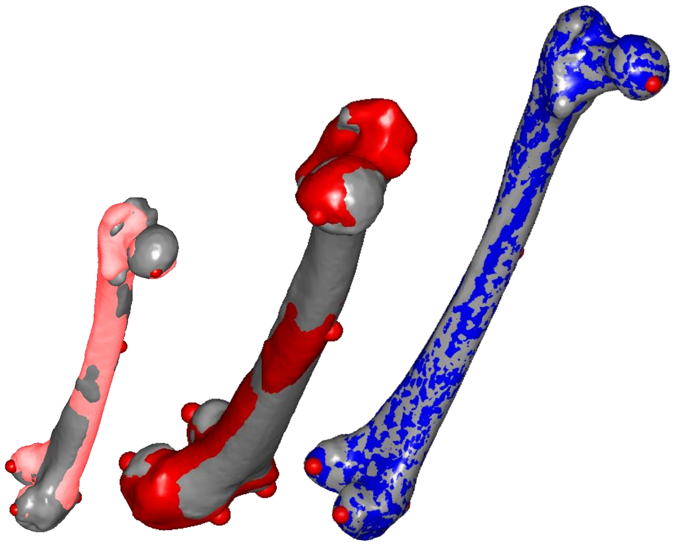


Personalize your musculoskeletal models based on medical image data

PAVEL GALIBAROV, PHD

ANYBODY TECHNOLOGY

Outline

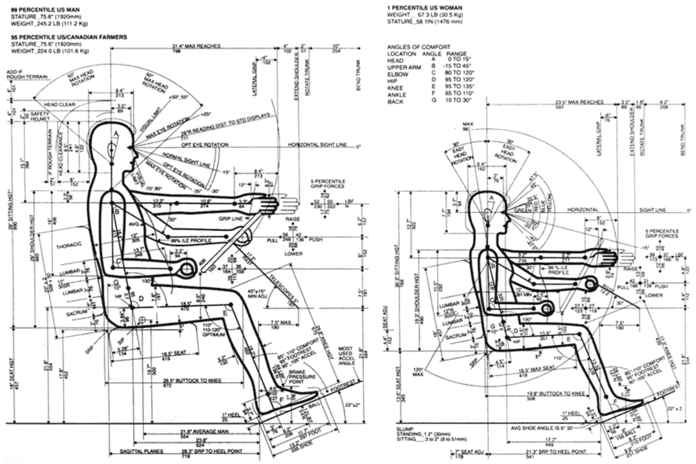
- Motivation for subject-specific modeling
- Individualization principles and techniques
  - Anthropometric scaling
  - Individualization of body parts
  - Available tools
- Code examples:
  - Full bone morphing
  - Bone morphing based on partially available surface information
  - MoCap model with a subject-specific bone
- Conclusion remarks

Pavel Galibarov
(Presenter)

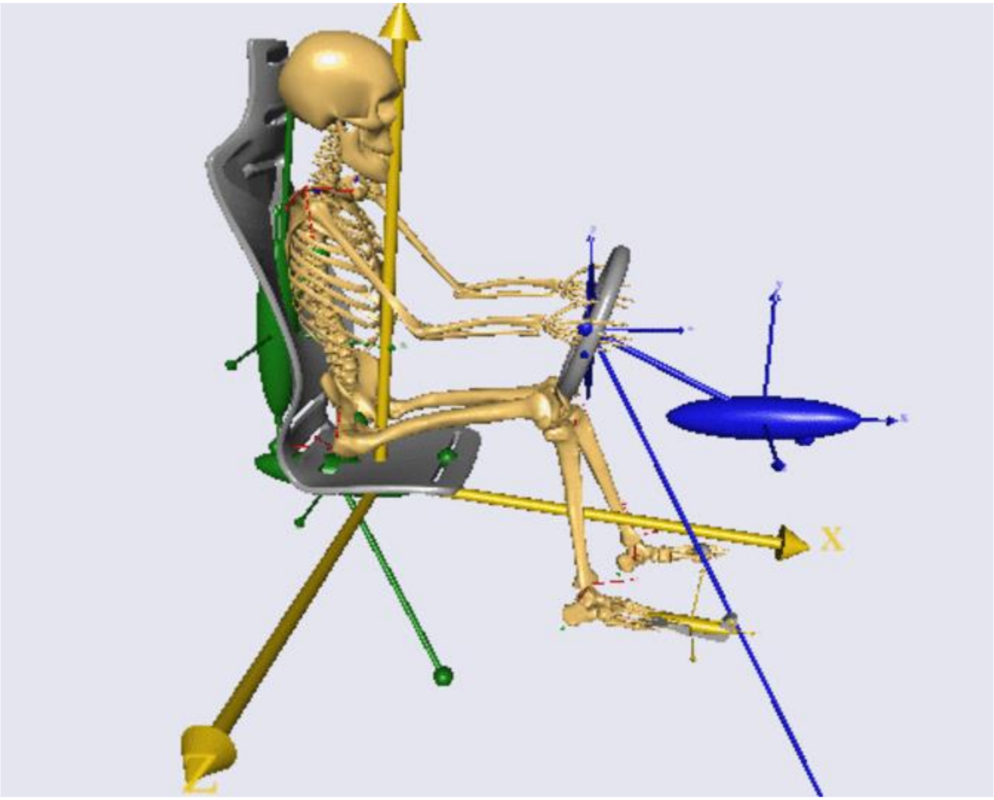
Motivation: mass *vs.* personal products

- <u>Mass products</u>
  - Target as many customers as possible
    - Vehicle interior/exterior
    - Workspace
    - Mobile device
  - Design based on anthropometric databases
    - ANSUR
    - CAESAR
    - NHANES
    - etc.
  - Make product that are ‘personal’, but mass produced
- <u>Personal products</u>
  - Best function for an individual as a selling point
  - Example: Shoe-insoles, medical devices, etc.

- <u>Computational models reduce costs</u>
  - Allow virtual testing on a wide range of subjects
  - Virtually cost free ergonomic and other analyses
  - Ethically sound product development
  - Shorter time-to-market



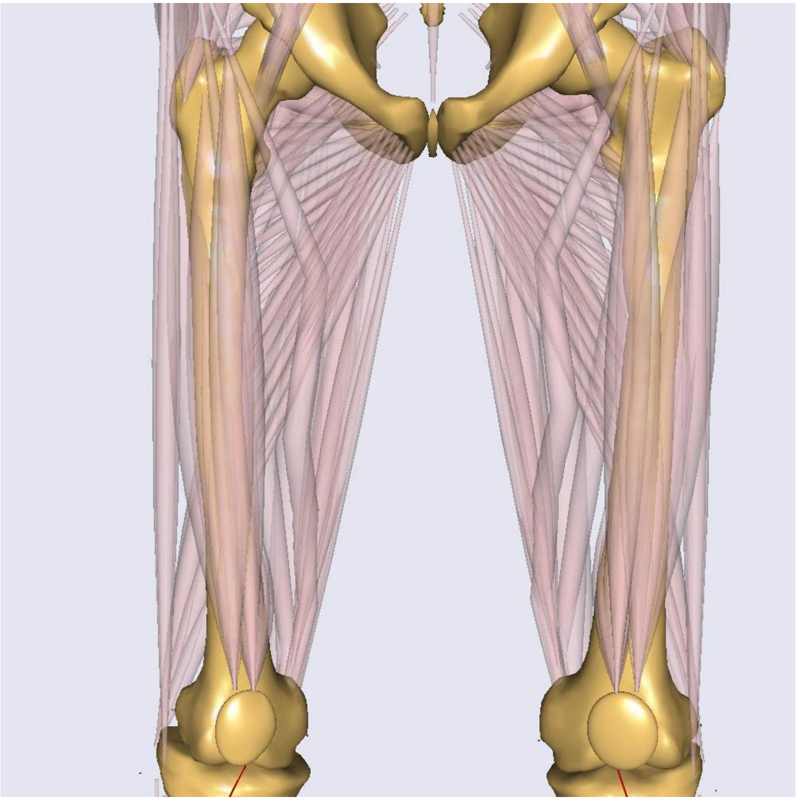
Motivation: Ergonomic design



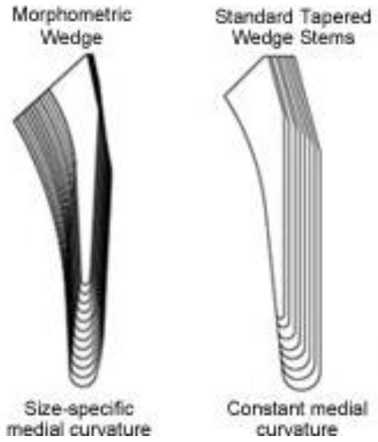
- Seat and steering wheel adjustment mechanisms
- Previous webcast:
  - *Modeling Of Population Ergonomics With AnyBody*
  - www.youtube.com/anybodytech
  - 16.000 subject-specific musculoskeletal models
  - Cross-correlated bone dimensions

Motivation: Anatomic products

- Linear scaling is simple – problem solved?
- Not completely…
  - same bone length
  - different shape
  - different inner/outer cortex proportions
- Muscle attachment configuration changes
  - Leads to different muscle activation patterns
  - Different joint loading



Motivation: Anatomic products



Accolade II Wedge Stems

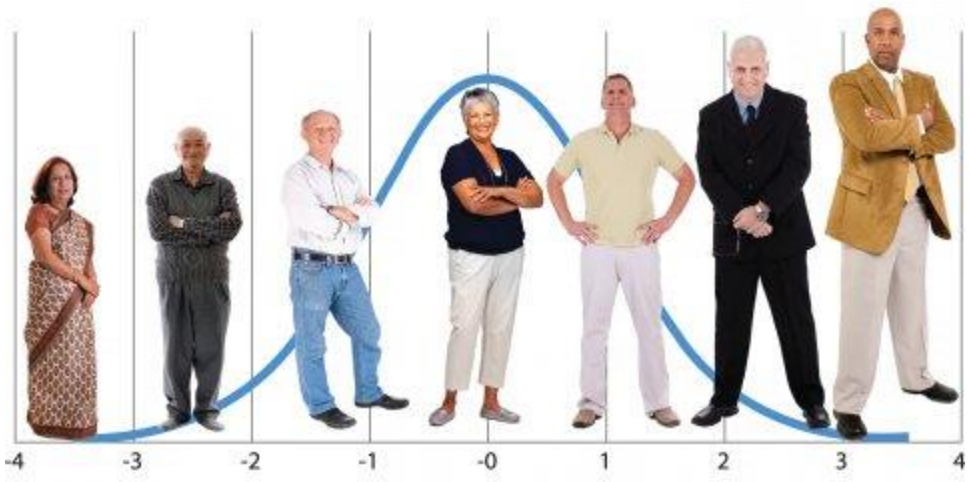
http://www.stryker.com/

“SOMA (Stryker Orthopaedics Modeling and Analytics system), is a proprietary population-based design environment”

AxSOS 3 Plating System

Motivation: Anatomic products

Materialise ADaM™ for Population Analysis

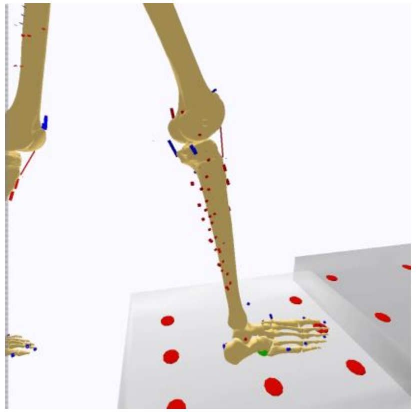


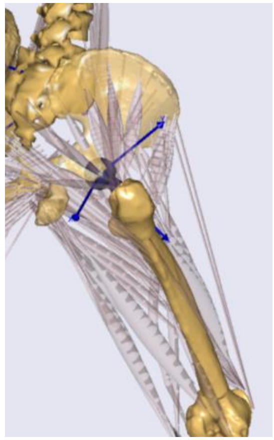
biomedial.materialise.com

www.djoglobal.com/products/djo-surgical/taperfill-hip-system

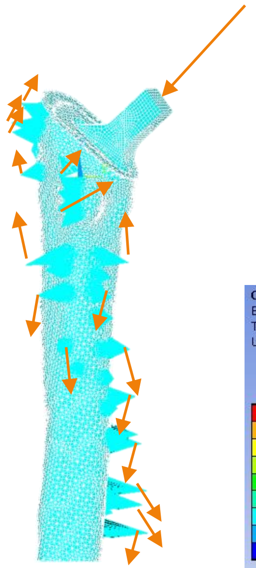
”Virtual population analysis improves orthopedic implant design”
S. Kolk, M. Lawrenchuk, Orthopedic Design and Technology Magazine

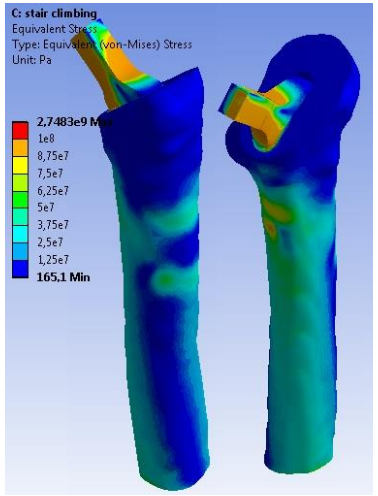
Motivation: Multiscale modeling





- Consistent geometries in FEA and musculoskeletal models
- Ensures quality of the muscular and joint reaction force transfer
- Required for high level of precision to address such questions as, for example, bone remodeling

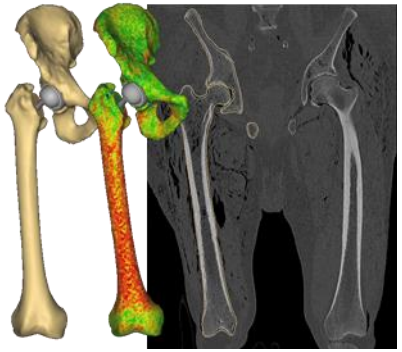




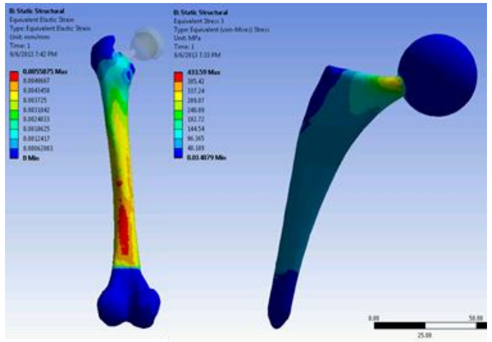
Courtesy of Prof. Sebastian Dendorfer, OTH Regensburg, Germany

Presented previously workflow

Materialise
innovators you can count on

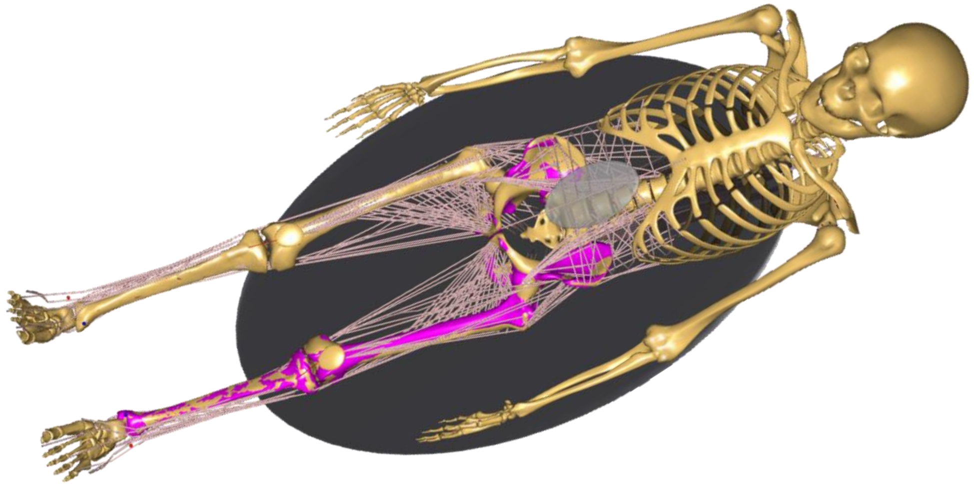


ANYBODY



ANSYS

Motivation: Anatomic accuracy



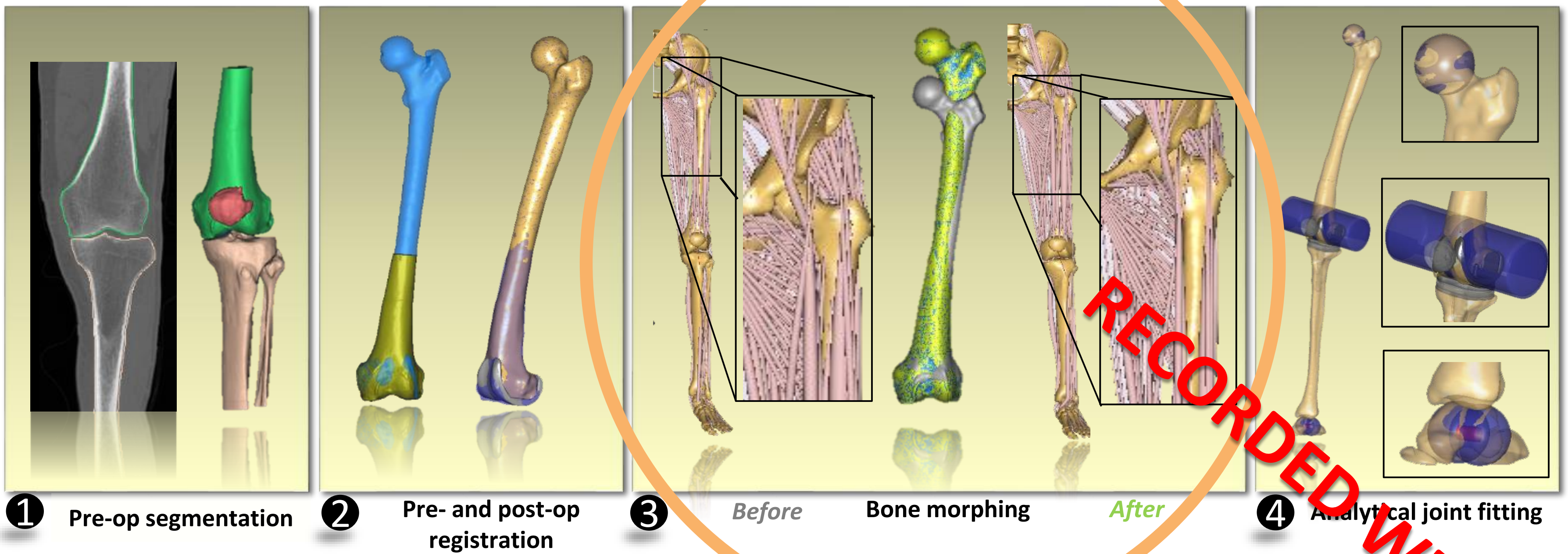
- (Purple) Lower extremity bones from MRI
- Personalized and registered AnyBody model
- TLEMsafe project

youtube.com/anybodytech

Motivation: Result accuracy

Patient-specific Musculoskeletal Modelling of Total Knee Arthroplasty using Force-dependent Kinematics

Twente Lower Extremity Model (TLEM) v. 2.0 (Fluit et al, 2013)



Winner of the 5th Grand Challenge Competition to Predict In Vivo Knee Loads!!!

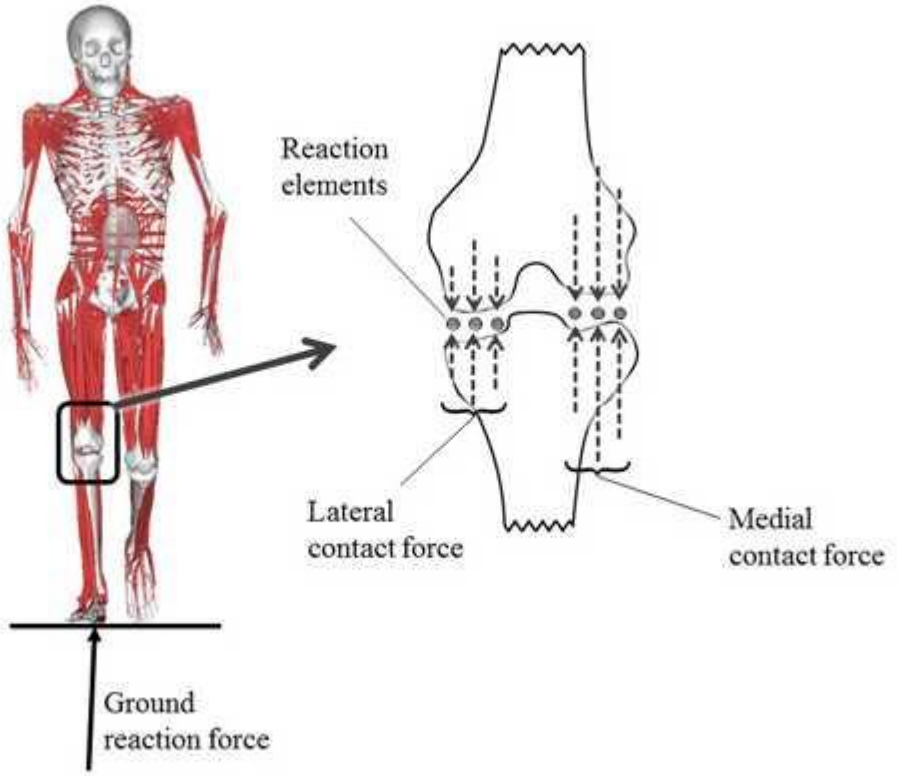
Motivation:
Contact modeling for accurate results

Next AnyBody webcast:

Knee contact force estimation using force-reaction elements

“The proposed knee contact model with subject-specific joint model could predict in vivo knee contact forces with reasonable accuracy.”

Winner of the 6th Grand Challenge Competition to Predict In Vivo Knee Loads!!!



NB. Individualization scheme might differ from the presented in this webcast

Personalization scheme

1. Apply overall size of the modeled human
   - Anthropometric scaling to reach needed height/weight
   - Regression equations to distribute individual bone lengths
   - Data sources: MoCap, collected height/segment length measurements

Personalization scheme

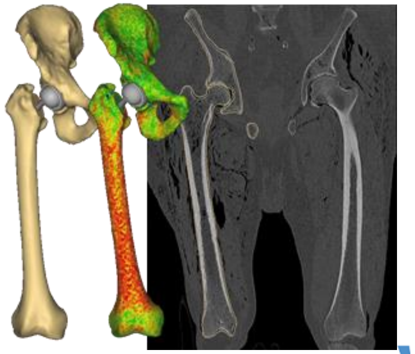
1. Apply overall size of the modeled human
   - Anthropometric scaling to reach needed height/weight
   - Regression equations to distribute individual bone lengths
   - Data sources: MoCap, collected height/segment length measurements
2. Correct individual bone shapes / muscle attachment/insertion locations
   - Use available medical imaging data
     - X-ray measurements
     - Reconstructed surfaces from CT/MRI
     - Partial reconstructions
   - Use consistent bone surfaces for contact analyses

Personalization scheme

1. Apply overall size of the modeled human
   - Anthropometric scaling to reach needed height/weight
   - Regression equations to distribute individual bone lengths
   - Data sources: MoCap, collected height/segment length measurements
2. Correct individual bone shapes / muscle attachment/insertion locations
   - Use available medical imaging data
     - X-ray measurements
     - Reconstructed surfaces from CT/MRI
     - Partial reconstructions
   - Use consistent bone surfaces for contact analyses
3. **Use available muscle volume data to correct default/scaled muscle strength, muscle attachment, etc.**

Personalization scheme

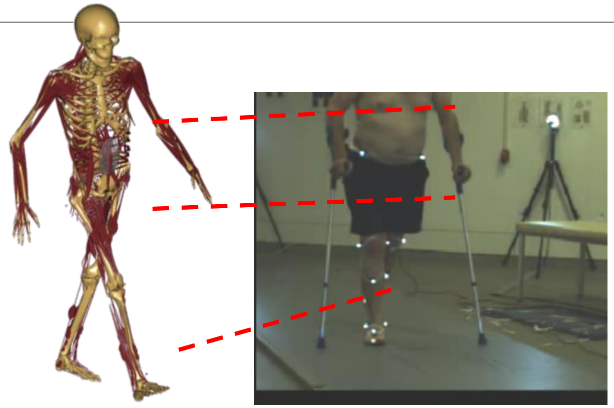
1. Anthropometrics:
Weight, height, fat percentage, individual segment lengths, etc.



2. Bone morphing:
Segmented surface files, landmark protocols, etc.

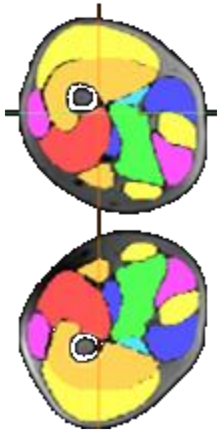
Kinematics & Kinetics

Geometry



- Motion capture to define subj-spec. motion
- Force-plates to define kinetics

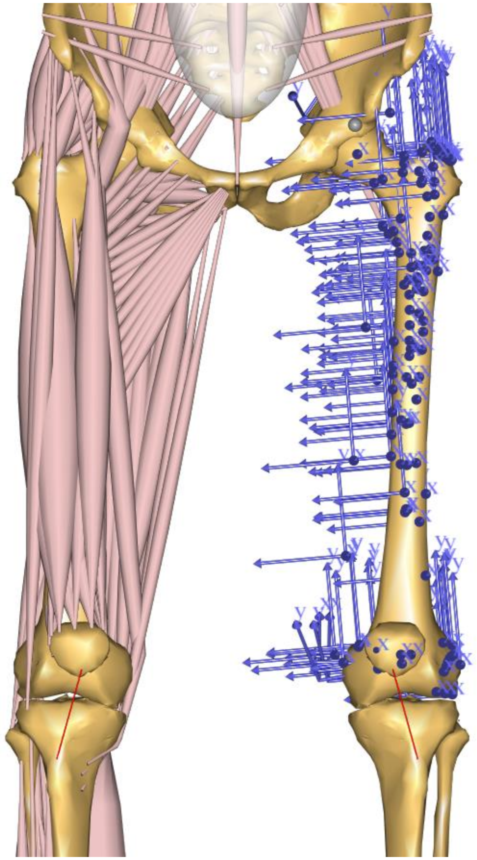
3. Further improvements:
Muscle volumes, muscle/ligament attachment nodes correction, implant position, etc.



Personalization: complex? a lot of work?

- Model looks very detailed
- A lot of structures need to be modified and adjusted
  - Soft tissue insertion/attachment locations
  - Via points
  - Muscle wrapping objects
- Does it take a lot of time and effort to model a new person?

- No!
  - Model is organized to allow quick individualization
  - Basic anthropometric scaling laws require a couple of numbers
  - Segment individualization is a little more complex, but…
  - Most common solutions can be reused and
    will be provided as the last part of this webcast!



Model architecture

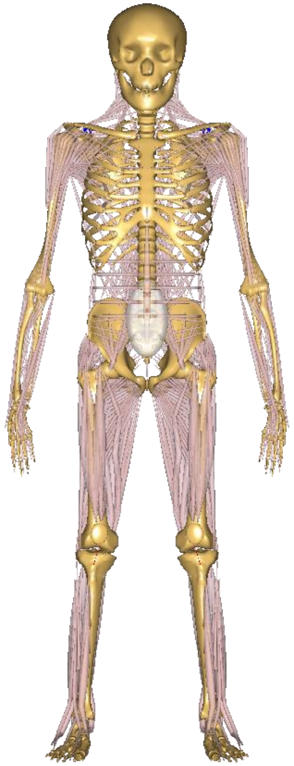
- Model is organized in a tree like structure:
  - Segment
    - Branches: body parts and working folders/objects
    - Leave type: muscle insertion position vectors subjected to a 3D scaling function
    - Leave type: scaling 3D function as a leave of the second last branch
  - Anthropometric scaling law is a tree:
    - Branches organized corresponding to the body parts
    - Leaves: linear transformations
- Everything is connected and picked up automatically!

```
// Anthropometric scaling law
AnyFolder <AnthroLaw> = …
        AnyFolder <BodyPart> = …
                AnyFunction < MyScalingFunction > = …
        …

// Segment
AnySeg <SegmentName> = …
        AnyFunction <MyScalingFunction> = …
        AnyRefNode <NodeName> = …
                sRel = .MyScalingFunction( std. location);
```

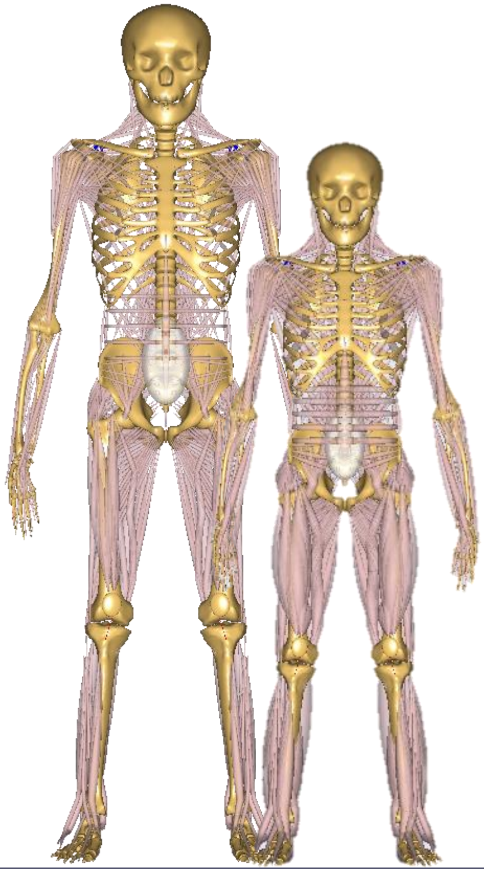
Anthropometric scaling

Generic AnyBody model



***Anthropometrics*:**
mass/height/fat percentage

Standard
Uniform
Length/Mass
Length/Mass/Fat
External Measurements



Anthropometric scaling law: ”standard”

```
AnyFolder Scaling = {
  AnyFolder MassScaling = {
    AnyFolder Pelvis = { AnyVar MassScale = 1.0; };
    ...
  AnyFolder StrengthScaling = {
    AnyVar p = (2/3); //Power
    AnyFolder Pelvis = { AnyVar StrengthScale = 1.0^.p; };
    ...
  };
  AnyFolder FiberLengthScaling = {
    AnyFolder Pelvis = { AnyVar FiberLengthScale = 1.0; };
    ...
  };
  AnyFolder GeometricalScaling = {
    AnyFolder Trunk = {
      AnyVar GeomScale= 1.0;
      AnyFunTransform3DLin ScaleFunction = {
        ScaleMat ={{.GeomScale,0,0},{0,.GeomScale,0},{0,0,.GeomScale}};
        Offset={0,0,0};
      };
    };
    AnyFolder &ThoraxSeg = Trunk;
  };
};
```

No scaling!!!

// switching ON – nothing or the following line

```
#define BM_SCALING CONST_SCALING_STANDARD
```

ANYBODY TECHNOLOGY

Anthropometric scaling law: ”uniform”

```
AnyFolder Scaling = {
  AnyFolder MassScaling = {
    AnyFolder Pelvis = { AnyVar MassScale =
..AnthroData.pelvis / ..StandardParameters.Pelvis.Mass; };
  };

  AnyFolder StrengthScaling = {
    AnyVar p = (2/3); //Power
    AnyFolder Pelvis = { AnyVar StrengthScale = (..AnthroData.pelvis/
..StandardParameters.Pelvis.Mass)^.p; };
  };

AnyFolder GeometricalScaling = {
    AnyFolder Pelvis = {
      AnyVar GeomScale = (..AnthroSegmentLengths.PelvisWidth / ..StandardParameters.Pelvis.Width);
      AnyFunTransform3DLin ScaleFunction = {
        ScaleMat ={{.GeomScale,0,0},{0,.GeomScale,0},{0,0,.GeomScale}};
        Offset = {0,0,0};
      };
    };
  };

};
```

```
AnyFolder AnthroData = {
  AnyVar Body_Mass = 75 ;
  AnyVar body_height = 180 /100;
  ...
};

AnyFolder AnthroSegmentLengths = {
  AnyVar PelvisWidth =
0.176*.AnthroData.body_height/1.8;
  ...
};
```

AnyManUniform.any

// switching ON – the following two lines

```
#define BM_SCALING CONST_SCALING_UNIFORM
#path BM_SCALING_ANTHRO_FILE "Model\AnyFamily\AnyManUniform.any"
```

Anthropometric scaling law: ”uniform”

```
AnyFolder Scaling = {
  AnyFolder MassScaling = {
    AnyFolder Pelvis = { AnyVar MassScale =
..AnthroData.pelvis / ..StandardParameters.Pelvis.Mass; };
  };

  AnyFolder StrengthScaling = {
    AnyVar p = (2/3); //Power
    AnyFolder Pelvis = { AnyVar StrengthScale = (..AnthroData.pelvis/
..StandardParameters.Pelvis.Mass)^.p; };
  };

AnyFolder GeometricalScaling = {
    AnyFolder Pelvis = {
      AnyVar GeomScale = (..AnthroSegmentLengths.PelvisWidth / ..StandardParameters.Pelvis.Width);
      AnyFunTransform3DLin ScaleFunction = {
        ScaleMat ={{.GeomScale,0,0},{0,.GeomScale,0},{0,0,.GeomScale}};
        Offset = {0,0,0};
      };
    };
  };

};
```

```
AnyFolder AnthroData = {
  AnyVar Body_Mass = 75 ;
  AnyVar body_height = 180 /100;
  ...
};

AnyFolder AnthroSegmentLengths = {
  AnyVar PelvisWidth =
0.176*.AnthroData.body_height/1.8;
  ...
};
```

AnyManUniform.any

More in the tutorials!!!

```
// switching ON – the following two lines
#define BM_SCALING CONST_SCALING_UNIFORM
#path BM_SCALING_ANTHRO_FILE "Model\AnyFamily\AnyManUniform.any"
```

Individualizing segments: why?

- *To gain more accuracy*
- *Take into account individual bone morphology*
- *Make MSM bone surface consistent with an FEA model for load transfer*
- *Employ bone/cartilage surfaces reconstructed from CT/MRI for contact analysis*

Individualizing segments: how?

```
// Anthropometric scaling law
AnyFolder <Body> = …

        AnyFolder <BodyPart> = …

                AnyFunction < MyScalingFunction > = …

        …

// Segment
AnySeg <SegmentName> = …

        AnyFunction <MyScalingFunction> = …

        AnyRefNode <NodeName> = …
                sRel = .MyScalingFunction( std. location);
```

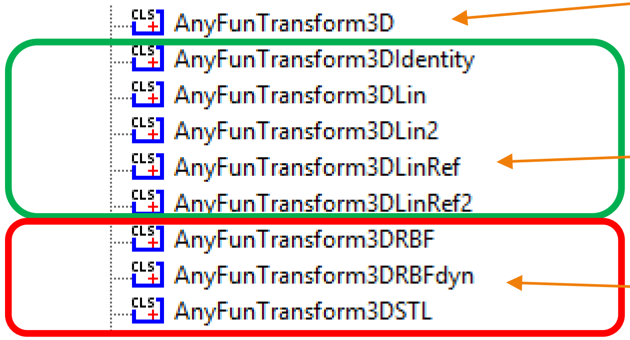
- *This function should not be looking up the anthropometric scaling law!*
- *Needs to be replaced by another 3D function…*
- *Custom segmental scaling function!!!*

Individualizing segments: how?

- *Task:*
  - *Define a 3D function that will make an AnyBody generic bone looking like the target bone*
  - *Ensure that all entities, on the surface and outside, deform to match subject's anatomy*

- *Solution:*
  - *A 3D inter/extrapolation function*
  - *Linear/Nonlinear behavior*
  - *Control over inter/extrapolative behavior*

Individualizing segments: functions



Abstract

Linear functions

Nonlinear functions (Radial Basis Functions)

+ combinations of these functions assembled into a pipeline

Linear functions

$$f(\vec{x}) = A\vec{x} + \vec{b}$$

```
AnyFunTransform3DLin <ObjectName> = {
  //PreTransforms = ;
  ScaleMat = {{1, 0, 0}, {0, 1, 0}, {0, 0, 1}};
  Offset = {0, 0, 0};
};
```

A (ScaleMat), b (Offset)

\\ AnyFunTransform3DIdentity

```
AnyFunTransform3DLin2 <ObjectName> = {
  //PreTransforms = ;
  Points0 = ; // Source landmarks, n x 3
  Points1 = ; // Target landmarks, n x 3
  //Mode = VTK_LANDMARK_RIGIDBODY;
};
```

A, b computed through least squares fitting

Uses VTK class: vtkLandmarkTransform

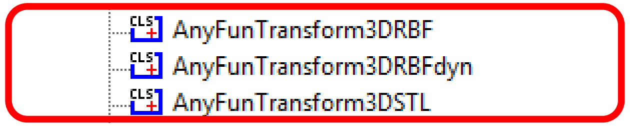
Inherits: rigid body, similarity, and affine modes

RBF functions

Interpolation transformation:

$$s(y) = \sum_{j=1}^{n} c_j \phi(\|y - x_j\|) + p(y)$$

$$\phi(r) = e^{-r^2}$$
$$\phi(r) = r^2 log(r)$$
$$\dots$$



$$s(x_k) = f_k$$

source and target landmarks

$$\Phi_{XX} c + P_X d = f_X$$
$$P_X = [e, X, X^2, \dots, X^q]$$

$$f_Y = \Phi_{YX} c + P_Y d$$

RBF functions

```
AnyFunTransform3DRBF <ObjectName> = {
    //PreTransforms = ;
    /*RBFDef = {
    Type = RBF_Gaussian;
    Param = 1;
    };*/
    Points0 = ;
    //RBFCoefs = ;
    //PolynomDegree = -1;
    //PolynomCoefs = ;
    //PointNames = ;
    //PointDescriptions = ;
    //Points1 = ;
    /*BoundingBox = {
    Type = BB_Cartesian;
    ScaleXYZ = {2, 2, 2};
    DivisionFactorXYZ = {1, 1, 1};
    };*/
    //BoundingBoxOnOff = Off;
  };
```

Constructs interpolation transformation based on two point sets as described on the previous slide

source landmarks

polynomial degree

target landmarks

bounding box parameters for extrapolative behaviour

RBF functions

```
AnyFunTransform3DSTL <ObjectName> = {
   //PreTransforms = ;
   /*RBFDef = {
   Type = RBF_Gaussian;
   Param = 1;
   };*/
   //FileName0 = ;
   //ScaleXYZ0 = {1, 1, 1};
   //SurfaceObjects0 = ;
   //FileName1 = ;
   //ScaleXYZ1 = {1, 1, 1};
   //SurfaceObjects1 = ;
   NumPoints = 0;
   //UseClosestPointMatchingOnOff = On;
   //PolynomDegree = -1;
   //PolynomCoefs = ;
   /*BoundingBox = { … };*/
   //BoundingBoxOnOff = Off;
 };
```

Similar to the previous, but points are selected automatically on the surfaces, requires preregistration of the surfaces

source surface

target surface

polynomial degree

bounding box parameters for extrapolative behaviour

Individualizing segments: pipeline

Levels of scaling

- anthropometric
- custom

overall scaling transformation

Levels of accuracy

- linear/uniform scaling
- rough nonlinear (using few points)
- surface based nonlinear (using many points)

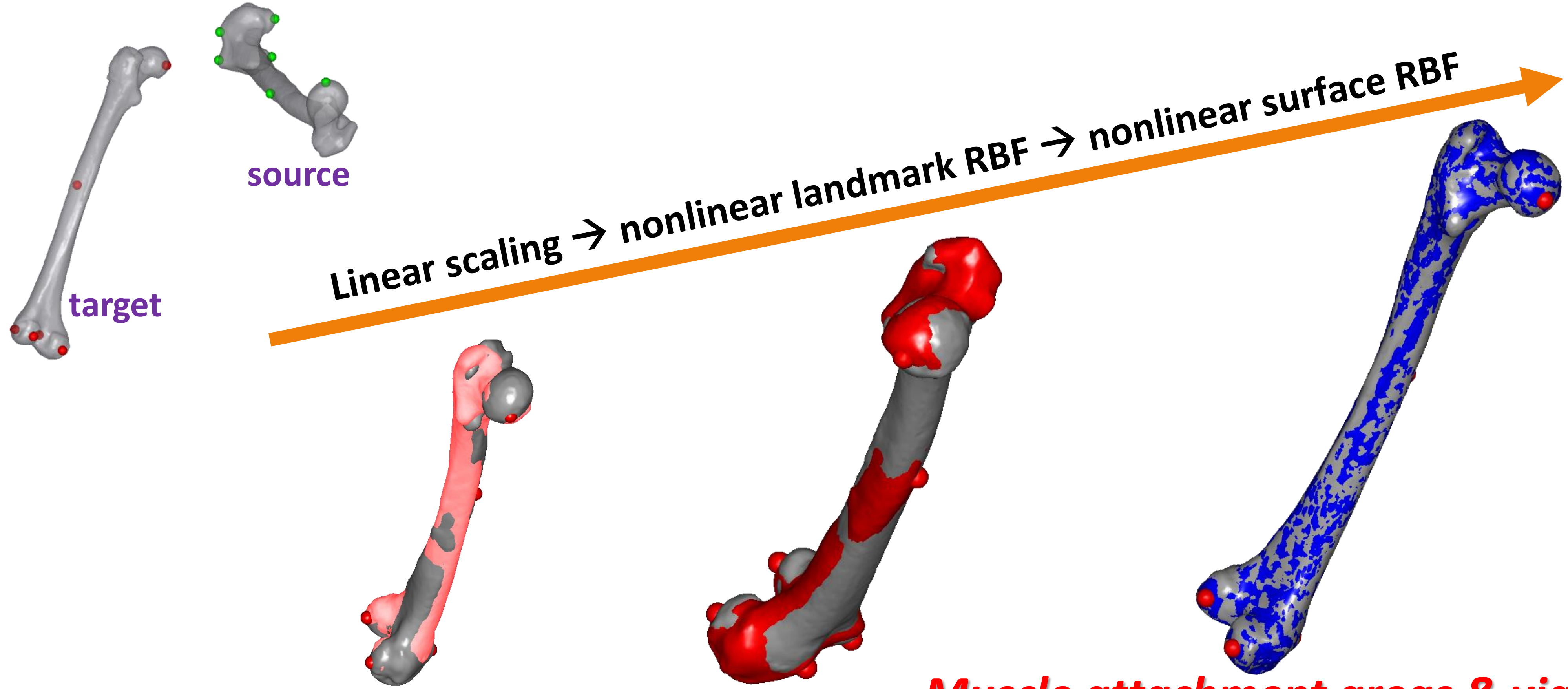
improved scaling

Different reference ref. frames

- anatomical
- segmental
- CT/MRI (target)

registration and reverse

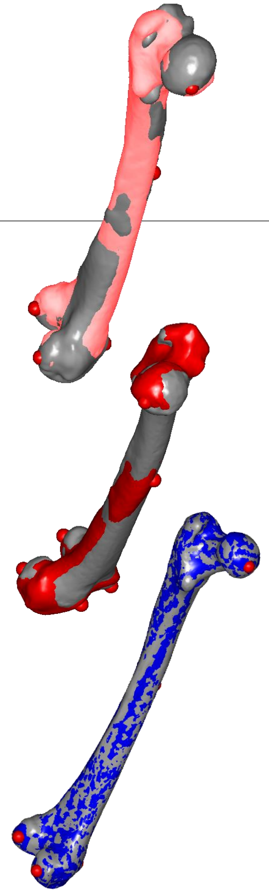
Individualizing segments: pipeline



Muscle attachment areas & via points follow!!!

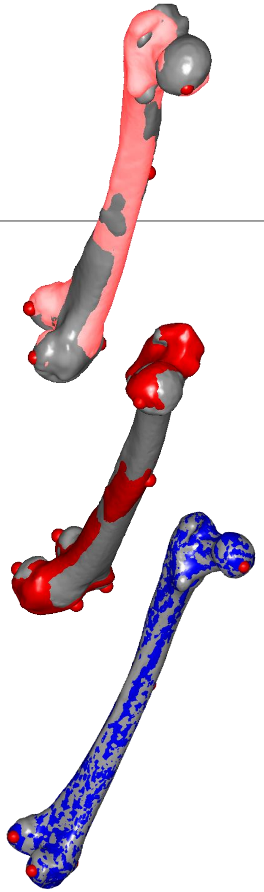
Individualization function code

```
AnyFunTransform3DLin2 MyAffineTransform = {
      Points0 = {...};
      Points1 = {...};
      Mode = VTK_LANDMARK_AFFINE;
    };
AnyFunTransform3DRBF MyRBFTransform = {
      PreTransforms = {&.MyAffineTransform};
      Points0 = .MyAffineTransform.Points0;
      Points1 = .MyAffineTransform.Points1;
     };
AnyFunTransform3DSTL MySTLTransform = {
      PreTransforms = {&.MyRBFTransform};
      FileName0 = "SourceFemur.stl";
      FileName1 = "TargetFemur.stl";
      ...
    };
AnyFunTransform3DLin2 MyReverseTransform = {
      Points0 = .MyTransform.Points1;
      Points1 = .MyTransform.Points0;
      Mode = VTK_LANDMARK_RIGIDBODY;
    };
AnyFunTransform3DIdentity Final = {
      PreTransforms = {&.MySTLTransform,&.MyReverseTransform};
    };
```



Individualization function code

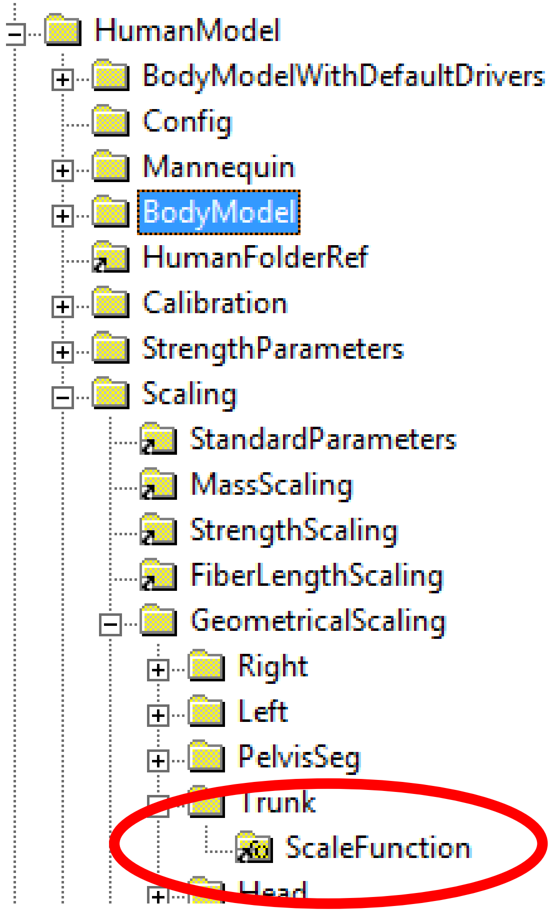
```
AnyFunTransform3DLin2 MyAffineTransform = {
       Points0 = {...};
       Points1 = {...};
       Mode = VTK_LANDMARK_AFFINE;
     };
AnyFunTransform3DRBF MyRBFTransform = {
       PreTransforms = {&.MyAffineTransform};
       Points0 = .MyAffineTransform.Points0;
       Points1 = .MyAffineTransform.Points1;
      };
AnyFunTransform3DSTL MySTLTransform = {
       PreTransforms = {&.MyRBFTransform};
       FileName0 = "SourceFemur.stl";
       FileName1 = "TargetFemur.stl";
       ...
     };
AnyFunTransform3DLin2 MyReverseTransform = {
       Points0 = .MyTransform.Points1;
       Points1 = .MyTransform.Points0;
       Mode = VTK_LANDMARK_RIGIDBODY;
     };
AnyFunTransform3DIdentity Final = {
       PreTransforms = {&.MySTLTransform,&.MyReverseTransform};
     };
```



This code can be reused and rarely changes!!

Input: Points0, Points1, FileName0, FileName1

Including custom scaling function



Configuration parameter

```
#define CUSTOM_SCALING_PelvisSeg
Main.HumanModel.Scaling.GeometricalScaling.PelvisSeg = {
  AnyFunTransform3D &AnthropometricScaling =..Scaling.GeomScaling.PelvisSeg.ScaleFunction;
  AnyFunTransform3DRBF ScaleFunction = {
    PreTransforms={&.AnthropometricScaling};
    RBFDef.Type = RBF_Gaussian;
    Points0 =..OriginalBone.Points0;
    Points1 = ..TargetBone.Points1;
  };
};
```

Anthropometric law added to pipeline

Actual scaling function

This code can be reused and rarely changes!!

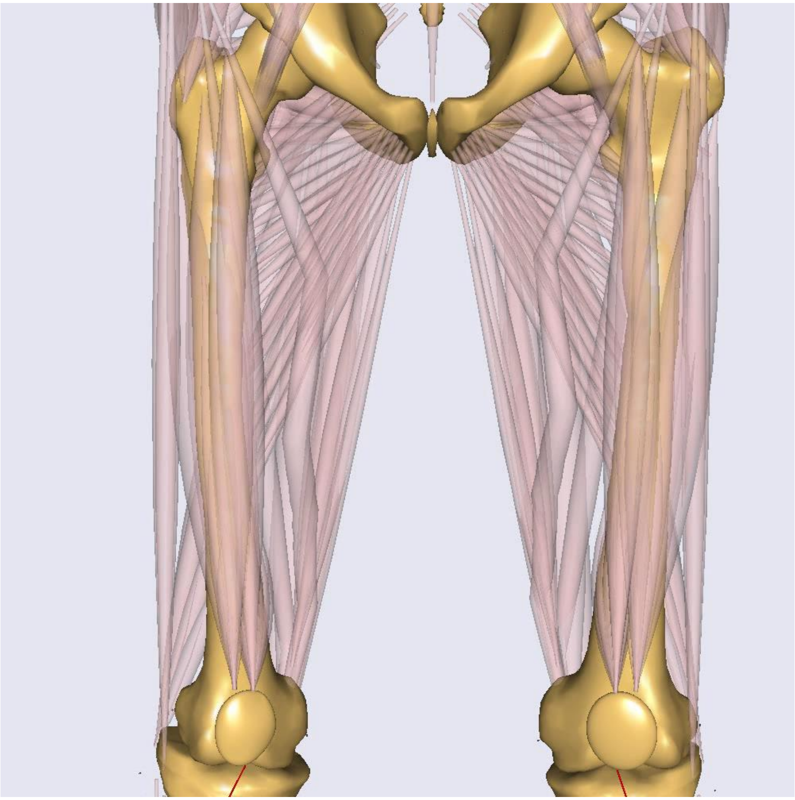
More information in tutorials!

Summary

- Choose an anthropometric scaling law
- Copy and modify an AnyMan file
  - `#define BM_SCALING CONST_SCALING_UNIFORM`
  - `#path BM_SCALING_ANTHRO_FILE "Model\AnyFamily\AnyManUniform.any"`
- Segmental individualization
  - Use proposed 3D transformation pipeline template
  - Provide 2 matrices of landmark coordinates (source/target)
  - Provide 2 filenames of the source and target STLs
- Incorporate new 3D scaling function into your model using precooked code

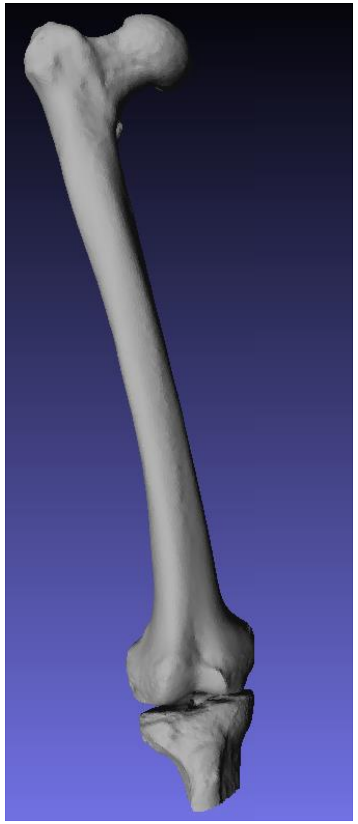
Summary

- Segmental individualization
  - Use proposed 3D transformation pipeline template
  - Provide 2 matrices of landmark coordinates (source/target)
  - Provide 2 filenames of the source and target STLs
- Possibility to automate this process
  - Automatic selection of landmarks (vertex indices of STL) using ”STL_Vertices” function
  - Requires matching STL topology (same number of vertices/faces, consistent locations)
  - Comes down to a single parameter: target STL filename/subject ID
  - Potential use of ADaM, SOMA, in-house PCA datasets
  - Example: 21 subjects, from -2 to +2 std. Deviations



Demo

- Hypothetical scenario:
  - Available CT or MRI scan
  - Entire surface of the femur can be segmented
  - Only proximal tibial surface can be seen segmented
  - Available MoCap data (walking)
- Task: to build a subject-specific musculoskeletal model
  - Step #0: Change anthropometrics (skipping, explained in the tutorials)
  - Step #1: Morph femur of the generic model to match subject’s one
  - Step #2: Morph tibia in the best possible way based partial surface information
  - Step #3: Combine MoCap segment length optimization and femur morphing



Solution, step #1

Normal approach is described in the tutorial,
Source and target bone are given as meshes (STL) and share the same topology (same vertex/face number + vertices have the same anatomic meaning)

```
AnyFile src = SOURCE_FEMUR;
AnyFile trg = TARGET_FEMUR;

AnyInt size = STL_Size(src, 1);    // get size of the stl
AnyInt VertNum = 200;              // specify landmark number to be extracted automatically
AnyInt vertices = iarr(1,VertNum)*floor(size[0]/VertNum);// evenly space landmarks on surface

AnyMatrix P0 = STL_Vertices(src, vertices , 1);          // extract vertices with given indices
AnyMatrix P1 = 0.001*STL_Vertices(trg, vertices , 1);    // conversion from mm to m
```

Solution, step #1

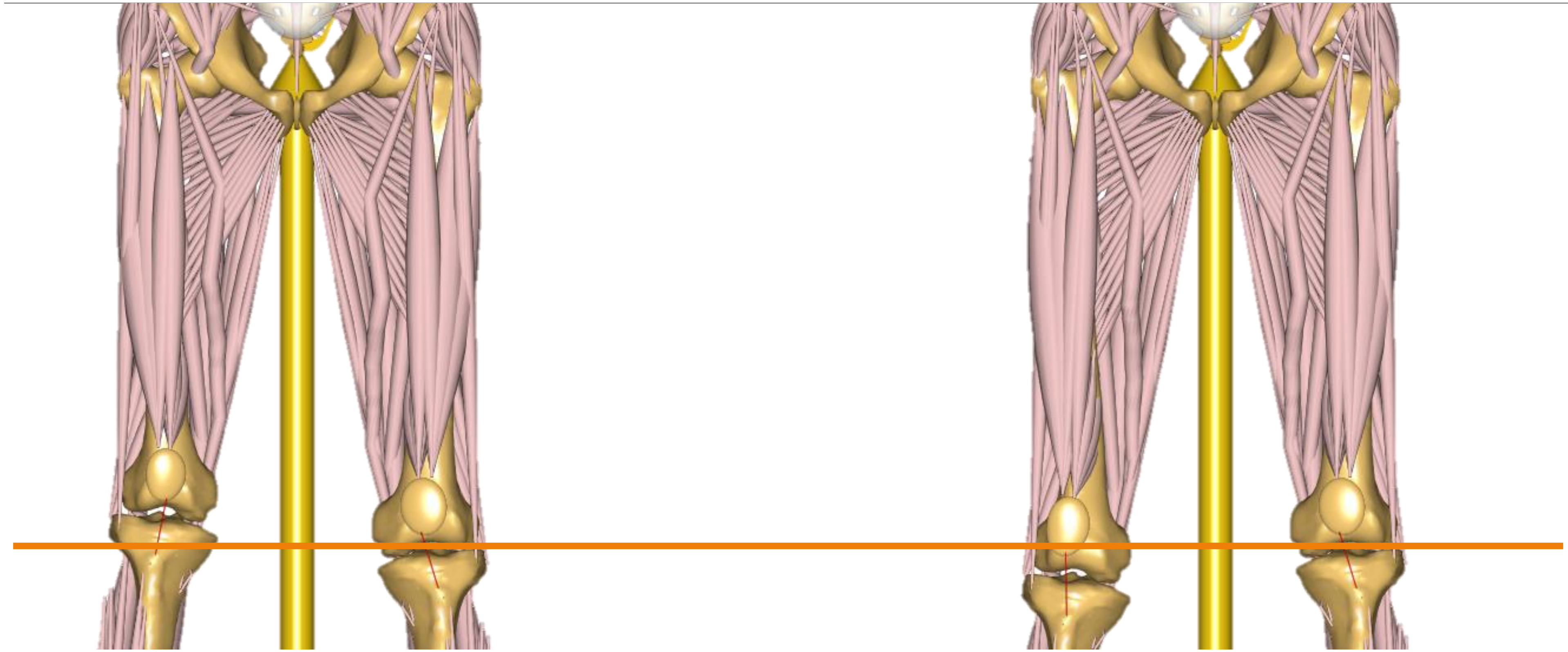
```
AnyFunTransform3DLin2 reg = {
  Points0 = .TSeg2ScaleFrame(.P0);
  Points1 = .P1;
  Mode = VTK_LANDMARK_AFFINE;
};
AnyFunTransform3DRBF rbf =
{
  PreTransforms = {&.reg};
  RBFDef.Type = RBF_Triharmonic;
  PolynomDegree = 1;
  Points0 = .reg.Points0;
  Points1 = .reg.Points1;
  BoundingBoxOnOff = On;
  BoundingBox.Type = BB_Cartesian;
  BoundingBox.ScaleXYZ={1,2,2}*1.2;
  BoundingBox.DivisionFactorXYZ ={1,1,1}*3;
};
AnyFunTransform3DLin2 inv = {
  Points0 = .reg.Points1;
  Points1 = .reg.Points0;
  Mode = VTK_LANDMARK_RIGIDBODY;
};

AnyFunTransform3DIdentity ScalingFunction = {
  PreTransforms = {&.rbf, &.inv};
};
```

Normal pipeline of transformations, except surface/surface RBF functions

- not needed because we can select sufficient landmarks from surfaces with consistent topology

Results, step #1



#define TARGET_FEMUR ”Data/1_FemurR.stl”
female

#define TARGET_FEMUR ”Data/10_FemurR.stl”
male

Solution, step #2

```
AnyFunTransform3D &AnthropometricScaling =...Scaling.GeometricalScaling.Right.Shank.ScaleFunction;

AnyMatrix proximal_tibial_landmarks_source = {
{0.0421311, -0.417261    , 0.0076043},
....
AnyMatrix distal_tibial_landmarks_source = {
{0.082874 , -0.556316    , 0.0316961},
...
AnyMatrix proximal_tibial_landmarks_target = {
{-71.6207, 30.8218   , -776.802},
...
AnyFunTransform3DLin2 RegProximalLandmarks = {
// aligning proximal target bone with expected anthropometrically scaled generic bone
  Points0 = 0.001*.proximal_tibial_landmarks_target;
  Points1 = .AnthropometricScaling(.proximal_tibial_landmarks_source);
};

AnyMatrix P0 = arrcat(
                  distal_tibial_landmarks_source,
                  proximal_tibial_landmarks_source);
AnyMatrix P1 = arrcat(
                  AnthropometricScaling(distal_tibial_landmarks_source),
                  RegProximalLandmarks(0.001*proximal_tibial_landmarks_target));
```

Solution, step #2

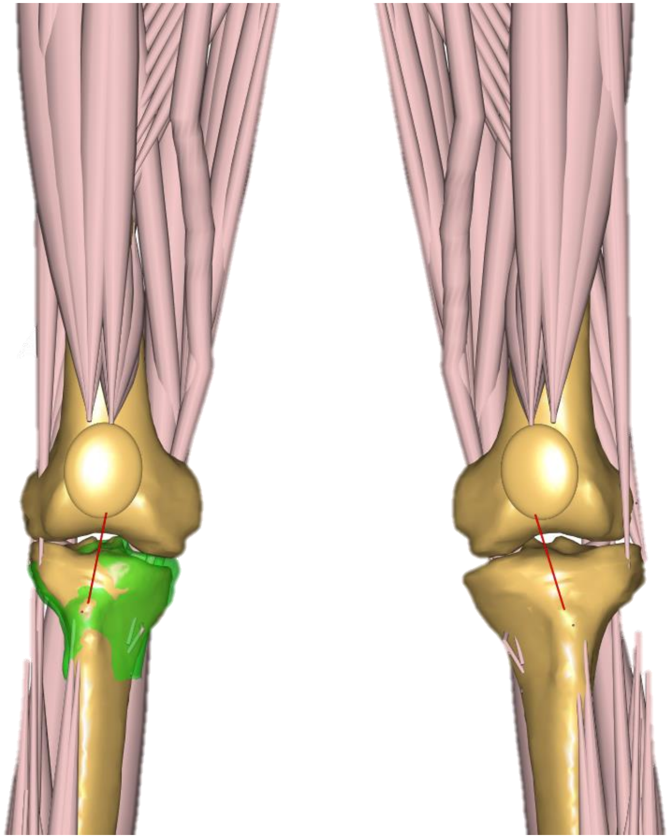
```
AnyFunTransform3DLin2 reg = {
  PreTransforms={&.AnthropometricScaling};
  Points0 = .TSeg2ScaleFrame(.P0);
  Points1 = .P1;
  Mode = VTK_LANDMARK_AFFINE;
};
AnyFunTransform3DRBF rbf = {    // we will only use an RBF
transformation, because full STL is not available
  PreTransforms = {&.reg};
  RBFDef.Type = RBF_Triharmonic;
  PolynomDegree = 1;
  Points0 = .reg.Points0;
  Points1 = .reg.Points1;
  BoundingBoxOnOff = On;
  BoundingBox.Type = BB_Cartesian;
  BoundingBox.ScaleXYZ={2,2,2}*1.2;
  BoundingBox.DivisionFactorXYZ ={1,1,1}*3;
};
AnyFunTransform3DLin2 inv = {
  Points0 = .reg.Points1;
  Points1 = .reg.Points0;
  Mode = VTK_LANDMARK_RIGIDBODY;
};

AnyFunTransform3DIdentity ScalingFunction = {
  PreTransforms = {&.rbf, &.inv};
};
```

Normal pipeline of transformations, except surface/surface RBF functions

- Is not used because we do not have sufficient surface information to use it

Results, step #2



Superimposed target surface (green)

Morphing is not perfect:

- possible cause – too few landmarks for the demo case
- their location is not optimal
- captures most of the sizes and can be used in combination with the real (green) surface for contact analyses

Solution, step #3

```
#define CUSTOM_SCALING_Right_Thigh

 Right.Thigh = {

   AnyFunTransform3D& TSeg2ScaleFrame = Main.Studies.HumanModel.BodyModel.Right.Leg.Seg.Thigh.Scale.T0;

   AnyFunTransform3DIdentity ScaleFunction = {
    PreTransforms = {&.rbf,&.inv, &.ScaleToOptimizedLength};
   };

   AnyFunTransform3DLin ScaleToOptimizedLength = {
     AnyVec3 HipJointUnscaled = .TSeg2ScaleFrame( Main.Studies.HumanModel.BodyModel.Right.Leg.Seg.Thigh.HipJoint.sRelUnscaled );
     AnyVec3 KneeJointUnscaled = .TSeg2ScaleFrame( Main.Studies.HumanModel.BodyModel.Right.Leg.Seg.Thigh.KneeJoint.sRelUnscaled );

     AnyVar OptimizedLength = Main.Studies.HumanModel.Scaling.AnthroSegmentLengths.Right.ThighLength;
     AnyVar InitialLength = vnorm( .rbf( HipJointUnscaled ) - .rbf(KneeJointUnscaled) );
     AnyVar ScalingFactor = OptimizedLength/InitialLength;

    ScaleMat = ScalingFactor * {{1,0,0},{0,1,0},{0,0,1}};
    Offset = {0,0,0};
   };

   #include "../../StandingModel/Morphing/FemurScalingFunction.auto.any"
 };
```

Refer to the previously explained morphing function from step 1.

Uniformly scaling a morphed bone to minimize kinematic errors

Results, step #3

Demonstration in the AnyBody Modeling System

Conclusions

- One can use bone surfaces segmented from CT/MRI to personalize and improve a MS model
- The entire procedure comes down to:
  - Reuse available code
  - Select a corresponding set of landmarks in an automated/semi-automated manner
  - In special cases may require combining different geometric information to fill the missing gaps
- *Personalization gives a great value at a relatively low cost!*

Questions?

Visit:

www.youtube.com/anybodytech

www.anybodytech.com

Acknowledgements:

- Emmanuel Audenaert and Jan Van Houcke, University of Ghent
- Sebastian Dendorfer, OTH Regensburg
- TLEMsafe project