between a set of multiple frames (Fig. 2) by using a numerical sampling method as opposed to the analytical method used in the two frame overlap problem. This is required in our subsequent use of a frame sieving approach to identify redundant, or largely redundant, frames within a given set (Section 4.3).

In this approach, the interior of the frame bounding box of image is initialised with multiple sampling points from which every frame covering the bounding box is examined (Fig. 2). The algorithm determines which of the sampling points represent redundancy because of coverage by other frames within the mosaic by examining sampling point to frame boundary intersection for all surrounding frames. After all such frames have been examined, the percentage of visible (i.e. uncovered, non-redundant) area is equal to the percentage of sampling points left. The idea is depicted in Fig. 2 where the blue frame represents the potentially redundant frame and the green and red frames represent existing, potentially covering, frames. In this example, there are 34 visible sampling points (blue) remaining from an original 63 resulting in a visible area equal of $34/63 \simeq 53\%$ (Fig. 2).

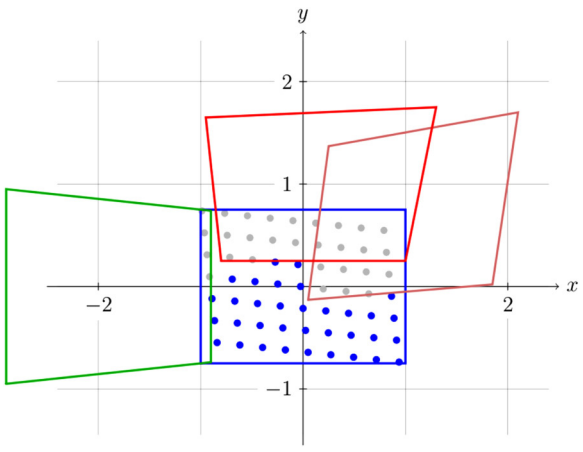


Fig. 2 *Visible area calculation in the case of multiple frame overlap*

Since the majority of frames are not significantly rotated relative to the subsequent frames, the pattern of the sampling points is chosen to provide higher accuracy of the area estimation when considering such cases of slight inter-frame rotation. As such, each sample row and column is slightly offset to the previous one by a small angle, θ_s, of the sampling density as shown in Fig. 2. Here, we can see that the left-most column of sampling points lies only partially inside the bounding box of the blue frame in question. Empirically, this has been found to give improved accuracy in overlap calculations in place of an axis aligned mesh in the presence of the often minor inter-frame rotations encountered in a full-frame video input operating at $\sim$25 fps.

3.5 *Bundle adjustment*

Finally, bundle adjustment [7–9] addresses the problem of optimising the 3D structure of the reconstructed scene. In essence this presents a large, sparse, geometric parameter estimation problem. The 2D positions on images constitute the measurement set and the camera parameters (scale and rotation) with 3D coordinates (in our case the 3D coordinates describe the projective geometry) of the feature points are the parameters being sought. The goal is to minimise the *re-projection error*, that is, a sum of squares of euclidean distances of observed and estimated image features.

Following [9], the *Levenberg–Marquardt* algorithm has proven to be the best suited in solving this non-linear least-squares problem. It can be thought as an interpolation between the *gradient descent* and *Gauss–Newton* algorithms. Despite the high dimensionality of the problem, the lack of dependence among most of the estimated parameters (i.e. the 3D points do not influence each other) makes fast calculation possible because the structure of the problem is sparse.

In general, a representation for the geometry used in the problem is not specifically assumed [9]. We specify the projection function f_{P} that computes the estimated measurement vector (i.e. the position of a point in the camera plane), given the camera and 3D point parameters. In our case the projection function f_{P} is given by the homography H_i (calculated from the scale and rotation parameters) of the camera i. For the estimated point $\boldsymbol{u}$ in the projective space we can calculate its ith camera coordinates (i.e. its position on the ith image). These coordinates in terms of projective geometry are described by vector $\boldsymbol{u}_i$, which can be calculated by applying the homography to the $\boldsymbol{u}$ point

$$\boldsymbol{u}_i = H_i \boldsymbol{u} \tag{8}$$

To calculate the 2D image coordinates $(x_i,\ y_i)$ of this point, we need to transform the coordinates from the projective geometry to the image coordinate frame:

$$x_i = \frac{\boldsymbol{u}_{ix}}{\boldsymbol{u}_{iz}} \quad y_i = \frac{\boldsymbol{u}_{iy}}{\boldsymbol{u}_{iz}} \tag{9}$$

These transformations are thoroughly described in [7] and here provide a robust reconstruction methodology within a real-time performance framework.

doi: 10.1049/joe.2015.0016

4 Maintaining real-time performance

On the basis of this prior overview of a video mosaicking approach, we now outline a specific methodology for maintaining real-time performance, by way of a continuously updated 'live' mosaic, for a largely unconstrained video input providing densely sampled, highly redundant scene imagery.

4.1 Pairwise and global image registration

Bundle adjustment is commonly performed over the entire image set to obtain a maximally global consistent mosaic [2, 4]. By contrast here, within real-time constraints, we cannot readily afford to perform global bundle adjustment over all prior video frames every time a new frame is introduced. However, it is still desirable to use global bundle adjustment in order to prevent accumulated error (drift) which otherwise occurs when only concatenated pairwise image alignment is used in mosaic construction [1, 5]. Our use of image alignment has been divided into two concurrent operations: (i) primary – pairwise frame alignment and (ii) secondary – global bundle adjustment in a similar vein to the work of [3] and [25]. We hence use bundle adjustment globally for overall accumulated error reduction (i.e. to remove drift accumulated from iterative pairwise frame-to-frame alignment and) within the overall mosaic representation.

Pairwise frame alignment is instigated for every new video frame occurrence. It takes only two frames, transforming the second to align it optimally to the first one (i.e. it does not change the parameters of the first frame). This is performed iteratively over the set of video frames occurring since the last global bundle adjustment. As each application of *pairwise frame alignment* only takes two video frames, it facilitates real-time alignment of incoming frames relative to those already present and globally adjusted within the mosaic.

Global bundle adjustment is performed periodically in parallel to the pairwise alignment case. This maximally aligns all of the current mosaic images simultaneously taking into account the overall structure of the mosaic and thus correcting errors accumulated from prior pairwise image alignment. Post-calculation of all the video frame alignment transformation parameters are updated in the visualisation. Although global bundle adjustment is computationally expensive (order of magnitude seconds for 10+ video frames present in the mosaic), it is effectively implemented as a parallel task periodically updating the overall inter-image alignment within the mosaic following the *Levenberg–Marquardt* algorithm [8] presented in Section 3.5.

Both image alignment methods (pairwise and global) require an initial estimate of the camera transformation parameters for each frame. For a new video frame, where these are unknown, we initialise these parameters with a coarse approximation using those of the mosaic frame to which this new frame has the most feature-based correspondences (see Section 3.2). Although coarse, this has empirically proven to be itself to be a sufficient initial estimate. The more common approach of initialising the input frame transformation with the parameters derived from a RANSAC-based estimation (see Section 3.2) have not shown any general improvement while occasionally resulting in significant mis-transformation of the new incoming video frame. In this paper, RANSAC is thus uniquely used solely for rapidly confirming the presence of a suitable match within the current mosaic frames and for eliminating the outliers from the image-to-image feature matching. It is the pairwise frame alignment process, over the remaining inliers, that is, used to compute the final image alignment registration of the new incoming frame to the existing mosaic presentation.

Global bundle adjustment requires prior knowledge of inter-frame overlap (i.e. spatially matching frames) within the current mosaic. This facilitates the extraction of additional image-to-image feature matches which have not been present in the previous pairwise chain of matching (i.e. image 1 ↔ image 2 ↔ image 3 ↔ etc.). These additional matches occur because the camera may re-acquire certain portions of the mosaic within its motion. For example, Fig. 3 presents a case where the eigth frame provides an additional match with second frame. Essentially, this is the case of either localised or global loop closing as discussed in prior work [3, 5] (Fig. 3).

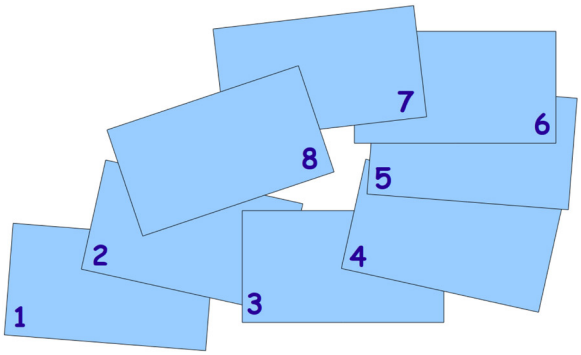


Fig. 3 *Illustrative example of loop-closing with the mosaicking of frames 1 ↔ 8*

Image overlap within the mosaic is identified directly by the relative inter-image geometry recovered (initially) from pairwise image alignment using the technique outlined in Section 3.4. When such an overlap event occurs then the identified frame pairs feature point matching as described in Section 3.2 using the per-calculated features from their initial feature extraction. Post matching the RANSAC sieve is again used for eliminating the outliers from these newly computed image-to-image feature correspondences and for confirming suitable matching has been found. From all of these identified pairwise image-to-image feature correspondences extracted we use the *union-find* algorithm [30] to merge the correspondences between feature points on different images and thus derive a global set of multi-point matches. This set of global feature correspondences from the input to global bundle adjustment and are essential in eliminating the accumulated error associated with drift within the mosaic.

4.2 Dealing with frame mismatches

Under certain conditions a valid feature matching between the last most video frame captured and the currently captured video frame cannot be found. There are many practical reasons for this in-sequence break in transmission: camera malfunction, large-scale movement within the scene or a featureless image frame (e.g. plain white wall). In such a case, our approach simply discards the input video frame and proceeds with the next received. However, if this occurs repeatedly we initiate a search for a global match – that is, we assume a possible significant movement of the camera and attempt to match a current video frame with anyone of all the current frames encompassing the video mosaic. This is based on the assumption that during the 'outage period' of the image matching the camera may have moved position within the global scene view and therefore it is reasonable to assume that a match may be found against any portion of the previously captured scene imagery. If the result of this search is successful, then the new frame is aligned with the identified matching frame. From this point, the regular operation of the dual pairwise image alignment and global bundle adjustment continues. Determination of a suitable match between the current video frame post-outage and a frame currently existing within the global set of mosaic frames is made on a simple threshold basis. First, we have a condition that there must be a sufficient amount of *statistical inliers* present in the set of feature point matches post-RANSAC sieve as outlined in Section 3.2. Additionally, the percentage of *inliers* in the set of

J Eng 2015
doi: 10.1049/joe.2015.0016

feature point matches must be greater than a set threshold t_{m2} (empirically set to 80%).

Overall the operation as described is desirable in several real case scenarios. Often the mismatch is temporary, and after one or two video frames that cannot be matched, the next received can be successfully registered due to the fact that camera movement is usually not significant over short periods of time. However, when a significant 'outage period' occurs our approach initiates searching for a global match based on the assumption of a potential significant camera movement within the scene. This global search is not performed instantly but instead after a given number of missed frames (dependent on the frame-rate) for two primary reasons. First, it is computationally expensive and thus to be avoided and second empirically it has been found that instant operation does not improve the overall performance of the video mosaicking approach as perceived by the viewer.

4.3 Key frames and frame sieve

In general, the input video frames can be considered to be temporally and spatially dense with most of these video frames having a significant spatial overlap resulting in high spatial frame redundancy. The concept of key frames is introduced to provide means of reducing this redundancy and identifying portions of the image data that are to be retained while others can be discarded because of spatial duplication (using methodology of Section 3.4). Key frames are dominant frames composing the mosaic (with relatively low-redundant information content).

Despite the fact that only a portion of the video stream is retained and contributes to final video mosaic, initially all of the input video frames are pairwise aligned and displayed for visualisation. Only after the current video frame (frame t) has been captured and aligned within the mosaic can a redundancy the decision about the previous one (frame $(t-1)$) be made. This decision is made in the concept of the *frame sieve* which essentially works on the identification of the key spatial frames within the overall video sequence (i.e. key frames).

Two criteria are used to classify a frame as redundant: (i) the percentage of area, threshold t_k, that is common to the last identified key-frame (Section 3.4) and (ii) the temporal distance to the last such key frame measured in terms of the frame index in the sequence (i.e. significant temporal separation) (This assumes a constant video frame-rate from a video source device.). Initially all the input frames are retained and displayed – both key frames and non-key frames. However, the frame sieve iterates over the set of video mosaic frames removing all non-key frames, except the most recently captured as a separate parallel process. This approach assures that the most recent video frame is always displayed within the mosaic in addition to those which give significant spatial coverage of the environment.

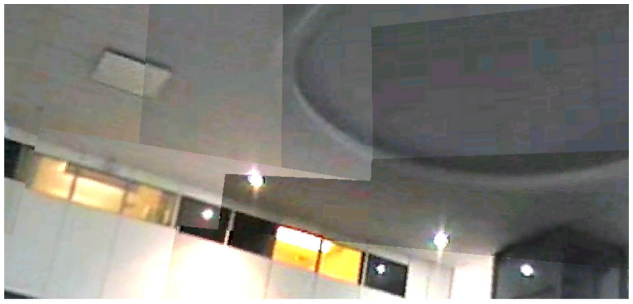
The second stage of the frame sieve erases all frames that are completely spatially covered by newer (temporally more recent) frames and thus are not visible within the mosaic. In practice, this heuristic procedure is slightly more complex. First, frames are interrogated in temporal order. If a frame is only partially covered by more recent frames (the specific level of coverage is specified by a threshold value, t_k), then we identify that this frame would leave holes within the mosaic if removed. The final stage in the overall pipeline of frame sieving is to erase the oldest frames if the number of globally recorded frames exceeds a given frame limit value f_l based on the memory management of a practical video mosaic implementation from a standard frame-rate (i.e. 25–30 fps) video source at reasonable spatial resolution. The frame sieve is thus composed of three main stages of frame filtering: (i) non-key-frame removal, (ii) overlapping frame removal and (iii) temporal frame removal. This delivers a highly practical, yet effective, frame management solution that in turn both manages the complexity growth of the *global bundle adjustment* approach and the storage requirements of the video mosaic for usage cases over a range environments (see results in Section 6).

5 Real-time mosaic visualisation

Individual video frames within the mosaic are represented as independent, textured rectangular graphic primitives in 3D space. Relative transformation parameters for every video frame, obtained from bundle adjustment (Section 3.5), are directly applied to arrange these graphics primitives appropriately using hardware accelerated visualisation. This independence of each frame, as opposed to the connected 3D geometry approach of [3], lends itself well to the *global bundle adjustment* and *frame sieve* approaches outlined previously (Sections 3.5 and 4.3) as it readily supports independent frame adjustment and removal as required. Furthermore, recent prior work on real-time mosaicking [3, 5, 6] does not address the issues of inter-frame *blending* and *gain compensation* in real time. These have been shown to be required for effective artefact free visualisation in similar work on still imagery [4].

5.1 Inter-frame blending

Video frame blending solves the problem of visible seams on the resulting image mosaic by blending the video frame border with the overlapped one. Brown and Lowe [4] suggest a multi-band blending methodology to merge the images in the composite panoramic image but such an approach is not readily possible within real-time bounds. Hence, we use a much simpler approach by associating an *α-channel* with each video frame. This channel specifies the opacity of a given part of the image. It is set to be completely opaque in the video frame centre with an increasing transparency towards the edges following a linear distribution. Despite the simplicity of the approach, the experimental results show that it is effective. In Fig. 4, we can see the seams apparent within the mosaic prior to blending (Fig. 4 upper) and an increase in the perceived quality of the mosaic post blending (Fig. 4 lower).



a

b

Fig. 4 *Example of inter-frame blending*
a Mosaic without the blending
b Mosaic with the blending applied

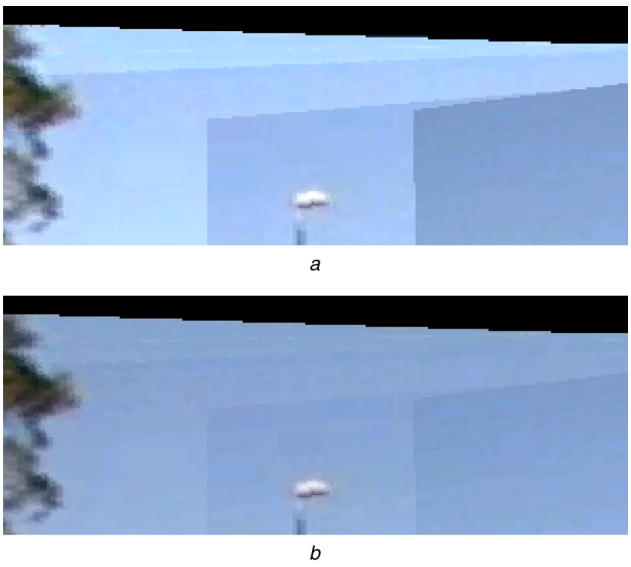


Fig. 5 *Application of gain compensation*
a Mosaic without gain compensation
b Mosaic with gain compensation

5.2 Gain compensation

Most modern video cameras are equipped with *AGC*, which automatically adjusts the camera exposure to achieve an automatically regulated level of image brightness and dynamic range based on the illumination conditions within the image [31]. However, this independence in gain between video frames in the sequence can consequently introduce undesirable effects into the video mosaic because the dynamic range of each video frame thus varies independently based on localised changes in lighting levels within the scene (see Fig. 5 upper). If we consider the example provided in Fig. 5 (upper), we see a camera that is initially viewing a darker portion of the scene (e.g. tree on left), pans right towards the brighter area of the sky (Fig. 5, upper right). During the transition the AGC decreases the overall camera gain, darkening the image as the overall illumination level entering the camera increases. Mosaicking a video frame sequence such as this shows that the sky becomes darker from frame-to-frame because of the effect of the AGC (Fig. 5 upper). Correcting these differences is required to improve the overall quality of the output video mosaic and mitigate the effect of the gain compensation introduced by the AGC (see example Fig. 5 lower).

The method for calculating such gain compensation is detailed in [4]. The compensation works in terms of minimising an error function, essentially the intensity differences between overlapping regions of the mosaic. The error function is defined as:

$$e = \sum_{i=1}^{n} \sum_{j=1}^{n} N_{ij} \left((g_i I_{ij} - g_j I_{ji})^2 \frac{1}{\sigma_N^2} + (1 - g_i)^2 \frac{1}{\sigma_g^2} \right) \tag{10}$$

where N_{ij} is the number of pixels in image i that overlaps with image j (note that N_{ij} does not necessarily equal N_{ji}), g_i is the gain parameter for image i we are seeking and I_{ij} is the mean value of intensity values of pixels in image i that overlaps with image j. The σ parameters are standard deviations of normalised intensity error and gain. Following from the prior work of [4], we choose these values to be $\sigma_N = 10$ and $\sigma_g = 0.1$ but the $(1 - g_i)^2$ term has been added to keep the gain parameters close to unity. Without it the optimal solution to the problem would be $\boldsymbol{g} = 0$, that is, all the images black.

The optimisation problem in this case can be solved analytically by setting the derivative of the error function to zero as follows:

$$\frac{\partial e}{\partial g_1} = 0; \quad \frac{\partial e}{\partial g_2} = 0; \quad \ldots; \quad \frac{\partial e}{\partial g_n} = 0 \tag{11}$$

This results in a linear system of equations which we solve via a Gaussian elimination method. This solution results in a recovery of the gain parameter vector $\boldsymbol{g}$ which contains the gain parameters for every video frame, that is, g_1, g_2, …, g_n. This is then applied to the video mosaic graphics primitives as texture parameters (separately for each frame) to result in the effect shown in Fig. 5 (lower) where we see a reduction in the AGC related artefacts in comparison to Fig. 5 (upper).

This solution solves the problem of gain compensation, in general; however, if applied as-is ((10) and (11)) a significant calculation has to be carried out for each new video frame. Owing to the real-time requirements we introduce, analogously to the bundle adjustment detailed in Section 3.5, the concepts of pairwise and global gain compensation. Global gain compensation performs the calculation of all gain parameters, g_i, for all video frames i present within the video mosaic. These are processed using the error metric as described in (10). The execution of global gain compensation is not bound to video frame capture but operates as a similar parallel task to that of global bundle adjustment (Section 4.1).

However, this introduces a problem of calculating the gain parameter for most recently captured video frames. Consider a mosaic with n globally gain compensated video frames. After a few subsequently captured video frames our mosaic will have m video frames, where $m > n$. However, the global gain compensation would not immediately calculate the new gain parameters, because of its parallel 'batch' nature, for all the m video frames. As a result we introduce a fast, pairwise gain compensation for temporary estimation of the gains g_{n+1}, g_{n+2}, …, g_m. This pairwise gain compensation takes only a single pair of frames at a time and adjusts the gain of the secondary frame to match that of the first (which is kept constant). It iterates from the last globally gain compensated video frame (frame n in the case presented above) to the most recently captured one (frame m) and thus calculates all the unknown gains g_{n+1}, g_{n+2}, …, g_m, which can be instantly supplied to the graphics visualisation and subsequently optimised in the next round of global gain compensation.

6 Results

We outline some example results of our technique over a range of both camera and workstation hardware to show illustrative results following the, largely subjective, evaluation methodology of prior work [1, 3, 5–6, 12, 16] in the field.

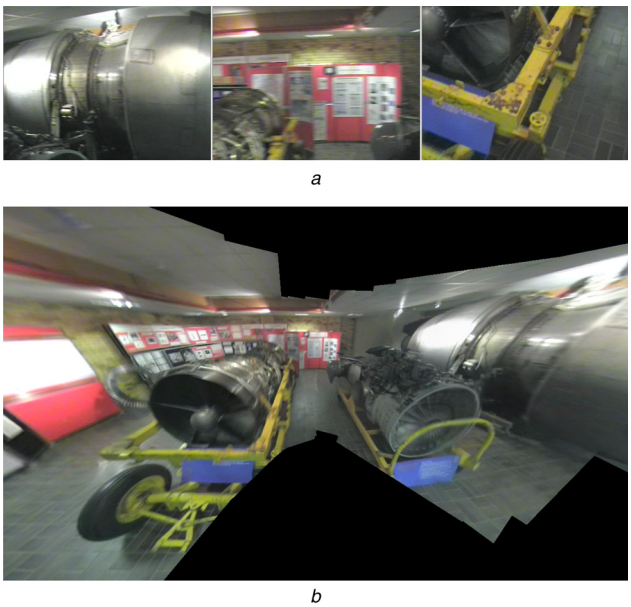
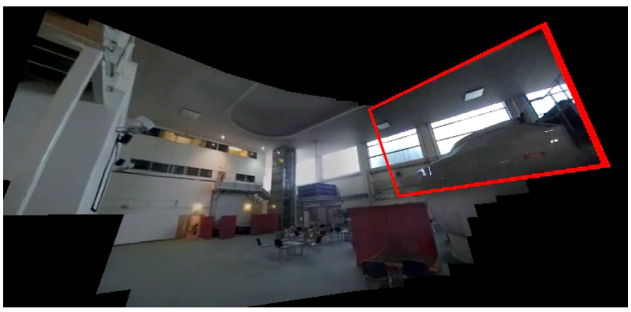


Fig. 6 *Wide angle video mosaic of the engine display room, School of Engineering, Cranfield University*
a Individual input video frames
b The constructed mosaic

J Eng 2015
doi: 10.1049/joe.2015.0016



a

b

Fig. 7 *Video mosaic from an interior scene comprising varying cross-scene illumination*
a Camera is pointed directly at the bright window
b Camera points at the dark portion of the scene

6.1 Equipment and environment

Results are illustrated on a range of differing video source equipment of varying quality and spectral response: a hand-held consumer camcorder (32× optical zoom and deinterlaced 352 × 288 pixel resolution), a near infra-red (IR) camera (wavelength 850 nm, IR diode lighting, deinterlaced 352 × 288 pixel resolution) and a low-cost consumer webcam (640 × 480 pixels, low-quality USB camera).

The presented method has been evaluated in various environments under varying lighting conditions including both indoor and outdoor environments. A few of the scenarios had a very unstable lighting conditions (Fig. 7) and others were fairly uniformly illuminated (Fig. 6). The case of a heavy movement present in the scene (i.e. considerable per cent of the video frame area contains moving objects) has also been examined (Fig. 9).

Our methodology was primarily evaluated using *Intel Core i7* (4 core) central processing unit (CPU)-based computer with a *Nvidia GeForce 9800 GT* graphics card. In addition some testing was also carried out on a standard *Intel Pentium M* 1700 MHz (single core common laptop) with an *ATI Fire GL T2* graphics card. Overall we aim to present performance on both high-end and low-end hardware.

6.2 Illustrative results

The first example, shown in Fig. 1, depicts an outdoor panorama with considerable zoom. This mosaic has been built up from an interesting camera movement. First, the camera swept the scene without any zooming (Fig. 1, upper), then it zoomed in and started to update the mosaic with a much higher effective resolution (the captured frames had constant resolution but the camera has zoomed considerably hence the information density for objects on the scene has increased, Fig. 1 (middle)). This is clearly visible within Fig. 1 (lower) where we can see the approximate, blurred nature of the scene detail on the left-hand side of the scene, whereas on the right-hand side we can see updated, high-resolution detail within the scene context based on the zoomed (higher resolution) information. This shows the overall robustness of the methodology and its components to variations in scale/zoom of the source video frame.

Let us analyse the second example – another standard case of the video mosaic, Fig. 6. The field of view of the visualisation is set to be wide, hence the mosaic looks distorted (especially at the corners of the view) which is to be expected in the case of the wide angle perspective projection. The black parts of the mosaic represent unknown regions, that is, parts of the scene that have not been captured by the camera. Despite the difficulties in maintaining a stationary hand-held camera and a short distance to the objects in the scene the video mosaic is still constructed properly. To be specific, the distance from the camera to each of the engines (not including the large one on the right side of the mosaic) was equal to approximately 1 m. This short distance amplifies the errors that result from the violation of the stationary camera assumption attributable to the hand-held nature of this video capture sequence. The result (Fig. 6 lower) shows the robustness of this method in the presence of minor disturbances.

The third example, Fig. 7, shows the test of indoor performance in case of varying lighting conditions (i.e. large lighting gradient within the indoor scene because of influx of light from windows within the environment). In this figure, the red box shows the position of the input video frame in the mosaic (Fig. 7). Comparing Fig. 7 (upper) with Fig. 7 (lower) one can observe the importance of the *gain compensation* described in Section 5.2. In Fig. 7 (lower) we point the camera at a darker portion of the scene, whereas Fig. 7 (upper) presents a case of '*blinding*' the camera with a direct light entering the lens. Despite the fact that the AGC of the camera changes the exposure considerably, the gain

Fig. 8 *Mosaic constructed from a top-down unmanned aerial vehicle (UAV) camera footage*

Fig. 9 *Motion (office chair and fan) present within the scene. The red box shows the current field of view of the camera*

compensation accounts for that and the mosaic is globally consistent in terms of brightness (Fig. 7 upper/lower).

Another example is a video mosaic constructed from a top-down UAV camera (Fig. 8). The problem of mosaicking an aerial footage is that it employs a different geometry – a mosaic of a flat surface captured by a camera moving and filming it from the above. Hence, we slightly modify our approach to approximate this geometry using an extended focal length parameter within the prior formulation (Section 3.3). Although the approach was not directly designed for such a task, one can see that it gives promising results

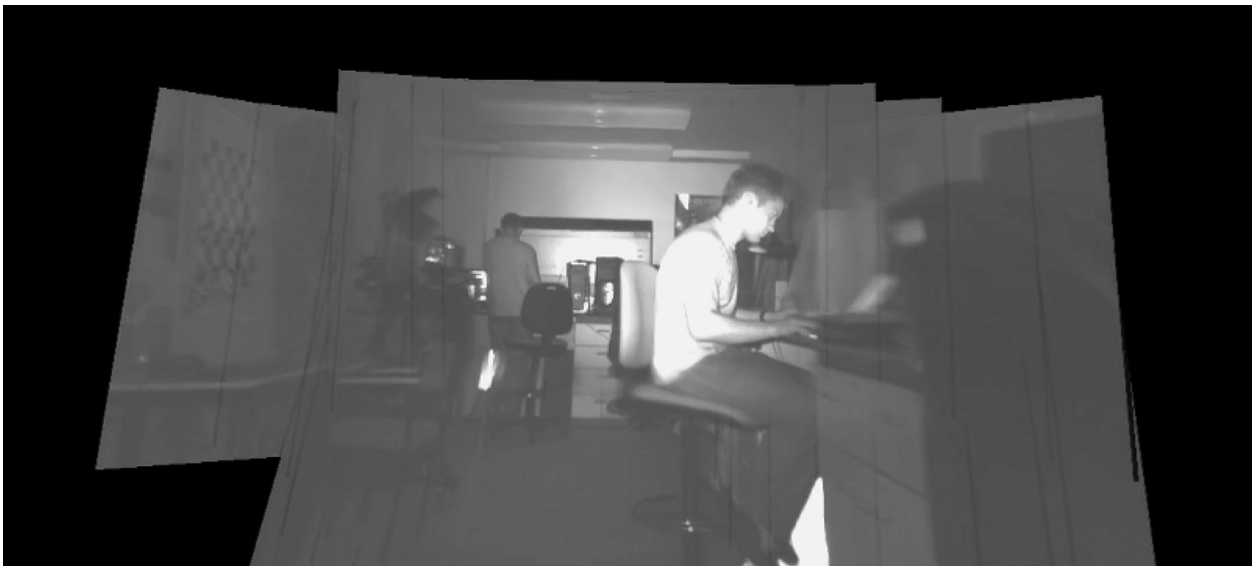


Fig. 10 *Mosaic constructed from a near IR imagery*

J Eng 2015
doi: 10.1049/joe.2015.0016

Fig. 11 *360° scene mosaic comprising drift-free loop closing*

indicating that a feature point-based matching could be employed with success to the problem of the aerial photography stitching.

The fifth example presents the case of a significant motion within the mosaic scene. In Fig. 9, the person is rotating on an office chair and together with rotating fan introduces a significant motion disturbance which must be recognised and dealt with. In our methodology, the application of a robust statistical RANSAC sieve (Section 3.2) allows for correct mosaic construction behaviour under such conditions. In the case where the methodology cannot find a suitable frame-to-frame feature match, the procedure of global search (Section 4.2) is used as a means to recover and proceed with ongoing mosaic construction from the video source. As we can see from Fig. 9 (upper) significant motion is present in primary area of the scene but as shown in Fig. 9 (lower) the mosaic is successfully constructed.

The sixth case presents the use of the methodology for the construction of a mosaic from a monochromatic Infrared (IR) video source. Fig. 10 depicts this case thus showing that the implemented method can operate on the footage taken using wavelengths outside the visible light spectrum.

The final case presents the 360° mosaic, that is, all of the visible horizon have been captured around the camera position. This is presented in Fig. 11 where we see it presents a wide angle view of a mosaic including effective 360° loop closing without obvious effects of drift.

Overall although some mild alignment and/or frame blending artefacts may be visible (Fig. 11), we can see that in the majority of cases (Figs. 1, 6–10) such artefacts are not present under varying lighting, video source and motion conditions.

6.3 *Performance characterisation*

We can characterise the real-time performance of our video mosaicking approach by considering the different aspects of the main processing requirements. In most cases, the core processing loop takes about 50 ms from which the SURF extraction step is the most computationally expensive. In more demanding environments, especially in those producing more feature points this core processing loop can take as long as 150 ms (with SURF extraction taking ~100 ms). From our experimentation, a single video frame is processed in a mean of ~75 ms over a range of environments. This translates into a frame-rate of ~13 fps which is clearly within the bounds of real-time performance for the tasks under consideration.

The performance of the polygon-driven visualisation also meets the real-time requirement with an average display refresh time of ~45–50 ms (average to worst case). This translates to ~20 fps (worst case) which is highly satisfactory for an interactive visualisation. As the frame-rate of the visualisation is greater than that of the main mosaic construction operation, the presentation of all captured video frames is assured.

The update frequency of the parallelised global bundle adjustment varies with the number of video frames currently registered within the mosaic. Although it is somewhat dependent on the relative positioning of the video frames and on the number of global

frame-to-frame matched feature points, for a low image count (∼20) the update algorithm does not take more than 500 ms to globally optimise the registration of the current mosaic. This provides an approximate global alignment update every half second into mosaic visualisation. As we consider larger mosaics (e.g. 80–100+ frames), this global bundle adjustment time can grow to ∼3–5 s but empirically appears to remain sufficient for the task of eliminating accumulated errors (Section 3.5).

Overall, performance on a modern quad-core CPU (allowing parallelisation as identified) facilitates the primary mosaicking of source video footage at 13 fps and the subsequent visualisation at 20 fps which is sufficient for real-time performance, visualisation and interactive user display (user specified PTZ within the mosaic visualisation itself).

6.4 Performance characterisation – low-end CPU

On a dual-core CPU platform, a moderately sized video mosaic, consisting of 40 video frames, was constructed for illustrative purposes. Despite the reduction in computational resource, the application still performed in a real-time manner. The primary processing loop took ∼200 ms on average (peaking at 300 ms, feature dependent), while the visualisation loop processing time was stable at ∼100 ms. The global bundle adjustment update rate was 2 s for frame registration optimisation. This translates as 5 fps for input video frame-rate and 10 fps for the visualisation.

The methodology was also tested on the standard single-core platform as specified (Section 6.1). Overall the performance was poor with a mosaic constructed from 20 frames resulting in 3.3 fps with a visualisation of ∼10 fps. Global bundle adjustment was again no more than 2 s. From this testing, we can see that with a reduction in computational resource the proposed methodology becomes increasingly less viable in terms of the parallelised aspects of global bundle adjustment and global gain compensation which extend earlier works within this field [1, 4]. We can see that while the approach is moderately viable on dual CPU parallelisation performance significantly drops for a single CPU platform.

7 Conclusions

In this paper, we have presented a feature point-based approach for the task of real-time video mosaicking. We present a variation on the prior approaches proposed within the field [3, 5], extending current mosaicking approaches to deal with in-sequence changes in scale (i.e. camera zoom) [3, 5, 6] and illustrating a flexible real-time visualisation architecture adaptable to both spherical (handheld) and planar (UAV) scene mosaicking tasks. Furthermore, we make explicit provision for effective mosaic visualisation, via online frame filtering and blending [4], and effective frame management that is overlooked in prior work [3, 5]. Overall an effective pipeline for flexible real-time mosaicking is realised with the context of a practical real-time application incorporating both a novel online mosaic construction approach, integration of frame blending and redundancy management aspects and a graphics primitive driven visualisation strategy.

The approach is shown to be robust to motion in the scene, varying lighting conditions and varying video source characteristics over a diverse range of environmental conditions. Real-time performance is characterised over varying computational platforms. Future work will investigate an extension to the combined use of real-time video mosaicking and stereo depth modelling from multi-camera systems in addition to addressing the aspects of wide-area deployment, usage and visualisation.

8 References

[1] Robinson J.: ‘Collaborative vision and interactive mosaicking’. Proc. Vision, Video and Graphics, 2003

[2] Szeliski R.: ‘Image alignment and stitching: a tutorial’, *Found. Trends Comput. Graph. Vis.*, 2006, **2**, (1), pp. 1–104

[3] Lovegrove S., Davidson A.: ‘Real-time spherical mosaicking using whole image alignment’, Proc. European Conference on Computer Vision, 2010, pp. 73–86

[4] Brown M., Lowe D.: ‘Automatic panoramic image stitching using invariant features’, *Int. J. Comput. Vis.*, 2007, **74**, (1), pp. 59–73

[5] Civera J., Davison A.J., Magallón J., *ET AL.*: ‘Drift-free real-time sequential mosaicking’, *Int. J. Comput. Vis.*, 2008, **81**, (2), pp. 128–137

[6] Wagner D., Mulloni A., Langlotz T., *ET AL.*: ‘Real-time panoramic mapping and tracking on mobile phones’. Proc. Virtual Reality Conf., 2010, pp. 211–218

[7] Hartley R., Zisserman A.: ‘Multiple view geometry in computer vision’ (Cambridge University Press, Cambridge, UK, 2003)

[8] Triggs B., McLauchlan P., Hartley R., *ET AL.*: ‘Bundle adjustment – a modern synthesis’, *Vis. Algorithms, Theory Pract.*, 1999, **1883**, pp. 153–177

[9] Lourakis M., Argyros A.: ‘SBA: A software package for generic sparse bundle adjustment’, *ACM Trans. Math. Softw.*, 2009, **36**, (1), pp. 1–30

[10] Steedly D., Pal C., Szeliski R.: ‘Efficiently registering video into panoramic mosaics’. Proc. Tenth Int. Conf. on Computer Vision, 2005, pp. 1300–1307

[11] Adams A., Gelfand N., Pulli K.: ‘Viewfinder alignment’, *Comput. Graph. Forum*, 2008, **27**, (2), pp. 597–606

[12] DiVerdi S., Wither J., Hollerei T.: ‘Envisor: online environment map construction for mixed reality’. Proc. Virtual Reality Conf., 2008, pp. 19–26

[13] Benosman R., Kang S. (Eds.): ‘Panoramic vision’ (Springer-Verlag, London, UK, 2001)

[14] Gledhill D., Tian G.Y., Taylor D., *ET AL.*: ‘Panoramic imaging – a review’, *Comput. Graph.*, 2003, **27**, (3), pp. 435–445

[15] Hartley R.I., Zisserman A.: ‘Multiple view geometry in computer vision’ (Cambridge University Press, Cambridge, UK, 2004, 2nd edn.), ISBN: 0521540518

[16] Brown M., Lowe D.: ‘Recognising panoramas’. Proc. Int. Conf. on Computer Vision, 2003, vol. **2**, pp. 1218–1225

[17] Szeliski R., Shum H.: ‘Creating full view panoramic image mosaics and environment maps’. Proc. of the 24th Annual Conf. on Computer Graphics and Interactive Techniques, 1997, pp. 251–258

[18] Sawhney H., Hsu S., Kumar R.: ‘Robust video mosaicking through topology inference and local to global alignment’. Proc. European Conference on Computer Vision, 1998, pp. 103–119

[19] Capel D., Zisserman A.: ‘Automated mosaicking with super-resolution zoom’. Proc. 1998 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, 1998, pp. 885–891

[20] Kourogi M., Kurata T., Hoshino J., *ET AL.*: ‘Real-time image mosaicking from a video sequence’. Proc. Int. Conf. on Image Processing, 1999, pp. 133–137

[21] Marks R., Rock S., Lee M.: ‘Real-time video mosaicking of the ocean floor’, *IEEE J. Oceanic Eng.*, 1995, **20**, (3), pp. 229–241

[22] Morimoto C., Chellappa R.: ‘Fast 3d stabilization and mosaic construction’. Proc., 1997 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, 1997, pp. 660–665

[23] Kim D., Hong K.: ‘Real-time mosaic using sequential graph’, *J. Electron. Imaging*, 2006, **15**, (2), pp. 47–63

[24] Zhu Z., Xu G., Riseman E., *ET AL.*: ‘Fast generation of dynamic and multi-resolution 360 panorama from video sequences’. IEEE Int. Conf. on Multimedia Computing and Systems, 1999, vol. **1**, pp. 400–406

[25] Klein G., Murray D.: ‘Parallel tracking and mapping for small AR workspaces’. Sixth IEEE and ACM Int. Symp. on Mixed and Augmented Reality, 2007. ISMAR, 2007, pp. 225–234

[26] Bay H., Ess A., Tuytelaars T., *ET AL.*: ‘Speeded-up robust features (SURF)’, *Comput. Vis. and Image Underst.*, 2008, **110**, (3), pp. 346–359

[27] Fischler M., Bolles R.: ‘Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography’, *Commun. ACM*, 1981, **24**, (6), pp. 381–395

[28] Lowe D.: ‘Distinctive image features from scale-invariant keypoints’, *Int. J. Comput. Vis.*, 2004, **60**, (2), pp. 91–110

[29] Koks D.: ‘A roundabout route to geometric algebra’. in ‘Explorations in mathematical physics’ (Springer Science, 2006), pp. 147–184.

[30] Cormen T., Leiserson C., Rivest R.: ‘Introduction to algorithms’ (McGraw-Hill, New York, USA, 2001)

[31] Solomon C., Breckon T.: ‘Fundamentals of digital image processing: a practical approach with examples in MATLAB’ (Wiley-Blackwell, Chichester, UK, 2010)