Non-Rigid Alignment

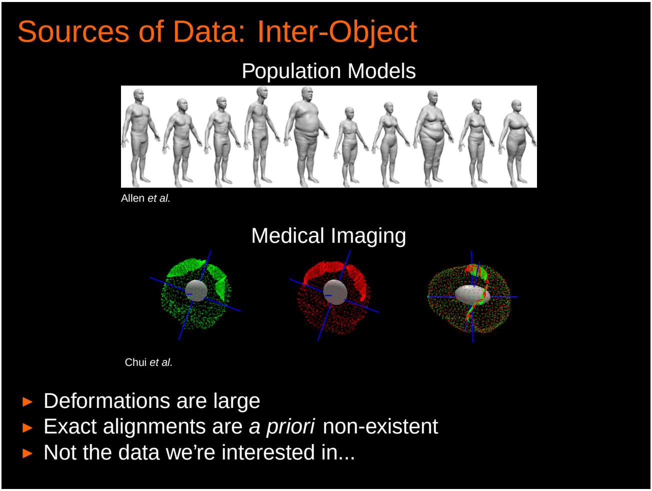
3D Scan Matching and Registration, Part III
ICCV 2005 Short Course

Benedict J. Brown

Princeton University

Outline

- Non-rigid error
- Thin-plate splines
- Alignment Algorithms
    - Softassign/deterministic annealing
    - Hierarchical ICP
    - Piecewise ICP
    - Global non-rigid alignment



Sources of Data: Inter-Object

Population Models

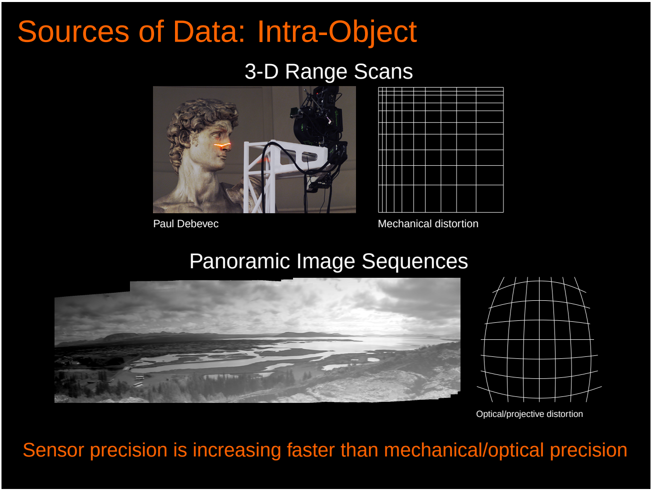
Allen *et al.*

Medical Imaging

Chui *et al.*

- Deformations are large
- Exact alignments are *a priori* non-existent
- Not the data we're interested in...

Aligning objects of the same class, such as bodies or organs to each other or to a template is a very common application. But it differs from range scan alignment in that there is only a somewhat loose relationship between the different objects, so only certain landmarks or structures should be aligned. In model acquisition, however, we wish to align overlapping scans of the same surface. Deformations are typically small because they come from acquisition error, but we want exact alignments everywhere. So we won't address the data on this slide any more in this talk.



Sources of Data: Intra-Object

3-D Range Scans

Paul Debevec

Mechanical distortion

Panoramic Image Sequences

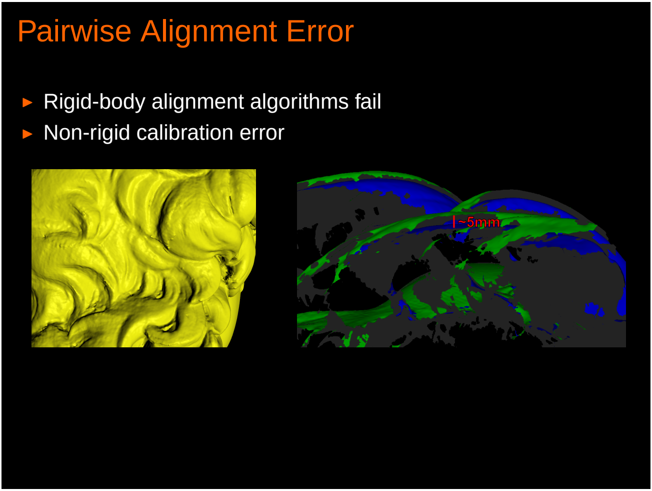
Optical/projective distortion

Sensor precision is increasing faster than mechanical/optical precision

In scan (and image) alignment, warp typically arises from calibration error. In some cases, such as the scanner used for the David, this is chiefly mechanical. For example, to keep the warp within the required tolerances for .25 mm accuracy on this scanner, lead screws needed to be calibrated to within a few microns. As this was not achievable, the spacing between samples diverges non-linearly from the expected values. The result is a slight curl in all the range scans.

Image sequences often exhibit substantial optical and perspective distortion. While optical distortion and vignetting can be compensated for in a carefully calibrated system, doing so is extremely difficult with consumer equipment (such as the DV camera used for this sequence), being used by consumers, in uncontrolled environments. Some additional sources of non-linear error are autofocus, auto white-balance, and parallax.

We believe that the ability to calibrate cameras and scanners well will get harder, not easier. A major reason for this is that sensor resolution is increasing rapidly, leading to lower calibration tolerances. But the precision of motors and optics is not increasing nearly as quickly. This is especially true in consumer-grade equipment, because of the high cost of manufacturing and assembling precision components.



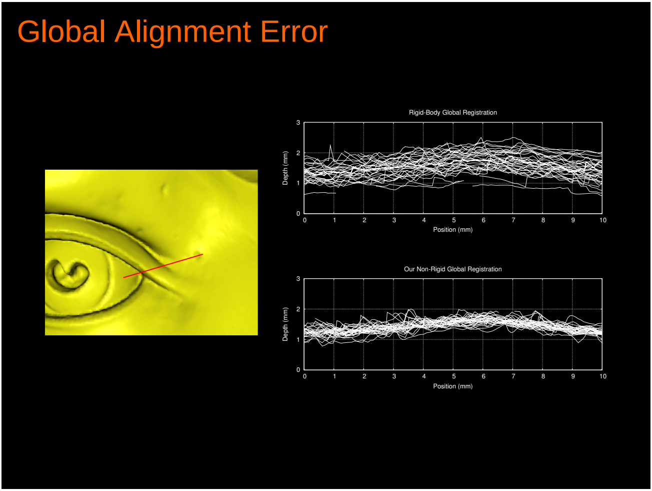
Pairwise Alignment Error

- Rigid-body alignment algorithms fail
- Non-rigid calibration error

Previous range scan alignment algorithms have assumed a rigid-body transform. But even the best rigid-body alignment fail when the data is warped.

The Problem in the Digital Michelangelo data is one of calibration. The David is about 7.5m high, including the pedestal, and the desired scan accuracy is .25mm, leading to particular stringent calibration requirements. Furthermore, the scanner had to be highly reconfigurable to be able to capture the entire surface (without ever running the risk of touching it). Meeting the calibration requirements under these conditions was simply not possible.

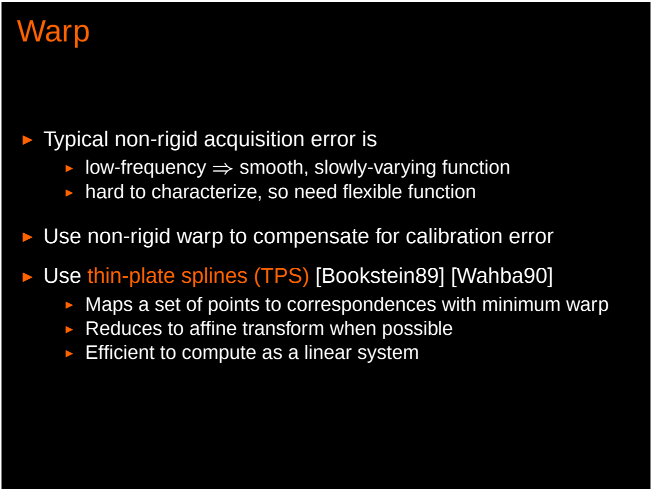
Other types of scanners run into other problems. Photometric stereo scanners estimate normals, and integrate the normal field to obtain the surface. In this case a bias in the normal estimations will cause a small warp to each range scan. Images will suffer warp if the camera is not calibrated for lens parameters, vignetting, radial distortion, chromatic aberration, *etc.* Even then, autofocus, auto white balance, parallax, and scene movement will cause non-rigid image distortions. All of these parameters (in 2-D and 3-D) are of course compounded when manufacturing cheap, consumer-oriented devices.



Global Alignment Error

Global alignment magnifies all pairwise alignment problems, because you're finding a consensus alignment for everything. The upper graph gives the relative displacements of all the rigid-body-aligned range scans overlapping the red line on David's left eye. Note that there is about 1mm misalignment normal to the surface. On many areas of the model, particular in the hair, we have observed much greater misalignments, even exceeding 15mm.

The lower graph shows the relative displacements of the non-rigid-aligned scans. Most range scans are now aligned to within less than .5mm of each other. This results in a sharper, more detailed final model (shown at left).



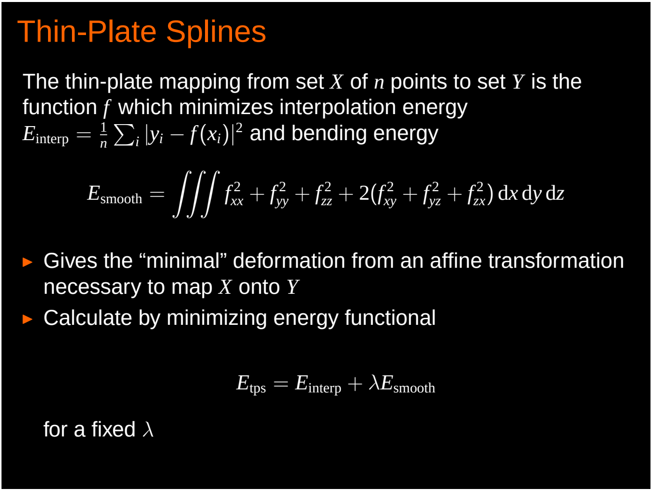
Warp

- Typical non-rigid acquisition error is
  - low-frequency ⇒ smooth, slowly-varying function
  - hard to characterize, so need flexible function
- Use non-rigid warp to compensate for calibration error
- Use thin-plate splines (TPS) [Bookstein89] [Wahba90]
  - Maps a set of points to correspondences with minimum warp
  - Reduces to affine transform when possible
  - Efficient to compute as a linear system

Our range scans have thousands to millions of points, and the spacing between points is about the same as the desired alignment quality. The low frequency assumption is key to maintaining a manageable system, and to prevent overfitting to a bad alignment.

Outline

- Non-rigid error
- Thin-plate splines
- Alignment Algorithms
    - Softassign/deterministic annealing
    - Hierarchical ICP
    - Piecewise ICP
    - Global non-rigid alignment



Thin-Plate Splines

The thin-plate mapping from set X of n points to set Y is the function f which minimizes interpolation energy $E_{\text{interp}} = \frac{1}{n}\sum_i |y_i - f(x_i)|^2$ and bending energy

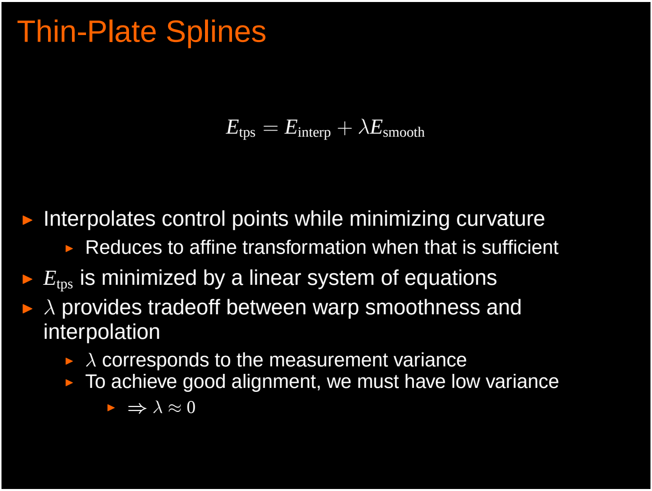
$$E_{\text{smooth}} = \iiint f_{xx}^2 + f_{yy}^2 + f_{zz}^2 + 2(f_{xy}^2 + f_{yz}^2 + f_{zx}^2)\,\mathrm{d}x\,\mathrm{d}y\,\mathrm{d}z$$

- Gives the “minimal” deformation from an affine transformation necessary to map X onto Y
- Calculate by minimizing energy functional

$$E_{\text{tps}} = E_{\text{interp}} + \lambda E_{\text{smooth}}$$

for a fixed λ

Thin-plate splines were first introduced by [Duchon77], who proved the form their form, existence, and unicity. They were later picked up by [Bookstein89] and [Wahba90], the latter of which contains a version of the proof in English. [Ruhr96, Ruhr03] has extended them in various ways, such as constraining normal directions, which is useful in some alignment applications with sparse features.

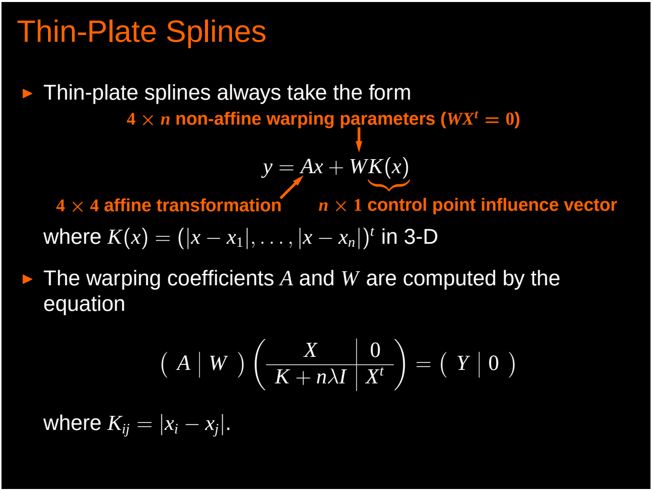


Thin-Plate Splines

$$E_{\text{tps}} = E_{\text{interp}} + \lambda E_{\text{smooth}}$$

- Interpolates control points while minimizing curvature
  - Reduces to affine transformation when that is sufficient
- E_{tps} is minimized by a linear system of equations
- λ provides tradeoff between warp smoothness and interpolation
  - λ corresponds to the measurement variance
  - To achieve good alignment, we must have low variance
    - $\Rightarrow \lambda \approx 0$

The first two properties are what we really like about the thin-plate spline. We get a rigid-body alignment when we can, and it's not too onerous to compute. The system is pretty unstable if $\lambda = 0$, so we use $\lambda = 0.00001$.



Thin-Plate Splines

- Thin-plate splines always take the form

$$y = \underset{\textbf{4 × 4 affine transformation}}{A}x + \overset{\textbf{4 × } n \textbf{ non-affine warping parameters } (WX^t = 0)}{W}\underbrace{K(x)}_{n \times 1 \textbf{ control point influence vector}}$$

  where $K(x) = (|x - x_1|, \ldots, |x - x_n|)^t$ in 3-D

- The warping coefficients A and W are computed by the equation

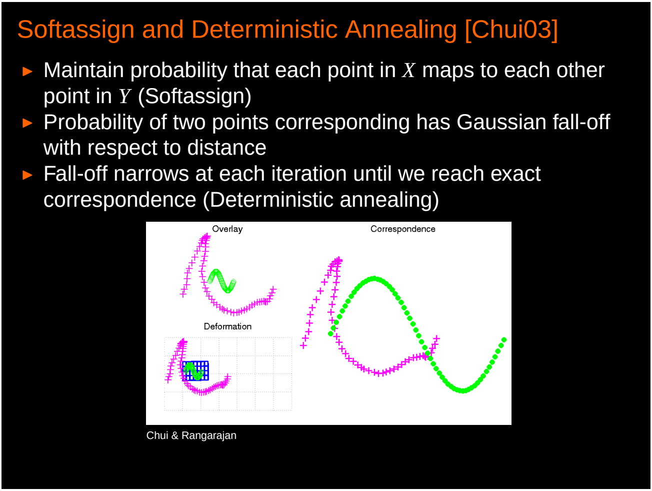
$$\left(\begin{array}{c|c} A & W \end{array} \right) \left(\begin{array}{c|c} X & 0 \\ \hline K + n\lambda I & X^t \end{array} \right) = \left(\begin{array}{c|c} Y & 0 \end{array} \right)$$

  where $K_{ij} = |x_i - x_j|$.

Computing a thin-plate spline with n control points requires inverting an $n \times n$ matrix; applying it requires multiplying an $n \times 4$ matrix with each point for a total of $4n$ scalar multiplications per scan point. To get good alignments, we use a few hundred to a thousand control points per range scan, which is fine if we only need to compute one warp per scan.

Outline

- Non-rigid error
- Thin-plate splines
- Alignment Algorithms
    - Softassign/deterministic annealing
    - Hierarchical ICP
    - Piecewise ICP
    - Global non-rigid alignment

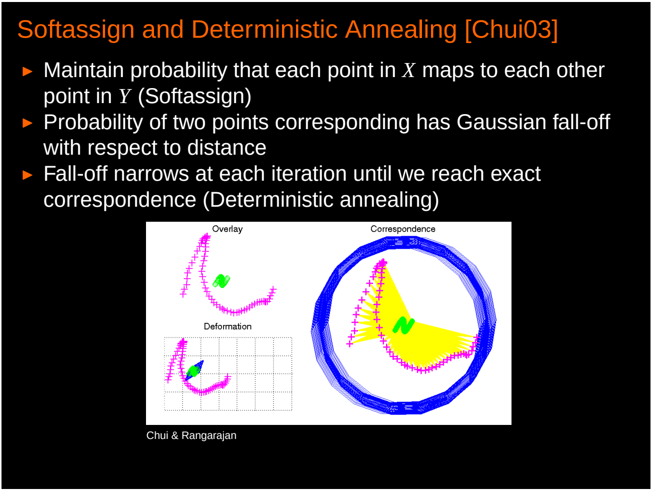


Softassign and Deterministic Annealing [Chui03]

- Maintain probability that each point in X maps to each other point in Y (Softassign)
- Probability of two points corresponding has Gaussian fall-off with respect to distance
- Fall-off narrows at each iteration until we reach exact correspondence (Deterministic annealing)

Chui & Rangarajan

For full details refer to [Chui03]. Code is available at http://www.cise.ufl.edu/ãnand/publications.html. The course bibliography contains additional citations of related work, including more recent work which guarantees a diffeomorphic warp. The advantage of using a diffeomorphism is that it guarantees a surface will not fold back on itself, or do other horrible things. For range scans, this turns out to be unnecessary because the warp is so small relative to the surface.



Softassign and Deterministic Annealing [Chui03]

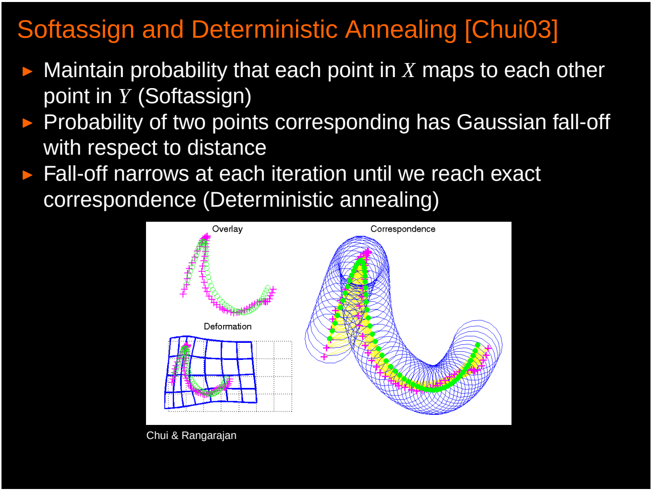
- Maintain probability that each point in X maps to each other point in Y (Softassign)
- Probability of two points corresponding has Gaussian fall-off with respect to distance
- Fall-off narrows at each iteration until we reach exact correspondence (Deterministic annealing)

Overlay
Correspondence
Deformation

Chui & Rangarajan

For full details refer to [Chui03]. Code is available at http://www.cise.ufl.edu/ãnand/publications.html. The course bibliography contains additional citations of related work, including more recent work which guarantees a diffeomorphic warp. The advantage of using a diffeomorphism is that it guarantees a surface will not fold back on itself, or do other horrible things. For range scans, this turns out to be unnecessary because the warp is so small relative to the surface.

First iteration of deterministic annealing. The blue circles shows the standard deviation of the Gaussian falloff used at this iteration. Because the initial radius is very large, all source points are nearly equally likely to match any target, so they are mapped nearly to the target points' center of mass.



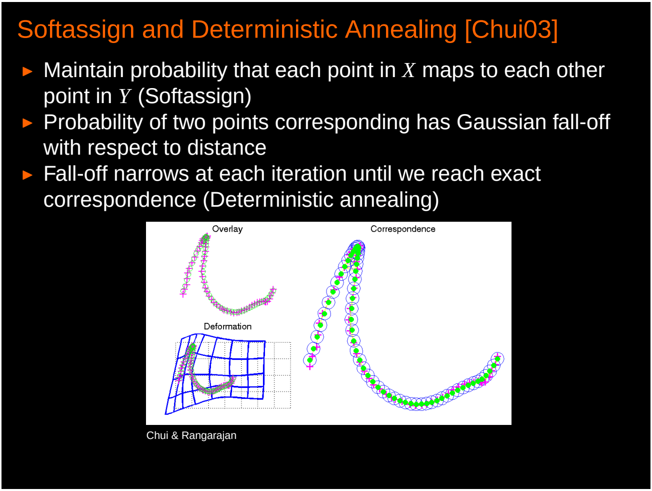
Softassign and Deterministic Annealing [Chui03]

- Maintain probability that each point in X maps to each other point in Y (Softassign)
- Probability of two points corresponding has Gaussian fall-off with respect to distance
- Fall-off narrows at each iteration until we reach exact correspondence (Deterministic annealing)

Chui & Rangarajan

For full details refer to [Chui03]. Code is available at http://www.cise.ufl.edu/ãnand/publications.html. The course bibliography contains additional citations of related work, including more recent work which guarantees a diffeomorphic warp. The advantage of using a diffeomorphism is that it guarantees a surface will not fold back on itself, or do other horrible things. For range scans, this turns out to be unnecessary because the warp is so small relative to the surface.

After 15 iterations, the source points are starting to converge to their final positions. They are now far more likely to match nearby target points than far off ones.



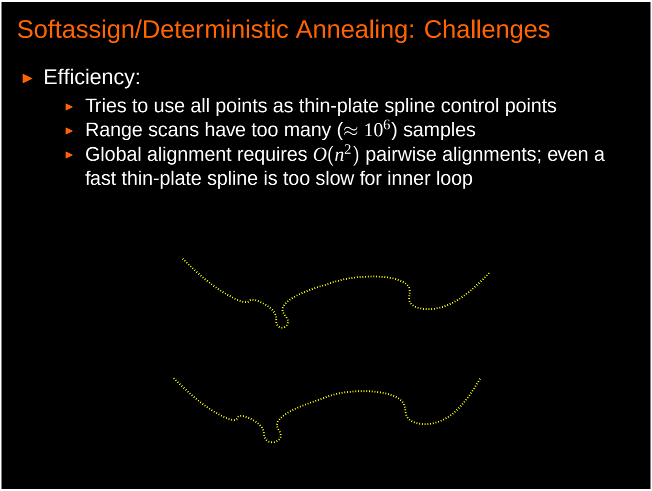
Softassign and Deterministic Annealing [Chui03]

- Maintain probability that each point in X maps to each other point in Y (Softassign)
- Probability of two points corresponding has Gaussian fall-off with respect to distance
- Fall-off narrows at each iteration until we reach exact correspondence (Deterministic annealing)

Chui & Rangarajan

For full details refer to [Chui03]. Code is available at http://www.cise.ufl.edu/ãnand/publications.html. The course bibliography contains additional citations of related work, including more recent work which guarantees a diffeomorphic warp. The advantage of using a diffeomorphism is that it guarantees a surface will not fold back on itself, or do other horrible things. For range scans, this turns out to be unnecessary because the warp is so small relative to the surface.

After 29 iterations, the matching is nearly perfect, and correspondences and outliers can be determined.

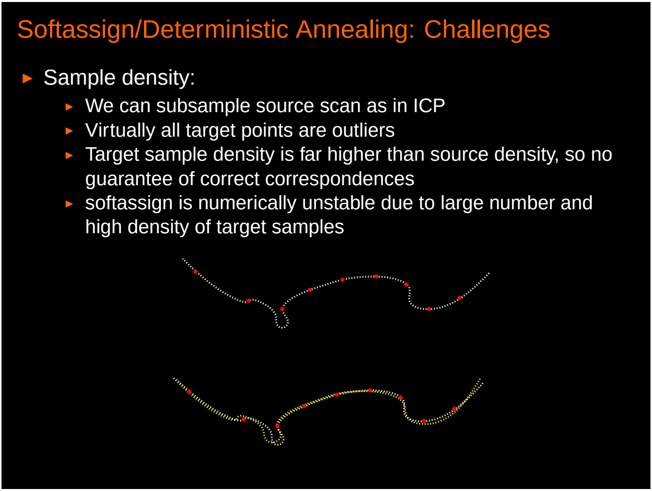


Softassign/Deterministic Annealing: Challenges

- Efficiency:
    - Tries to use all points as thin-plate spline control points
    - Range scans have too many ($\approx 10^6$) samples
    - Global alignment requires $O(n^2)$ pairwise alignments; even a fast thin-plate spline is too slow for inner loop

The figure on this slide is slightly misleading, because it's only a curve, not a surface. Computing a thin-plate spline with n control points requires inverting an $n \times n$ matrix, which is $O(n^2)$ at best, and $O(n^3)$ at worst. Remember that the numerical stability of matrix inversion degrades as the matrix increases in size as well. This puts a practical limit of about 1000 control points or so, and even this will be inordinately slow.

For the range scans in David's head, it takes us about 9 seconds per range scan on average to compute and apply a thin-plate spline(typically using a few hundred control points). That means, as a very rough estimate, that a pairwise alignment would take at least a minute. Remember, global rigid-body alignment has ICP in an $O(n^2)$ inner loop, and David's head has 1400 range scans. Even considering that not all range scans overlap, that would still mean about a year of computation.



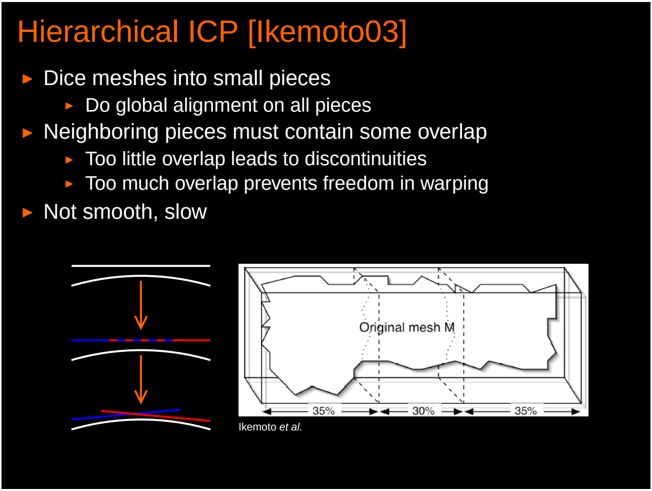
Softassign/Deterministic Annealing: Challenges

- Sample density:
  - We can subsample source scan as in ICP
  - Virtually all target points are outliers
  - Target sample density is far higher than source density, so no guarantee of correct correspondences
  - softassign is numerically unstable due to large number and high density of target samples

Subsampling is the obvious way of dealing with enormous data sets. Just as in ICP, you can imagine selecting only a few hundred points (or less) from the source data to keep the problem manageable. However that means, partial overlap aside, the number of target points exceeds the number of source points by several orders of magnitude, and the spacing between target points is no greater than the desired alignment accuracy. The matching is therefore guaranteed to be influenced far more by outliers than by correspondences.

In the figure, imaging using only the red points from the white scan, which can match anything on the yellow scan. A rigid-body transformation can provide an excellent alignment of these points to the yellow scan, but because the correspondences are poor, the remainder of the white scan (which is not counted in the error metric) is badly aligned. Although we have not seen alignments get worse using this approach as the figure might imply, we have routinely seen initial alignment fail to improve because the algorithm locks onto poor correspondences.

Finally, the amount of warp is only a few millimeters, whereas range scans are often tens of centimeters long. Chui & Rangarajan anneal λ (and a second λ_2 parameter) along with the width of the Gaussian. But we need these to be very nearly zero to obtain any meaningful warp at all at this scale. This in turn leads to overfitting bad correspondences and to numerical instability.

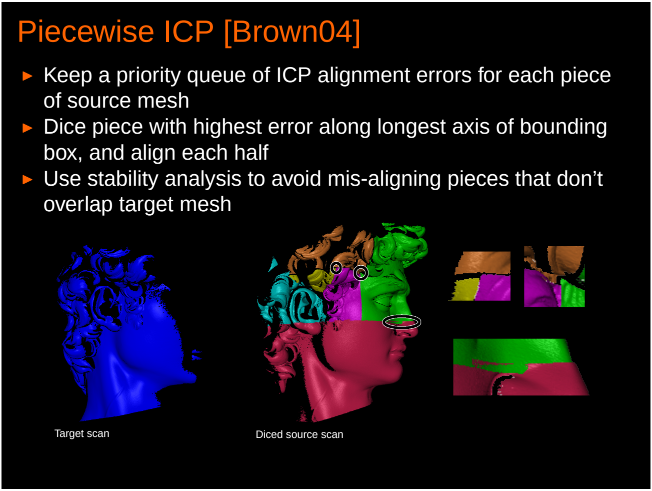


Hierarchical ICP [Ikemoto03]

- Dice meshes into small pieces
  - Do global alignment on all pieces
- Neighboring pieces must contain some overlap
  - Too little overlap leads to discontinuities
  - Too much overlap prevents freedom in warping
- Not smooth, slow

Ikemoto *et al.*

Hierarchical ICP approximates the low frequency warp as a piecewise-linear function. There are two main drawbacks—dicing increases the number of scans to align, which increases running time quadratically; also, overlapping areas of diced scans move some relative to each other, causing some undesirable smoothing of high frequency information.



Piecewise ICP [Brown04]

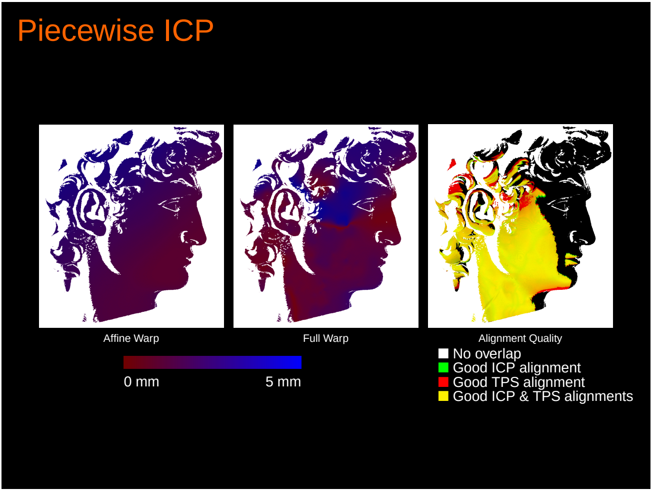
- Keep a priority queue of ICP alignment errors for each piece of source mesh
- Dice piece with highest error along longest axis of bounding box, and align each half
- Use stability analysis to avoid mis-aligning pieces that don't overlap target mesh

Target scan

Diced source scan

We like the thin-plate spline warp in [Chui03], especially for the preservation of high frequency information, but the piecewise linear approximation of [Ikemoto03] is faster and more robust. This is a way of combining the two.

The idea is to dice the source mesh into non-overlapping pieces, and perform ICP on each piece against the target mesh. This is very discontinuous at the piece boundaries, but yields good alignments everywhere else. So we can use the piecewise alignments to find good correspondences for a thin-plate spline warp.



Piecewise ICP

Affine Warp

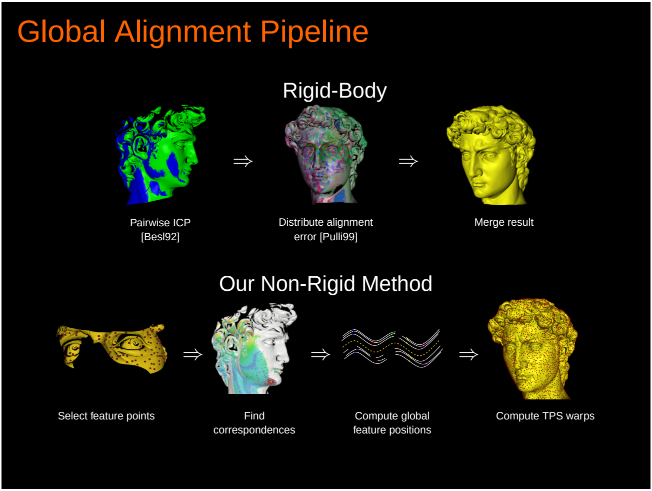
Full Warp

Alignment Quality

⬜ No overlap
🟩 Good ICP alignment
🟥 Good TPS alignment
🟨 Good ICP & TPS alignments

Here we visualize the warp associated with this alignment. The left figure shows the affine portion of the warp; the right figure shows the non-rigid portion. Blue indicates a larger warp, red a small one. The right figure shows the overall relative alignment qualities of thin-plate spline *vs.* ICP alignment. In this figure, red areas have good non-rigid alignment, but poor rigid alignment (chiefly the hair and the chin), green is the inverse, and yellow are areas where both alignments are good. (More specifically, the red color channel shows the non-rigid alignment quality, with brighter being better, and the green channel shows the ICP alignment quality; yellow is just the sum of the red and green channels.)

There are two principal drawbacks to this algorithm. The first is that it's pairwise, not global; the second is that although we obtain good alignments, we have observed much higher warp at the boundaries between pieces than elsewhere on the mesh (as is evident from the graph of non-rigid warp). That's an indication of poor correspondences near the boundaries which are resulting in unnecessary warp.



Global Alignment Pipeline

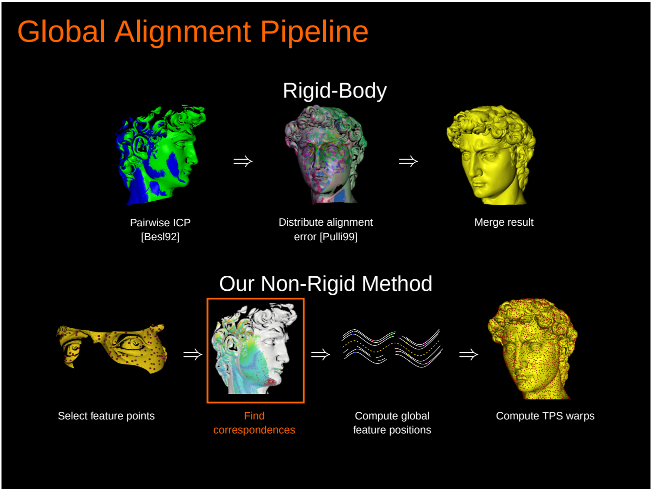
Rigid-Body

Pairwise ICP [Besl92] ⇒ Distribute alignment error [Pulli99] ⇒ Merge result

Our Non-Rigid Method

Select feature points ⇒ Find correspondences ⇒ Compute global feature positions ⇒ Compute TPS warps

We recall the global rigid-body alignment pipeline, and compare it to our non-rigid pipeline, whose steps we will describe in the following slides.



Global Alignment Pipeline

Rigid-Body

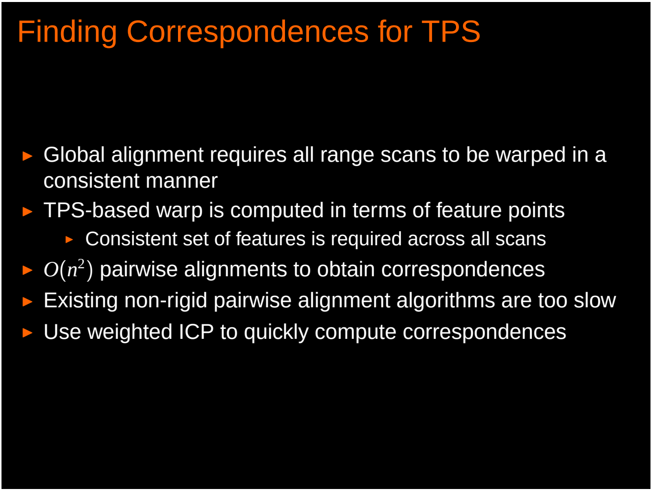
Pairwise ICP [Besl92] ⇒ Distribute alignment error [Pulli99] ⇒ Merge result

Our Non-Rigid Method

Select feature points ⇒ Find correspondences ⇒ Compute global feature positions ⇒ Compute TPS warps

The key to doing global non-rigid alignment is to have a consistent set of features across all scans. If we know the position of a feature point p on every range scan, we can compute warping transformations for each range scan that all map p to the same global position in space. To ensure we have enough features to control the warp, we select feature points from every range scan, then find their correspondences on every other scan. This is $O(n^2)$ in the number of scans, just like the first phase of global rigid-body alignment.

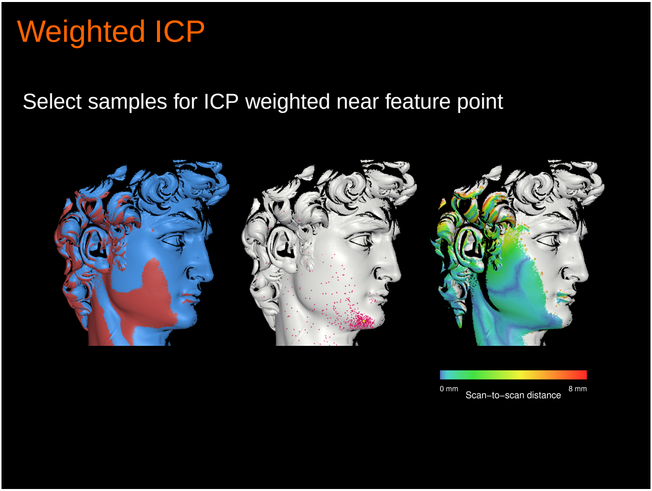
We select half our samples according to a uniform random distribution, and half according to stability sampling. This ensures that will get features for which we can reliably find correspondences, without starving smoother areas of the surface. The features are select at a constant sampling rate on each scan. This leads to more features than are strictly necessary in areas with many overlapping range scans. But it our experience too many features doesn't hurt you (in the worst case you can always throw some out later), whereas too few leads to a poor alignment.



Finding Correspondences for TPS

- Global alignment requires all range scans to be warped in a consistent manner
- TPS-based warp is computed in terms of feature points
    - Consistent set of features is required across all scans
- $O(n^2)$ pairwise alignments to obtain correspondences
- Existing non-rigid pairwise alignment algorithms are too slow
- Use weighted ICP to quickly compute correspondences

Finding feature correspondences isn’t all that different from doing all pairwise alignments. If the pairwise alignment is good, you can just read off closest points to get the feature correspondences. But we can speed things up by avoiding a full non-rigid alignment, which is important inside an $O(n^2)$ loop. Weighted ICP gives us that speedup.



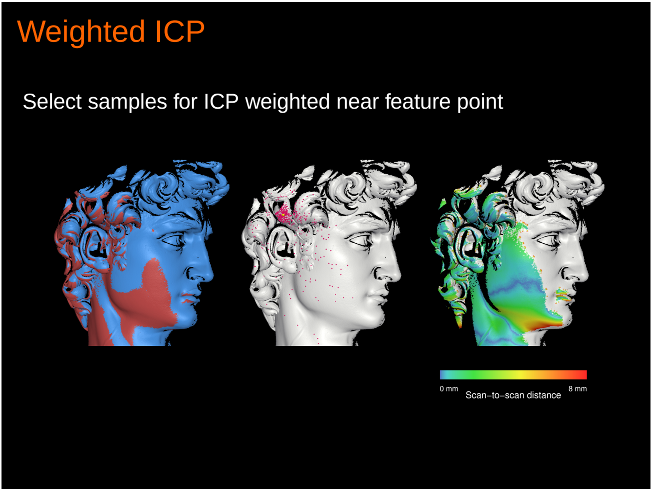
Weighted ICP

Select samples for ICP weighted near feature point

In the example shown here, we'd like to find a correspondence to the yellow point on the chin. So we do ICP using lots of samples clustered there. The result is a good alignment on the chin, but a lousy one in the hair (right image). That's ok, we'll just record the correspondence to the yellow point, and move on.

This isn't all that different from piecewise ICP. Conceptually, you're just giving yourself a different small piece centered around each feature. That's still very fast because each new alignment is pretty similar to the previous one so ICP converges in just a few iterations.

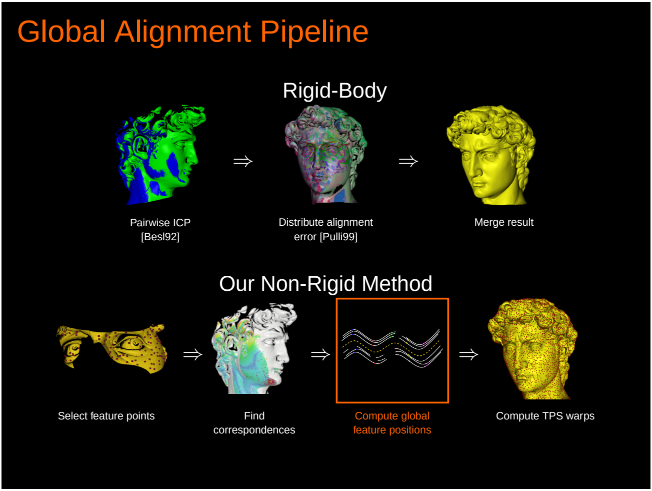
Stability sampling is very important to keep local alignments robust. In flat areas, where there's not much to constrain the rigid-body transformation, the stability part of the probability distribution selects points farther from the feature so as to obtain a stable alignment.



Weighted ICP

Select samples for ICP weighted near feature point

Here's another example, where we're looking for a correspondence in the hair. Now the alignment is good in the hair where we need the correspondence, but poor on the chin.



Global Alignment Pipeline

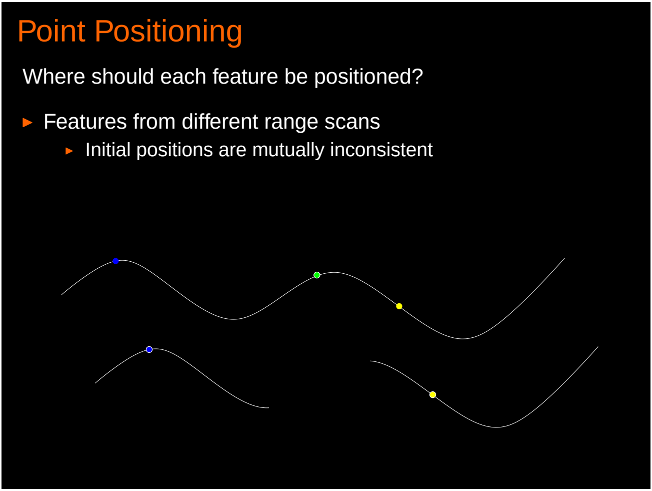
Rigid-Body

Pairwise ICP [Besl92] ⇒ Distribute alignment error [Pulli99] ⇒ Merge result

Our Non-Rigid Method

Select feature points ⇒ Find correspondences ⇒ Compute global feature positions ⇒ Compute TPS warps

Now that we have feature correspondences, we can compute a consistent global alignment. But we're dealing with a real object, which has an *a priori* correct shape. If we map all corresponding features to the wrong global position, we'll get a consistent alignment of the range scans, but we won't get the right shape.

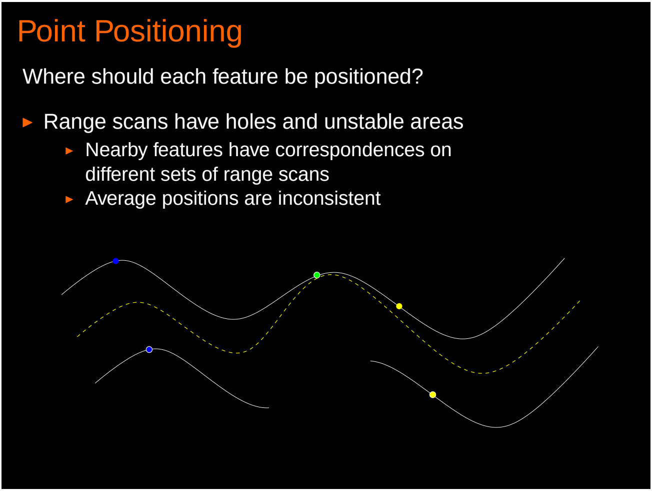


Point Positioning

Where should each feature be positioned?

- Features from different range scans
  - Initial positions are mutually inconsistent

The initial positions of the features from the scans on which we selected them are the easiest things to use. They're obviously wrong because the scans were not well aligned to start with, and lead to substantial pitting and dimpling of the final surface.

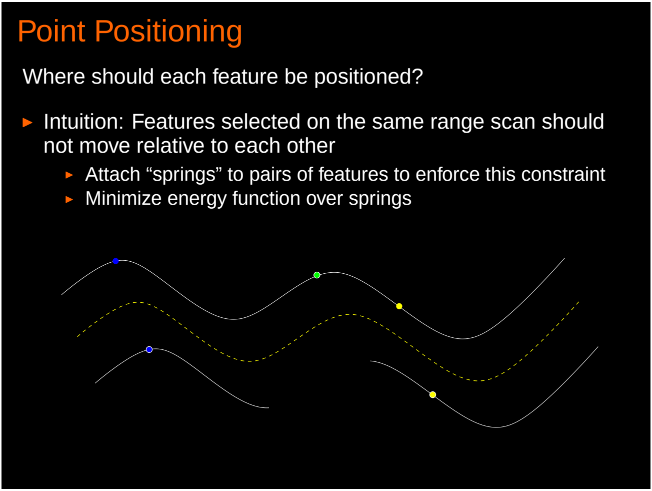


Point Positioning

Where should each feature be positioned?

- Range scans have holes and unstable areas
  - Nearby features have correspondences on different sets of range scans
  - Average positions are inconsistent

The average position over all features is much better, and we start our optimization process from these positions. But there are still inconsistencies due to gaps in range scans. We also don’t use correspondences if the weighted ICP was not stable since they probably aren’t good.



Point Positioning

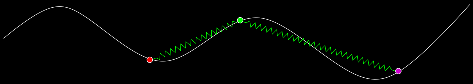
Where should each feature be positioned?

- Intuition: Features selected on the same range scan should not move relative to each other
  - Attach "springs" to pairs of features to enforce this constraint
  - Minimize energy function over springs

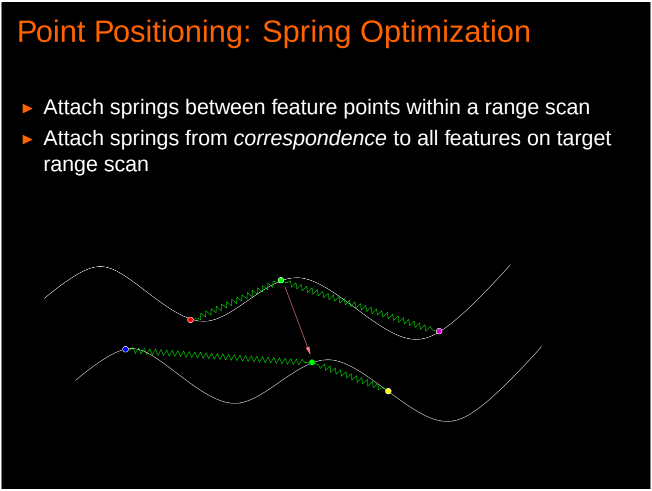
Here's what we want: something that interpolates the positions of all the scans while maintaining their shape.

Point Positioning: Spring Optimization

- Attach springs between feature points within a range scan



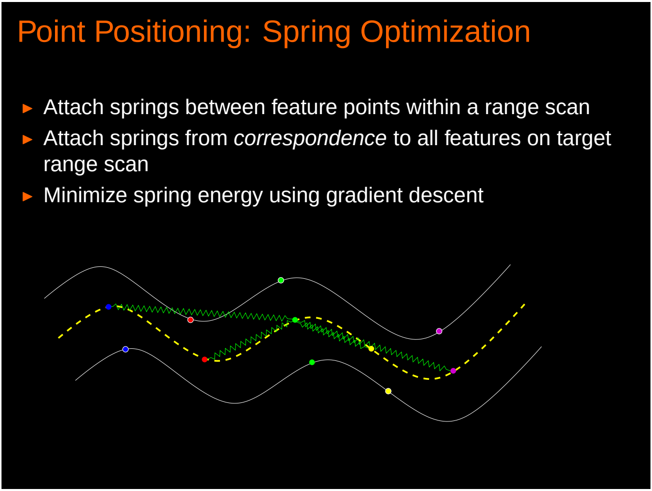
These springs ensure that no scan gets warped much. A rigid-body transformation won't affect the spring lengths at all. We weight short springs more than long springs because a low-frequency warp should be affecting the lengths of long springs more than short ones.



Point Positioning: Spring Optimization

- Attach springs between feature points within a range scan
- Attach springs from *correspondence* to all features on target range scan

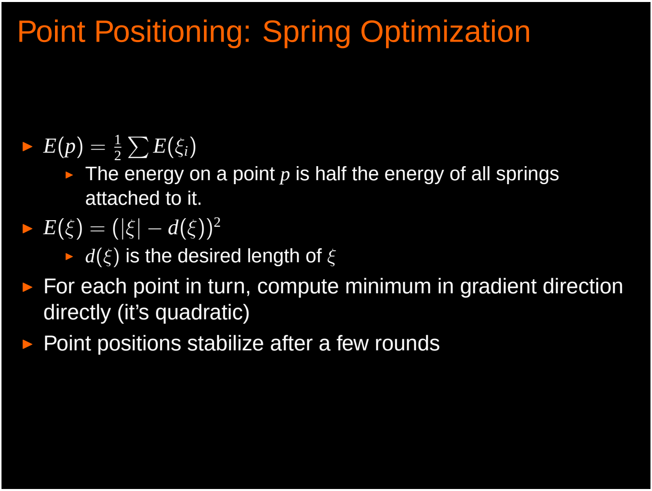
These springs ensure that scans align well to each other. Note than one scan can serve as a bridge between two others; if scans *A* and *C* both have stable correspondences to scan *B*, but not to each other, we can still get good alignments because the features which relate *A* and *C* to *B* are themselves linked by springs within *B*. This situation arises frequently in David’s hair, for example, where the range scans are small, noisy, and have many holes.



Point Positioning: Spring Optimization

- Attach springs between feature points within a range scan
- Attach springs from *correspondence* to all features on target range scan
- Minimize spring energy using gradient descent

Here's what happens when we minmize the spring energy. The energy on a spring ξ with rest length d is just $|\xi - d|^2$, and the total energy is the sum of all spring energies, which is quadratic. We just move each feature point in turn to the minimum along its gradient. The only drawback that we've observed is that memory requirements are quadratic in the number of features.

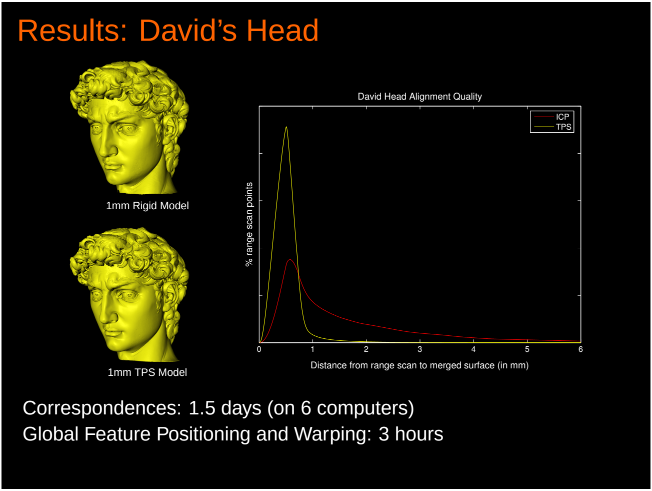


Point Positioning: Spring Optimization

- $E(p) = \frac{1}{2}\sum E(\xi_i)$
  - The energy on a point p is half the energy of all springs attached to it.
- $E(\xi) = (|\xi| - d(\xi))^2$
  - $d(\xi)$ is the desired length of ξ
- For each point in turn, compute minimum in gradient direction directly (it's quadratic)
- Point positions stabilize after a few rounds

We use a very simple iterated gradient descent algorithm. This has worked so well and so quickly in practice that we have not moved to a more principled method.

We did find that Newton's method often overshoots the minimum, and doesn't converge. But since the error function is quadratic, it's trivial to compute the minimum directly. The only detail is that the error function can be nearly flat, leading to instability. In that case we just don't move the point.



Results: David's Head

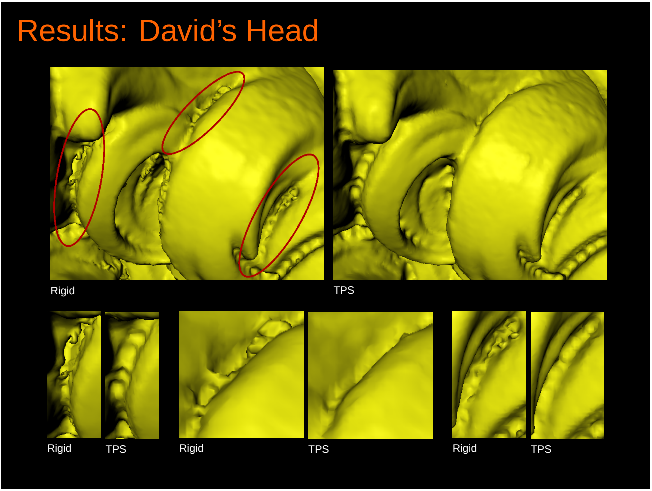
1mm Rigid Model

1mm TPS Model

Correspondences: 1.5 days (on 6 computers)

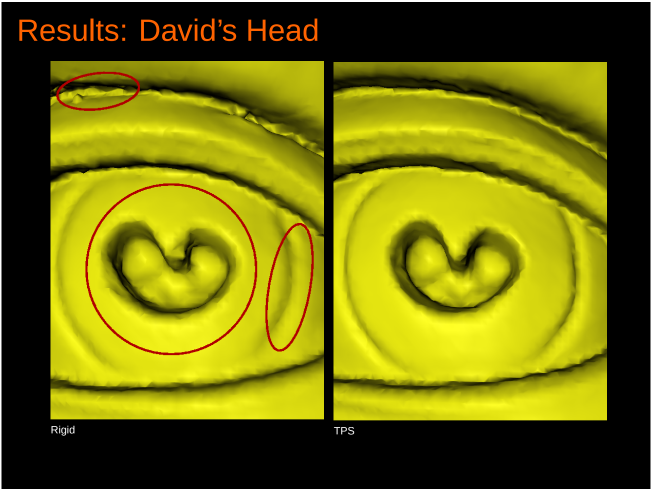
Global Feature Positioning and Warping: 3 hours

The graph is a histogram of alignment error, under rigid-body and non-rigid alignment. In each case, all scans are aligned global, and a final, merged is created. The the distance from every point on the aligned ranged scans to the merged surface is measured, and this is plotted as a histogram. While the peak alignment error in both cases is about .5mm, the long tail on the rigid-body histogram (in red) shows that many scans are poorly aligned. In the non-rigid case, the tail is largely eliminated. The difference in quality is immediately visible in the closeups on the following slides.



Results: David’s Head

This is a closeup of David’s hair above the left ear. The three circled areas (show along the bottom) exhibit artifacts and bridging between surfaces, which are eliminated in the non-rigid alignment. Also, especially in the first case, detailed drill marks on the surface are revealed in the improved alignment.



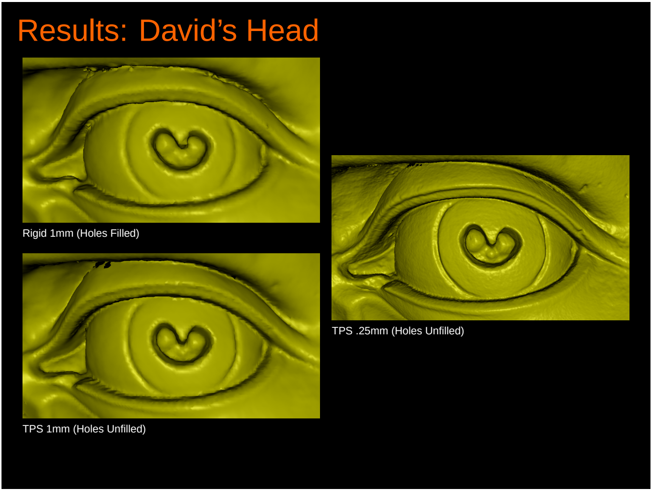
Results: David's Head

Rigid

TPS

This is David's left eye. The non-rigid model on the right has fewer artifacts and sharper details. The artifacts occur when the VRIP algorithm jumps between misaligned surfaces, particularly in deep grooves. The smoothing occurs due to misalignment along the surface

Of particular interest is the pupil on the eye. In the rigid-body model, it is skewed, and squared off. Comparisons to photographs and individual range scans confirm that the non-rigid version is in fact the correct shape.



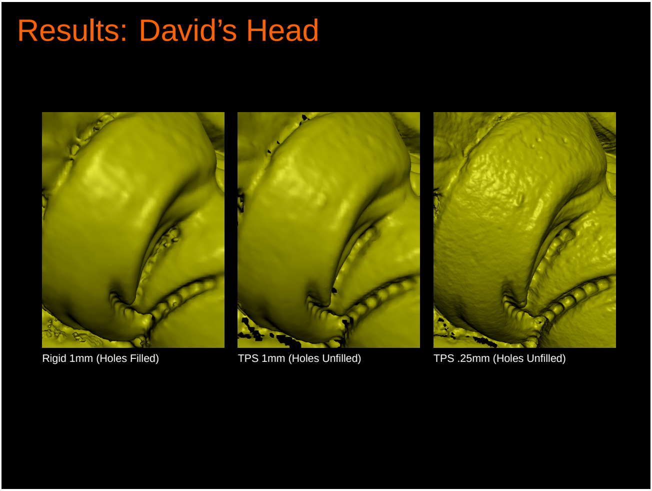
Results: David’s Head

Rigid 1mm (Holes Filled)

TPS 1mm (Holes Unfilled)

TPS .25mm (Holes Unfilled)

The Digital Michelangelo data was acquired at approximately .25mm resolution. Most of the models, including the David have only been released at 1mm resolution or less, simply because the alignments have not been good enough to produce meaningful .25mm results. We have now produced a .25mm model of David’s head, using the non-rigid alignments, and a closeup is shown hear alongside the 1mm models for comparison. Not only are all features much sharper than in the 1mm result, but surface texture at a finer scale than 1mm is revealed. Again, the input to VRIP is the same for the 1mm and .25mm non-rigid models—we have simply merged them at a higher resolution.



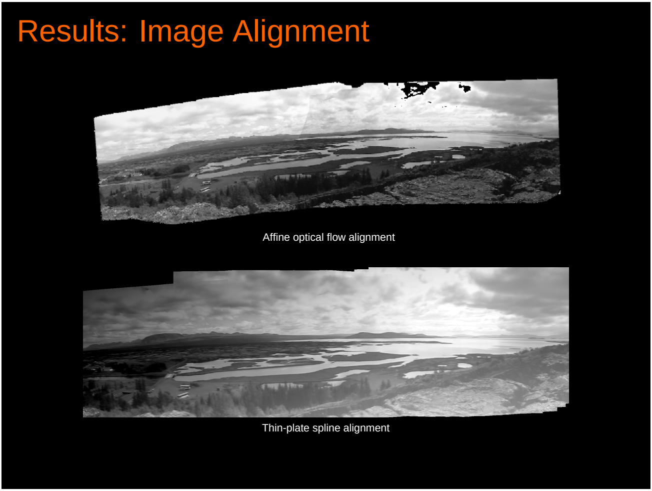
Results: David’s Head

Rigid 1mm (Holes Filled)

TPS 1mm (Holes Unfilled)

TPS .25mm (Holes Unfilled)

Here’s a closeup above David’s ear, again showing the increased sharpness and surface detailed revealed at .25mm.



Results: Image Alignment

Affine optical flow alignment

Thin-plate spline alignment

We are currently adapting our algorithm to work with image sequences. These panoramas are generated from footage taken with an uncalibrated, handheld consumer digital video camera (with auto focus and white balance). The top image, generated using affine optical flow shows substantial geometric distortion, although we did not generate it using state-of-the-art techniques. For the bottom panorama, we simply converted the images to height fields, and ran them through our range scan pipeline. Without modeling the perspective transform, camera parameters, camera motion, or parallax, the distortion is largely eliminated.

The upshot of this investigation is that the global feature positioning seems to work quite well. If there is significant zoom present, the images would need to be rescaled, but this would not be hard. On the other hand, weighted (or unweighted) ICP does not seem to be well-suited to aligning and finding feature correspondences in images. An image-based method for finding those correspondences will work much better, and likely faster.