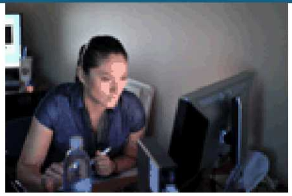
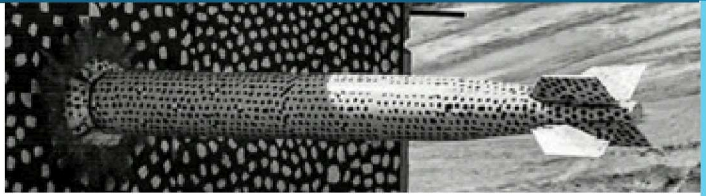
SAND2020-1457C

Predicting 3D Motions from Single-Camera Optical Data





IMAC XXXVIII

Dan Rohe and Bryan Witt

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Motivation

Is there a way that we can extract 3D motions from a set of 2D images?

For example:

Cities are generally full of cameras; if 3D motions could be extracted from a single camera dataset, engineers could make predictions of damage due to events such as earthquakes, dramatically increasing the ability to surveil civil structures.

Even in a laboratory setting, it may be difficult to obtain stereo-calibrated images for some testing methods, perhaps for radiographic images.

When a pair of cameras is available, the most efficient use may be to view multiple sides of a test article rather than pointing both at a single surface.

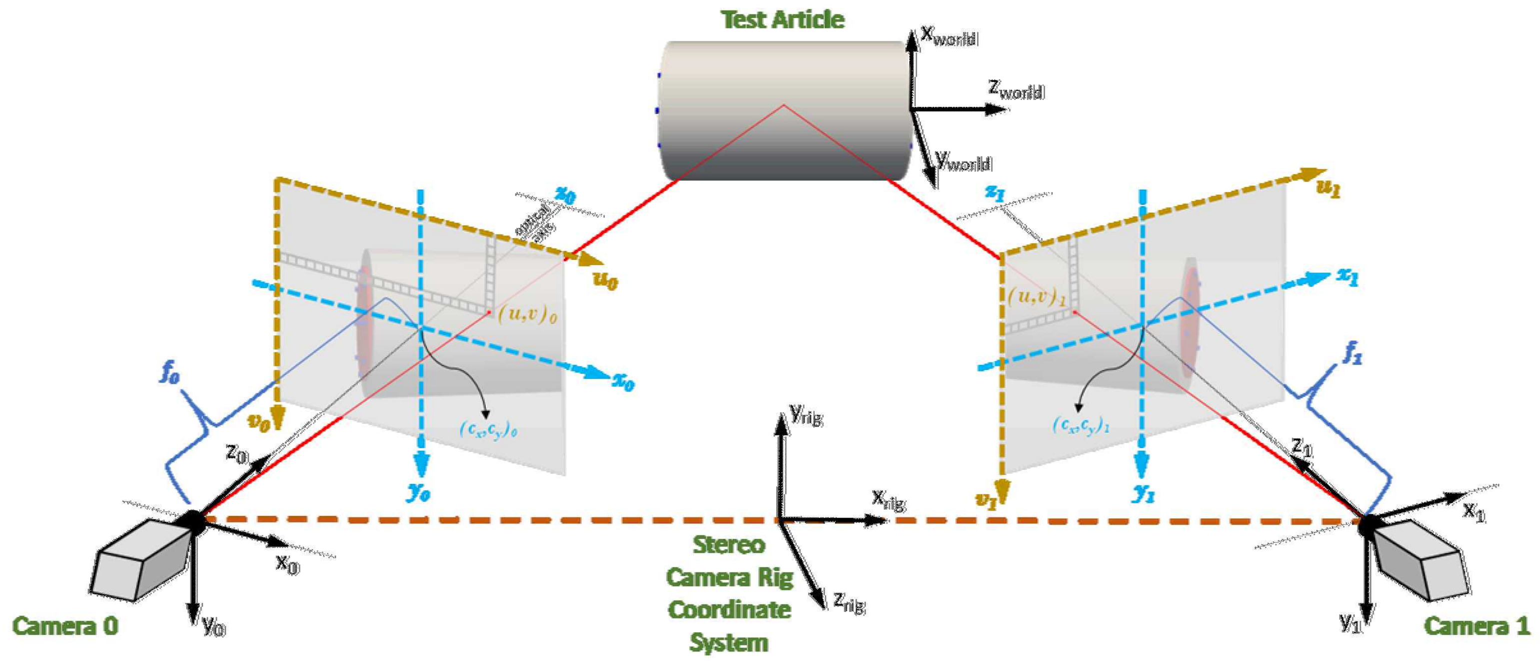
3D Motions from Single Camera Test Setup

Typically stereo cameras are needed to extract 3D motions from an optical test.

A single camera will only be able to identify motions in the image plane; motions perpendicular will be missed, or at least be ambiguous.

Given knowledge of the test article and applied excitation, we can infer out-of-plane motions from single camera images.

More precisely, we can treat a single camera image as a case of measured degrees of freedom, and use finite element expansion techniques to expand from measured degrees of freedom to finite element degrees of freedom.



Finite Element Expansion using SEREP

System Equivalent Reduction Expansion Process (SEREP)[1] uses a spatial filter to identify which shapes are active in a given displacement.

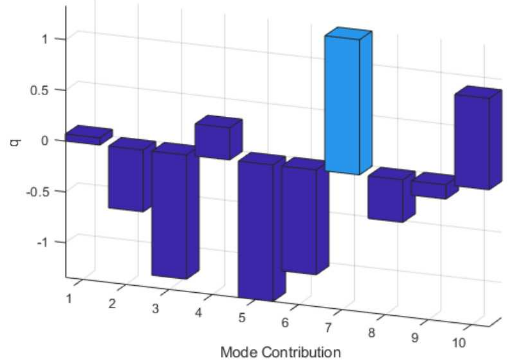
$$\mathbf{x}_a = \mathbf{\Phi}_a \mathbf{q} \qquad \mathbf{q} = \mathbf{\Phi}_a^+ \mathbf{x}_a$$

Then by multiplying this coefficient by those same shapes (but at a expanded set of degrees of freedom), predictions can be made at other degrees of freedom

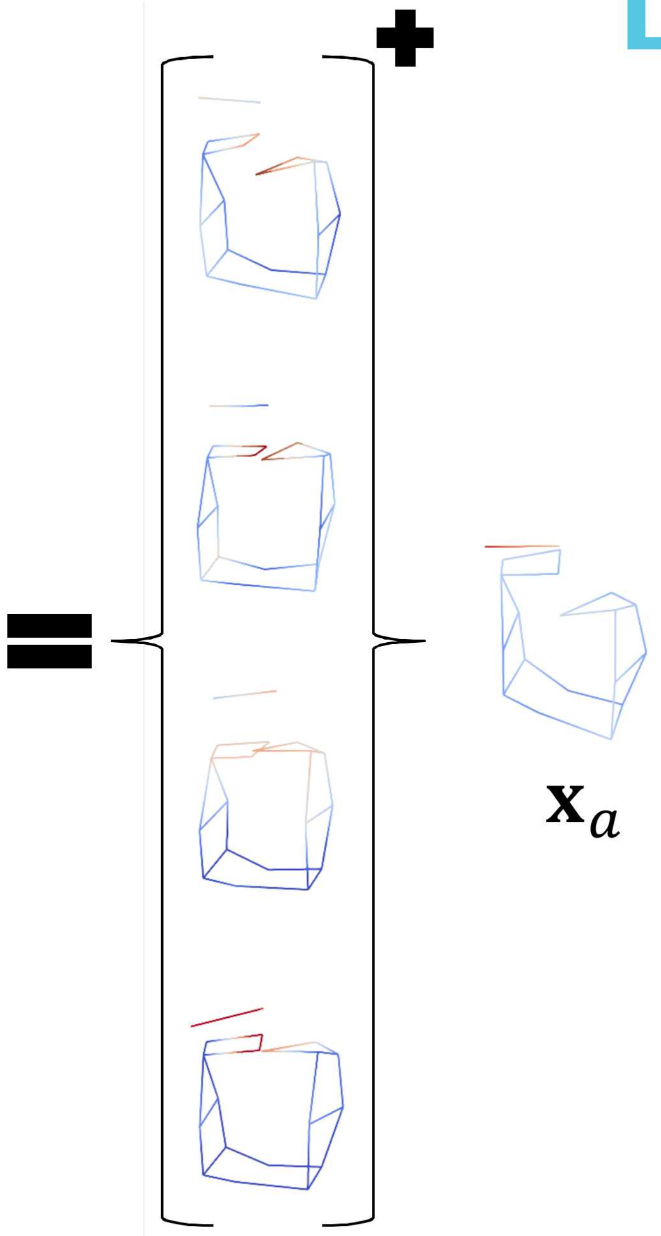
$$\mathbf{x}_n = \mathbf{\Phi}_n \mathbf{q}$$

This results in the well-known SEREP equation

$$\mathbf{x}_n = \mathbf{\Phi}_n \mathbf{\Phi}_a^+ \mathbf{x}_a$$



$\mathbf{q}$



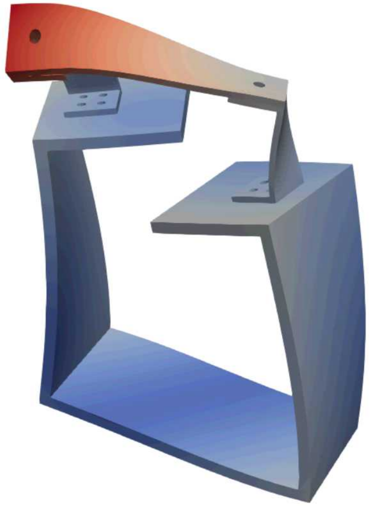
$\mathbf{\Phi}_a$ $\mathbf{x}_a$

[1] System Equivalent Reduction Expansion Process (SEREP), J.O'Callahan, P.Avitabile, R.Riemer, Seventh International Modal Analysis Conference, Las Vegas, Nevada, February 1989

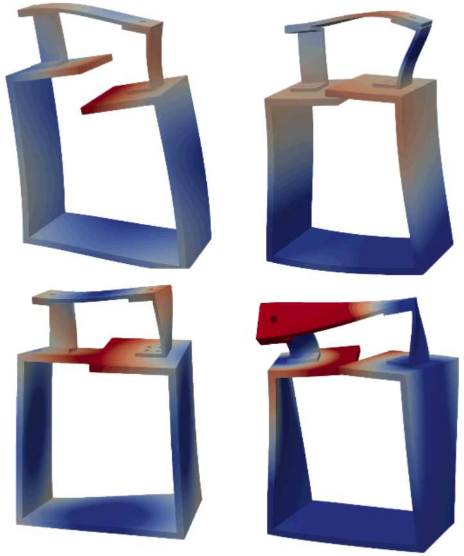
Casting the 3D response from 2D image problem into SEREP

$$\mathbf{x}_n = \mathbf{\Phi}_n \mathbf{\Phi}_a^{+} \mathbf{x}_a$$

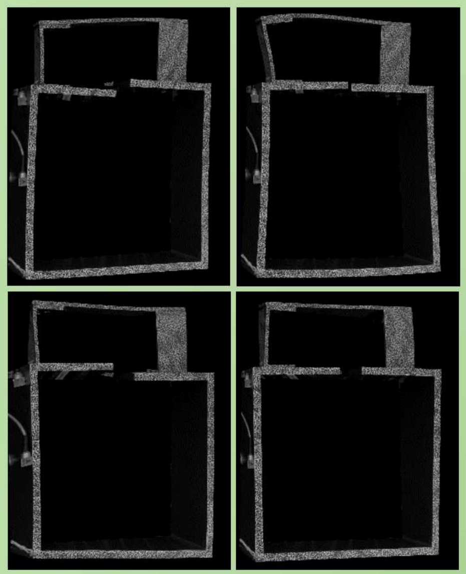
3D, Full-Field Motions



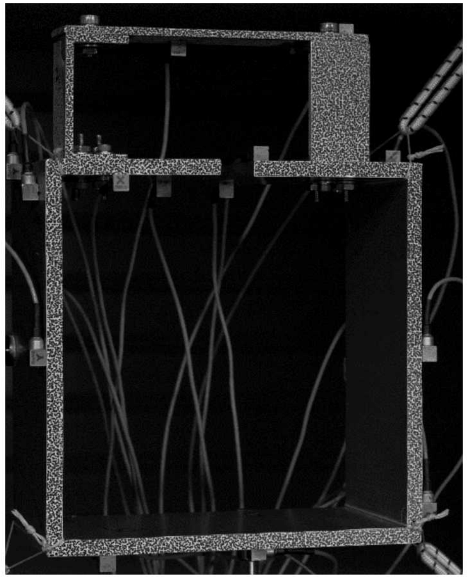
Finite Element Shapes in 3D Space



Finite Element Shapes in Image Space?



A series of images



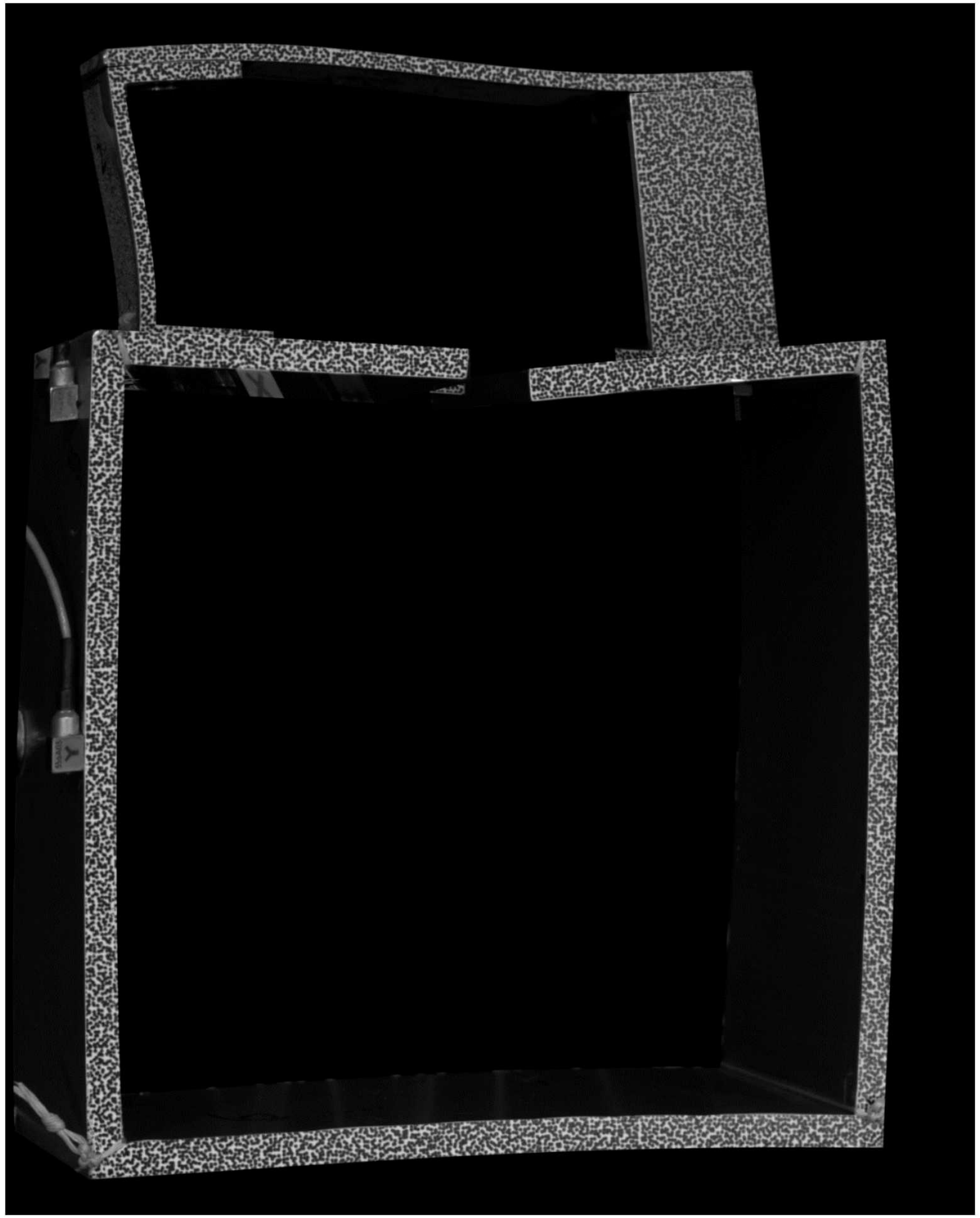
Generating Synthetic Finite Element Images of Modeshape Deflections

Synthetic Images are created using the open-source Blender software.

- Has both graphical and Python programming interface
- Blender uses surfaces to define geometry; volumetric finite element blocks should be skinned.
- For more information:
  - An Optical Test Simulator Based on the Open-Source Blender Software #8013, Thursday 12:10 p.m.

Steps to generate synthetic finite element images:

- Import the mesh into the render software
- Set up cameras in-software to mimic the experimental cameras
  - 2D calibration gives intrinsic parameters
  - Perspective-n-Point algorithm gives position and orientation
- Create a local texture pattern on the image
  - Project test image to mesh using camera equations
- Deform mesh into each mode shape and render image
  - Shapes deformed at 0.5 pixel peak deflection
  - Images shown at 50 pixel deflection (100x actually used) so deflections are



Degrees of freedom extracted from mode shape images

26 mode shapes (6 rigid body and 20 elastic) were rendered on the image at 0.5 px peak deflection.

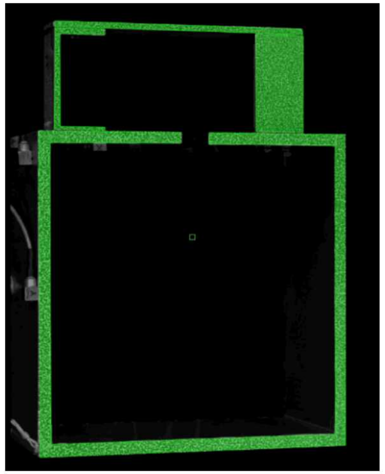
From each rendered shape image, subsets were defined that would be tracked using 2D digital image correlation.

814 subsets (21x21 pixel, step size 20) with 2 degrees of freedom

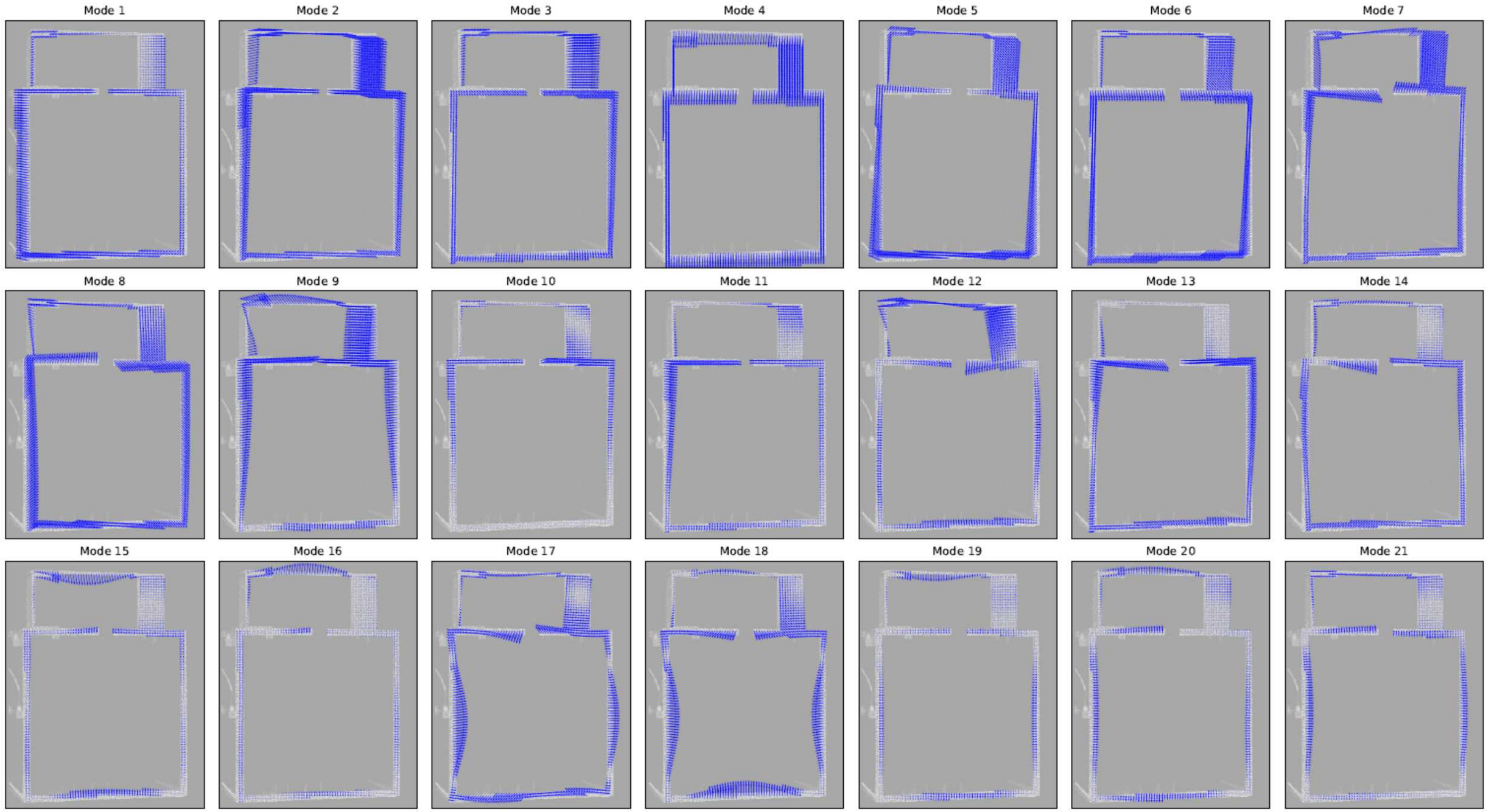
Subset displacements from first 26 modes were assembled into a 1628x26 $\boldsymbol{\Phi}_a$ matrix.

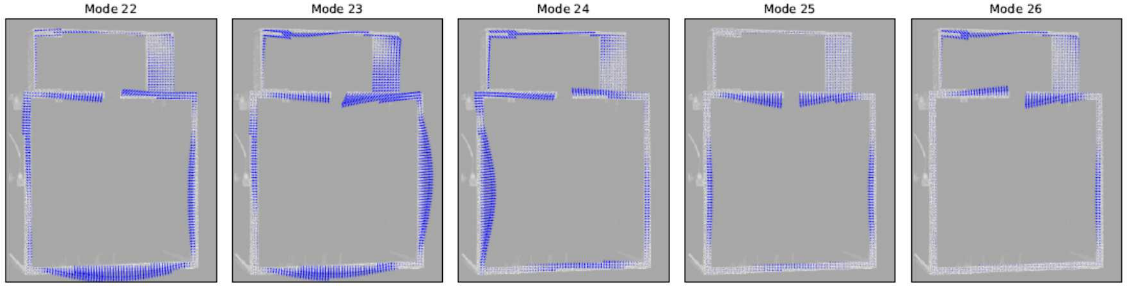
- Condition number: 148.

Identical subsets will be used on the response images.

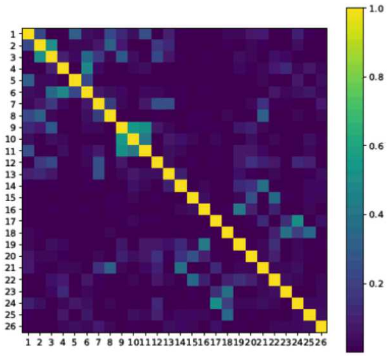


Areas of Interest in DIC analysis





Subset displacements for first 26 modes.



Expanding from a Synthetic Deformation Image

A synthetic deformation image was created to test the technique.

- Assembled from linear combination of mode shapes 1-20 scaled randomly between -0.3 and 0.3 pixels for each mode.

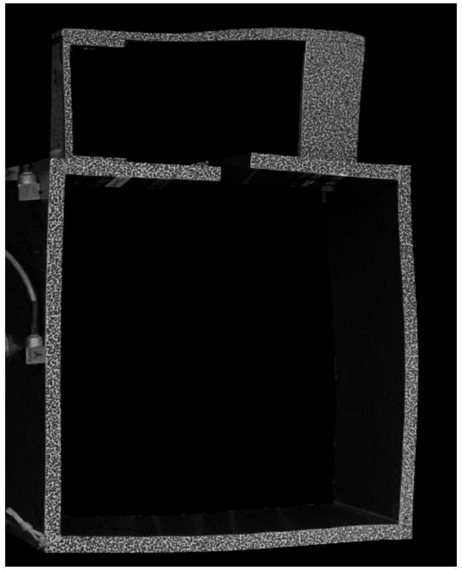
$\mathbf{q}$s were estimated using SEREP to determine whether or not they could be recovered using the technique.

Prescribed and Estimated full-field displacements reconstructed by $\mathbf{x}_n = \mathbf{\Phi}_n \mathbf{q}$

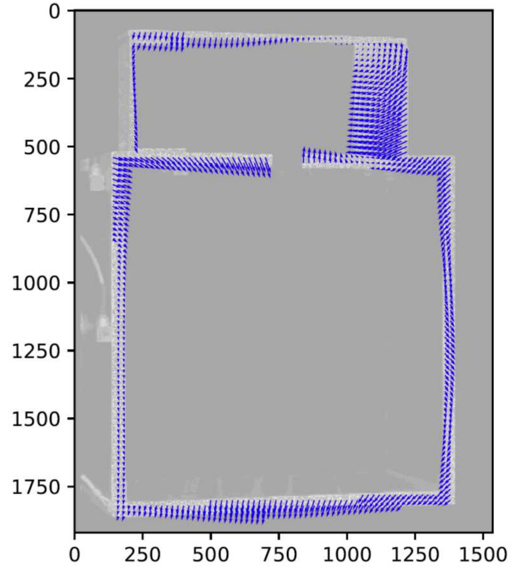
Errors approximately 5% on $\mathbf{q}$ and 3% on $\mathbf{x}_n$, noting that the majority of this error occurs far away from the measurement degrees of freedom.

Errors in $\mathbf{q}$ and $\mathbf{x}_n$ investigated as a function of pixel displacement in the rendered mode shape images: minimum near 0.5 px

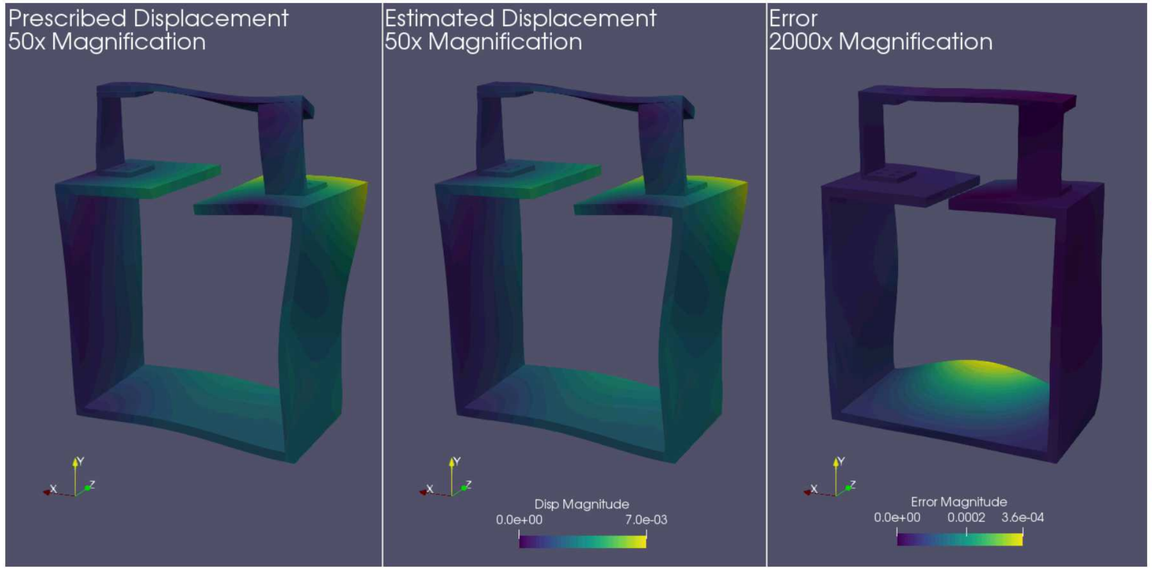
Pixel Displacement	Error Magnitude
3	0.1252
1	0.0725
0.5	0.0580
0.25	0.1422
0.125	0.2010



Synthetic Deformation Image rendered 50x actual displacement



2D subset displacements from 2D DIC at a-set degrees of freedom



Validation using Experiment

Stereo DIC setup to provide truth data

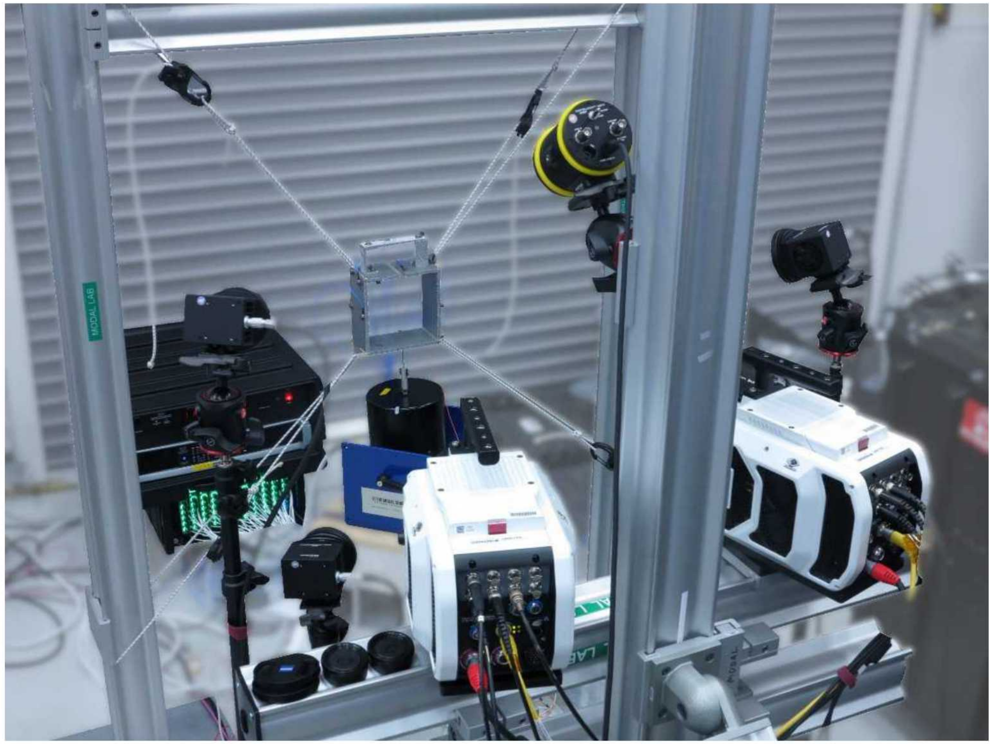
Expansion uses just left camera

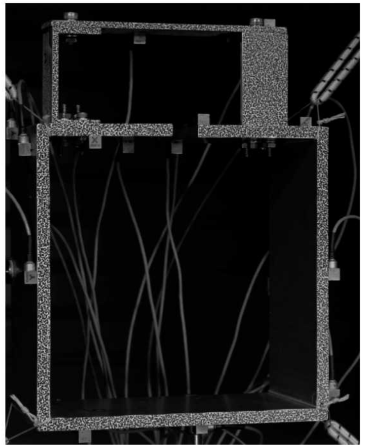
Excitation:
- Pseudorandom Flat-Force, 100-2000 Hz
- 1 lb RMS
- Approximately 0.3 px displacement.

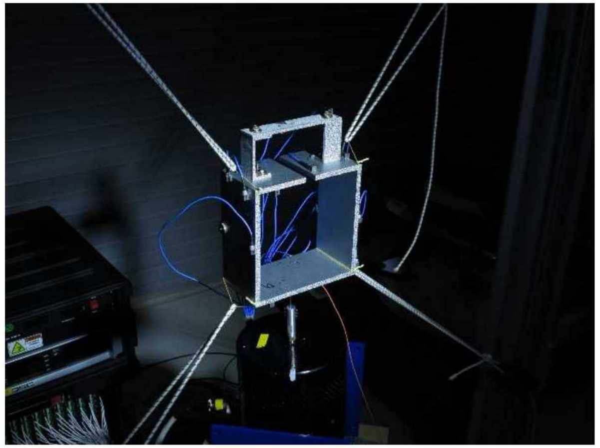
Sampling Parameters
- Sample Rate: 4096 Hz
- Frame Length: 1 s
- Exposure Time: 50 μs
- 10 Averages
- 40960 images stored

Processing:
- Image averaging to reduce 40960 to 4096 images.

Accelerometers and force measurements were also obtained, used for diagnostics.





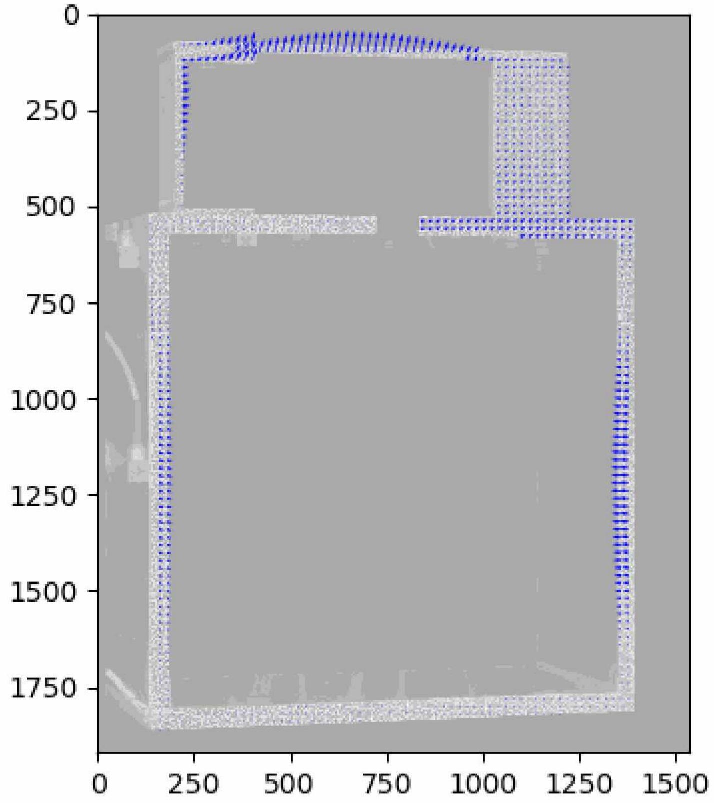


Subset Extraction

2D DIC was performed and subsets deformations were extracted from each time step.

Time data was high-pass filtered at 100 Hz using 4th order Butterworth Filter to remove distortions due to heat waves and other experimental errors.

Time data was assembled into a 1628x4096 matrix $\mathbf{x}_a$ and the SEREP expansion was performed to extract 3D motions from the 2D responses.



Comparison to Truth Data: Overview

60 locations were selected to compare response against truth data.

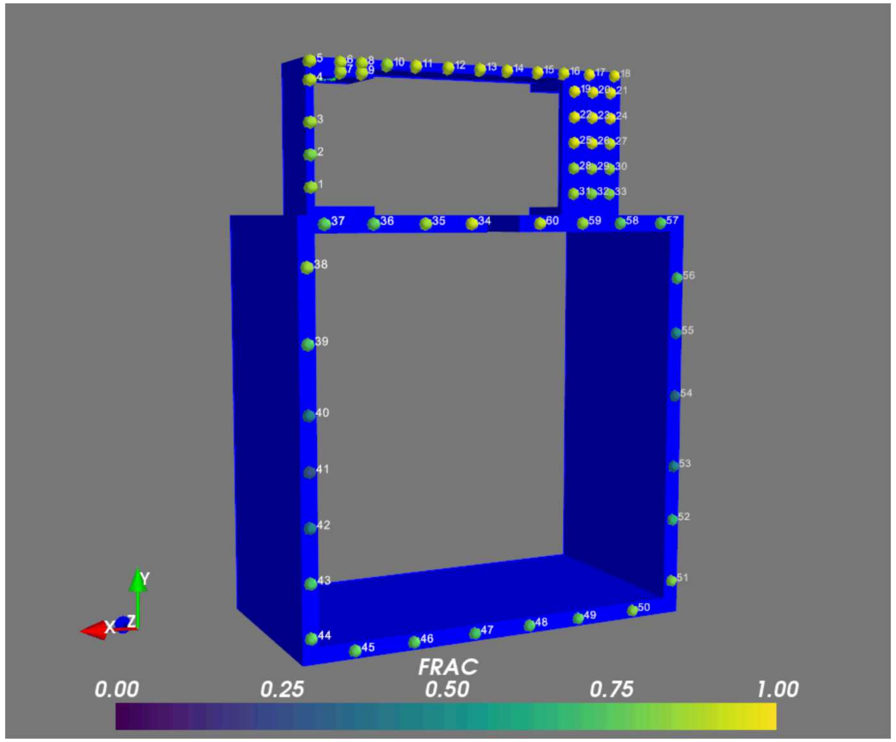
TRAC and FRAC were computed to compare time and frequency responses.

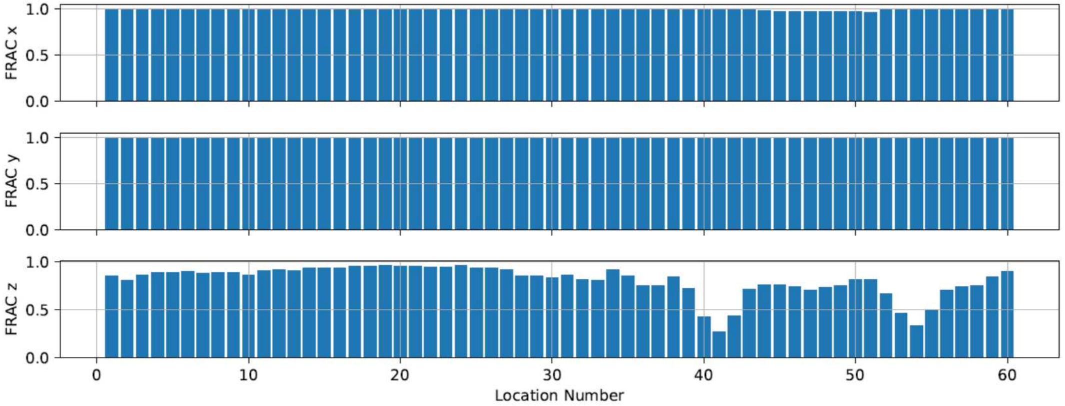
Responses in the x and y (closely aligned with image plane) directions were nearly perfectly matched.

Responses in the z direction did not match as well at all degrees of freedom.

Largest errors were on the vertical portions of the "box".

Best matches were on the removable component.



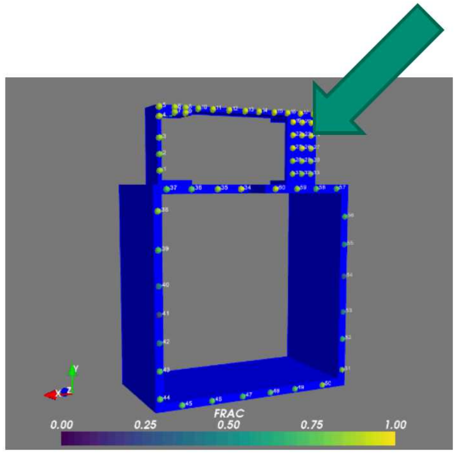


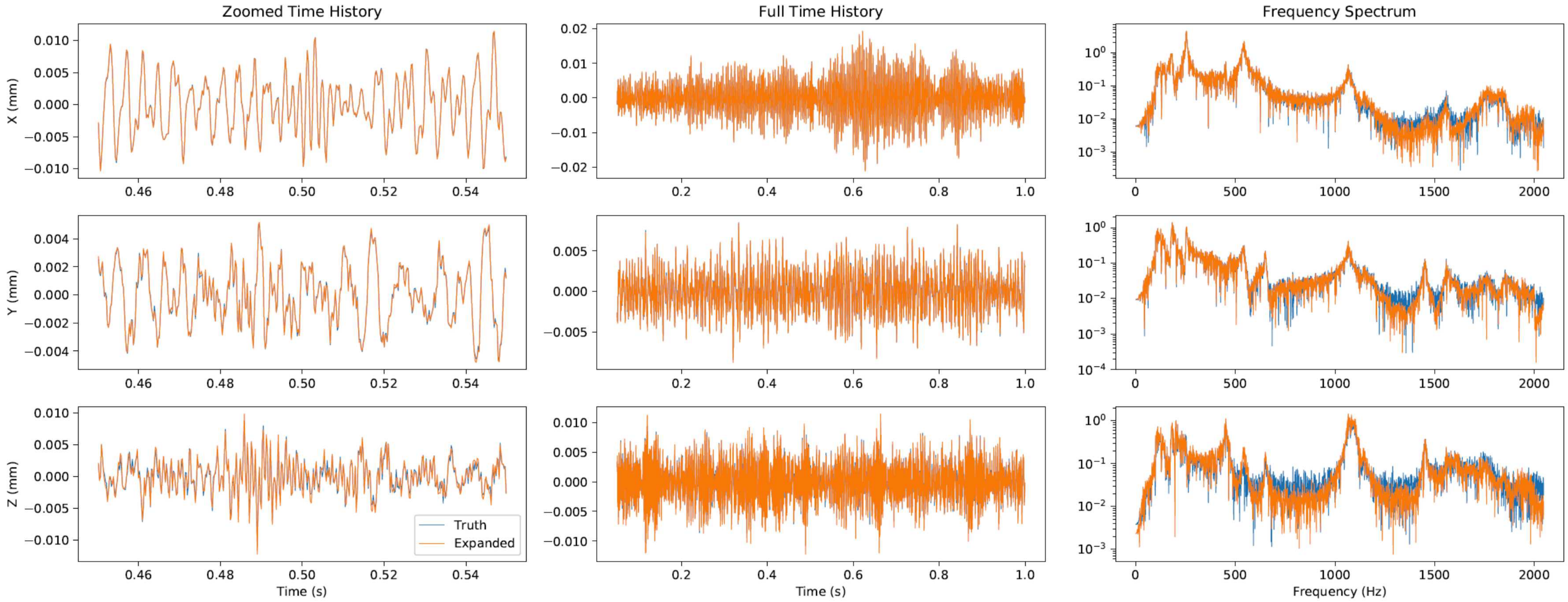
Comparison to Truth Data: Best Match

Best results were found on the removable component.

Predicted motions were nearly line-on-line for all components of response.

Only differences found in valleys of the response spectra (dominated by noise).





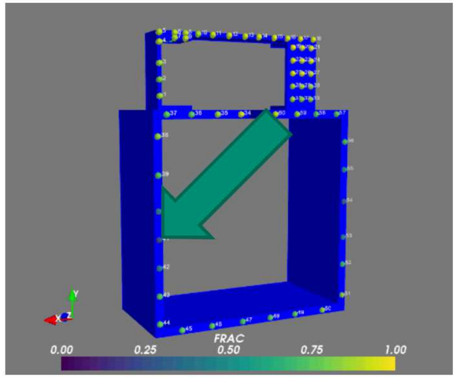
Comparison to Truth Data: Worst Match

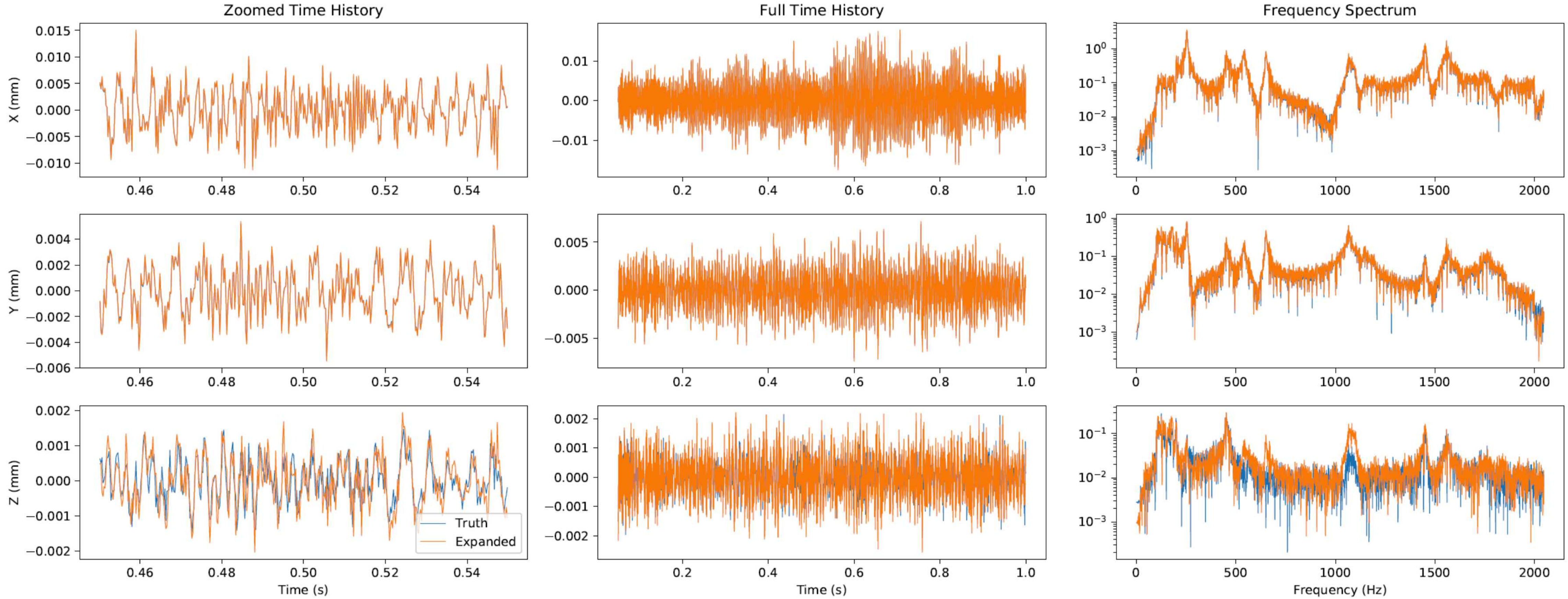
Worst results found on the vertical of the box.

Results look better than a FRAC of <0.5 implies.

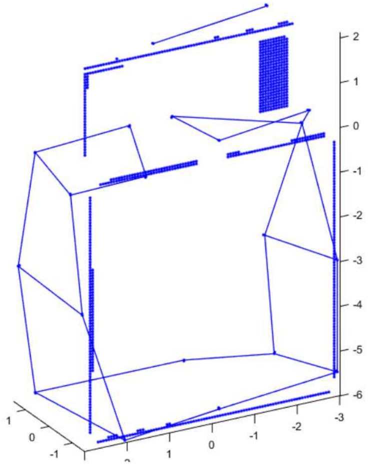
Errors primarily at the 1000-1200 Hz frequency band.

Note the Z response is only 1/5 of the magnitude at the best predicted locations.





What's with the 1000-1200 Hz band?



SEREP can incorrectly expand when:

- The FEM is not well-correlated to the structure, so the mode shapes are not good interpolants to the measured data.
- The degrees of freedom are insufficient or too many modes are used.

SEREP includes least squares smoothing of experimental error

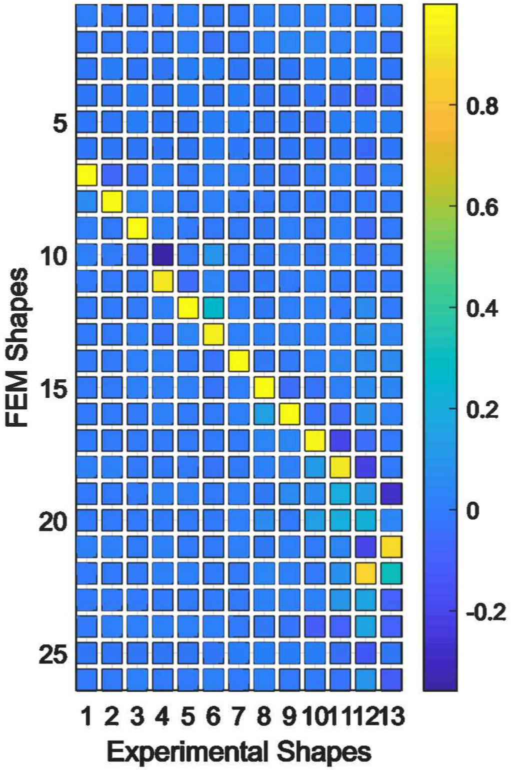
- If too many modes or insufficient degrees of freedom are used, motion will be fit to the errors rather than being smoothed out.

Test expansion using stereo DIC and Accelerometer data.

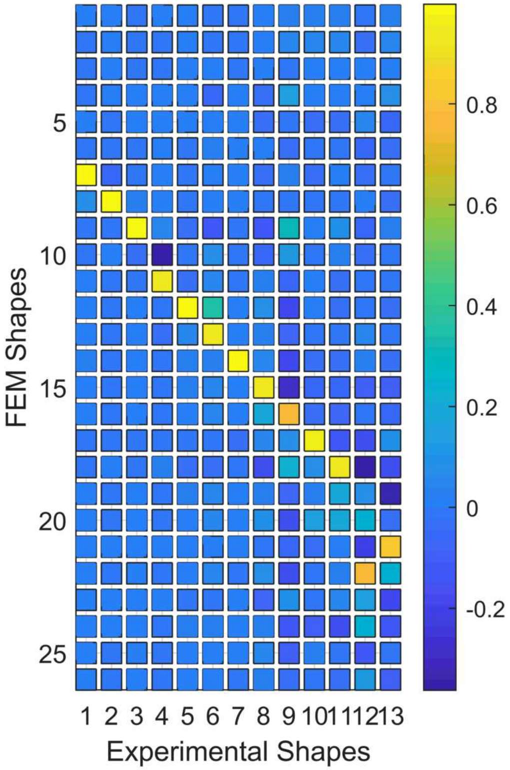
- Fit modes to accelerometer and stereo DIC data.
- Expand mode shapes using SEREP using different sets of degrees of freedom
  - Accelerometer + DIC: data spread across part, provides "truth" data when expanded.
  - DIC: data limited to front surface, provides case close to experiment

Modes 8 and 9 were found to change q composition when the accelerometers were removed from the expansion.

- Suggests that the degrees of freedom are not quite sufficient to fit these modes accurately
- These modes have frequencies at 1058 and 1123 Hz.



qs with accels



qs without accels

Overcoming Limited Degrees of Freedom: Multi-View Analysis

All degrees of freedom on a few surfaces may not be sufficient for expansion.

- Depends on test article

Procedure can easily be modified to accommodate multiple views:

- Must calibrate all cameras (2D calibration)
- Must estimate position and orientation of all cameras (PnP)
- Must generate synthetic images for all cameras for all mode shapes
- Additional DoFs are simply appended as extra rows to Φ_a and x_a.
- Expand to 3D shapes as usual.

All this work to not use two cameras, now you're telling me to use two cameras?!

- Stereo DIC on just a few surfaces may also not be sufficient to understand all the mode shapes of the system.
- Testing usually performed to understand what the **entire part** is doing, not just instrumentation locations.
- May be a better use of your two cameras to **not** use a stereo pair, but instead view two different surfaces and do 2D DIC on each.
- Addition of a few accelerometers may also be sufficient

Conclusions

Technique developed to expand 2D image data to 3D motion using synthetic mode shape images as a basis for the transformation.

Developed techniques to replicate test images in rendering software so that camera parameters and local contrast on the part is reproduced.

- Utilized 2D camera calibration to estimate intrinsic parameters
- Utilized PnP algorithms to estimate position and orientation
- Experimental reference image applied as texture using camera equations to define the projection of the 3D model to the image.

2D DIC performed on experimental response images and synthetic finite element images to extract measured (a-set) degrees of freedom.

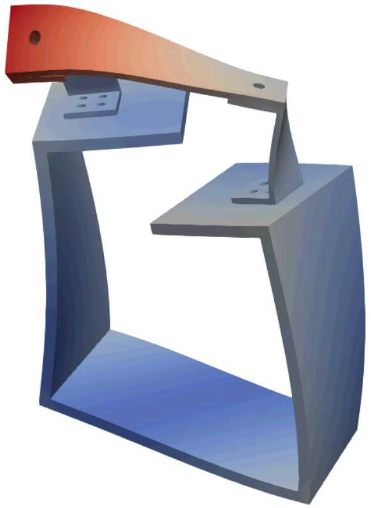
- Experimental response displacements filtered by mode shape image displacements to produce modal coefficients q
- Modal coefficients multiplied by full finite element space finite element shapes to estimate full field responses from image data.

Responses showed good match to 3D DIC if data has been high-pass filtered to remove low frequency errors thought to arise from heat waves.

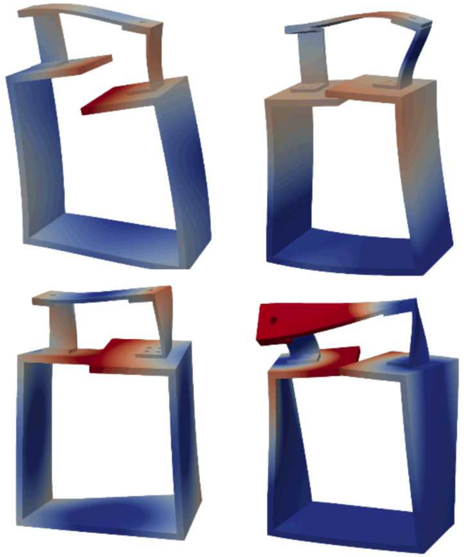
Questions?

$$\mathbf{x}_n = \boldsymbol{\Phi}_n \boldsymbol{\Phi}_a^+ \mathbf{x}_a$$

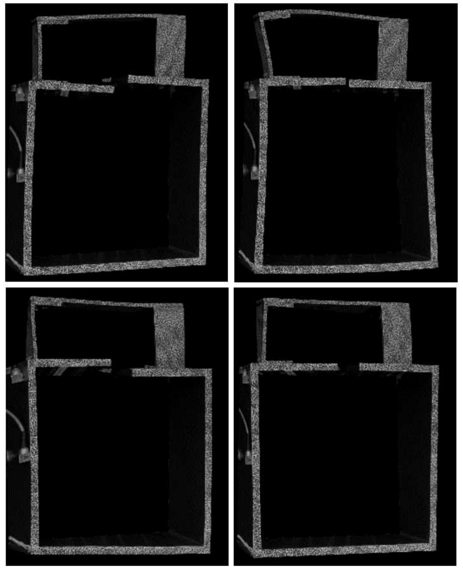
3D, Full Field Motions



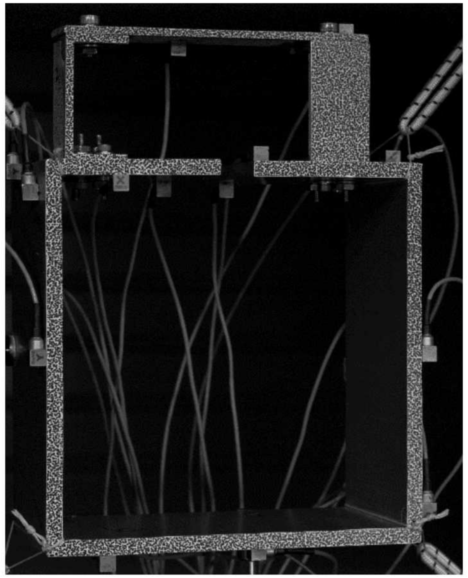
Finite Element Shapes in 3D Space



Finite Element Shapes in Image Space



A series of images



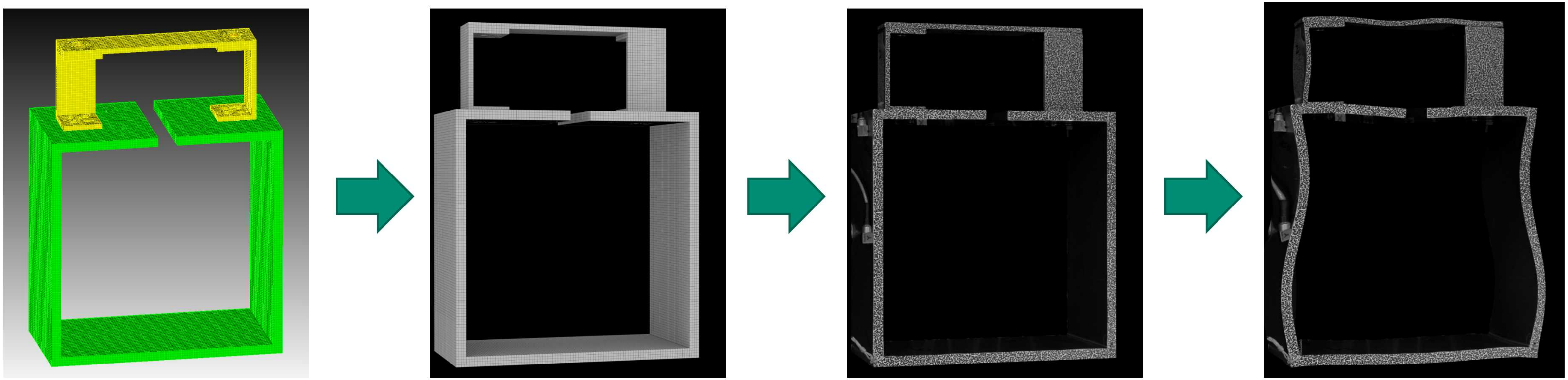
Creating Synthetic Finite Element Shape Images

The process to reproduce finite element shapes on-image is not trivial.

- Export the finite element model to rendering software.
- Set up cameras in-software to mimic the experimental cameras
- Create a local texture pattern on the image
- Deform mesh into each mode shape and render image

Synthetic Images are created using the open-source Blender software.

- Has both graphical and Python programming interface



Blender can be downloaded at: https://www.blender.org/

Preparing and Importing the Finite Element Model

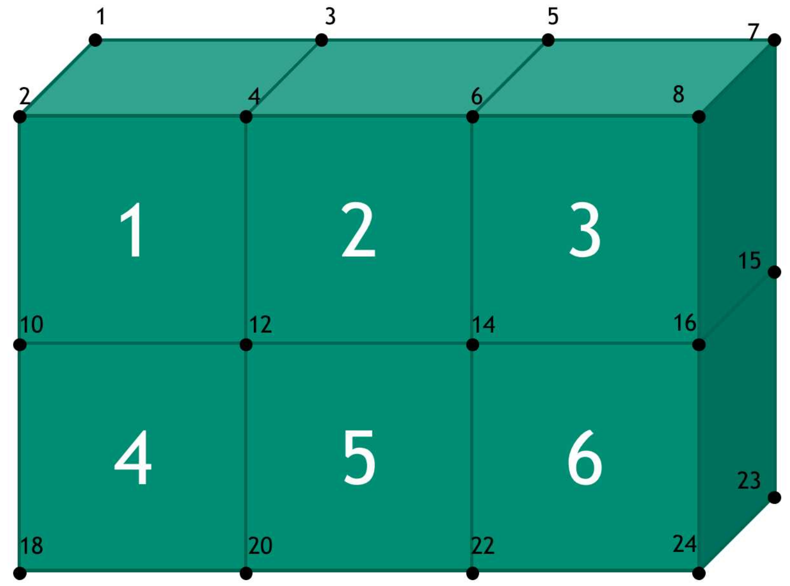
Blender uses surface meshes, rather than volume meshes, to define geometry.

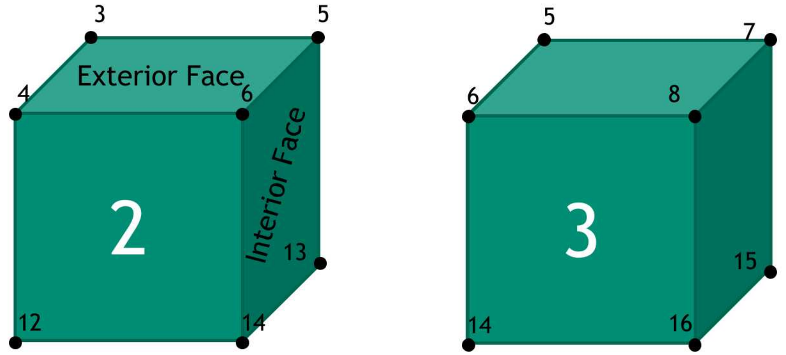
Finite element model must be "skinned" prior to import.

- Assemble vector of all faces in all elements
- Faces found in multiple elements (face 5,6,14,13 is found in elements 2 and 3) are interior faces
- Faces found in only 1 element (face 3,4,6,5 is found only in element 2) are exterior faces

Face connectivity arrays must be created, which can be imported into Blender using the Python interface.

- Vertices are an $n \times 3$ array where the rows are the vertex indices and columns are (x, y, z) coordinates.
- Faces are $m \times 3$ (triangles) or $m \times 4$ (quads) arrays where the rows are the face number and the columns are indices into the Vertices rows.
- Similar to `patch` Matlab function, except remember Python uses 0-based indexing!





Setting Up Cameras in-software

Blender uses the perfect pinhole camera model.

To specify a camera in Blender, we need to specify:

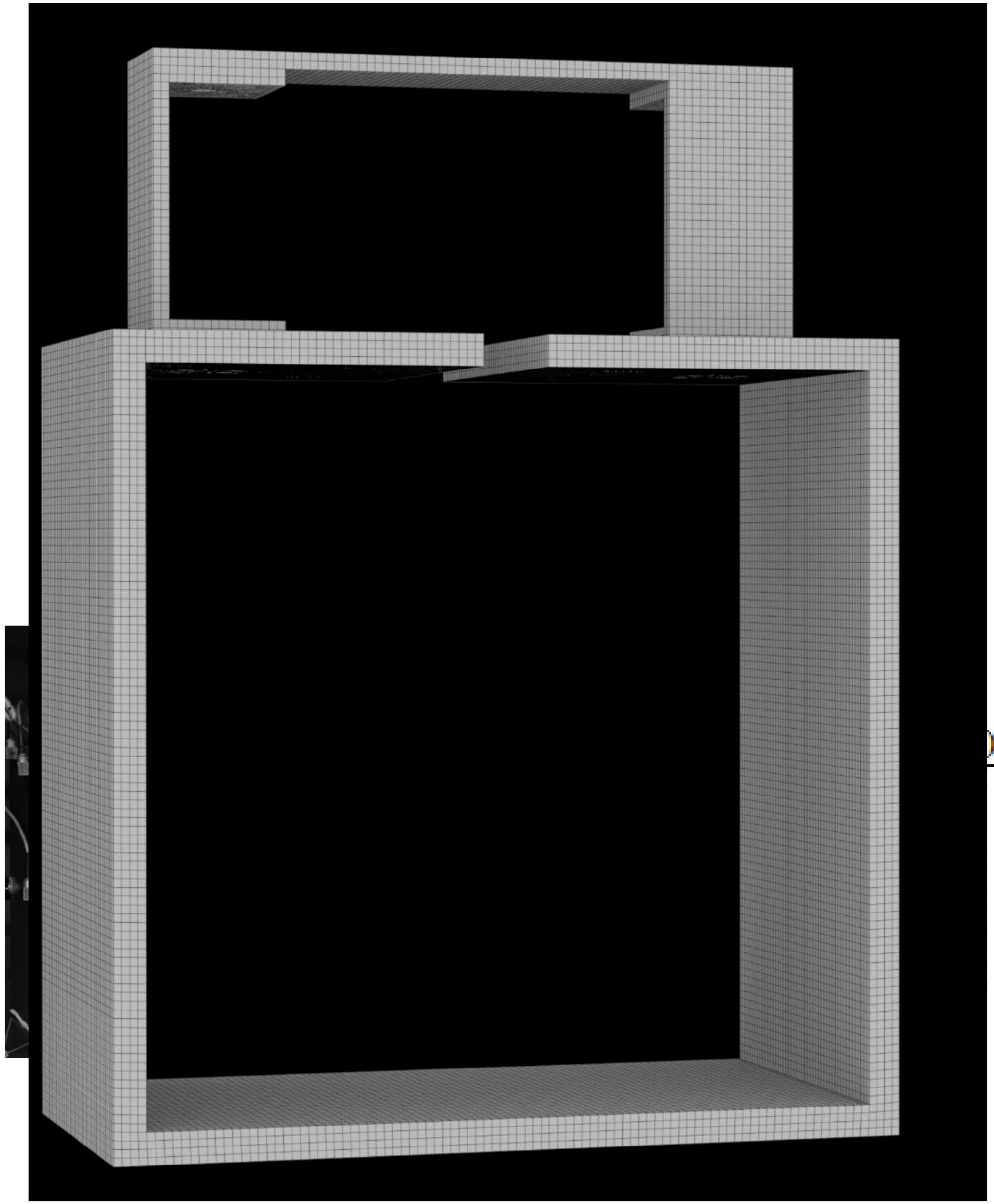
- Focal Length of Lens in millimeters
- Sensor Size in millimeters
- Image resolution in pixels
- Image shift as fraction of image size
- (x, y, z) position in the finite element coordinate system.
- Orientation in the finite element coordinate system (Euler Angles, Rotation Matrix, or Quaternions)

These can be derived from camera matrices $[K]$ and $[R|t]$

2D Calibration can give intrinsic parameters $[K]$

Camera position and orientation $[R|t]$ can be defined using Perspective-n-Point (PnP) Algorithm

OpenCV has calibration and PnP algorithms, and will take into account lens distortions from the calibration as well.



Mimicking Local Texture

To match arbitrary image degrees of freedom, the local image texture should also be applied to the mesh.

- Not strictly necessary if image degrees of freedom are DIC subsets (subsets should be identically placed, though)
- Necessary if feature tracking or gradient-based methods are used.

General computer graphics problem of texturing a 3D model using a 2D image (UV mapping)

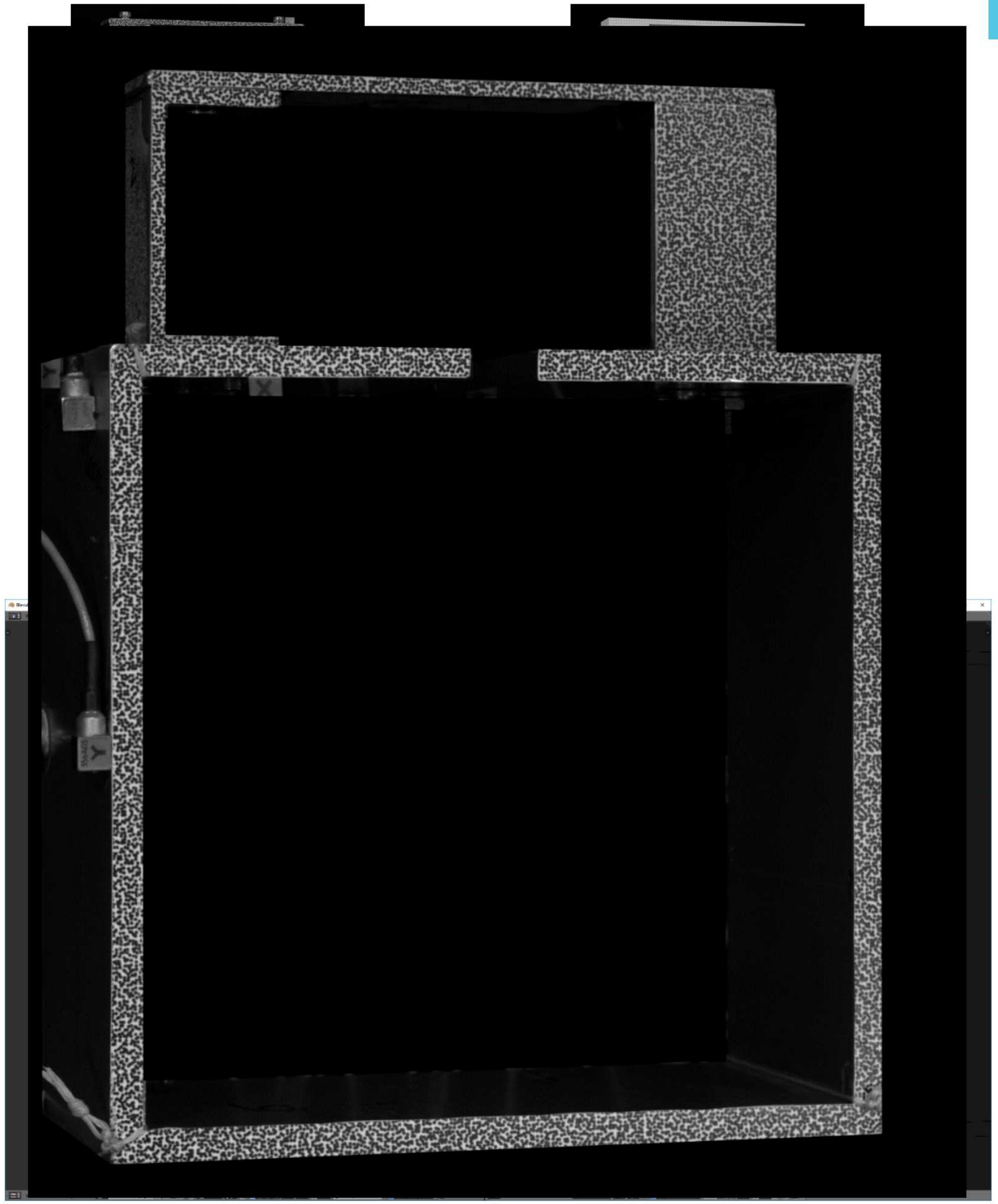
- 3D Model must be unwrapped or projected to 2D image.
- Each face is defined a portion of the image
- Image portion is mapped back to flat face on 3D model.

If texture image is identical to render image, a projection is already defined using the camera matrices.

Blender software has project from view capabilities.

Flat white lighting should be used. Shadows and specularities already baked into texture.

Projection will not look good from other views!



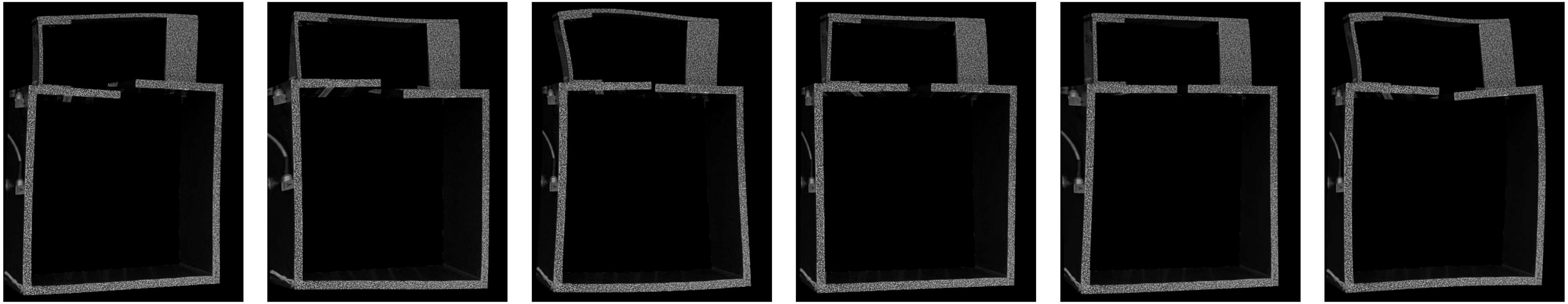
Deforming and Rendering Images

The finite element mesh can be arbitrarily deformed within the rendering software.

- Mode shapes are extracted to external files
- Files are loaded in via Blender Python interface
- Coordinates of the model vertices are updated
- **Must track node map!**
    - When the mesh is skinned, some nodes will be discarded. Need to partition displacements to only nodes kept in the skinned model.

How large to draw the deflections?

- Deflections must be small
    - Image processing techniques may suffer for large displacements (subsets become warped, gradients move past linear range)
    - Camera projection is nonlinear (normalization due to homogenous coordinates)
    - Surfaces hidden in reference view will not be textured correctly. Large displacements will reveal hidden surfaces.
- But not too small
    - Signal to noise will suffer if too small
- Deflection set to 0.5 pixel displacement at peak deflection location



First 6 elastic modes rendered at 50 pixel peak displacement (100x larger than used for analysis)

Other Discussion Points: Perspective and View Angles

Technique relies on perspective to give image-plane motion to 3D motions along the view axis of the camera.

- Perspective effects will be largest with test article closer to the camera with a wide angle lens.
- As lens focal length increases, the view angle decreases, and lines become more parallel.
- In the limit as focal length goes to infinity (orthographic view using telecentric lens), there is zero enlargement due to perspective.
- Expansion breaks down, one rigid body translation would be indistinguishable from the other two.

However:

- Wider angle lenses may have more lens distortions
- A closer object to the camera will have a narrower depth of field for a given aperture
- Lenses will have a minimum focus distance; can't get closer.

If work were repeated:

- Use wider-angle lens than 85mm
- Use a more oblique angle to the xy plane
- This can be investigated using synthetic images

Other discussion Points: Heat Waves

Heat waves are well known in DIC analysis

- Bright lights on a surface will cause heating; index of refraction changes cause the part to deform on the image.
- These deformation shapes are **not** in the finite element model shapes, so expansion suffers.

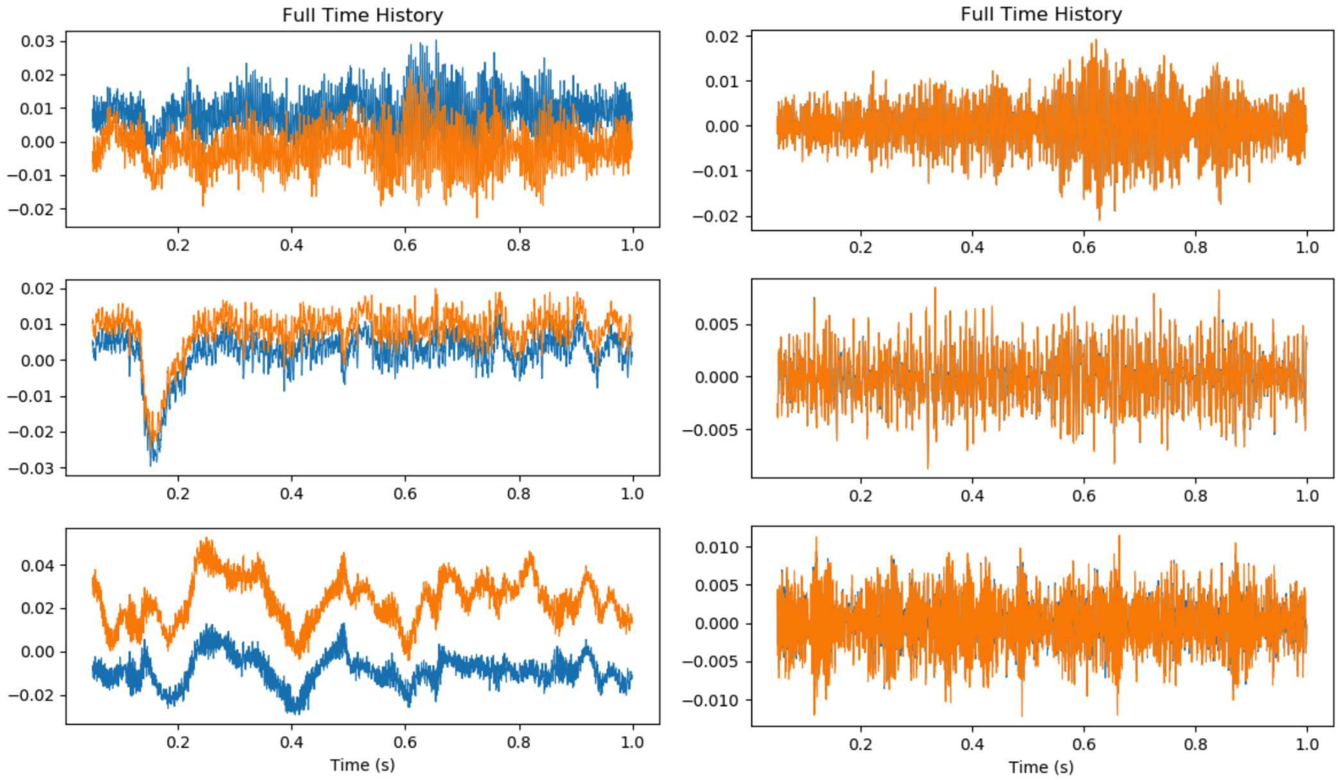
Image averaging can reduce some effect of heat waves

- Heat waves are not periodic with the excitation, and may therefore be averaged out.

Heat waves have low temporal frequency

- If modes are high enough in frequency, heat wave data can be filtered out.

Very poor results were obtained if data were not filtered below 100 Hz.



No Filtering

High-pass Filtering at 100 Hz