COMMUNICATION

Software techniques for two- and three-dimensional kinematic measurements of biological and biomimetic systems

To cite this article: Tyson L Hedrick 2008 Bioinspir. Biomim. 3 034001

View the article online for updates and enhancements.

Related content

- Two-step calibration of a stereo camera system for measurements in large volumes
  M Machacek, M Sauter and T Rösgen

- Refraction corrected calibration for aquatic locomotion research: application of Snell's law improves spatial accuracy
  Sebastian Henrion, Cees W Spoor, Remco P M Pieters et al.

- A complete 3D particle tracking algorithm and its applications to the indoor airflow study
  Pascal Henry Biwole, Wei Yan, Yanhui Zhang et al.

Recent citations

- Identification of optimal feedback control rules from micro-quadrotor and insect flight trajectories
  Imraan A. Faruque et al

- Female–female aggression functions in mate defence in an Asian agamid lizard
  Yayong Wu et al

- Multi-camera volumetric PIV for the study of jumping fish
  Leah Mendelson and Alexandra H. Techet
COMMUNICATION

Software techniques for two- and three-dimensional kinematic measurements of biological and biomimetic systems

Tyson L Hedrick

Department of Biology, University of North Carolina at Chapel Hill, CB 3280, Chapel Hill, NC 27599, USA
E-mail: thedrick@bio.unc.edu

Received 30 March 2008
Accepted for publication 22 May 2008
Published 1 July 2008
Online at stacks.iop.org/BB/3/034001

Abstract
Researchers studying aspects of locomotion or movement in biological and biomimetic systems commonly use video or stereo video recordings to quantify the behaviour of the system in question, often with an emphasis on measures of position, velocity and acceleration. However, despite the apparent simplicity of video analysis, it can require substantial investment of time and effort, even when performed with adequate software tools. This paper reviews the underlying principles of video and stereo video analysis as well as its automation and is accompanied by fully functional and freely available software implementation.

Introduction
Researchers studying animal locomotor dynamics at spatial and temporal scales ranging from the motion of bacteria within a drop of fluid (Kogure et al 1998) to the gaits of elephants (Hutchinson et al 2006) use video analysis to measure movement. As techniques and ideas move from one disciple to another, video analysis has also found a home in the analysis of freely behaving biomimetic systems (Long et al 2006). In each case video analysis presents an attractive measurement technique because it requires little to no manipulation or instrumentation of the system or animal to be measured, potentially provides high spatial and temporal resolution, and is easy to implement at laboratory spatial scales. Simple video measurements of position are often extended to measures of velocity, acceleration and even estimation of mechanical power (e.g. Burrows 2003). However, analysis of video data can require significant investment in time and effort even if undertaken with appropriate analysis software and techniques. Furthermore, calculation of positional derivatives can become error prone when attempted without concurrent analysis of measurement uncertainty. A number of different vendors have developed proprietary software packages to facilitate video analysis. Unfortunately, these packages are often targeted at clinical studies of human gait and are a poor fit to the needs of researchers in other disciplines. Furthermore, these are often costly, and cannot be modified to suit the particular needs of individual research laboratories or projects. General-purpose image analysis programs, such as ImageJ (Rasband 1997–2008), an open source image processing tool developed by the US National Institute of Health, can perform simple two-dimensional (2D) measurements on images but are not well
suited to 3D or video operations. Here, I review techniques for video analysis and reconstruction of three-dimensional (3D) trajectories from multiple calibrated video sources with particular application to biological applications and provide a general-purpose MATLAB application suitable for use as a two- or three-dimensional video analysis tool. The source code for the application, DLTdv3, is available (see supplementary material stacks.iop.org/BB/3/034001); the application can be modified with little effort to suit individual projects.

Extraction of positional information from video data incorporates two coupled problems. Firstly, specific objects or landmarks must be identified within the video frames. In this context, objects are entire robots or organisms while landmarks are locations of particular interest on an object such as the centre of a joint or tip of a wing. Depending on the objects or landmarks in question, they may be automatically identified in each frame, manually identified in a few frames and tracked through prior or subsequent frames, or manually identified in all frames. Completely automatic object identification usually only works in carefully controlled situations such as insect flight arenas (Fry et al 2004) and is generally limited to identifying entire organisms rather than specific landmarks on an organism. Secondly, the objects have been identified a reconstruction operation is required to map pixel coordinates \([u, v]\) from one or more cameras to 2D \([x, y]\) or 3D \([x, y, z]\) coordinates. Reconstruction operations require calibrated cameras; calibrations range from a linear scale factor applied to a single camera view to complex multi-camera schemes which resolve 3D coordinates from a combination of two or more camera views while also correcting for optical distortion. Although these two operations, image acquisition and reconstruction, are often intertwined, they are conceptually distinct and will be covered separately below.

Object and landmark tracking

Many video analysis efforts seek to track discrete objects within the field of view of the camera, examining changes in position through time. These objects range in size from entire animals or robots, markers attached to parts of an animal or natural landmarks, such as the eyes or even distinctive colouration patterns. The simplest possible object identification implementation is for a human operator to identify each point of interest in each frame by use of a computer mouse or other input device. This process is usually referred to as digitizing the video. Although generally acceptable and in some cases partially or wholly unavoidable, manual video processing can be very time consuming. Instead, tracking of image landmarks may be automated in software as a dot or cross. The techniques outlined below work equally well on natural colouration patterns and therefore facilitate markerless tracking.

Matching objects from frame \(n\) to \(n + 1\) may be accomplished via two-dimensional cross-correlation, which offers a simple and powerful tool for assessing the alignment of two matrices or two arrays of pixels with respect to one another. Furthermore, the signal-to-noise ratio of the cross-correlation, the maximum value of the output divided by the average value, provides an estimate of the goodness of fit for the alignment (figure 1). Note that cross-correlation techniques assume that the inputs are the result of a stochastic, stationary process and therefore perform best on normalized data, i.e. those with a mean value of zero and no trends across any of the array dimensions. Furthermore, the signal-to-noise ratio of the cross-correlation depends on the signal-to-noise ratio in the original pixel arrays. Therefore, in practice the signal-to-noise ratio that denotes a successful match between frames \(n\) and \(n + 1\) varies somewhat and should be subject to operator modification. Additionally, different portions of
an image typically have different trajectories across frames. Thus, cross-correlation tracking is usually performed on a subsection of the image. Typically, cross-correlation tracking performs best when the array of pixels is approximately twice the size of the object or landmark in question; this allows for normalization and is unlikely to include nearby, similar landmarks. However, use of image subsections leads to a secondary problem: cross-correlation-based tracking cannot function if the pixel array from frame $n + 1$ does not include the landmark in question. This problem can be addressed by increasing the pixel array size, but this may lead to erroneous output if the relative pixel position is not maintained across the entire array. Iterative or multiple-pass cross-correlation algorithms solve this problem by first performing a cross-correlation on a large pixel array and then refining it with successively smaller pixel arrays centred on the offset returned by the prior iteration (Roesgen and Totaro 1995). Alternatively, the location of the landmark in frame $n + 1$ can be predicted from the time series of prior (or subsequent) positions, increasing the likelihood that a small pixel array will suffice.

A number of different predictive tracking algorithms have been applied to the marker tracking problem, notably Kalman (1960) and double exponential filters (e.g. LaViola 2003). Both of these have practical limitations because filter parameters must be estimated from a limited set of prior data. Furthermore, 3D motion projected into the 2D plane captured by the camera may not be well described by a constant set of coefficients. Despite these limitations, adding a predictive algorithm to a cross-correlation-based landmark tracking algorithm can greatly improve performance.

In cases where the landmarks in question are regular shapes such as circular markers, explicit identification of the shapes and their centroids may also facilitate tracking. Object centroids are typically identified by examining the entirety of the pixel array, computing a threshold value and then reducing the pixels to a binary array based on their value with respect to the threshold. The object centroid is then computed from the position of the remaining above (or below) threshold pixels. Classic threshold identification algorithms (e.g. Otsu 1979) function most effectively when using only the image region near the marker, rather than the image as a whole. Therefore, centroid identification functions best when coupled with the techniques described above. All the aforementioned techniques and algorithms are implemented in the DLTdv3 package along with many user-configurable parameters including the cross-correlation pixel array size, predictive tracking algorithm and centroid colour. Additionally, the modular nature of the package allows addition of different predictive or explicit object identification algorithms as needed. For example, in a set of videos acquired under identical conditions, a fixed background threshold level may be more effective than computing a new threshold during each tracking operation.

### Three-dimensional reconstruction

Many applications of photogrammetric or video analysis use the information gathered from two or more cameras to measure movement in three dimensions. A number of different techniques, all falling under the general rubric of camera calibration, facilitate this process. Note that calibration typically refers to the process of creating a camera model while reconstruction refers to using the camera model and known pixel coordinates to reconstruct a 3D location. Camera calibration is an active area of research within the computer vision community, with most efforts devoted towards rapid self-calibration of off-the-shelf cameras with little spatial separation between the cameras (e.g. Heikkila and Silven 1997, Zhang 2000). These algorithms often prove difficult to implement in cases with widely dispersed cameras; in such cases, it is common to rely on older algorithms using 3D reference objects often referred to as calibration cubes (Abdel-Aziz and Karara 1971, Tsai 1987). The older of these techniques, direct linear transformation (DLT), has become well established in the biomechanics community and continues in widespread use due to its flexibility with respect to camera placement, acceptable accuracy of reconstruction and quantity of earlier literature establishing best practices in experimental application (Wood and Marshall 1986, Challis and Kerwin 1992, Chen et al 1994). Moreover, DLT has been extended and applied to many different imaging applications ranging from calibration through materials with differing refractive indices (Kwon and Casebolt 2006) to stereoradiography (Choo and Oxland 2003). Although DLT is relatively simple to implement and well understood, it does not offer the highest possible reconstruction accuracy because it does not take nonlinear lens distortions into account and assumes zero error in the calibration cube. More advanced techniques such as bundle adjustment improve on the accuracy at some cost in implementation difficulty and robusticity to noisy input data but may be desirable in situations where the highest accuracy is required. Researchers interested in implementing these techniques should consult a modern computer vision textbook such as Hartley and Zisserman (2003). Conversion between different camera calibration algorithms is possible but, because of the diversity of calibration systems and algorithms, a complete set of conversions has not been developed.

Calibration of cameras with DLT requires an object with a minimum of six non-coplanar control points in view of each camera. However, many more control points are often used because (1) additional points allow for an overdetermined solution to the camera calibration coefficients, improving accuracy via least squares minimization, and (2) for accurate reconstruction, control points should be distributed throughout the volume of interest. Each control point need not be in view of all cameras. In some cases it is possible to develop automated methods for identification of calibration points in the camera image, but no method, both general and automated, exists to cover all situations. Therefore, the DLTdv3 package includes a specialized sub-program (DLTcal3) for manual identification of control points and automated identification of outliers. Users with repeated regular calibration tasks may choose to automate control point identification through custom software or implementations of other published methods such as the laser pointer candle technique of Svoboda et al (2005). Calibrations are sensitive to changes in any camera
optical parameters including position, orientation, focus, zoom, aperture width and even the wavelength of light used to illuminate the scene. Calibrations are not sensitive to changes to non-optical parameters such as the exposure duration or sensor gain.

**Tracking in calibrated cameras**

Calibrated cameras make available some additional region and marker tracking options because identification of the landmark of interest in one camera of a calibrated camera set defines an epipolar plane and therefore provides information on where the landmark must lie in the other cameras (Hartley and Zisserman 2003). This information can be used to constrain the search for the object in these other views. Alternatively, the constraint can be displayed to a human operator and used to identify the object in cases of slightly obscured markers or sets of markers with similar appearance (figure 2). These relationships have also been used in more specialized applications such as tracking the arm motion of freely behaving octopus (Yekutieli et al 2007). Both tracking with a constraint based on additional information from other cameras and visualization of that information are supported in DLTdv3. Additionally, use of three-dimensional trajectories may allow more accurate predictive tracking (e.g. Black and Ellis 2006).

**Measurement error**

The error in three-dimensional reconstructions arises from error in the individual camera calibrations and from disagreement between the pixel coordinates of markers identified in different camera views. These may be the result of human error, intrinsic limitations in the physical devices or in the assumptions used to model the cameras. Errors in the calibrations may arise due to nonlinear lens distortion not accounted for in the particular calibration implementation, inaccuracies in the control point identification process or inaccuracies in the control point measurements. Errors in calibration can be measured during the calibration process but also appear as reconstruction errors. Reconstruction errors occur when there is no exact solution \([x, y, z]\) for the set of camera calibration coefficients and recorded camera pixel coordinates \([u, v]\). This situation may arise from errors in the recorded pixel coordinates (i.e. human or tracking error) or in the camera calibration coefficients. Although errors may be minimized by use of appropriate calibration and digitization methodologies, all reconstructed points have some error or uncertainty associated with them. This error should be measured and used to create confidence intervals for measurements and in statistical filtering techniques.

Because reconstruction errors arise from inconsistencies in the pixel coordinates used to compute a 3D location, both camera calibration and reconstruction errors are typically measured in pixels rather than dimensional coordinates. For example, the reconstruction error or residual reported by DLTdv3 is the root mean square difference between the input pixel coordinates and the ideal pixel coordinates, or those that would result in the same 3D location but with no reconstruction error. Note that the ideal pixel coordinates may fall outside the boundaries of the image and may not be physically realizable. Conversion of these quantities to dimensional measures appropriate for publication and for estimating the error of derived quantities such as velocity or rigid body orientation requires several additional steps. The relationship between the dimensional error and pixel error varies among the different dimensions as well as among the cameras used in the reconstruction and the location of the point in question. Given these complex interactions, dimensional errors are best measured with bootstrap or Monte Carlo techniques (Zar 1999) that introduce additional error of the appropriate magnitude in the camera pixel coordinates.
Repeatedly perturbing the original \([u, v]\) camera coordinates with values randomly drawn from a normal distribution with a mean value centred on the known camera pixel error results in a distribution of 3D coordinate values. This distribution of 3D coordinates can then be used to establish confidence intervals for the coordinates, for instance by finding the range of values for a particular coordinate at a particular instant in time that contains 95% of the perturbed coordinates. This information may then be used to display confidence intervals for the original 3D coordinates or used with a smoothing spline to find a sequence of points that minimizes higher derivatives of the signal while remaining within the confidence intervals, allowing conservative measurement of higher derivatives. The DLTdv3 package implements both of these strategies and generates 95% confidence intervals when used on any MATLAB platform (version 7 or later) and spline-smoothed data on platforms with the spline toolbox installed. Note that in cases where the epipolar line was used to identify markers in one or more of the cameras, it is desirable to add additional pixel error to the bootstrap analysis beyond that indicated by the reconstruction residuals themselves. This is because the epipolar line represents a line of zero reconstruction error, so choosing points along it effectively propagates error from the first camera into the second camera without generating any reconstruction uncertainty.

**Software implementation**

The video analysis and calibration techniques described here have been implemented in a freely available MATLAB (version 7 or later) application, DLTdv3, available for download as supplementary material to this publication. Extensive documentation including video tutorials and sample data are available from the author’s website. In addition to implementing the methods described above, DLTdv3 has an easy-to-use graphical interface exposing these functions, is designed to exchange data freely with data analysis packages and compensates for synchronization errors in multicamera recording. DLTdv3 is designed as a general-purpose application, useful in many different video analysis tasks but not specialized for any. Source code is available and project-specific extensions may be added as necessary.

**Future work**

Given the current degree of interest in computer vision applications, improvements in the algorithms available for camera calibration, landmark tracking and object identification are expected. As these advances are refined to the point where they may be easily applied by non-specialist users, they may be incorporated into the DLTdv3 application and made available for download. Additionally, although implementation in MATLAB facilitates modification by many researchers in the biological and engineering sciences due to widespread expertise in that platform, MATLAB itself is costly. Reimplementation of DLTdv3 in a wholly free and open source programming environment such as Python would overcome this disadvantage but might also reduce the ability of individual research groups to customize the software for their own needs.

**References**

Abdel-Aziz Y I and Karara H M 1971 Direct linear transformation into object space coordinates in close-range photogrammetry *Proc. Symp. on Close-Range Photogrammetry* (University of Illinois at Urbana-Champaign)


Challis J H and Kerwin D G 1992 Accuracy assessment and control point configuration when using the DLT for photogrammetry *J. Biomech.* 25 1053–8

Chen L, Armstrong C W and Raftopoulos D D 1994 An investigation on the accuracy of 3-dimensional space reconstruction using the direct linear transformation technique *J. Biomech.* 27 493–500

Choo A M T and Oxland T R 2003 Improved RSA accuracy with DLT and balanced calibration marker distributions with an assessment of initial-calibration *J. Biomech.* 36 259–64


Kalman R E 1960 A new approach to linear filtering and prediction problems *Trans. AMSE D* 82 35–45

Kogure K, Ikemoto E and Morisaki H 1998 Attachment of Vibrio alginolyticus to glass surfaces is dependent on swimming speed *J. Bacteriol.* 180 932–7

Kwon Y H and Casebolt J B 2006 Effects of light refraction on the accuracy of camera calibration and reconstruction in underwater motion analysis *Sports Biomech.* 5 95–120


Roessgen T and Totaro R 1995 Two-dimensional on-line particle imaging velocimetry *Exp. Fluids* 19 188–93


