SUPPLEMENTARY MATERIAL

for

Topology-based three-dimensional reconstruction of delicate skeletal fossil remains and the quantification of their taphonomic deformation

Oliver E. Demuth[1,2,3,*], Juan Benito[1,4], Emanuel Tschopp[5,6,7], Stephan Lautenschlager[8], Heinrich Mallison[5,9], Niklaus Heeb[3], Daniel J. Field[1,10]

[1] Department of Earth Sciences, University of Cambridge, Cambridge, UK

[2] Structure and Motion Laboratory, Royal Veterinary College, Hatfield, UK

[3] Department Design, Zurich University of the Arts, Zurich, Switzerland

[4] Department of Biology & Biochemistry, Milner Centre for Evolution, University of Bath, Bath, UK

[5] Universität Hamburg, Hamburg, Germany

[6] Division of Paleontology, American Museum of Natural History, New York, USA

[7] GeoBioTec, NOVA School of Science and Technology, Caparica, Portugal

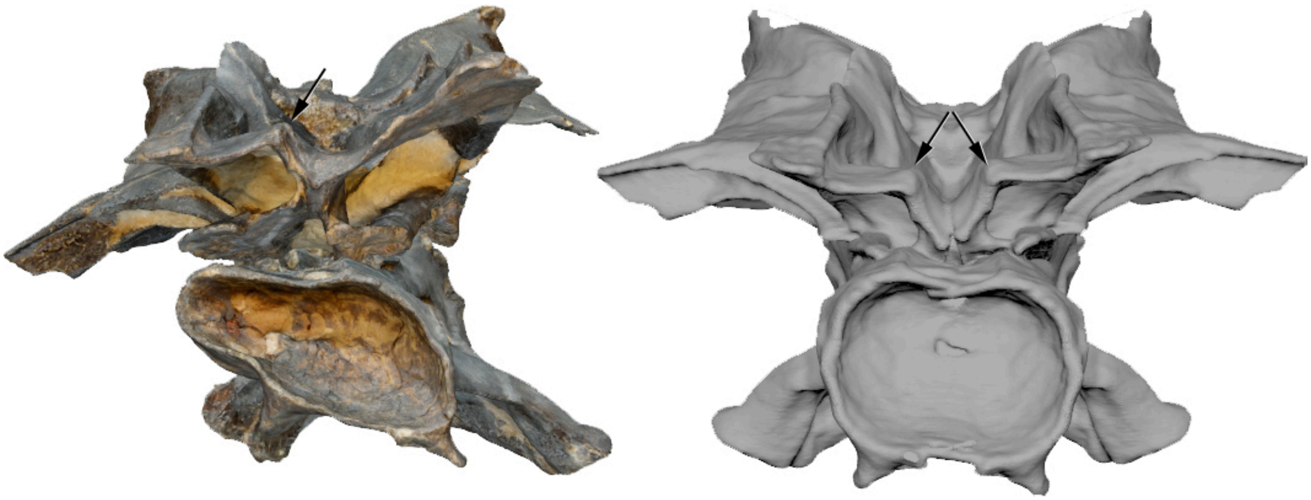
[8] School of Geography, Earth and Environmental Sciences, University of Birmingham, Birmingham, UK

[9] Palaeo3D, Rain am Lech, Germany

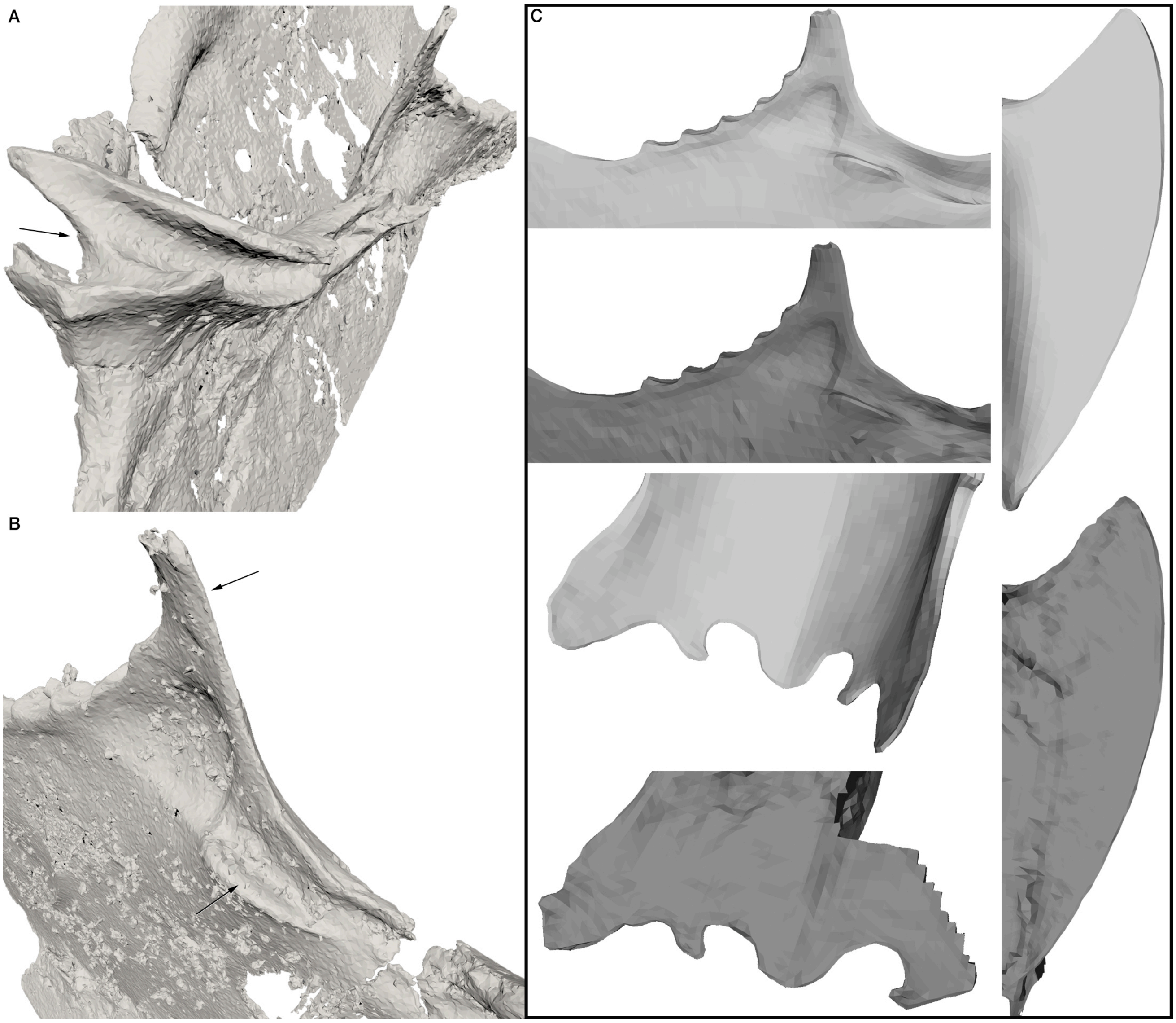
[10] Museum of Zoology, University of Cambridge, Cambridge, UK

*** Correspondence:**
Oliver E. Demuth
oed24@cam.ac.uk

1 Supplementary Figures



Supplementary Figure S1. Deformed and semi-automated reconstruction of *Galeamopus* SMA 0011/NMZ 1000011 CV 8 in caudal view. The semi-automated reconstruction approach following Lautenschlager (2016) resulted in only partial retrodeformation. Although the retrodeformed vertebra is fully symmetrical, deformation artefacts are still present (e.g., folded postzygapophyses, see arrows), indicating that further manual reconstruction would be required and that automated methods are not universally applicable.



Supplementary Figure S2. Well-preserved parts in the *Ichthyornis* specimens. (A) Crossed coracoid sulci of FHSM VP-18702. (B) Craniolateral process and ventral lip of NHMUK PV A 905 (arrows). (C) Comparison between reconstruction (light grey) and retopologised (dark grey) well-preserved parts of the individual specimens.

2 Supplementary Tables

Supplementary Table S1. All digital elements from this study and where to find them.

Specimen	Taxon	Element	Step	DOI
NMZ 1000011	*Galeamopus pabsti*	Cervical vertebra 1 (atlas)	Original photogrammetry mesh	10.17602/M2/M412553
NMZ 1000011	*Galeamopus pabsti*	Cervical vertebra 2 (axis)	Original photogrammetry mesh	10.17602/M2/M412580
NMZ 1000011	*Galeamopus pabsti*	Cervical vertebra 3	Original photogrammetry mesh	10.17602/M2/M414476
NMZ 1000011	*Galeamopus pabsti*	Cervical vertebra 4	Original photogrammetry mesh	10.17602/M2/M414482
NMZ 1000011	*Galeamopus pabsti*	Cervical vertebra 5	Original photogrammetry mesh	10.17602/M2/M414490
NMZ 1000011	*Galeamopus pabsti*	Cervical vertebra 6	Original photogrammetry mesh	10.17602/M2/M414493
NMZ 1000011	*Galeamopus pabsti*	Cervical vertebra 7 (neural arch)	Original photogrammetry mesh	10.17602/M2/M414499
NMZ 1000011	*Galeamopus pabsti*	Cervical vertebra 7 (centrum)	Original photogrammetry mesh	10.17602/M2/M414502
NMZ 1000011	*Galeamopus pabsti*	Cervical vertebra 8 (neural arch)	Original photogrammetry mesh	10.17602/M2/M414505
NMZ 1000011	*Galeamopus pabsti*	Cervical vertebra 8 (neural arch)	Retopologised mesh	10.17602/M2/M414546
NMZ 1000011	*Galeamopus pabsti*	Cervical vertebra 8 (neural arch)	Retrodeformed mesh	10.17602/M2/M414549
NMZ 1000011	*Galeamopus pabsti*	Cervical vertebra 8 (centrum)	Original photogrammetry mesh	10.17602/M2/M414511
NMZ 1000011	*Galeamopus pabsti*	Cervical vertebra 8 (centrum)	Retopologised mesh	10.17602/M2/M414552
NMZ 1000011	*Galeamopus pabsti*	Cervical vertebra 8 (centrum)	Retrodeformed mesh	10.17602/M2/M414555
NMZ 1000011	*Galeamopus pabsti*	Cervical vertebra 8 (composite)	Merged automatically retrodeformed mesh	10.17602/M2/M414542
NMZ 1000011	*Galeamopus pabsti*	Cervical vertebra 8 (prototype)	Prototype	10.17602/M2/M414574
NMZ 1000011	*Galeamopus pabsti*	Cervical vertebrae 9 & 10	Original photogrammetry mesh	10.17602/M2/M414528
NMZ 1000011	*Galeamopus pabsti*	Cervical vertebra 9	Retopologised mesh	10.17602/M2/M414558
NMZ 1000011	*Galeamopus pabsti*	Cervical vertebra 9	Retrodeformed mesh	10.17602/M2/M414561
NMZ 1000011	*Galeamopus pabsti*	Cervical vertebra 9	Prototype	10.17602/M2/M414577
NMZ 1000011	*Galeamopus pabsti*	Cervical vertebra 10	Retopologised mesh	10.17602/M2/M414564
NMZ 1000011	*Galeamopus pabsti*	Cervical vertebra 10	Retrodeformed mesh	10.17602/M2/M414567
NMZ 1000011	*Galeamopus pabsti*	Cervical vertebra 10	Protoype	10.17602/M2/M414580
NMZ 1000011	*Galeamopus pabsti*	Cervical vertebra 12 & dorsal vertebra 1	Original photogrammetry mesh	10.17602/M2/M414539

Supplementary Table S1. (continued).

Specimen	Taxon	Element	Step	DOI
FHSM VP 18702	*Ichthyornis dispar*	Sternum	Original CT mesh	10.17602/M2/M399358
FHSM VP 18702	*Ichthyornis dispar*	Sternum	Retopologised mesh	10.17602/M2/M414671
FHSM VP 18702	*Ichthyornis dispar*	Sternum	Retrodeformed mesh	10.17602/M2/M411923
KUVP 1189673	*Ichthyornis dispar*	Sternum	Original CT mesh	10.17602/M2/M406297
KUVP 1189673	*Ichthyornis dispar*	Sternum	Retopologised mesh	10.17602/M2/M414682
KUVP 1189673	*Ichthyornis dispar*	Sternum	Retrodeformed mesh	10.17602/M2/M411926
NHMUK A 905	*Ichthyornis dispar*	Sternum	Original CT mesh	10.17602/M2/M406247
NHMUK A 905	*Ichthyornis dispar*	Sternum	Retopologised mesh	10.17602/M2/M414674
NHMUK A 905	*Ichthyornis dispar*	Sternum	Retrodeformed mesh	10.17602/M2/M411932
-	*Ichthyornis dispar*	Sternum	Prototype	10.17602/M2/M414595

Supplementary Table S2. µCT scan settings and scan resolutions for the *Ichthyornis* sterna.

Specimen	Voltage (kV)	Current (µA)	Voxel size (mm)	Number of slices
FHSM VP-18702	170	225	0.058	1727
KUVP 119673	200	140	0.0643	1758
NHMUK PV A 905	155	260	0.0363	1764

Supplementary Table S3. *Galeamopus* retopology properties.

Specimen	Vertices	Quadrilateral faces	Triangles	Median global deformation [cm]	Median local deformation [degrees]
SMA 0011/NMZ 1000011 CV8 centrum	3365	3261	6522	1.875	22.866
Reconstruction CV8 centrum	3484	3482	6964	-	-
SMA 0011/NMZ 1000011 CV8 neural arch	7213	6830	13660	1.606	20.815
Reconstruction CV8 neural arch	8840	8842	17684	-	-
SMA 0011/NMZ 1000011 CV9	7391	7189	14378	1.280	12.122
Reconstruction CV9	12172	12176	23352	-	-
SMA 0011/NMZ 1000011 CV10	7679	7427	14854	1.622	13.980
Reconstruction CV10	12660	12664	22884	-	-

Supplementary Table S4. *Ichthyornis* retopology properties.

Specimen	Vertices	Quadrilateral faces	Triangles	Median global deformation [cm]	Median local deformation [degrees]
FHSM VP-18702	10836	10646	21292	0.592	25.807
KUVP 119673	7949	7832	15664	0.441	22.080
NHMUK PV A 905	6748	13220	6610	0.250	15.491
Reconstruction	11444	11442	22884	-	-

3 Supplementary Code

Supplementary Code S1. MAYA Embedded Language (MEL) script for visualisation of CSV data as vertex colours.

```
/* Vertex attribute color

    This script applies color based on a data file (.csv) to an object's vertices, i.e. it
    generates and applies a heat map. The colormap is loaded from another CSV file. Both
    the data and colormap csv files need to be in the same folder specified in this script.
    Select the mesh and run script.

    Written by Oliver Demuth 22.01.2020
    Last updated 01.12.2021 - Oliver Demuth

    Note the number of vertices must match the number of rows of the input data file. Make
    sure to have vertex colors on: Mesh Display > Toggle Display Colors Attribute

*/

//==========================================================//

/************** SET VARIABLE NAMES BELOW *******************/


string $array[] = `ls -sl`; //select two objects to compare

string $impPath = ("YOUR/FILE/PATH"); // set file path where the colormaps and data are saved

string $nc = "COLORMAP";     // name of colormap; change as appropriate, e.g. "parula", "jet",
"hot", etc. based on your predefined colormaps in your folder (without '.csv' suffix)

string $DATA = "DATAFILE"; // name of data file (without '.csv' suffix)

int $setRefSpace = 0;     // set reference space (0 = world space, 1 = object space)


//==========================================================//

/**************  NO CHANGES NEEDED BELOW *******************/

// check if one object is selected, otherwise issue error //

int $selArraysize = size($array);

string $impFile2 = ($impPath + $DATA + ".csv");
string $impData2[] = `fileread($impFile2)`;     // load data CSV file
int $DataSize = size($impData2);     //count # rows of input CSV

string $ErrorString1 = "No objects selected. Cannot perform vertex distance color transformation.
Please select a mesh.";
string $ErrorString2 = "Too many objects selected. Cannot perform vertex distance color
transformation. Please select only one mesh.";

if ($selArraysize == 0) {
    select -cl;
    error $ErrorString1;
}

if ($selArraysize > 1) {
    select -cl;
    error $ErrorString2;
}


// check if object's vertex count matches # rows of data, otherwise issue error //

int $numbVerts[] = `polyEvaluate -v $array[0]`; // count number of vertices
```

```
int $numbV[];

$numbV[0] = $numbVerts[0];
$numbVDiff = $numbV[0] - $DataSize;

string $ErrorString3 = "The number of vertices of target " + $array[0] + "Shape does not match
the size of " + $DATA + ".csv. Cannot perform vertex distance color transformation. Please check
input data.";
if ($numbVDiff != 0) {
    select -cl;
    error $ErrorString3;
}


// if checks passed execute script //


if ($setRefSpace < 0) $setRefSpace = 0;    //check reference space inputs
if ($setRefSpace > 0) $setRefSpace = 1;

string $refSpace[];    // set reference space
$refSpace[0] = "-ws";
$refSpace[1] = "-os";

string $impFile = ($impPath + $nc + ".csv");
string $impData[] = `fileread($impFile)`;    // load colormap CSV file
int $colorSize = size($impData);    //count # rows of input CSV

string $t[];
vector $colorVec[];
float $colorMap[];

int $k = 0;
for ($k = 0;$k<$colorSize;$k++)    // goes through CSV and reads data points into array
{
    tokenizeList($impData[$k], $t);

    $colorMap[0] = $t[0];
    $colorMap[1] = $t[1];
    $colorMap[2] = $t[2];

    $colorVec[$k] = << $colorMap[0], $colorMap[1], $colorMap[2]>>; // represents CSV as vector
array
}

string $u[];
float $NormArray[];

float $global_max = 0.0001;

int $m = 0;
for ($m = 0;$m<$DataSize;$m++)    // goes through CSV and reads data points into array
{
    tokenizeList($impData2[$m], $u);

    $NormArray[$m] = $u[0];

    if ($NormArray[$m] >= $global_max)    // set $global_max to the maximum of the input data
    $global_max = $NormArray[$m];
}

float $colorNorm = $colorSize / $global_max; // normalize colorscale based on $global_max

// set progressBar

progressBar -edit
    -beginProgress
    -isInterruptable true
    -status "Set Vertex Colours"
    -maxValue $numbVerts[0]
    $gMainProgressBar;

int $j = 0;
for($j; $j < $numbVerts[0]; $j++)    // go through each vertex individually
```

```
{
    float $rowfloat = $NormArray[$j] * $colorNorm * (1/(float)$colorSize * ($colorSize - 1));
// put vertex into colormap bin
    $rowfloat = python("'%.0f' % "+$rowfloat);

    int $row = $rowfloat;
    if ($row > $colorSize -1)    // if maximum is manually set instead of $global_max, restrict
possible bins to actual bins
        $row = $colorSize -1;

    vector $colorRow = $colorVec[$row];    // temporary vector to extract vector components

    float $R = $colorRow.x;
    float $G = $colorRow.y;
    float $B = $colorRow.z;

    select -r ($array[0] + ".vtx["+$j+"]");    // select individual vertex
    polyColorPerVertex -rgb $R $G $B;    // paint vertex color
    select -cl;

    progressBar -edit
    -status "Set Vertex Colours"
    -step $j $gMainProgressBar;
}

progressBar -edit
    -endProgress
```

4 Supplementary References

Lautenschlager, S. (2016). Reconstructing the past: methods and techniques for the digital restoration of fossils. *Royal Society Open Science*, *3*(10), 160342. https://doi.org/10.1098/rsos.160342