Shading-Based Surface Editing

Yotam Gingold
Denis Zorin

New York University

Shading

[Michelangelo]

[Dürer]

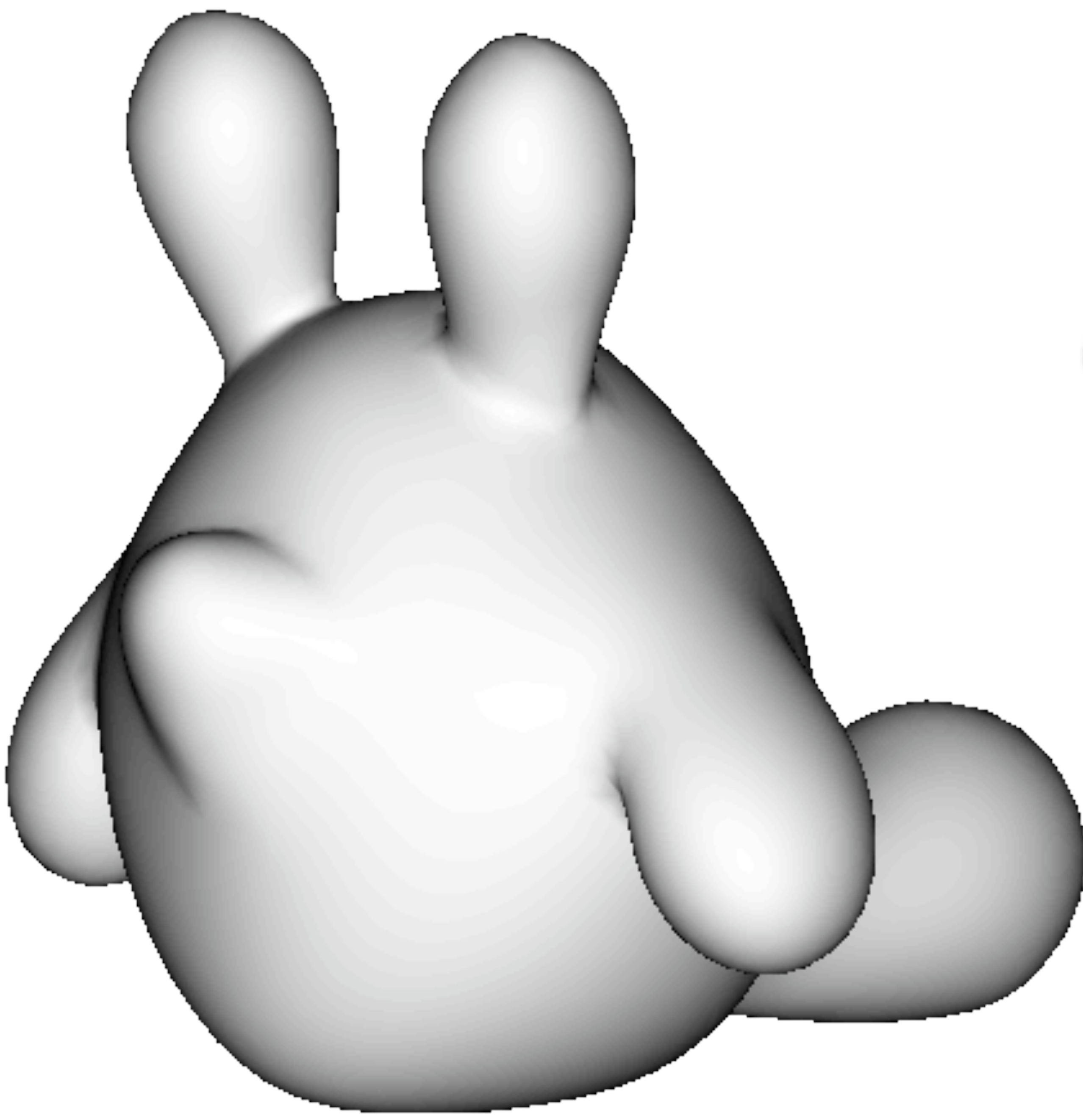
The subject of this presentation is using shading to modify 3D models.

Shading is an important part of our visual perception; everything in the real world appears shaded, and it plays a large role in how we interpret 3D geometry. Shading provides relief information to any view of a 3D model.

Specifically, shading indicates the amount of light that reaches a point on the surface and is reflected towards our eyes.

Artists learn to add shading to their drawings and can be quite accurate.

Shaded 3D Models



[Nealen et al. 2007]

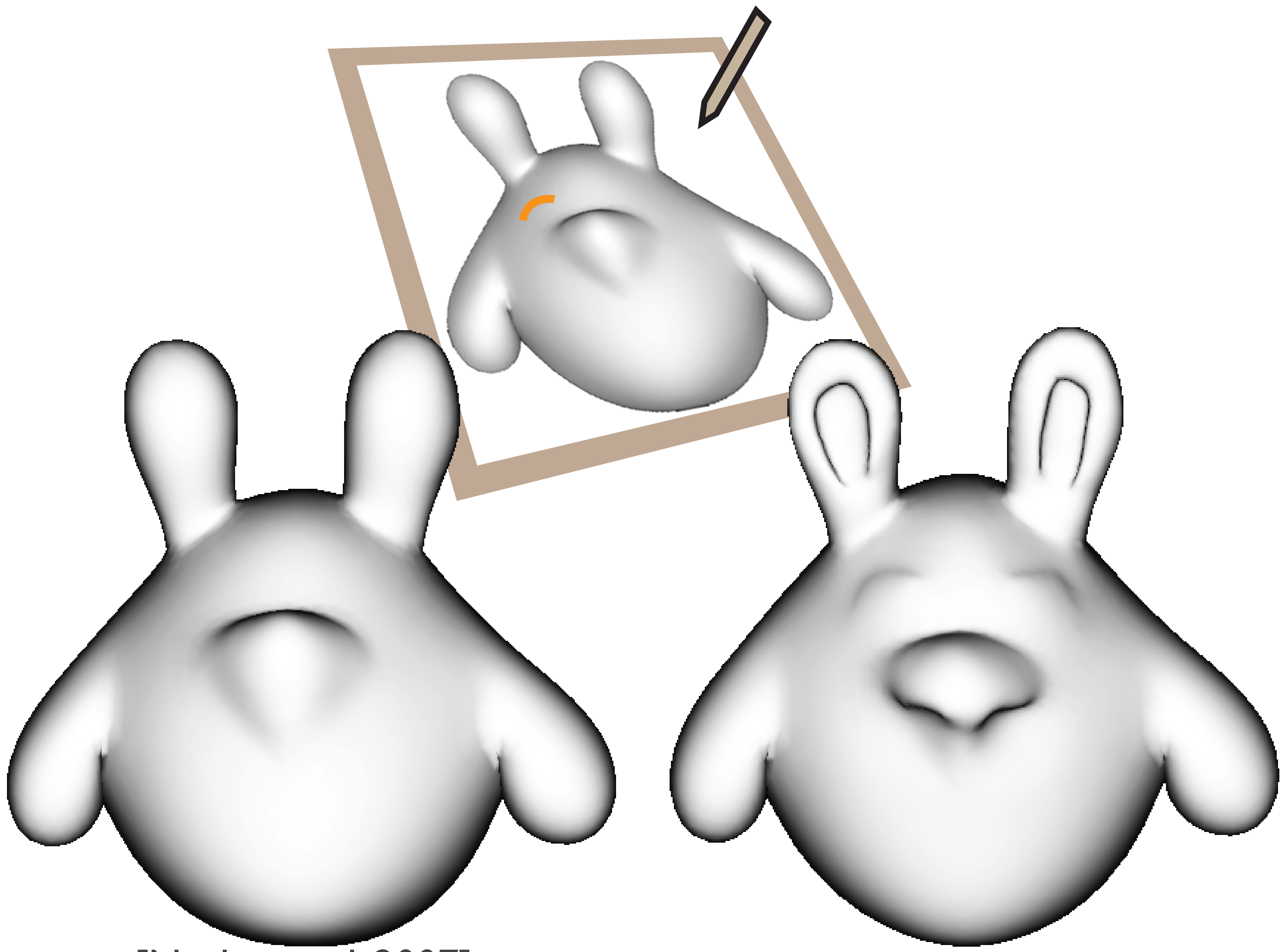
[Malanjo]

Here are computer generated images of 3D models.
We always see 3D geometry with shading, in the real world as well as on computers.

If you are a 2D artist familiar with shading, you might wish to modify 3D models by shading over 2D images of them.
Unfortunately, although we always see shaded models, there is no tool for 3D modeling by directly editing the shading.
In other words, you can't "draw what you want to see".

Our Approach

Obtain a new 3D model by shading over an existing one.



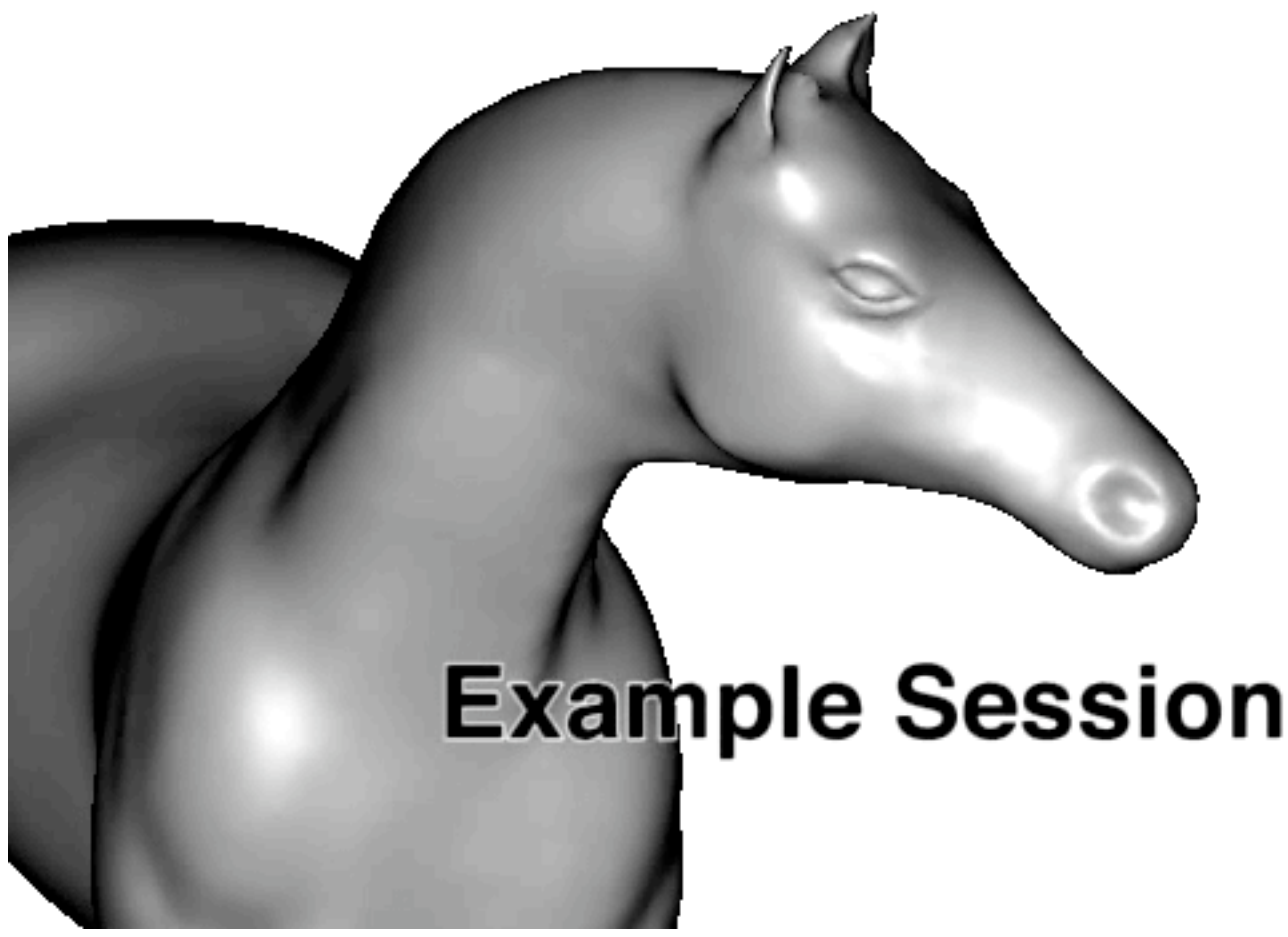
[Nealen et al. 2007]

Our system allows just this: users paint shading over the image of an existing model, and, after every stroke, our system computes a new model which has the desired appearance.

Here is a video of our system in use.

Our Approach

Obtain a new 3D model by shading over an existing one.



The user loads a coarse horse model, zooms into the eye, and proceeds to draw shading strokes. The resulting horse model contains a detailed eye.
Strokes are shown in yellow for purposes of illustration.

Contributions

An interactive tool for surface editing by "drawing what you see."

Our contribution is an interactive tool for surface editing by "drawing what you see."
- Our tool leverages artists' experience with shading,
- and has brush parameters similar to those found in 2d paint programs like Photoshop.
- To do this, we provide a stable, predictable, approximate solution for a special case of the Shape-from-Shading problem;
- our solution minimizes a quadratic energy, so it is fast and robust, and is not limited to a height field.

Contributions

An interactive tool for surface editing by “drawing what you see.”

- Leverages artists’ experience with shading

Our contribution is an interactive tool for surface editing by “drawing what you see.”
- Our tool leverages artists’ experience with shading,
- and has brush parameters similar to those found in 2d paint programs like Photoshop.
- To do this, we provide a stable, predictable, approximate solution for a special case of the Shape-from-Shading problem;
- our solution minimizes a quadratic energy, so it is fast and robust, and is not limited to a height field.

Contributions

An interactive tool for surface editing by “drawing what you see.”

- Leverages artists’ experience with shading
- Brush parameters similar to paint programs

Our contribution is an interactive tool for surface editing by “drawing what you see.”
- Our tool leverages artists’ experience with shading,
- and has brush parameters similar to those found in 2d paint programs like Photoshop.
- To do this, we provide a stable, predictable, approximate solution for a special case of the Shape-from-Shading problem;
- our solution minimizes a quadratic energy, so it is fast and robust, and is not limited to a height field.

Contributions

An interactive tool for surface editing by "drawing what you see."

- Leverages artists' experience with shading
- Brush parameters similar to paint programs
- Stable, predictable, approximate solution for a special case of Shape-from-Shading

Our contribution is an interactive tool for surface editing by "drawing what you see."
- Our tool leverages artists' experience with shading,
- and has brush parameters similar to those found in 2d paint programs like Photoshop.
- To do this, we provide a stable, predictable, approximate solution for a special case of the Shape-from-Shading problem;
- our solution minimizes a quadratic energy, so it is fast and robust, and is not limited to a height field.

Contributions

An interactive tool for surface editing by “drawing what you see.”

- Leverages artists’ experience with shading
- Brush parameters similar to paint programs
- Stable, predictable, approximate solution for a special case of Shape-from-Shading
- Quadratic energy, not limited to height field

Our contribution is an interactive tool for surface editing by “drawing what you see.”
- Our tool leverages artists’ experience with shading,
- and has brush parameters similar to those found in 2d paint programs like Photoshop.
- To do this, we provide a stable, predictable, approximate solution for a special case of the Shape-from-Shading problem;
- our solution minimizes a quadratic energy, so it is fast and robust, and is not limited to a height field.

Related Work

Relief Modeling

[van Overveld 1996]

[Zhang et al. 2001]

[Bourguignon et al. 2004]

[Zeng et al. 2005]

[Wu et al. 2007]

[Ng et al. 2007]

Shape-from-Shading

[Rushmeier et al. 2003]

[Prados 2004]

3D Sculpting

ZBrush [Pixologic]

Mudbox [Autodesk]

Sketch-based modeling

[Igarashi et al. 1999]

[Cheutet et al. 2004]

[Lawrence and Funkhouser 2004]

[Kara et al. 2006]

[Karpenko and Hughes 2006]

[Nealen et al. 2007]

Silhouette Editing

[Nealen et al. 2005]

[Zimmermann et al. 2007]

Shape-Preserving Deformation

[Sorkine et al. 2004]

[Yu et al. 2004]

There is a large body of related work.
There are a variety of surface editing tools, each complementary to ours. In particular, no relief editing tools allows users to simply "draw what they want to see."
-- click --
For the complementary task of silhouette editing, [Nealen et al. 2005] introduced one of the first techniques in this direction. Our machinery is similar to theirs, as well.
-- click --
3D sculpting tools, such as ZBrush and Mudbox, may be preferable for users with a sculpting background, although the interaction will be via a mouse or tablet instead of a physical tool in the real-world. However, note that these programs directly modify geometry, not shading. The class of edits which are easy to make with our tool are difficult to reproduce with a 3D sculpting tool, and vice versa. I will return to this difference later in the talk.
-- click --
Our approach can be considered an approximate, special case of Shape-from-Shading, where we use existing geometry to make incremental shading changes.

--- refuse ---
? [McCann and Pollard 2008] SIGGRAPH 2008 "Real-time Gradient-domain painting"

TODO:

show less, focus on the ones I do show.

mention explicitly that [nealen, zimmerman] work uses a similar machinery (laplacian energy) applied to moving silhouettes, while we apply it to changing gradients.

emphasize that we use existing geometry to solve this specific case of shape from shading. add after "related work".

emphasize difference with previous work, specifically sculpting.
- maybe show the comparison movie.
- maybe show a diagram: previous system creates bumps in or out, while our system changes the gradient and smoothes it out.

Related Work

Relief Modeling

[van Overveld 1996]

[Zhang et al. 2001]

[Bourguignon et al. 2004]

[Zeng et al. 2005]

[Wu et al. 2007]

[Ng et al. 2007]

Shape-from-Shading

[Rushmeier et al. 2003]

[Prados 2004]

3D Sculpting

ZBrush [Pixologic]

Mudbox [Autodesk]

Sketch-based modeling

[Igarashi et al. 1999]

[Cheutet et al. 2004]

[Lawrence and Funkhouser 2004]

[Kara et al. 2006]

[Karpenko and Hughes 2006]

[Nealen et al. 2007]

Silhouette Editing

[Nealen et al. 2005]

[Zimmermann et al. 2007]

Shape-Preserving Deformation

[Sorkine et al. 2004]

[Yu et al. 2004]

There is a large body of related work.
There are a variety of surface editing tools, each complementary to ours. In particular, no relief editing tools allows users to simply "draw what they want to see."
-- click --
For the complementary task of silhouette editing, [Nealen et al. 2005] introduced one of the first techniques in this direction. Our machinery is similar to theirs, as well.
-- click --
3D sculpting tools, such as ZBrush and Mudbox, may be preferable for users with a sculpting background, although the interaction will be via a mouse or tablet instead of a physical tool in the real-world. However, note that these programs directly modify geometry, not shading. The class of edits which are easy to make with our tool are difficult to reproduce with a 3D sculpting tool, and vice versa. I will return to this difference later in the talk.
-- click --
Our approach can be considered an approximate, special case of Shape-from-Shading, where we use existing geometry to make incremental shading changes.

--- refuse ---
? [McCann and Pollard 2008] SIGGRAPH 2008 "Real-time Gradient-domain painting"

TODO:

show less, focus on the ones I do show.

mention explicitly that [nealen, zimmerman] work uses a similar machinery (laplacian energy) applied to moving silhouettes, while we apply it to changing gradients.

emphasize that we use existing geometry to solve this specific case of shape from shading. add after "related work".

emphasize difference with previous work, specifically sculpting.
- maybe show the comparison movie.
- maybe show a diagram: previous system creates bumps in or out, while our system changes the gradient and smoothes it out.

Related Work

Relief Modeling
[van Overveld 1996]
[Zhang et al. 2001]
[Bourguignon et al. 2004]
[Zeng et al. 2005]
[Wu et al. 2007]
[Ng et al. 2007]

Sketch-based modeling
[Igarashi et al. 1999]
[Cheutet et al. 2004]
[Lawrence and Funkhouser 2004]
[Kara et al. 2006]
[Karpenko and Hughes 2006]
[Nealen et al. 2007]

Shape-from-Shading
[Rushmeier et al. 2003]
[Prados 2004]

Silhouette Editing
[Nealen et al. 2005]
[Zimmermann et al. 2007]

3D Sculpting
ZBrush [Pixologic]
Mudbox [Autodesk]

Shape-Preserving Deformation
[Sorkine et al. 2004]
[Yu et al. 2004]

There is a large body of related work.
There are a variety of surface editing tools, each complementary to ours. In particular, no relief editing tools allows users to simply "draw what they want to see."
-- click --
For the complementary task of silhouette editing, [Nealen et al. 2005] introduced one of the first techniques in this direction. Our machinery is similar to theirs, as well.
-- click --
3D sculpting tools, such as ZBrush and Mudbox, may be preferable for users with a sculpting background, although the interaction will be via a mouse or tablet instead of a physical tool in the real-world. However, note that these programs directly modify geometry, not shading. The class of edits which are easy to make with our tool are difficult to reproduce with a 3D sculpting tool, and vice versa. I will return to this difference later in the talk.
-- click --
Our approach can be considered an approximate, special case of Shape-from-Shading, where we use existing geometry to make incremental shading changes.

--- refuse ---
? [McCann and Pollard 2008] SIGGRAPH 2008 "Real-time Gradient-domain painting"

TODO:
show less, focus on the ones I do show.

mention explicitly that [nealen, zimmerman] work uses a similar machinery (laplacian energy) applied to moving silhouettes, while we apply it to changing gradients.

emphasize that we use existing geometry to solve this specific case of shape from shading. add after "related work".

emphasize difference with previous work, specifically sculpting.
maybe show the comparison movie.
maybe show a diagram: previous system creates bumps in or out, while our system changes the gradient and smoothes it out.

Related Work

Relief Modeling
[van Overveld 1996]
[Zhang et al. 2001]
[Bourguignon et al. 2004]
[Zeng et al. 2005]
[Wu et al. 2007]
[Ng et al. 2007]

Sketch-based modeling
[Igarashi et al. 1999]
[Cheutet et al. 2004]
[Lawrence and Funkhouser 2004]
[Kara et al. 2006]
[Karpenko and Hughes 2006]
[Nealen et al. 2007]

Shape-from-Shading
[Rushmeier et al. 2003]
[Prados 2004]

Silhouette Editing
[Nealen et al. 2005]
[Zimmermann et al. 2007]

3D Sculpting
ZBrush [Pixologic]
Mudbox [Autodesk]

Shape-Preserving Deformation
[Sorkine et al. 2004]
[Yu et al. 2004]

There is a large body of related work.
There are a variety of surface editing tools, each complementary to ours. In particular, no relief editing tools allows users to simply “draw what they want to see.”
-- click --
For the complementary task of silhouette editing, [Nealen et al. 2005] introduced one of the first techniques in this direction. Our machinery is similar to theirs, as well.
-- click --
3D sculpting tools, such as ZBrush and Mudbox, may be preferable for users with a sculpting background, although the interaction will be via a mouse or tablet instead of a physical tool in the real-world. However, note that these programs directly modify geometry, not shading. The class of edits which are easy to make with our tool are difficult to reproduce with a 3D sculpting tool, and vice versa. I will return to this difference later in the talk.
-- click --
Our approach can be considered an approximate, special case of Shape-from-Shading, where we use existing geometry to make incremental shading changes.

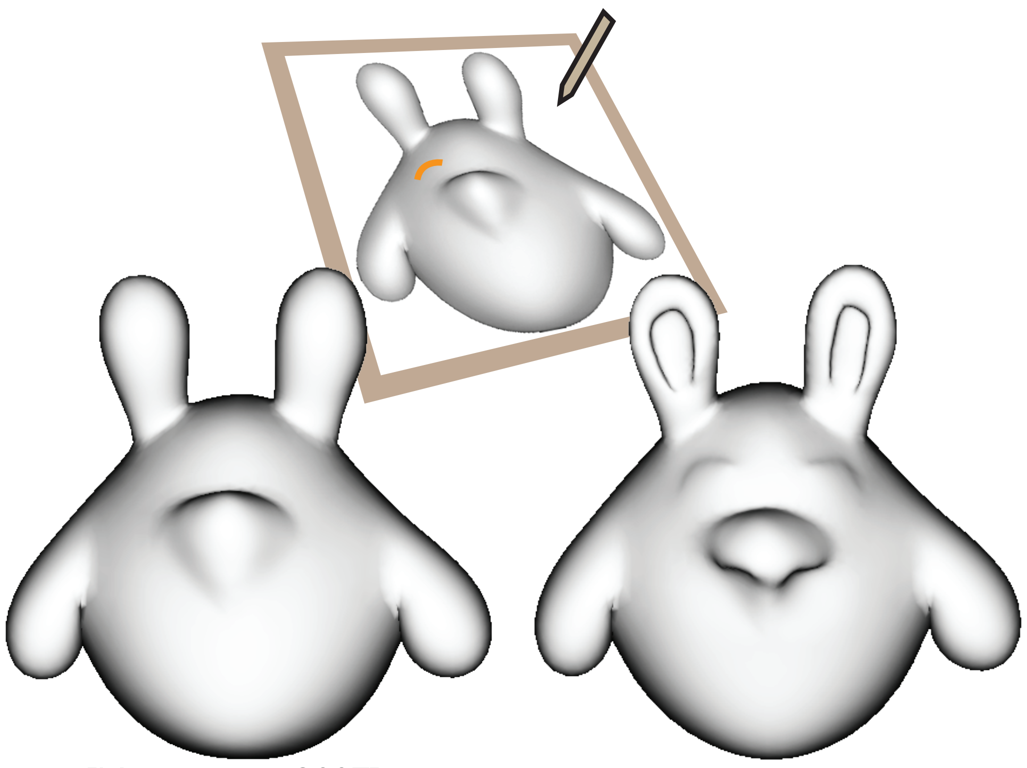
--- refuse ---
? [McCann and Pollard 2008] SIGGRAPH 2008 “Real-time Gradient-domain painting”

TODO:
show less, focus on the ones I do show.

mention explicitly that [nealen, zimmerman] work uses a similar machinery (laplacian energy) applied to moving silhouettes, while we apply it to changing gradients.

emphasize that we use existing geometry to solve this specific case of shape from shading. add after "related work".

emphasize difference with previous work, specifically sculpting.
- maybe show the comparison movie.
- maybe show a diagram: previous system creates bumps in or out, while our system changes the gradient and smoothes it out.

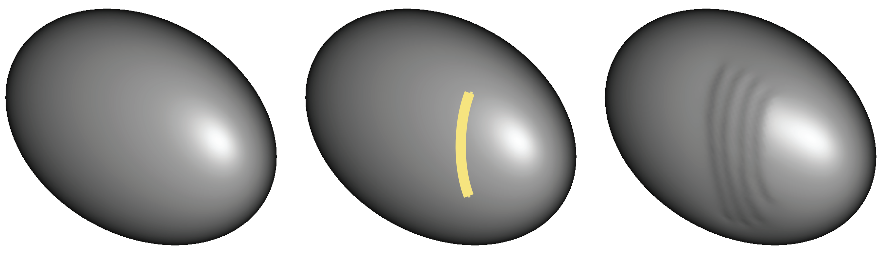


[Nealen et al. 2007]

User Interface

I will begin with a presentation of our system's three primary tools.
The algorithms used will be presented in the second half of this talk.

Shading Strokes

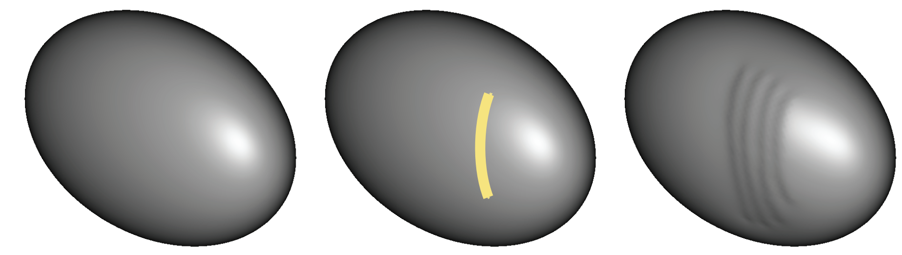


The primary strokes in our system are shading strokes.
-- click --
Here we see thin, dark shading strokes drawn on the surface of an ellipsoid. Note that the geometry is modified after every stroke to achieve the desired appearance. Also note that the surface away from the stroke does not appear to change.
Here are thicker strokes drawn on the same ellipsoid.

--- next slide:
Shading strokes have varying width, opacity, and softness --- similar to parameters in painting programs like Photoshop.

live demo? :-(

Shading Strokes

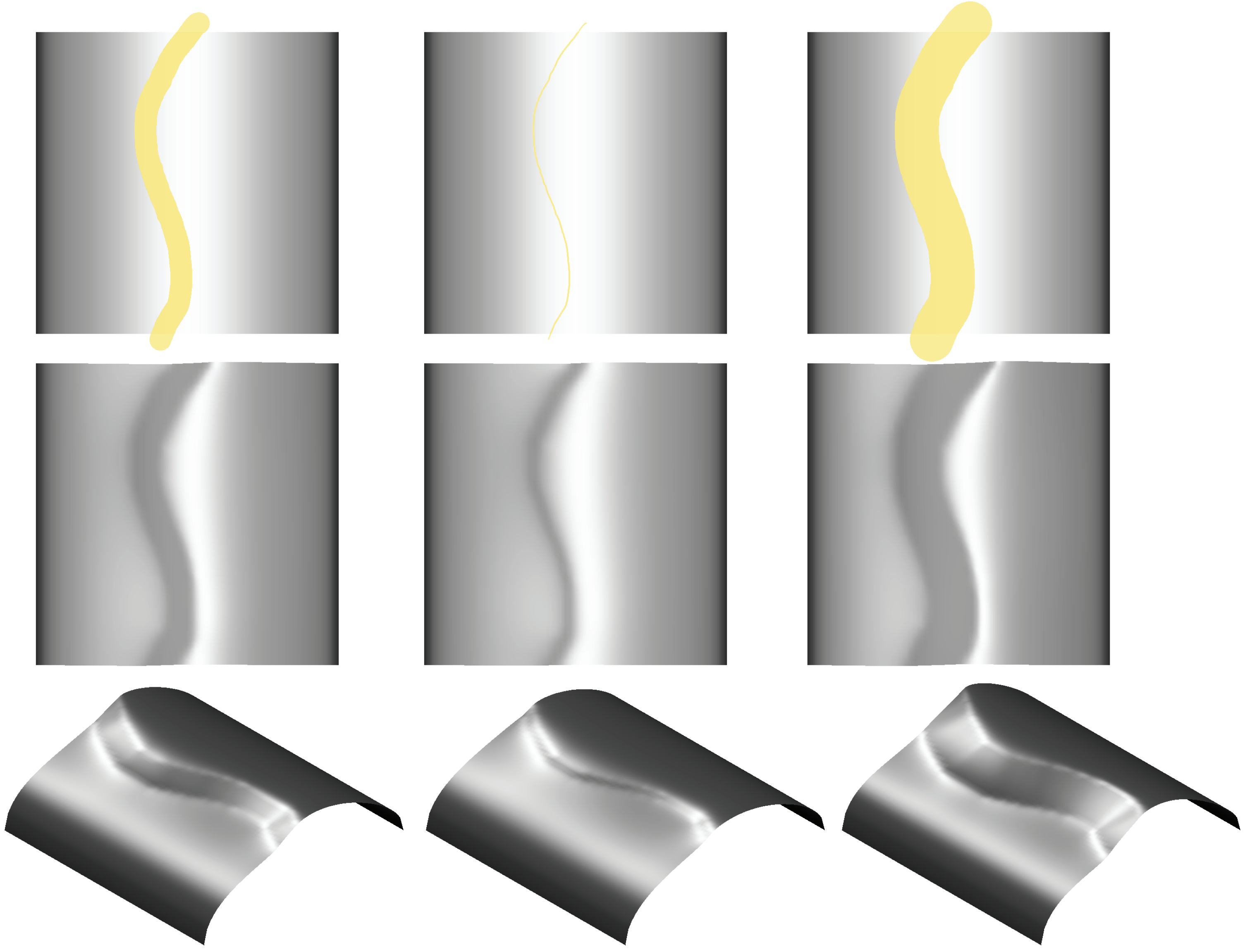


The primary strokes in our system are shading strokes.
-- click --
Here we see thin, dark shading strokes drawn on the surface of an ellipsoid. Note that the geometry is modified after every stroke to achieve the desired appearance. Also note that the surface away from the stroke does not appear to change.
Here are thicker strokes drawn on the same ellipsoid.

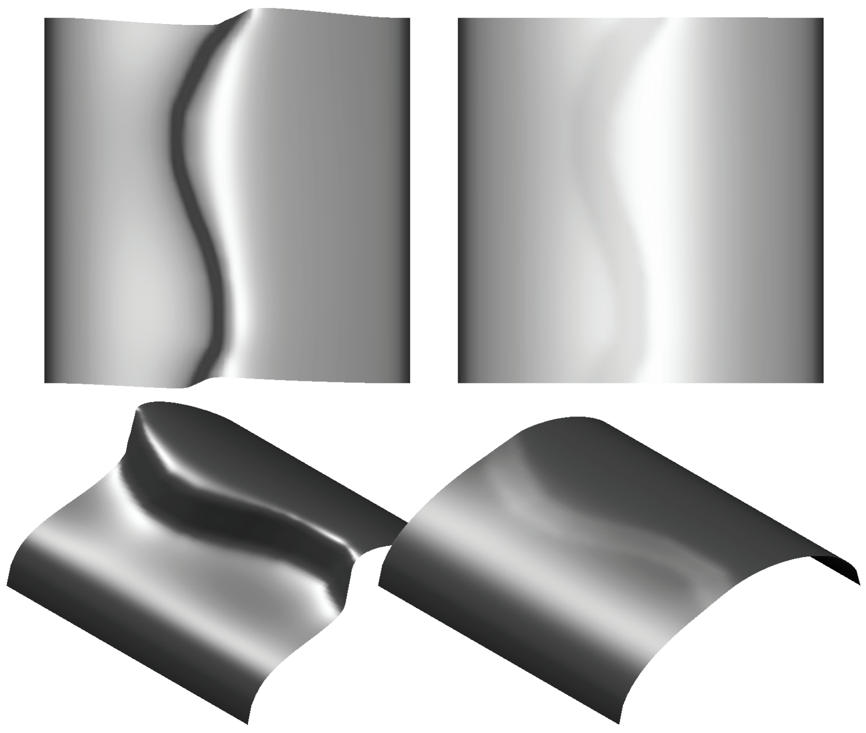
--- next slide:
Shading strokes have varying width, opacity, and softness --- similar to parameters in painting programs like Photoshop.

live demo? :-(

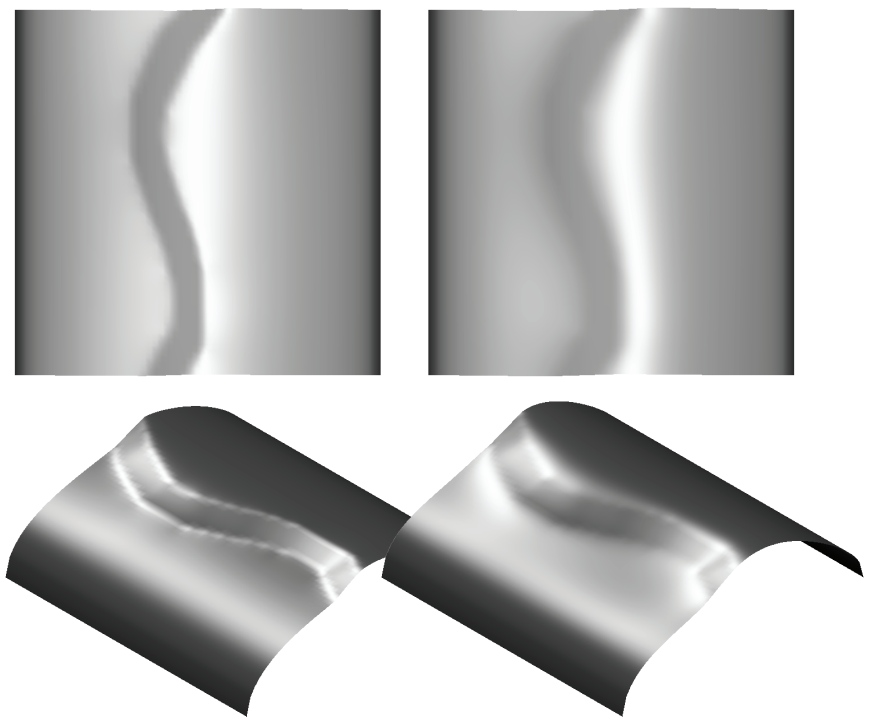
Stroke Attributes



width



opacity



softness

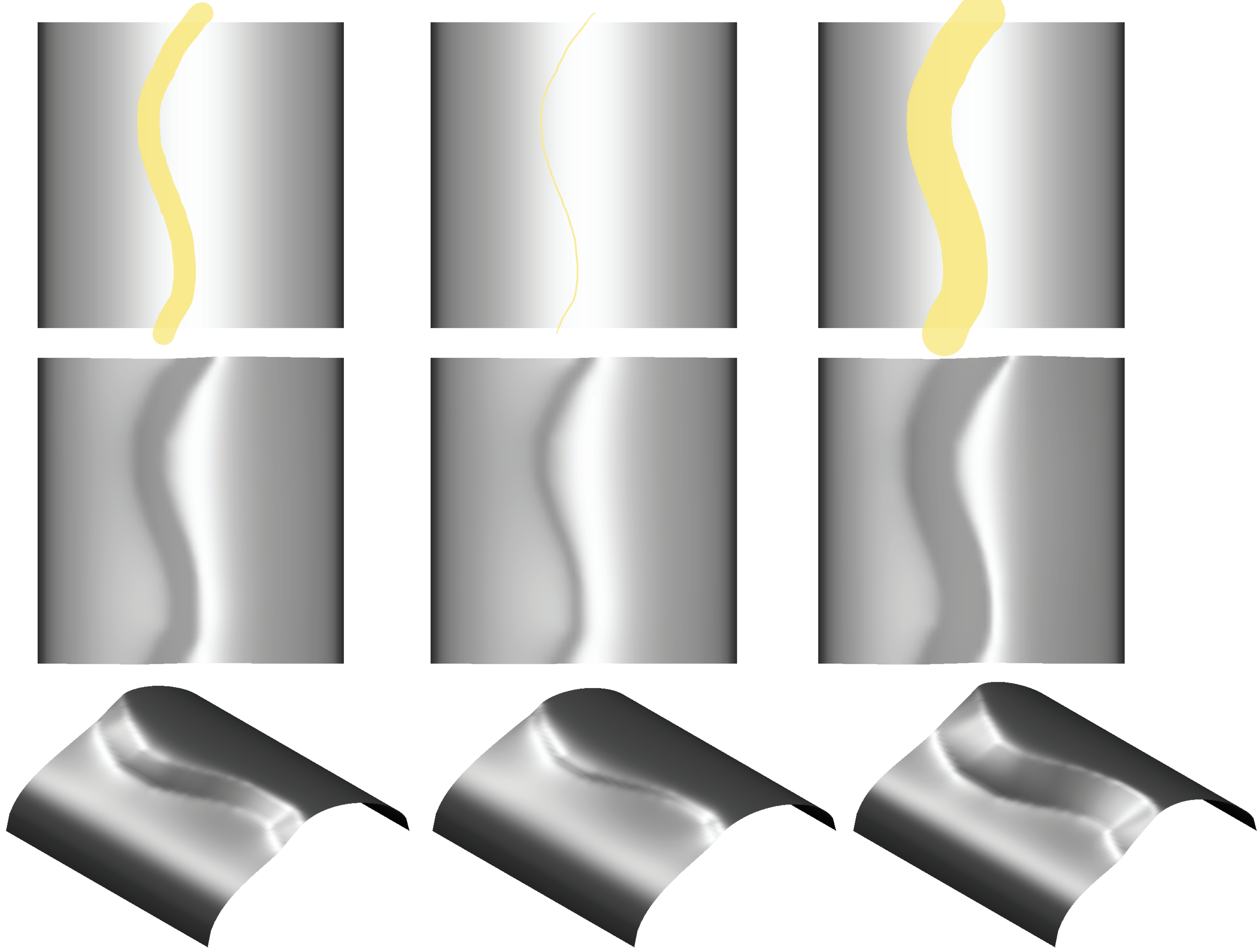
Shading strokes have varying width, opacity, and softness --- similar to parameters in painting programs like Photoshop.

These diagrams depict the effects of strokes with varying attributes applied to a piece of cylinder.

-- click --
Width controls the size of the brush

LIVE DEMO: draw a few strokes on the shallow ellipse, rotate.

Stroke Attributes: Width



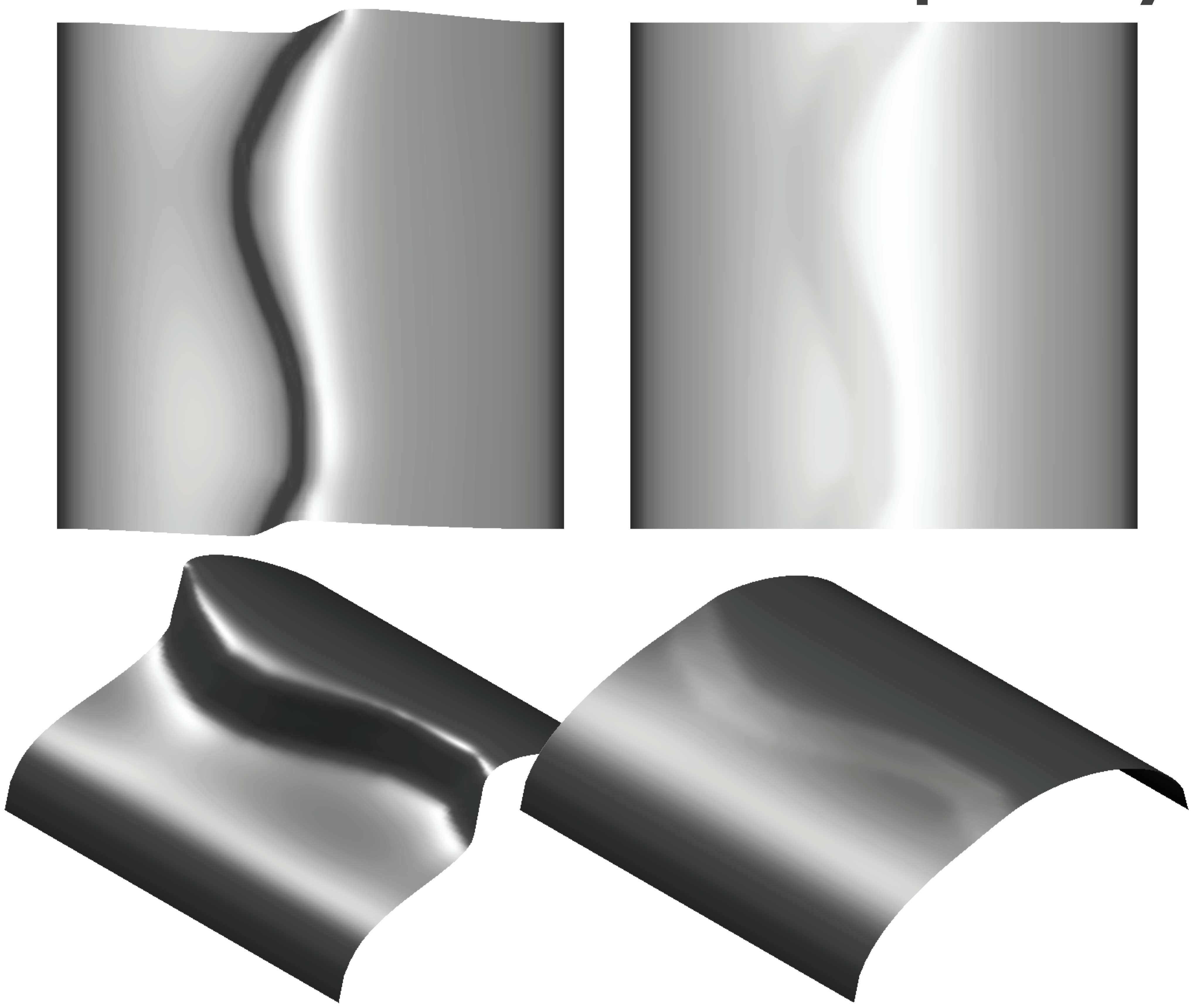
Shading strokes have varying width, opacity, and softness --- similar to parameters in painting programs like Photoshop.

These diagrams depict the effects of strokes with varying attributes applied to a piece of cylinder.

-- click --
Width controls the size of the brush

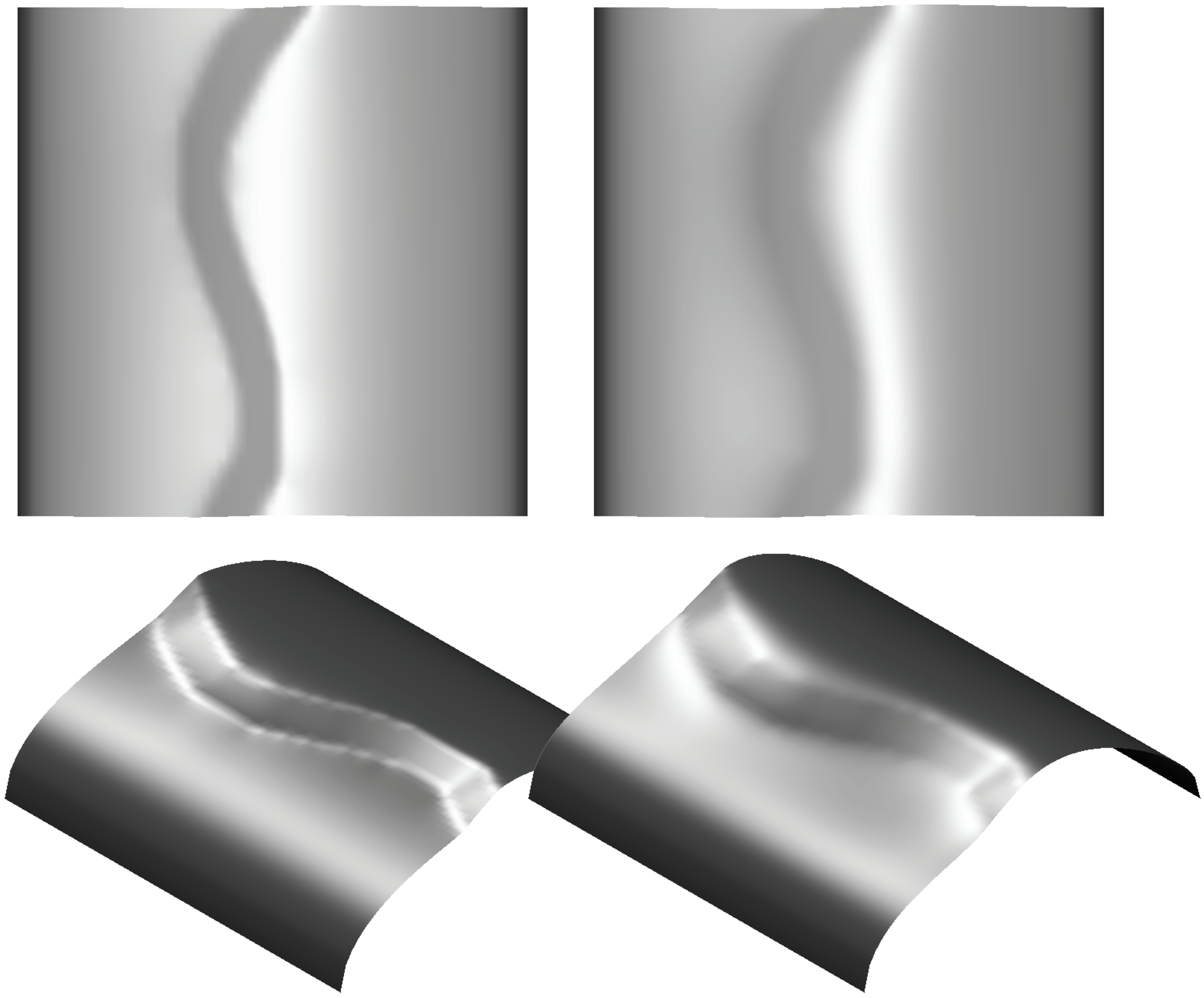
LIVE DEMO: draw a few strokes on the shallow ellipse, rotate.

Stroke Attributes: Opacity



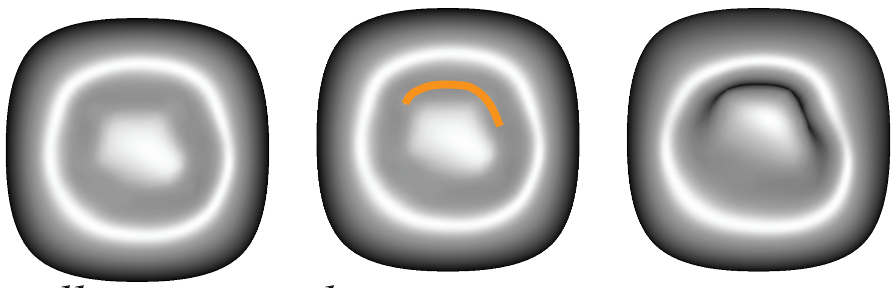
Opacity determines the blend between the brush color and the current mesh appearance.

Stroke Attributes: Softness



Softness determines the brush falloff, whether it has a sharp of soft edge.

Silhouette Stroke



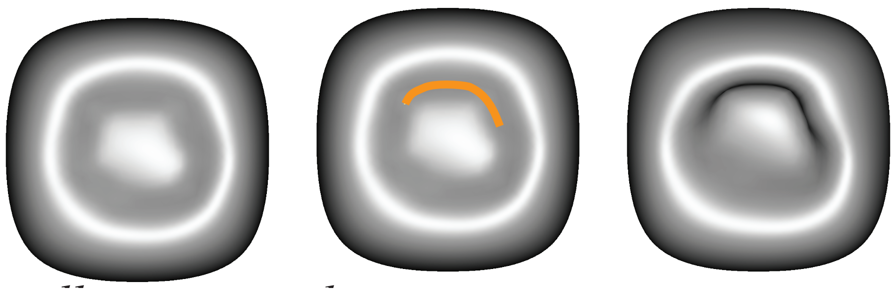
Next we have silhouette strokes, which create new silhouettes under the stroke.
-- click --
Silhouette strokes have varying width and softness, which controls the width of the fold and smoothness leading in and out of it.

skip: [Silhouette strokes have the same effect as 100% darkness shading strokes when using a headlight.]

Once a silhouette is created, it can be edited with a silhouette editing tool such as [Nealen et al. 2005] or [Zimmermann et al. 2007].

live demo? :-(

Silhouette Stroke



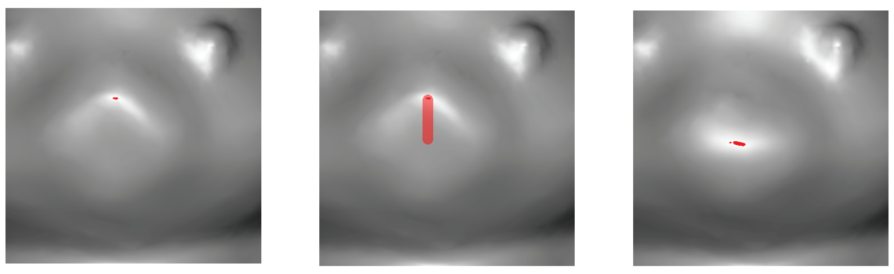
Next we have silhouette strokes, which create new silhouettes under the stroke.
-- click --
Silhouette strokes have varying width and softness, which controls the width of the fold and smoothness leading in and out of it.

skip: [Silhouette strokes have the same effect as 100% darkness shading strokes when using a headlight.]

Once a silhouette is created, it can be edited with a silhouette editing tool such as [Nealen et al. 2005] or [Zimmermann et al. 2007].

live demo? :-(

Highlight Moving



Special care must be taken when manipulating highlights, as
highlight points are actually singularities in the shading field and cannot in general be removed without a large, undesirable surface change. I will touch upon this later; For a detailed explanation of this statement, please see the paper.

-- play video --

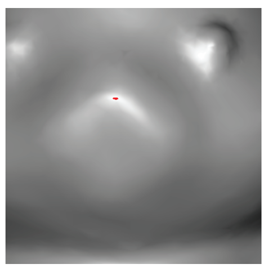
For this reason, we have a special tool for moving highlight points.

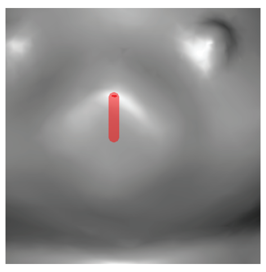
For changing the shape of a highlight, shading strokes can be used.

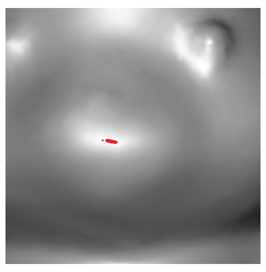
We consider our highlight tool to be somewhat limited, and future work is required for precise highlight control.

live demo? :-(

Highlight Moving



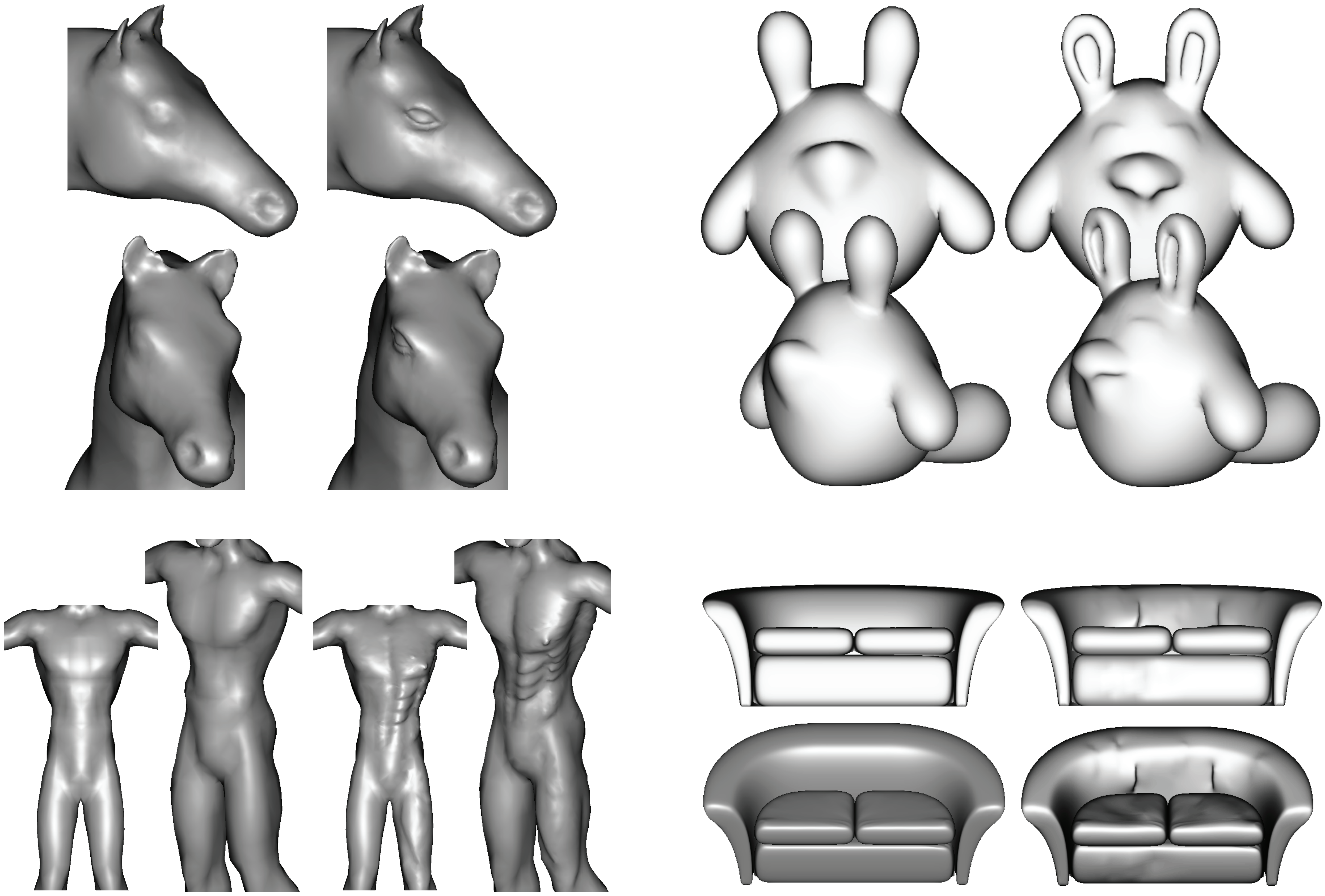




Special care must be taken when manipulating highlights, as
highlight points are actually singularities in the shading field and cannot in general be removed without a large, undesirable surface change. I will touch upon this later; For a detailed explanation of this statement, please see the paper.
-- play video --
For this reason, we have a special tool for moving highlight points.
For changing the shape of a highlight, shading strokes can be used.
We consider our highlight tool to be somewhat limited, and future work is required for precise highlight control.

live demo? :-(

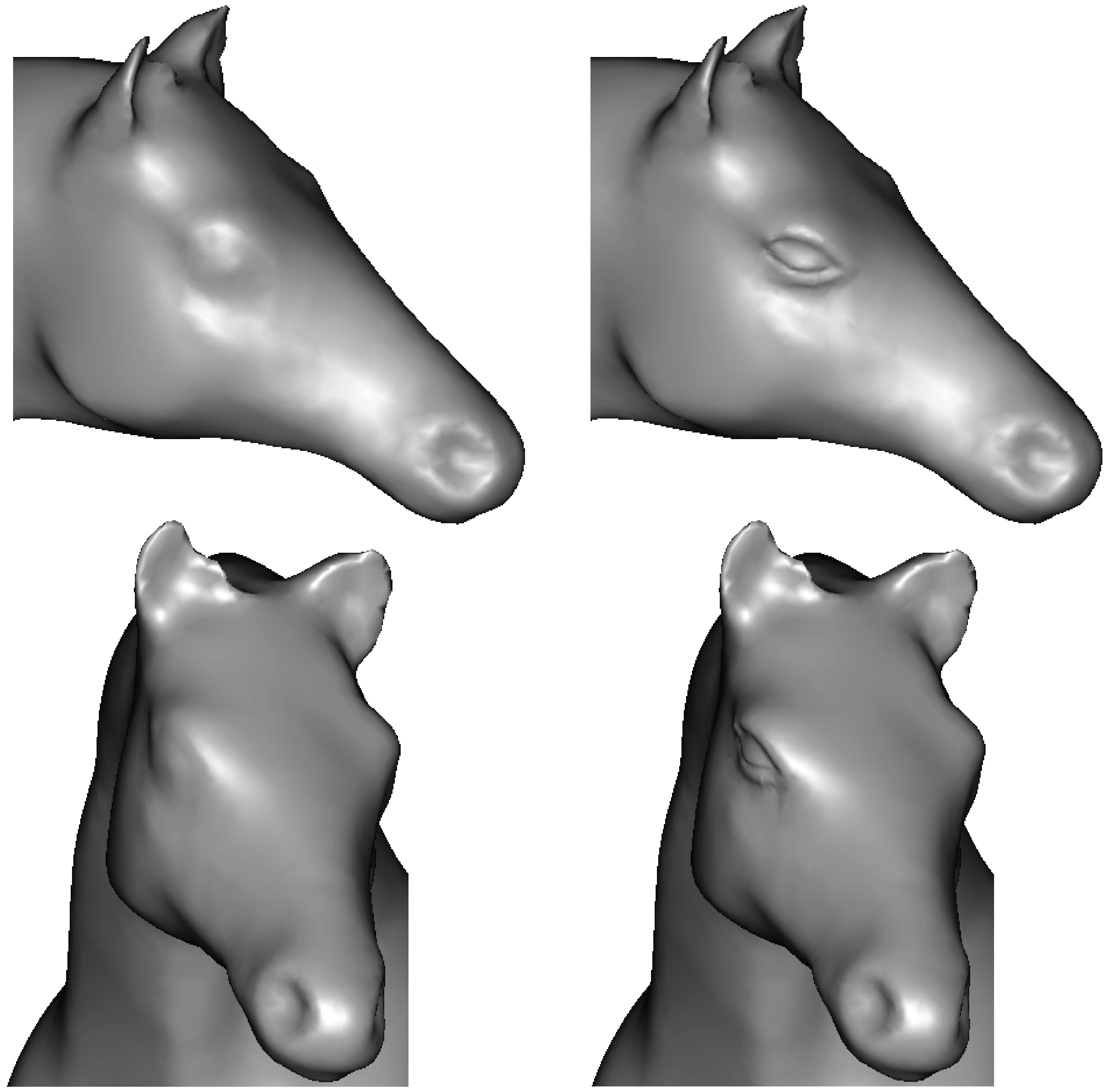
Results



Here are some results created using our system.

-- click --
In this example, we have added an eye to the simple horse model by shading it.
Note that the initial mesh is quite coarse.

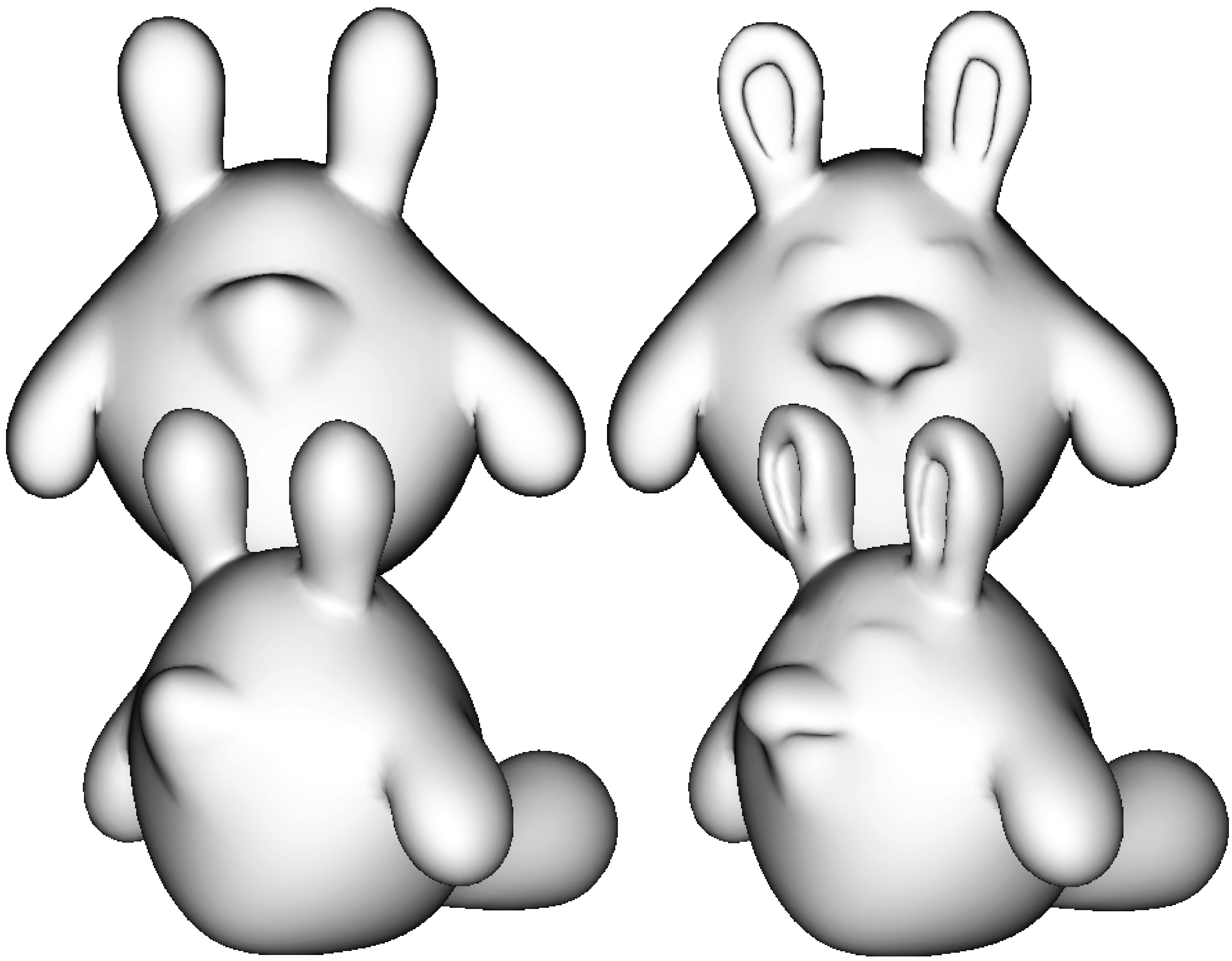
Results



Here are some results created using our system.

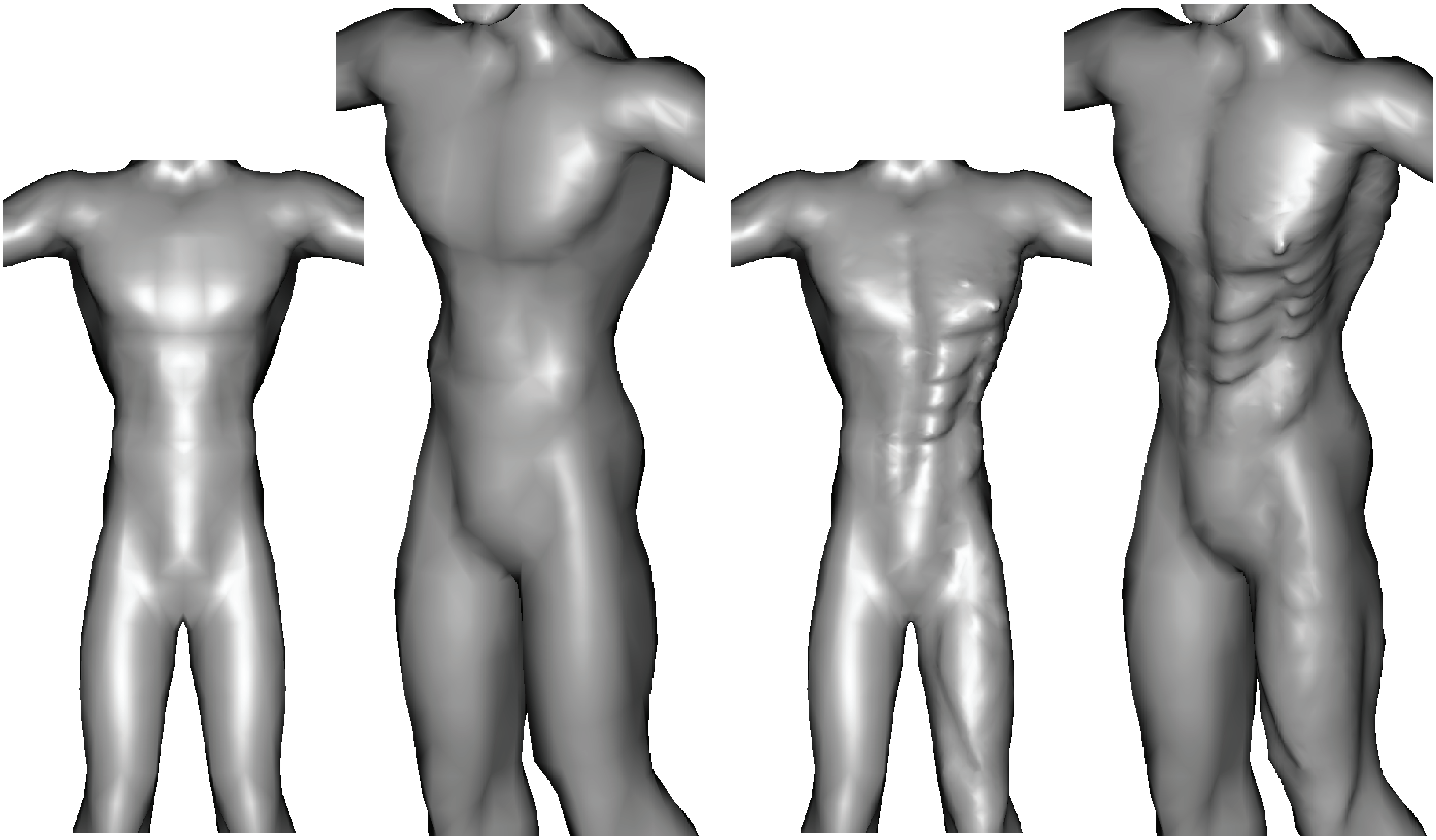
-- click --
In this example, we have added an eye to the simple horse model by shading it.
Note that the initial mesh is quite coarse.

Results



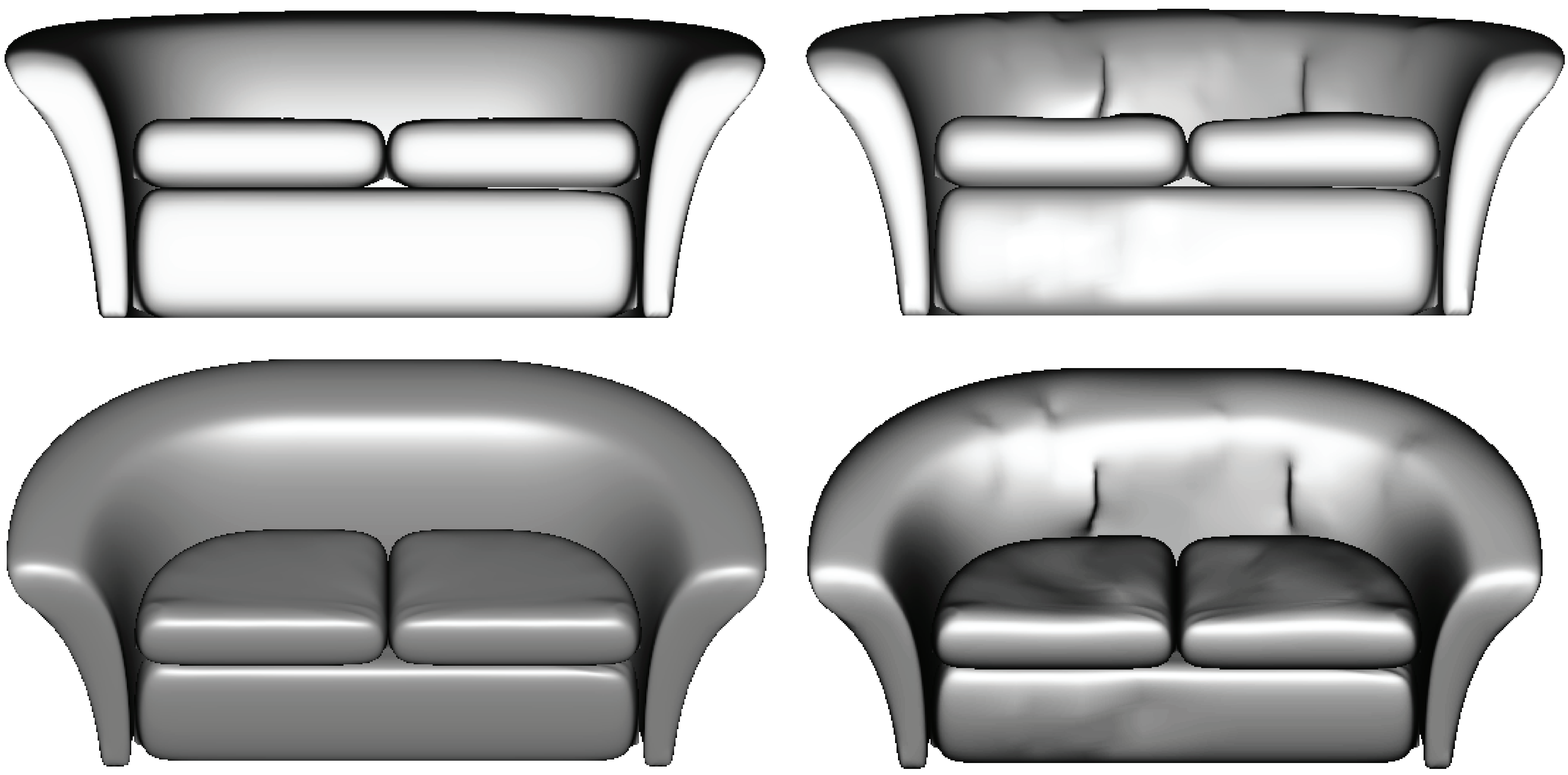
In this result, we have added facial features — ears, eyes, and nostrils — to a mesh created in the FiberMesh system, which was presented at last year's SIGGRAPH.
Many sketch-based modeling systems do not allow users to edit surface relief; complementarily, our tool can be used for this purpose.

Results



Here, muscles have been added to the abdomen and leg of a human. This example was created by a professional animator using our tool.

Results



In this example, we have added detail to a couch. While the geometric effects of these folds and crumples may be difficult to specify, the desired shading is easy to draw.

Video

Here is a video of our system in use.
-- click --
[Show FiberMesh example and Mannequin head, then cut.]

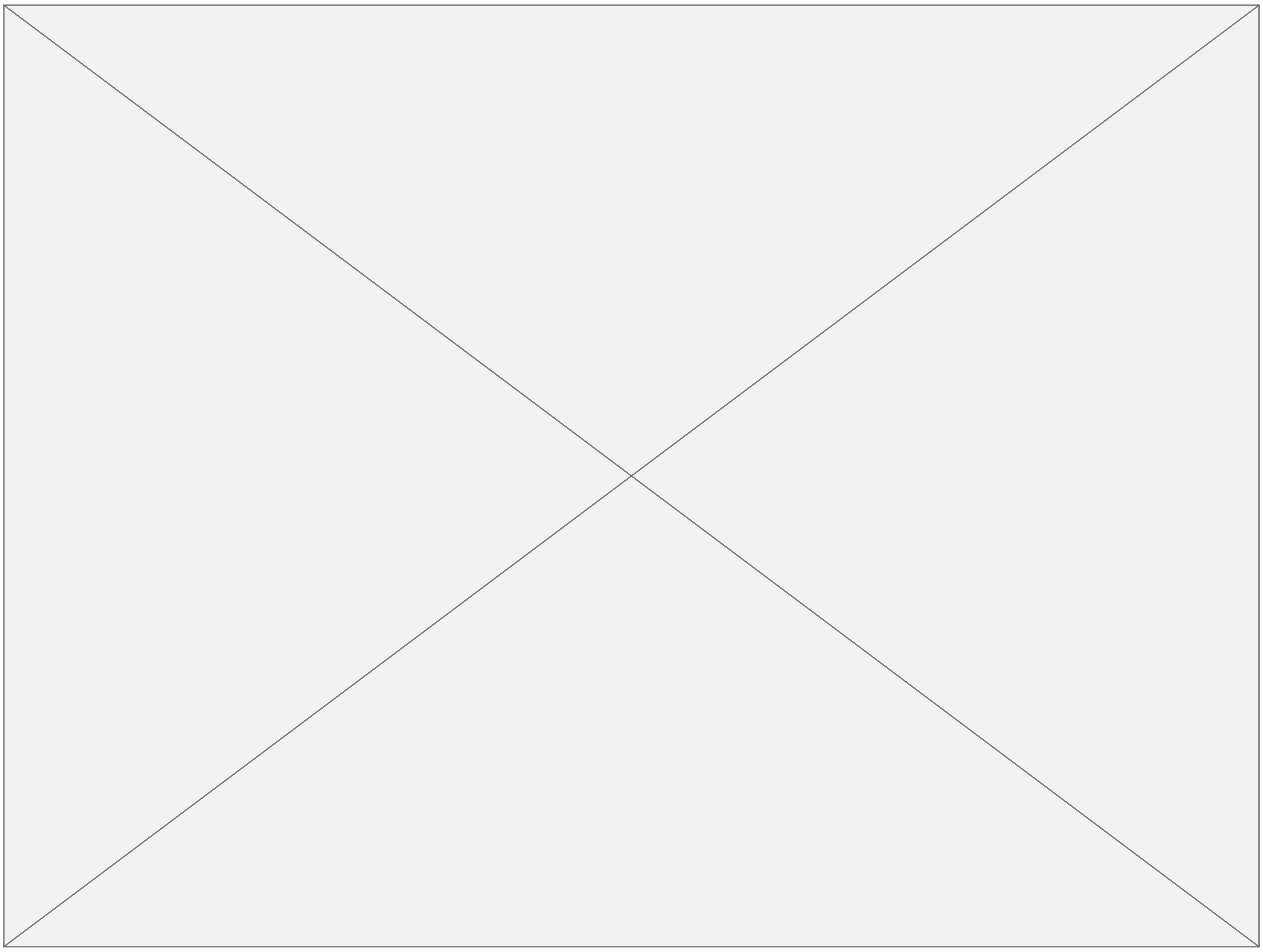
[TODO: transcribe the audio from the video?]

First, the user draws several light shading strokes to add eyes to a model created in the FiberMesh system. Next, silhouette strokes are used to create nostrils. Finally, silhouette strokes are drawn to add ear cavities. The ear cavities are initially convex, and the users clicks a “flip” button to reverse the orientation of the stroke.

In this next example, we perform several edits on the mannequin head. First the user draws shading strokes to add puffy cheeks to the mannequin’s left and right sides. This is accomplished with two strokes on each side. Then the user zooms in to the mannequin’s eye. A variety of shading strokes are used to add detail to the eye.

live demo :-(

Video



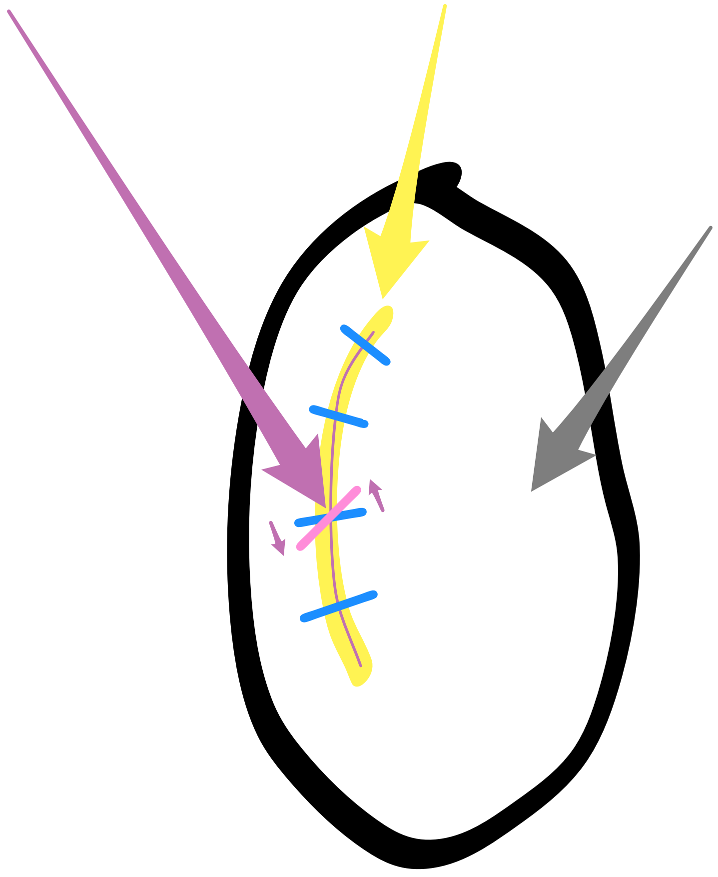
Here is a video of our system in use.
-- click --
[Show FiberMesh example and Mannequin head, then cut.]

[TODO: transcribe the audio from the video?]

First, the user draws several light shading strokes to add eyes to a model created in the FiberMesh system. Next, silhouette strokes are used to create nostrils. Finally, silhouette strokes are drawn to add ear cavities. The ear cavities are initially convex, and the users clicks a “flip” button to reverse the orientation of the stroke.

In this next example, we perform several edits on the mannequin head. First the user draws shading strokes to add puffy cheeks to the mannequin’s left and right sides. This is accomplished with two strokes on each side. Then the user zooms in to the mannequin’s eye. A variety of shading strokes are used to add detail to the eye.

live demo :-(



Technical Presentation

In the second half of the talk, I will describe the technical aspects of our system.
This includes our approach, formulation, and algorithms.

Q?: replace image.
A: I kind of like it as an icon.

I will now present the problems and our solutions for shade-based surface editing.

Criteria for Controllability

Our interface must balance:

In order for our tool to be "controllable," it should have the following properties:

Stability means that small changes produce small effects. For example, the surface should change slightly if the user draws a light stroke.

The surface under or near the stroke should change accordingly, but the surface and its shading elsewhere should appear to remain unchanged.

Finally, the effects of a stroke should be predictable.

Criteria for Controllability

Our interface must balance:

- Stability (small changes produce small effects)

In order for our tool to be “controllable,” it should have the following properties:

Stability means that small changes produce small effects. For example, the surface should change slightly if the user draws a light stroke.

The surface under or near the stroke should change accordingly, but the surface and its shading elsewhere should appear to remain unchanged.

Finally, the effects of a stroke should be predictable.

Criteria for Controllability

Our interface must balance:

- Stability (small changes produce small effects)
- Appearance and shape preserved elsewhere

In order for our tool to be "controllable," it should have the following properties:

Stability means that small changes produce small effects. For example, the surface should change slightly if the user draws a light stroke.

The surface under or near the stroke should change accordingly, but the surface and its shading elsewhere should appear to remain unchanged.

Finally, the effects of a stroke should be predictable.

Criteria for Controllability

Our interface must balance:

- Stability (small changes produce small effects)
- Appearance and shape preserved elsewhere
- Predictability

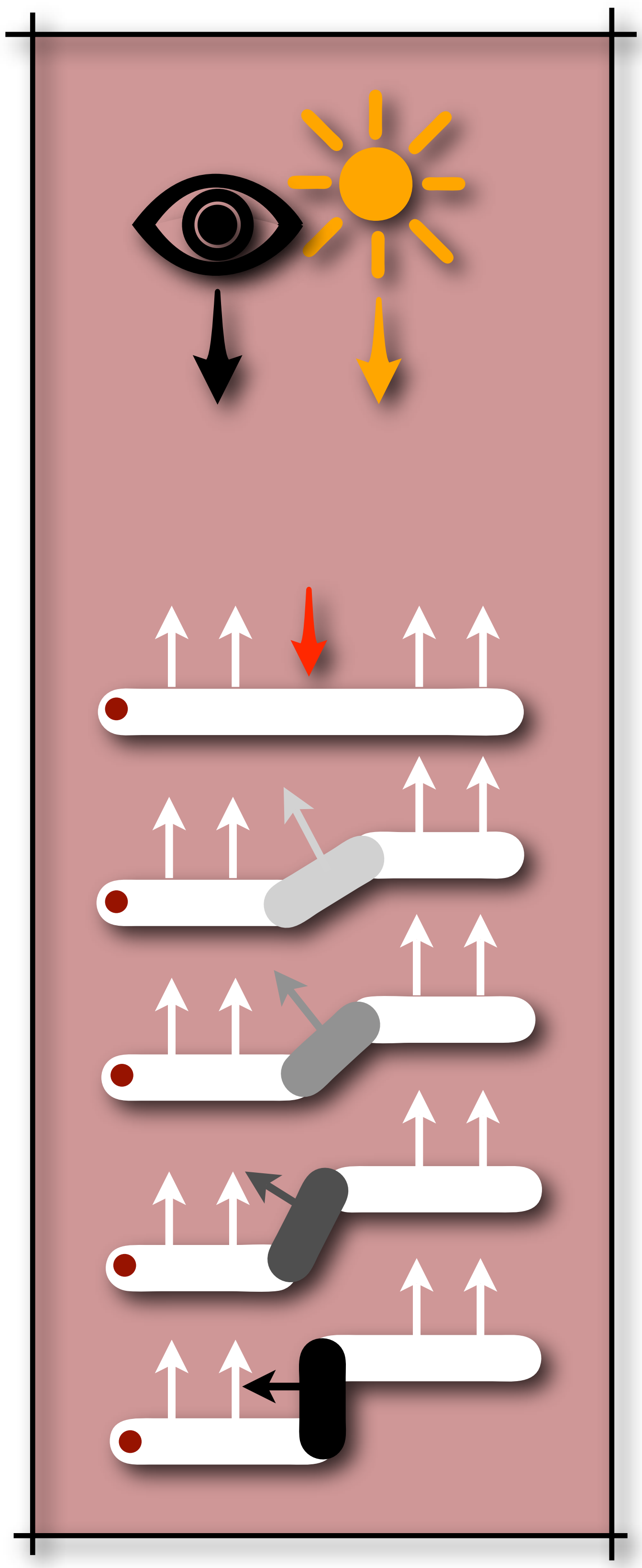
In order for our tool to be “controllable,” it should have the following properties:

Stability means that small changes produce small effects. For example, the surface should change slightly if the user draws a light stroke.

The surface under or near the stroke should change accordingly, but the surface and its shading elsewhere should appear to remain unchanged.

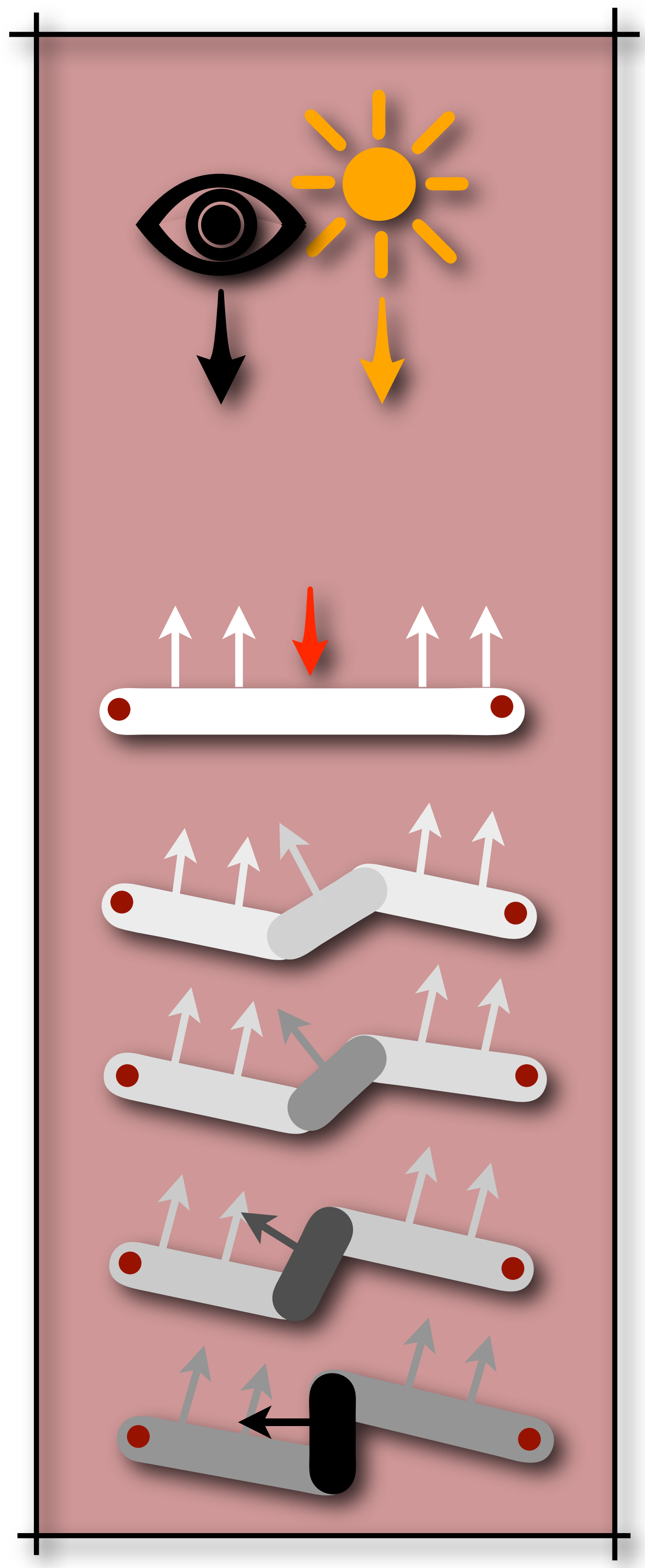
Finally, the effects of a stroke should be predictable.

Why Is This Hard?



Why is this a hard problem?
Let's consider the 2-dimensional analog, where our surface is a line, and the viewer and light are positioned above, looking down.
In this diagram, the red dot fixes the left endpoints of the lines.
As shading is a function of the normal and light direction,
If we wish to darken the line's shading in the region indicated by the red arrow, that part of the line must turn away from the light source.
If we wish for the rest of the line's appearance to remain unchanged, it must continue to face the light. We can do exactly this if we only constrain one end of the line.

Why Is This Hard?

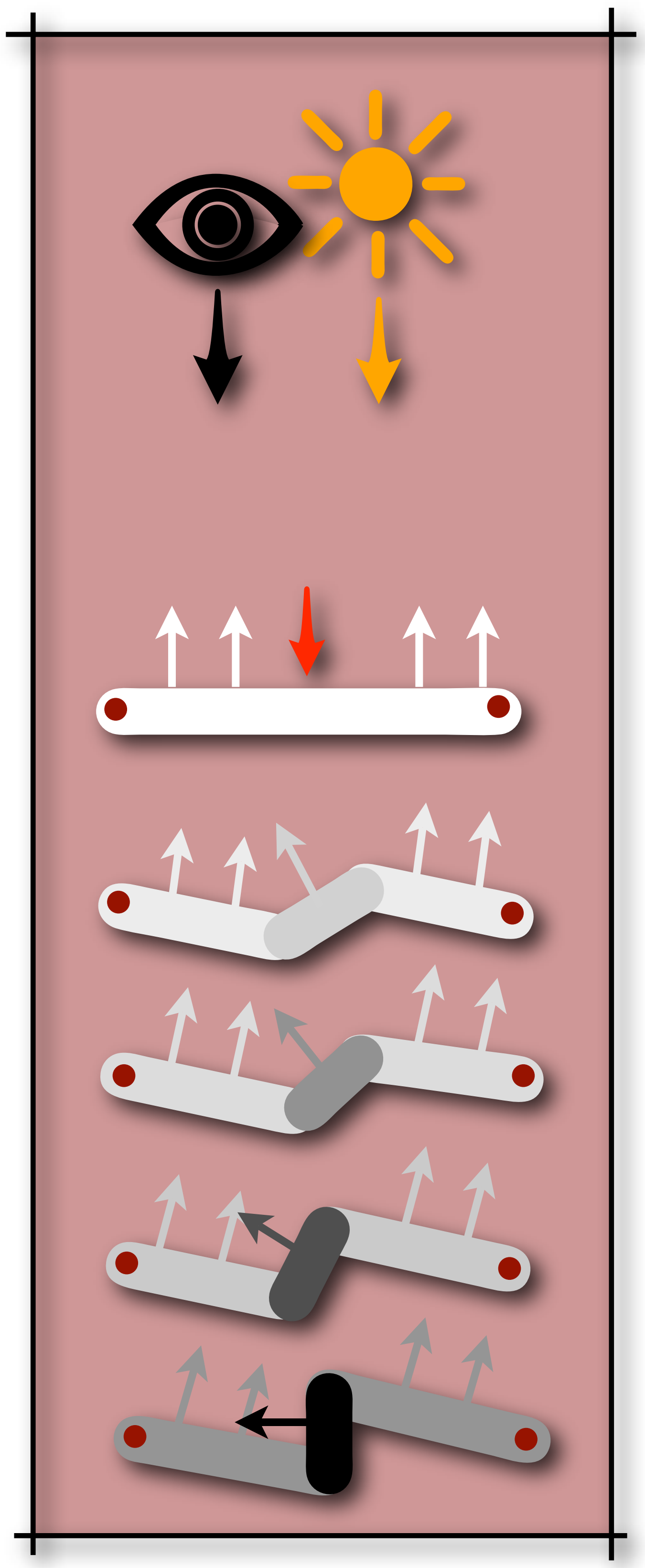


However, if both ends of the line are fixed, then we can't keep the rest of the line's appearance unchanged. The solution shown here preserves the appearance in the least squares sense.

Note that in both cases modifying the shading is a global operation; the solution may modify the entire line.

Why Is This Hard?

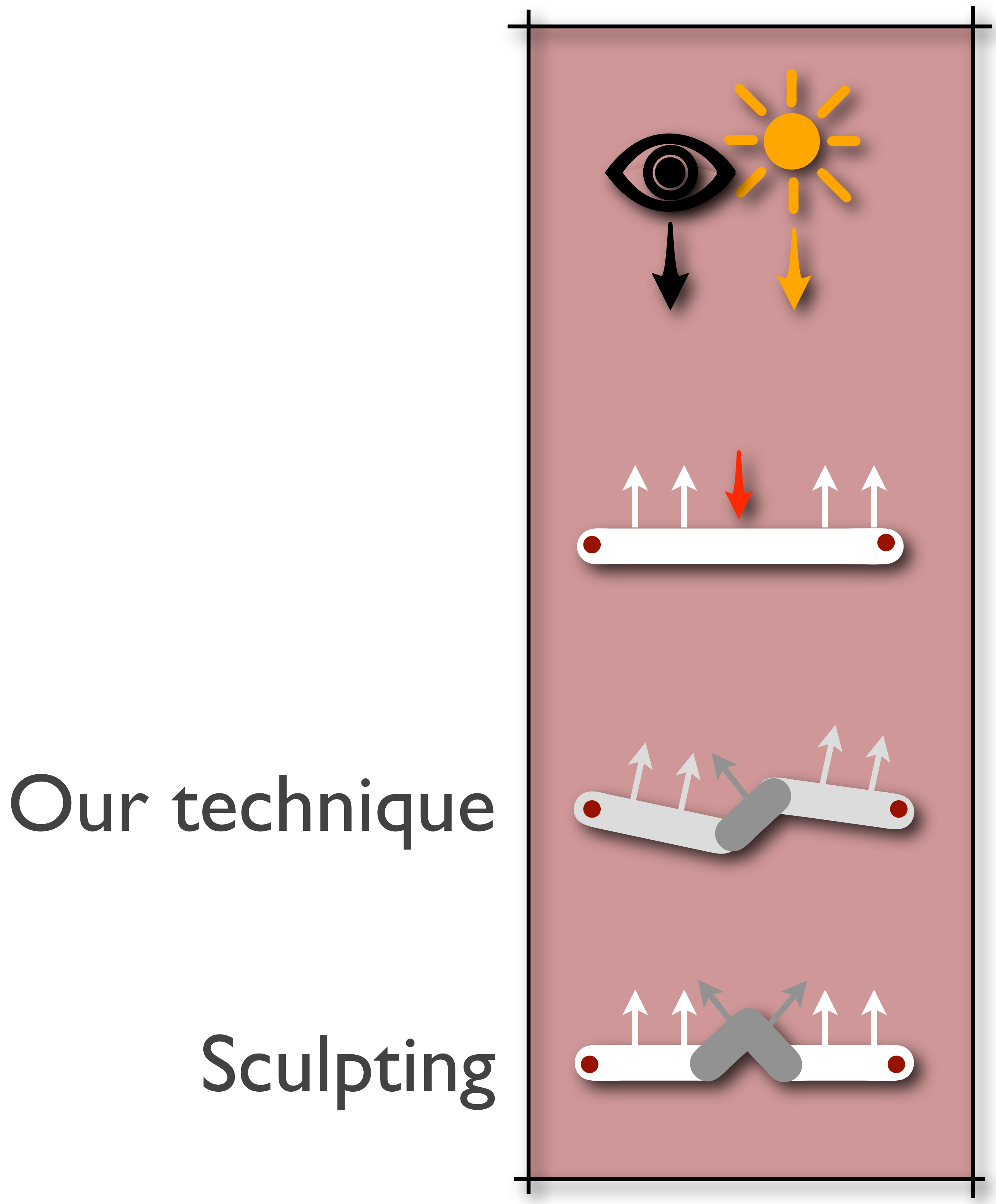
The change is global



However, if both ends of the line are fixed, then we can't keep the rest of the line's appearance unchanged. The solution shown here preserves the appearance in the least squares sense.

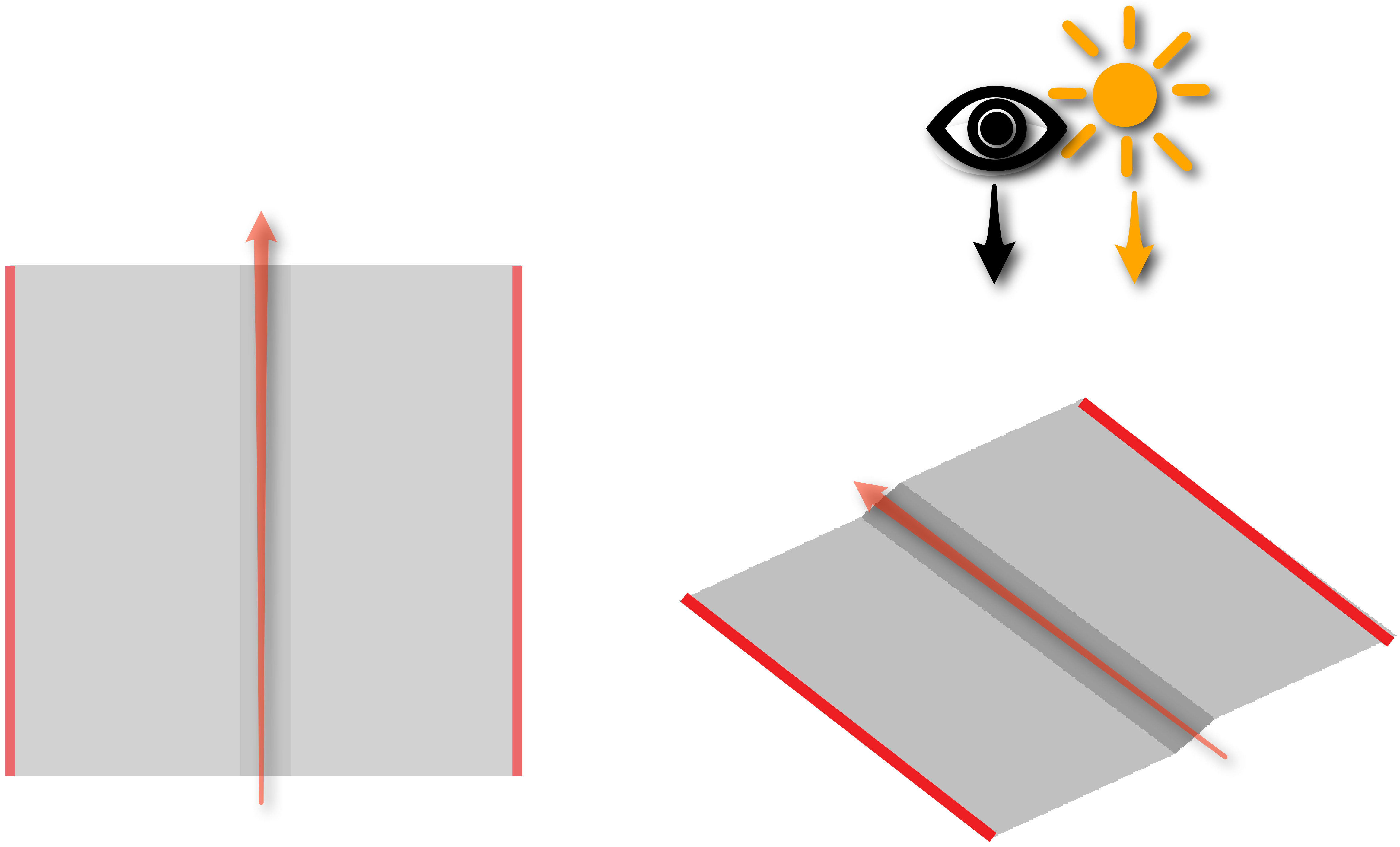
Note that in both cases modifying the shading is a global operation; the solution may modify the entire line.

Comparison with Sculpting



Before we move to 3D shading editing, here is an example to illustrate the difference between shading-based editing and 3D sculpting tools. A direct shading edit creates a single slope, rather than a hill. The hill created by a sculpting edit, however, is a local change.

Why Is This Hard?



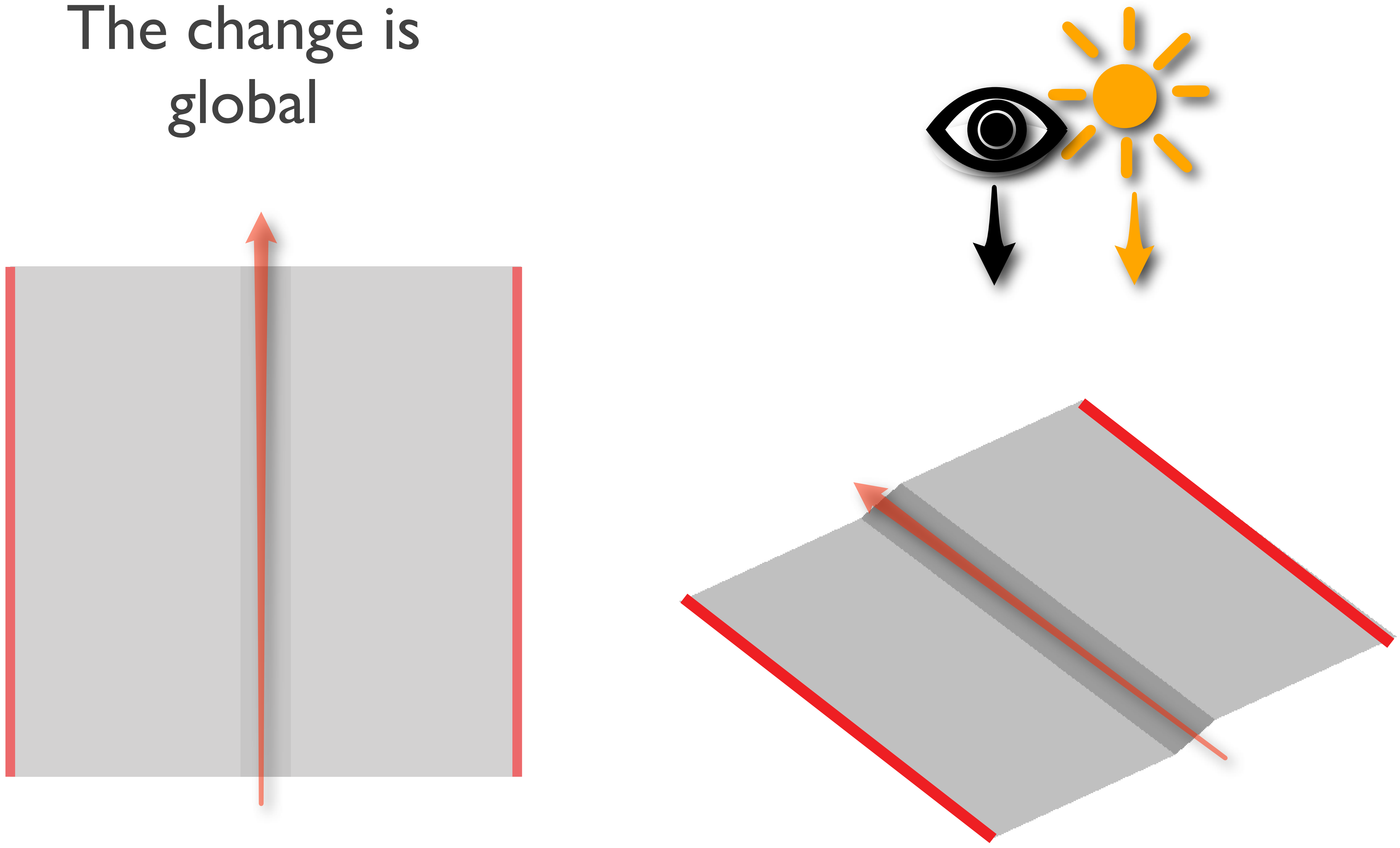
Here is the 3-dimensional version of shading editing. Here, the surface is a flat square patch and we wish to darken the surface along a vertical stroke, shown as a red arrow. With both sides of the square patch constrained, the best we can do is a least squares approximation.
-- click --
Note that the solution is again a global change, and the entire surface will be modified.

However, there is a problem with the exact shape-from-shading solution: It sometimes violates our stability and predictability principles, especially near highlights.
Moreover, it is slow, because in general a non-linear system needs to be solved.
[We would need to deal with the same set of problems as general Shape-from-Shading approaches.]

-- next slide:
However, in the case of a single darkening stroke, a good approximation to the nonlinear solution can be achieved by rotating the surface under the stroke about the stroke. This is the important observation that enables our paradigm.

Why Is This Hard?

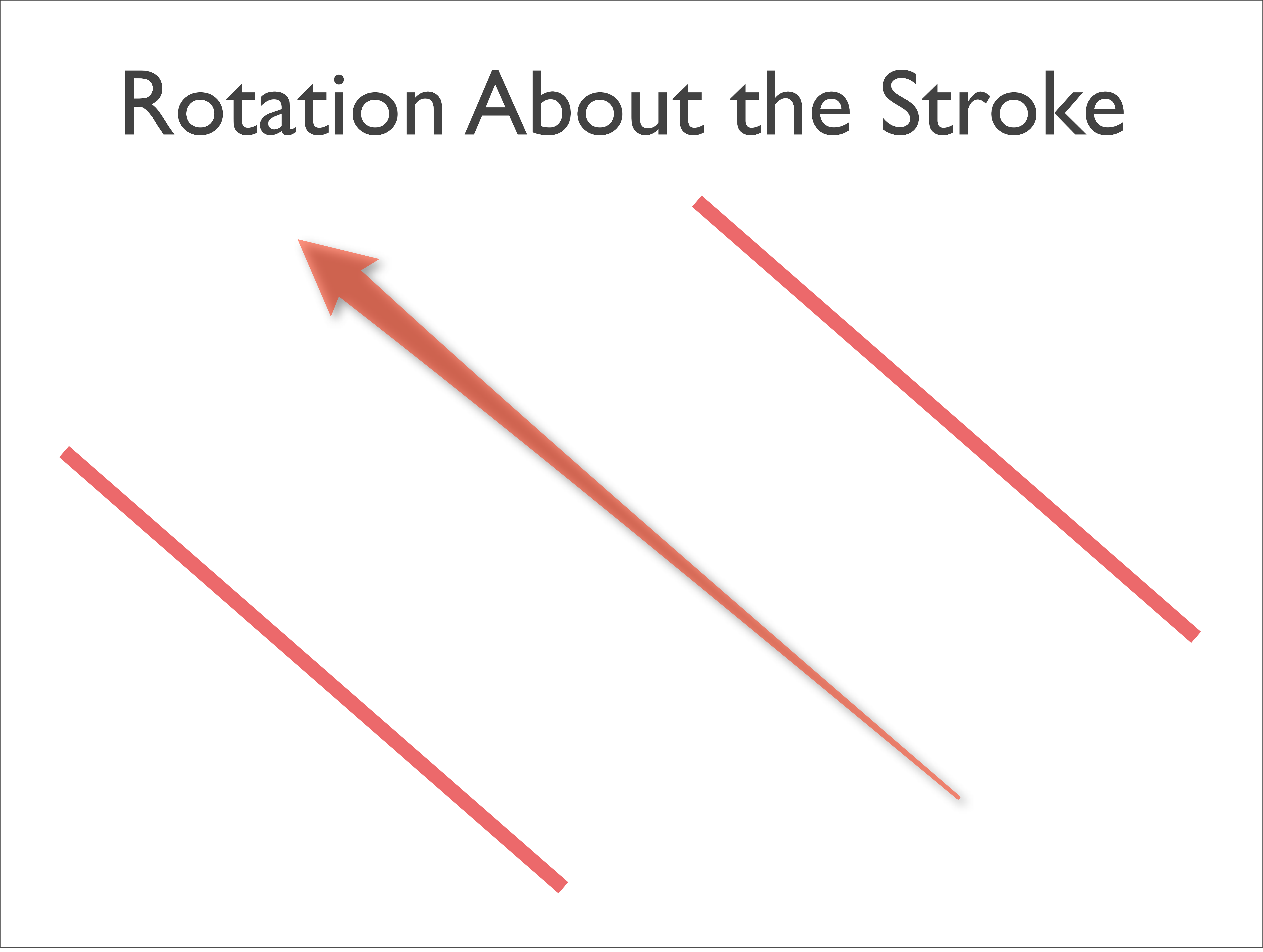
The change is global



Here is the 3-dimensional version of shading editing. Here, the surface is a flat square patch and we wish to darken the surface along a vertical stroke, shown as a red arrow. With both sides of the square patch constrained, the best we can do is a least squares approximation.
-- click --
Note that the solution is again a global change, and the entire surface will be modified.

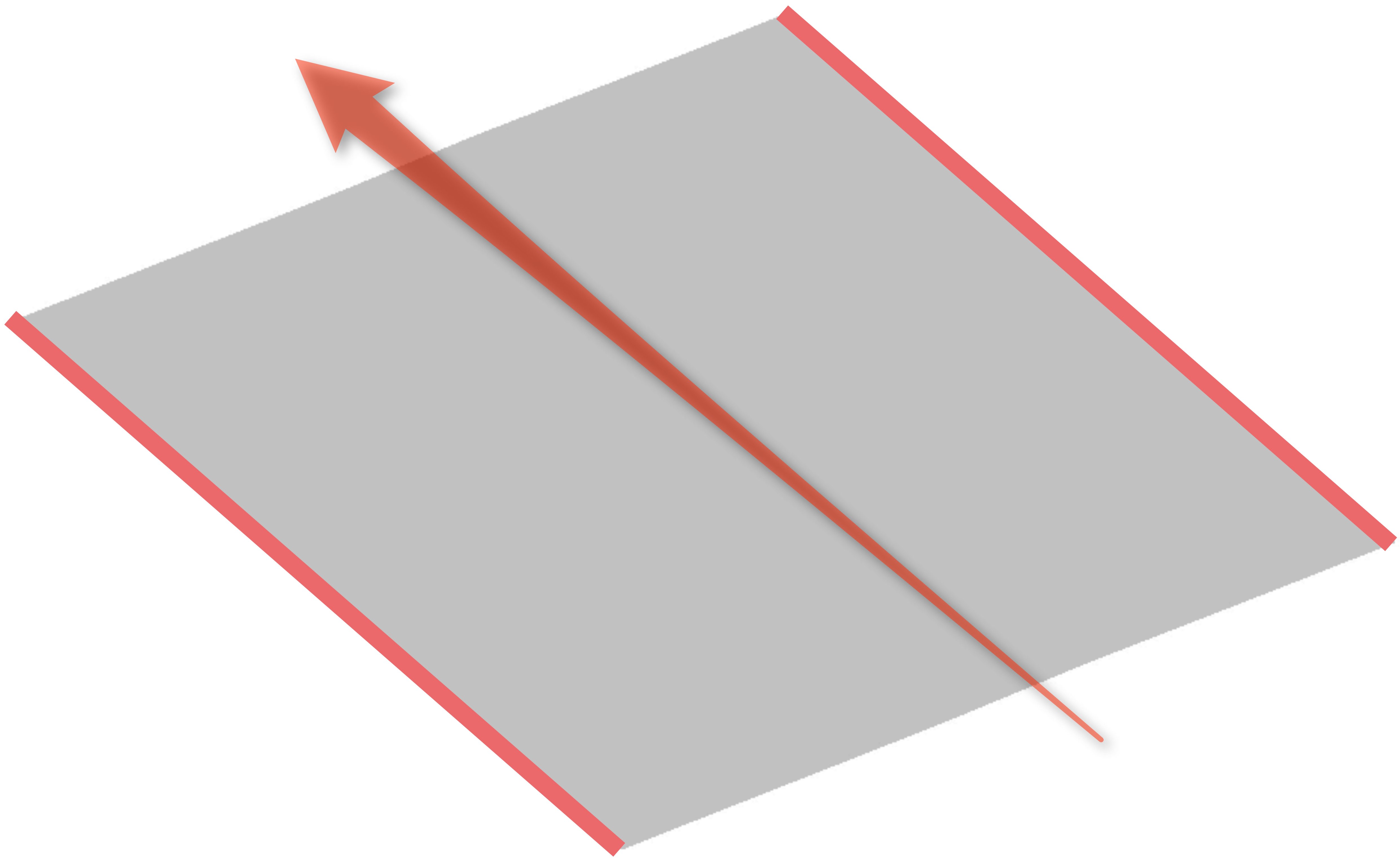
However, there is a problem with the exact shape-from-shading solution: It sometimes violates our stability and predictability principles, especially near highlights.
Moreover, it is slow, because in general a non-linear system needs to be solved.
[We would need to deal with the same set of problems as general Shape-from-Shading approaches.]

-- next slide:
However, in the case of a single darkening stroke, a good approximation to the nonlinear solution can be achieved by rotating the surface under the stroke about the stroke. This is the important observation that enables our paradigm.



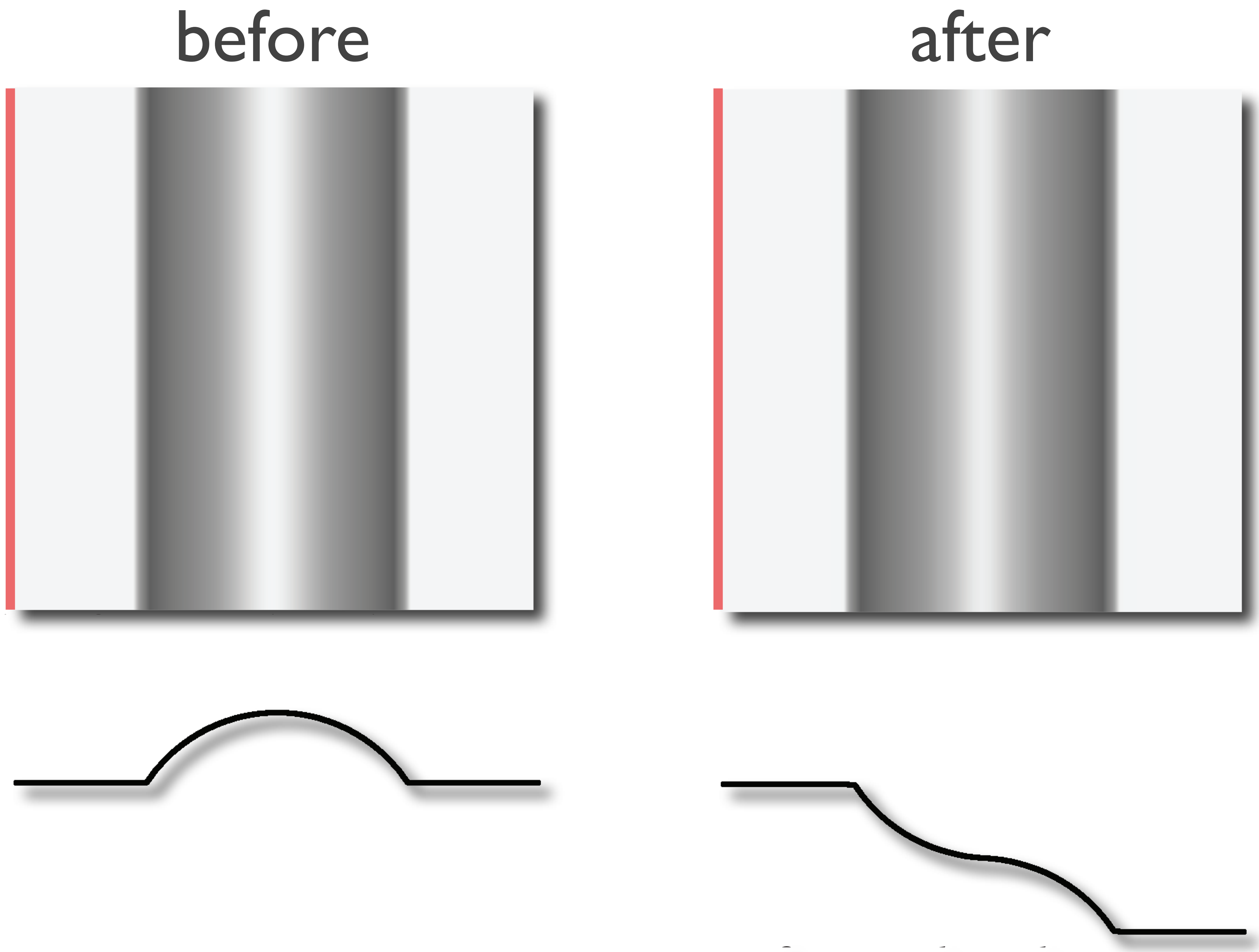
However, in the case of a single darkening stroke, a good approximation to the nonlinear solution can be achieved by
-- click --
rotating the surface under the stroke about the stroke, as shown in this video. This is the important observation that enables our paradigm.

Rotation About the Stroke



However, in the case of a single darkening stroke, a good approximation to the nonlinear solution can be achieved by
-- click --
rotating the surface under the stroke about the stroke, as shown in this video. This is the important observation that enables our paradigm.

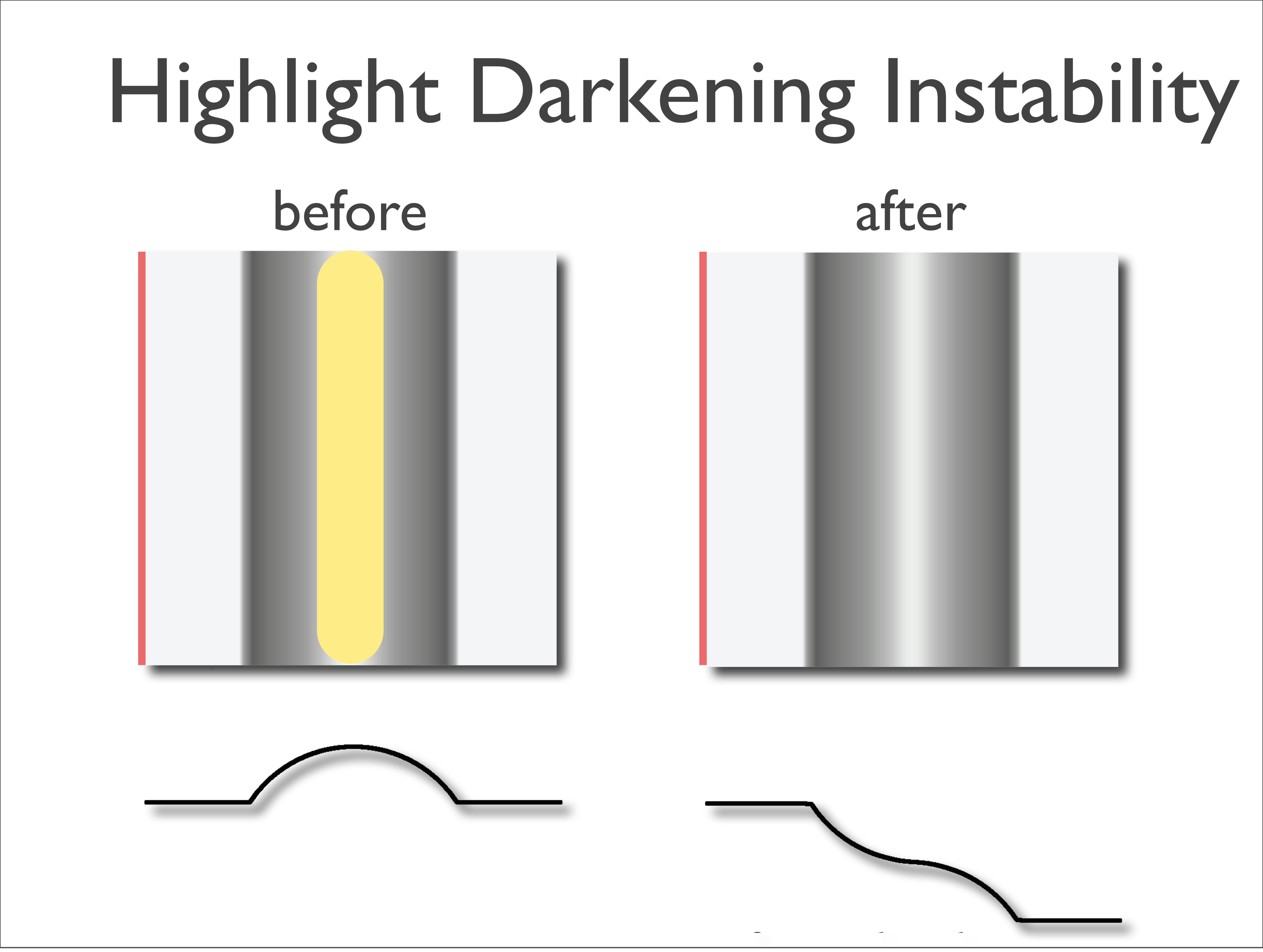
Highlight Darkening Instability



However, darkening strokes cannot be applied everywhere.

This figure illustrates the issue with highlight points: that they are local maxima in the shading field and cannot in general be removed without a large, undesirable surface change. A highlight maxima represents a hill, and the highlight cannot be removed with anything short of cutting off the entire hill.
The cylindrical cross section on the left has been darkened 1%
-- click --
to produce an image that appears similar but lacks an actual highlight and so represents a drastically different surface.
-- click --
Our solution is to detect and protect highlights from shading strokes, by terminating a stroke before it crosses a highlight.

It is for this reason that we provide the highlight moving tool.

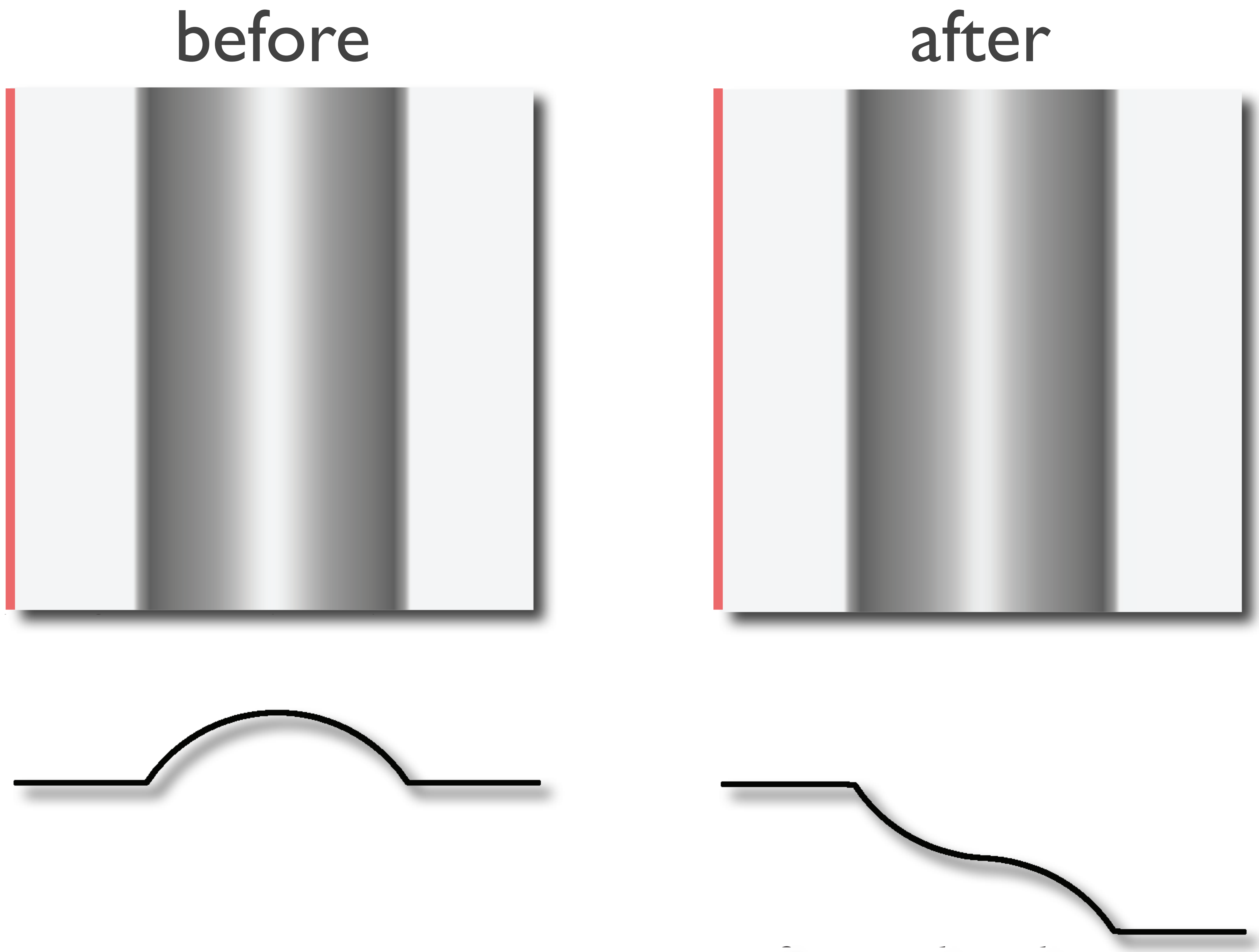


However, darkening strokes cannot be applied everywhere.

This figure illustrates the issue with highlight points: that they are local maxima in the shading field and cannot in general be removed without a large, undesirable surface change. A highlight maxima represents a hill, and the highlight cannot be removed with anything short of cutting off the entire hill.
The cylindrical cross section on the left has been darkened 1%
-- click --
to produce an image that appears similar but lacks an actual highlight and so represents a drastically different surface.
-- click --
Our solution is to detect and protect highlights from shading strokes, by terminating a stroke before it crosses a highlight.

It is for this reason that we provide the highlight moving tool.

Highlight Darkening Instability

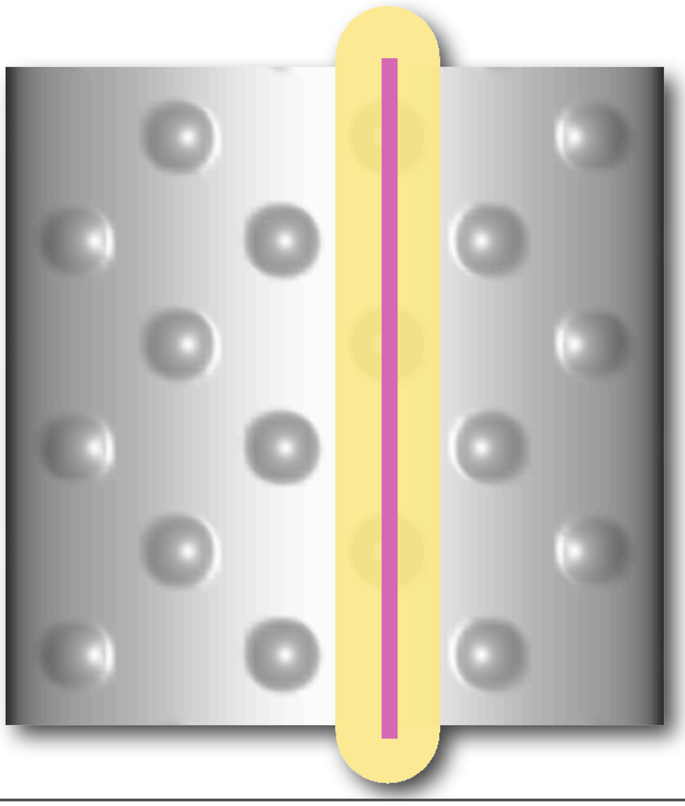


However, darkening strokes cannot be applied everywhere.

This figure illustrates the issue with highlight points: that they are local maxima in the shading field and cannot in general be removed without a large, undesirable surface change. A highlight maxima represents a hill, and the highlight cannot be removed with anything short of cutting off the entire hill.
The cylindrical cross section on the left has been darkened 1%
-- click --
to produce an image that appears similar but lacks an actual highlight and so represents a drastically different surface.
-- click --
Our solution is to detect and protect highlights from shading strokes, by terminating a stroke before it crosses a highlight.

It is for this reason that we provide the highlight moving tool.

Approach



This brings us to our approach.

After each stroke, we solve for the new surface.

-- click --
Along the centerline of the stroke (shown in purple), we rotate the surface about the projected stroke as the axis, to obtain the desired lighting. This is stable and predictable.
We also place image-plane position constraints along the centerline to keep the surface there from moving.

-- click --
Elsewhere, we use Laplacian Editing Energy, due to [Sorkine et al. 2004], to preserve surface details and overall surface shape and appearance. This energy minimizes the change in laplacians.

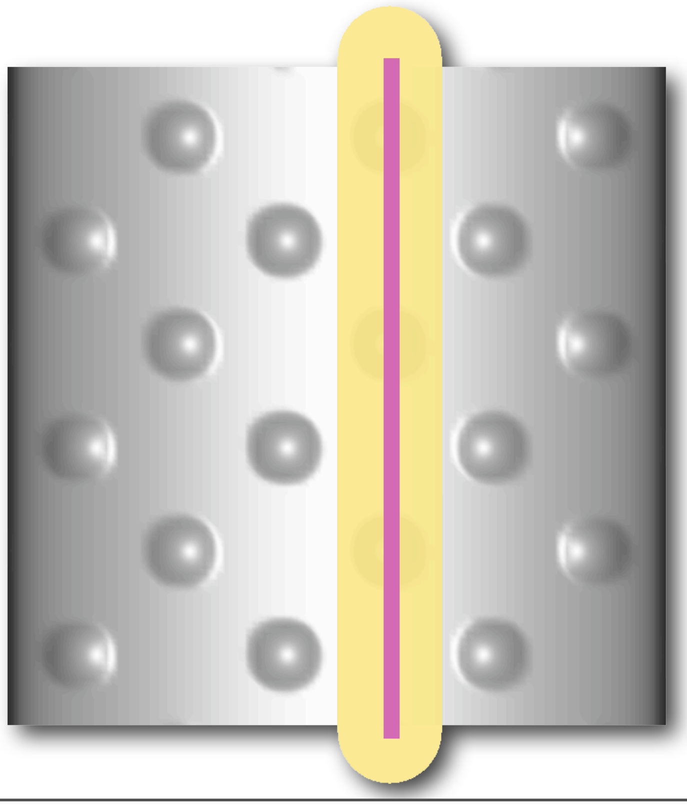
-- click --
Underneath the stroke but away from the centerline (shown in yellow), we vary vertex weights in the Laplacian Editing Energy. We do this to implement the brush attributes width and sharpness.
Details about our variable vertex weights are in the paper.

-- click --
Note that prescribed rotations are linear constraints, and laplacian editing energy is quadratic. So, our energy is quadratic, has a unique minimum, and a closed form solution to reach it --- solving a sparse linear system of equations. This is why our approach is faster than Shape-from-Shading and stable.

Approach

Centerline of stroke: rotate surface about the stroke

- stable, predictable



This brings us to our approach.

After each stroke, we solve for the new surface.

-- click --
Along the centerline of the stroke (shown in purple), we rotate the surface about the projected stroke as the axis, to obtain the desired lighting.
This is stable and predictable.
We also place image-plane position constraints along the centerline to keep the surface there from moving.

-- click --
Elsewhere, we use Laplacian Editing Energy, due to [Sorkine et al. 2004], to preserve surface details and overall surface shape and appearance.
This energy minimizes the change in laplacians.

-- click --
Underneath the stroke but away from the centerline (shown in yellow), we vary vertex weights in the Laplacian Editing Energy. We do this to implement the brush attributes width and sharpness.
Details about our variable vertex weights are in the paper.

-- click --
Note that prescribed rotations are linear constraints, and laplacian editing energy is quadratic. So, our energy is quadratic, has a unique minimum, and a closed form solution to reach it --- solving a sparse linear system of equations. This is why our approach is faster than Shape-from-Shading and stable.

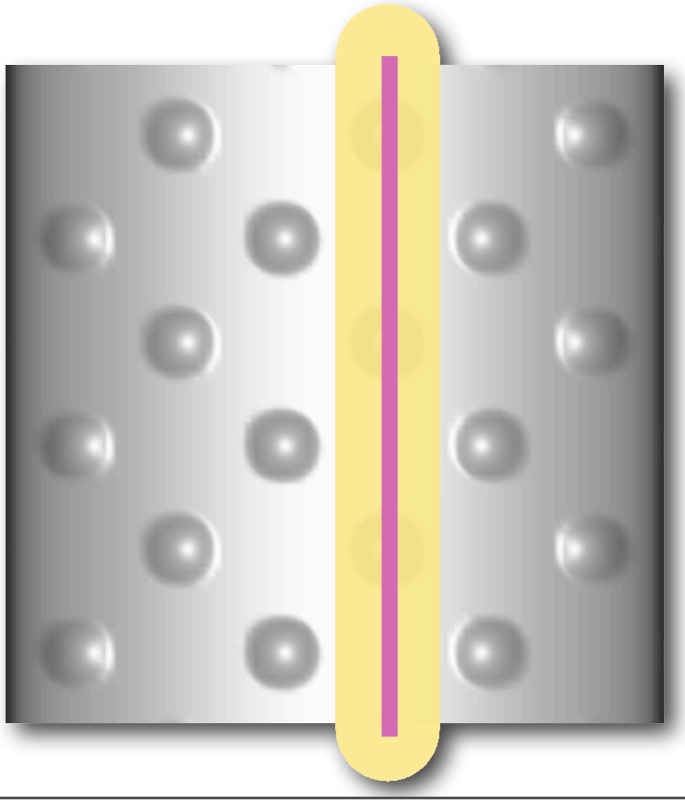
Approach

Centerline of stroke: rotate surface about the stroke

- stable, predictable

Elsewhere: Laplacian Editing Energy

- preserves appearance & shape



This brings us to our approach.

After each stroke, we solve for the new surface.

-- click --
Along the centerline of the stroke (shown in purple), we rotate the surface about the projected stroke as the axis, to obtain the desired lighting. This is stable and predictable.
We also place image-plane position constraints along the centerline to keep the surface there from moving.

-- click --
Elsewhere, we use Laplacian Editing Energy, due to [Sorkine et al. 2004], to preserve surface details and overall surface shape and appearance. This energy minimizes the change in laplacians.

-- click --
Underneath the stroke but away from the centerline (shown in yellow), we vary vertex weights in the Laplacian Editing Energy. We do this to implement the brush attributes width and sharpness.
Details about our variable vertex weights are in the paper.

-- click --
Note that prescribed rotations are linear constraints, and laplacian editing energy is quadratic. So, our energy is quadratic, has a unique minimum, and a closed form solution to reach it --- solving a sparse linear system of equations. This is why our approach is faster than Shape-from-Shading and stable.

Approach

Centerline of stroke: rotate surface about the stroke

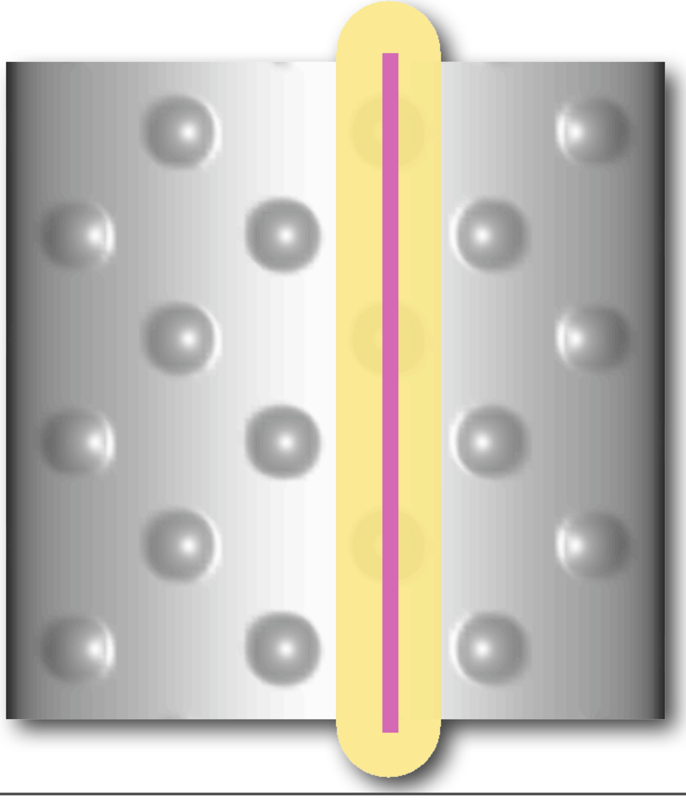
- stable, predictable

Elsewhere: Laplacian Editing Energy

- preserves appearance & shape

Variable vertex weights for our brush parameters

- controllable



This brings us to our approach.

After each stroke, we solve for the new surface.

-- click --
Along the centerline of the stroke (shown in purple), we rotate the surface about the projected stroke as the axis, to obtain the desired lighting. This is stable and predictable.
We also place image-plane position constraints along the centerline to keep the surface there from moving.

-- click --
Elsewhere, we use Laplacian Editing Energy, due to [Sorkine et al. 2004], to preserve surface details and overall surface shape and appearance. This energy minimizes the change in laplacians.

-- click --
Underneath the stroke but away from the centerline (shown in yellow), we vary vertex weights in the Laplacian Editing Energy. We do this to implement the brush attributes width and sharpness.
Details about our variable vertex weights are in the paper.

-- click --
Note that prescribed rotations are linear constraints, and laplacian editing energy is quadratic. So, our energy is quadratic, has a unique minimum, and a closed form solution to reach it --- solving a sparse linear system of equations. This is why our approach is faster than Shape-from-Shading and stable.

Approach

Centerline of stroke: rotate surface about the stroke

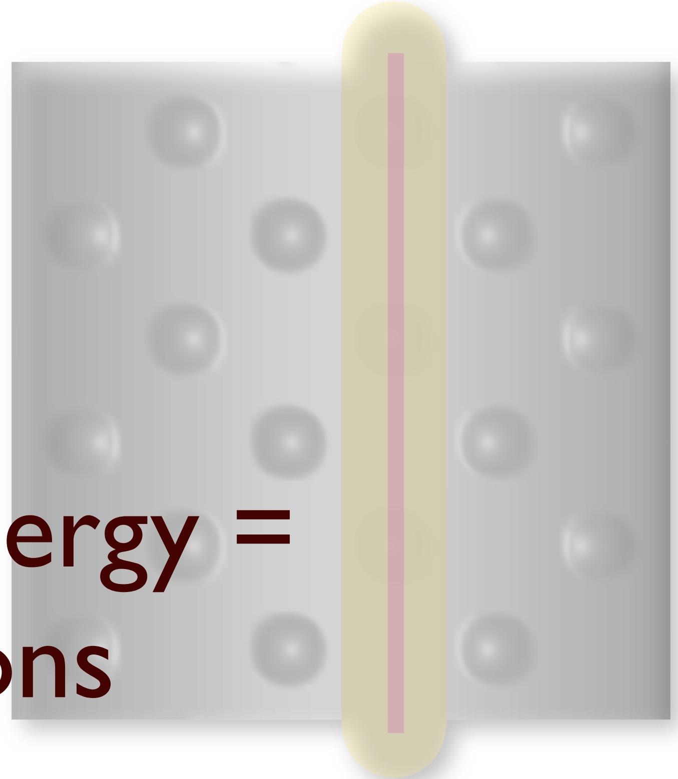
- stable, predictable

Elsewhere: Laplacian Editing Energy

- preserves appearance & shape

Variable vertex weights for our brush parameters

- controllable



Linear Constraints + Quadratic Energy =
sparse linear system of equations

This brings us to our approach.

After each stroke, we solve for the new surface.

-- click --
Along the centerline of the stroke (shown in purple), we rotate the surface about the projected stroke as the axis, to obtain the desired lighting.
This is stable and predictable.
We also place image-plane position constraints along the centerline to keep the surface there from moving.

-- click --
Elsewhere, we use Laplacian Editing Energy, due to [Sorkine et al. 2004], to preserve surface details and overall surface shape and appearance.
This energy minimizes the change in laplacians.

-- click --
Underneath the stroke but away from the centerline (shown in yellow), we vary vertex weights in the Laplacian Editing Energy. We do this to implement the brush attributes width and sharpness.
Details about our variable vertex weights are in the paper.

-- click --
Note that prescribed rotations are linear constraints, and laplacian editing energy is quadratic. So, our energy is quadratic, has a unique minimum, and a closed form solution to reach it --- solving a sparse linear system of equations. This is why our approach is faster than Shape-from-Shading and stable.

Quadratic Energy

vertices
(degrees of freedom)

$$E(V') =$$

I will now introduce the system we build and solve after every stroke. I will illustrate each term step-by-step.

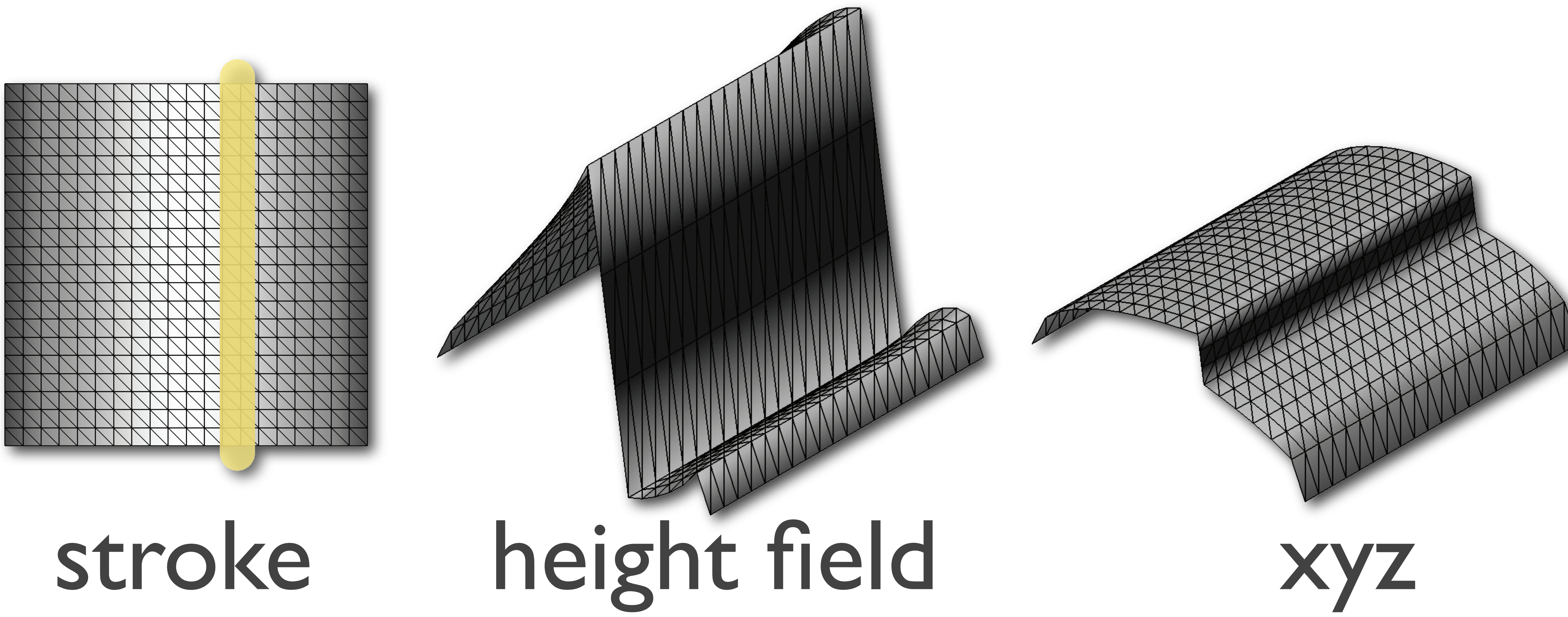
Before we look at the right hand side of our energy, note that our degrees of freedom are vertices, x, y, and z. This is similar to [Ng et al. 2007].
Let us contrast this with a height field approach, where only z is free.

-- click --
Consider a single vertical darkening stroke on a shallow mesh. To achieve the desired darkening, our surface must take on a certain steep slope.
If we solve on a height field, with only z free, our only solution contains a very large steep region.
If we allow all three degrees of freedom to move, we can achieve the desired darkening with a much less distorted surface.

Quadratic Energy

vertices
(degrees of freedom)

$$E(V') =$$



I will now introduce the system we build and solve after every stroke. I will illustrate each term step-by-step.

Before we look at the right hand side of our energy, note that our degrees of freedom are vertices, x, y, and z. This is similar to [Ng et al. 2007].
Let us contrast this with a height field approach, where only z is free.

-- click --
Consider a single vertical darkening stroke on a shallow mesh. To achieve the desired darkening, our surface must take on a certain steep slope.
If we solve on a height field, with only z free, our only solution contains a very large steep region.
If we allow all three degrees of freedom to move, we can achieve the desired darkening with a much less distorted surface.

Quadratic Energy

vertices
(degrees of freedom)

$$E(V') =$$

-- click --
Our energy is the difference in laplacian, squared.

--- click ---
Using the laplacian editing energy gives us good surface and detail preservation. However, unlike the typical usage of difference-in-laplacian energies, we do not allow laplacians to rotate, as we prefer to preserve shaded appearance rather than the geometry itself.

Quadratic Energy

vertices
(degrees of freedom)

$$E(V') = \underbrace{\sum_{i=1}^{n} g(v_i)\|\Delta v_i - \Delta v_i'\|^2}_{\text{difference in laplacian, squared}}$$

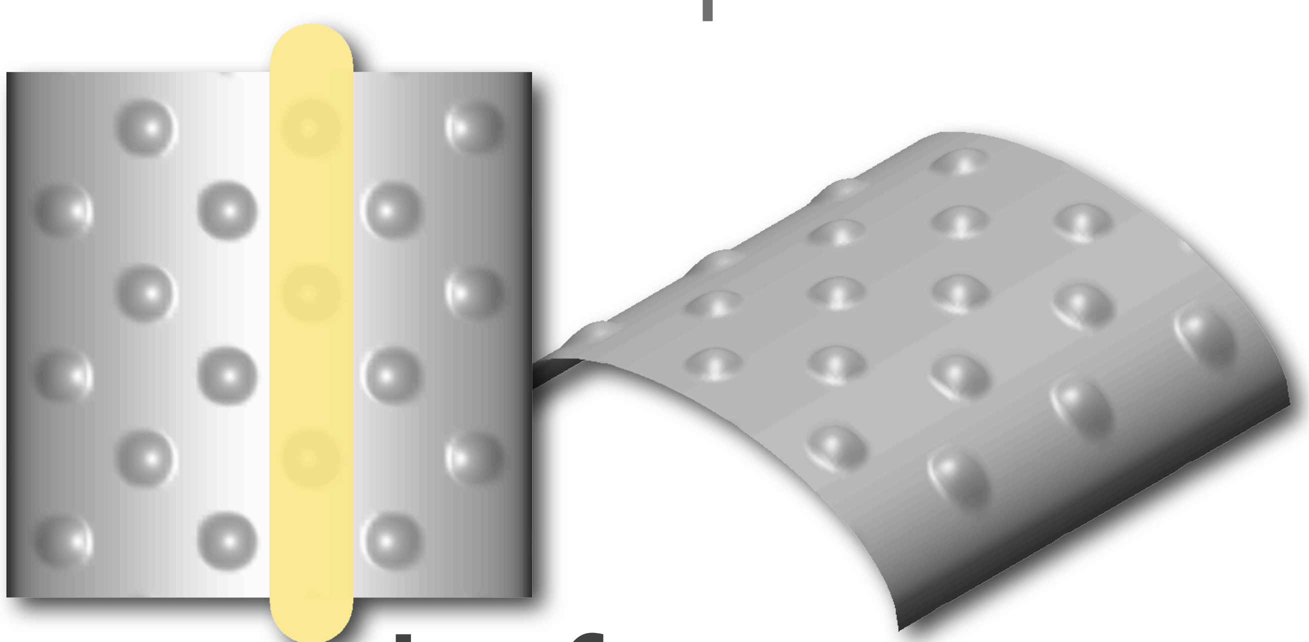
-- click --
Our energy is the difference in laplacian, squared.

--- click ---
Using the laplacian editing energy gives us good surface and detail preservation. However, unlike the typical usage of difference-in-laplacian energies, we do not allow laplacians to rotate, as we prefer to preserve shaded appearance rather than the geometry itself.

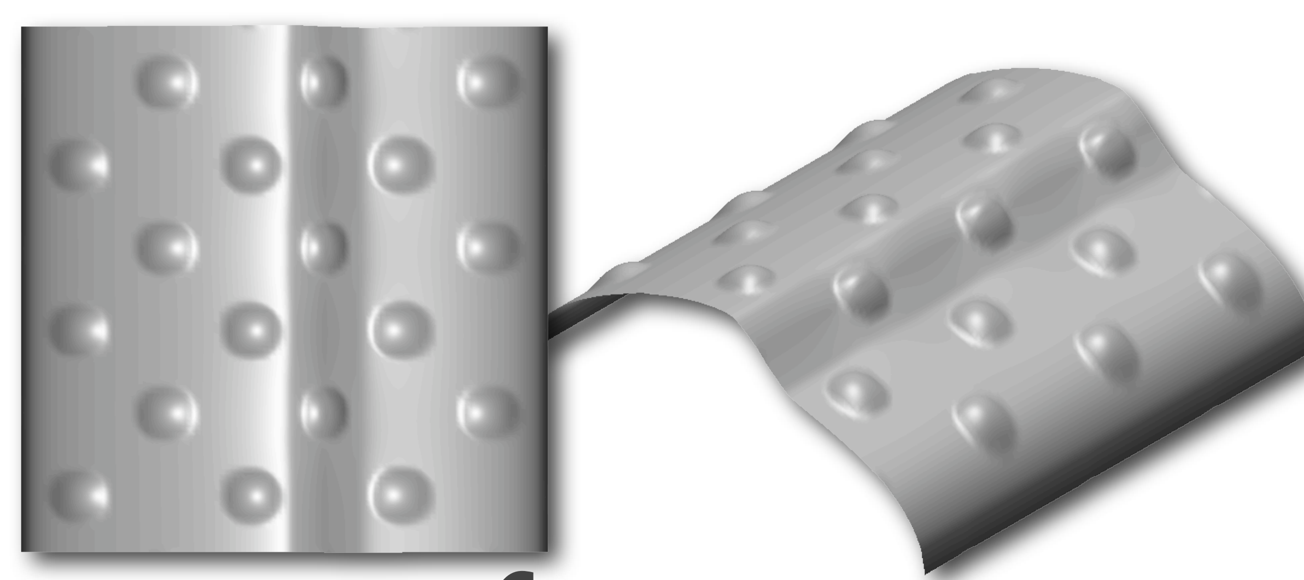
Quadratic Energy

vertices
(degrees of freedom)

$$E(V') = \underbrace{\sum_{i=1}^{n} g(v_i)\|\Delta v_i - \Delta v_i'\|^2}_{\text{difference in laplacian, squared}}$$



before



after

-- click --
Our energy is the difference in laplacian, squared.

--- click ---
Using the laplacian editing energy gives us good surface and detail preservation. However, unlike the typical usage of difference-in-laplacian energies, we do not allow laplacians to rotate, as we prefer to preserve shaded appearance rather than the geometry itself.

Quadratic Energy

vertices
(degrees of freedom)

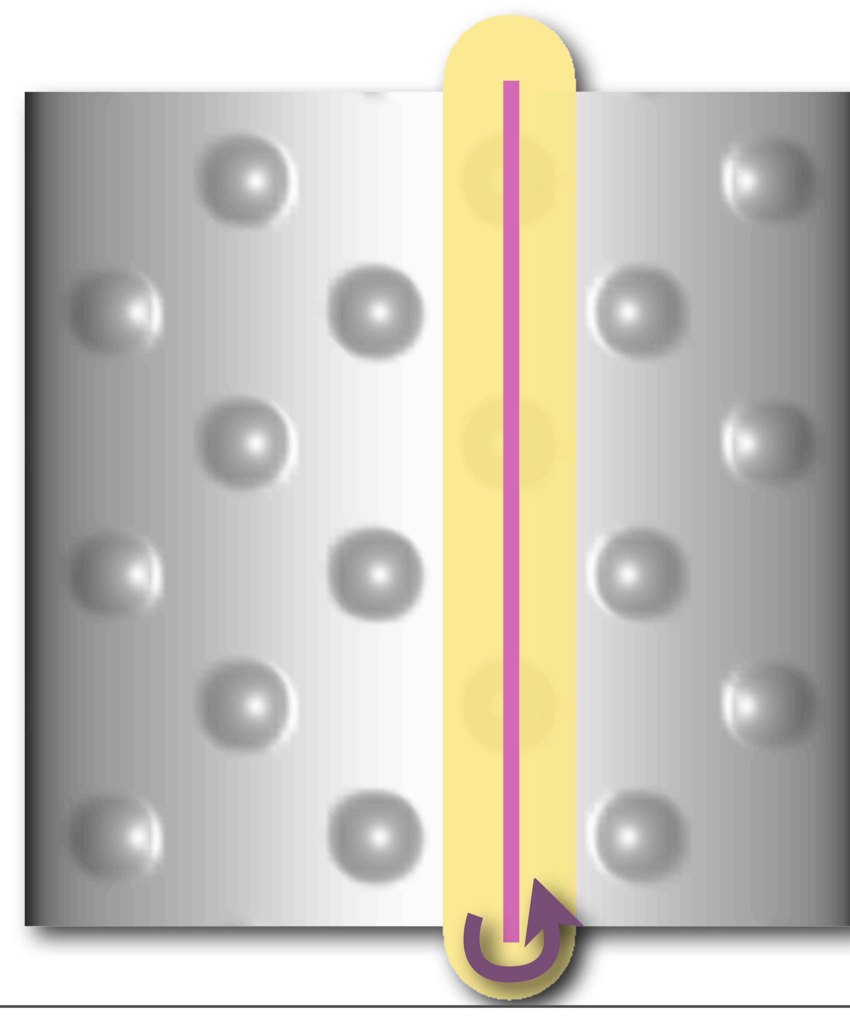
$$E(V') = \underbrace{\sum_{i=1}^{n} g(v_i) \|\Delta v_i - \Delta v_i'\|^2}_{\text{difference in laplacian, squared}}$$

-- click --
We use least squares constraints to rotate the surface under the stroke.
-- click --
Edges intersected by the stroke are rotated about the projected stroke as their axis.
Note that rotating edges leads to linear constraints, while rotating normals does not.

Quadratic Energy

vertices
(degrees of freedom)

$$E(V') = \underbrace{\sum_{i=1}^{n} g(v_i)\|\Delta v_i - \Delta v_i'\|^2}_{\text{difference in laplacian, squared}} + \overbrace{w_{lsq} \sum_{(i,j)=e\in C} \|(v_i' - v_j') - e^{trg}\|^2}^{\text{constraint rotating the surface under the stroke}}$$



-- click --
We use least squares constraints to rotate the surface under the stroke.
-- click --
Edges intersected by the stroke are rotated about the projected stroke as their axis.
Note that rotating edges leads to linear constraints, while rotating normals does not.

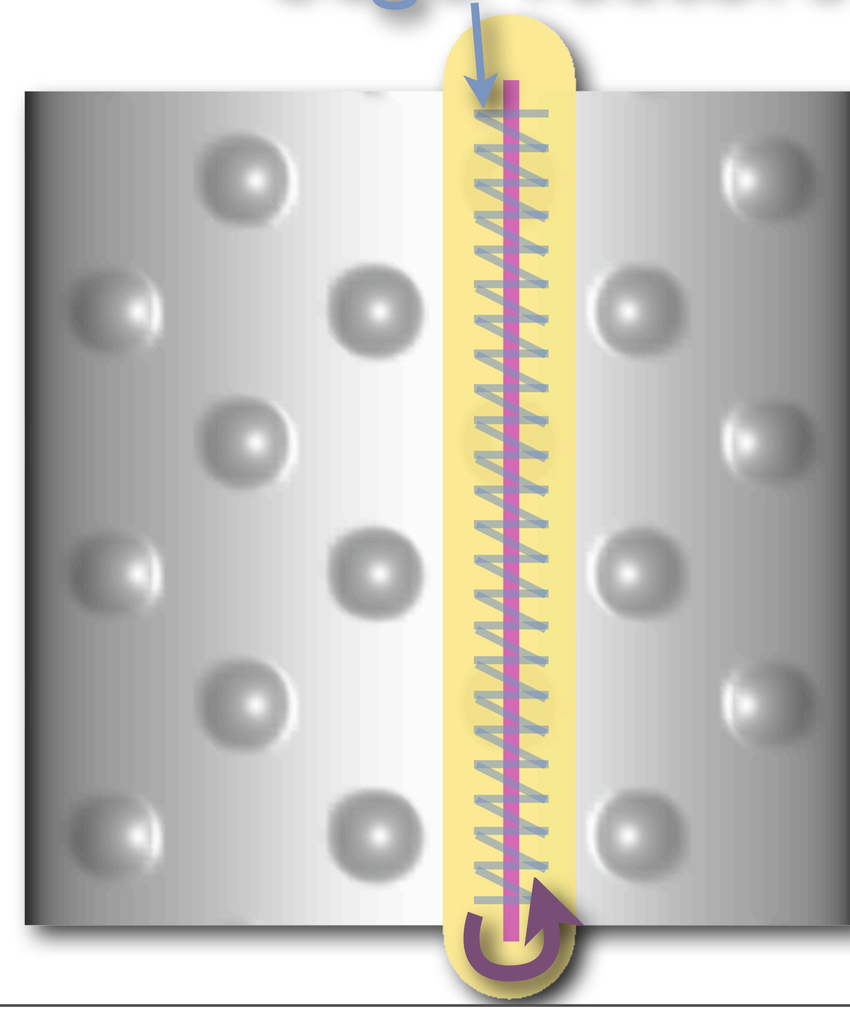
Quadratic Energy

vertices (degrees of freedom)

constraint rotating the surface under the stroke

$$E(V') = \underbrace{\sum_{i=1}^{n} g(v_i)\|\Delta v_i - \Delta v_i'\|^2}_{\text{difference in laplacian, squared}} + \overbrace{w_{lsq} \sum_{(i,j)=e\in C} \|(v_i' - v_j') - e^{trg}\|^2}^{\text{constraint rotating the surface under the stroke}}$$

edge vectors



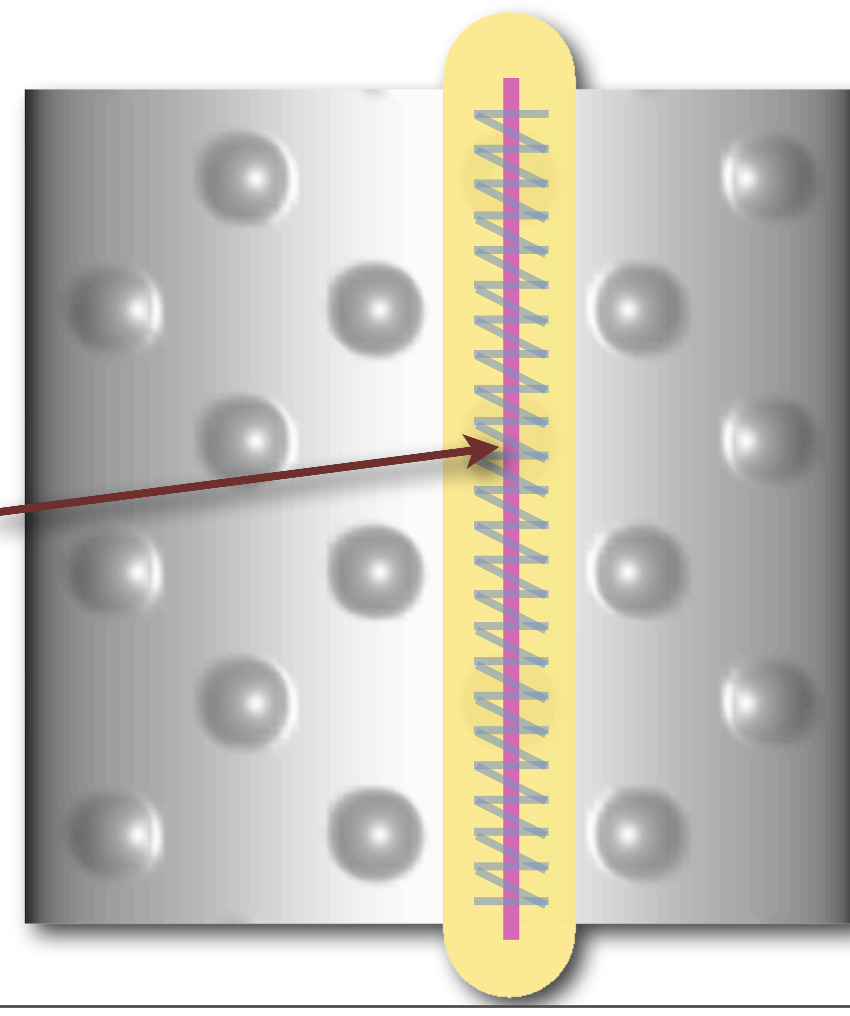
-- click --
We use least squares constraints to rotate the surface under the stroke.
-- click --
Edges intersected by the stroke are rotated about the projected stroke as their axis.
Note that rotating edges leads to linear constraints, while rotating normals does not.

Quadratic Energy

vertices
(degrees of freedom)

$$E(V') = \underbrace{\sum_{i=1}^{n} g(v_i)\|\Delta v_i - \Delta v_i'\|^2}_{\text{difference in laplacian, squared}} + \overbrace{w_{lsq} \sum_{(i,j)=e\in C} \|(v_i' - v_j') - e^{trg}\|^2}^{\text{constraint rotating the surface under the stroke}}$$

image-plane constraints
(not shown in E)



Finally, we also place image-plane positional constraints underneath the stroke.

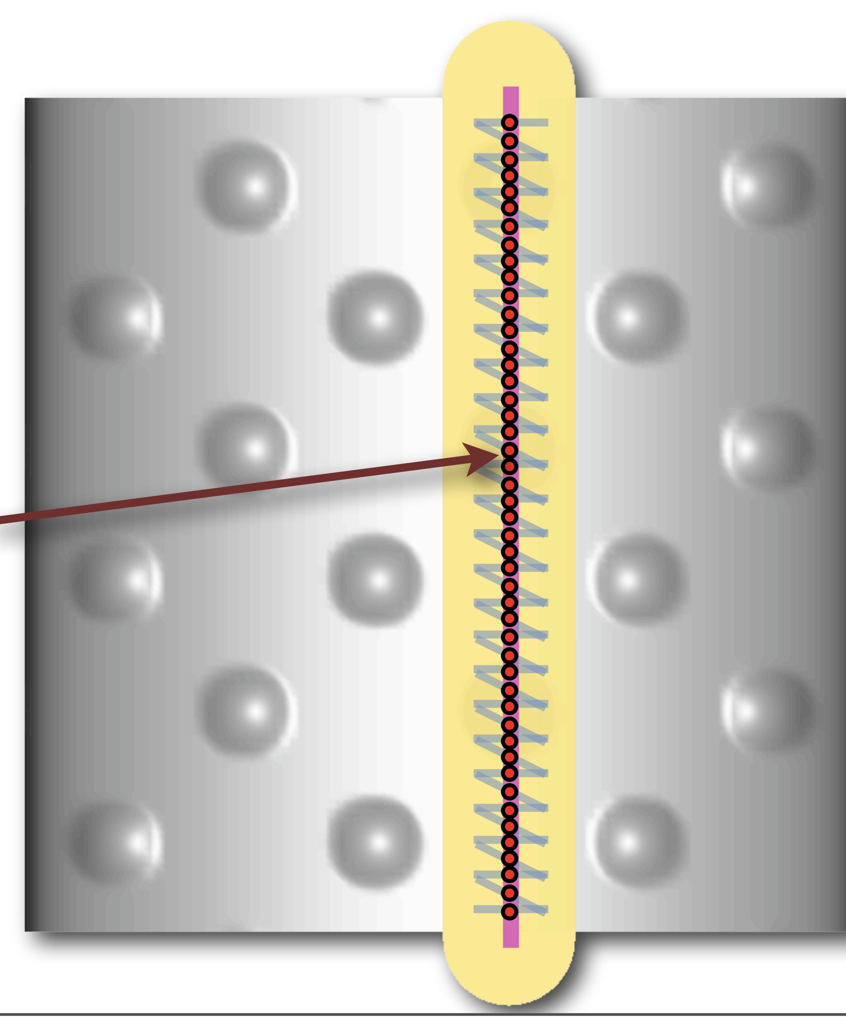
--- refuse ---
Note that our boundaries are vertices near silhouettes.
and 3D position constraints at silhouettes, which are our boundaries.

Quadratic Energy

vertices (degrees of freedom)

$$E(V') = \underbrace{\sum_{i=1}^{n} g(v_i) \|\Delta v_i - \Delta v_i'\|^2}_{\text{difference in laplacian, squared}} + w_{lsq} \overbrace{\sum_{(i,j)=e \in C} \|(v_i' - v_j') - e^{trg}\|^2}^{\text{constraint rotating the surface under the stroke}}$$

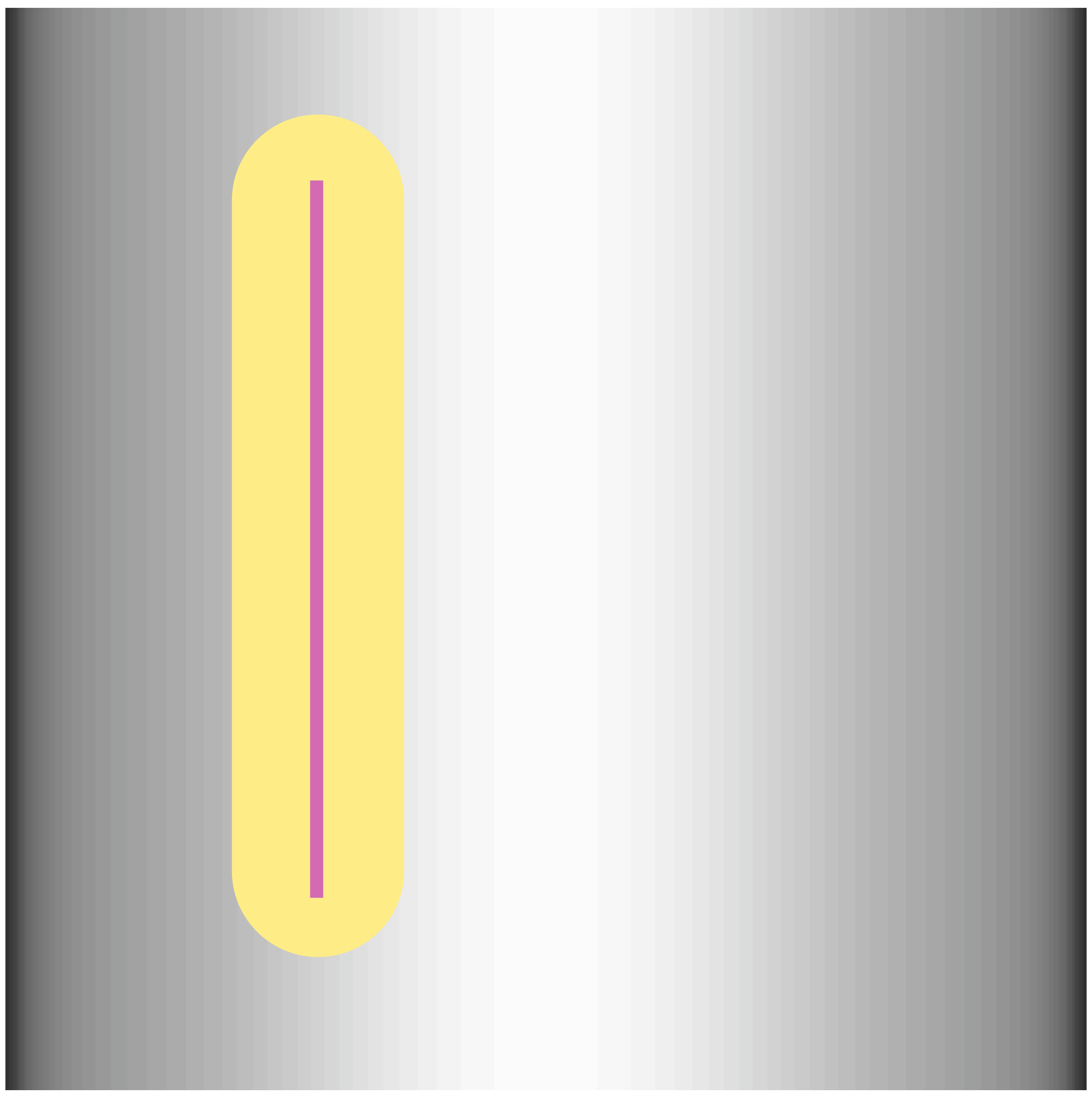
image-plane constraints (not shown in E)



Finally, we also place image-plane positional constraints underneath the stroke.

--- refuse ---
Note that our boundaries are vertices near silhouettes.
and 3D position constraints at silhouettes, which are our boundaries.

Region of Interest

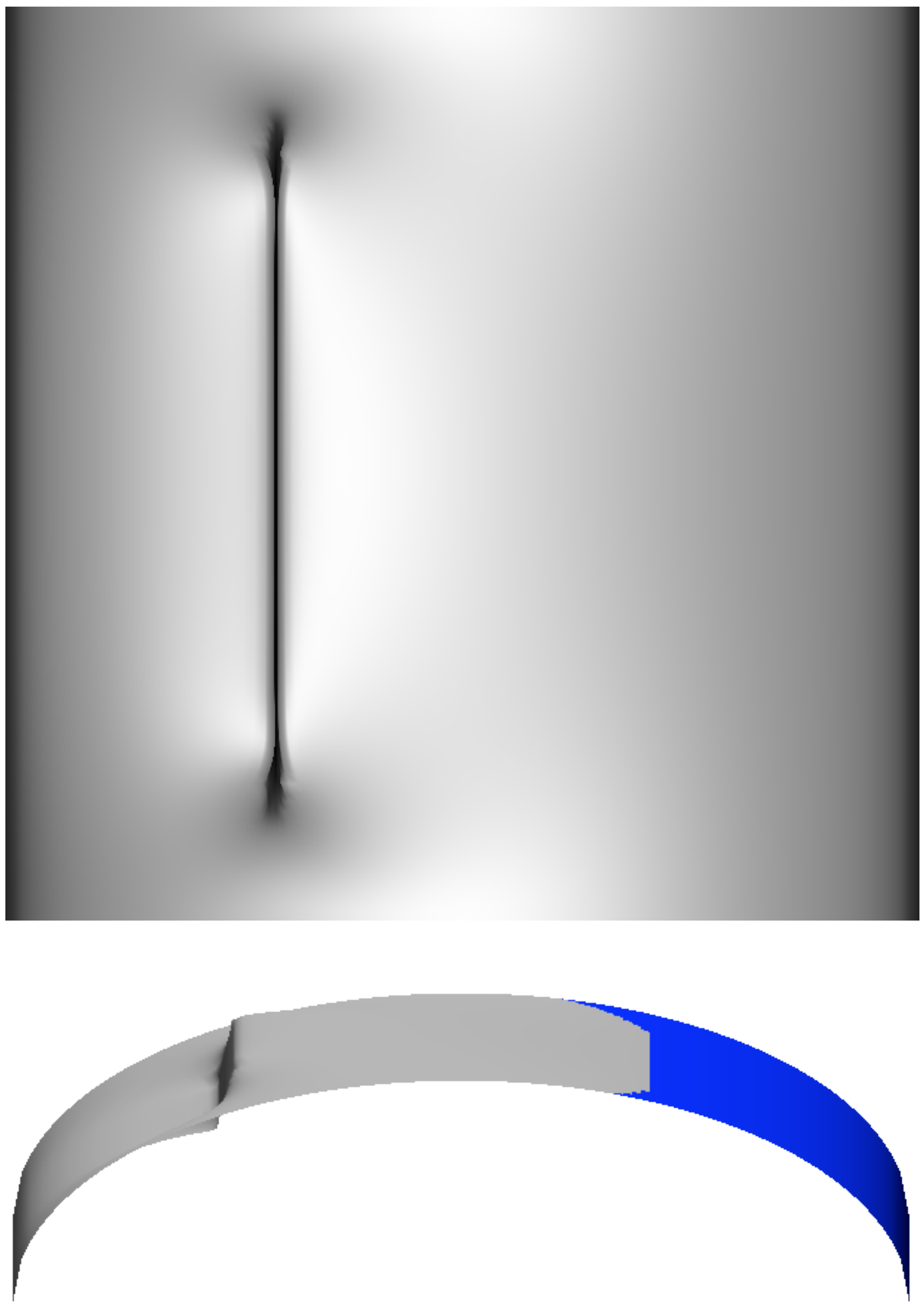


Although changes are global, the user can still limit the effect of changes by freezing parts of the mesh.
-- click --
In this diagram, the blue region of mesh is frozen and unaffected by the stroke.

When the changeable area of mesh is too small,
-- click --
a distinct bright area appears around the darkening stroke.

Freezing parts of the mesh decreases the system size, which improves performance. By freezing the mesh sufficiently far from the stroke, we improve performance without introducing noticeable artifacts.

Region of Interest

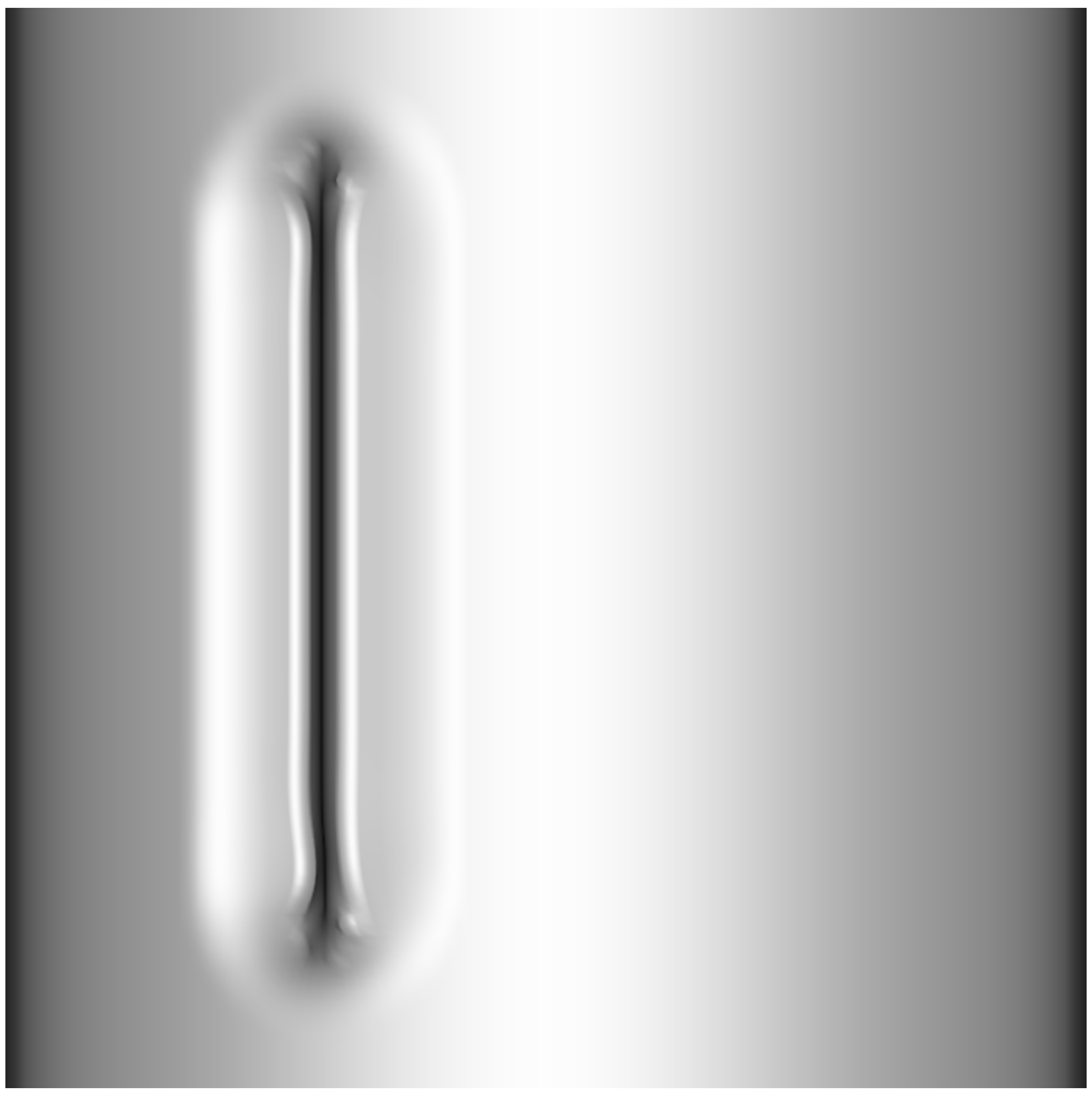


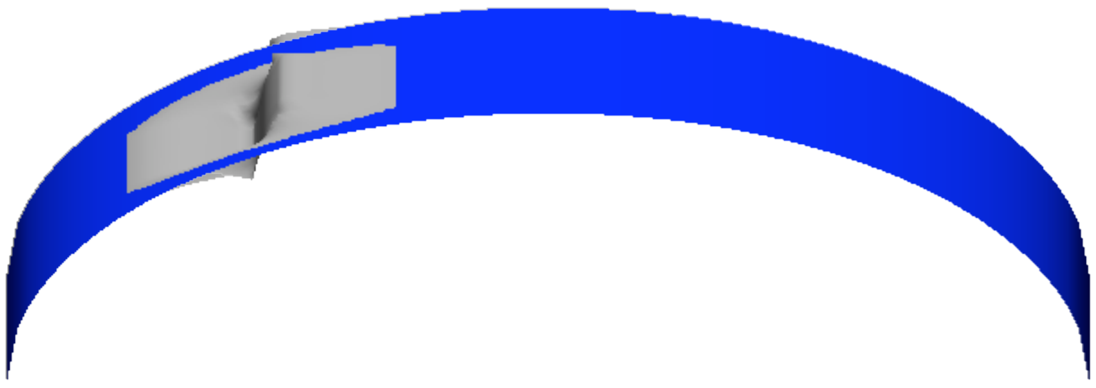
Although changes are global, the user can still limit the effect of changes by freezing parts of the mesh.
-- click --
In this diagram, the blue region of mesh is frozen and unaffected by the stroke.

When the changeable area of mesh is too small,
-- click --
a distinct bright area appears around the darkening stroke.

Freezing parts of the mesh decreases the system size, which improves performance. By freezing the mesh sufficiently far from the stroke, we improve performance without introducing noticeable artifacts.

Region of Interest



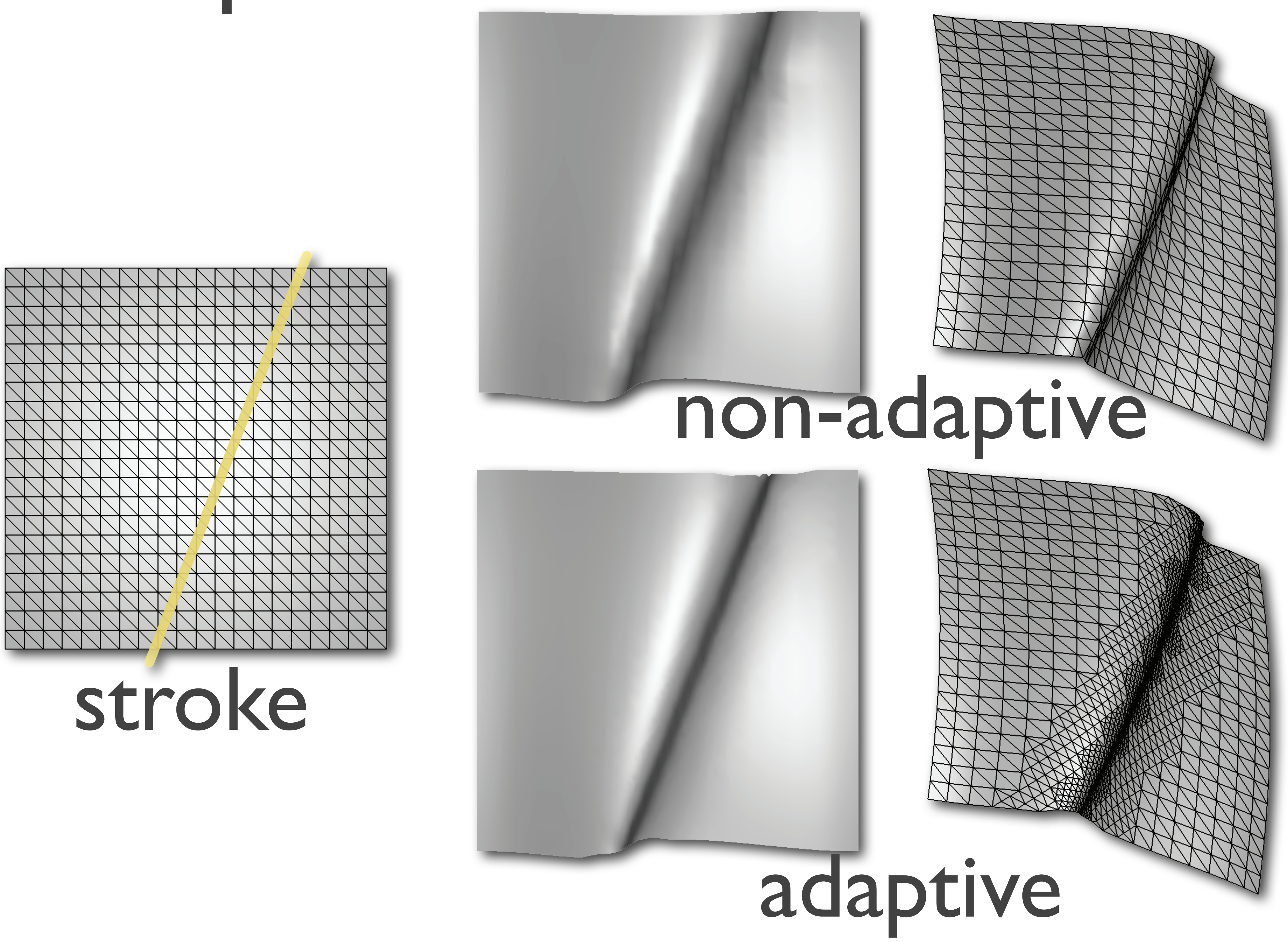


Although changes are global, the user can still limit the effect of changes by freezing parts of the mesh.
-- click --
In this diagram, the blue region of mesh is frozen and unaffected by the stroke.

When the changeable area of mesh is too small,
-- click --
a distinct bright area appears around the darkening stroke.

Freezing parts of the mesh decreases the system size, which improves performance. By freezing the mesh sufficiently far from the stroke, we improve performance without introducing noticeable artifacts.

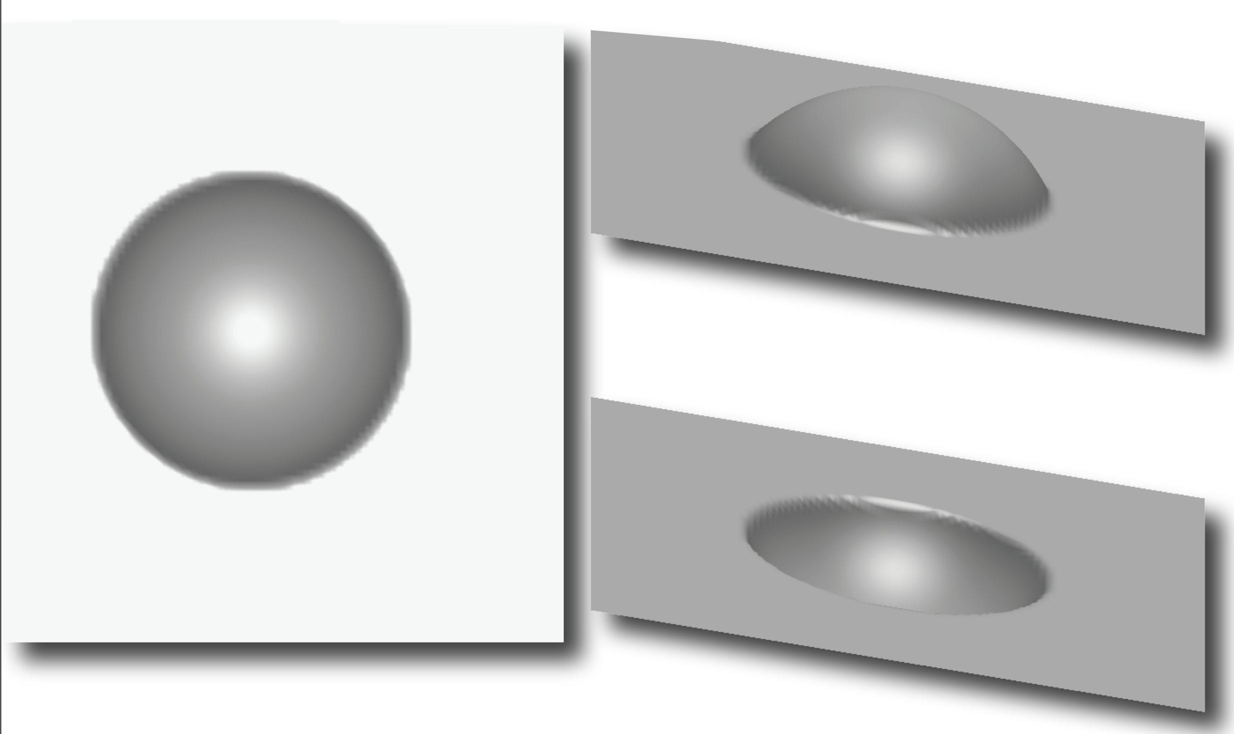
Adaptive Subdivision

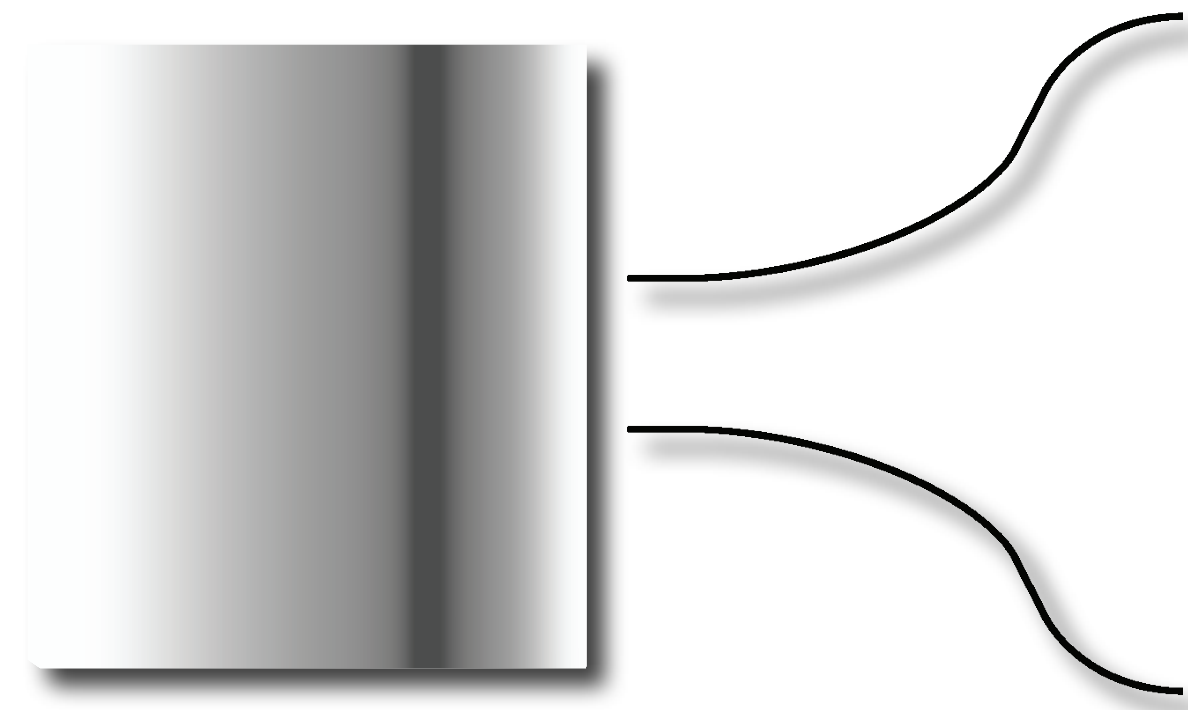


Adaptive subdivision lets us add fine detail to low resolution meshes.
Note the aliasing artifacts in the non-adaptive mesh.

This feature also improves performance, as additional degrees of freedom are only introduced where needed.

Flip Ambiguity





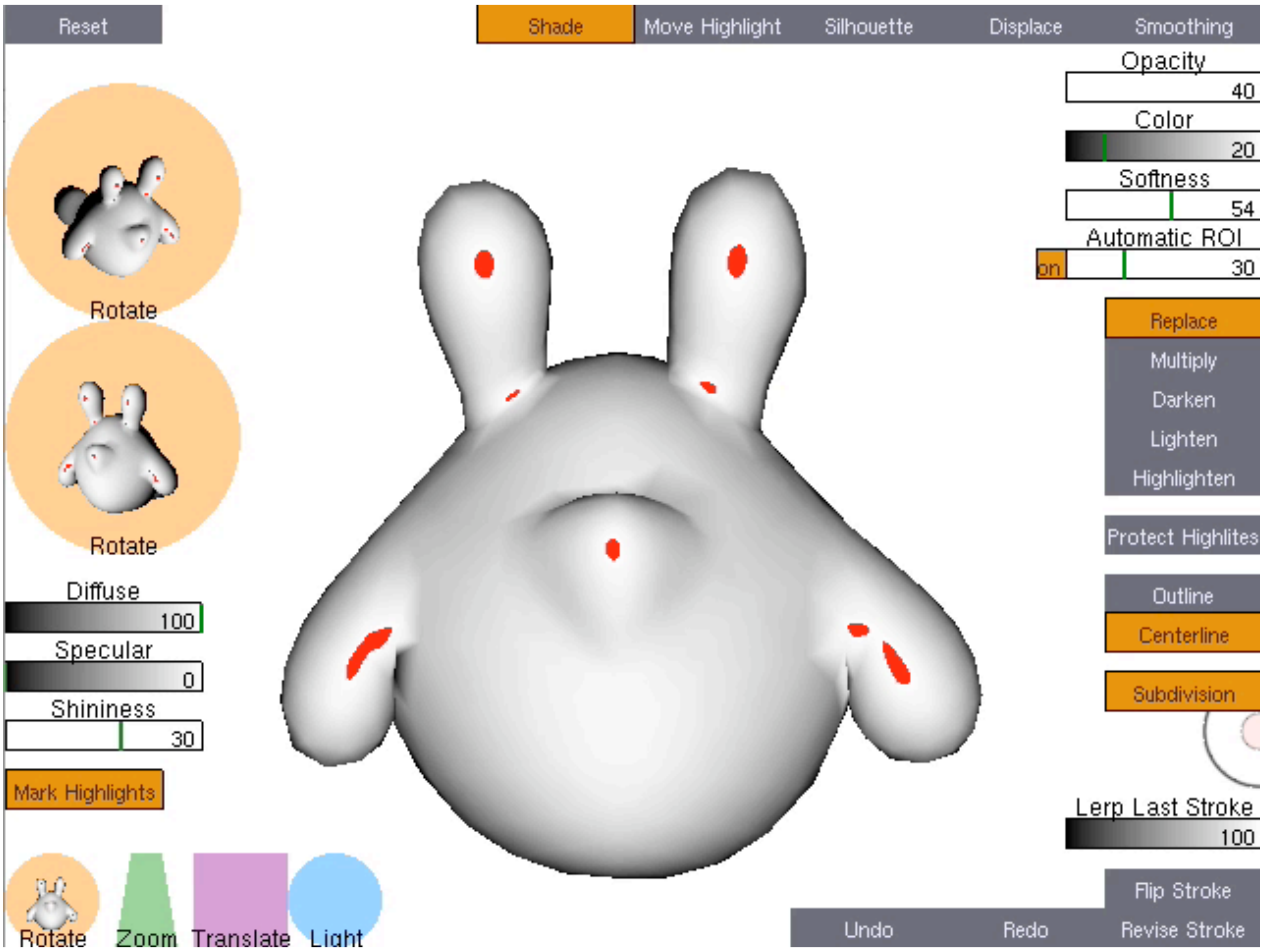
Finally, there is an inherent ambiguity in shape-from-shading, and we provide a tool for this.

This ambiguity is often known as the concave/convex ambiguity.
A hemisphere viewed from above looks the same whether it is concave or convex.
Similarly, a sloping square looks the same whether it slopes towards or away from the viewer.

When adding cavities to the cartoon animal’s ears [and for the first stroke on the mannequin’s cheeks] the user was confronted with this ambiguity.

Because we are modifying an existing surface, by default we resolve the ambiguity for a given stroke by choosing the slope which modifies the surface the least. The user can override this choice after drawing a stroke by clicking the “flip” button. This is only a binary ambiguity, and the default is usually correct.

Summary



-- no click --
Here we see a reprise of the cartoon example shown earlier, with decorations.
Highlights are shown in red.
After each stroke, the centerline rotation constraints appear as normals in purple.
Frozen mesh vertices are shown in blue.

Note the adaptive subdivision performed on the initially coarse mesh.

For the ear cavities, the orientation of the stroke is reversed using the flip button.

Limitations and Future Work

I will conclude with Limitations and Future Work.

-- click --
The ability to shade requires artistic expertise. This is not a system for novices, although it may be a useful learning tool for artists learning to shade.

-- click --
Because shading modifications induce global modifications, we need to solve a system of equations on a region-of-interest a good deal larger than the stroke. Sculpting, on the other hand, is a local operation: only a small area under or nearly under the stroke is modified.

-- click --
Our tools are useful for certain operations, but sometimes users just want to create a bump or a ridge, and our tool is the wrong tool for that. We should integrate with a sculpting tool so that users can use the most appropriate tool for a given operation. Also, integration with a silhouette editing system would allow users to edit silhouettes created using our silhouette tool.

-- click --
Finally, we can move and reshape highlights if the user is careful, but care must be taken not to try to darken them with a shading stroke, as this is impossible. [This is discussed in the paper.] We have no tool for explicitly merging or deleting highlights.

Limitations and Future Work

Shading requires expertise.

I will conclude with Limitations and Future Work.

-- click --
The ability to shade requires artistic expertise. This is not a system for novices, although it may be a useful learning tool for artists learning to shade.

-- click --
Because shading modifications induce global modifications, we need to solve a system of equations on a region-of-interest a good deal larger than the stroke. Sculpting, on the other hand, is a local operation: only a small area under or nearly under the stroke is modified.

-- click --
Our tools are useful for certain operations, but sometimes users just want to create a bump or a ridge, and our tool is the wrong tool for that. We should integrate with a sculpting tool so that users can use the most appropriate tool for a given operation. Also, integration with a silhouette editing system would allow users to edit silhouettes created using our silhouette tool.

-- click --
Finally, we can move and reshape highlights if the user is careful, but care must be taken not to try to darken them with a shading stroke, as this is impossible. [This is discussed in the paper.] We have no tool for explicitly merging or deleting highlights.

Limitations and Future Work

Shading requires expertise.

Speed: modifications aren’t local.

I will conclude with Limitations and Future Work.

-- click --
The ability to shade requires artistic expertise. This is not a system for novices, although it may be a useful learning tool for artists learning to shade.

-- click --
Because shading modifications induce global modifications, we need to solve a system of equations on a region-of-interest a good deal larger than the stroke. Sculpting, on the other hand, is a local operation: only a small area under or nearly under the stroke is modified.

-- click --
Our tools are useful for certain operations, but sometimes users just want to create a bump or a ridge, and our tool is the wrong tool for that. We should integrate with a sculpting tool so that users can use the most appropriate tool for a given operation. Also, integration with a silhouette editing system would allow users to edit silhouettes created using our silhouette tool.

-- click --
Finally, we can move and reshape highlights if the user is careful, but care must be taken not to try to darken them with a shading stroke, as this is impossible. [This is discussed in the paper.] We have no tool for explicitly merging or deleting highlights.

Limitations and Future Work

Shading requires expertise.

Speed: modifications aren't local.

Lack of integration with sculpting tools and silhouette editing.

I will conclude with Limitations and Future Work.

-- click --
The ability to shade requires artistic expertise. This is not a system for novices, although it may be a useful learning tool for artists learning to shade.

-- click --
Because shading modifications induce global modifications, we need to solve a system of equations on a region-of-interest a good deal larger than the stroke. Sculpting, on the other hand, is a local operation: only a small area under or nearly under the stroke is modified.

-- click --
Our tools are useful for certain operations, but sometimes users just want to create a bump or a ridge, and our tool is the wrong tool for that. We should integrate with a sculpting tool so that users can use the most appropriate tool for a given operation. Also, integration with a silhouette editing system would allow users to edit silhouettes created using our silhouette tool.

-- click --
Finally, we can move and reshape highlights if the user is careful, but care must be taken not to try to darken them with a shading stroke, as this is impossible. [This is discussed in the paper.] We have no tool for explicitly merging or deleting highlights.

Limitations and Future Work

Shading requires expertise.

Speed: modifications aren't local.

Lack of integration with sculpting tools and silhouette editing.

Highlight control is limited.

I will conclude with Limitations and Future Work.

-- click --
The ability to shade requires artistic expertise. This is not a system for novices, although it may be a useful learning tool for artists learning to shade.

-- click --
Because shading modifications induce global modifications, we need to solve a system of equations on a region-of-interest a good deal larger than the stroke. Sculpting, on the other hand, is a local operation: only a small area under or nearly under the stroke is modified.

-- click --
Our tools are useful for certain operations, but sometimes users just want to create a bump or a ridge, and our tool is the wrong tool for that. We should integrate with a sculpting tool so that users can use the most appropriate tool for a given operation. Also, integration with a silhouette editing system would allow users to edit silhouettes created using our silhouette tool.

-- click --
Finally, we can move and reshape highlights if the user is careful, but care must be taken not to try to darken them with a shading stroke, as this is impossible. [This is discussed in the paper.] We have no tool for explicitly merging or deleting highlights.

Acknowledgements

NYU Computer Science colleagues for their help.

The anonymous reviewers for their suggestions.

Zach Shukan for his artwork.

gingold@cs.nyu.edu

[Finally, I would like to thank, [1], [2], and [3].]

Thank you, and I will be glad to take questions.

End