

iBind: Smooth Indirect Binding using Segmented Thin-Layers

Chung-An Lin

Erick Miller

Gene S. Lee

Walt Disney Animation Studios

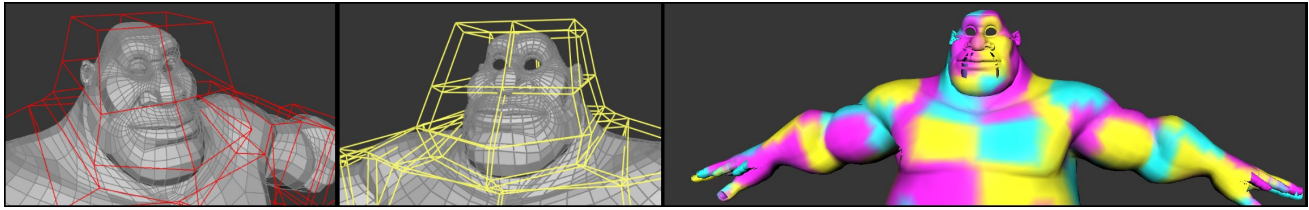


Figure 1: (a) User control cage (b) Segmented overlapping thin-layers

(c) Heat diffusion segment weights

This talk introduces iBind, a novel indirect binding technique improving on mean value coordinates (MVC)[1]. iBind smoothly deforms vertices using a control cage by uniquely leveraging heat diffusion on closed, thin layers across a structured set of mean value coordinates. The talk begins by discussing the limitations of previous techniques, and proceeds with an explanation of iBind, which produces fast, stable, and appealing spacial deformation.

Previous Techniques

Previous binding techniques have limitations for production users. Green coordinates [2] suffers from scaling artifacts, no control over localized regions, and being numerically unstable. Harmonic coordinates [3] encounters problems when binding points outside the control cage, and is inaccurate unless used with a dense solver (but the dense solver is slow). MVC is fast and appealing for low resolution cages, but has a large computation footprint as resolution of the control cage is increased to real-world examples. MVC also has limited local controls, and must perform a rebinding operation if the incoming vertex of the deformed mesh is moving. Finally, neither of the three algorithms generally allow arbitrary control cages with open-holes, and each requires a special case using barycentric coords if a vertex lies coplanar with a control cage face.

Segmented Thin-layers

iBind begins by building segmented, thin-layers from a user-created control cage (1a). A thin-layer is an extruded, mesh segment that expands one face of the control cage by its n surrounding faces. The vertices of the thin-layer are extruded along the normals of the control cage, and the layer is set to a variable thickness of value t . The results of this step creates a thin-layer, closed form, geometric structure, which serves as a cage for storing and evaluating a four dimensional MVC structure(1b).

Heat Diffusion Weights

Influence weights for each thin-layer to a vertex is achieved by performing surface-based, heat diffusion to the underlying mesh (1c). Given $\phi(\vec{r}, t)$ as the density of the thin segment at vertex \vec{r} and $D(\phi, \vec{r})$ as the diffusion coefficient, heat diffusion is computed as follows:

$$\frac{\partial \phi(\vec{r}, t)}{\partial t} = \nabla \cdot (D(\phi, \vec{r}) \nabla \phi(\vec{r}, t)),$$

Afterwards, the weights are normalized, and a (structured set of) MVC of each vertex is computed for each non-zero influencing thin-layer. These MVC values are then used with a static binding method to blend between overlapping thin-layers.

Static Binding vs. Dynamic Re-Binding

The static binding method of iBind performs a sampling of the MVC space of each layer to support dynamic vertices. This extension to MVC permits the vertices of the underlying mesh to freely change position without being dynamically rebound. In cg production, when articulating character skin, dynamic rebinding is undesirable since it often rebinds vertices to unintended locations,

such as lips being rebound to a nose during facial animation. Dynamic rebinding is still useful, so iBind implements it, but for most cases of character articulation, a static binding option is used.

Quadruple Structured Coordinates

The static binding method of iBind is stable and controllable due to its use of Quadruple Structured Coordinates (QSC). QSC are constructed using the original vertex and three additional coordinates sampled along each Cartesian axis. QSC are the mean value coordinates stored for all four samples, which are then applied to build a non-singular matrix for transforming, and ultimately deforming the dynamically changing vertices:

$$g[v] = \frac{\eta[v] - v}{\delta} \{ \hat{f}[v + \delta e_x] - \hat{f}[v], \hat{f}[v + \delta e_y] - \hat{f}[v], \hat{f}[v + \delta e_z] - \hat{f}[v] \}$$

Linear precision, smoothness and interpolation of this method relies on the stability and behaviors of the mean value interpolant.

Evaluation & Future Work

The use of segmented thin-layers with QSC overcomes the drawbacks of MVC. First, the local influence of the control cage can be defined explicitly by allowing variable expansion of the surrounding face segments in conjunction with heat diffusion settings. Second, control cages with open holes are handled elegantly and result in smooth deformations. Third, the memory footprint is dramatically reduced, and run-time performance scales well to real-world examples. And, lastly, it inherently resolves numeric instabilities found in co-planar conditions, without requiring special case barycentric coordinates. The thin-layer extrusion stage checks for and simply avoids such conditions.

iBind is efficient, stable, and practical. It adds predictability to geometric lattice deformations, and allows the incoming vertex position to change freely while staying statically bound. iBind's static binding is faster, uses less memory and is more effective than dynamic binding of MVC. Our method can be more than 30x faster than dynamic binding, due to the isolated segment approach.

iBind was implemented with OpenMP in C++, yielding nearly a 5x improvement in speed due to parallelization, and resulting in interactive deformations on real-world cg production examples. Ideas for extending this algorithm include the use of greater interpolation methods for blending multiple overlapping, weighted control cages, as well as a smooth subdivision surface scheme for evaluating the segmented thin-layers to an implicit limit surface.

References

- [1] Ju, T., Schaefer, S., and Warren, J. 2005 *Mean Value Coordinates for Closed Triangular Meshes*. In Siggraph '05, ACM Press, 561-566.
- [2] Lipman, Y., Levin D., and Cohen-Or, D. 2008 *Green Coordinates*, In Siggraph '08, ACM Press. 78:1-78:10
- [3] Joshi, P., et. al., 2007 *Harmonic Coordinates for Character Articulation*. In Siggraph '07, ACM Press, 71:1-71:9