

Imagine Day

続・レンダリスト養成講座

田村 尚希
川瀬 正樹

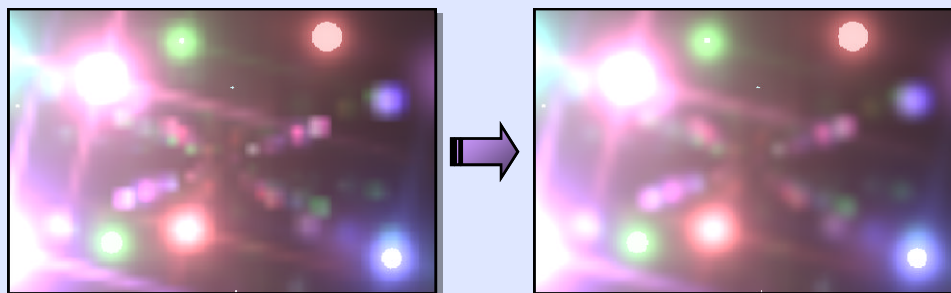
シリコンスタジオ株式会社

本セッションの流れ

前半発表者：田村



後半発表者：川瀬



発表の流れ

1. 発表の概要
2. 論文紹介

「Line Drawing via Abstracted Shading」



3. 発表のまとめ

発表の流れ

1. 発表の概要

2. 論文紹介

「Line Drawing via Abstracted Shading」



3. 発表のまとめ

1.1 発表の概要

- 紹介論文:

Line Drawing via Abstracted Shading [Lee et al. SIGGRAPH 2007]



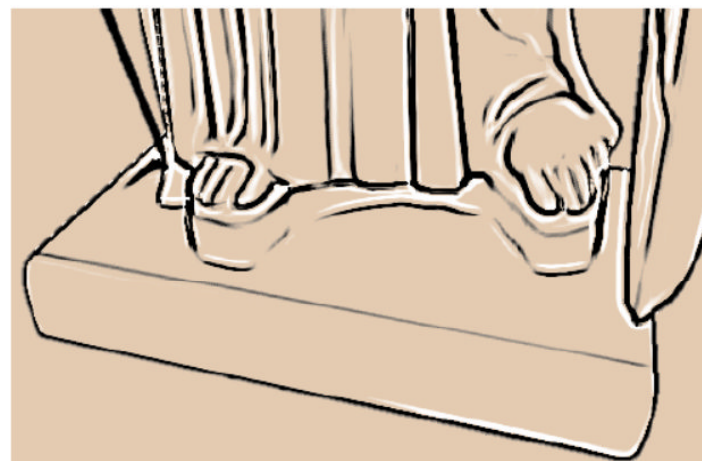
当初、今年のSIGGRAPHのNPRの論文紹介しようと思いましたが、ゲーム向きじゃなかったので止めました。

1.2 手法の概要 (1/3)

- 手法の特徴



入力画像



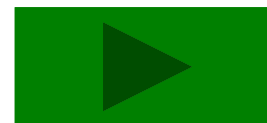
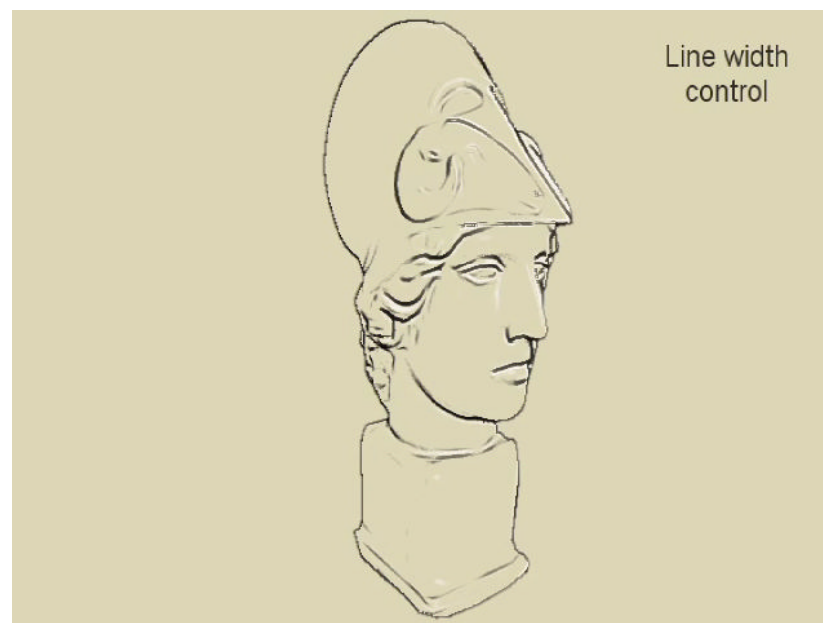
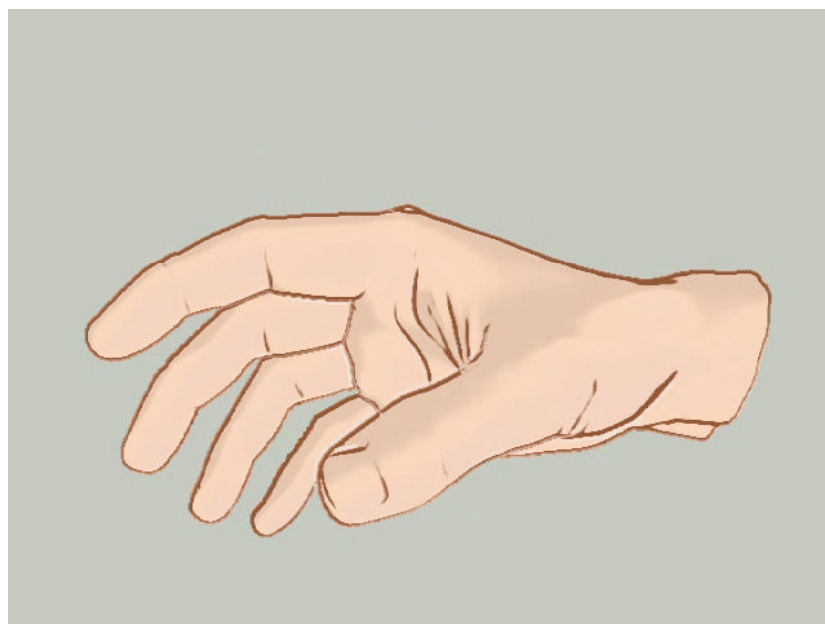
結果画像

- NPRにおいて:

山・谷線、折り目線、輪郭線を統一的に描画できる手法

1.2 手法の概要 (2/3)

- 動画:



1.2 手法の概要 (3/3)

- 手法の長所
 - 完全な2次元処理
 - 2D静止画 / レンダリングした3Dシーン両方対応可
 - メッシュに特殊なデータを仕込んだりは不要
 - ラインの太さは可変
 - 単純なSobel Filterなどでは不可能
 - 大元となるアルゴリズムに応用の余地がある
 - SSAO, Tone Map,
- 手法の短所
 - 少々計算時間がかかる
 - 3Dシーンに適用すると、結果が視点依存
 - エイリアシングが発生する

発表の流れ

1. 発表の概要

2. 論文紹介

「Line Drawing via Abstracted Shading」



3. 発表のまとめ

アルゴリズム全体像

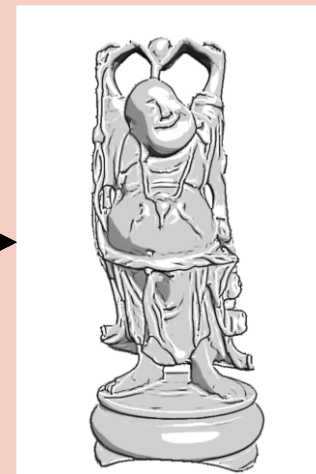
トーンシェーディング



入力メッシュ



トーン
シェーディング

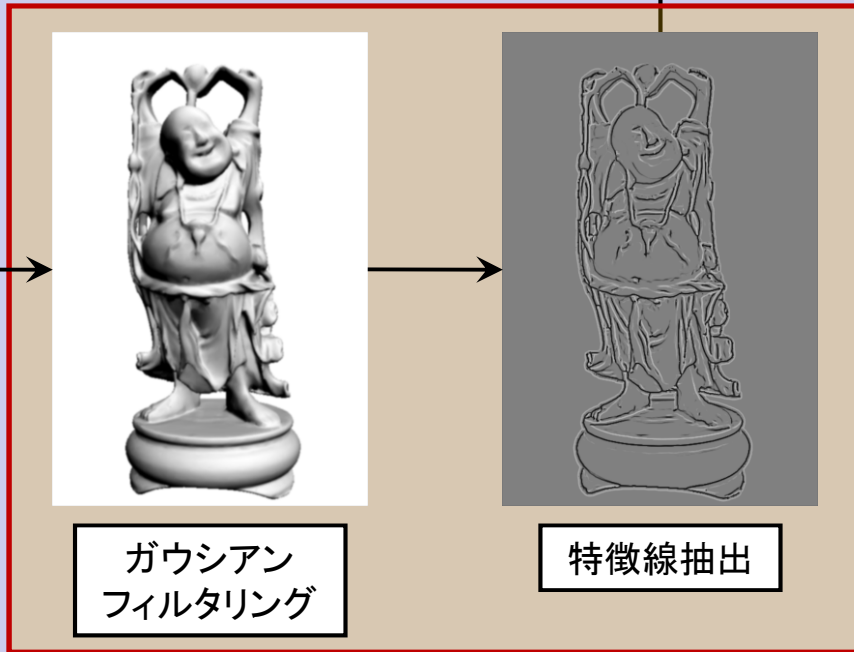


結果画像

特徴線抽出

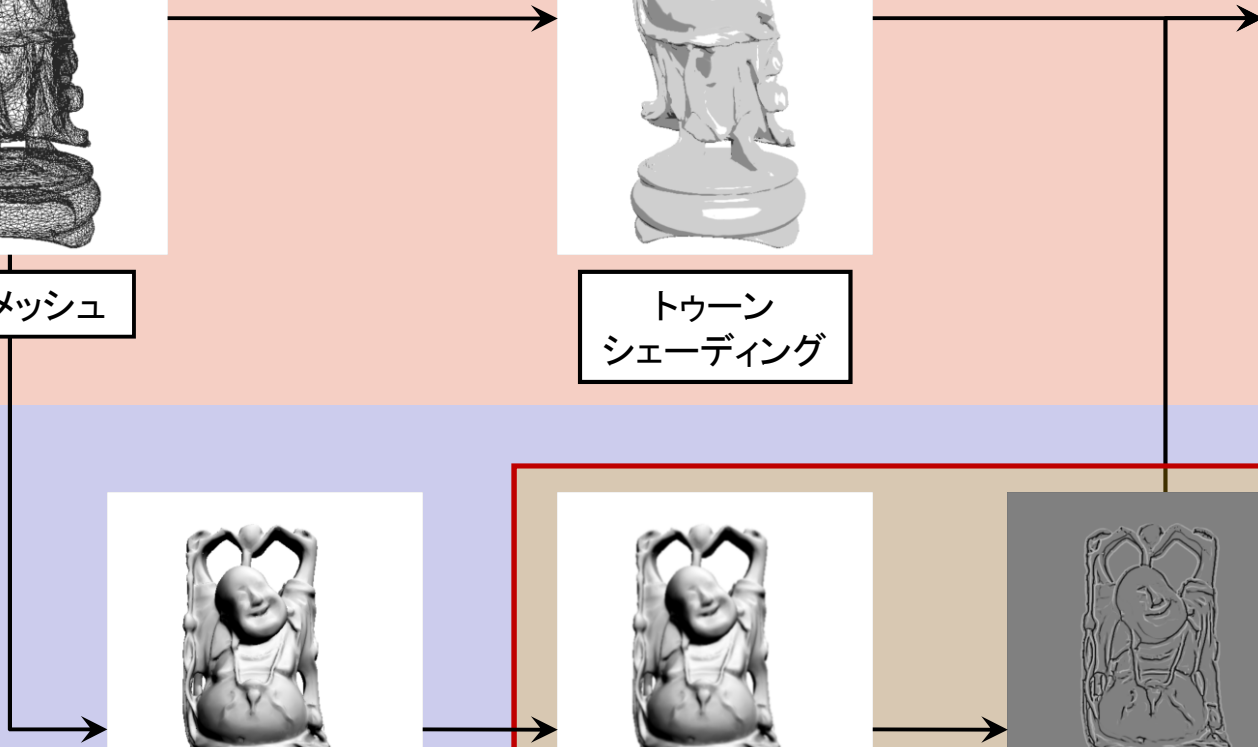


ライティング

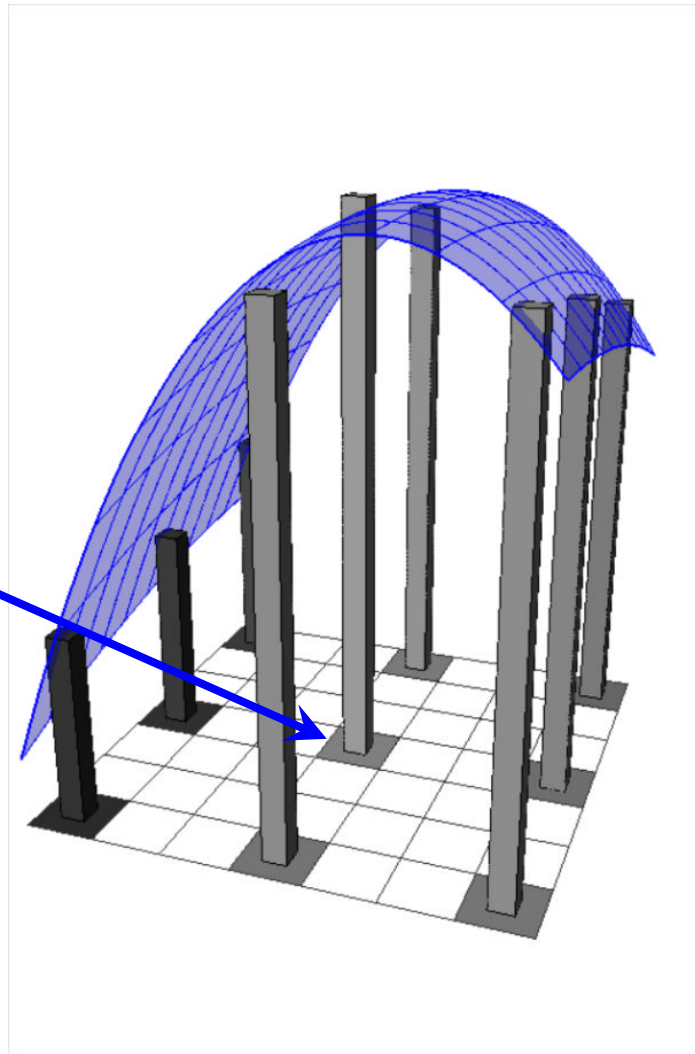


ガウシアン
フィルタリング

特徴線抽出



2.1 キーアイデア (1/3)



手順1:

{(描きたい線の太さ)/2}の間隔で
3×3のサンプル点を配置する

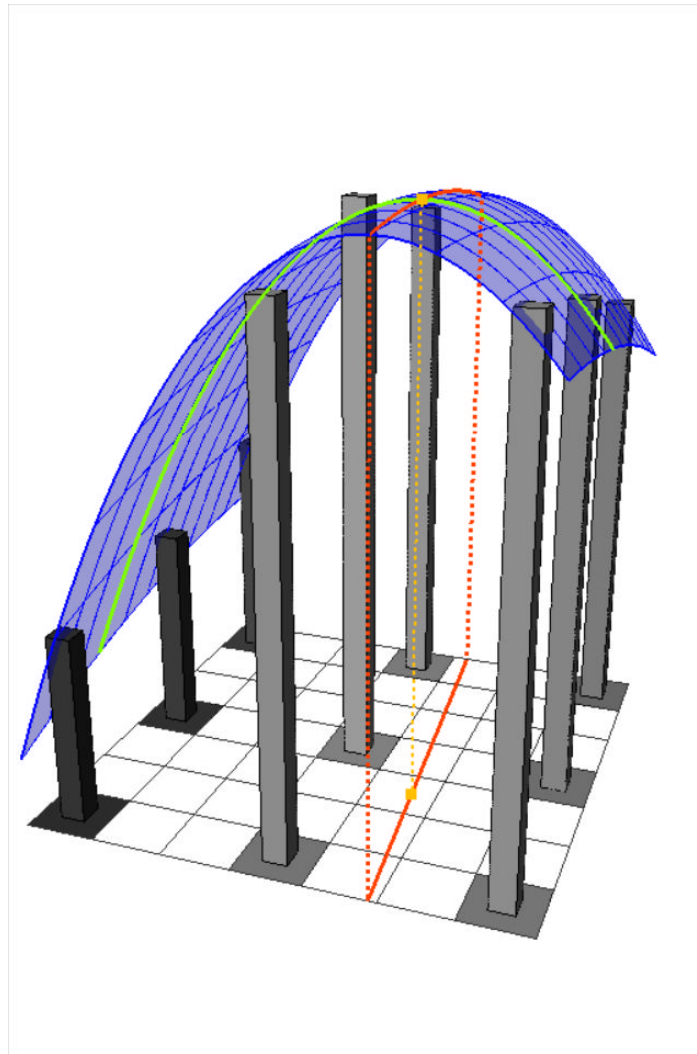
手順2:

色の濃さを高さとしたハイトマップを
構築

手順3:

ハイトマップを放物面で近似

2.1 キーアイデア (2/3)



手順1:

{(描きたい線の太さ)/2}の間隔で
3×3のサンプル点を配置する

手順2:

色の濃さを高さとしたハイトマップを
構築

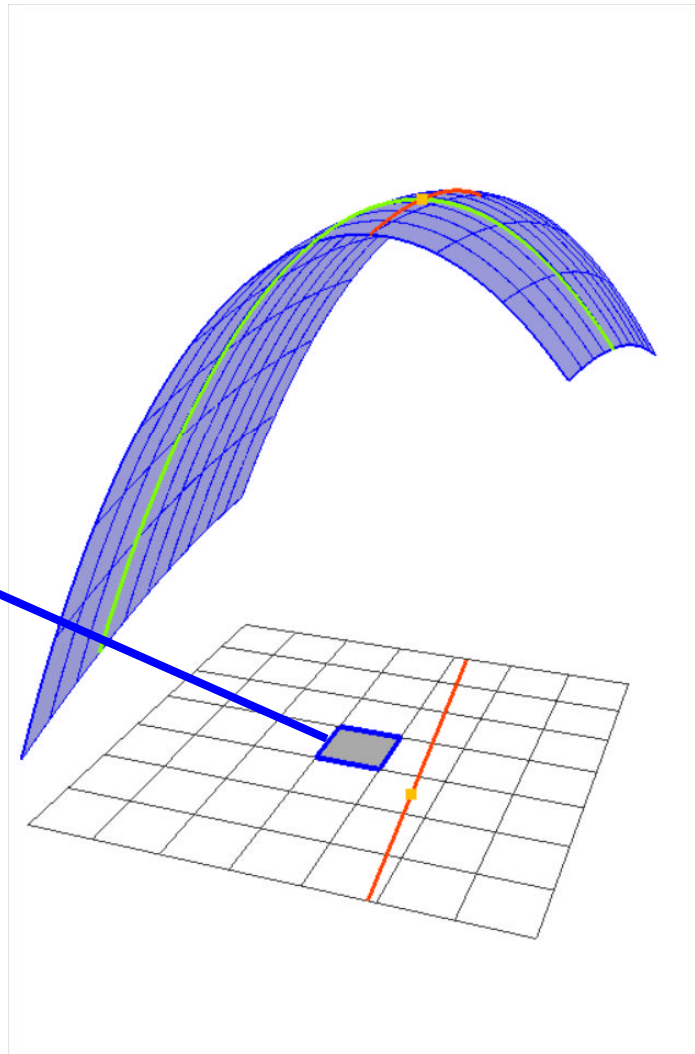
手順3:

ハイトマップを放物面で近似

手順4:

頂点・曲率・曲線方向を計算

2.1 キーアイデア (3/3)



手順1 :

{(描きたい線の太さ)/2}の間隔で
3×3のサンプル点を配置する

手順2 :

色の濃さを高さとしたハイトマップを
構築

手順3 :

ハイトマップを放物面で近似

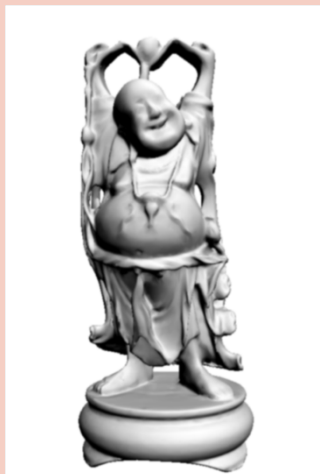
手順4 :

頂点・曲率・曲線方向を計算

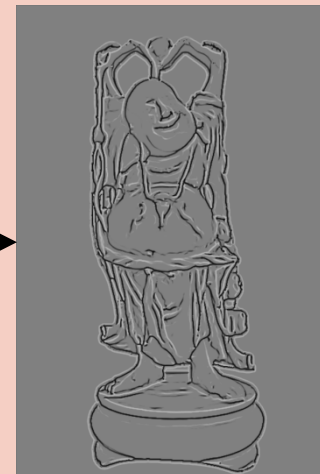
手順5 :

頂点・曲率・曲線方向から色の補正
係数を計算

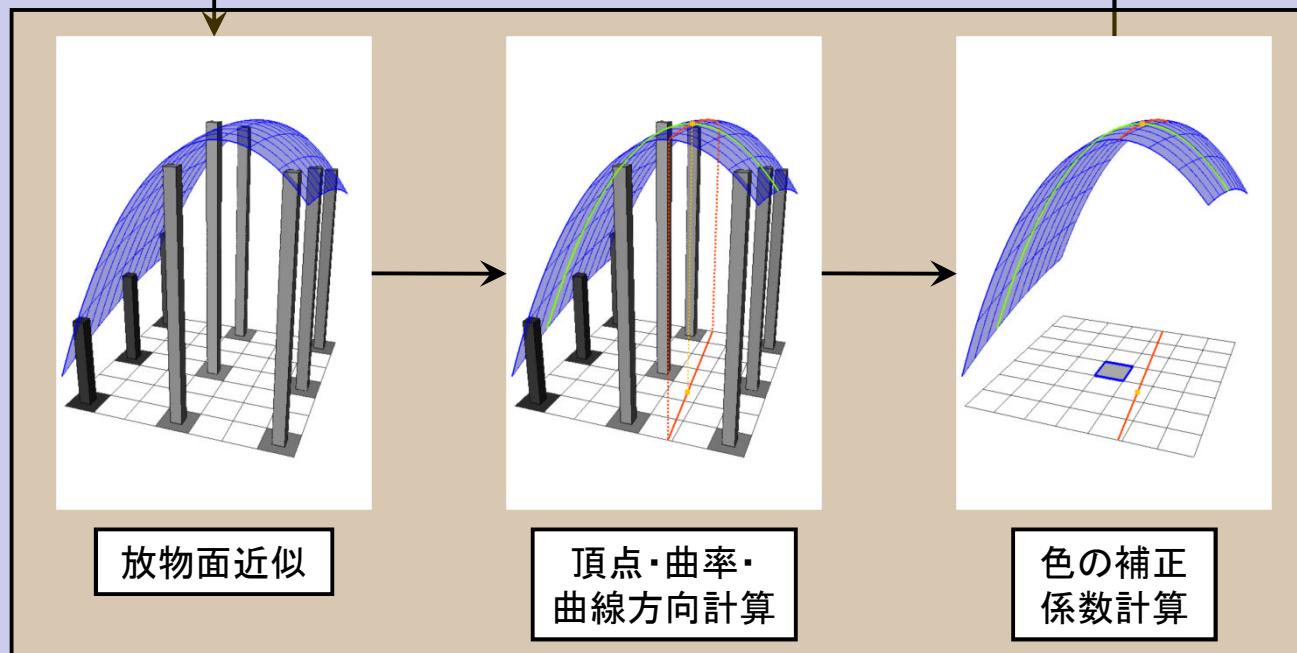
特徴線抽出手順



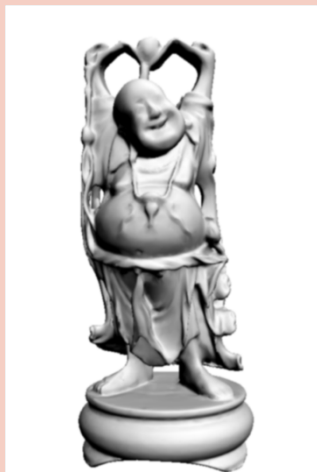
ガウシアン
フィルタリング



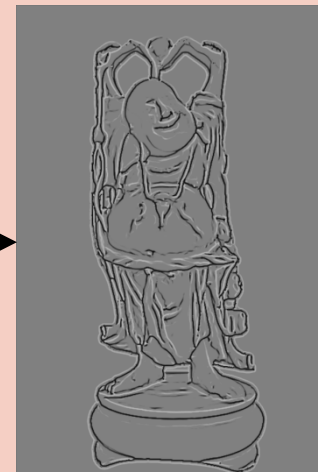
特徴線抽出



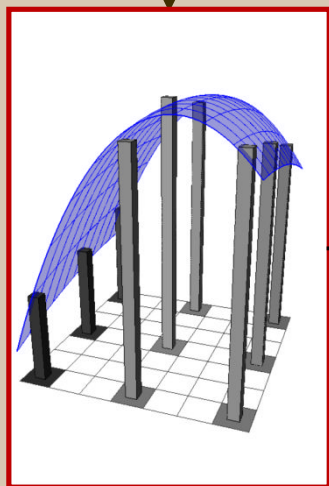
特徴線抽出手順



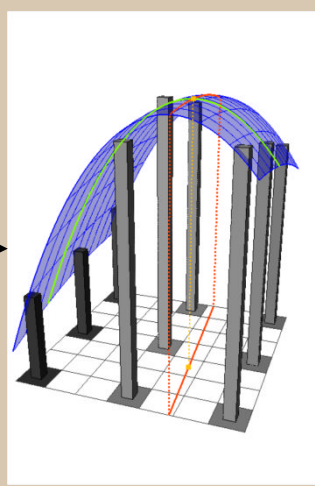
ガウシアン
フィルタリング



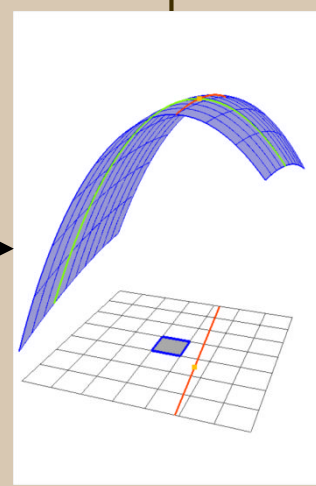
特徴線抽出



放物面近似



頂点・曲率・
曲線方向計算



色の補正
係数計算

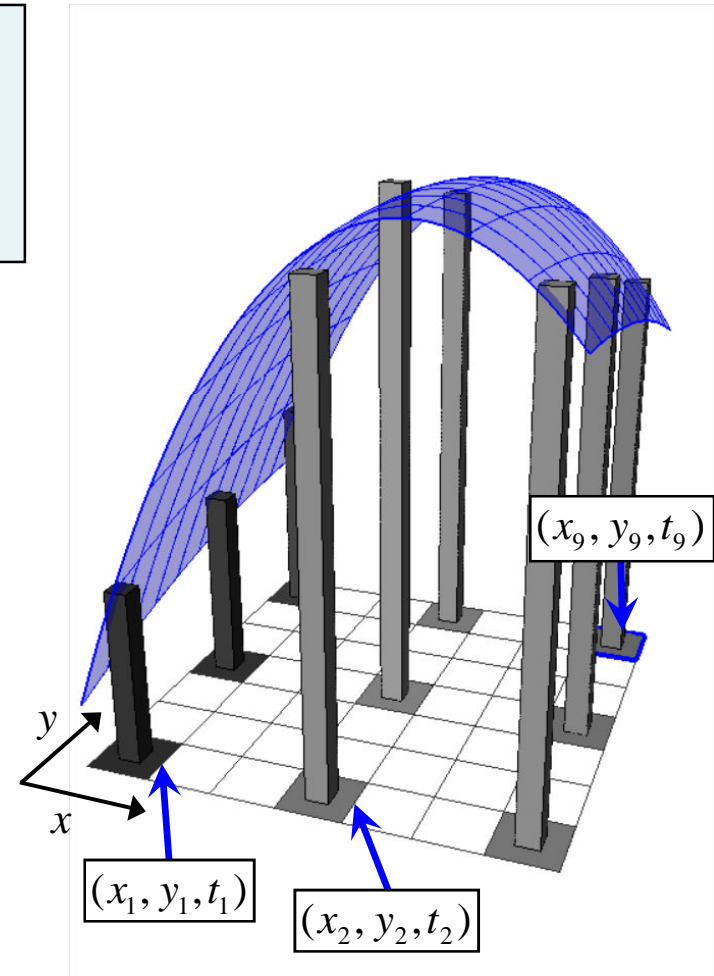
2.2.1 放物面近似 (1/4)

放物線:

$$f(x, y) = a_0x^2 + 2a_1xy + a_2y^2 + a_3x + a_4y + a_5$$

の a_0, a_1, \dots, a_5 を線形システムを用いて求める

$$\Leftrightarrow \begin{bmatrix} x_1^2 & 2x_1y_1 & y_1^2 & x_1 & y_1 & 1 \\ x_2^2 & 2x_2y_2 & y_2^2 & x_2 & y_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_9^2 & 2x_9y_9 & y_9^2 & x_9 & y_9 & 1 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix} = \begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ t_9 \end{bmatrix}$$



2.2.1 放物面近似 (2/4)

放物線:

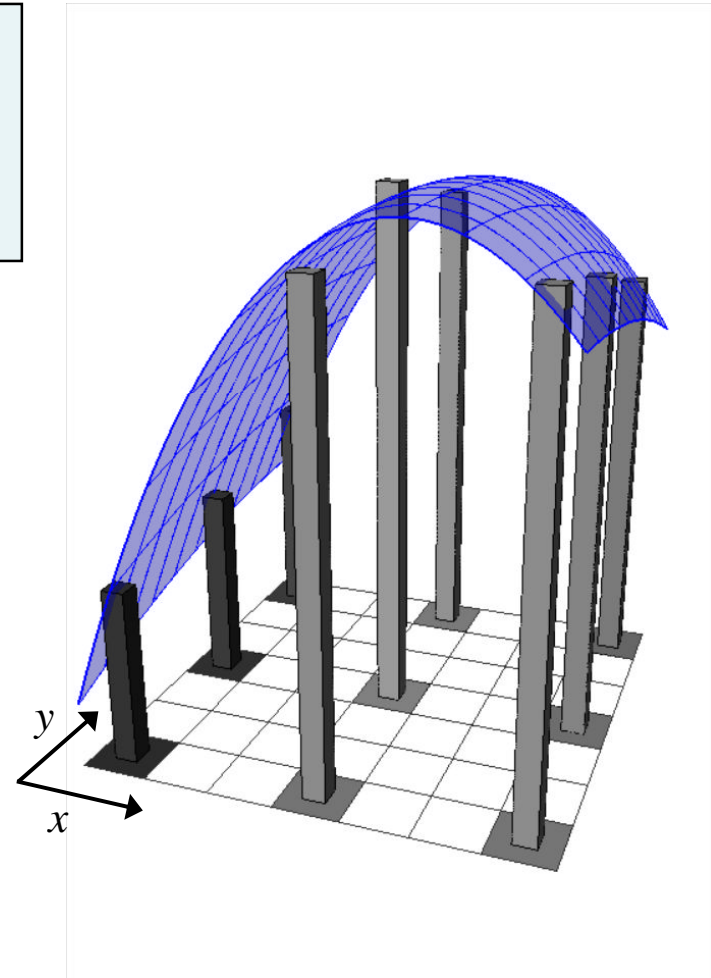
$$f(x, y) = a_0x^2 + 2a_1xy + a_2y^2 + a_3x + a_4y + a_5$$

の a_0, a_1, \dots, a_5 を線形システムを用いて求める

$$\begin{array}{cccccc|c|c}
 x_1^2 & 2x_1y_1 & y_1^2 & x_1 & y_1 & 1 & a_0 & t_1 \\
 x_2^2 & 2x_2y_2 & y_2^2 & x_2 & y_2 & 1 & a_1 & t_2 \\
 & & \vdots & & & & a_2 & \vdots \\
 x_9^2 & 2x_9y_9 & y_9^2 & x_9 & y_9 & 1 & a_3 & t_9 \\
 & & & & & & a_4 & \\
 & & & & & & a_5 &
 \end{array} =
 \begin{array}{c}
 t_1 \\
 t_2 \\
 \vdots \\
 t_9
 \end{array}$$

\Downarrow \Downarrow \Downarrow
 X $A = T$

$XA = T$ という線形システムを解けば良い



2.2.1 放物面近似 (3/4)

放物線:

$$f(x, y) = a_0x^2 + 2a_1xy + a_2y^2 + a_3x + a_4y + a_5$$

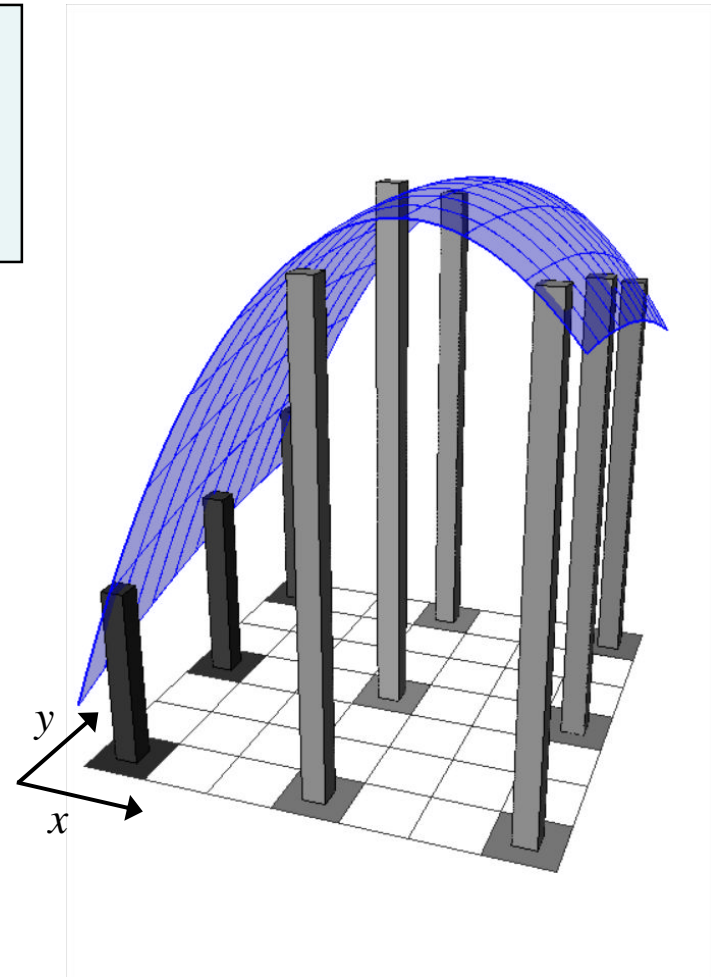
の a_0, a_1, \dots, a_5 を線形システムを用いて求める

$XA = T$ という線形システムを解けば良い

$A = X^{-1}T$ で求められる?

⇒ X は正方行列でないため逆行列は計算不可能

$$X = \begin{bmatrix} x_1^2 & 2x_1y_1 & y_1^2 & x_1 & y_1 & 1 \\ x_2^2 & 2x_2y_2 & y_2^2 & x_2 & y_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_9^2 & 2x_9y_9 & y_9^2 & x_9 & y_9 & 1 \end{bmatrix}$$



2.2.1 放物面近似 (4/4)

放物線:

$$f(x, y) = a_0x^2 + 2a_1xy + a_2y^2 + a_3x + a_4y + a_5$$

の a_0, a_1, \dots, a_5 を線形システムを用いて求める

$XA = T$ という線形システムを解けば良い

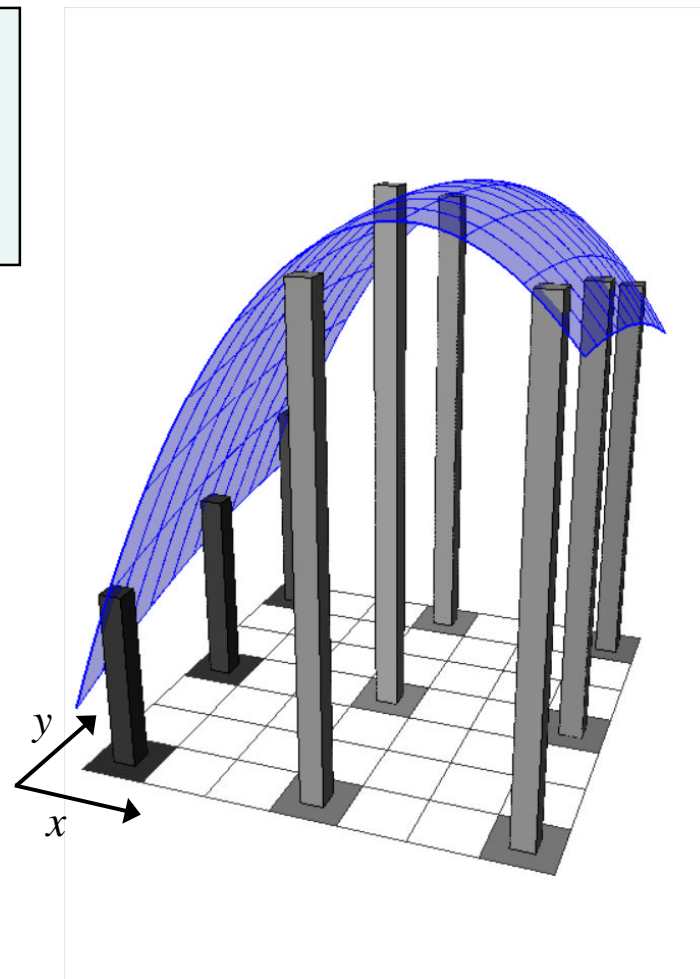
$A = X^{-1}T$ で求められる?

⇒ X は正方行列でないため逆行列は計算不可能

⇒ 疑似逆行列 X^+ を用いて $A = X^+T$ と計算可能

$$X^+ = (X^T X)^{-1} X^T$$

最小二乗法の解と一致する



2.2.2 放物面近似の計算手順 (1/11)

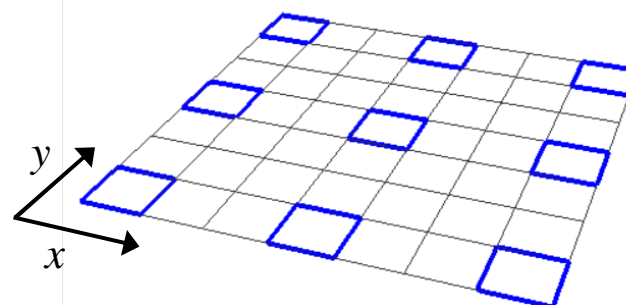
手順1:
Xを計算する

- ローカル座標系で定義することで
Xを共通化

前計算

描画処理

$$X = \begin{bmatrix} x_1^2 & 2x_1y_1 & y_1^2 & x_1 & y_1 & 1 \\ x_2^2 & 2x_2y_2 & y_2^2 & x_2 & y_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_9^2 & 2x_9y_9 & y_9^2 & x_9 & y_9 & 1 \end{bmatrix}$$



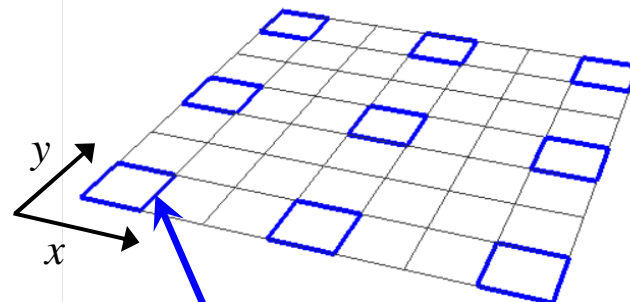
2.2.2 放物面近似の計算手順 (2/11)

手順1:
Xを計算する

- ローカル座標系で定義することで
Xを共通化

前計算

描画処理

$$X = \begin{bmatrix} x_1^2 & 2x_1y_1 & y_1^2 & x_1 & y_1 & 1 \\ x_2^2 & 2x_2y_2 & y_2^2 & x_2 & y_2 & 1 \\ \vdots & & & & & \\ x_9^2 & 2x_9y_9 & y_9^2 & x_9 & y_9 & 1 \end{bmatrix}$$


$(x_1, y_1) = (-3, -3)$

2.2.2 放物面近似の計算手順 (3/11)

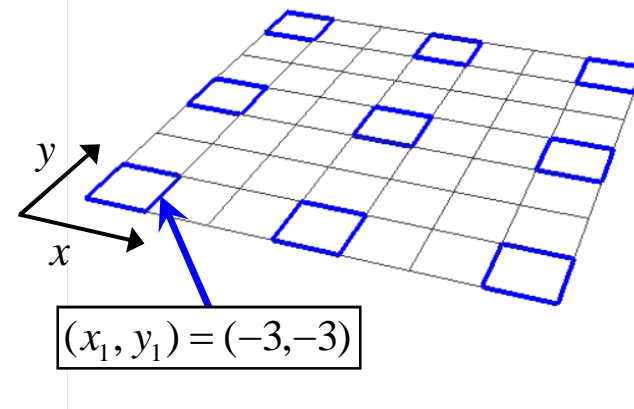
手順1:
Xを計算する

- ローカル座標系で定義することで
Xを共通化

前計算

描画処理

$$X = \begin{bmatrix} 9 & 18 & 9 & -3 & -3 & 1 \\ x_2^2 & 2x_2y_2 & y_2^2 & x_2 & y_2 & 1 \\ \vdots & & & & & \\ x_9^2 & 2x_9y_9 & y_9^2 & x_9 & y_9 & 1 \end{bmatrix}$$



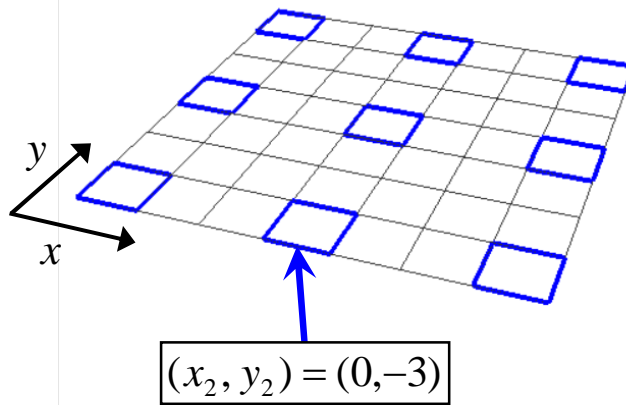
2.2.2 放物面近似の計算手順 (4/11)

手順1:
Xを計算する

- ローカル座標系で定義することで
Xを共通化

前計算

描画処理

$$X = \begin{bmatrix} 9 & 18 & 9 & -3 & -3 & 1 \\ x_2^2 & 2x_2y_2 & y_2^2 & x_2 & y_2 & 1 \\ \vdots & & & & & \\ x_9^2 & 2x_9y_9 & y_9^2 & x_9 & y_9 & 1 \end{bmatrix}$$


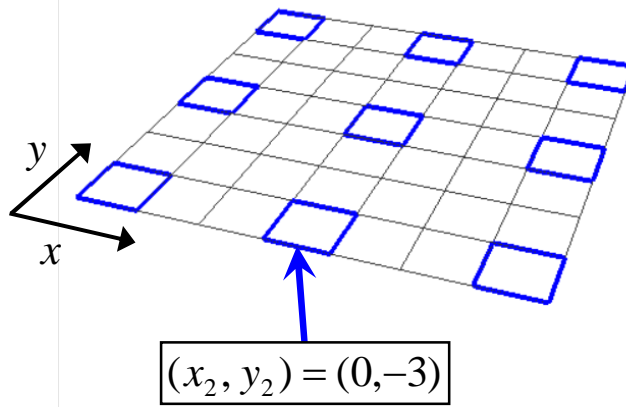
2.2.2 放物面近似の計算手順 (5/11)

手順1:
Xを計算する

- ローカル座標系で定義することで
Xを共通化

前計算

描画処理

$$X = \begin{bmatrix} 9 & 18 & 9 & -3 & -3 & 1 \\ 0 & 0 & 9 & 0 & -3 & 1 \\ \vdots & & & & & \\ x_9^2 & 2x_9y_9 & y_9^2 & x_9 & y_9 & 1 \end{bmatrix}$$


$(x_2, y_2) = (0, -3)$

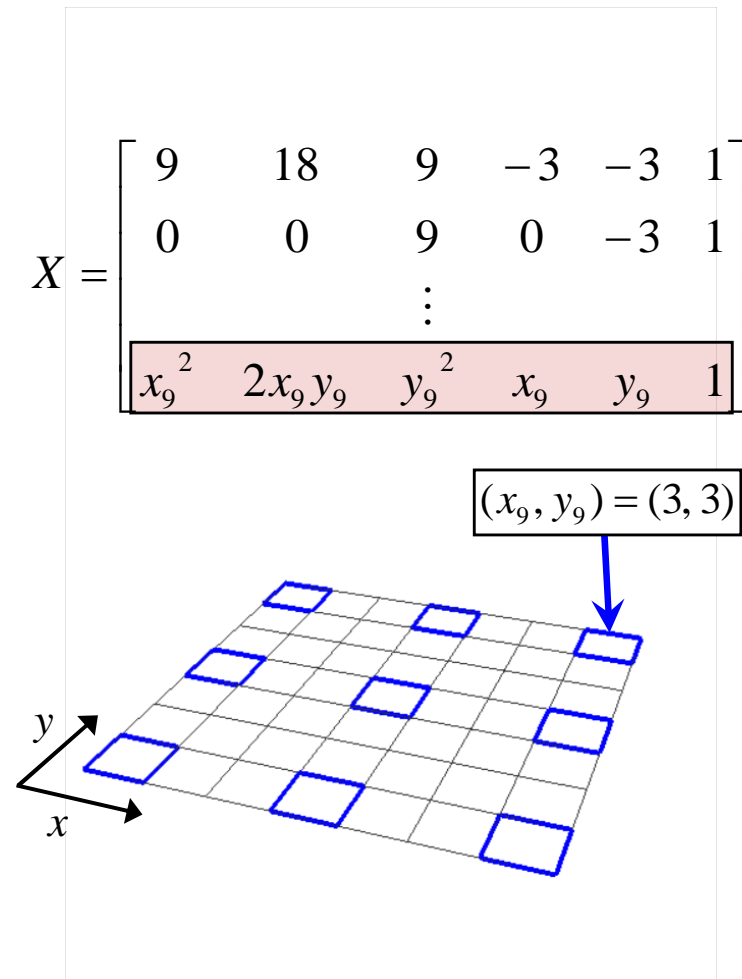
2.2.2 放物面近似の計算手順 (6/11)

手順1:
Xを計算する

- ローカル座標系で定義することで
Xを共通化

前計算

描画処理



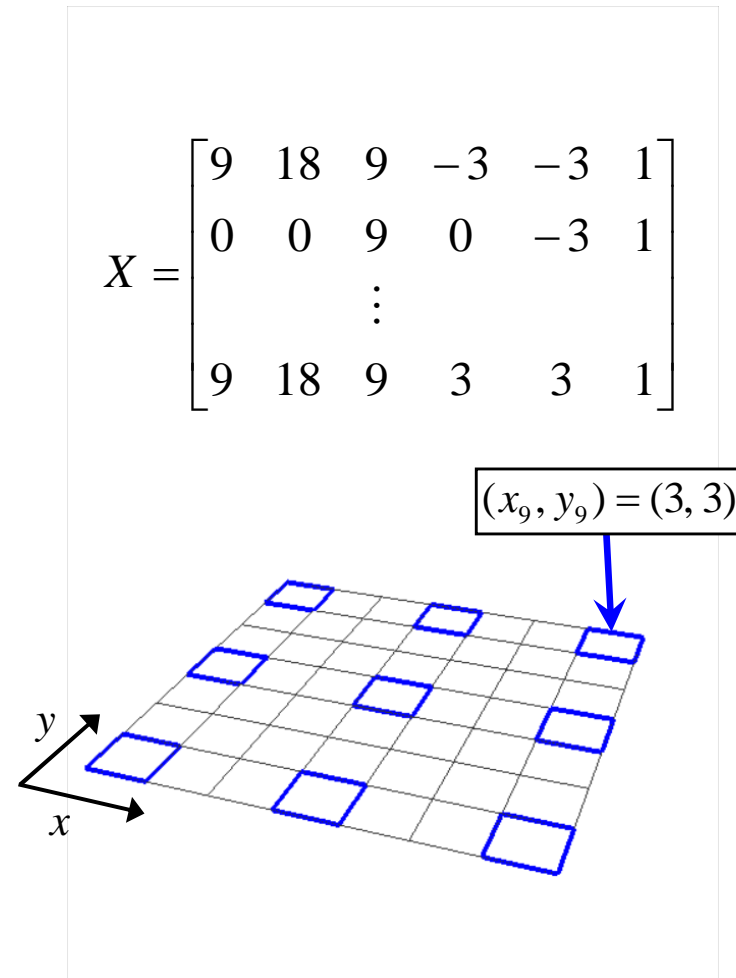
2.2.2 放物面近似の計算手順 (7/11)

手順1:
Xを計算する

- ローカル座標系で定義することで
Xを共通化

前計算

描画処理



2.2.2 放物面近似の計算手順 (8/11)

手順1：
Xを計算する

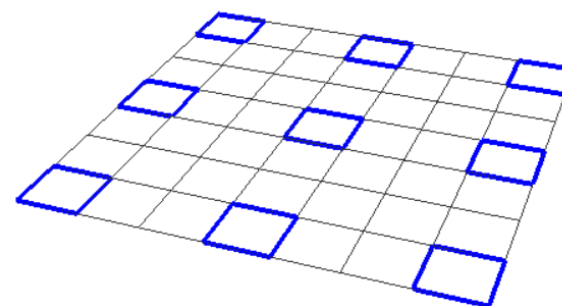
- ローカル座標系で定義することで Xを共通化
- 描きたい線の太さにのみ依存

手順2：
 $X^+ = (X^T X)^{-1} X^T$ を計算する

前計算

描画処理

$$X = \begin{bmatrix} 9 & 18 & 9 & -3 & -3 & 1 \\ 0 & 0 & 9 & 0 & -3 & 1 \\ & & \vdots & & & \\ 9 & 18 & 9 & 3 & 3 & 1 \end{bmatrix}$$



2.2.2 放物面近似の計算手順 (9/11)

手順1：
 X を計算する

- ローカル座標系で定義することで X を共通化
- 描きたい線の太さにのみ依存

手順2：
 $X^+ = (X^T X)^{-1} X^T$ を計算する

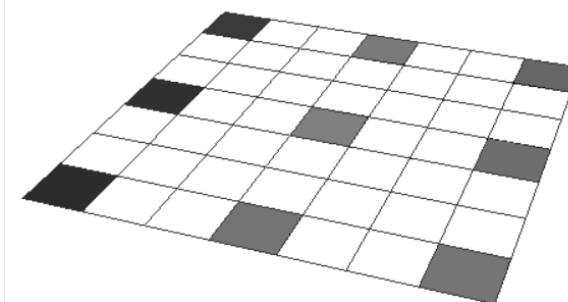
前計算

描画処理

手順3：
 X^+ を Pixel Shader の Uniform 変数
に設定する

手順4：
Pixel Shader で近傍のサンプル点
をフェッチし、 T を構築する

$$T = \begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ t_9 \end{bmatrix} = \begin{bmatrix} 0.27 \\ 0.73 \\ \vdots \\ 0.64 \end{bmatrix}$$



2.2.2 放物面近似の計算手順 (10/11)

手順1：
 X を計算する

- ローカル座標系で定義することで X を共通化
- 描きたい線の太さにのみ依存

手順2：
 $X^+ = (X^T X)^{-1} X^T$ を計算する

前計算

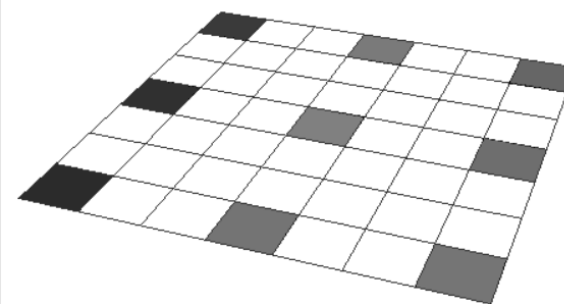
描画処理

手順3：
 X^+ を Pixel Shader の Uniform 変数
に設定する

手順4：
Pixel Shader で近傍のサンプル点
をフェッチし、 T を構築する

手順5：
 $X^+ T$ を計算して A を求める

$$A = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix} = \begin{bmatrix} -0.029 \\ -0.002 \\ -0.002 \\ 0.063 \\ 0.002 \\ 0.783 \end{bmatrix}$$



2.2.2 放物面近似の計算手順 (11/11)

手順1：
Xを計算する

- ローカル座標系で定義することでXを共通化
- 描きたい線の太さにのみ依存

手順2：
 $X^+ = (X^T X)^{-1} X^T$ を計算する

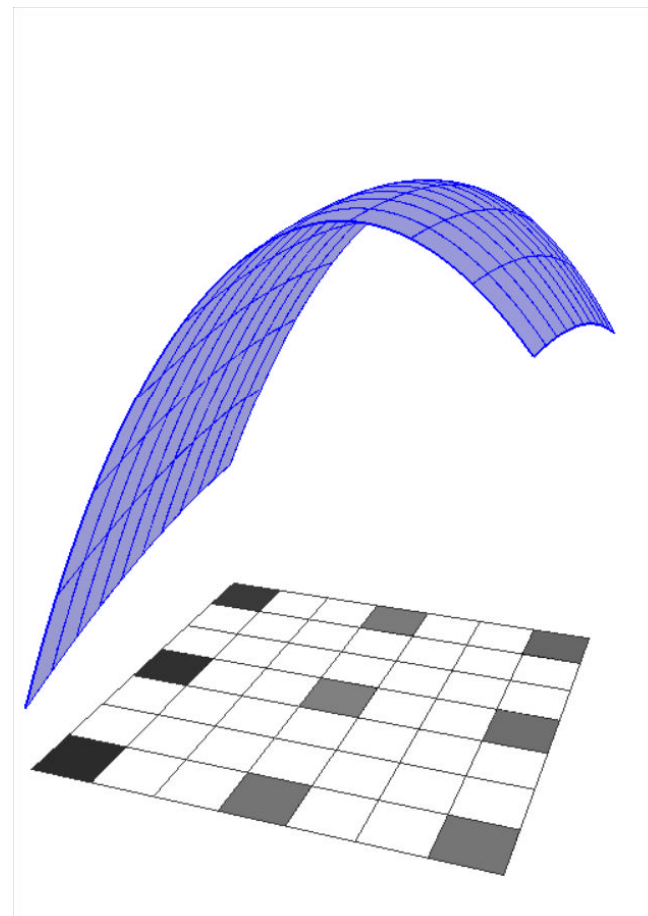
前計算

描画処理

手順3：
X⁺をPixel ShaderのUniform変数に設定する

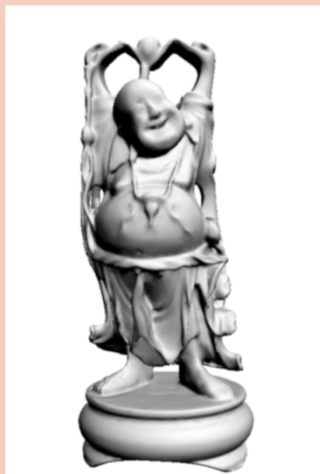
手順4：
Pixel Shaderで近傍のサンプル点をフェッチし、Tを構築する

手順5：
X⁺Tを計算してAを求める

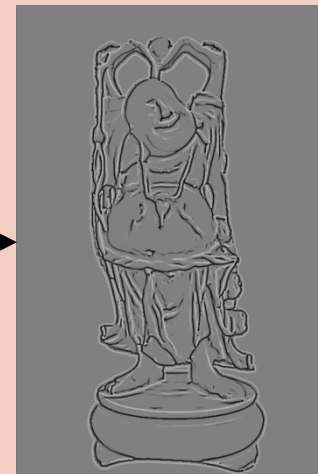


$$f(x, y) = a_0 x^2 + 2a_1 xy + a_2 y^2 + a_3 x + a_4 y + a_5$$

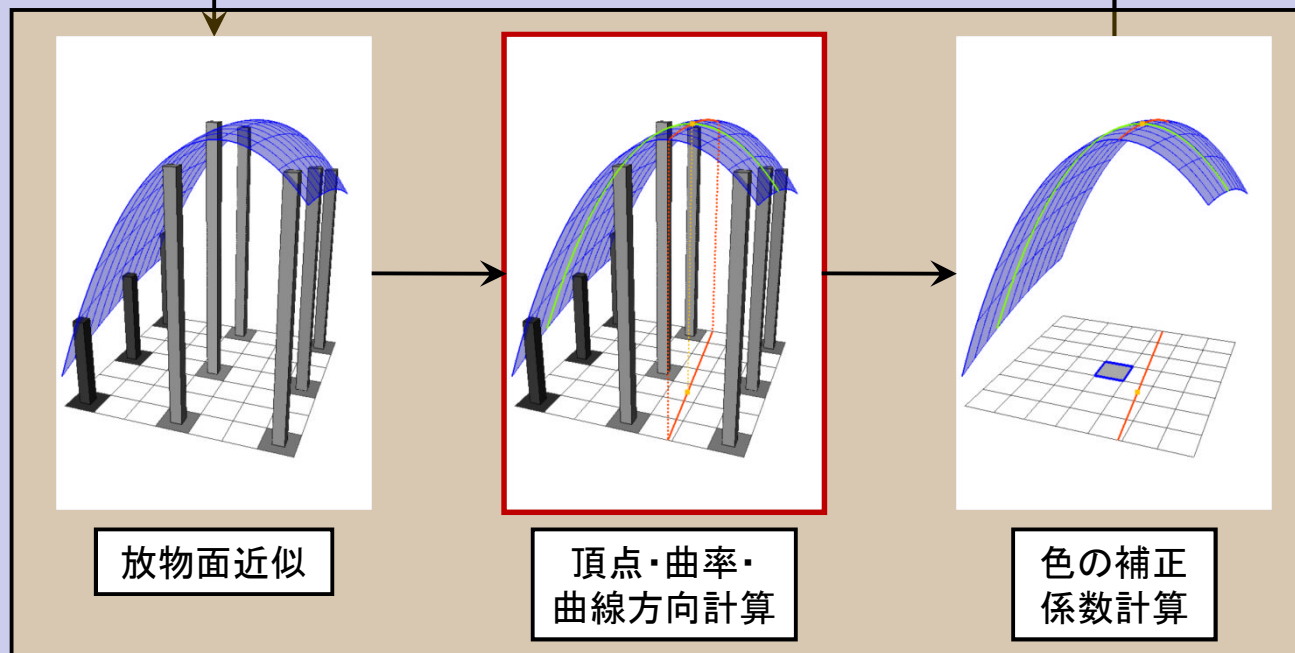
特徴線抽出手順



ガウシアン
フィルタリング



特徴線抽出



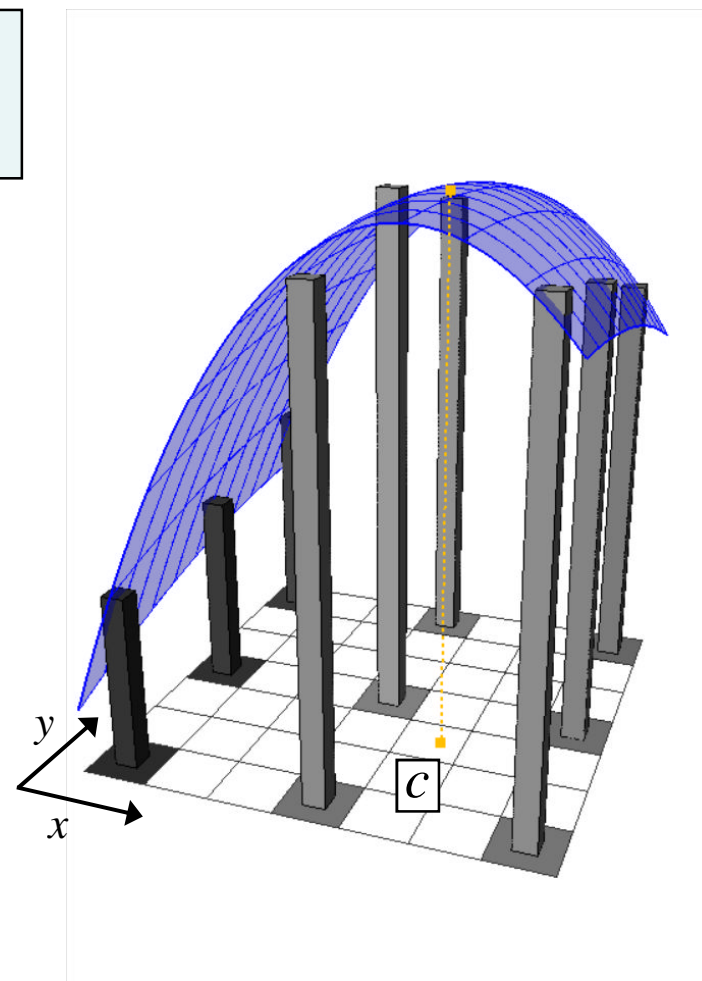
2.3.1 放物面の頂点

放物線:

$$f(x, y) = a_0x^2 + 2a_1xy + a_2y^2 + a_3x + a_4y + a_5$$

放物線の頂点:

$$c = \left(\frac{a_1a_4 - a_2a_3}{2(a_0a_2 - a_1^2)}, \frac{a_1a_3 - a_0a_4}{2(a_0a_2 - a_1^2)} \right)$$



2.3.2 放物面の主曲率・曲線方向

放物線:

$$f(x, y) = a_0x^2 + 2a_1xy + a_2y^2 + a_3x + a_4y + a_5$$

$$\text{主曲率: } \alpha_1 = \frac{(a_0 + a_2) + \sqrt{(a_0 + a_2)^2 + 4a_1^2}}{2}$$

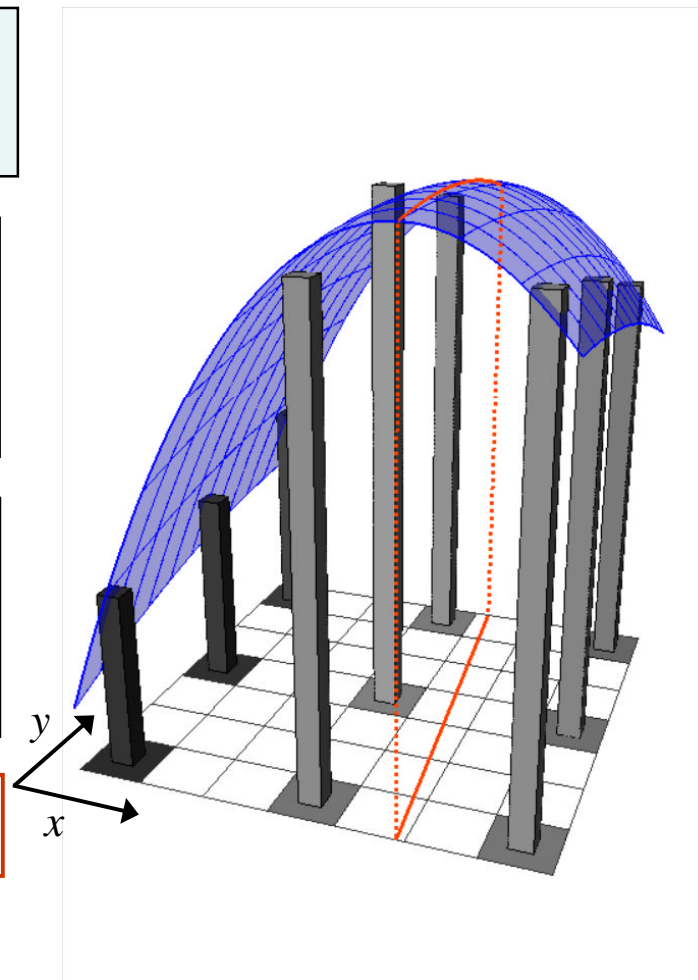
$$\text{曲線方向: } \vec{d}_{\alpha_1} = \text{normalize}(-2a_1, (a_0 - a_2) - \sqrt{(a_0 + a_2)^2 + 4a_1^2})$$

$$\text{主曲率: } \alpha_2 = \frac{(a_0 + a_2) - \sqrt{(a_0 + a_2)^2 + 4a_1^2}}{2}$$

$$\text{曲線方向: } \vec{d}_{\alpha_2} = \text{normalize}(-2a_1, (a_0 - a_2) + \sqrt{(a_0 + a_2)^2 + 4a_1^2})$$

主曲率の絶対値が小さい組が尾根線

山は主曲率が負、谷は主曲率が正



2.3.2 放物面の主曲率・曲線方向

放物線:

$$f(x, y) = a_0x^2 + 2a_1xy + a_2y^2 + a_3x + a_4y + a_5$$

$$\text{主曲率: } \alpha_1 = \frac{(a_0 + a_2) + \sqrt{(a_0 + a_2)^2 + 4a_1^2}}{2}$$

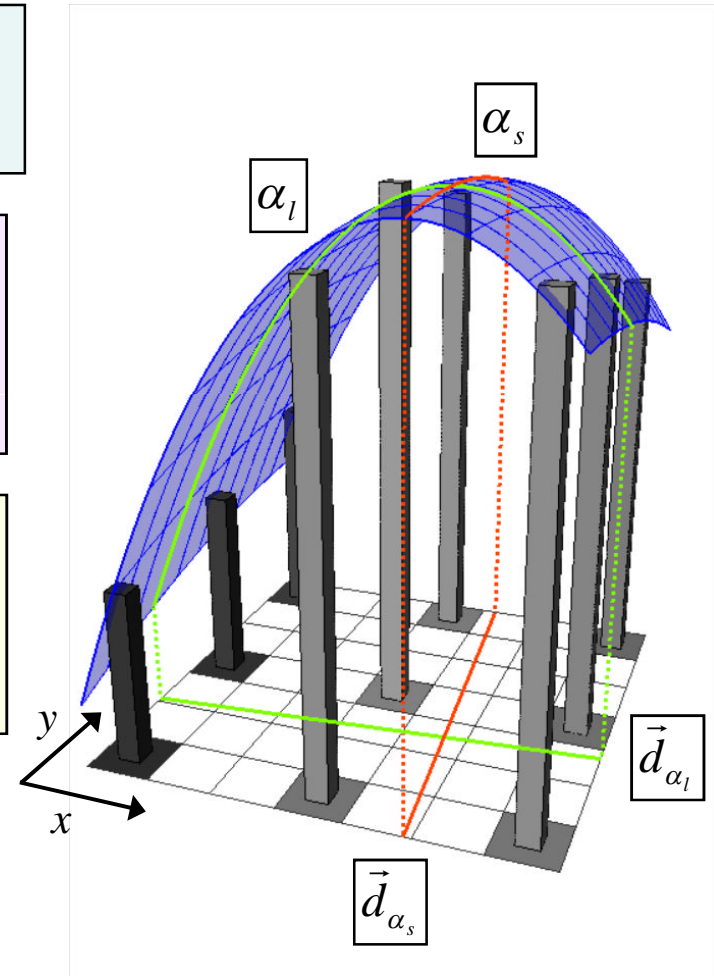
$$\text{曲線方向: } \vec{d}_{\alpha_1} = \text{normalize}(-2a_1, (a_0 - a_2) - \sqrt{(a_0 + a_2)^2 + 4a_1^2})$$

$$\text{主曲率: } \alpha_2 = \frac{(a_0 + a_2) - \sqrt{(a_0 + a_2)^2 + 4a_1^2}}{2}$$

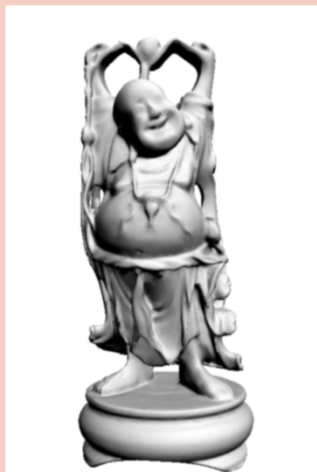
$$\text{曲線方向: } \vec{d}_{\alpha_2} = \text{normalize}(-2a_1, (a_0 - a_2) + \sqrt{(a_0 + a_2)^2 + 4a_1^2})$$

主曲率の絶対値の小さい組: $\alpha_s, \vec{d}_{\alpha_s}$

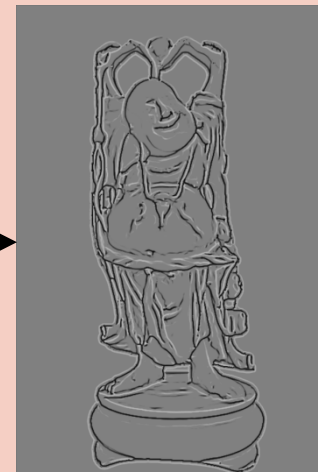
主曲率の絶対値の大きい組: $\alpha_l, \vec{d}_{\alpha_l}$



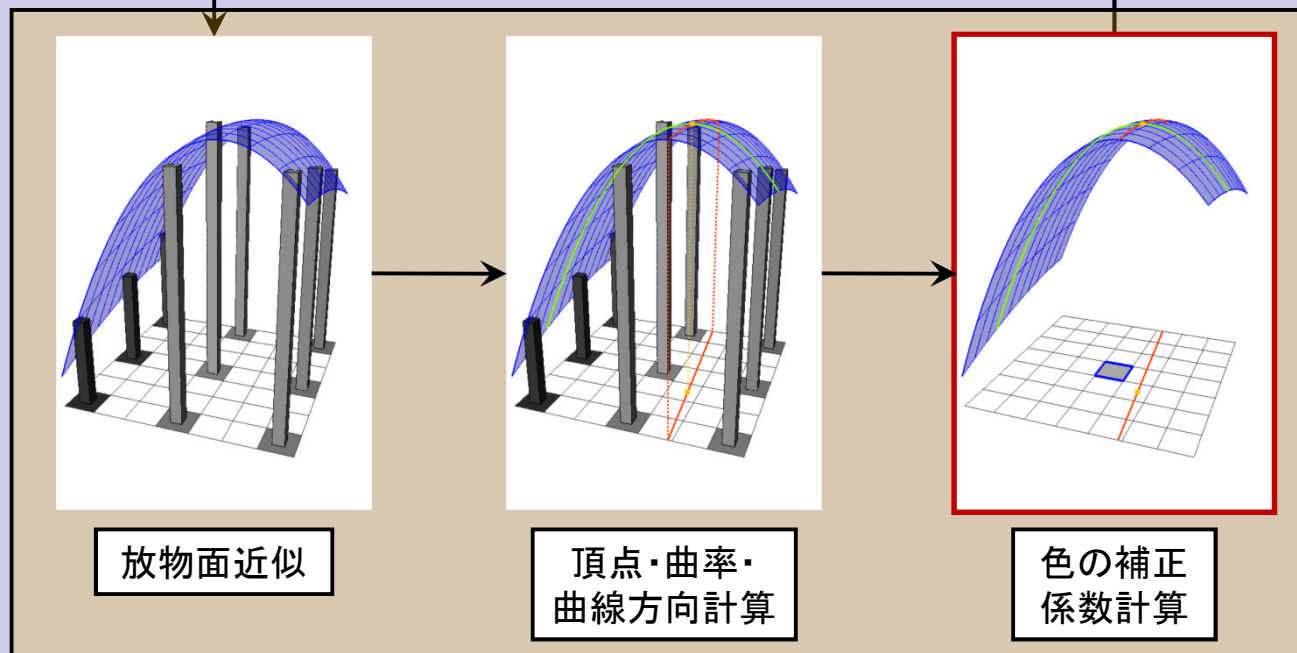
特徴線抽出手順



ガウシアン
フィルタリング



特徴線抽出



放物面近似

頂点・曲率・
曲線方向計算

色の補正
係数計算

2.4 色の補正係数 (1/5)

山線・谷線への距離による係数 : S_d (0.0~1.0)

- 山線・谷線に近いほど値が大きい

曲率の大きさによる係数 : S_c (0.0~1.0)

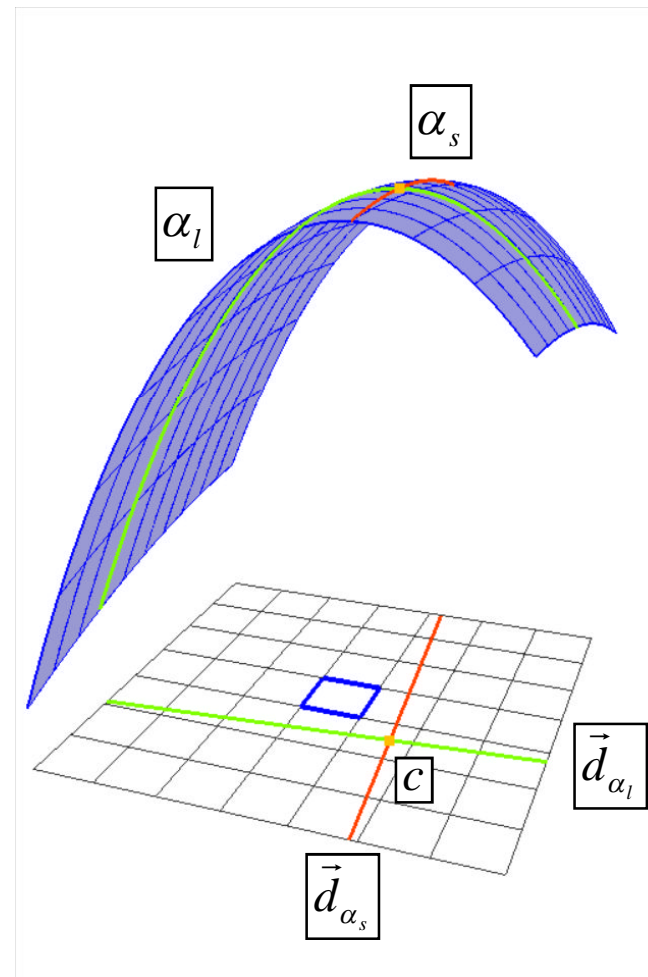
- 放物面の曲率が大きいほど値が大きい

曲面の種類による補正 : S

- 山は明るく、谷は暗く色を変化

山の場合 (α_l が負) $S = 1.0 + (S_d \times S_c)$

谷の場合 (α_l が正) $S = 1.0 - (S_d \times S_c)$



2.4 色の補正係数 (2/5)

山線・谷線への距離による係数 : S_d (0.0~1.0)

- 山線・谷線に近いほど値が大きい

曲率の大きさによる係数 : S_c (0.0~1.0)

- 放物面の曲率が大きいほど値が大きい

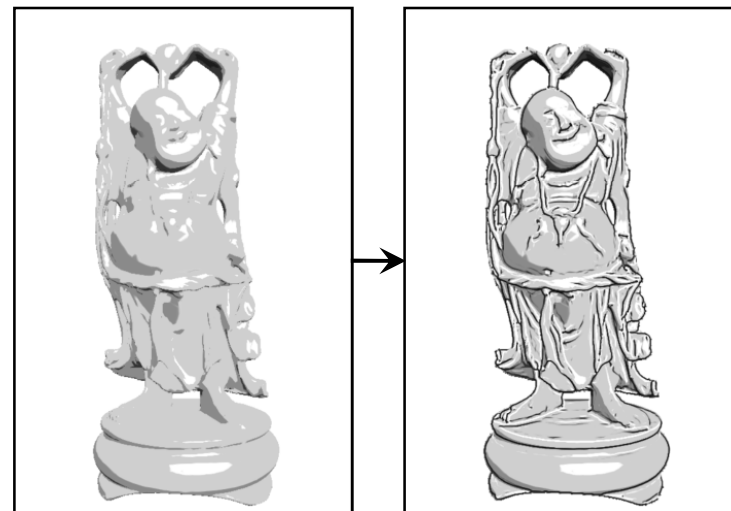
曲面の種類による補正 : S

- 山は明るく、谷は暗く色を変化

山の場合 (α_l が負) $S = 1.0 + (S_d \times S_c)$

谷の場合 (α_l が正) $S = 1.0 - (S_d \times S_c)$

S を乗算



2.4 色の補正係数 (3/5)

山線・谷線への距離による係数 : S_d (0.0~1.0)

- 山線・谷線に近いほど値が大きい

$$S_d = \max(1.0 - (d / \{LineWidth / 2\}), 0.0)$$

曲率の大きさによる係数 : S_c (0.0~1.0)

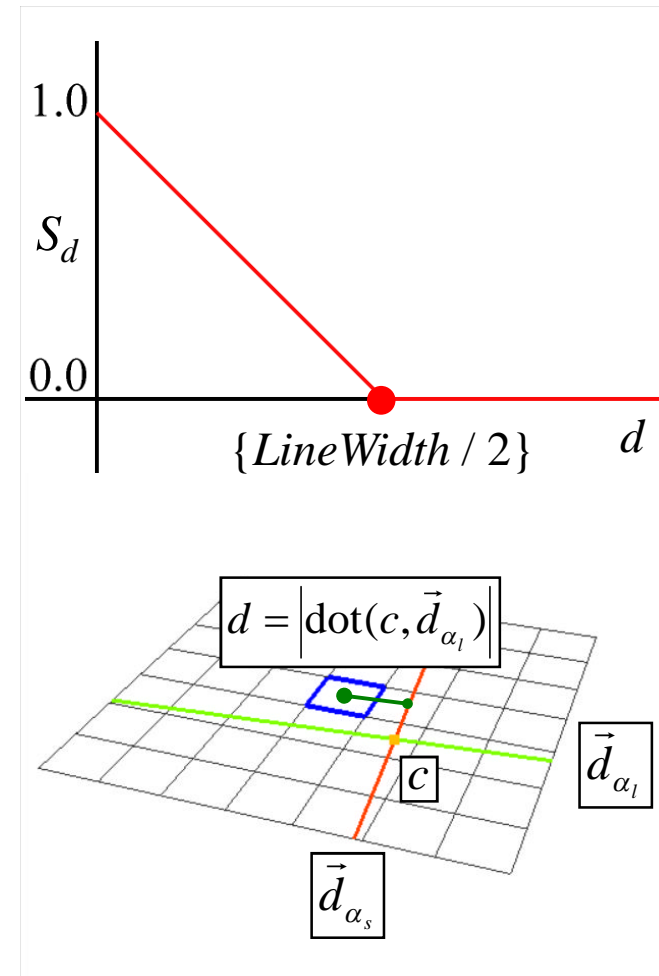
- 放物面の曲率が大きいほど値が大きい

曲面の種類による補正 : S

- 山は明るく、谷は暗く色を変化

$$\text{山の場合 } (\alpha_l \text{ が負}) \quad S = 1.0 + (S_d \times S_c)$$

$$\text{谷の場合 } (\alpha_l \text{ が正}) \quad S = 1.0 - (S_d \times S_c)$$



2.4 色の補正係数 (4/5)

山線・谷線への距離による係数 : S_d (0.0~1.0)

- 山線・谷線に近いほど値が大きい

$$S_d = \max(1.0 - (d / \{LineWidth / 2\}), 0.0)$$

曲率の大きさによる係数 : S_c (0.0~1.0)

- 放物面の曲率が大きいほど値が大きい

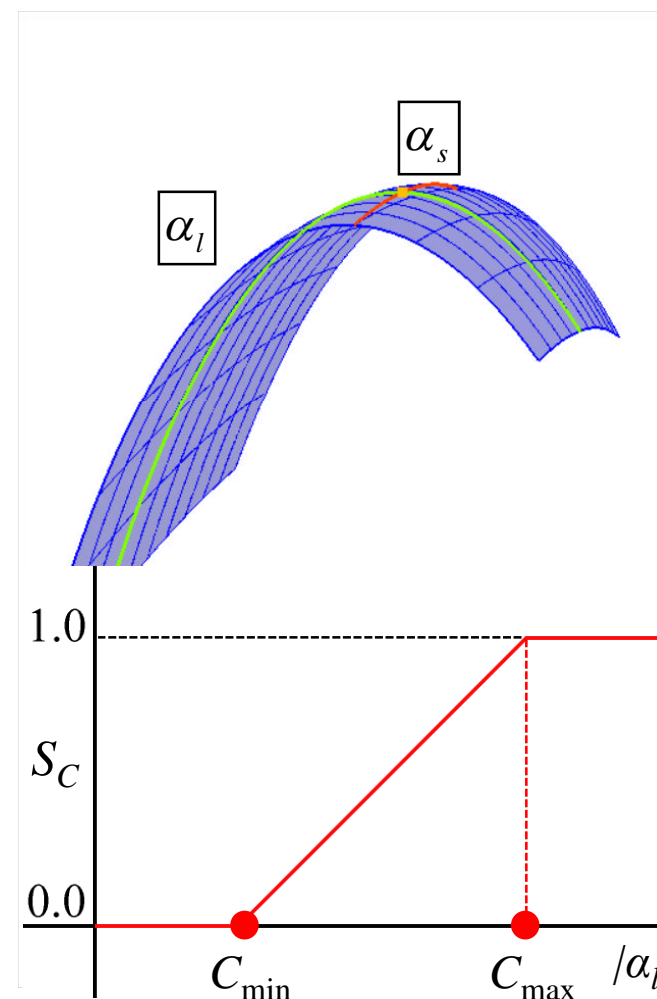
$$S_c = \text{clamp}\left(\frac{|\alpha_l| - C_{\min}}{C_{\max} - C_{\min}}, 0.0, 1.0\right)$$

曲面の種類による補正 : S

- 山は明るく、谷は暗く色を変化

$$\text{山の場合 } (\alpha_l \text{ が負}) \quad S = 1.0 + (S_d \times S_c)$$

$$\text{谷の場合 } (\alpha_l \text{ が正}) \quad S = 1.0 - (S_d \times S_c)$$



2.4 色の補正係数 (5/5)

山線・谷線への距離による係数 : S_d (0.0~1.0)

- 山線・谷線に近いほど値が大きい

$$S_d = \max(1.0 - (d / \{LineWidth / 2\}), 0.0)$$

曲率の大きさによる係数 : S_c (0.0~1.0)

- 放物面の曲率が大きいほど値が大きい

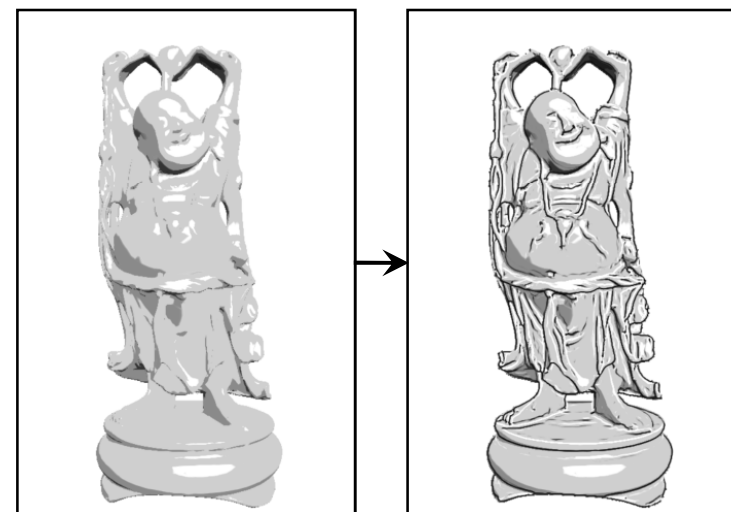
$$S_c = \text{clamp}\left(\frac{|\alpha_l| - C_{\min}}{C_{\max} - C_{\min}}, 0.0, 1.0\right)$$

曲面の種類による補正 : S

- 山は明るく、谷は暗く色を変化

山の場合 (α_l が負) $S = 1.0 + (S_d \times S_c)$

谷の場合 (α_l が正) $S = 1.0 - (S_d \times S_c)$



Sを乗算

2.5 まとめ・考察 (1/2)

- まとめ

1. 各ピクセルで、近傍の色情報の放物面近似によって自分が山・谷に居るのかを判断
山に居る場合は色を明るく、谷に居る場合は色を暗く変化させる
2. 放物面の係数から、放物面の曲率や山・谷線方向といった付加情報を計算可能
3. 山・谷線に近いほど、また、曲面の曲率が大きいほど大きく色を変化させる

2.5 まとめ・考察 (2/2)

- 考察

1. 山・谷の区別だけではなく、曲率や山・谷線方向を導出し、レンダリングに反映させている点は面白い
2. 現在は二次元処理なので、三次元情報を上手く利用すれば、結果の向上が図れるかも
3. 近傍の色情報の放物面近似は、SSAOやトーンマップにも応用可能かも

発表の流れ

1. 発表の概要
2. 論文紹介

「Line Drawing via Abstracted Shading」



3. 発表のまとめ

3 発表のまとめ

- 紹介論文:

Line Drawing via Abstracted Shading [Lee et al. SIGGRAPH 2007]



3 発表のまとめ

著作権について

本スライド内で引用した論文の図・動画の
著作権は全て論文著者に帰属します

その他の図・スライド本体・挿絵等の著作権は
株式会社シリコンスタジオに帰属します

最後に

CEDEC2009
CESA Developers Conference

Silicon  Studio

御清聴ありがとうございました

本セッションの流れ

前半発表者：田村



後半発表者：川瀬

