Reconstruction of the Pose of Uncalibrated Cameras via User-Generated Videos

Stuart Bennett
Cambridge University
sb476@cam.ac.uk

Joan Lasenby
Cambridge University
jl221@cam.ac.uk

Anil Kokaram
Google Inc.
anilkokaram@google.com

Sasi Inguva
Google Inc.
isasi@google.com

Neil Birkbeck
Google Inc.
birkbeck@google.com

ABSTRACT

Extraction of 3D geometry from hand-held unsteady uncalibrated cameras faces multiple difficulties: finding usable frames, feature-matching and unknown variable focal length to name three. We have built a prototype system to allow a user to spatially navigate playback viewpoints of an event of interest, using geometry automatically recovered from casually captured videos. The system, whose workings we present in this paper, necessarily estimates not only scene geometry, but also relative viewpoint position, overcoming the mentioned difficulties in the process. The only inputs required are video sequences from various viewpoints of a common scene, as are readily available online from sporting and music events. Our methods make no assumption of the synchronization of the input and do not require file metadata, instead exploiting the video to self-calibrate. The footage need only contain some camera rotation with little translation — for hand-held event footage a likely occurrence.

Categories and Subject Descriptors

I.4.1 [**Digitization and Image Capture**]: Camera calibration; I.4.8 [**Scene Analysis**]: Stereo; I.4.9 [**Image Processing and Computer Vision**]: Applications

Keywords

Camera pose estimation, 3D reconstruction, user-generated videos, self-calibration

1. INTRODUCTION

User-generated video is a huge and growing form of online media: over 100 hours of footage are uploaded to YouTube every minute. With such vast amounts of video, the challenge becomes one of presenting useful ways for the content consumer to navigate it all. A common form of online video is that of outdoor events, sporting or musical, where many hundreds of audience members each make their own recordings on their smartphones. A search will reveal many videos for the event, but selecting between these results may require several attempts to meet an individual's tastes. Presenting a spatial interpretation of the positions from which the videos were recorded (as in Figure 1) helps users make a more informed choice.

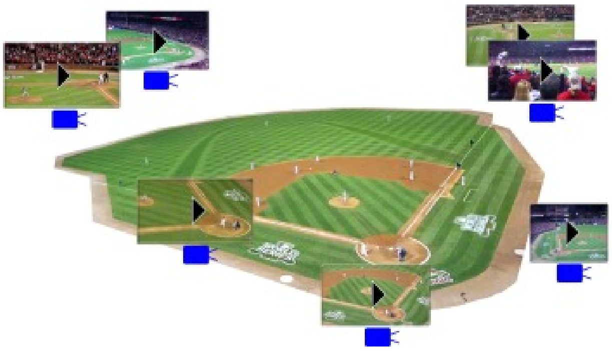


Figure 1: Relative localization of camera positions from crowd-sourced video clips in a 3D playback interface

From this application, comes the subject of this paper: a system capable of estimating *approximate* (to within a few metres) relative 3D camera locations, when given unsynchronized video sequences covering a common scene. Furthermore, the videos will be of variable quality, with issues of random occlusions, poor lighting, low cost image sensors, and unsteady camera motion all stemming from the use of inexpensive consumer devices by non-professional camera operators: the system necessarily copes with such conditions as a matter of course, in some cases exploiting these characteristics.

We presume no details of the optical system, nor indeed any knowledge of the camera model itself — video metadata are not nearly so complete as that from digital still camera files: no camera model, image sensor dimensions, or focal length information. We must therefore infer everything from the image stream itself, while also dealing with motion-blur and extracting useful and correct feature correspondences. The target application for this system is in reconstruction of camera positions at events where there is likely to be much available footage recorded on consumer-grade devices: sports matches and music concerts being key examples, but its use is not necessarily restricted to these two cases.

This paper proceeds as follows: after a discussion of work addressing somewhat similar problems, Section 2 details and discusses the design of the system, illustrated with results from a typical real-world dataset, which inform and validate our design. Section 3 gives accuracy results of the system on data with known ground-truth. Section 4 briefly presents further results on an alternative real data scenario, while Section 5 discusses and concludes with the advances made by the proposed system, and highlights possible future directions.

1.1 Related work

The problem posed bears many similarities to Structure from Motion (SfM), where a scene, or object, is observed over time by a moving camera which records differing 2D projections of the scene as it travels. From such data, approaches exist to variously recover the 3D information of the scene, the track of the camera's path, and the pose of the camera at each instant. The *n* frames captured by the moving camera may instead be seen as *n* static cameras each capturing one frame, which is not dissimilar to the user-generated video problem posed above. The main difference lies in the *n* cameras in the event scenario all being different, each with unknown parameters describing their optical system, which in turn greatly affects how the projection of the 3D real world is captured in 2D by the camera.

Much previous work has considered the processing of photographs. [23] describes a complete system for combining Internet-sourced photographs of tourist attractions, to form full 3D models, with the follow-up paper 'Building Rome in a day' [1] extending the approach to city scale. While these use a data-source with significantly different characteristics, the underlying approach is SfM, and several of the methods used — SIFT feature matching ([15]) and RANSAC parameter estimation — apply to the video-based problem. One of the differences is availability of camera optical parameters; Snavely et al. start construction of their camera network by only considering images for which a focal length estimate is available, which typically comes from EXIF data.

The issues of image quality and calibration affect the applicability of many photograph-based systems. Work exists in abundance for applications spanning urban reconstruction and outdoor navigation where calibrated cameras are presumed. The 'Videoscapes' paper [24] is an example of an uncalibrated video-based city exploration system, where moving cameras allow the user to move between common points in purpose-captured videos, and to swap video at such a 'portal' via an aesthetically pleasing transition. A graph of the ways videos are connected between portals is provided, but true spatial positioning is only available with supplementary GPS and orientation data; geolocation from video is left as future work. Another example of a system augmented by GPS (and inertial navigation) is [18].

[3] also deals with rendering viewpoint transitions, noting that their geometry recovery may have questionable 3D accuracy; they merely require the rendered scene to *look* correct. Other systems also focus on the scene reconstruction, rather than on camera positions, the depth estimate of which is particularly challenging. Ballan et al. further use audio to synchronize the videos, which vastly simplifies feature matching; unfortunately in sports event scenarios the recorded audio is highly localized however.

Other works dealing with unsynchronized videos need large silhouettes of performers (e.g. [21]), or the availability of dense point correspondences (e.g. [13], [14] and [7]) — unlikely given the expectation of poor quality videos taken from quite different viewpoints somewhat distant from the performance. Donate and Liu, along with [11] and many other simultaneous localization and mapping (SLAM) publications, also exploit the presumption of a roving camera, whereas event footage is often taken from essentially static viewpoints. Several of these systems calibrate using [19], which in turn is based on [17]. This paper describes a linear method of calibrating, specifically with a view to recovering focal length when other parameters are known — relevant conditions for our case. They find that simplified calibration algorithms not trying to exactly recover all parameters can counter-intuitively lead to better results.

Other notable self-calibration literature includes [9], which describes a practical algorithm supposing at least three images from fixed zoom cameras are available, and [6], which allows different optical parameters to be fixed or vary (with varying numbers of images required), permitting variable zoom. In both cases it is presumed the camera is stationary: fixed in space, but free in orientation, an appropriate scenario for event footage. [5] is an interesting paper, noting that previous approaches over-parameterize the problem and discounting non-focal length optical parameters as being essentially irrelevant to real data. They use algebraic geometry techniques to estimate focal lengths from two images at different zoom-levels separated by a pure rotation, using only three point correspondences, and claim greater noise tolerance and accuracy than common bundle-adjustment methods.

Exploratory experiments in the early stages of our research used the 'Bundler' software (from the 'Building Rome in a day' research) [22] with event stills, but distorted or highly implausible reconstructions were produced. Similarly flawed reconstructions resulted from the state-of-the-art VisualSfM software [25]. These failures are probably in no small part due to failing to correctly estimate focal lengths from images without EXIF data. No canonical example exists of a system to deal with crowd/event videos (little camera motion relative to scene) and infer camera location. Processing burden has to be some explanation for this, and this consideration is reflected in our preference for straightforward and parallelizable techniques.

2. SYSTEM DESCRIPTION

In this section we present the multiple stages of processing that must be performed on the input video sequences.

2.1 Overview

In order, the stages of the system are as follows:

Frame selection Reducing the quantity of data to process by finding suitable frames with many interest points from each video.

Feature matching Extracting and matching feature points, with a high degree of confidence, between frames both from differing viewpoints (for 3D reconstruction) and from the same video sequence (for camera calibration).

Camera parameter estimation Inferring parameters for each camera's optical system, particularly focal length and pixel aspect ratio (intrinsic calibration).

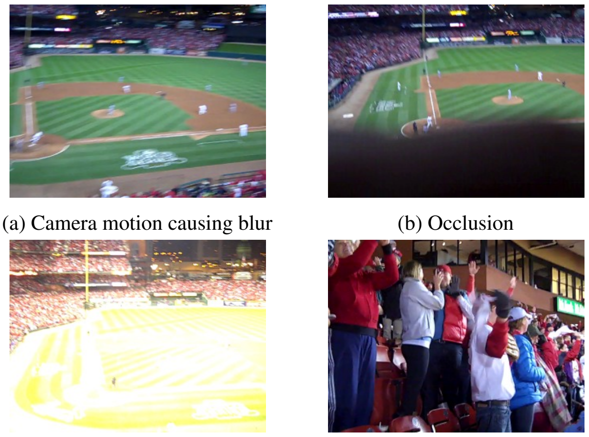
3D reconstruction Taking the intrinsic calibration parameters, and matched feature sets, and calculating camera pose (extrinsic calibration) and 3D scene co-ordinates.

With fully calibrated cameras we have achieved the desired output of the system, namely approximate relative camera positions, but it should be noted that being now in the possession of a full camera geometry, additional outputs, for example 3D structure modelling and viewpoint warping, can potentially also be realized.

2.2 Frame selection

A 5 minute video clip, at 30 frames per second, has 9000 frames. With a 1920 × 1280 pixel resolution, exhaustive processing would have to consider 22,000 million pixels. For reasons of computational tractability, it is vital to first cull the video sequence into some more manageable collection of frames, at low computational expense.

User-generated event footage tends to be unstable, with camera-shake and low-cost hardware leading to many blurry frames. Such frames are much less useful for accurate extraction of feature points, and so are good candidates for culling, presuming some fast method



(a) Camera motion causing blur

(b) Occlusion

(c) Lighting/sensor issues

(d) Extraneous cutaway shot

Figure 2: Examples of discardable frames from typical footage

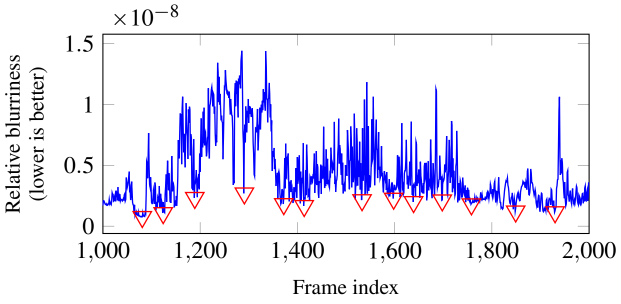


Figure 3: Measure of blurriness for 1001 frames of a baseball clip. Minima (red triangles) found using 2s sliding window

of detecting them. Figure 2 gives some examples of these issues observed in a test video. Any simple sampling-in-time approach must be ruled out: in the presence of camera-shake, missing 'good' frames is very likely. While it may appear that poor quality frames carry less information, and hence a measure of entropy might be indicative, experiments showed this not to be a discriminative metric.

Instead we use a 'relative blurriness' measure, b_t, comparing blurriness between frames from one video sequence, taken from the video stabilization literature ([16]):

$$b_t = \frac{1}{\sum_{\mathbf{p}_t} \left\{ \left((f_x * I_t)(\mathbf{p}_t) \right)^2 + \left((f_y * I_t)(\mathbf{p}_t) \right)^2 \right\}}, \tag{1}$$

where f_x and f_y are two derivative filters along the x and y directions respectively, and the pixel co-ordinates of the image at frame index t (I_t) are given by $\mathbf{p}_t$. As noted in [16], this *inverse of the sum of squared gradients* is robust to image alignment error and, with two simple gradient convolutions in x and y per frame, it parallelizes trivially.

b_t gives a comparative measure from one frame to the next, and we select minima (i.e. frames with comparatively low blurriness) in this measure across all t, using a sliding window approach. A two second window ensures that brief changes in video subject are not lost, but guards against excessive repetition of barely changing scenes. The locations of these minima for one video are shown by red triangles in Figure 3. It is apparent from the figure that the measure is highly discriminative, with the blurriness score varying significantly from frame to frame.

The reconstruction stage *needs* around 10 matched points, but this lower bound relies heavily on those points being both true matches and very accurately localized. In an automated process it is very likely that some matches will be erroneous, so in fact it is desirable to have many matches (over 100), of which the vast majority (80-90%) are good, so that by randomly sampling the matches a consensus of in-/out-liers can be reached, via methods such as RANSAC. While the comparative blurriness filtering produces sharp, textured frames, there is no guarantee that the frames will have many interest points, prerequisites for forming matches, when processed by a general-purpose automated feature detector.

A second filtering is therefore performed, by applying a general-purpose feature detector to each of the selected frames, and simply taking the frames with most features. Our implementation uses the very efficient FAST-9 detector ([20]), being the most reliable of the FAST-*n* detectors. For short single-viewpoint video clips, as are typical online, disregarding all but the highest scoring ten frames gives satisfactory results. For longer sequences, selecting more frames, separated in time, is required.

The frames considered optimal by the above processes are sharp, strongly textured, and featureful. In empirical evaluation the selected frames are always 'good' by human standards; those that one would manually select as appearing sharp and featureful.

In footage of sports events the frames selected are generally not of the playing surface — wide or crowd shots are much more common. Likewise in concert video, selected frames are rarely of zoomed foreground shots — background set or stage shots are more likely. Such frames are essentially never recorded simultaneously in multiple views; hence matching of frames between videos must be performed without regard to video synchronization. However, since video, especially sports footage, is challenging to automatically synchronize, the forcing of an unsynchronized approach by the frame selection method is in fact advantageous. The caveat is that should the input videos not contain some essentially static scene elements the subsequent matching and reconstruction stages are unlikely to succeed: this is accepted as a case beyond the scope of our system; for the intended footage most videos do *not* suffer from this issue.

2.3 Feature matching

Robust feature matching is necessary to have reliable point correspondences for 3D reconstruction and for camera calibration. Given the range of potential input videos, specialized feature detectors and matchers are problematic: they are likely to be superior for their specific task (say, detecting a tennis court), but apart from then needing an immense array of purpose-made algorithms, *choosing* the correct one to use becomes an additional undertaking. Both due to this, and consideration of the (in)ability to deal with videos of situations for which no tuned algorithm exists, we only consider general-purpose feature detectors.

While not the fastest feature detecting and describing technique, the SIFT algorithm ([15]) has been found in several studies to provide one of the best performing descriptors. We therefore process all of the highest scoring frames with Lowe's SIFT algorithm to obtain a set of keypoint descriptors and respective location information for each frame.

The first stage of matching is a standard technique described in Section 7.1 of the SIFT paper. Each SIFT descriptor has a point in a 128 dimensional space associated with it. A putative match with a feature from frame a is made by finding the nearest descriptor from frame b's set of features. Lowe makes the observation that if the *second-nearest* match has a similar distance to that of the first, i.e. the ratio of distances is close to 1, we should have little reason to trust the match: the strong second-best might be true, or even both

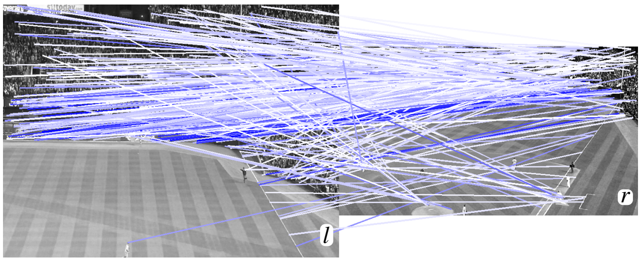


Figure 4: Matches after ratio-of-distances filtering. Matches with weaker ratio have lower saturation (more white than blue).

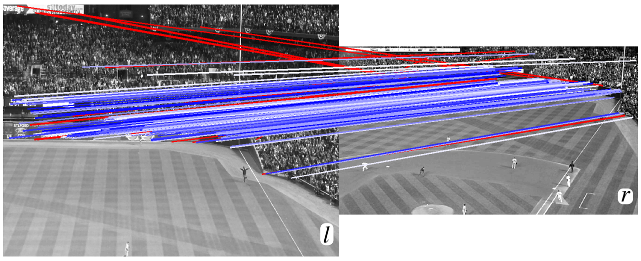


Figure 5: Matches after scale and orientation filtering. Matches with weaker ratio have lower saturation (more white than blue), incorrect matches in red.

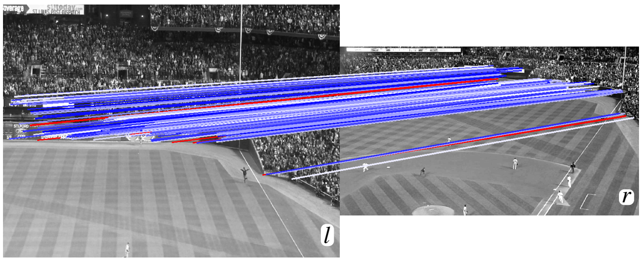


Figure 6: Matches after displacement filtering. Matches with weaker ratio have lower saturation (more white than blue), incorrect matches in red.

may be false, the true match having been missed altogether.

Experimentally, reliable matches tend to occur frequently when accepting a distance ratio less than 0.6. As the ratio gets closer to 1, using say 0.8, more valid matches are found, but the proportion of false matches increases simultaneously. This situation is seen in Figure 4, where matches at 0.6 or below are dark blue, and mostly good, while weaker matches, those at 0.8 being white, are more likely to be erroneous. We would like to retain the extra matches found by using a 0.8 acceptance threshold, but to reliably filter out many of the accompanying false matches.

Using a naïve RANSAC-like technique alone in the presence of many false matches is computationally onerous. The second stage of matching then exploits the expectation that the scene contains static features (since the videos are unsynchronized we cannot rely on the scene contents moving/deforming in the same way over time). Fortunately many event scenes contain featureful fixed background structures: buildings, stages, grandstands, advertisements, even trees, for example. Events footage commonly has an audience all contained by these background structures in an arena, with the purposefully recorded action in the foreground. Thus there is a significant depth from the camera to the background, and the views of a given background will not change greatly from cameras a little distance apart. SIFT annotates each keypoint with scale and orientation measures. We observe that while there is no need for these to be the *same* between two images, bearing in mind the depth and similarity-of-view considerations, they ought to be *related* for good frame matches on background features:

- *Scale* ought to be related by a (roughly) constant factor
- *Orientation* ought to be related by a (roughly) constant difference.

The key idea here is to form histograms of scaling factors/orientation differences over all matches and only take those matches falling in the histogram bin with the greatest count, essentially finding the modal scale factor/orientation difference. This approach will work well and quickly even in the presence of many false positives (unlike RANSAC), assuming that erroneous scales/orientations are uniformly distributed. For the scaling measure the technique is identical to that proposed in [2], in the context of matching gravity-aligned orthophotos of buildings, but no similar orientation method is proposed. The technique is also closely related to that of Weak Geometric Consistency by Jegou et al. [10], successfully used for efficient image retrieval from large databases. In Figure 5 the scale and orientation consensus filtering cleans the matches up markedly. Only 25 (13%) incorrect matches remain, while weaker yet correct matches have been retained.

There are two important implementational details: the first is that in view of the '(roughly)' above, the bin widths of the histogram are important. We use fairly tight widths: the scale window is $\pm 10\%$ and the orientation difference window is $\pm 7°$. Linearization of scale differences into equal width bins is easily achieved by considering scale in the logarithmic domain. Once the best bin is known, the acceptance widths are a little larger: $\pm 25\%/\pm 15°$ of the respective bin centres ensures that even features suffering significant perspective distortion are not unnecessarily pruned.

The second detail relates to the bin edges: in order to not bias the binning by using some particular bin-edge alignment, one should attempt the histogramming with all possible offsets. In practice we approximate this by iteratively forming the histogram, sliding the bin-edges along using a small discrete step-size.

It is worth noting that this histogram strategy lends itself to a similar ratio-of-improvement threshold (as used by the descriptor distance matching), where further processing of a frame is declined if there is no clear modal bin, implying no obvious overall scale/orientation, and hence no matches can be accepted.

The third stage makes use of the feature position data; the coordinates of each localized keypoint. We further assume that background objects present some continuous face toward the cameras. This being the case, all the per-object feature-point displacements from the frames of one view to another ought again to be similar. We transform the matched keypoint displacement vectors into polar co-ordinates, expressing each vector as a length and angle. These quantities are respectively amenable to modal scale and orientation filtering as described previously.

Finding one mode across the whole frame assumes that the observed displacements are all at roughly the same depth from the camera. A straightforward solution to this unjustified limitation is to tile the frame, into say 4 $\times 4$ rectangles, and accept the modal displacement for each tile, subject to a tile having enough putative matches to permit a modal displacement to be meaningful. This gives differing modal motion vectors per tile, in turn allowing the use of feature matches at differing scene-depths. The degree of tiling is an application specific parameter, but can be dynamically assigned by considering the number of potential matches in each tile area: there must (in general) be sufficient to have confidence in the modal vector being correctly estimated.

The displacement vectors will clearly appear differently depending on which camera viewpoint they are relative to. A high degree of confidence in the overall accepted matches is obtained by taking the *intersection* of the accepted modal matches as gained when processed independently from *both* views. After displacement filtering, in Figure 6, only 12 invalid matches remain (7%), none of them obvious, and only 5 (3%) correct matches have been lost. The very small proportion of incorrect matches should not deteriorate the eventual reconstruction, due to the RANSAC approach in use, though improvement of matching remains a topic of research.

2.4 Camera intrinsic calibration

With user-generated video from unknown consumer cameras we have no idea of the optical system of the camera, and are unlikely to have convenient footage of a known calibration object. Having a good calibration of the cameras is however vital. While the effects of an incorrect calibration on the 3D reconstruction of feature points into world-space *can* be neglectably subtle, we wish to retrieve the positions of the cameras. For this the internal parameters make the difference between a camera physically located very close to the world points, and a distant one using a high magnification lens.

We may presume a pinhole camera model, minimal pixel skew, and an optical axis roughly coincident with the centre of the image sensor. The pinhole model restricts the system's use to captures without significant lens distortion, but the majority of modern consumer optics give sufficiently rectilinear projections for our fairly weak accuracy requirements. No simplifications, other than constraining sanity bounds, are available for the focal length parameters however, and unlike digital photographs these are not embedded in file metadata. We can however, exploit the properties of video image data: a series of frames close in time will capture almost the same scene and should the camera rotate, even by a few degrees, during this time, self-calibration is possible (assuming translation of the camera, relative to the distance to the world-points, is negligible). If the two image-planes formed from two frames are related by some rotation, the camera must lie at the point where the plane normals intersect, thus the depth-ambiguity is trivially resolved.

Clearly estimation of the intersection point relies heavily on finding the transformation relating the image planes, with the incorporation of multiple image planes making the estimation more stable. Unfortunately, the rare nature of stable, non-blurry, wide-angle shots means that one may only have two frames to estimate the relationship, and hence a technique capable of operating with only two image-planes is highly desirable, albeit with the knowledge that the results may not be ideal. The previous works [9] and [6] present practical and appropriate methods to find the calibration via camera rotation, but use multiple (three or more) images; the following develops from their approaches to only *require* two frames.

The intrinsic camera matrix, expressing the optical parameters of the camera, and part of the pinhole camera model, may be given as:

$$K = \begin{bmatrix} \alpha_x & \gamma & u_0 \\ & \alpha_y & v_0 \\ & & 1 \end{bmatrix}, \tag{2}$$

where α_x and α_y express the optical focal length in pixels (in x and y directions respectively), γ gives the pixel skewness coefficient, and u_0 and v_0 the co-ordinates of the principal point. We assume γ to be zero (i.e. non-skew pixels), and the principal point to lie at the centre of the image, which by choosing an appropriate origin for the 2D feature point co-ordinate system allows u_0 and v_0 to both also be zero. Even assuming this in error is not fatal: in [4] the over-parameterization of self-calibration on real images is criticized; imposing sane values on the parameter, rather than seeking it through minimization may in fact be more robust, with mere percent inaccuracies resulting. An assumption of square pixels (i.e. $\alpha_x = \alpha_y$) is not valid, as while true for many devices, various compression and encoding techniques change the image aspect ratio, and information to correct for this is not always available.

We take the equation from Proposition 3.2 in [9]:

$$P = KRK^{-1}, \tag{3}$$

where P is a unique two-dimensional projectivity (homography) mapping co-ordinates from one image to another from the same camera, and R is some rotation matrix. Correct formation of P is critical to the later extraction of focal lengths, and so a robust technique is required given the likely presence of some erroneous feature matches. We use the normalized Direct Linear Transform inside a RANSAC loop to find the homography with the most inliers, followed by optimizing a homography generated from all inliers using Levenberg-Marquardt, as described in Algorithm 4.6 of [8].

From Equation 3 $R = K^{-1}PK$ and for R to be a rotation matrix, $RR^T = I$. Therefore via substitution and rearrangement let

$$K^{-1}PK(K^{-1}PK)^T - I = D. \tag{4}$$

Due to inaccuracies in P, D can never be precisely zero, but we can optimize over α_x and α_y to find the smallest residual via

$$\operatorname*{arg\,min}_{\alpha_x,\ \alpha_y} \|\operatorname{vec}(D)\|_1, \tag{5}$$

using $\operatorname{vec}(D)$ to turn the matrix D into a vector before calculating a standard vector norm. This method easily allows estimation of α_x and α_y from just a pair of frames from one video, so long as some inter-frame rotation is present. Candidate frame-pairs are selected by requiring an adequate displacement of 2D keypoints from one frame to another.

As described, the zoom (focal length) must be the same in both frames; this is a consequence of using the same K twice in Equation 3. Such a requirement both limits the available choice of frames, and means we must have some way of detecting the absence of zoom-change in order to have any confidence in the estimated parameters.

We remove this limitation by extending the above approach, adding an extra scaling parameter s to describe the zoom-change, so now

$$P = K_1RK_2^{-1} \tag{6}$$

and

$$K_1 = \begin{bmatrix} \alpha_x & & \\ & \alpha_y & \\ & & 1 \end{bmatrix} \quad \text{and} \quad K_2 = \begin{bmatrix} s\alpha_x & & \\ & s\alpha_y & \\ & & 1 \end{bmatrix} \tag{7}$$

(incorporating the previous presumptions on γ, u_0 and v_0).

We can now optimize as before, but over α_x, α_y and s. Naturally, given the introduction of an extra parameter, this extension removes a valid constraint on estimation when the frames are at the same focal length, but results are rarely degraded by this, and allowing the use of frames of differing zoom level means the number of eligible frames is much greater.

Since the α values can vary over time, if a change of zoom level occurs, it is necessary to estimate the α values from the frames whose features will be used in 3D reconstruction — α values estimated at a different zoom level will lead to poor reconstruction. Reconstruction uses the frame with the most filtered feature matches to another frame in a different video; this is then also the reference frame used in α estimation. The reference frame is used pair-wise

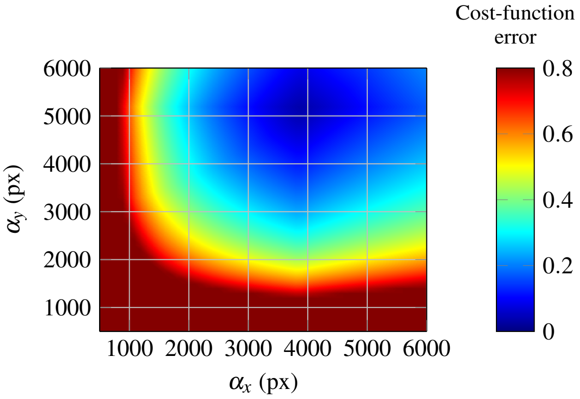


Figure 7: Error-surface for varying α_x and α_y, for scene shown in left half of Figure 6

against as many comparison frames as are suitable in the temporal vicinity of the reference frame. 'Suitable' frames are found by first expanding a window forwards and backwards in time from the reference frame (up to a maximum of ± 1 minute) and performing feature-based motion estimation against frames with a low relative blurriness. Taking the average of all the motion vectors, those frames with a mean feature displacement in x or y greater than 2%, but less than 20%, of the recording's resolution are then flagged as being usable for estimation of α_x or α_y respectively, subject to there being enough feature matches. Before taking the mean focal length of these flagged frames, the upper and lower quartiles of the estimate range are discarded as a crude outlier filter, with the range of the remaining distribution (the interquartile range (IQR)) giving a measure of confidence in the estimate.

In Figure 7 we see the results of evaluating Equation 4, at constant s, over a range of α_x and α_y, using two frames of the left-hand scene of Figure 6 (other temporally local frames being of poor quality or inappropriate displacement). A clearly defined minimum exists around (3900, 5200). Simple optimization (e.g. gradient descent) determines the minimum more precisely at (3871, 5188), s being 1 in this instance. This is therefore a case with a non-square (4:3) pixel aspect ratio (as is common for HD video), and the estimation method has properly permitted the later reconstruction stage to transparently correct for it.

Note that the rotations required to recover these parameters are quite small: in this case the overall rotation expressed by the homography is less than 6°, and the recovered rotation matrix may be decomposed into yaw, pitch and roll rotations of approximately 4.2°, 3.8° and 0.4° respectively.

The focal length estimations are sensitive to rotations between the images used in the homography, and as noted in Agapito et al., reliable *independent* estimation of α_x and α_y depends on having some rotation about the camera's optical axis. If no such rotation is apparent from the 2D keypoint displacement, we must try alternative reference frames from the 'top ten' frames (selected by the subsection 2.2 method) until comparison frames having some small axial rotation are found. While the α values derived will not be directly applicable to the reconstruction, we infer the pixel aspect ratio (PAR) of the video sequence, which will not change over time. Normally this ratio is 1:1, indicating square pixels, but 4:3, as seen above, is not uncommon. Having derived the PAR, estimation of α at the time of the reconstruction frame may be performed with the ratio of α_x to α_y constrained, yielding a viable answer even without axial rotation. Recourse to such extra computation may be avoided if trustworthy metadata for aspect ratio correction are embedded in the video.

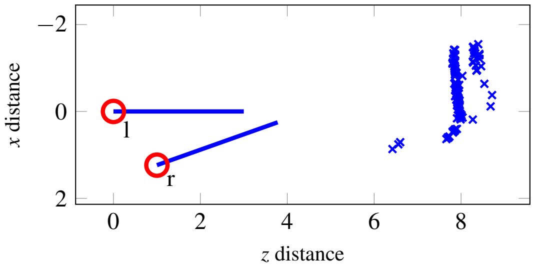


Figure 8: Plan view of reconstruction of world points (×), camera locations (○) and optical axes (—) from frames in Figure 6

2.5 3D reconstruction

Now having intrinsically calibrated cameras, and mostly correct feature matches between viewpoints, we proceed to extrinsic calibration, estimating the rotation and critically the translation between each video sequence's camera. This estimation is confined to the instant captured by the frames whose data are used in reconstruction, but the event situation implies that significant camera translation over time is unlikely. As stated above, reconstruction uses the frames with the most matches between viewpoints; using the static location presumption, two inter-camera matched frame-pairs formed using three cameras may use frames captured at different times by the common camera.

Many methods are available to perform pose estimation, but in our application, where having several viewpoints is likely, methods which optimally calibrate multiple cameras, exploiting the data of all cameras to constrain the overall geometry, are preferable. We employ an adaptation of [12], which, while equivalent to many other methods in the two camera case, can calibrate multiple cameras simultaneously and find least-squares-optimal estimates of the cameras' poses. The adaptation is to run many calibrations sampling different sets of the feature matches in a RANSAC framework, to allow for remaining false inter-viewpoint matches.

Using the features already highlighted in Figure 6, a plan view of reconstructed world points, along with camera poses, is given in Figure 8. Note that since only two cameras are used in this reconstruction it is not constrained by cameras at other viewpoints which typically *improve* the reconstruction.

Considering the world points first, we see the advertising hoardings have been correctly reconstructed as a flat plane, close to perpendicular to one of the cameras' optical axes (the 'l' camera corresponding to the left half of Figure 6). The angled corner hoardings toward the right of the scene have also been correctly reconstructed, angled and nearer to the cameras, while the higher level plane of matches, behind the hoardings, is also correctly positioned.

Considering the camera poses, the camera orientations, given by their optical axes in the plan, are entirely plausible. The x displacement between the cameras is also acceptable, while the distances of the cameras from the world points is broadly correct. The one erroneous item is the relative camera positions in z: with the cameras both in the grandstand the cameras' differing z distances appear too great. This is almost certainly due to some combination of two things:

1. The cameras' principal points not being quite centred in the image.

2. Limitations of the focal length estimation method.

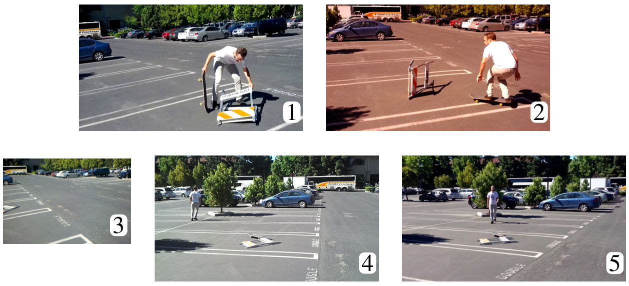


Figure 9: Skateboarding situation from five viewpoints/cameras, with the cameras positioned progressively further to the right

The focal length estimation is very sensitive to the formation of the homography, which is necessarily inexact due to noise in the feature co-ordinates, a rolling shutter capturing motion during the exposure, minor radial distortion in the lens, and the camera motion likely including some small translation. Figure 3 of [6] shows that, on real data, errors in focal length estimation of around 10% are not uncommon. Attenuating the right-most camera's focal length by 10% does indeed result in the two cameras being equidistant in depth from the advertisements.

Following reconstruction, all camera positions, relative to one 'anchor' camera, are known, as desired. The by-product of 3D world points may be of use to subordinate applications: creating a full 3D model of the scene for instance, or automatically synchronizing the videos. Our key application is in providing spatial context to content playback, making possible interfaces like that seen in Figure 1.

3. VERIFICATION OF SYSTEM ON DATA WITH KNOWN GROUND-TRUTH

While the results in Section 2 appear superficially plausible, it is instructive to have some factual basis to claiming the performance of the system. While the frame selection and feature matching methods can be verified empirically, unfortunately the real-world data have no ground truth against which to measure the reconstruction. Hence we present here accuracy results based on our own data for which the true values are known.

In the experiment, a wall of a building was recorded from multiple viewpoints, using a number of cameras, and modest panning (rotational) motions included in the recording.

In Table 1 the estimated focal lengths for each camera are compared to their calibrated values. In all cases α_x was within ± 10 px of α_y, so one figure for α is presented. It can be seen that the errors are generally in keeping with, or better than, the 10% recorded by Agapito et al.

Taking the reconstructed width of the wall as the scaling length, Table 2 compares other reconstructed lengths with their true values. The percentage errors are small, and the absolute errors are well within the tolerance of a few metres required by the application. It is no surprise that the reconstruction with the less-well calibrated cameras has lower accuracy.

4. A MULTI-CAMERA EXAMPLE

Figure 9 shows frames from footage of a person skateboarding in a car park, recorded by five cameras with overlapping but nonidentical views. As numbered, the views can be seen to proceed from camera 1 being left-most, to 5 being right-most. The reconstruction (Figure 10) uses features from the cars, the background coach, the fallen jump obstacle and the surrounding trees.

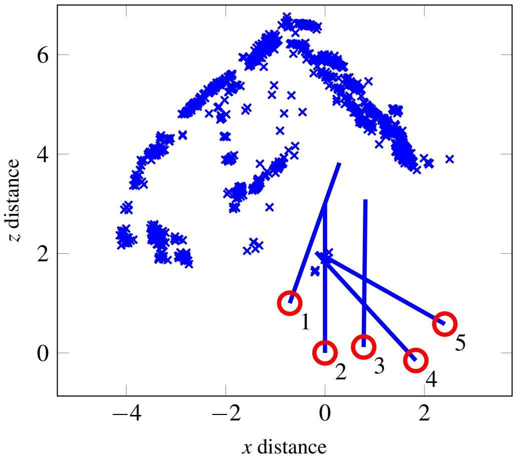


Figure 10: Plan view of reconstruction of world points (×), camera locations (○) and optical axes (—) from frames in Figure 9

As expected from examination of Figure 9 and Figure 10 the nearby car is correctly reproduced near-perpendicularly to camera 4's optical axis, the row of immature trees to the left of the car can clearly be seen along $x = -2$, the background coach appears parallel to the car, and the wall of large trees to the right, seen in the camera 1 image, form a solid line at right-angles to the coach in the reconstruction. Furthermore, and importantly, the cameras are correctly ordered left to right, with entirely visually plausible depths, and the view vectors correspond with the elements of the scene visible in the respective stills.

Perceptually correct (which is the measure that matters for our application) results like this have also been achieved with reconstructions we have formed using user-generated videos from other sporting and musical events, but are not included here for brevity.

5. CONCLUSIONS

In this paper we have presented the design for an implemented system taking unsynchronized, uncalibrated videos of an event, and producing a 3D reconstruction of the scene and the relative camera positions. On user-generated test data captured using various consumer camera-phones and camcorders in a number of differing environments the system presented has been capable of producing perceptually accurate 3D outputs, and on our own validation data the errors in reconstruction have been found to be acceptably low for our application.

The design described above builds on a variety of earlier work, and modifies and extends it where appropriate, forming an innovative combination of stages and yielding a system for an original application. Key features in the design, exploiting the event scenario, are dealing with poor quality frames via a blurriness measure, background feature matching permitting use of unsynchronized videos, and camera self-calibration through camera rotation, using a presumption of no gross camera motion relative to the scene.

The system design presented is still a subject of research, with improvements in robust feature matching and camera calibration, achieved via panoramic stitching methods, expected to improve already satisfactory results further.

Table 1: Comparison of estimated focal length parameters with truth (values rounded to nearest 10 px)

Camera	Calibrated α (px)	Estimated α (px)	Percentage error
Nikon D40 f=19.4mm (A)	2500	2730	9%
Nikon D40 f=28.3mm (B)	3470	3780	9%
Samsung Galaxy S III smartphone (C)	1570	1560	1%
Panasonic HC-V100 camcorder (D)	1790	1880	5%

Table 2: Comparison of reconstructed and actual distances, camera letters reference Table 1

Camera 1	Camera 2	Camera 1 distance to wall (m)			Camera 2 distance to wall (m)			Camera separation (m)		
		Real	Recons.	Error	Real	Recons.	Error	Real	Recons.	Error
A	B	23.2	25.1	8%	23.2	25.3	9%	14.1	16.0	13%
C	D	23.2	22.6	3%	21.6	21.4	1%	16.0	16.5	3%

6. REFERENCES

[1] S. Agarwal, N. Snavely, I. Simon, S. M. Seitz, and R. Szeliski. Building Rome in a day. In *12th International Conference on Computer Vision*, ICCV 2009, pages 72–79, Oct. 2009.

[2] G. Baatz, K. Köser, D. Chen, R. Grzeszczuk, and M. Pollefeys. Handling urban location recognition as a 2D homothetic problem. In *11th European Conference on Computer Vision*, volume VI of *ECCV 2010*, pages 266–279, Sept. 2010.

[3] L. Ballan, G. J. Brostow, J. Puwein, and M. Pollefeys. Unstructured video-based rendering: interactive exploration of casually captured videos. *ACM Transactions on Graphics*, 29(4):87:1–87:11, July 2010.

[4] S. Bougnoux. From projective to Euclidean space under any practical situation, a criticism of self-calibration. In *Sixth International Conference on Computer Vision*, ICCV 1998, pages 790–796, Jan. 1998.

[5] M. Brown, R. I. Hartley, and D. Nistér. Minimal solutions for panoramic stitching. In *IEEE Conference on Computer Vision and Pattern Recognition*, CVPR 2007, pages 1–8, June 2007.

[6] L. de Agapito, R. I. Hartley, and E. Hayman. Linear calibration of a rotating and zooming camera. In *1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, CVPR 1999, pages 15–21, June 1999.

[7] A. Donate and X. Liu. 3D feature extraction from uncalibrated video clips. In *Proceedings of the 2010 ACM Workshop on 3D Video Processing*, 3DVP 2010, pages 31–36, Oct. 2010.

[8] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2003.

[9] R. I. Hartley. Self-calibration of stationary cameras. *International Journal of Computer Vision*, 22(1):5–23, Feb. 1997.

[10] H. Jegou, M. Douze, and C. Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *10th European Conference on Computer Vision*, ECCV 2008, pages 304–317, Oct. 2008.

[11] R. Koch, M. Pollefeys, and L. J. V. Gool. Multi viewpoint stereo from uncalibrated video sequences. In *5th European Conference on Computer Vision*, ECCV 1998, pages 55–71, June 1998.

[12] J. Lasenby and A. X. S. Stevenson. Using geometric algebra for optical motion capture. In E. B. Corrochano and G. Sobczyk, editors, *Geometric Algebra with Applications in Science and Engineering*, pages 147–169. Birkhäuser, 2001.

[13] L. Ling, I. S. Burrent, and E. Cheng. A dense 3D reconstruction approach from uncalibrated video sequences. In *2012 IEEE International Conference on Multimedia and Expo Workshops*, ICMEW 2012, pages 587–592, July 2012.

[14] C. Lipski, C. Linz, K. Berger, A. Sellent, and M. Magnor. Virtual video camera: image-based viewpoint navigation through space and time. *Computer Graphics Forum*, 29(8):2555–2568, Dec. 2010.

[15] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, Nov. 2004.

[16] Y. Matsushita, E. Ofek, X. Tang, and H.-Y. Shum. Full-frame video stabilization. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, CVPR 2005, pages 50–57, June 2005.

[17] M. Pollefeys, R. Koch, and L. Van Gool. Self-calibration and metric reconstruction in spite of varying and unknown internal camera parameters. In *Sixth International Conference on Computer Vision*, ICCV 1998, pages 90–95, Jan. 1998.

[18] M. Pollefeys, D. Nistér, J.-M. Frahm, A. Akbarzadeh, P. Mordohai, B. Clipp, C. Engels, D. Gallup, S.-J. Kim, P. Merrell, C. Salmi, S. Sinha, B. Talton, L. Wang, Q. Yang, H. Stewénius, R. Yang, G. Welch, and H. Towles. Detailed real-time urban 3D reconstruction from video. *International Journal of Computer Vision*, 78(2-3):143–167, July 2008.

[19] M. Pollefeys, L. Van Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, and R. Koch. Visual modeling with a hand-held camera. *International Journal of Computer Vision*, 59(3):207–232, Sept. 2004.

[20] E. Rosten and T. Drummond. Machine learning for high-speed corner detection. In *9th European Conference on Computer Vision*, ECCV 2006, pages 430–443, May 2006.

[21] S. N. Sinha and M. Pollefeys. Camera network calibration and synchronization from silhouettes in archived video. *International Journal of Computer Vision*, 87(3):266–283, May 2010.

[22] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3D. *ACM Transactions on Graphics*, 25(3):835–846, July 2006.

[23] N. Snavely, S. M. Seitz, and R. Szeliski. Modeling the world from internet photo collections. *International Journal of Computer Vision*, 80(2):189–210, Nov. 2008.

[24] J. Tompkin, K. I. Kim, J. Kautz, and C. Theobalt. Videoscapes: exploring sparse, unstructured video collections. *ACM Transactions on Graphics*, 31(4):68:1–68:12, July 2012.

[25] C. Wu. Towards linear-time incremental structure from motion. In *2013 International Conference on 3D Vision*, 3DV 2013, pages 127–134, June 2013.