

## METHODS &amp; TECHNIQUES

## Validation of XMALab software for marker-based XROMM

Benjamin J. Knörlein<sup>1,\*</sup>, David B. Baier<sup>2,3</sup>, Stephen M. Gatesy<sup>3</sup>, J. D. Laurence-Chasen<sup>3</sup> and Elizabeth L. Brainerd<sup>3</sup>

## ABSTRACT

Marker-based XROMM requires software tools for: (1) correcting fluoroscope distortion; (2) calibrating X-ray cameras; (3) tracking radio-opaque markers; and (4) calculating rigid body motion. In this paper we describe and validate XMALab, a new open-source software package for marker-based XROMM (C++ source and compiled versions on Bitbucket). Most marker-based XROMM studies to date have used XrayProject in MATLAB. XrayProject can produce results with excellent accuracy and precision, but it is somewhat cumbersome to use and requires a MATLAB license. We have designed XMALab to accelerate the XROMM process and to make it more accessible to new users. Features include the four XROMM steps (listed above) in one cohesive user interface, real-time plot windows for detecting errors, and integration with an online data management system, XMAPortal. Accuracy and precision of XMALab when tracking markers in a machined object are  $\pm 0.010$  and  $\pm 0.043$  mm, respectively. Mean precision for nine users tracking markers in a tutorial dataset of minipig feeding was  $\pm 0.062$  mm in XMALab and  $\pm 0.14$  mm in XrayProject. Reproducibility of 3D point locations across nine users was 10-fold greater in XMALab than in XrayProject, and six degree-of-freedom bone motions calculated with a joint coordinate system were 3- to 6-fold more reproducible in XMALab. XMALab is also suitable for tracking white or black markers in standard light videos with optional checkerboard calibration. We expect XMALab to increase both the quality and quantity of animal motion data available for comparative biomechanics research.

**KEY WORDS:** Biomechanics, Kinematics, Motion analysis, X-ray, Camera calibration, Marker tracking, Precision, Accuracy

## INTRODUCTION

X-ray reconstruction of moving morphology (XROMM) is a set of methods for combining static bone shape data from a CT scan with bone motion data from X-ray videos. The result is a precise and accurate XROMM animation of 3D bone meshes moving in 3D space (Brainerd et al., 2010; Gatesy et al., 2010). Over the past 8 years, researchers have used XROMM to study *in vivo* skeletal motion in numerous behaviors and species including: terrestrial locomotion of alligators (Baier and Gatesy, 2013), dogs (Wachs et al., 2016), rats (Bonnan et al., 2016) and birds (Kambic et al., 2014, 2015); arboreal locomotion of sloths (Nyakatura and Fischer, 2010); jumping in frogs (Astley and Roberts, 2012) and humans (Miranda et al., 2013); feeding in pigs (Menegaz et al., 2015), ducks (Dawson et al., 2011), geckos (Montuelle and Williams, 2015) and

fish (Camp and Brainerd, 2015; Gidmark et al., 2012); flight in bats (Konow et al., 2015) and birds (Baier et al., 2013; Heers et al., 2016; Heers and Dial, 2012); lung ventilation in iguanas (Brainerd et al., 2015); and trackway formation in theropod dinosaurs (Falkingham and Gatesy, 2014). Methods include marker-based XROMM, in which radio-opaque markers are surgically implanted into skeletal elements (Brainerd et al., 2010; Tashman and Anderst, 2003), and markerless XROMM, which includes manual scientific roscoping (Gatesy et al., 2010) and semi-automated bone model registration methods (Banks and Hodge, 1996; Miranda et al., 2011; You et al., 2001).

Marker-based XROMM requires specialized software for correcting distortion introduced by fluoroscopic image intensifiers, calibrating cameras, tracking radio-opaque markers, and calculating rigid body motion from a set of three or more markers implanted in each rigid skeletal element (Brainerd et al., 2010). Here, we introduce and validate XMALab (X-ray motion analysis lab), a new open-source software package for marker-based XROMM. XMALab is designed for XROMM, but works equally well for motion analysis from standard video cameras. Prior software for marker-based XROMM, XrayProject (Brainerd et al., 2010), was developed in MATLAB and based on DLTdv3 (Hedrick, 2008). XMALab is written in C++; compiled versions for Mac OS and Windows and source code are available on Bitbucket ([bitbucket.org/xromm/xmalab](http://bitbucket.org/xromm/xmalab)). XMALab is fully open-source (offered under GNU General Public License version 3) and does not require MATLAB or any proprietary software to run.

The XrayProject software package produces results with high precision and accuracy for marker tracking (Brainerd et al., 2010), but is somewhat cumbersome to use, particularly for new users.

Our primary goal in the design of XMALab has been to improve the user experience and thereby make marker-based XROMM more efficient and accessible to a broader community of researchers. The correct use of XrayProject requires substantial expertise, and users can easily make mistakes that either form a barrier to finishing the XROMM process or introduce more subtle errors that may be hard to detect. To avoid such pitfalls and improve the quality of data produced by all users, XMALab provides the user with clear visual feedback on errors at all stages of data processing.

Another challenge with XROMM is file management, as XROMM requires at least seven files as inputs (raw data): two images for distortion correction, two images for calibration, a calibration frame specification text file and two X-ray movie files. Then, the XrayProject workflow produces 22 files as outputs: four images, two MATLAB files and 16 text files for every trial in an experiment. Thus, XrayProject requires or produces a total of 29 files per trial and 725 files for a small study with five individual animals and five trials per individual.

To solve this data management problem, we have developed an open-source, online database, XMAPortal ([xmaportal.org](http://xmaportal.org)), for managing calibration images and X-ray video files. XMALab is integrated with XMAPortal to facilitate data management and data

<sup>1</sup>Center for Computation and Visualization, Brown University, Providence, RI 02912, USA. <sup>2</sup>Department of Biology, Providence College, Providence, RI 02912, USA.

<sup>3</sup>Department of Ecology and Evolutionary Biology, Brown University, Providence, RI 02912, USA.

\*Author for correspondence ([benjamin\\_knorlein@brown.edu](mailto:benjamin_knorlein@brown.edu))

transfer. Calibration trials and sets of X-ray movie files for a given trial can be exported from XMAPortal and imported into XMALab. Then, most importantly, XMALab packages all data into a single file in our new .xma format. To keep the .xma files from becoming too large, paths to the movie files are saved, rather than the actual movies. All other data, including the images used for calibration and undistortion and data-specific settings are saved directly within the .xma file. Our goal with XMAPortal and XMALab is to facilitate data exchange, troubleshooting and reproducibility by maintaining a record of all data analysis and software settings. Any user can open an .xma file and see exactly how the data were processed to help with troubleshooting or to reproduce a prior result.

The purpose of this paper is to validate XMALab for marker-based XROMM, to measure its precision and accuracy, and to compare XMALab with XrayProject for precision and reproducibility. The use of complex, open-source software for research in biology can be problematic because many biologists do not have the programming skills to review and validate the code themselves (Deng, 2015). Similar to the process for approving new drugs and therapies in medicine, new scientific software should be shown to be safe (for producing correct results), effective and an improvement over existing tools. In this paper we: (1) use objects with known spacing between steel spheres (X-ray phantoms) to measure the precision and accuracy of marker tracking in XMALab; (2) compare XMALab with XrayProject for precision and reproducibility; and (3) use a frozen-specimen test (Menegaz et al., 2015) to measure the precision of measurements made using an anatomically meaningful coordinate system (i.e. a joint coordinate system).

## MATERIALS AND METHODS

### User interface

Here, we describe the user interface and main features of XMALab software as seen from the perspective of a user. The technical details of the algorithms employed for undistortion, calibration and marker tracking are described after this section.

The XMALab user interface contains three workspaces: Undistortion, Calibration and Marker tracking. In the Undistortion workspace (Fig. 1A), XMALab calculates the transform required to correct the complex distortion patterns introduced by fluoroscopic image intensifiers (Wang and Blackburn, 2000). In the Calibration workspace (Fig. 1B), users load in one or more sets of calibration images for two or more cameras. As in XrayProject, camera calibration in XMALab is based on images of a 3D calibration object, but identification of the calibration points is semi-automatic in XMALab. The user identifies only four locations on the 3D object, which can be marked with lead solder formed into uniquely identifiable geometric shapes (Fig. 1B; Supplementary Information 1). Then, XMALab detects all visible calibration points and provides the user with feedback on goodness of fit (error) of all detected points relative to the known geometry of the object. The user can then manually include or exclude calibration points and refine the locations of points if required. Calibration with checkerboard images for standard light cameras is also supported in XMALab.

The Marker tracking workspace provides a Toolbox (Fig. 1Ci) for automatically tracking one or all points. Specific points can be named and selected in the Points window (Fig. 1Cii). For each point, users can select black marker detection mode (default X-ray marker) or white marker mode for X-ray negative images or standard light cameras with white markers. Selecting a point makes the point active in the Camera views (Fig. 1Ciii), Detailed View window (Fig. 1Civ), and Plot window (Fig. 1Cv). The Points window is also used to group sets of markers located in the individual bones into

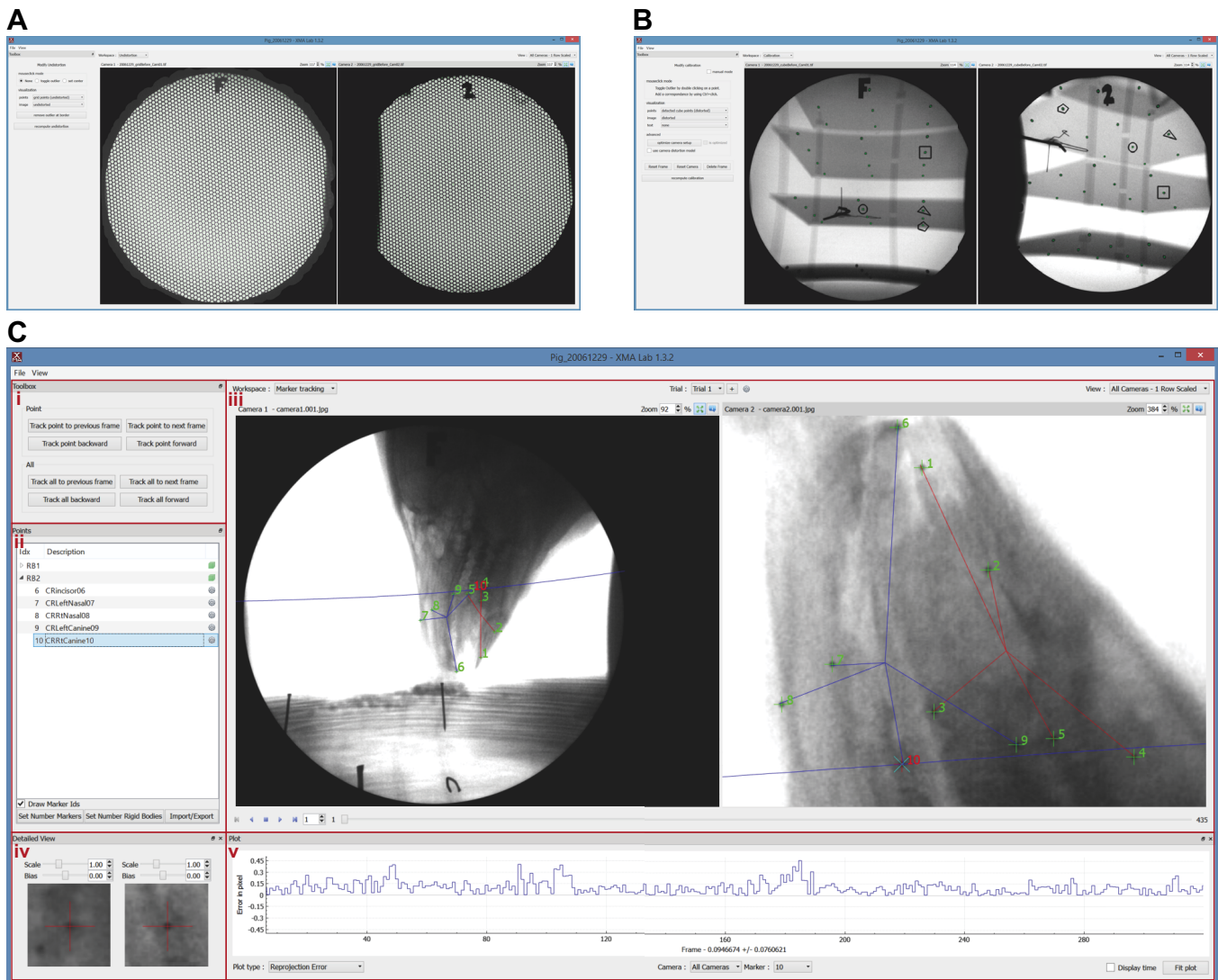
rigid bodies. Marker *xyz* coordinates from a CT scan can be imported, and transformations of the rigid bodies calculated for animating mesh models of the skeletal elements in Autodesk Maya (Brainerd et al., 2010). Navigation and interaction in the Camera view windows (Fig. 1Ciii) is simple and intuitive. Points can be set, moved or selected with the left mouse button and modifier keys, the mouse wheel zooms to the cursor position, and the right mouse button is used for panning. The Detailed View window (Fig. 1Civ) shows a magnified view of the selected point and permits manual refinement of its position. Manual refinement is necessary when automatic tracking fails, particularly when radio-opaque markers intersect or overlap with other radio-opaque objects in the scene.

The Plot window (Fig. 1Cv) offers six plot types: 2D positions, 3D positions, marker to marker distance, reprojection error, rigid body transformations and rigid body error (Fig. 2). Clicking on a frame in the plot window moves the Camera view and Detailed View windows to that frame, and shift-dragging over data and pressing the delete key removes tracked points from the selected markers and frames. The Plot window updates automatically as points are tracked or moved, providing instant feedback on the extent and quality of tracking. The Plot window also updates if the user goes back to fine-tune the undistortion or calibration, providing feedback on the effects of changes.

Erroneous point positions can be identified in the reprojection error plot (Fig. 2A), which shows the mean error in pixels of the calculated 3D point reprojected back onto the 2D image planes. Errors can also be detected in the marker-to-marker distance plot (Fig. 2B) of the distance between a pair of markers. This distance should be constant for markers implanted in the same rigid skeletal element, so co-osseous points with particularly large deviations reflect poorly tracked frames. Once rigid bodies are defined in the Points window, the rigid body error plot will also permit detection of problematic frames (Fig. 2C). XMALab includes Butterworth filtering of rigid body transformations with user control of frame rate and cutoff frequency. The filtered and unfiltered rigid body transformations can be viewed in the Plot window to detect over-filtering. Over-filtering can also be detected in the rigid body error plot as the goodness of fit of the rigid CT point geometry to the actual 3D points will decrease substantially with over-filtering.

Filtered and unfiltered rigid body transformations can be exported for animation in Autodesk Maya. XMALab also supports export of 2D points, 3D points and marker to marker distances. As in XrayProject, 2D points are tracked on distorted images and the undistortion transform applied to the 2D points (Brainerd et al., 2010). Hence, the same 2D points can be used to calculate 3D points with different sets of undistortion and calibration images. In XMALab, the 2D points for a trial and links to trial images (the X-ray movies) can be imported from another .xma file, and the 2D points from an XrayProject dataset can be imported. Rounding out export and import functions, files for placing virtual X-ray cameras in Autodesk Maya and undistorted trial images can be exported for Scientific Rotoscoping (Gatesy et al., 2010) and for visual validation of model registration in Maya.

As described in the Introduction, XMALab facilitates data management by coordination with the online XMAPortal data management system. XMALab saves all undistortion and calibration images, calibration data, links to X-ray movies or images, tracked points and data-specific settings as one file with the extension .xma. Sets of undistortion and calibration images can be selected on XMAPortal and downloaded as an .xma file. Metadata for the files are also automatically exported from the XMAPortal and made visible in XMALab.



**Fig. 1. Workspaces and user interfaces in XMA Lab.** (A) Undistortion. (B) Calibration. The calibration data shown here are the data from the XMA Lab pig feeding tutorial (linked on bitbucket.org/xromm/xmalab/). The four shapes (rectangle, circle, triangle, pentagon) shown in B were digitally added to the images for the tutorial. In a real project, geometric outlines formed from lead solder would be attached to the calibration object to identify specific reference locations. (C) Marker tracking. (i) Toolbox. (ii) Points window. (iii) Main video and automated point tracking window. (iv) Detailed View window for manual refinement of tracking. (v) Plot window showing reprojection error in pixels for marker 10. Four other plot types are available: 2D position, 3D position, marker-to-marker (intermarker) distance, rigid body motion and rigid body error (Fig. 2).

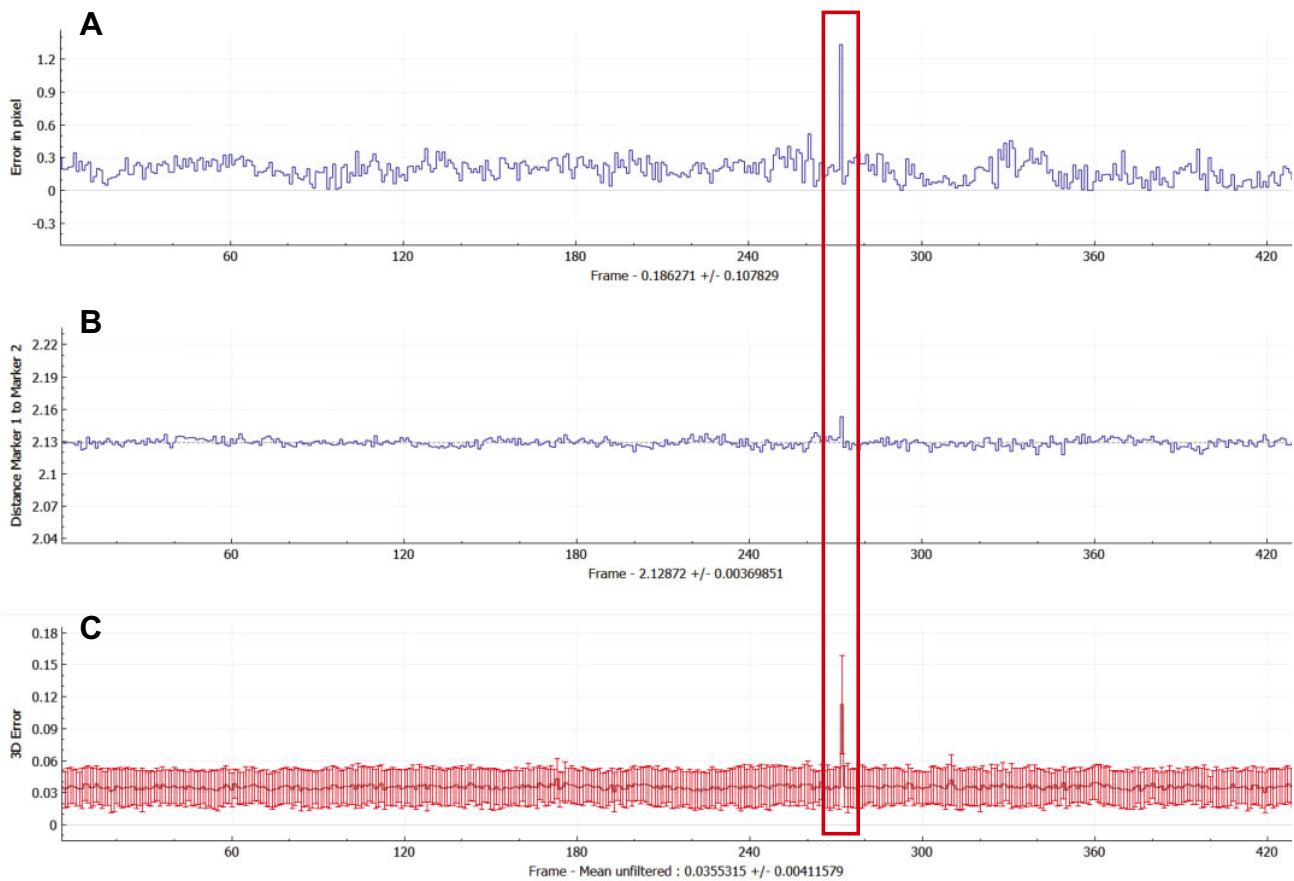
## Undistortion

Fluoroscopic videos are subject to complex image distortion (Verdonck et al., 1999). XMA Lab follows the same approach as already described (Brainerd et al., 2010) and corrects distortion by using local weighted mean (LWM) transformation (Goshtasby, 1988).

As input for the undistortion process (Fig. 1A), perforated metal sheets with a hexagonal pattern (McMaster-Carr, part number 9255T645) are mounted on the faces of the image intensifiers (Brainerd et al., 2010). XMA Lab automatically detects the holes by using a blob detection algorithm, which identifies connected components and filters the result based on appearance, e.g. size, circularity, convexity or inertia ratio (Bradski, 2000). After the blobs have been identified, a correspondence between distorted (detected) reference positions and undistorted position based on the structure of the hexagonal grid can be established. Based on these correspondences, the undistortion as well as the distortion

transformation can be computed by using the LWM algorithm (Goshtasby, 1988).

However, the transformations from distorted to undistorted or vice versa do not perfectly align when only the correspondences are swapped. This can result in differences between an original point and its position if it is distorted and consecutively undistorted or vice versa. In XMA Lab, point coordinates have to be transformed from distorted to undistorted image space, but also from undistorted to distorted, e.g. during rendering of epipolar lines. In this case, the 2D coordinate has to be undistorted and its epipolar line in the other camera computed, followed by a re-distortion for projection onto the target image. Even if the error introduced is small, we further addressed this problem by optimizing the position of transformed points, so that its distorted–undistorted and undistorted–distorted position always aligns with its original position. To do so, we distort points by using the LWM transformation as defined. However, when a point



**Fig. 2. The plot window facilitates error checking in XMA Lab.** Three plot-type options for the plot window (Fig. 1Cv) are shown. (A) Marker reprojection error in pixels for one marker. The mean reprojection error of 0.186 pixels with s.d. of 0.108 pixels is shown below the plot. (B) Intermarker distance between the marker in A and another marker in the same rigid body, with mean and s.d. below the plot (in the same units as for the calibration object specification file; in this case, cm). (C) Rigid body error in the same units as for the calibration object specification file (in this case, cm). The red box shows how error in tracking one marker can be detected in intermarker distance and rigid body error. These plots facilitate efficient error checking.

is undistorted, we first undistort the point using the LWM transformation from distorted to undistorted space. Then, we distort the computed point and calculate the distance between the original distorted coordinate and the undistorted–distorted point. As mentioned, these points usually do not perfectly match, so we add half the distance to the distorted position and recompute its undistorted position. This step is repeated until the undistorted–distorted coordinate matches with the original position in distorted image space below a threshold of 0.000001 pixels.

### Calibration

A projection from world coordinates  $(x, y, z)$  to image coordinates  $(u, v)$  can be expressed by using linear algebra as a multiplication in projective coordinates with the  $3 \times 4$  matrix  $\mathbf{P}$ :

$$\begin{pmatrix} u \cdot w \\ v \cdot w \\ w \end{pmatrix} = \mathbf{P} \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}. \quad (1)$$

In XMA Lab, the projection is performed by using a pinhole camera model described by:

$$\mathbf{P} = \mathbf{K} \cdot (\mathbf{R}|\mathbf{t}), \quad (2)$$

with the camera matrix  $\mathbf{K}$  containing the intrinsic parameters of the camera, the focal lengths  $f_x, f_y$ , and camera center  $c_x, c_y$ :

$$\mathbf{K} = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix}, \quad (3)$$

and the extrinsic parameters  $(\mathbf{R}|\mathbf{t})$ , which rotate and translate a point into the coordinate system of a camera.

To compute the intrinsic as well as the extrinsic camera parameters, the 2D image positions of the markers of a calibration object have to be known. We first detect all 2D locations in the image resembling a marker by using the same blob detection as during the undistortion step. As we do not know which 2D image position belongs to which marker, correspondences have to be established. To speed up this process, we developed a semi-automatic approach, which only requires the user to select the approximate location of four reference points (Fig. 1B). This provides four correspondences between 3D and 2D locations. However, as the projection  $\mathbf{P}$  has 11 degrees of freedom and its computation requires at least six correspondences, we use a random sampling consensus (Fischler and Bolles, 1981) in which we add two additional correspondences randomly selected from all possible combinations of detected 2D image locations and 3D locations of the calibration object. We then compute the projection  $\mathbf{P}$  based on

the six correspondences by using a direct linear transformation and project all 3D locations of markers back onto the image. These projected locations are then used to identify additional correspondences. The process is repeated 100,000 times. Finally, the largest set of 2D–3D correspondences is stored and users can manually add, remove or refine correspondences.

After the set of 2D–3D correspondences has been identified, the intrinsic and extrinsic camera parameters are computed using the OpenCV camera calibration (Bradski, 2000) based on Zhang (2000) and the Camera Calibration Toolbox for Matlab by Bouget ([https://www.vision.caltech.edu/bougetj/calib\\_doc/](https://www.vision.caltech.edu/bougetj/calib_doc/)). An initial guess on intrinsic and extrinsic parameters is acquired by computing the projection  $\mathbf{P}$  using direct linear transformation followed by a singular-values decomposition into intrinsic and extrinsic parameters.

XMALab supports calibrations using multiple images with different poses of the calibration object to improve calibration results. For this case, several optimizations have been made for the calibration as well as for the identification of 2D–3D correspondences. Intrinsic parameters for a camera are already computed when correspondences need to be established for additional calibration images. We therefore only use five correspondences during the randomized process and only compute the pose of the calibration object relative to the camera and not the projection matrix of the camera. This also makes it possible to calibrate images if only three of the references are visible with the addition of two randomized samples.

Furthermore, if the relationships between cameras have been established from one calibration object pose, then for any subsequent poses the correspondences are established entirely automatically if they are detected in only one camera. This not only leads to a reduction in user interaction when using additional calibration images but also permits the placement of the calibration object without the requirement that marked references are visible in all calibration images. The position and orientation of the calibration object implicitly define the coordinate system in which animated bone motions are reconstructed. The use and processing of additional images therefore makes it possible to define the coordinate system of the recovered data even in unwieldy experimental setups.

The additional images are then also used for improving calibration during a non-linear optimization of the camera parameters. The OpenCV camera calibration does not take into account that the transformation between cameras is fixed for all poses of the calibration object. The transformation between cameras can be computed by using their relative transformation to the pose of the calibration object, but this results in transformations that are slightly different for each pose of the calibration object. XMALab therefore optimizes the complete camera setup in a final non-linear optimization in which the optimization takes into account that transformations between cameras are the same for all images and that only the calibration object has been moved.

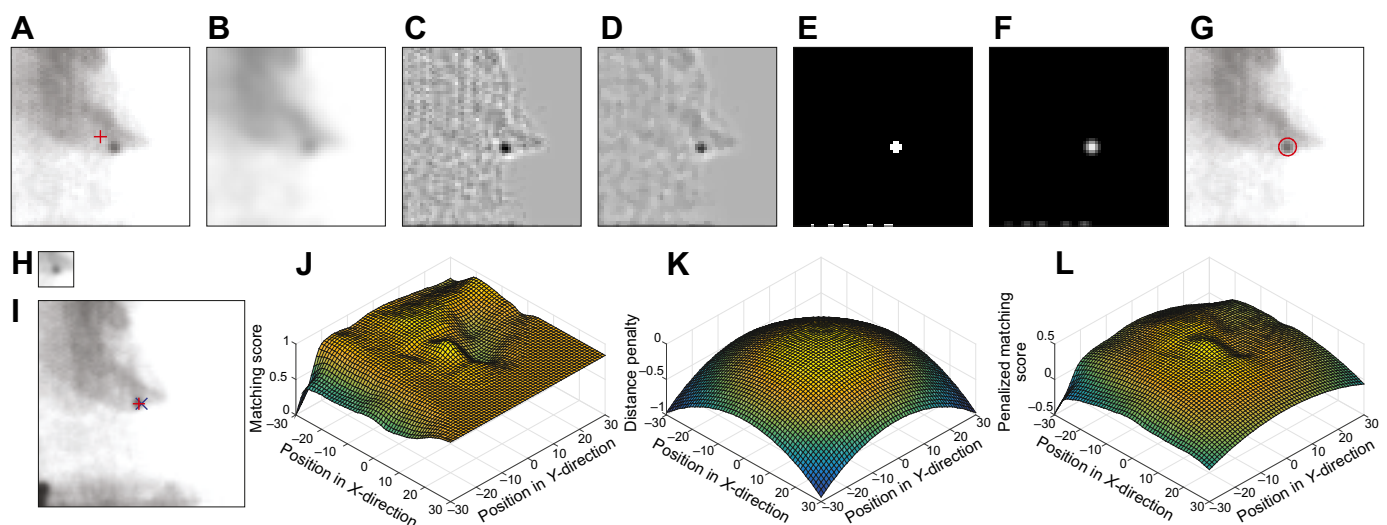
### Marker detection and tracking

During marker identification in XMALab, the user first selects a marker with a click into the image. The user-defined position does not have to be highly accurate as XMALab further optimizes it by examining a region of interest around it (Fig. 3A). To remove variations in the background, a blur is applied to the region of interest (Fig. 3B), subtracted from the original image (Fig. 3C) and filtered using a median filter (Fig. 3D). Then, the region of interest is normalized into the range  $[0,1]$  and converted into a binary image (Fig. 3E) using a heuristic threshold based on the intensity of the clicked position  $s$  and a user-defined offset  $o$ :

$$\text{Threshold} = 0.5 \cdot s + o. \quad (4)$$

The offset  $o$  is set by default to 0.08 and usually does not need to be modified. Finally, the binary image is blurred using a Gaussian kernel (Fig. 3F) and the marker position is determined as the center of the circle enclosing the contour closest to the clicked position (Fig. 3G). We also store the radius of the enclosing circle to improve further steps by adjusting the size of the region of interest during tracking and marker refinement.

For tracking markers automatically through time, XMALab uses template matching combined with a penalty score based on the



**Fig. 3. Marker detection and marker tracking.** (A–G) Marker detection. (A) Region of interest. The cross depicts the position clicked by the user. (B) The blurred region of interest. (C) Difference between the original and the blurred region of interest. (D) Median filtered difference. (E) Thresholded binary image. (F) Blurred thresholded image. (G) Detected enclosing circle of the marker. (H–L) Marker tracking. (H) Template used for the marker-tracking process. (I) Search region of the marker; the blue cross depicts the predicted position while the red cross depicts the tracked marker position. (J) The matching score of the template with the positions in the search region. (K) The penalty score based on the distance to the predicted position. (L) The penalized matching score.

distance to the predicted position of the marker. First, we read the template from the previous frame (Fig. 3H) and the position of the marker is predicted based on the position, velocity and acceleration in the previous frames (Fig. 3I). Note that the previously computed size of the markers is used to determine the size of the template to reduce the influence of the background. To find the best match of the template in the search area, we compute the normalized cross-correlation NCC between the template and positions in a search area surrounding the predicted position normalized in the range of 0 to 1 (Fig. 3J). The correlation does not take the distance to the predicted position into account and if multiple markers occur in the search region, incorrect matches may occur. We therefore add a negative score to the normalized cross-correlation, penalizing the distance to the predicted position (Fig. 3K). The penalty PEN is computed by using a 2D Gaussian function scaled to the range between 0 and 1 with a s.d. of 1.5 times the search region size. To modify the influence of the penalty, a weight  $w$  can be changed by the user for cases where the prediction lies too far from the actual position, e.g. in the case of very fast motions or low frame rates. The final matching score  $s$  (Fig. 3L) is then given by:

$$s = (1.0 - w) \cdot \text{NCC} + w \cdot \text{PEN}. \quad (5)$$

By default, a penalty weight of 0.5 equally weighing the normalized cross-correlation and the penalty is set, and in practice does not have to be modified by a user. Finally, as the template matching only computes the location of a marker in whole-pixel coordinates, which can result in a drift of the position when tracked over long sequences, we use the same marker detection as during selection by a user (Fig. 3A–G) with the tracked position as input. This computes the marker position to subpixel coordinates and also computes the radius of the marker for the tracked frame.

Even while being highly reliable, the detection of the position of a marker based on the enclosing circle is limited in its accuracy. Most markers are only 2–6 pixels wide and the detection does not directly take the intensity value of pixels into account. Therefore, the computed position of a marker is strongly affected by the discrete sampling of the image. To improve the sub-pixel accuracy of marker detection, we have added an optional step to further optimize marker locations after tracking by fitting a polynomial with Gaussian weight to the image intensity function (Rogers et al., 2007). The image intensity function (Fig. 4A) shows that X-ray images of radio-opaque markers have smooth transitions from the lighter border of a marker to the darker center, as more material has to be traversed in the center. The peak of the image intensity function is used to improve detected marker locations as well as radii (Fig. 4B) and in

most but not all cases, this optimization reduces the standard deviation of measured distances between markers of a rigid body.

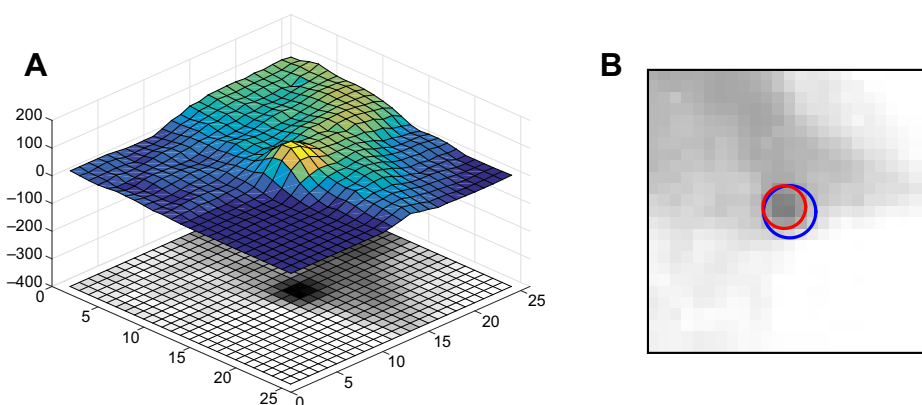
### 3D reconstruction, rigid body alignment and filtering

After the camera setup has been calibrated and the markers are tracked in the video sequences, the marker locations can be reconstructed in 3D. The 2D marker coordinates are first undistorted using the local weighted mean transformation. Then, reconstruction is performed by using linear triangulation with an iterative weight adjustment (Hartley and Sturm, 1996) and the reprojection errors (Fig. 2A) are computed based on the distances between the tracked marker's 2D positions and the marker's 3D position projected onto the images followed by a distortion. Next, 3D marker locations of a rigid body are aligned with the marker locations acquired from a CT scan of the specimen and the translation and rotation of the body are computed using unit quaternions (Horn, 1987). Finally, the translation and rotations in exponential coordinates of a rigid body can be smoothed with a low-pass Butterworth filter based on a user-defined filter frequency. After filtering, rigid body errors (Fig. 2C) are computed by animating the CT marker coordinates of a rigid body with the filtered and unfiltered transformations. Errors are computed in 3D by computing the distance between the animated CT locations and the reconstructed 3D locations of a marker, as well as in 2D by projecting the animated CT locations onto the image followed by a distortion and by comparing them with the tracked marker positions.

### Study 1: measurement of accuracy and precision

To determine the accuracy and precision of XMA Lab, we performed several trials with rigid X-ray phantoms, each consisting of two radio-opaque markers placed at a known distance in a relatively radio-lucent wand that was waved in the X-ray beams. These tests were conducted at the W. M. Keck Foundation XROMM Facility at Brown University with a biplanar videoradiography system: two Varian model G-1086 X-ray tubes, two EMD Technologies model EPS 45-80 pulsed X-ray generators, two Dunlee model TH9447QXH590 image intensifiers (40 cm diameter) and two Phantom v10 high-speed digital video cameras. The system was operated in pulsed mode with a pulse duration of 4 ms, source-image distance of 130 cm, 75 kV, 100 mA, magnification 0, with cameras recording at 1760×1760 pixels, 50 frames s<sup>-1</sup> and 1500 μs exposure time.

For each trial, we computed the mean intermarker distance as a measure of accuracy and the standard deviation of intermarker distances as a measure of precision (Brainerd et al., 2010; Tashman and Anderst, 2003; You et al., 2001). Camera calibration was performed using a new kind of X-ray calibration object constructed



**Fig. 4. Marker refinement based on image intensity function.** (A) Pixel intensity plotted as a 2D function across the region of interest. (B) Original marker position (blue) and refined marker position (red).

of Lego blocks (Lego Group, Billund, Denmark) and 5 mm diameter steel spheres (see Supplementary Information 1: Construction of a Lego Calibration Object). Six images for each camera with different orientations of the calibration object were used for calibration. One phantom consisted of a 3.2 mm-thick aluminium plate with markers (3 mm spherical steel beads) at a fixed bead distance of 38.1 mm machined at a tolerance of  $\pm 0.02$  mm (Brainerd et al., 2010). Three more phantoms were constructed from Lego bricks. Markers (5 mm spherical steel beads) were inserted into the Lego bricks at distances of 16.0, 32.0 and 64.0 mm. The steel beads were detected in the image using the same algorithms as during the calibration.

### Study 2: comparison of XMALab with XrayProject for precision and reproducibility

For the past 8 years we have used one specific trial of minipig feeding for tutorials and software testing for both XMALab and XrayProject (Brainerd et al., 2010). The dataset is available as part of an XMALab tutorial for new users (linked on [bitbucket.org/xromm/xmalab/](http://bitbucket.org/xromm/xmalab/)). These data were collected with a pair of modified mobile C-arm videofluoroscopes, with system and imaging parameters as described previously (Brainerd et al., 2010). The feeding sequence in this tutorial is 435 frames long and includes five chewing power strokes and four cycles of food gathering. It also includes examples of markers crossing over and occluding each other and areas where the image is dark and markers are hard to see and track. As part of routine instruction and software testing, nine users analyzed this tutorial dataset in both XrayProject and XMALab. The results were anonymized and repurposed for this study. Analysis in both programs included distortion correction, calibration, tracking 10 points over 435 frames, and creating two rigid bodies, one for the cranium and one for the lower jaw, each consisting of five markers. For the XMALab tutorial, four reference shapes were digitally drawn on the calibration images because geometric identifiers of reference object markers were not present in the originals (Fig. 1B).

To compare marker-tracking precision between XrayProject and XMALab we computed the standard deviation of intermarker distances for pairs of markers contained within each rigid body (Brainerd et al., 2010; Tashman and Anderst, 2003; You et al., 2001). To compare reproducibility among users for the rigid body transformations, we imported the minipig cranium and lower jaw models into Autodesk Maya and placed a joint coordinate system (JCS) at the temporomandibular joint (Menegaz et al., 2015). We calculated the six degree-of-freedom motions from the JCS for all

nine users and then calculated the mean and standard deviation for each frame across the nine users.

### Study 3: precision of measured joint kinematics

To determine the precision with which joint motion can be reconstructed by XROMM, the real motion of an object would have to be known *a priori*. As the real, world-space motion is not known, a previous study used a frozen specimen of a pig head with markers implanted into the skull and jaw to measure the precision of rotations and translations derived from a JCS placed at the temporomandibular joint (Menegaz et al., 2015). The frozen head was moved in the X-ray beams at approximately the frequency and amplitude of jaw motions *in vivo*. In a frozen specimen, all JCS rotations and translations should *a priori* be zero, as there is no actual motion between the jaw and cranium. The standard deviation of the apparent motion (noise) in each degree of freedom measured within the JCS is a measure of precision (Menegaz et al., 2015). The original study used XrayProject; we retraced the same markers in the same two trials with XMALab and recalculated the JCS precision (see Menegaz et al., 2015 for data collection methods).

## RESULTS AND DISCUSSION

### Study 1: measurement of accuracy and precision

We performed 11 trials in which four phantoms (four wands bearing two steel beads) were waved in front of both X-ray cameras. The X-ray systems were placed at an angle of about 90 deg and the wands were waved so that coverage of both camera images was as complete as possible. For each trial, we analyzed the frames in which both beads were visible in both cameras, resulting in a total of 5775 frames. Table 1 shows the resulting mean and standard deviation for all trials as well as the distance of beads in distorted image space, the bead velocity in 3D space and the number of frames used.

From the accuracy of the machined phantom (phantom 1), it can be seen that XMALab recovered the known distance of 38.1 mm up to an accuracy of 0.01 mm with a precision below 0.05 mm (Table 1). The actual distance of the wand could vary from 38.08 to 38.12 as a result of the manufacturing tolerance of  $\pm 0.02$  mm, which could introduce a slight deviation. Nonetheless, the high accuracy validates the accuracy of the Lego cube (see Supplementary Information 1) that was used for calibration. If the distances between the beads in the Lego calibration object were not correct, a strong deviation of the computed bead distance relative to the machined distance for phantom 1 would have been found. However, as this was not the case, it shows that beads can be placed in Lego blocks

**Table 1. Accuracy and precision of tracking two spherical steel beads embedded in wand-style phantoms at known distances for 2–3 trials per phantom**

Phantom ID	Material	Distance (mm) <sup>1</sup>	Accuracy (mean, mm) <sup>2</sup>	Precision (s.d., mm) <sup>3</sup>	No. of frames	Bead distance (pixels)	Bead velocity (mm s <sup>-1</sup> )
1	Aluminium	38.1	38.09	0.047	444	204.04 $\pm$ 14.64	2.47 $\pm$ 1.10
			38.11	0.041	522	196.00 $\pm$ 13.46	2.52 $\pm$ 0.99
			38.10	0.042	532	200.96 $\pm$ 11.73	2.39 $\pm$ 1.32
2	Lego	16.0	16.05	0.026	560	81.08 $\pm$ 6.78	2.68 $\pm$ 1.07
			16.05	0.025	494	80.07 $\pm$ 6.12	2.25 $\pm$ 1.03
			16.05	0.027	498	80.47 $\pm$ 6.30	2.10 $\pm$ 0.88
3	Lego	32.0	32.02	0.042	583	153.16 $\pm$ 17.53	1.97 $\pm$ 0.91
			31.98	0.042	517	155.52 $\pm$ 14.80	1.79 $\pm$ 0.79
			32.00	0.043	450	140.12 $\pm$ 26.88	2.49 $\pm$ 1.04
4	Lego	64.0	64.04	0.077	588	314.52 $\pm$ 16.39	2.20 $\pm$ 1.06
			64.04	0.067	587	314.84 $\pm$ 16.56	1.89 $\pm$ 1.00

<sup>1</sup>Known distance from phantom geometry. <sup>2</sup>Measured mean distance for 2–3 trials per wand. <sup>3</sup>Standard deviation of the measured distance across all frames of the video.

and assembled at a high precision. This validated the use of additional Lego phantoms for measuring accuracy and precision.

For phantom 3, we recovered the expected bead distance with an accuracy of 0.02 mm and precision below 0.05 mm (Table 1). The results for phantoms 2 and 4 differ slightly from the value assumed based on the Lego construction. For both phantoms, several trials resulted in beads being consistently 0.04 and 0.05 mm further apart than expected (Table 1). These offsets are most likely not an artifact of the processing in XMA Lab, but result from the construction of the Lego phantom itself. These offsets suggest that the precision of pressing steel beads into Lego bricks and then assembling the bricks for construction of phantoms or calibration objects is approximately 0.05 mm. It should be mentioned that the precision with which the wands can be manufactured has to be taken into account when validating setups and calibrations. We therefore recommend using wands that are manufactured at high precision or, if Lego bricks are used to acquire the measurement, using multiple wands to compensate for inaccuracy that could occur during the construction.

The different bead distances in the three Lego wands yielded an unexpected result for precision. The precision shows the variation of the measured length over a sequence and is independent of the manufacturing precision. It is therefore expected to be the same in all wands with two beads because it should be two times the precision for tracking each individual bead, no matter how far apart they are. Instead, we found that precision decreases as the distance between the beads increases (Table 1). Precision was 0.025 mm for beads placed 16 mm apart (phantom 2) and substantially worse at 0.075 mm for beads placed 64 mm apart (phantom 4). No bias could be detected in terms of whether the beads were at the periphery or the center of the image. We therefore suspect that the cause lies in the complex nature of distortions introduced by the fluoroscopic image intensifiers. The distortions are local and vary between different parts of the image. Our algorithm computes undistortions locally, but small variations can still occur due to the sampling of the grid points used, and distortions can slightly change over time (Verdonck et al., 1999). However, beads that are closer together will always be subject to similar distortions and therefore less affected, while beads that are further apart will be subject to more distinct distortion fields, resulting in higher deviations from the mean. To keep these errors at a minimum, we recommend acquiring distortion and calibration data before and after recording sessions, and using the standard deviations of marker to marker distances for pairs of beads within a rigid body to decide which distortion correction and calibration are best to use.

### Study 2: comparison of XMA Lab with XrayProject for precision and reproducibility

We compared marker-tracking precision and rigid body motion consistency (reproducibility) across users ( $N=9$ ) for XMA Lab and XrayProject. Standard deviations of pairwise intermarker distances were  $0.062\pm 0.010$  mm for XMA Lab and  $0.148\pm 0.035$  for XrayProject (mean across all marker pairs and all users,  $\pm$ s.d.). These means are significantly different (one-way repeated measures ANOVA,  $F_{1,15}=54.79$ ,  $P<0.0001$ ), with the marker-tracking precision from XMA Lab being more precise (one-tailed paired  $t$ -test,  $t_8=-7.5657$ ,  $P<0.0001$ ). XrayProject precision measured here is not as good as values ( $<0.1$  mm) obtained previously for the same pig feeding example data (Brainerd et al., 2010). Further examination suggests that the reason for this difference is the skill and care with which users track points for a tutorial dataset, versus tracking markers for their own research. Indeed, many studies have used XrayProject and obtained precision better than 0.1 mm

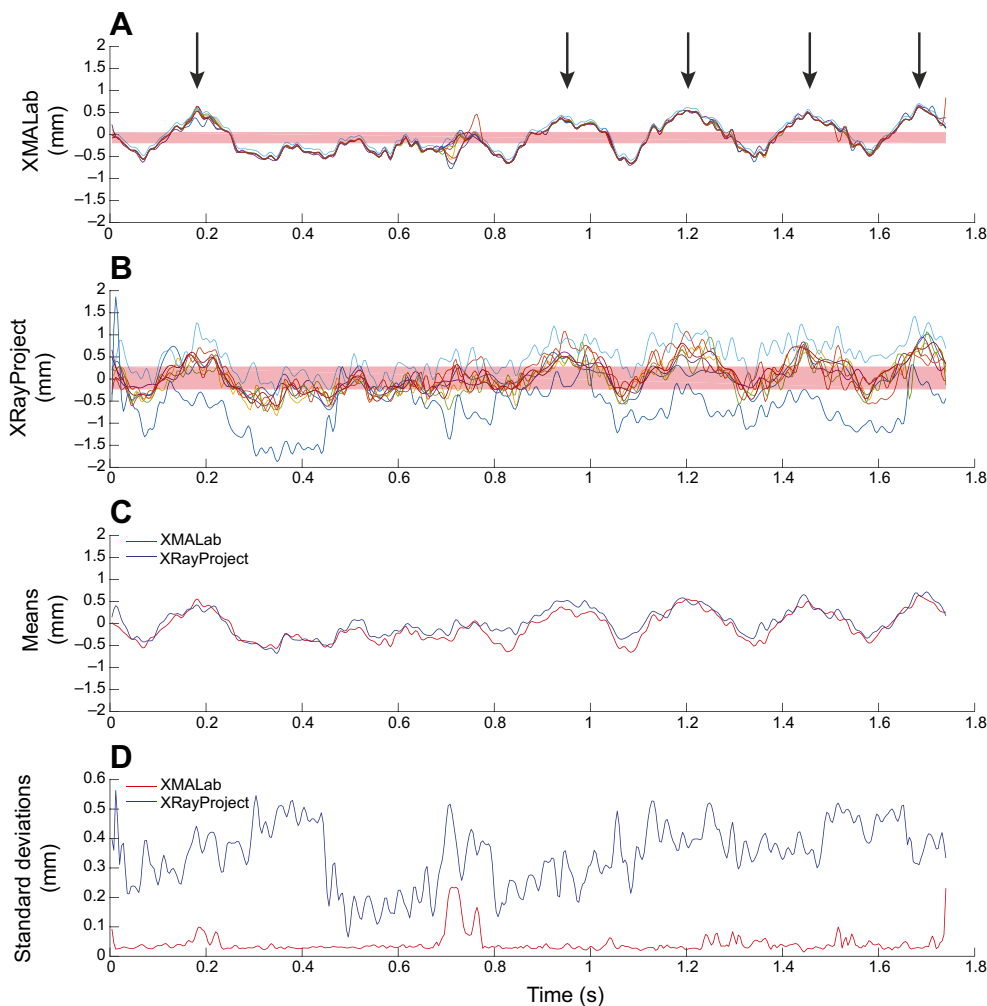
(e.g. Camp and Brainerd, 2015; Dawson et al., 2011; Gidmark et al., 2012; Konow et al., 2015). Despite this difference in user effort, the comparison is a good test for XMA Lab because it shows high precision and repeatability (see below) with XMA Lab, even for a tutorial dataset when users are novices and are not putting extra effort into data refinement.

The s.d. of the mean standard deviation of intermarker distance among users was 0.01 mm in XMA Lab and 0.035 mm in XrayProject, indicating that marker tracking is more consistent (reproducible) across users in XMA Lab. To see to what extent data are uniform across users, we compared results across users in more detail. First, we investigated the differences in the computed 3D points. For each frame and each point, we computed the mean 3D point for all users for XMA Lab and XrayProject and compared the result with the data from each user. The results from XMA Lab and XrayProject were significantly different (one-way repeated measures ANOVA,  $F_{1,15}=51.05$ ,  $P<0.0001$ ) with the 3D points in XMA Lab being more similar to each other (one-tailed paired  $t$ -test,  $t_8=-6.6615$ ,  $P<0.0001$ ). The deviation of 3D points from the mean 3D points across users differed between the programs by an order of magnitude, with a mean across users of  $0.019\pm 0.0072$  mm for XMA Lab and  $0.19\pm 0.073$  mm for XrayProject.

In addition to the similarity of 3D points, we also looked at the computed movement of the mandible relative to the cranium from a temporomandibular JCS (Brainerd et al., 2010; Menegaz et al., 2015) and compared the results from each user with the average of all users. As an example of one of six degrees of freedom from the JCS, the translations of the mandible along the  $y$ -axis relative to the cranium are less variable among users in XMA Lab (Fig. 5A) than in XrayProject (Fig. 5B). As expected, the mean values of  $y$ -axis translation across all users are very similar (Fig. 5C), indicating that the two programs are producing the same mean result. The lower standard deviation of all frames for XMA Lab compared with XrayProject (Fig. 5D) shows the greater reproducibility of data in XMA Lab. Similar plots for all rotations and translations from the temporomandibular JCS are in Supplementary Information 2. For all translations and rotations, the data from users in XMA Lab were 3- to 6-fold less variable ( $P<0.0001$  for all repeated measures ANOVA and one-tailed paired  $t$ -tests; see Supplementary Information 2, Table S3 for results of the statistical tests).

The more precise intermarker distances achieved with XMA Lab also showed an effect when marker  $xyz$  coordinates from a CT scan were aligned with the reconstructed 3D points by using the five points of each rigid body. The mean distance between the aligned CT coordinates and the reconstructed marker locations was  $0.40\pm 0.0035$  mm for XMA Lab and  $0.43\pm 0.038$  mm for XrayProject. These rigid body residuals are statistically significantly lower in XMA Lab (one-tailed paired  $t$ -test,  $t_8=-2.6452$ ,  $P=0.0147$ ), but a misalignment between CT data and reconstructed points was present in both programs. To determine whether the cause lies in tracking, undistortion and calibration errors, or error in the CT coordinates, we computed for each program a rigid marker constellation based on the reconstructed 3D points. We used just 99% of the frames containing the constellations that matched each other best to compensate for outliers. Alignment of the new constellation with the tracked points showed greater differences between programs than the alignment with the CT locations (one-way repeated measures ANOVA,  $F_{1,15}=51.05$ ,  $P<0.0001$ ) and the residuals were more than 2-fold lower at  $0.05\pm 0.0032$  mm in XMA Lab and  $0.13\pm 0.036$  mm in XrayProject for all users (one-tailed paired  $t$ -test,  $t_8=-6.6615$ ,  $P<0.0001$ ). The results are also in





**Fig. 5. Variation among users in results from a joint coordinate system (JCS) measuring motion of the mandible relative to the cranium in a miniature pig.** One of six degrees of freedom is shown: translation along the  $y$ -axis (in mm) which represents compression of the temporomandibular joint (Menegaz et al., 2015). (A) Results from XMA Lab for nine users. (B) Results from XrayProject for nine users. Each user is represented by a different color in A and B. (C) Mean result for all users for the two programs. (D) Standard deviation for each time point among users. (See Supplementary Information 2, Figs S6–S11, for plots of all six degrees of freedom.) Precision threshold from a frozen specimen study is shown as a salmon-colored bar centered at the mean of the data in A and B. Arrows indicate the five power strokes of chewing in this sequence.

accordance with the measured standard deviations of marker to marker distances as both measure the precision with which a constellation of markers can be tracked. These data and the results from study 1 show that the high residuals for the alignment with the CT coordinates can be attributed to errors in the CT marker coordinates. We could not determine whether these artifacts are a result of the CT scanner, the process to extract CT marker coordinates from the scans, or changes of the specimen between the study and the CT data acquisition.

These results show that XMA Lab yields 2D points that are about 2-fold more precise and 3-fold more consistent across users, resulting in 3D points that are 10-fold more consistent across users than in XrayProject. The similarity of results across users in XMA Lab can be attributed to the centroid-finding algorithms (Figs 3 and 4), which result in the same 2D location of a marker for each user. It is possible that consistent but erroneous centroid detection could produce the same erroneous 3D point for all users, but the more precise intermarker distances in XMA Lab indicate that the 3D points were more accurately reconstructed than with previous methods. However, it should be noted that when marker images touched or completely overlapped, individual centroids could not be accurately detected. In this case, users still had to manually refine the points, which led to small deviations between users in XMA Lab for some frames of the sequence (Fig. 5A, between 0.70 and 0.78 s). Overall, the derived motion data were much more consistent across users in XMA Lab, even though

several of the users were novices learning XMA Lab through completion of the tutorial for the first time.

Finally, in addition to more precise and reproducible marker tracking, the computation speed and design of the user interface of XMA Lab facilitate error checking at all stages of data collection and analysis. The preprocessing steps of undistortion and calibration are substantially faster in XMA Lab because of rapid computation of the undistortion transform, automatic calibration by using four lead shapes and the improved workflow of XMA Lab with linked workspaces for each step (Fig. 1). The fast preprocessing even permits users to perform calibrations during initial data collection to validate setup configurations. During marker tracking, incorporation of plot windows (Fig. 2) provides real-time feedback for detection of tracking errors. The rigid body error plot (Fig. 2C) is particularly valuable for detecting errors in marker identification, over-filtering, or dislocation of markers between video collection and CT scanning. Some errors, particularly dislocation of markers, can be hard to detect and can even result in reasonable looking but incorrect animations. As a best practice, all bone animations should routinely be checked visually against X-ray videos before further analysis. This verification step is supported in XMA Lab through export of files for placing virtual X-ray cameras in Autodesk Maya and undistorted trial images, and the newest version, XMA Lab 1.3.7 (released during revision of this paper), supports the display of animated mesh models for visual evaluation of animation quality.

### Study 3: precision thresholds for measured joint kinematics

For measuring the precision threshold for rigid body motions in XMA Lab, we used the same two trials from a frozen-specimen study previously analyzed in XrayProject (Menegaz et al., 2015). Standard deviations of intermarker distances indicated greater marker-tracking precision in XMA Lab (0.080 mm) than the reported results by Menegaz et al. (2015) (0.15 mm). The specimen in this precision study had nine markers, instead of the 10 markers in the specimen for the tutorial data, and marker-tracking precision was slightly worse (0.080 versus 0.062 mm) than in the tutorial data, indicating lower image quality. Thus, the thresholds calculated from this frozen-specimen study are conservative when plotted relative to the tutorial data.

XMA Lab produced smaller rigid body precision thresholds than XrayProject (Fig. 5), permitting the measurement of more subtle motions in XMA Lab. For example, potential compression of the joint space within the temporomandibular joint during the power stroke of chewing would appear as *Y*-axis translation in the temporomandibular JCS. With the more consistent results among users and low precision threshold in XMA Lab (Fig. 5A), it is clear that the joint is compressing with every power stroke. In contrast, with the larger precision threshold from XrayProject, it would not be possible to say with confidence that the joint is compressing from any one of the users' data (Fig. 5B).

### Concluding remarks

As open-source software for research becomes more and more complex, it is important to document in peer-reviewed publications that the new software is valid for the purpose intended. Here, we have demonstrated the accuracy, precision and reproducibility of results using XMA Lab. The higher accuracy makes it possible to investigate movements that were previously not possible with the existing tools. We also showed that the user interface, the workflows and the improved algorithms make it possible for even novice users to quickly obtain accurate results. XMA Lab facilitates exchange of data and reproducibility of results by retaining all settings and data in a single file. This eases collaboration and communication among peers, as well as troubleshooting of problems, as each user will see the same view of the data. Combined with the XMAPortal for data management, the open-source XMA Lab software will improve the speed of motion analysis, the ease of learning for novice users, and the quality of data from both experienced and novice users for XROMM studies.

### Acknowledgements

Thanks to E. G. Tavares for help with data acquisition, to H. Cryst for contributions to calibration object design, to R. A. Menegaz for allowing us to use her data and to P. Yan for her help with the statistical analysis. The XMAPortal code was written by Kia Huffman with user interfaces by Tom Hoogendyk.

### Competing interests

The authors declare no competing or financial interests.

### Author contributions

E.L.B., D.B.B. and S.M.G. developed the initial concept for new X-ray motion analysis software, B.J.K. designed and wrote the XMA Lab software, B.J.K., E.L.B., D.B.B. and S.M.G. designed the accuracy, precision and reproducibility tests, B.J.K. and J.D.L.-C. collected and analyzed the accuracy and precision data, B.J.K. analyzed the reproducibility data and did all statistical tests, J.D.L.-C. and B.J.K. designed, built and tested the Lego calibration object, B.J.K. and E.L.B. wrote the manuscript and all authors read the manuscript, provided critical comments for revision and approved the final version.

### Funding

This work was supported by the US National Science Foundation under grant numbers 1120967 and 1262156.

### Data availability

Data have been deposited in the XMAPortal (xmaportal.org), in the study 'Data for Software and Hardware Validation' with permanent ID BROWN17.

### Supplementary information

Supplementary information available online at <http://jeb.biologists.org/lookup/doi/10.1242/jeb.145383.supplemental>

### References

- Astley, H. C. and Roberts, T. J. (2012). Evidence for a vertebrate catapult: elastic energy storage in the plantaris tendon during frog jumping. *Biol. Lett.* **8**, 386-389.
- Baier, D. B. and Gatesy, S. M. (2013). Three-dimensional skeletal kinematics of the shoulder girdle and forelimb in walking Alligator. *J. Anat.* **223**, 462-473.
- Baier, D. B., Gatesy, S. M. and Dial, K. P. (2013). Three-dimensional, high-resolution skeletal kinematics of the avian wing and shoulder during ascending flapping flight and uphill flap-running. *PLoS ONE* **8**, e63982.
- Banks, S. A. and Hodge, W. A. (1996). Accurate measurement of three-dimensional knee replacement kinematics using single-plane fluoroscopy. *IEEE Trans. Biomed. Eng.* **43**, 638-649.
- Bonnan, M. F., Shulman, J., Varadharajan, R., Gilbert, C., Wilkes, M., Horner, A. and Brainerd, E. (2016). Forelimb kinematics of rats using XROMM, with implications for small eutherians and their fossil relatives. *PLoS ONE* **11**, e0149377.
- Bradski, G. (2000). The opencv library. *Doctor Dobbs J.* **25**, 120-126.
- Brainerd, E. L., Baier, D. B., Gatesy, S. M., Hedrick, T. L., Metzger, K. A., Gilbert, S. L. and Crisco, J. (2010). X-ray Reconstruction of Moving Morphology (XROMM): precision, accuracy and applications in comparative biomechanics research. *J. Exp. Zool.* **313A**, 262-279.
- Brainerd, E. L., Moritz, S. and Ritter, D. A. (2015). XROMM analysis of rib kinematics during lung ventilation in the green iguana, *Iguana iguana*. *J. Exp. Biol.* **219**, 404-411.
- Camp, A. L. and Brainerd, E. L. (2015). Reevaluating musculoskeletal linkages in suction-feeding fishes with X-ray reconstruction of moving morphology (XROMM). *Integr. Comp. Biol.* **55**, 36-47.
- Dawson, M. M., Metzger, K. A., Baier, D. B. and Brainerd, E. L. (2011). Kinematics of the quadrate bone during feeding in Mallard ducks. *J. Exp. Biol.* **214**, 2036-2046.
- Deng, B. (2015). ToolBox: see how they run. *Nature* **525**, 145-146.
- Falkingham, P. L. and Gatesy, S. M. (2014). The birth of a dinosaur footprint: subsurface 3D motion reconstruction and discrete element simulation reveal track ontogeny. *Proc. Natl. Acad. Sci. USA* **111**, 18279-18284.
- Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* **24**, 381-395.
- Gatesy, S. M., Baier, D. B., Jenkins, F. A. and Dial, K. P. (2010). Scientific Rotoscoping: a morphology-based method of 3-D motion analysis and visualization. *J. Exp. Zool.* **313A**, 244-261.
- Gidmark, N. J., Staab, K. L., Brainerd, E. L. and Hernandez, L. P. (2012). Flexibility in starting posture drives flexibility in kinematic behavior of the kinethmoid-mediated premaxillary protrusion mechanism in a cyprinid fish, *Cyprinus carpio*. *J. Exp. Biol.* **215**, 2262-2272.
- Goshtasby, A. (1988). Image registration by local approximation methods. *Image Vis. Comput.* **6**, 255-261.
- Hartley, R. I. and Sturm, P. (1996). Triangulation. *Comput. Analysis Images Patterns* **970**, 190-197.
- Hedrick, T. L. (2008). Software techniques for two- and three-dimensional kinematic measurements of biological and biomimetic systems. *Bioinsp. Biomim.* **3**, 6pp.
- Heers, A. M. and Dial, K. P. (2012). From extant to extinct: locomotor ontogeny and the evolution of avian flight. *Trends Ecol. Evol.* **27**, 296-305.
- Heers, A. M., Baier, D. B., Jackson, B. E. and Dial, K. P. (2016). Flapping before flight: high resolution, three-dimensional skeletal kinematics of wings and legs during Avian development. *PLoS ONE* **11**, e0153446.
- Horn, B. K. P. (1987). Closed-form solution of absolute orientation using unit quaternions. *J. Optic. Soc. Amer.* **4**, 629-642.
- Kambic, R. E., Roberts, T. J. and Gatesy, S. M. (2014). Long-axis rotation: a missing degree of freedom in avian bipedal locomotion. *J. Exp. Biol.* **217**, 2770-2782.
- Kambic, R. E., Roberts, T. J. and Gatesy, S. M. (2015). Guineafowl with a twist: asymmetric limb control in steady bipedal locomotion. *J. Exp. Biol.* **218**, 3836-3844.
- Konow, N., Cheney, J. A., Roberts, T. J., Waldman, J. R. S. and Swartz, S. M. (2015). Spring or string: does tendon elastic action influence wing muscle mechanics in bat flight? *Proc. R. Soc. B Biol. Sci.* **282**, 20151832.
- Menegaz, R. A., Baier, D. B., Metzger, K. A., Herring, S. W. and Brainerd, E. L. (2015). XROMM analysis of tooth occlusion and temporomandibular joint kinematics during feeding in juvenile miniature pigs. *J. Exp. Biol.* **218**, 2573-2584.

- Miranda, D. L., Schwartz, J. B., Loomis, A. C., Brainerd, E. L., Fleming, B. C. and Crisco, J. J.** (2011). Static and dynamic error of a biplanar videoradiography system using marker-based and markerless tracking techniques. *J. Biomech. Eng.* **133**, 1-8.
- Miranda, D. L., Fadale, P. D., Hulstyn, M. J., Shalvoy, R. M., Machan, J. T. and Fleming, B. C.** (2013). Knee biomechanics during a jump-cut maneuver: effects of sex and ACL surgery. *Med. Sci. Sports Exerc.* **45**, 942-951.
- Montuelle, S. J. and Williams, S. H.** (2015). In Vivo Measurement of Mesokinesis in Gekko gecko: the role of cranial kinesis during gape display, feeding and biting. *PLoS ONE* **10**, e0134710.
- Nyakatura, J. A. and Fischer, M. S.** (2010). Functional morphology and three-dimensional kinematics of the thoraco-lumbar region of the spine of the two-toed sloth. *J. Exp. Biol.* **213**, 4278-4290.
- Rogers, S. S., Waigh, T. A., Zhao, X. and Lu, J. R.** (2007). Precise particle tracking against a complicated background: polynomial fitting with Gaussian weight. *Phys. Biol.* **4**, 220-227.
- Tashman, S. and Anderst, W.** (2003). In vivo measurement of dynamic joint motion using high speed radiography and CT: application to canine ACL deficiency. *J. Biomech. Eng.* **125**, 238-245.
- Verdonck, B., Bourel, P., Coste, E., Gerritsen, F. A. and Rousseau, J.** (1999). Variations in the geometrical distortion of x-ray image intensifiers. *Med. Imaging* **1999**, 226-275.
- Wachs, K., Fischer, M. S. and Schilling, N.** (2016). Three-dimensional movements of the pelvis and the lumbar intervertebral joints in walking and trotting dogs. *Vet. J.* **210**, 46-55.
- Wang, J. and Blackburn, T. J.** (2000). The AAPM/RSSNA physics tutorial for residents: X-ray image intensifiers for fluoroscopy. *RadioGraphics* **20**, 1471-1477.
- You, B.-M., Siy, P., Anderst, W. and Tashman, S.** (2001). In vivo measurement of 3-D skeletal kinematics from sequences of biplane radiographs: application to knee kinematics. *IEEE Trans. Med. Imag.* **20**, 514-525.
- Zhang, Z.** (2000). A flexible new technique for camera calibration. *IEEE Trans. Pattern Anal. Mach. Intell.* **22**, 1330-1334.

## Supplementary Information 1: Construction of a Lego Calibration Object

Supplement to: B.J. Knörlein, D.B. Baier, S.M. Gatesy, J.D. Laurence-Chasen and E.L. Brainerd. Validation of XMALab Software for Marker-based XROMM.

In this document we describe and provide assembly instructions for a 64-point X-ray calibration object made from Lego bricks with 5.0 mm steel spheres press-fit into select bricks as radio-opaque marker points (Figs. S1-S3). Lego bricks have been validated in the past for calibration of external view cameras (Baronti et al., 2010), and here we extend this use to X-ray cameras.

In prior XROMM work we used a custom-machined calibration object made from acrylic sheets with machined spacers and holes drilled for the steel spheres (Brainerd et al., 2010). Building an acrylic object requires access to high-precision machine tools, and the assembled object can warp and marker beads can move over time. As an alternative, to maximize precision and simplify construction, we developed an X-ray calibration object made out of Lego bricks.

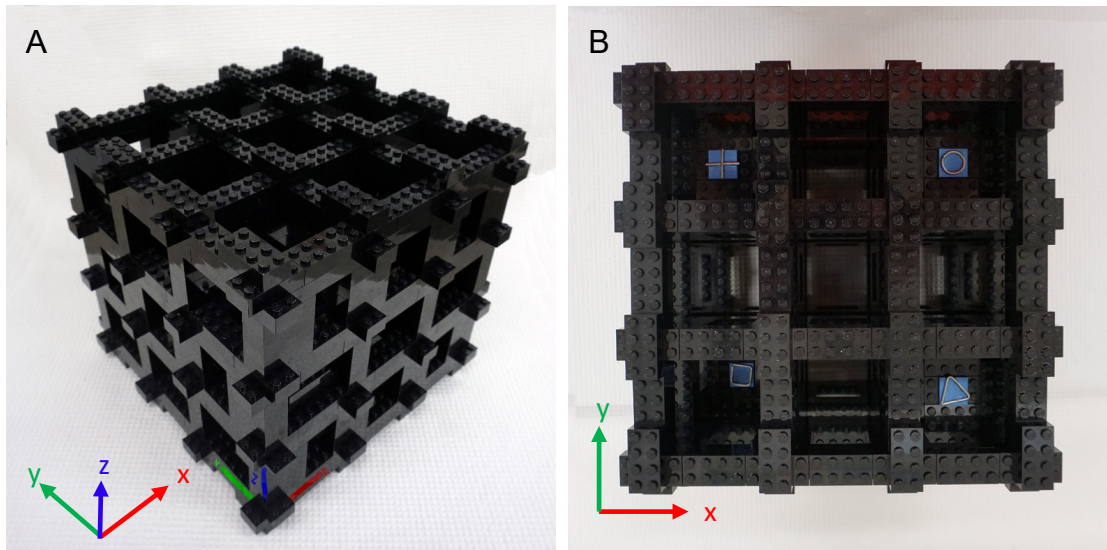


Figure S1. External views of a 64-point Lego calibration cube. (A) Oblique view. (B) Top views showing lead solder shapes used as reference points for determining the pose of the object.

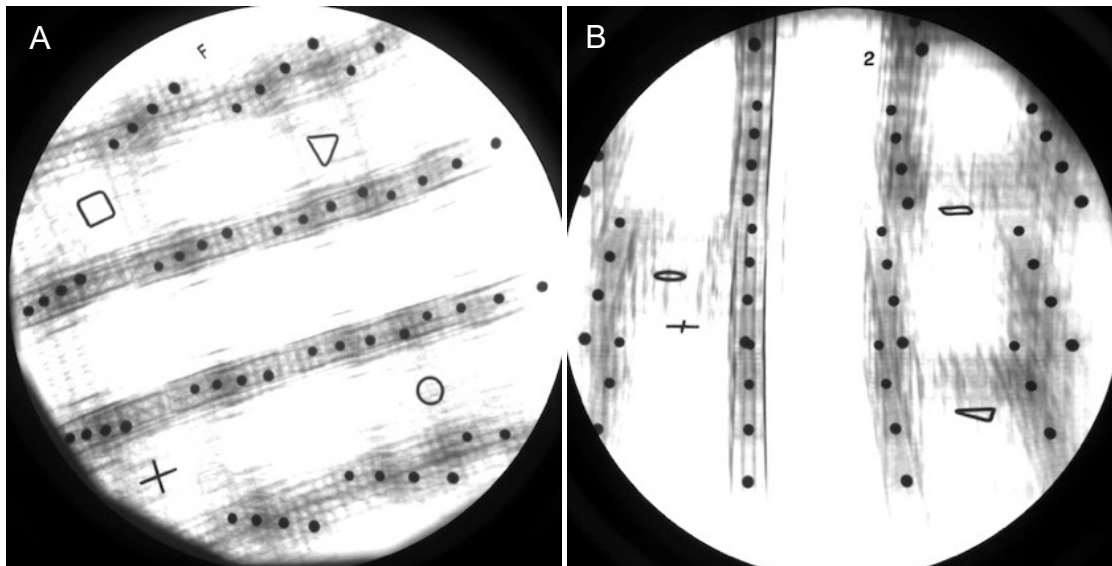


Figure S2. Biplanar X-ray images of a 64-point Lego calibration cube showing steel marker balls and lead solder reference shapes (square, circle, triangle, cross) for determining the pose of the object. The stainless steel balls are press-fit into selected Lego bricks to create a 4x4x4 lattice with 64 mm spacing. (A) Camera 1. (B) Camera 2.

The stainless steel balls in this calibration cube form a 4x4x4 lattice with 64 mm (6.4 cm) spacing between the beads (Fig. S2). The overall size of the marker lattice is 192x192x192 mm, and the overall size of the Lego calibration object is 224 mm x 224 mm x 211.2 mm. This is a large cube and typically not all of the points will appear in the X-ray images (Fig. S2). Construction of a smaller calibration object would be necessary for smaller fields of view. In the future, designs for other Lego calibration objects will be posted on the XMALab Bitbucket site ([bitbucket.org/xromm/xmlab/](http://bitbucket.org/xromm/xmlab/)).

Lego bricks are manufactured with high precision (tolerance 0.002 mm, from: [http://cache.lego.com/downloads/aboutus/LEGO\\_company\\_profile\\_UK.pdf](http://cache.lego.com/downloads/aboutus/LEGO_company_profile_UK.pdf)), as they must fit together firmly, yet be easily disassembled. These consistent dimensions make it possible to construct a robust Lego frame that holds steel balls at easily calculated and highly accurate

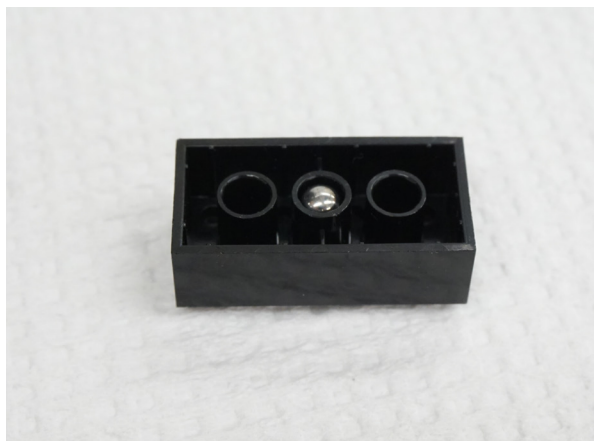


Figure S3. 2x4 Lego brick with a 5.0 mm stainless steel ball inserted into the center cylinder. Consistent depth is ensured by pressing ball in with brick inverted against a flat surface.

positions. It is also convenient that Lego bricks have internal cylinders into which the 5.0 mm diameter steel spheres can be press-fit tightly (Fig. S3). The pegs of an interlocking brick fit around the cylinders, so the bricks still interlock tightly with the steel balls in place.

Below we describe the assembly process of the Lego cube and the framespec (Table S1) and reference (Table S2) files required for XMALab calibration. The frame specification file contains the xyz coordinates of the steel balls (Table S1). In the past we have expressed marker positions in framespec files in centimeters, but starting with this Lego object, we recommend writing all framespec files for XROMM in millimeters. In the past we have also placed the lead solder shapes

around existing markers in the cube. Here we give them their own locations as points 65-68 in the framespec file (Table S1). The references file is a space-separated text file with the extension .ref that contains just the marker numbers and names of the shapes (Table S2). The user can also select reference points within the XMALab interface if some of the shapes are not visible but other points are known.

#### Required Parts:

- 64 5.0 mm spherical stainless steel balls (McMaster-Carr Part # 1598K26)
- 1.5 mm lead solder
- 492 2x4 Lego bricks Part #3001
- 104 2x6 Lego bricks Part #2456
- 4 2x2 Lego tiles Part #3068a

First, each bead was inserted into a 2x4 brick (Fig. S3). To ensure the beads were placed at identical depths, each bead was first fit partway into the center cylinder, then the brick was firmly pressed against a flat surface until the bottom of the bead was flush with the bottom of the brick.

**Table S1.** Lego cube frame specification file values and format (comma separated values; framespec.csv) for the 64-point Lego cube with 64 mm spacing between the beads, expressed in centimeters because cm units are more convenient for subsequent XROMM animation in Autodesk Maya.

x,y,z	
0,0,0	0,0,11.52
6.4,0,0	6.4,0,11.52
12.8,0,0	12.8,0,11.52
19.2,0,0	19.2,0,11.52
0,6.4,0	0,6.4,11.52
6.4,6.4,0	6.4,6.4,11.52
12.8,6.4,0	12.8,6.4,11.52
19.2,6.4,0	19.2,6.4,11.52
0,12.8,0	0,12.8,11.52
6.4,12.8,0	6.4,12.8,11.52
12.8,12.8,0	12.8,12.8,11.52
19.2,12.8,0	19.2,12.8,11.52
0,19.2,0	0,19.2,11.52
6.4,19.2,0	6.4,19.2,11.52
12.8,19.2,0	12.8,19.2,11.52
19.2,19.2,0	19.2,19.2,11.52
0,0,5.76	0,0,17.28
6.4,0,5.76	6.4,0,17.28
12.8,0,5.76	12.8,0,17.28
19.2,0,5.76	19.2,0,17.28
0,6.4,5.76	0,6.4,17.28
6.4,6.4,5.76	6.4,6.4,17.28
12.8,6.4,5.76	12.8,6.4,17.28
19.2,6.4,5.76	19.2,6.4,17.28
0,12.8,5.76	0,12.8,17.28
6.4,12.8,5.76	6.4,12.8,17.28
12.8,12.8,5.76	12.8,12.8,17.28
19.2,12.8,5.76	19.2,12.8,17.28
0,19.2,5.76	0,19.2,17.28
6.4,19.2,5.76	6.4,19.2,17.28
12.8,19.2,5.76	12.8,19.2,17.28
19.2,19.2,5.76	19.2,19.2,17.28
	3.2,3.2,9.76
	16,16,9.76
	16,3.2,15.52
	3.2,16,15.52

\*the last 4 points are points 65-68 and contain the coordinates of the reference points used to determine calibration object orientation in XMALab (see Table S2).

**Table S2.** Lego cube references file values and format (space separated values, with file extension .ref, such as references.ref).

65 Square  
 66 Circle  
 67 Triangle  
 68 Cross

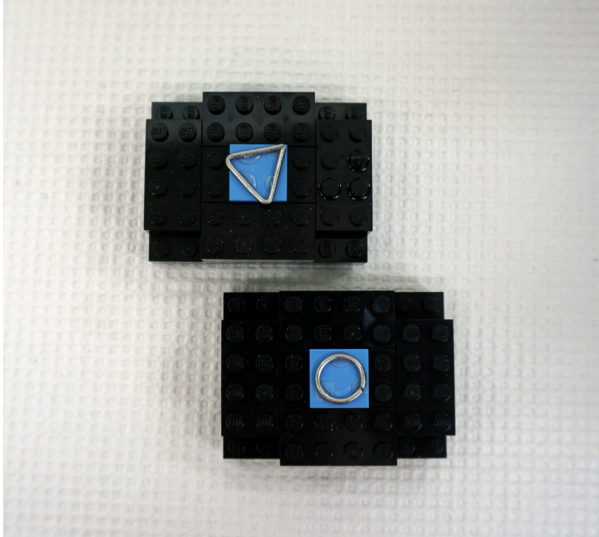


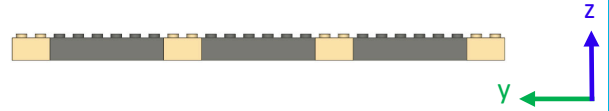
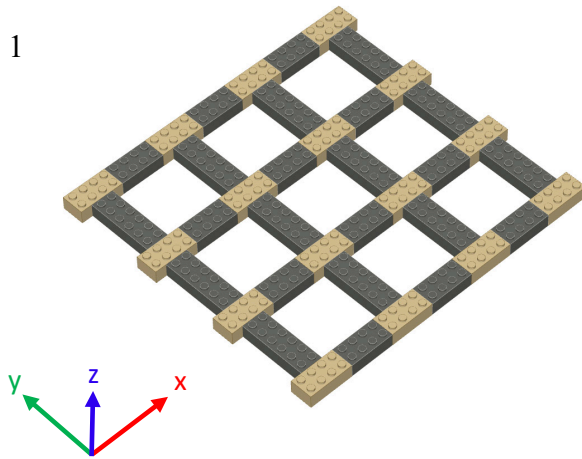
Figure S4: Shapes made of 1.5 mm lead solder were attached to 2x2 Lego tiles and serve as reference points for XMA Lab calibration.

To create the reference shapes, 1.5 mm lead solder was cut and bent into a square, circle, cross, and triangle, each the approximate size of a 2x2 Lego tile (Fig. S4). The shapes were laid flat on the tiles and secured in place with superglue. Once in place, the shapes provide a visual indication of the orientation of the cube.

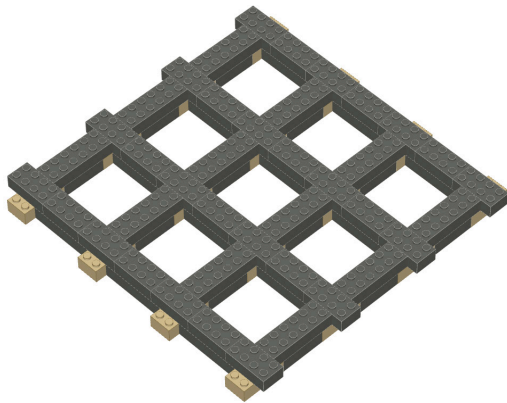
Once all 64 beads were fit into the bricks and the lead solder shapes were made, we began assembly of the cube itself. Figure S5 on the following pages is a step-by-step construction guide, created with the open-source software Bricksmith (Allen Smith). While the bricks in the actual cube can be any color, in the guide, bricks that contain steel beads are beige. Steps 6-10 detail the assembly of the segments that support the four reference shapes that are used for automatic calibration in XMA Lab. However, we found it helpful to add a set of external axes labels, to make it easy to quickly determine the cube's orientation (Fig S1).

Figure S5. Lego cube assembly steps. Beige bricks contain a 5.0 mm lead bead.

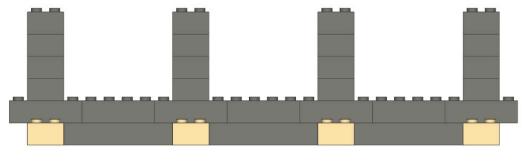
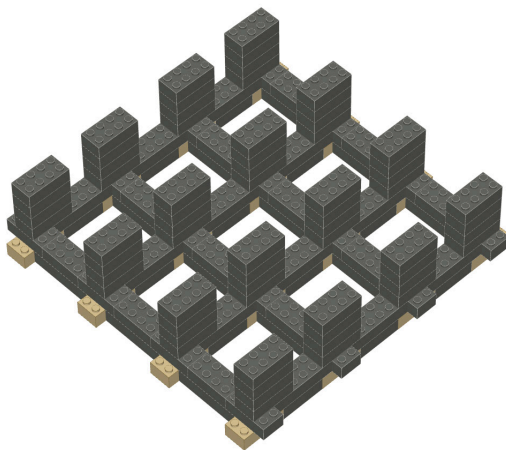
1



2

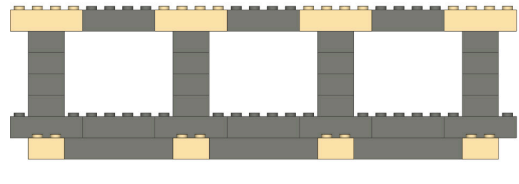
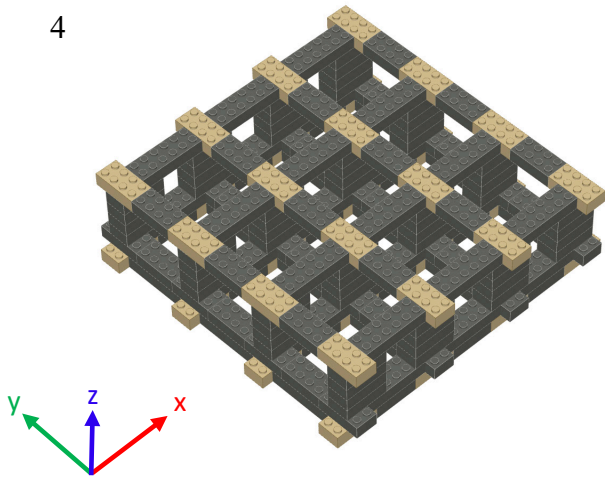


3

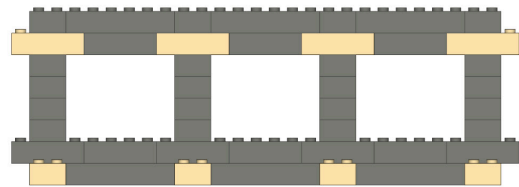
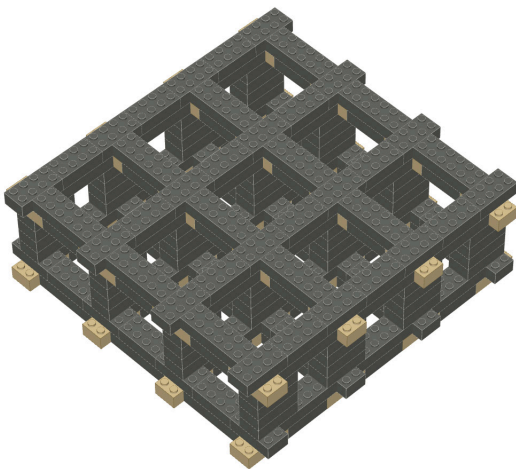




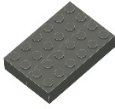
4



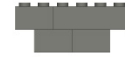
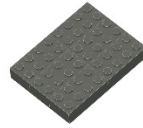
5



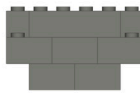
6\*



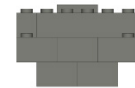
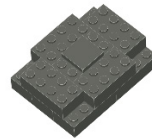
7



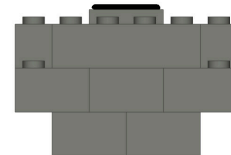
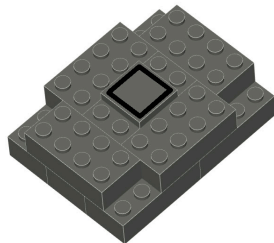
8



9

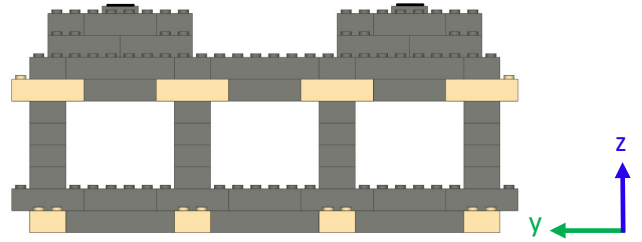
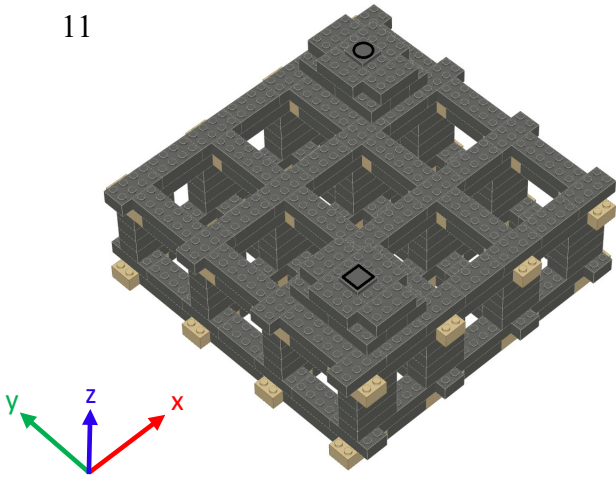


10

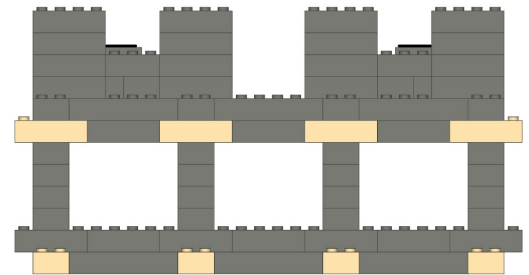
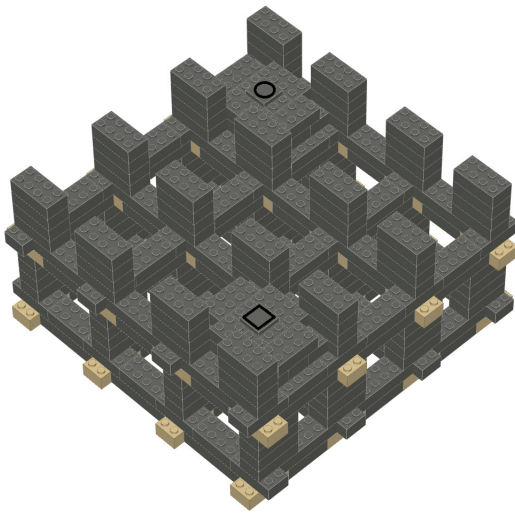


\*Steps 6-10 repeated four times, one for each lead solder shape: Square, circle, cross, and triangle. Shapes attached to tiles with superglue.

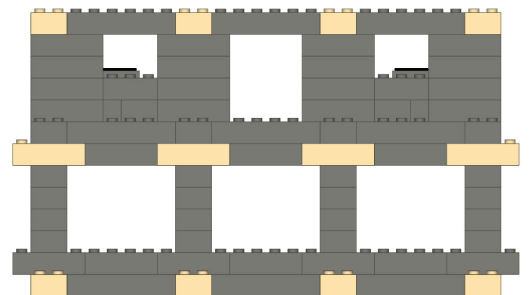
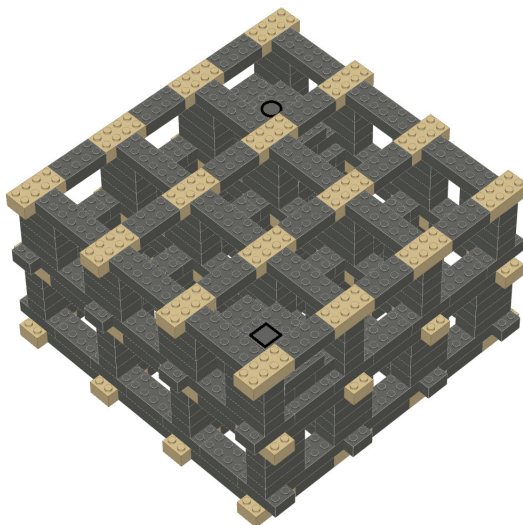
11



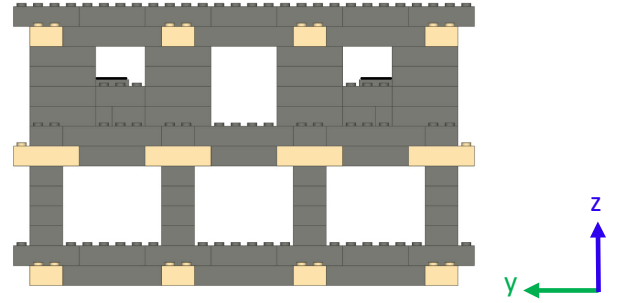
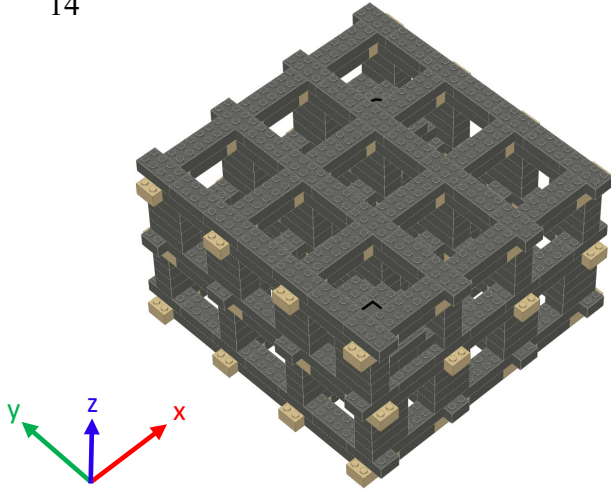
12



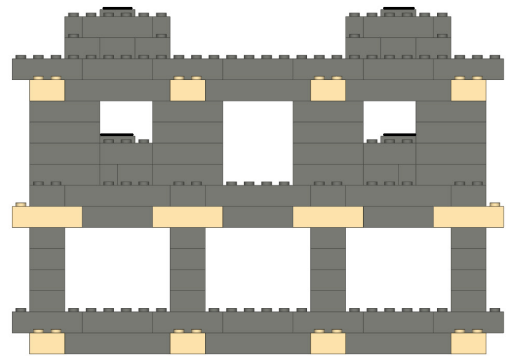
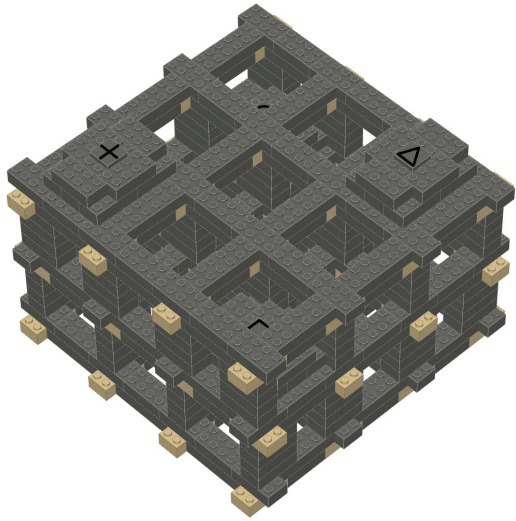
13



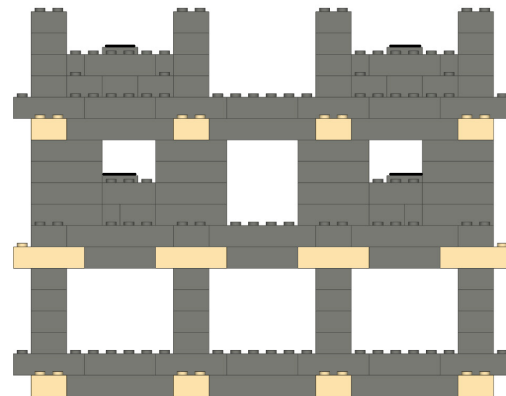
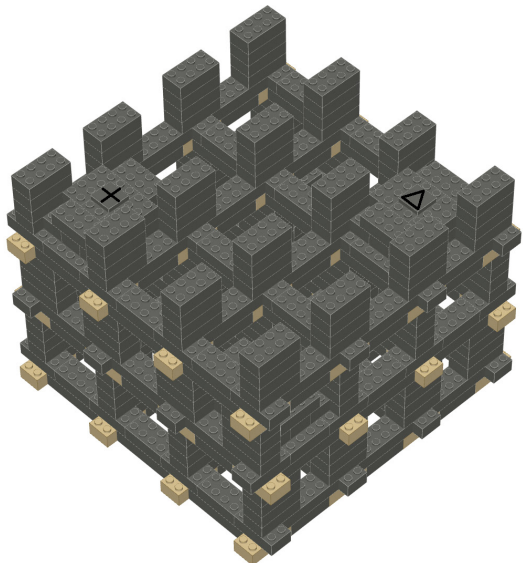
14



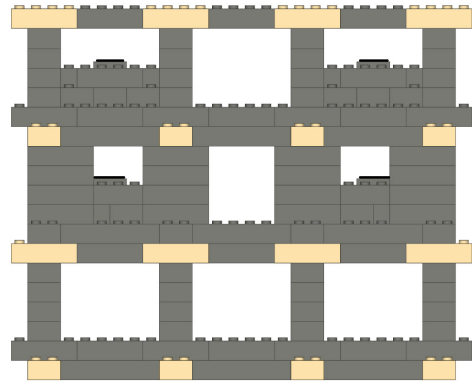
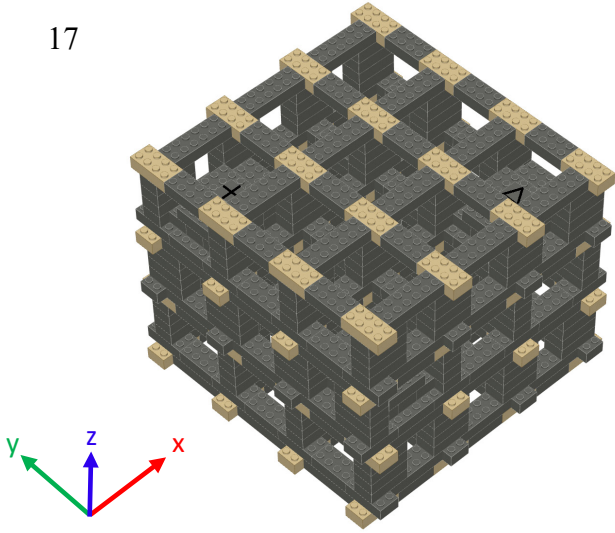
15



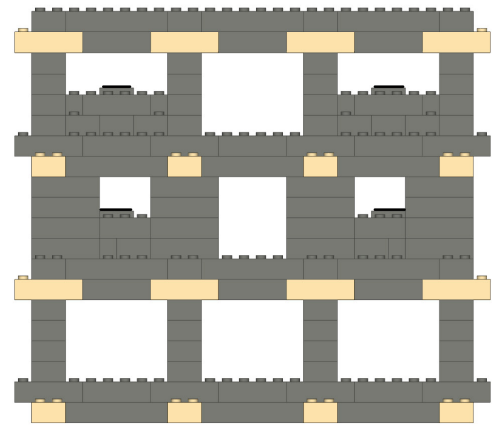
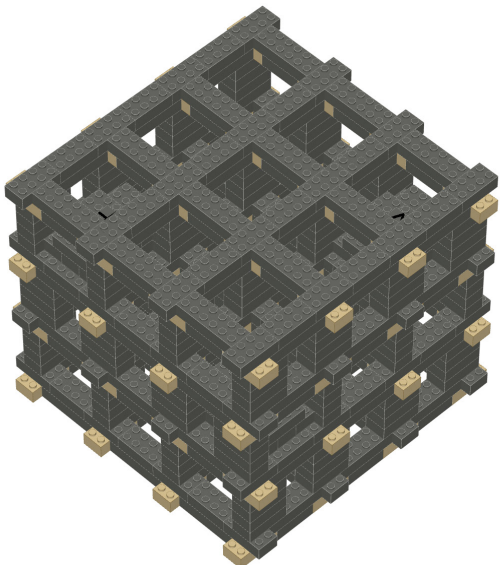
16



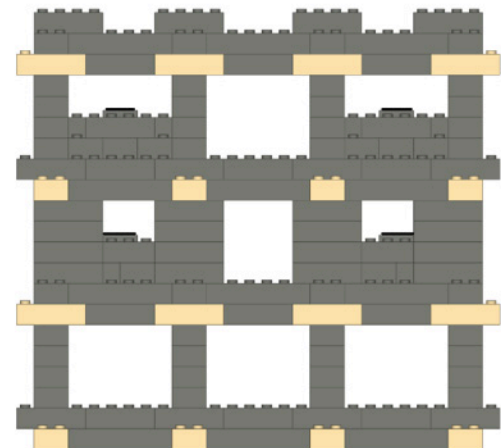
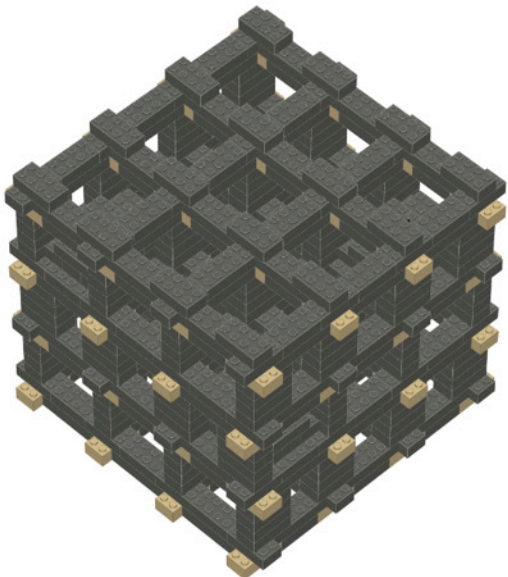
17

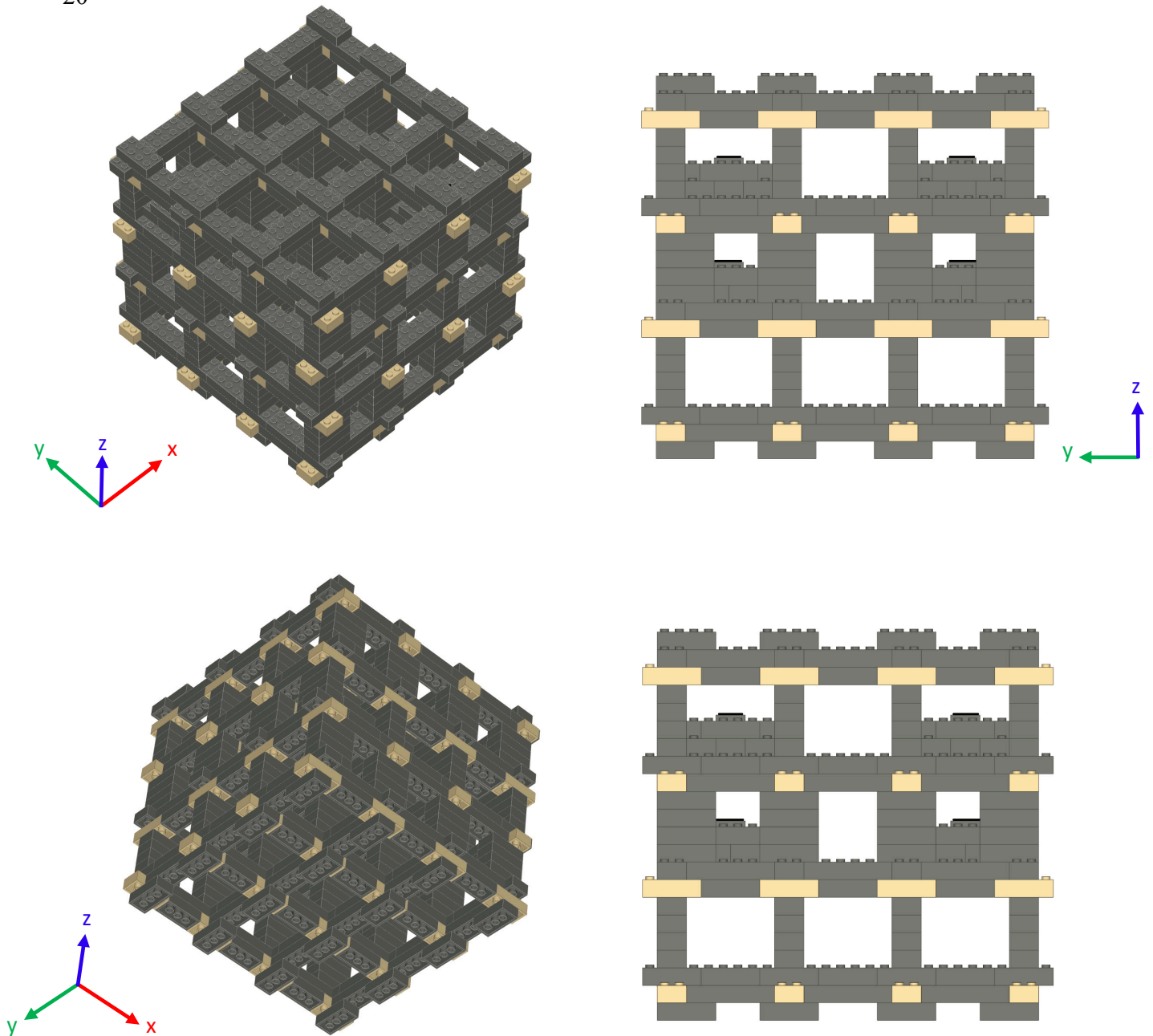


18



19





**Baronti, L., Dellepiane, M. and Scopigno, R. (2010).** Using Lego Pieces for Camera Calibration: a Preliminary Study. In *Eurographics 2010 - Short Papers*, eds. H. P. A. Lensch and S. Seipel): The Eurographics Association.

**Brainerd, E. L., Baier, D. B., Gatesy, S. M., Hedrick, T. L., Metzger, K. A., Gilbert, S. L. and Crisco, J. (2010).** X-ray Reconstruction of Moving Morphology (XROMM): precision, accuracy and applications in comparative biomechanics research. *J. Exp. Zool.* **313A**, 262-279.

## Supplementary Information 2: Repeatability of Rigid Body Motions Across Users

In this supplementary document we report repeatability results for all six degrees of freedom from the JCS analysis for nine users. To compare reproducibility among users for the rigid body transformations we imported the minipig cranium and lower jaw models into Autodesk Maya and placed a joint coordinate system (JCS) at the temporomandibular joint. We calculated the six-degree-of-freedom motions from the JCS for all nine users and then calculated the deviation from the mean for each frame for the nine users, and took the mean and standard deviation of all of those values across 435 frames.

The means and standard deviations and results of statistical tests comparing XMA Lab to XrayProject are presented in Table S3. Graphs of the data for nine users for the six degrees of freedom are presented in Figures S6-S11. The graphs include a precision threshold determined from a frozen specimen test (Menegaz et al., 2015). The threshold values are also available in Table S4.

Table S3. Repeatability XMA Lab and XrayProject for rigid body motions measured with a mandibular JCS relative to the mean value of all users (\*\*\*\*  $p < 0.0001$  )

	One-way repeated measures ANOVA F(1,15)	One-tailed paired t-test t(8)	Mean Deviation <sup>1</sup> (in mm or deg)		SD (in mm or deg)	
			XMA Lab	XrayProj.	XMA Lab	XrayProj.
Translation X	256.7 ****	15.541 ****	0.004	0.024	0.00081	0.0038
Translation Y	85.33 ****	8.4162 ****	0.011	0.061	0.0043	0.016
Translation Z	57.98 ****	7.1563 ****	0.022	0.082	0.0074	0.023
Rotation X	93.75 ****	8.8012 ****	0.012	0.055	0.0041	0.013
Rotation Y	85.13 ****	8.5546 ****	0.013	0.047	0.0042	0.011
Rotation Z	419.2 ****	18.226 ****	0.0064	0.041	0.0026	0.0043

<sup>1</sup>Mean deviation for all users across the 435 frames in the test sequence

Table S4. Precision thresholds for XMA Lab and XrayProject

	XMA Lab	XrayProject <sup>1</sup>
Translation X	0.043	0.06
Translation Y	0.13	0.26
Translation Z	0.19	0.44
Rotation X	0.089	0.26
Rotation Y	0.082	0.21
Rotation Z	0.067	0.13

<sup>1</sup> from Menegaz et al., 2015

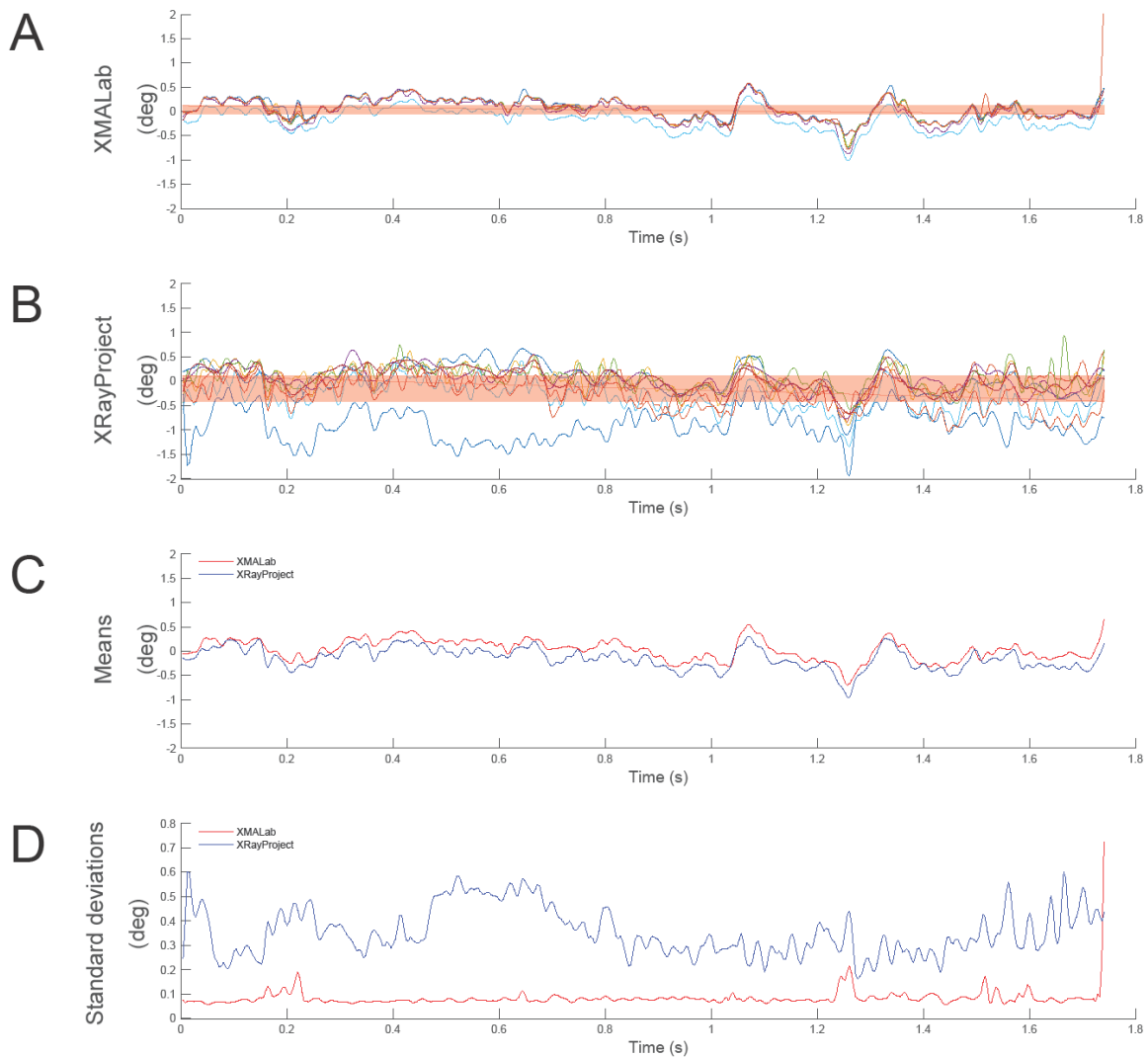


Figure S6. Variation among users in x-axis rotation (in degrees) from a joint coordinate system measuring motion of the mandible relative to the cranium in a miniature pig (Menegaz et al., 2015). (A) Results from XMA Lab for nine users. Line for each user is a different color. The precision threshold from a frozen specimen test for this degree of freedom is shown as a salmon-colored bar. (B) Results from XrayProject for nine users. Line for each user is a different color. Salmon-colored bar shows precision threshold. (C) Mean result for all users for the two programs; XMA Lab in red and XrayProject in blue. (D) Standard deviation for each time point among users; XMA Lab in red and XrayProject in blue.



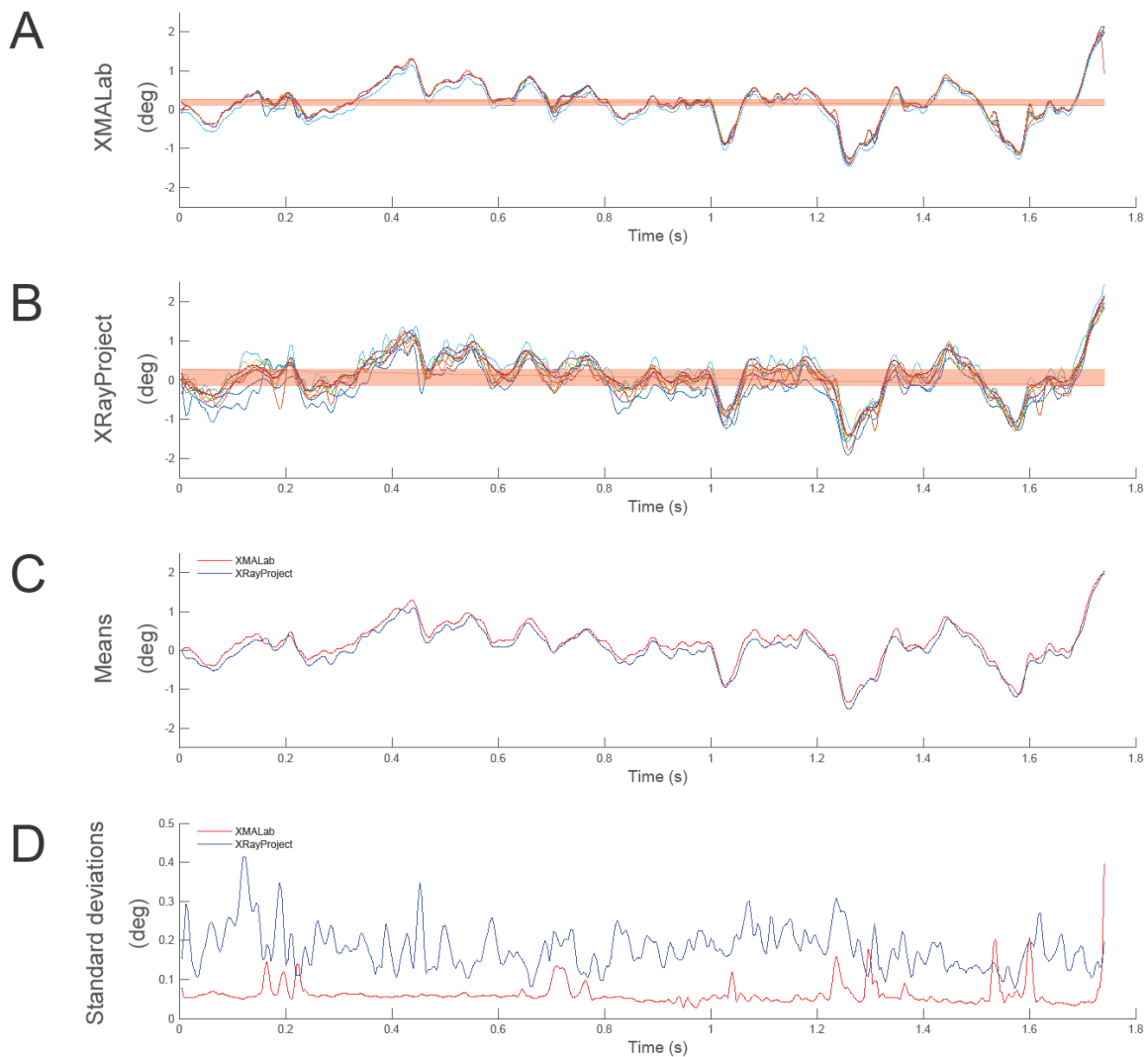


Figure S7. Variation among users in y-axis rotation (in degrees) from a joint coordinate system measuring motion of the mandible relative to the cranium in a miniature pig (Menegaz et al., 2015). (A) Results from XMALab for nine users. Line for each user is a different color. The precision threshold from a frozen specimen test for this degree of freedom is shown as a salmon-colored bar. (B) Results from XrayProject for nine users. Line for each user is a different color. Salmon-colored bar shows precision threshold. (C) Mean result for all users for the two programs; XMALab in red and XrayProject in blue. (D) Standard deviation for each time point among users; XMALab in red and XrayProject in blue.

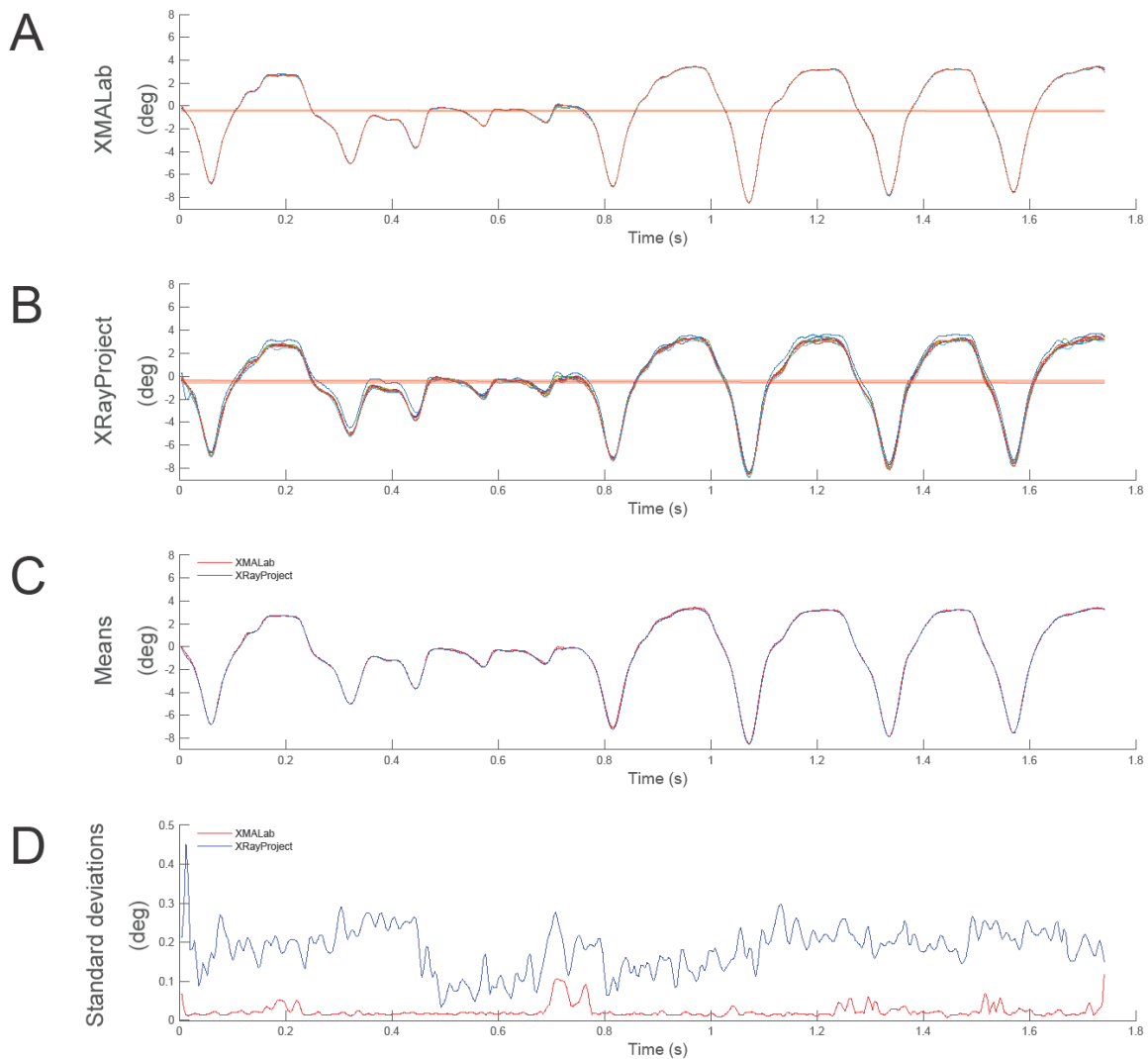


Figure S8. Variation among users in z-axis rotation (in degrees) from a joint coordinate system measuring motion of the mandible relative to the cranium in a miniature pig (Menegaz et al., 2015). (A) Results from XMA Lab for nine users. Line for each user is a different color. The precision threshold from a frozen specimen test for this degree of freedom is shown as a salmon-colored bar. (B) Results from XrayProject for nine users. Line for each user is a different color. Salmon-colored bar shows precision threshold. (C) Mean result for all users for the two programs; XMA Lab in red and XrayProject in blue. (D) Standard deviation for each time point among users; XMA Lab in red and XrayProject in blue.

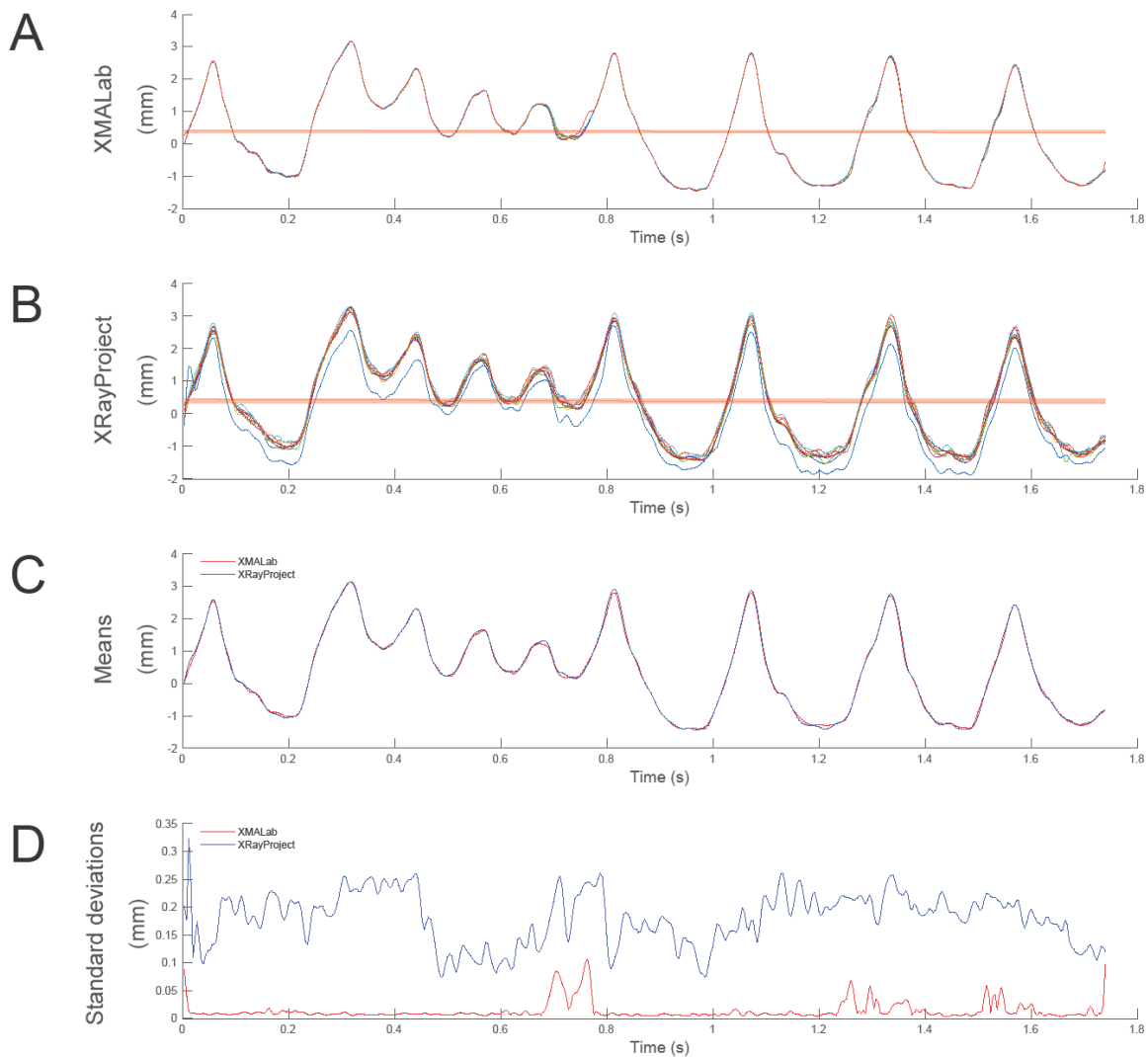


Figure S9. Variation among users in x-axis translation (in mm) from a joint coordinate system measuring motion of the mandible relative to the cranium in a miniature pig (Menegaz et al., 2015). (A) Results from XMALab for nine users. Line for each user is a different color. The precision threshold from a frozen specimen test for this degree of freedom is shown as a salmon-colored bar. (B) Results from XrayProject for nine users. Line for each user is a different color. Salmon-colored bar shows precision threshold. (C) Mean result for all users for the two programs; XMALab in red and XrayProject in blue. (D) Standard deviation for each time point among users; XMALab in red and XrayProject in blue.

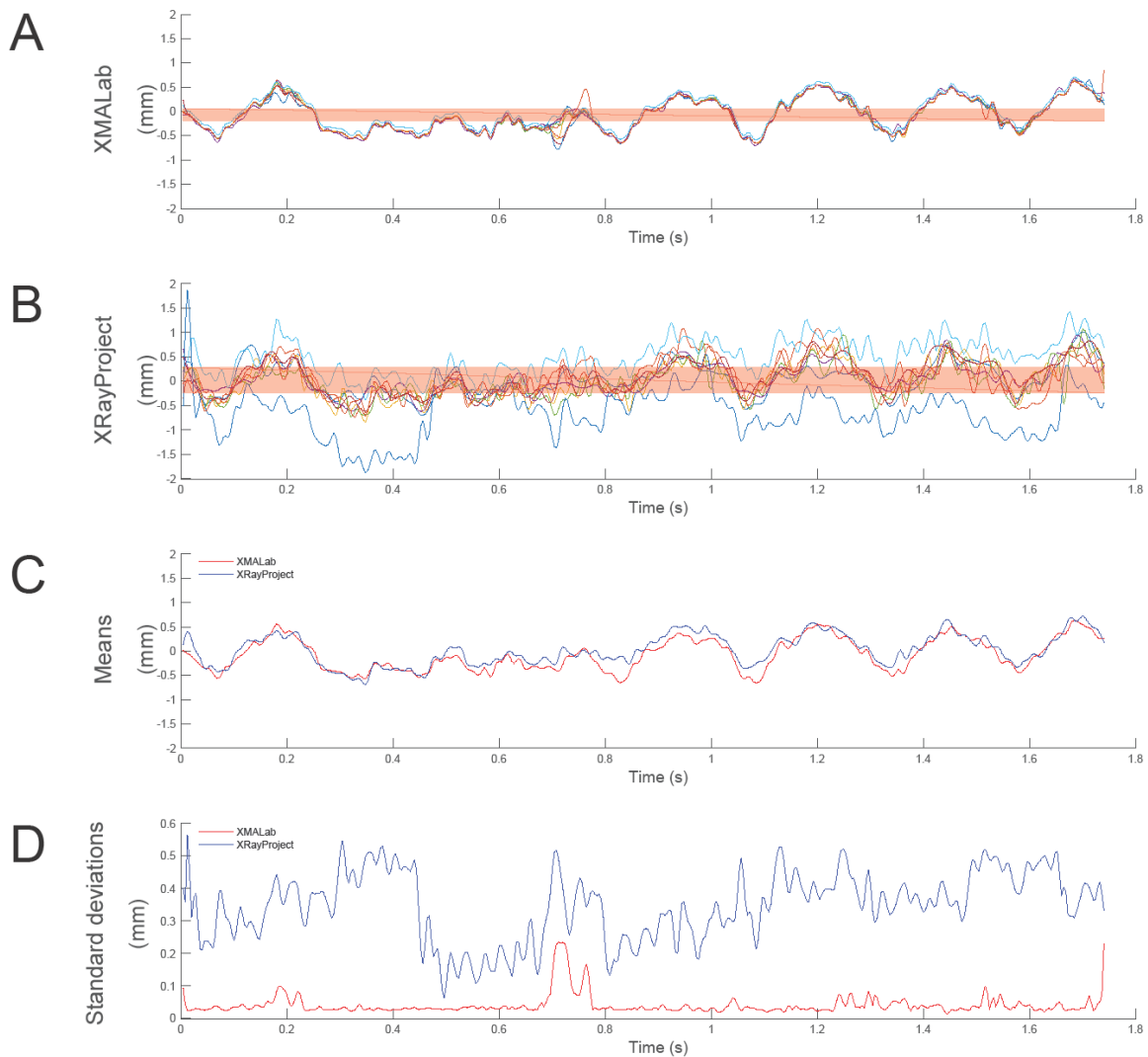


Figure S10. Variation among users in y-axis translation (in mm) from a joint coordinate system measuring motion of the mandible relative to the cranium in a miniature pig (Menegaz et al., 2015). (A) Results from XMALab for nine users. Line for each user is a different color. The precision threshold from a frozen specimen test for this degree of freedom is shown as a salmon-colored bar. (B) Results from XrayProject for nine users. Line for each user is a different color. Salmon-colored bar shows precision threshold. (C) Mean result for all users for the two programs; XMALab in red and XrayProject in blue. (D) Standard deviation for each time point among users; XMALab in red and XrayProject in blue.

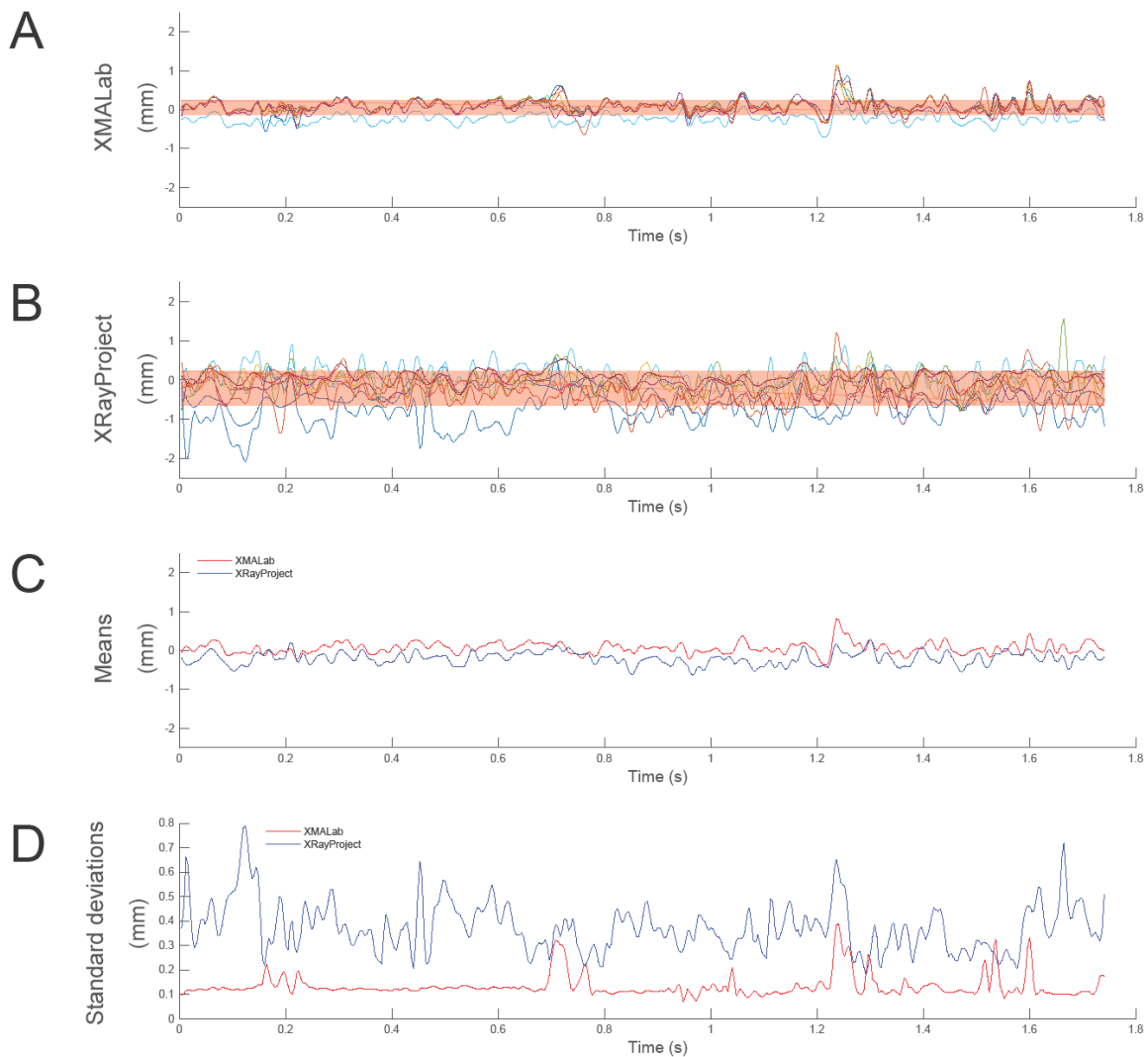


Figure S11. Variation among users in z-axis translation (in mm) from a joint coordinate system measuring motion of the mandible relative to the cranium in a miniature pig (Menegaz et al., 2015). (A) Results from XMALab for nine users. Line for each user is a different color. The precision threshold from a frozen specimen test for this degree of freedom is shown as a salmon-colored bar. (B) Results from XrayProject for nine users. Line for each user is a different color. Salmon-colored bar shows precision threshold. (C) Mean result for all users for the two programs; XMALab in red and XrayProject in blue. (D) Standard deviation for each time point among users; XMALab in red and XrayProject in blue.

**Menegaz, R. A., Baier, D. B., Metzger, K. A., Herring, S. W. and Brainerd, E. L. (2015).** XROMM analysis of tooth occlusion and temporomandibular joint kinematics during feeding in juvenile miniature pigs. *J Exp Biol* **218**, 2573-84.