Fundamentals of Computer Graphics

Shape visualization II

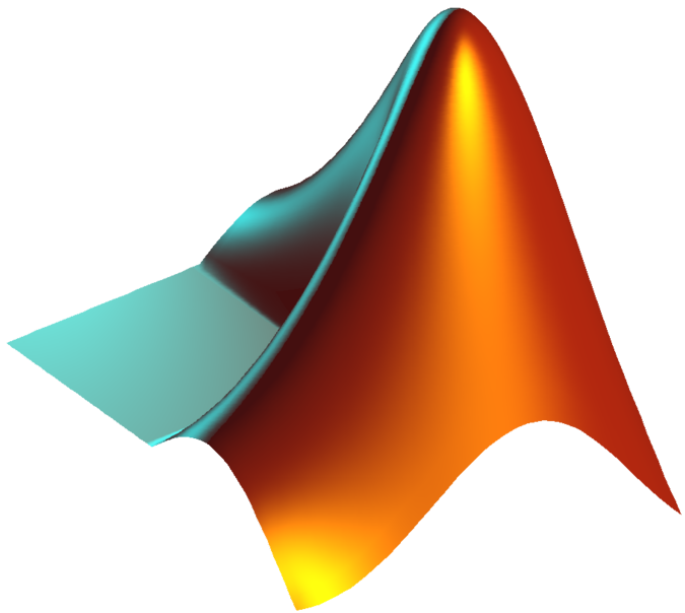
Emanuele Rodolà
rodola@di.uniroma1.it

SAPIENZA
UNIVERSITÀ DI ROMA

1st semester a.y. 2018/2019 · October 15, 2018

Exercises

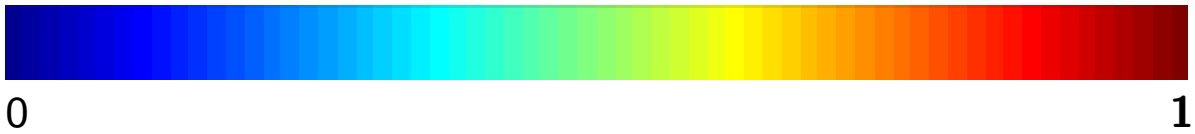
- Voronoi basis
- ICP



Colormap limits

Sometimes it is useful to modify the colormap to get saturation effects

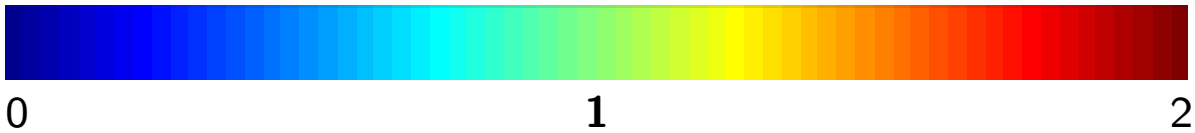
We can increase the maximum value represented in the colormap:



Colormap limits

Sometimes it is useful to modify the colormap to get saturation effects

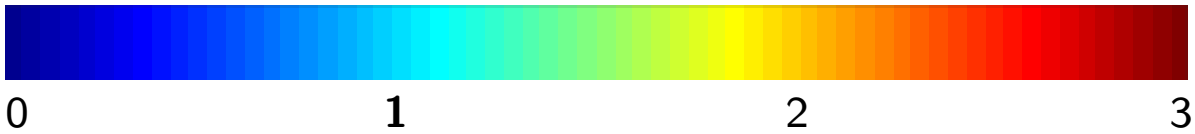
We can increase the maximum value represented in the colormap:



Colormap limits

Sometimes it is useful to modify the colormap to get saturation effects

We can increase the maximum value represented in the colormap:

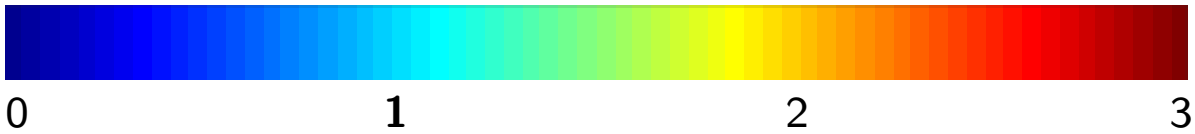


The colors remain fixed, but the mapping from values to colors changes

Colormap limits

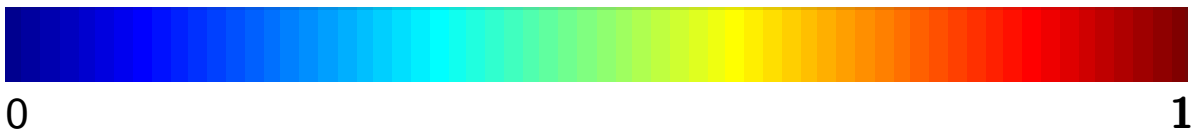
Sometimes it is useful to modify the colormap to get saturation effects

We can increase the maximum value represented in the colormap:



The colors remain fixed, but the mapping from values to colors changes

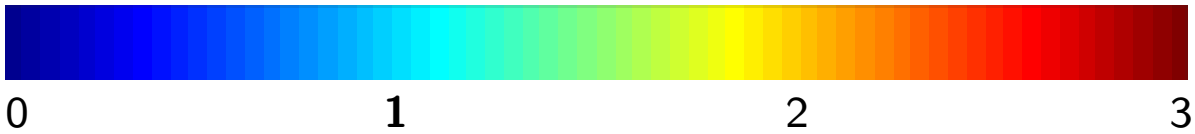
Similarly, we can decrease the max limit:



Colormap limits

Sometimes it is useful to modify the colormap to get saturation effects

We can increase the maximum value represented in the colormap:



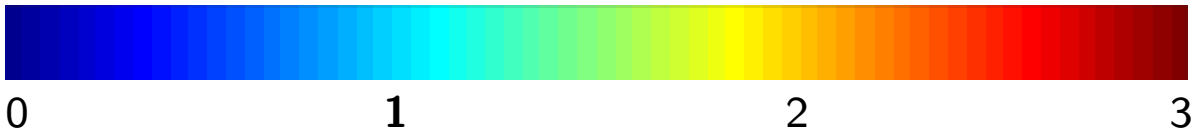
The colors remain fixed, but the mapping from values to colors changes

Similarly, we can decrease the max limit:

Colormap limits

Sometimes it is useful to modify the colormap to get saturation effects

We can increase the maximum value represented in the colormap:



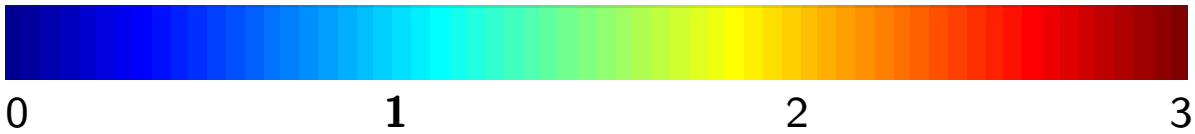
The colors remain fixed, but the mapping from values to colors changes

Similarly, we can decrease the max limit:

Colormap limits

Sometimes it is useful to modify the colormap to get saturation effects

We can increase the maximum value represented in the colormap:

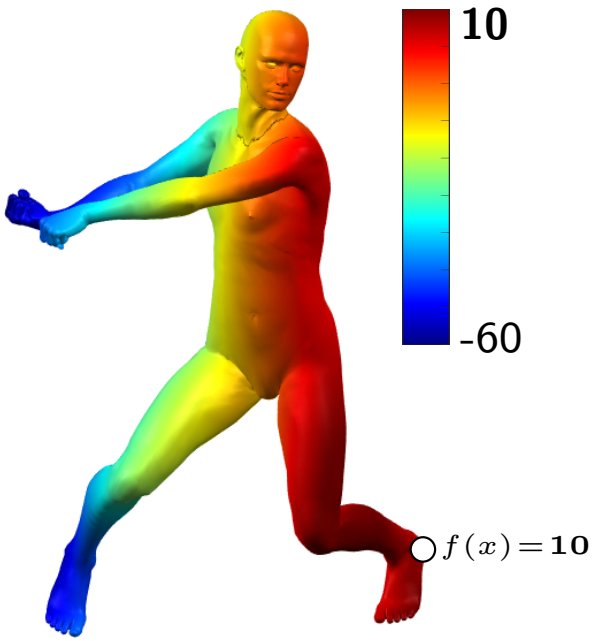


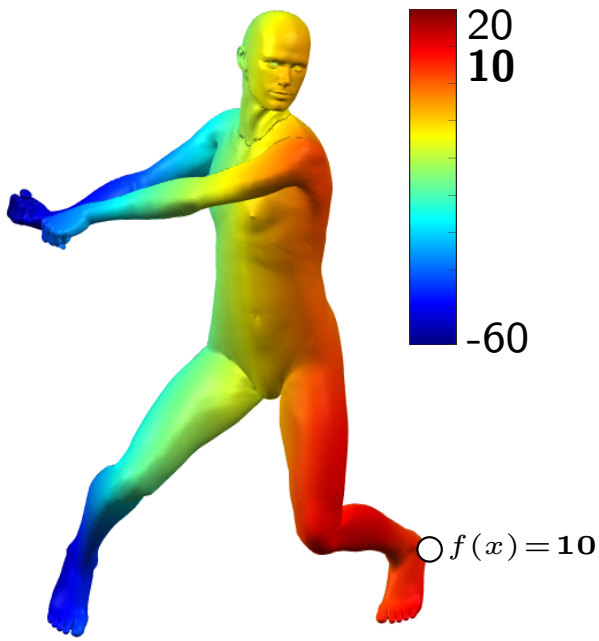
The colors remain fixed, but the mapping from values to colors changes

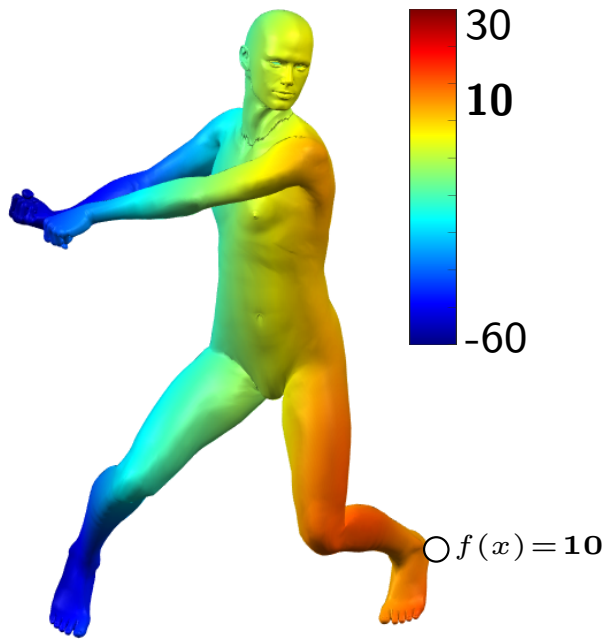
Similarly, we can decrease the max limit:

- Values $\geq$ the max limit are mapped to the max
- Values $\leq$ the min limit are mapped to the min
- Values between min and max are linearly interpolated

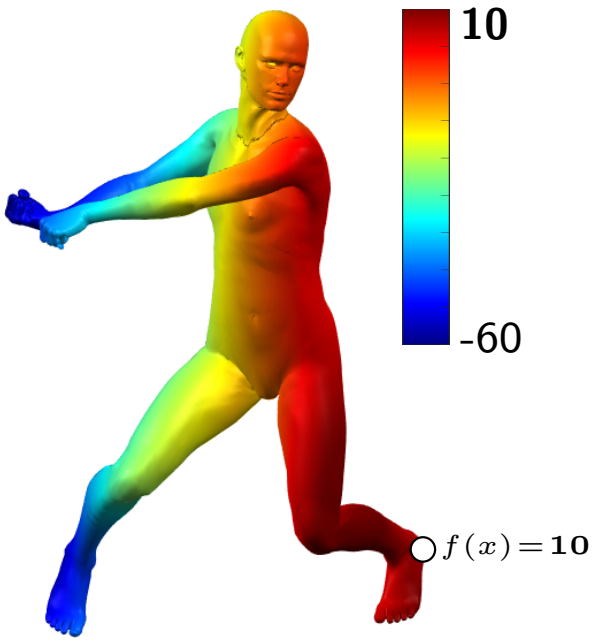
Increasing the max limit

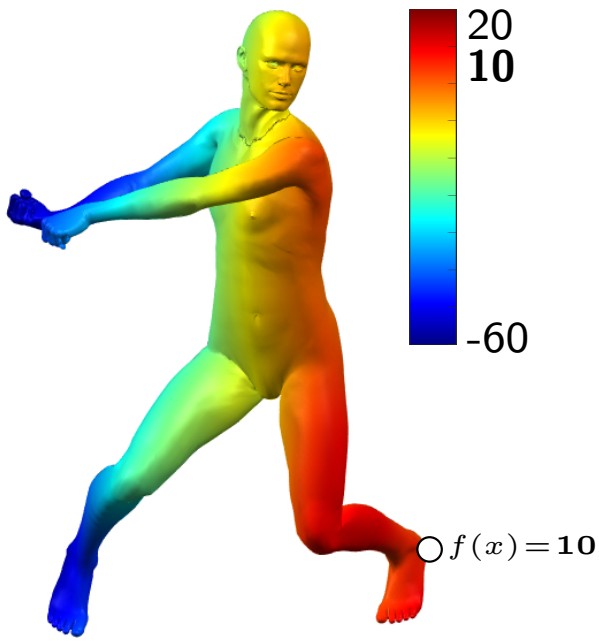


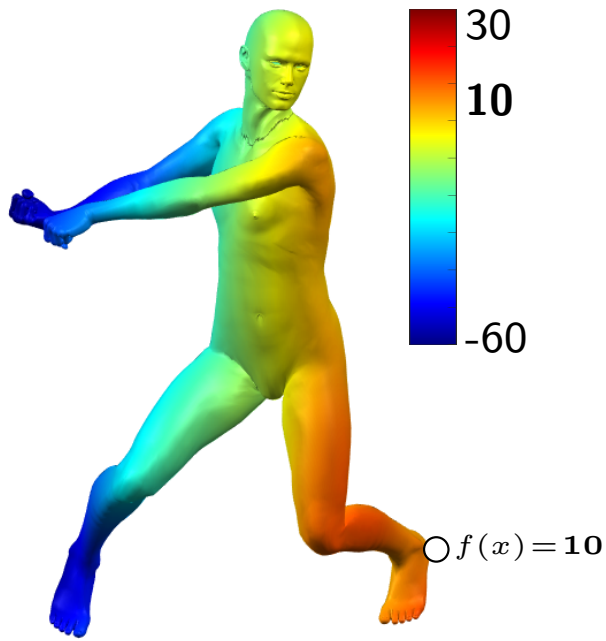




Increasing the max limit

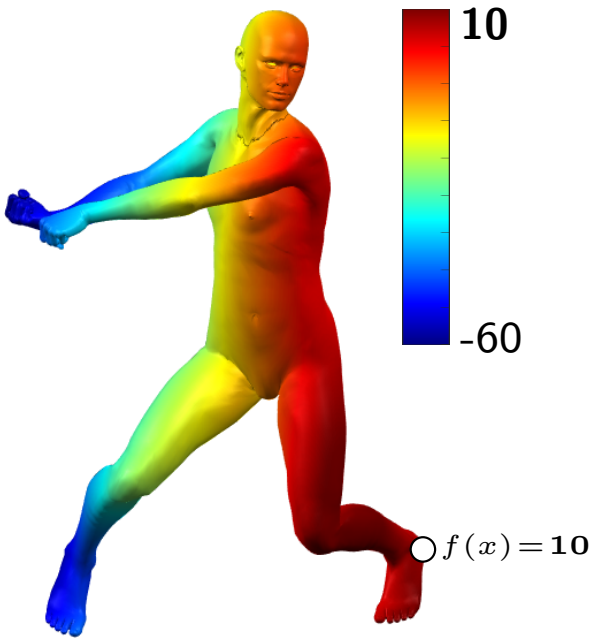


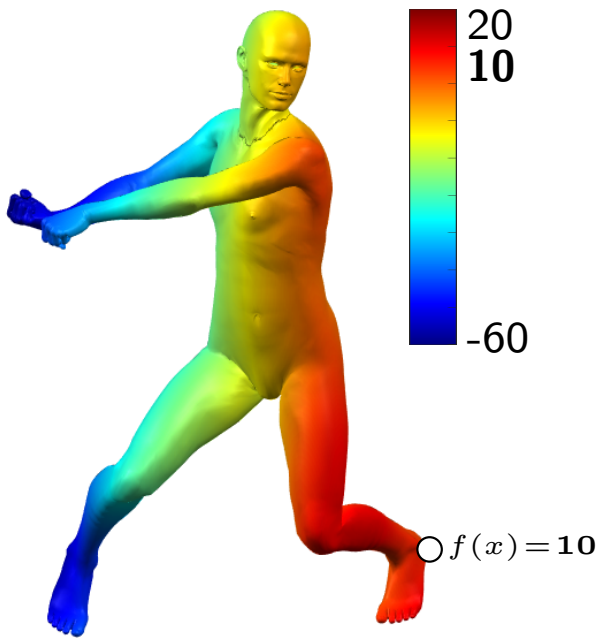


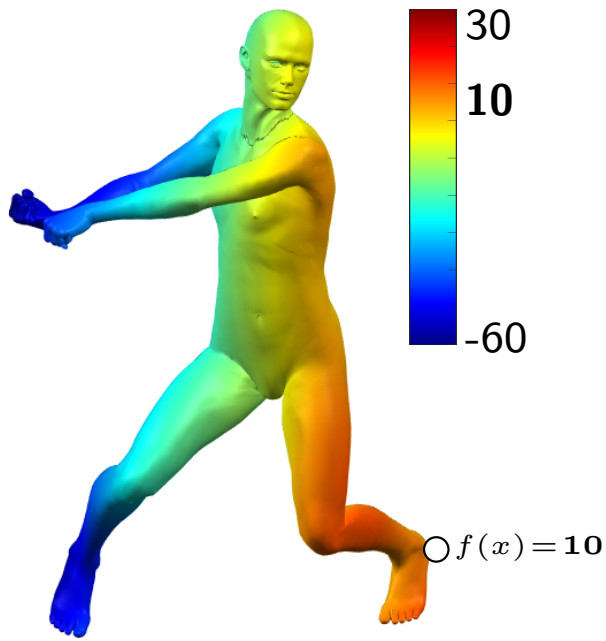


For smooth colormaps, the effect is to "flatten out" the colors and make variations less evident

Increasing the max limit



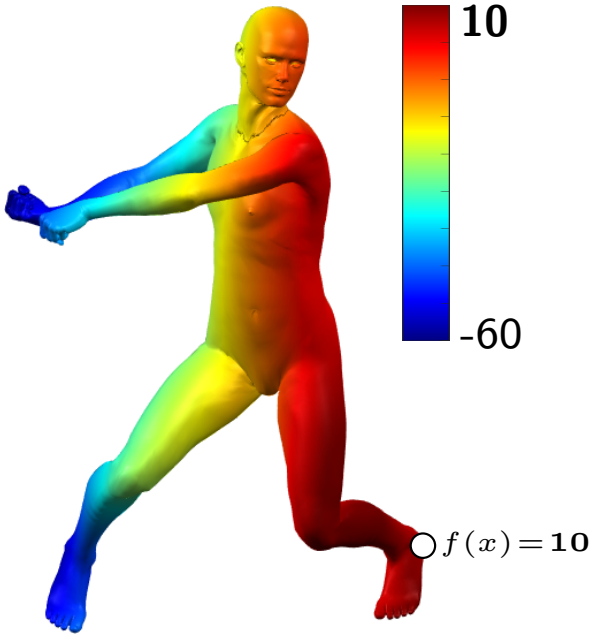


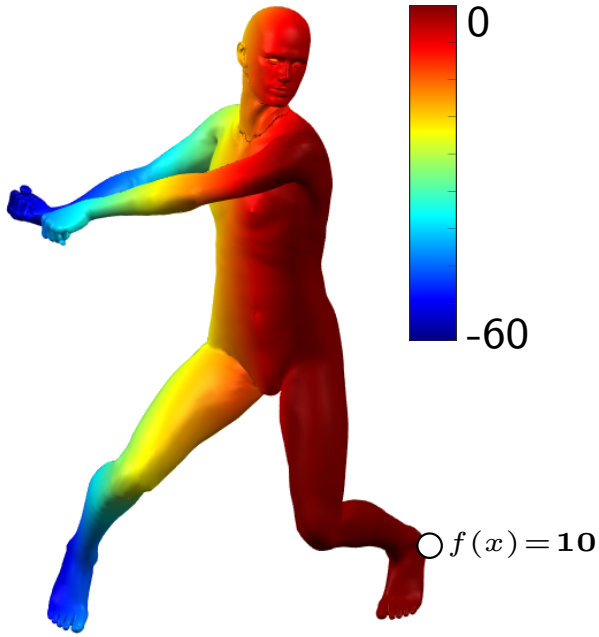


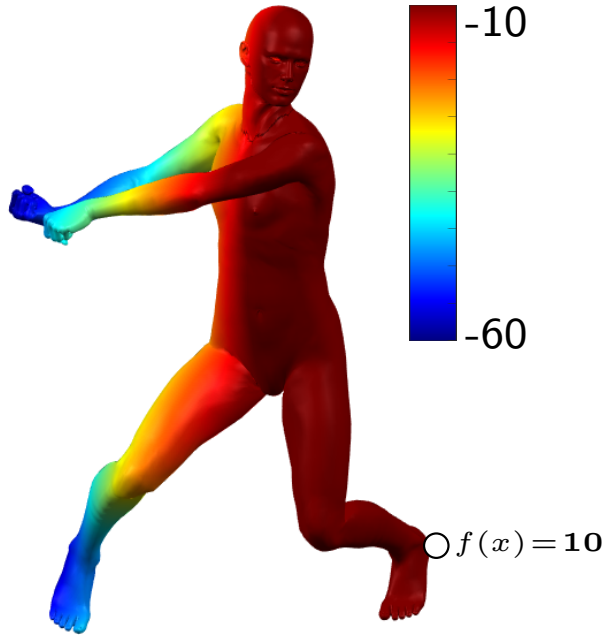
For smooth colormaps, the effect is to "flatten out" the colors and make variations less evident

In Matlab: caxis ([min max]) sets the colormap limits and does all the saturation and interpolation for us

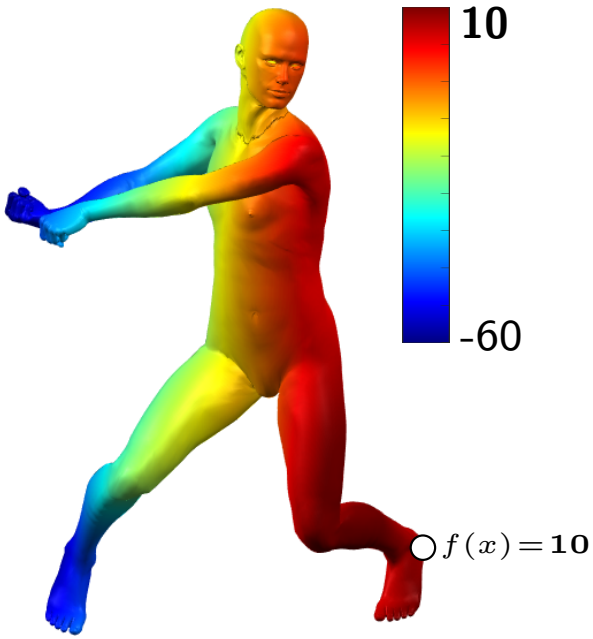
Decreasing the max limit

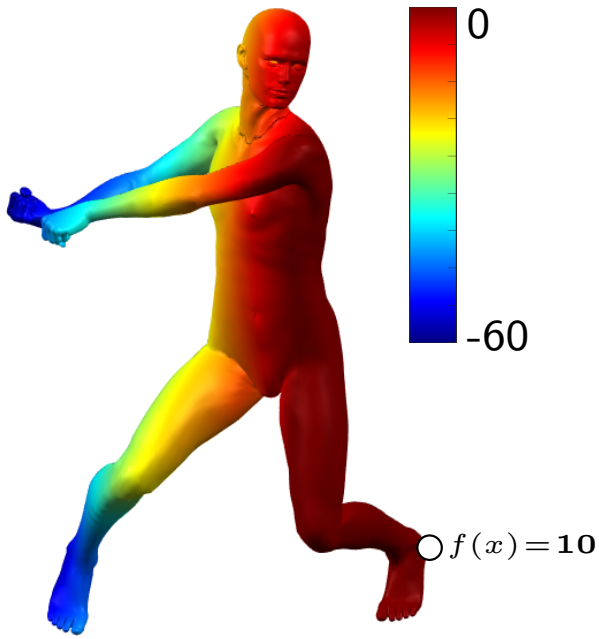


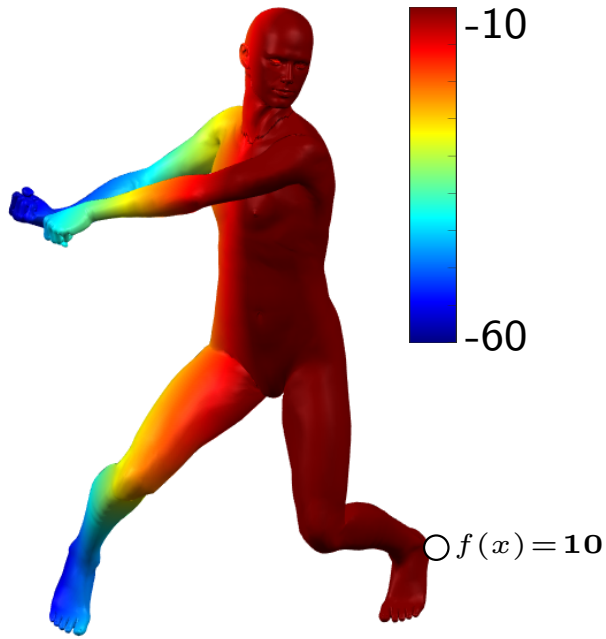




Decreasing the max limit



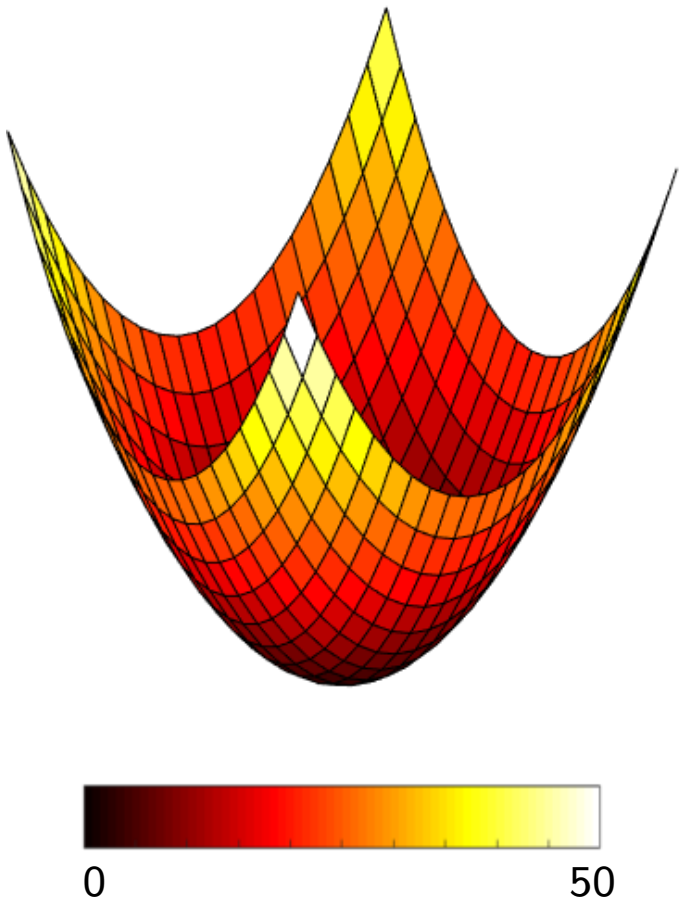


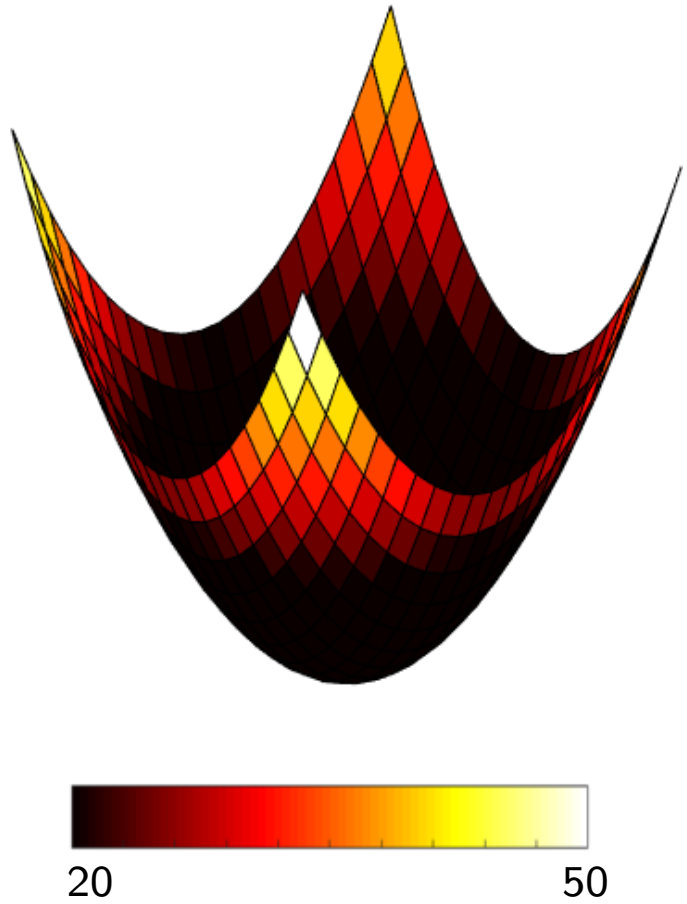


For smooth colormaps, the effect is to saturate the colors and make variations more evident

Example: Visualizing maxima

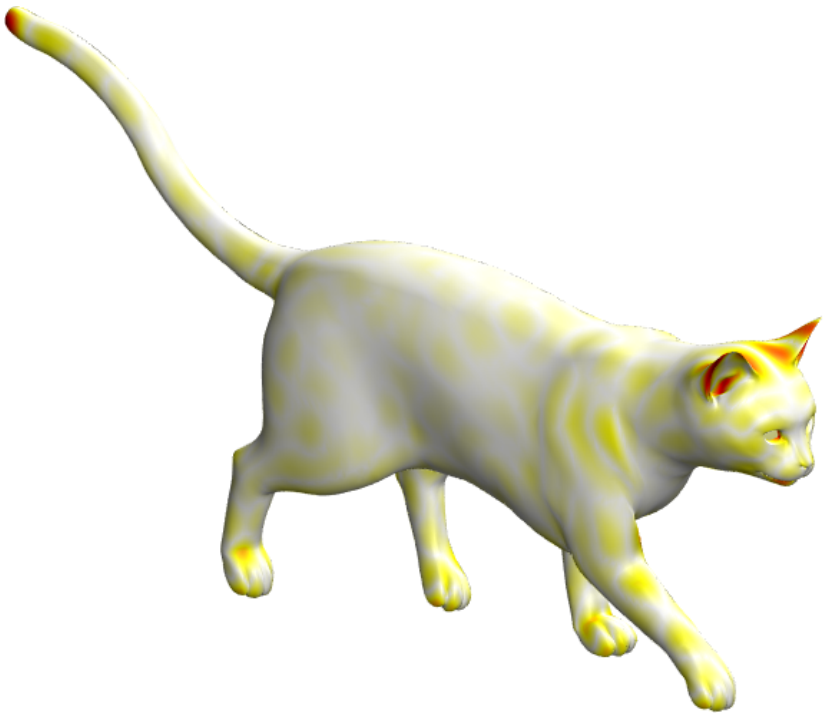
Increasing the min limit can be useful for visualizing maxima





Example: Visualizing point-wise error

Emphasize areas of large error:



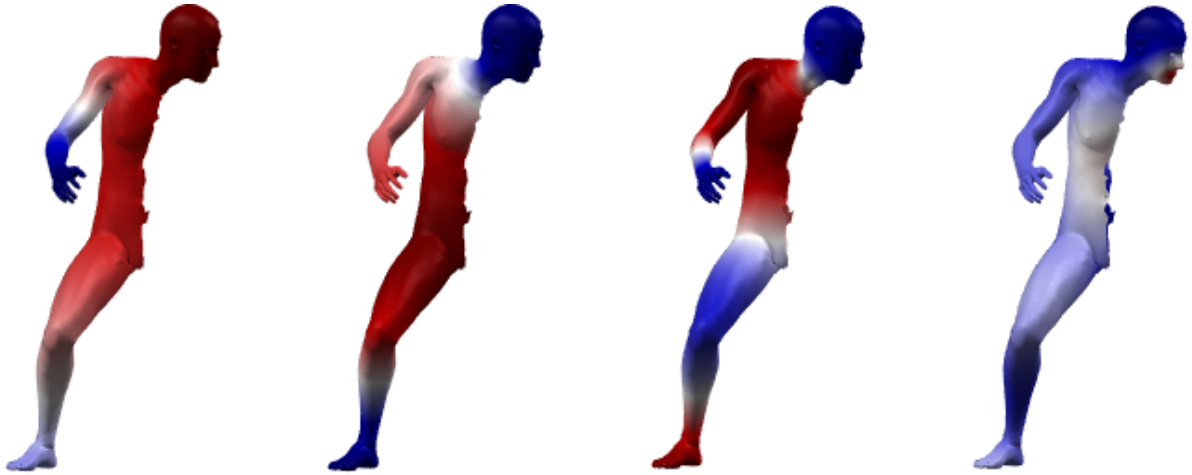
no saturation

saturate at 0.5

Example: Zero-centered functions

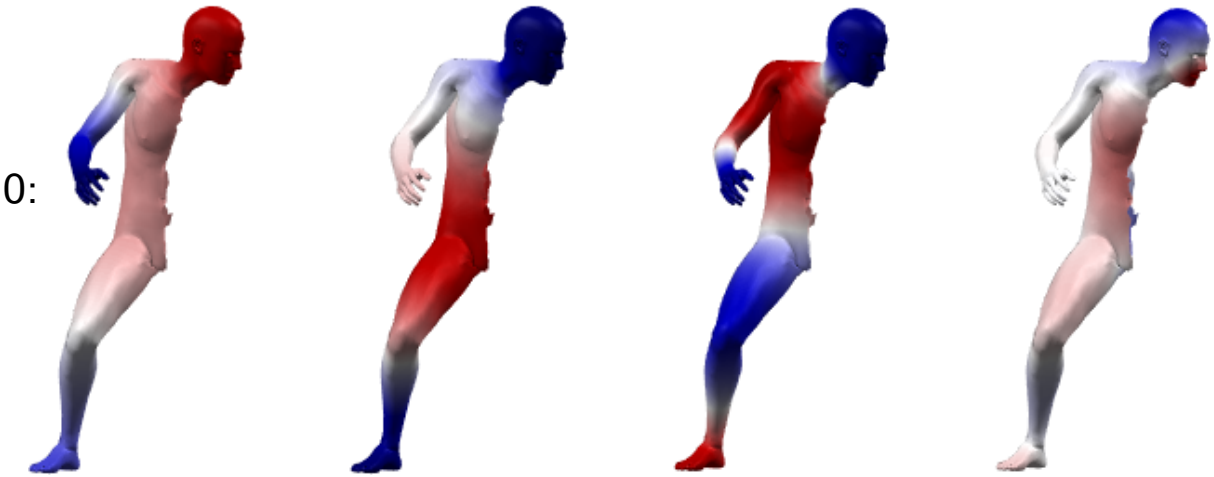
If $f :\to [-1, 1]$ but f is not surjective, it can be useful to recenter colors

not centered:



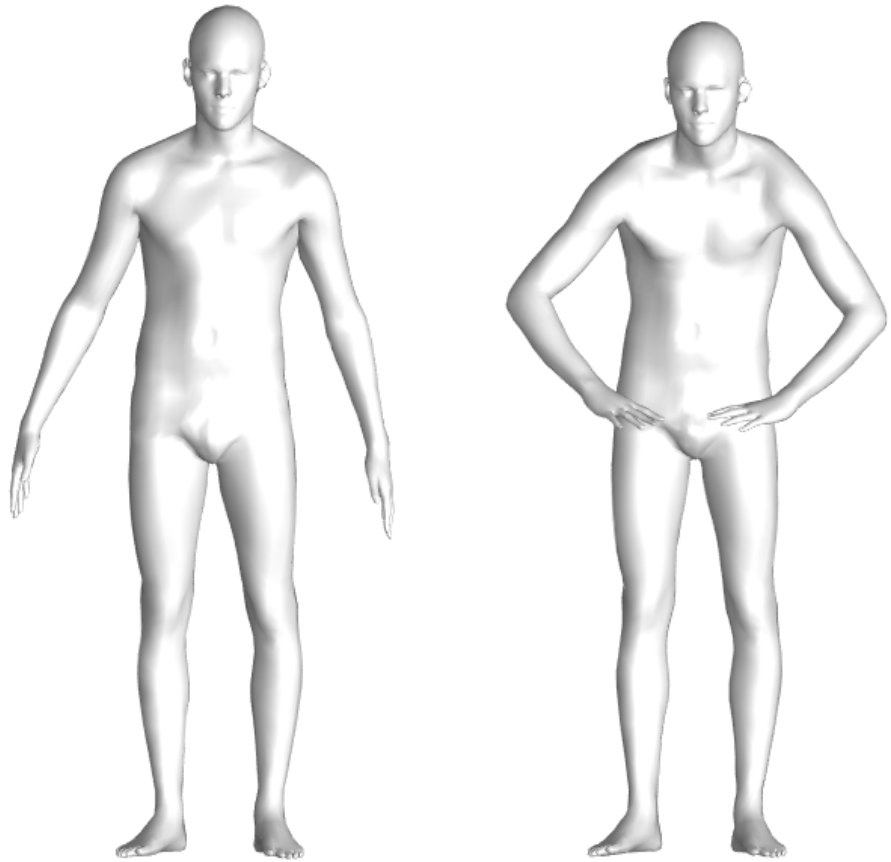
f_1 f_2 f_3 f_4

centered at 0:



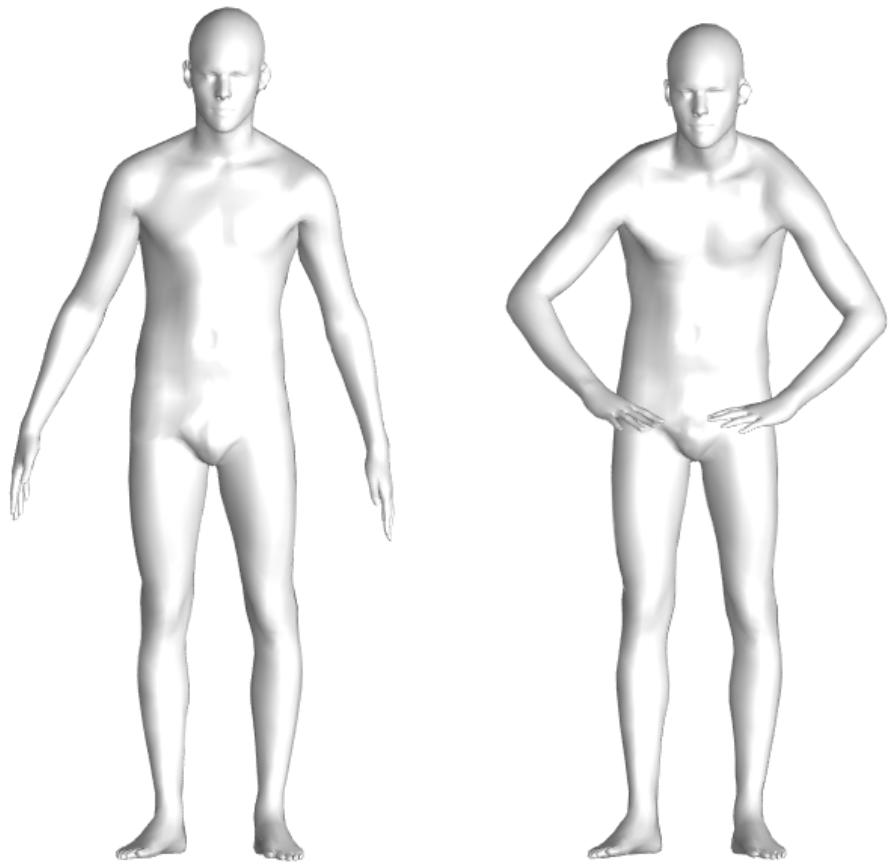
Visualizing correspondence

How to visualize correspondence between shapes?



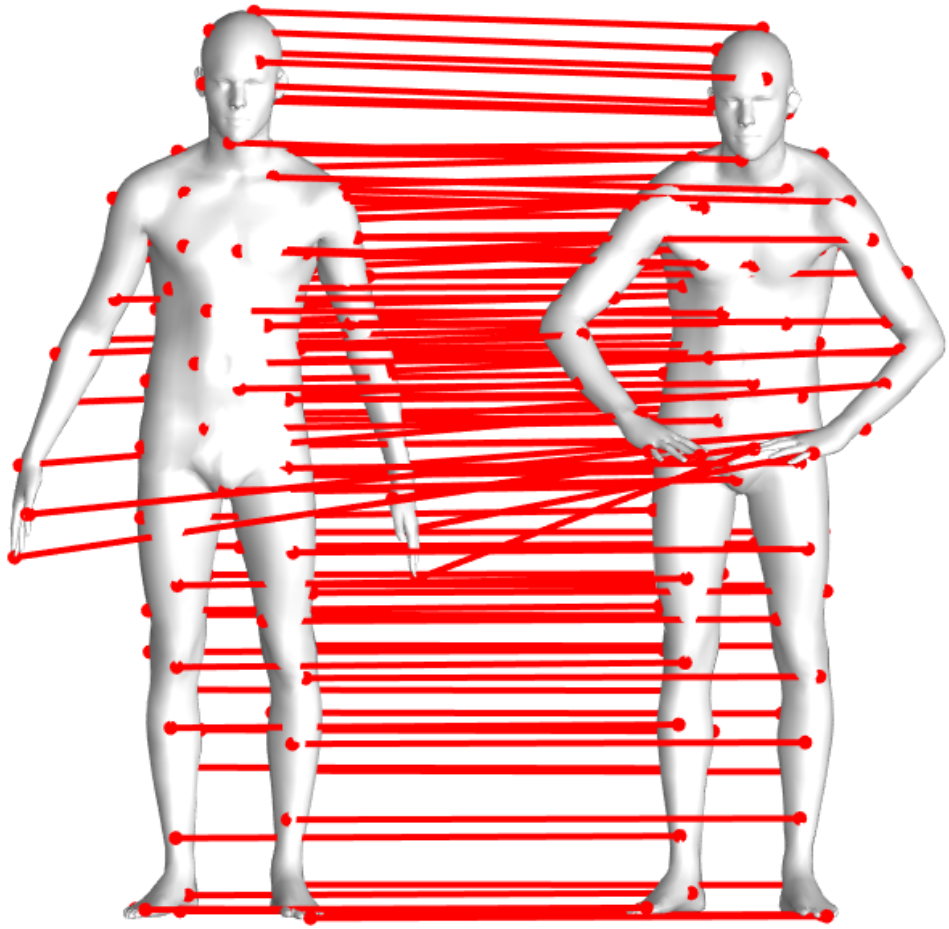
Visualizing correspondence

How to visualize correspondence between shapes?

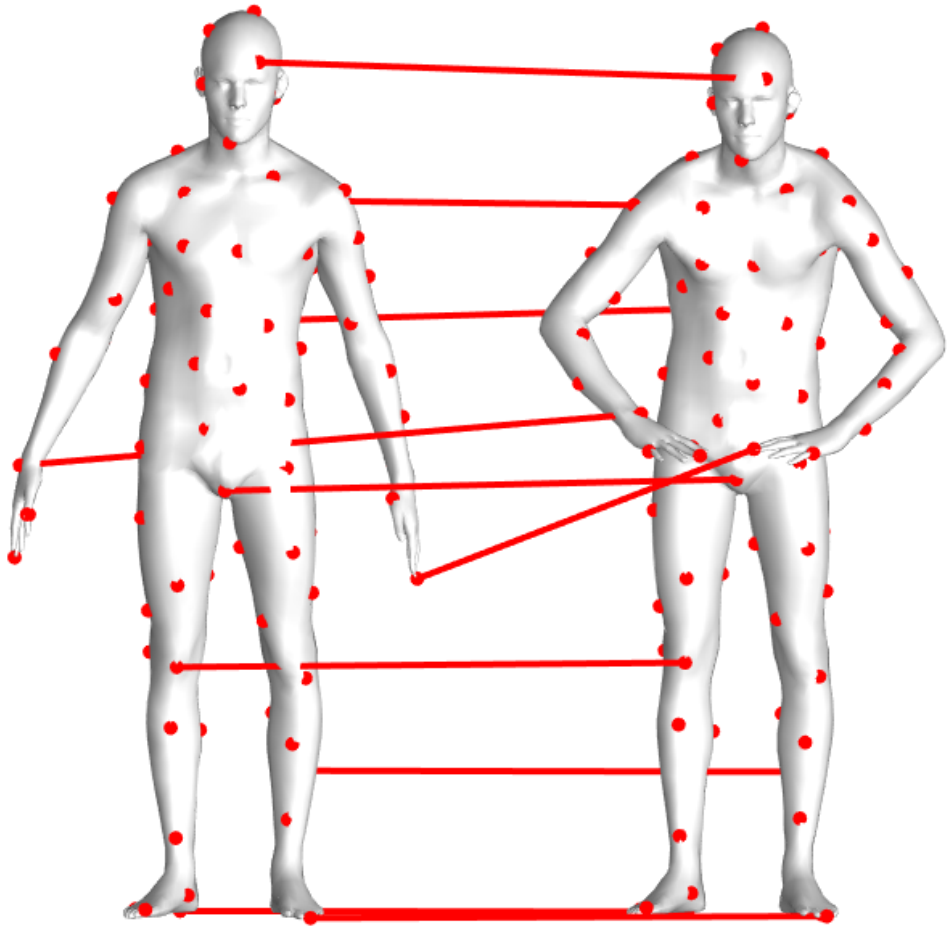


- It might be not one-to-one
- It might be not surjective (formally, not a correspondence)
- It might be sparse

Visualizing correspondence: Points and lines

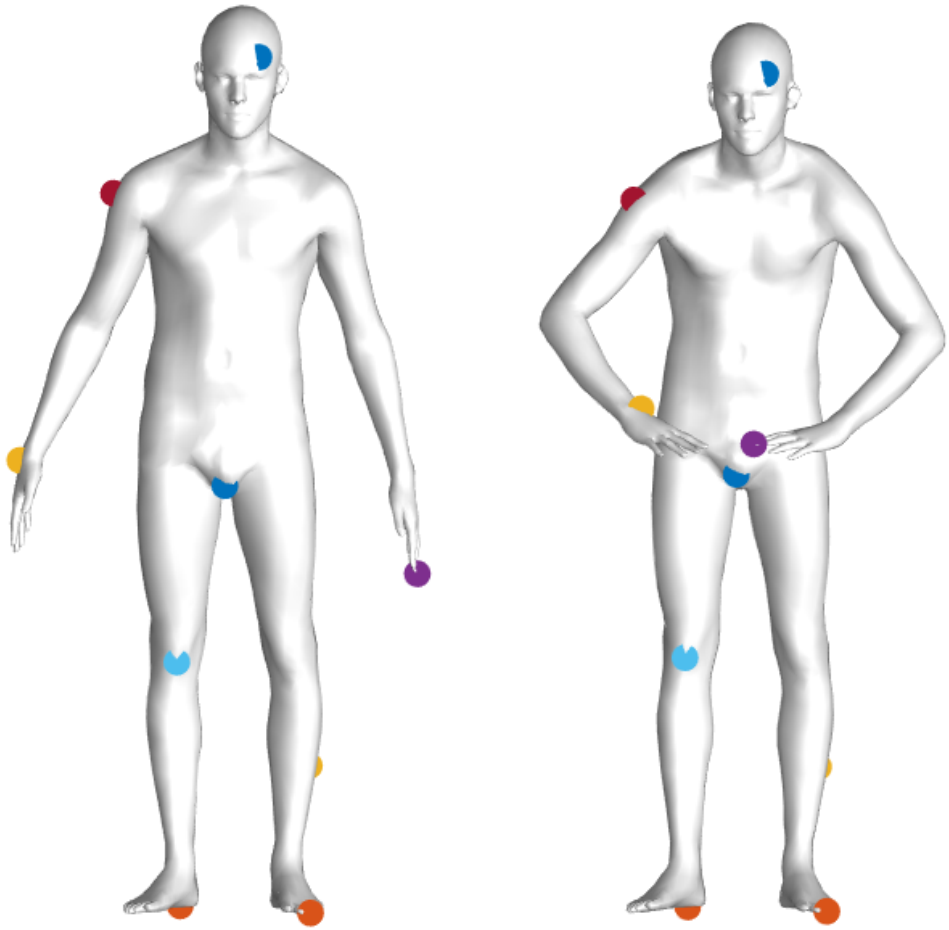


Visualizing correspondence: Points and lines



Useful for visualizing sparse matches

Visualizing correspondence: Points and colors



Useful for visualizing sparse matches (e.g., between landmarks)

Visualizing correspondence: Dense colors

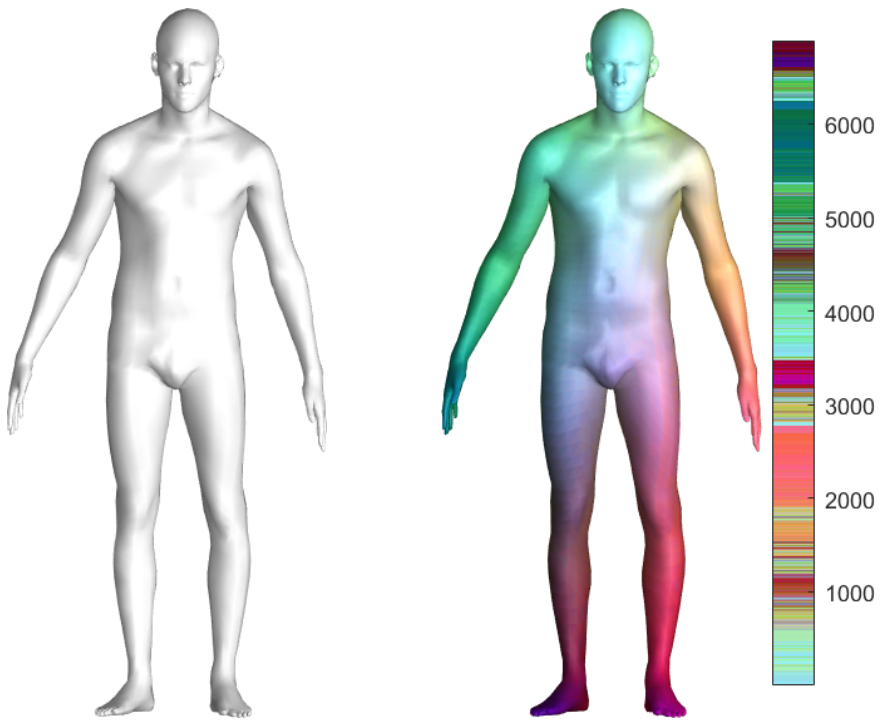
Dense matches (∼all points) can be used to color the entire mesh

- Create a colormap with one color for each point ($n \times 3$ matrix)

Visualizing correspondence: Dense colors

Dense matches (∼all points) can be used to color the entire mesh

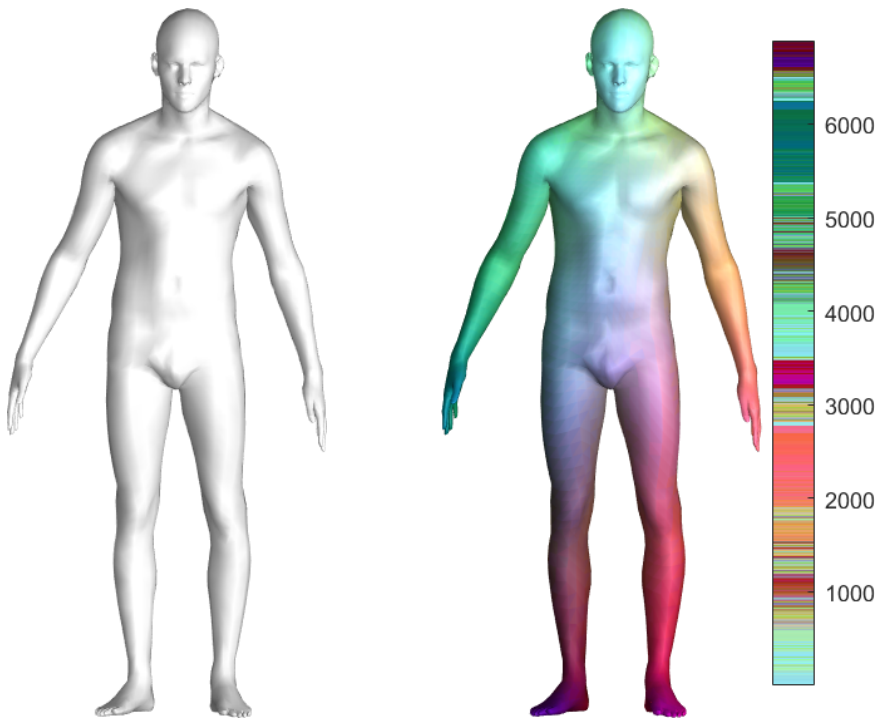
- Create a colormap with one color for each point ($n \times 3$ matrix)
- Plot the vector $(1, 2, \ldots, n)$



Visualizing correspondence: Dense colors

Dense matches (∼all points) can be used to color the entire mesh

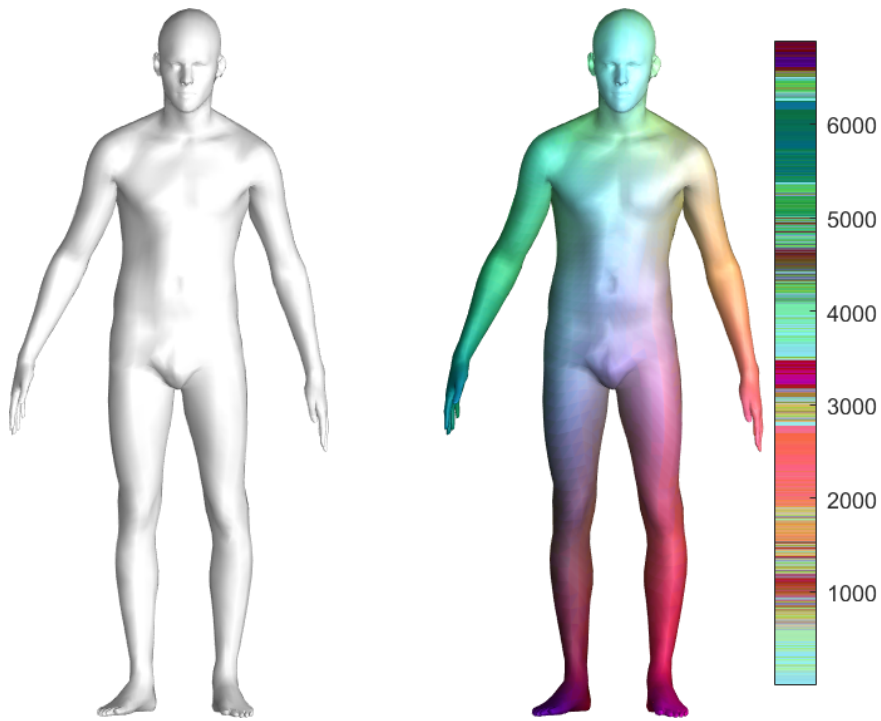
- Create a colormap with one color for each point ($n \times 3$ matrix)
- Plot the vector $(1, 2, \ldots, n)$
- For smooth colors: interpret x, y, z as R, G, B



Visualizing correspondence: Dense colors

Dense matches (∼all points) can be used to color the entire mesh

- Create a colormap with one color for each point ($n \times 3$ matrix)
- Plot the vector $(1, 2, \ldots, n)$
- For smooth colors: interpret x, y, z as R, G, B

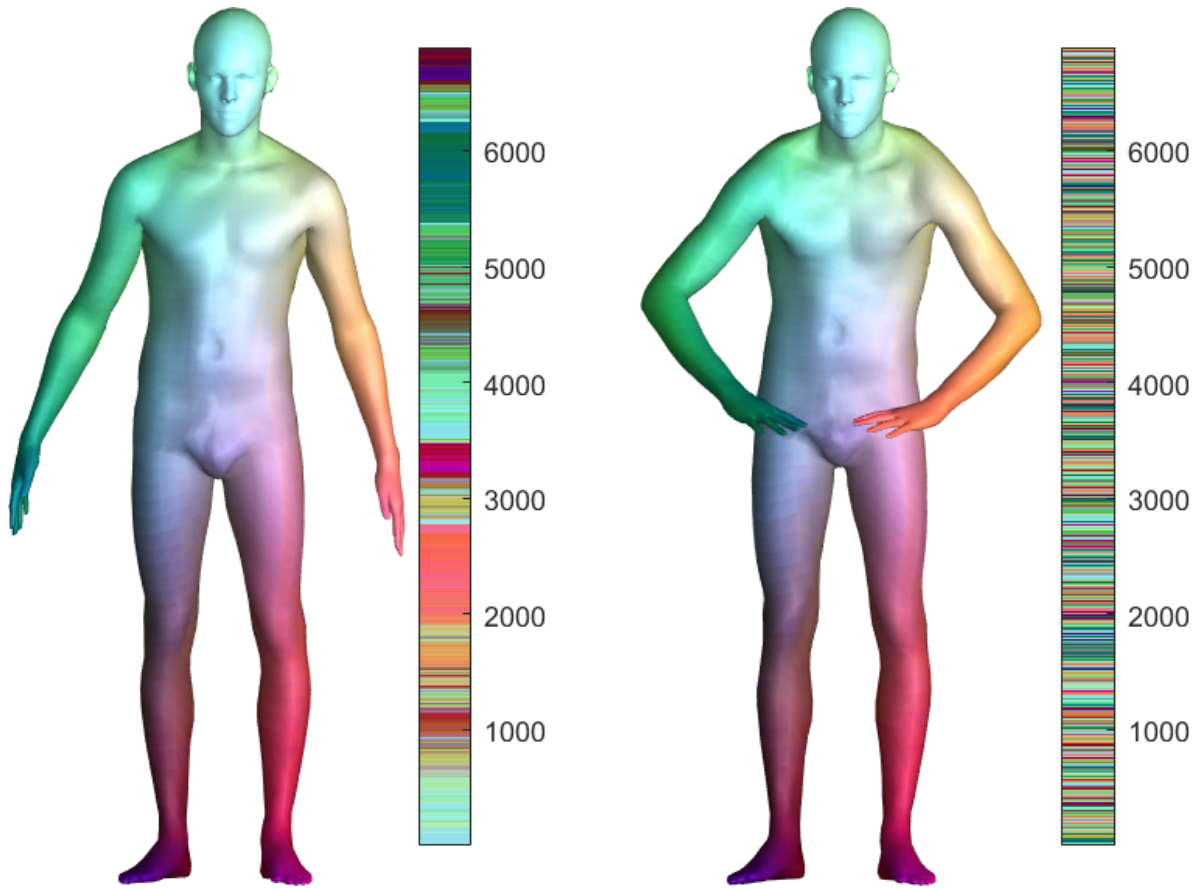


Warning: New versions of Matlab require shading flat (not interp)

Visualizing correspondence: Dense colors

Case 1: One match for each source point, surjective (one-to-one map)

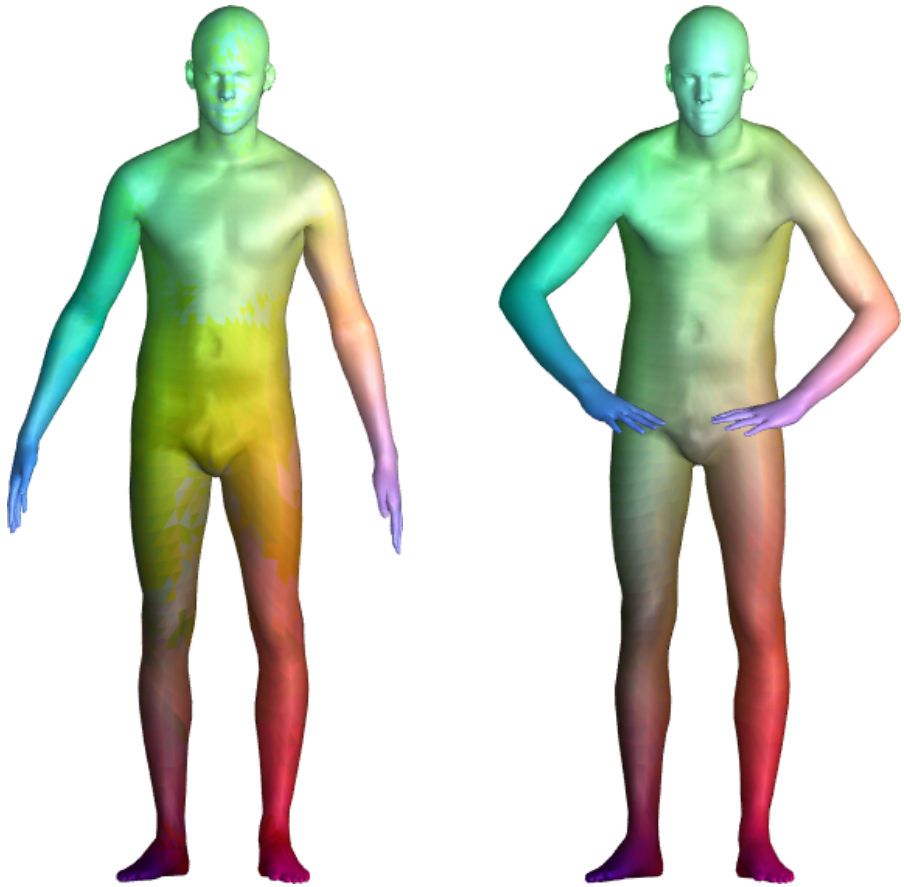
Obtain the target colormap by permuting the rows of the source colormap



Visualizing correspondence: Dense colors

Case 2: One match for each source point, not surjective

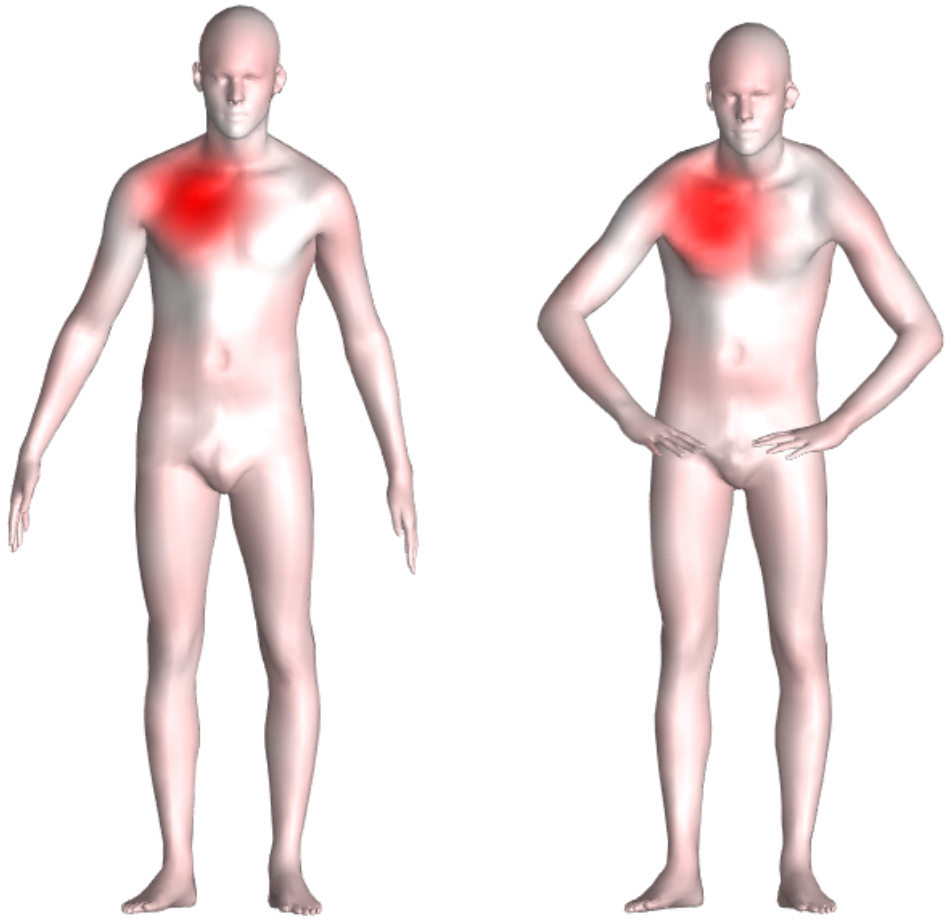
Obtain the source colormap by pulling back the target colormap



This hides the lack of surjectivity in the correspondence

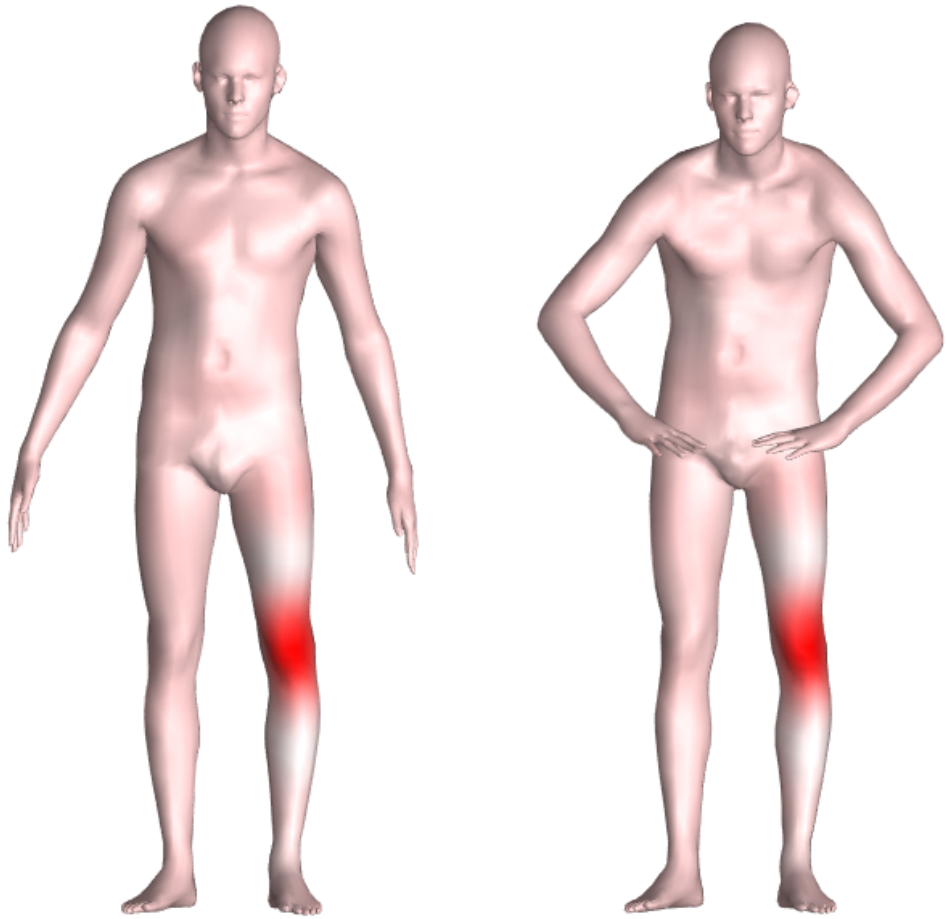
Visualizing correspondence: Function mapping

The correspondence can be used to map functions to functions



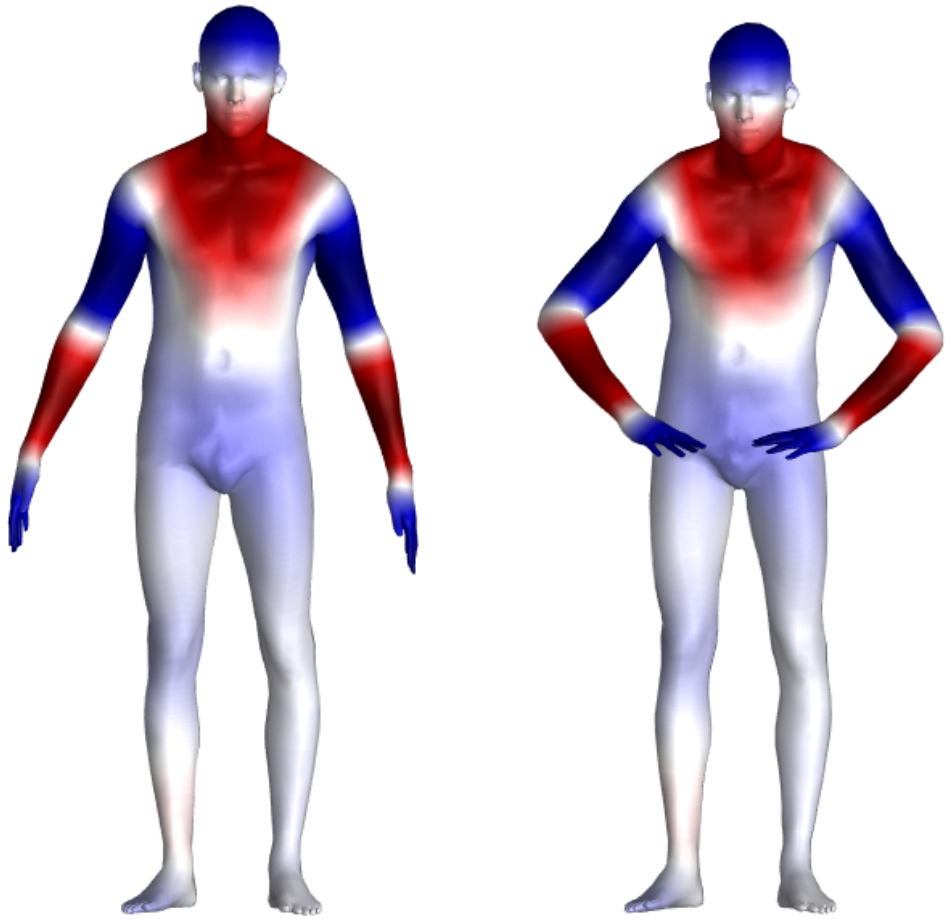
Visualizing correspondence: Function mapping

The correspondence can be used to map functions to functions



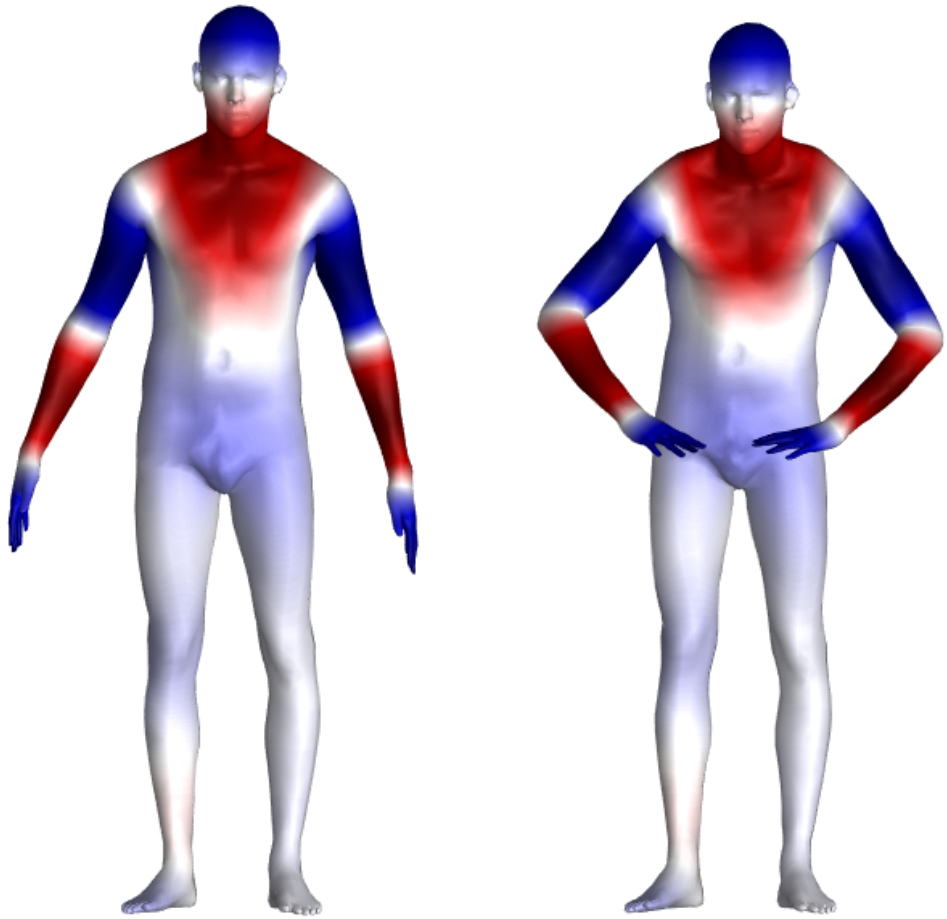
Visualizing correspondence: Function mapping

The correspondence can be used to map functions to functions



Visualizing correspondence: Function mapping

The correspondence can be used to map functions to functions



This also works for non-surjective and in general soft maps

Pov-RAY

Exercise: Rendering of 1d Euclidean embedding

Replicate in Pov-RAY the rendering from the last exercise of the Oct 04 lecture

- You must use the bluewhitered colormap
- Remember to extend the solution from FPS to entire mesh using the Voronoi regions

Use materials and lights as you like.

Send your renderings in .png format to rodola@di.uniroma1.it, using [FundCG] as the email subject.