

イメージシンセシス

イメージ・ビデオベースCG技術

岩手大学大学院
総合科学研究科 理工学専攻
デザイン・メディア工学コース
藤本 忠博

従来のCGの問題点

■ 従来のCG

3次元幾何モデル(モデリング) → 表示処理(レンダリング) → CG画像



■ 問題点

- 複雑な形状を扱うのが困難
 - 幾何モデルの作成が大変
 - レンダリングに時間がかかる
- 実世界の物体を直接的に扱えない

イメージ・ビデオベースCG技術

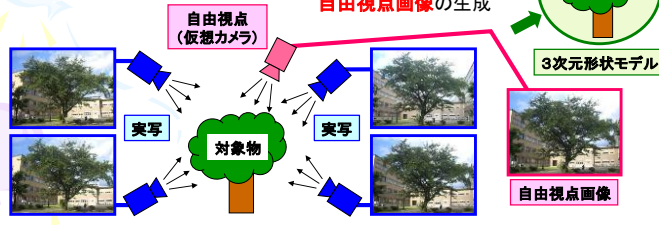
■ 従来のCG

3次元幾何モデル(モデリング)
→ 表示処理(レンダリング)
→ CG画像



■ イメージ・ビデオベースCG

複数の実写画像(参照画像) → 3次元形状モデルの構築
自由視点画像の生成



Plenoptic function の実現

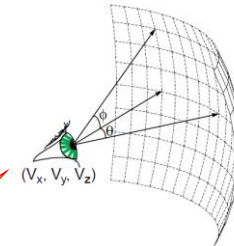
■ 実世界の全てのシーンを表現する光の情報

■ L. McMillan, G. Bishop, "Plenoptic Modeling: An Image-Based Rendering System", SIGGRAPH 95

■ 光の輝度 p を次式で表現する.

$$p = P(\theta, \phi, \lambda, V_x, V_y, V_z, t)$$

θ, ϕ : 視線方向
 λ : 光の波長
 V_x, V_y, V_z : 視点位置
 t : 時間



これが分かれば、
実世界の全てのシーンが
CGで再現できる！

L. McMillan, G. Bishop, "Plenoptic Modeling: An Image-Based Rendering System", SIGGRAPH 95

イメージ・ビデオベースCG技術の分類

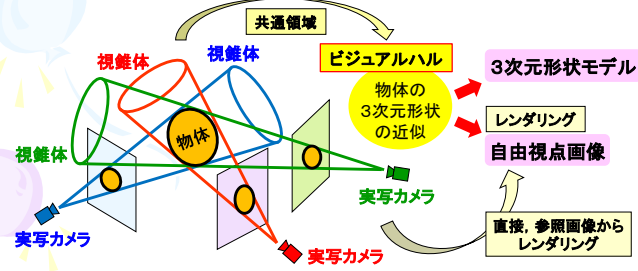
出力	3次元形状モデル			2.5次元モデル	仮想カメラからの自由視点画像
	ボクセル	ポリゴン	点群	仮想カメラからのデプス画像	
参照画像からの入力					
物体輪郭(シルエット)	視体積交差法	Polyhedral visual hull			Image-based visual hull
画素 色	ボクセルカラーリング			ブレンスリーブ ステレオマッチング	ビュー補間 ライトフィールド ドレンダリング, ルミグラフィ フォーカル ブレン手法
	デプス		デプスハル	層化デプス画像(LDI)	ポイントスプラッチング

イメージ・ビデオベースCG技術の分類

出力	3次元形状モデル			2.5次元モデル	仮想カメラからの自由視点画像
	ボクセル	ポリゴン	点群	仮想カメラからのデプス画像	
参照画像からの入力					
物体輪郭(シルエット)	視体積交差法	Polyhedral visual hull			Image-based visual hull
画素 色	ボクセルカラーリング			ブレンスリーブ ステレオマッチング	ビュー補間 ライトフィールド ドレンダリング, ルミグラフィ フォーカル ブレン手法
	デプス		デプスハル	層化デプス画像(LDI)	ポイントスプラッチング

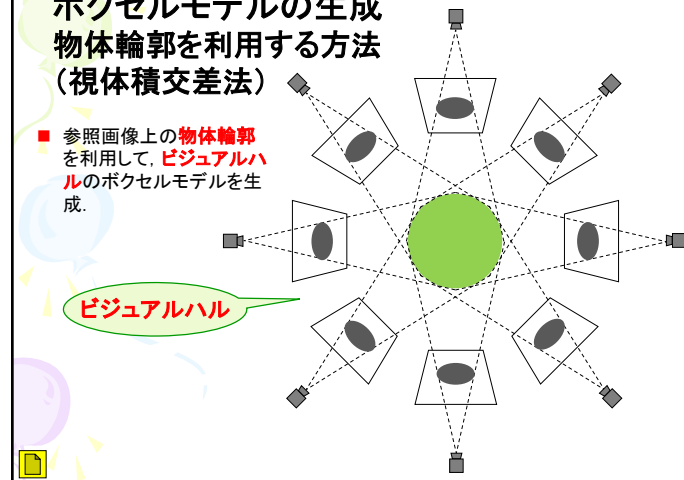
物体輪郭を利用する方法

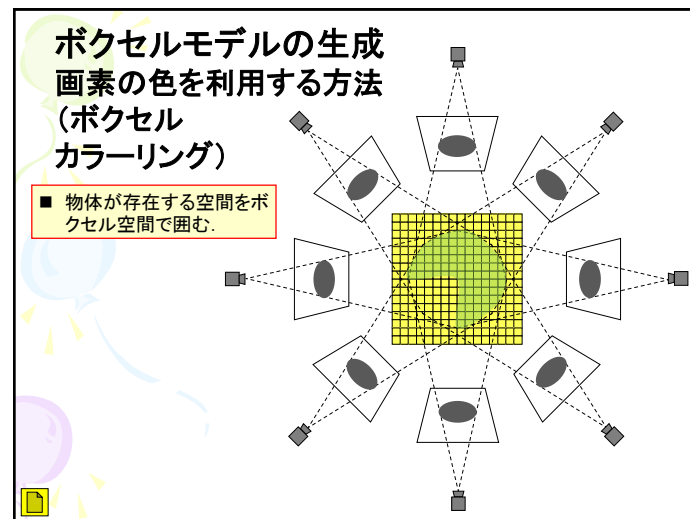
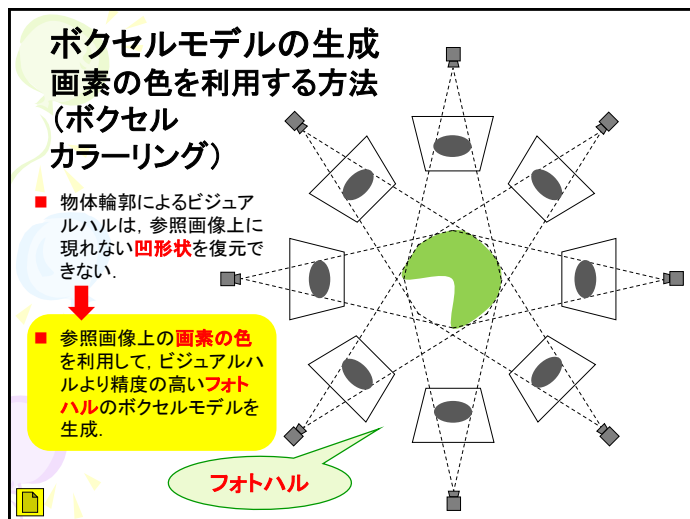
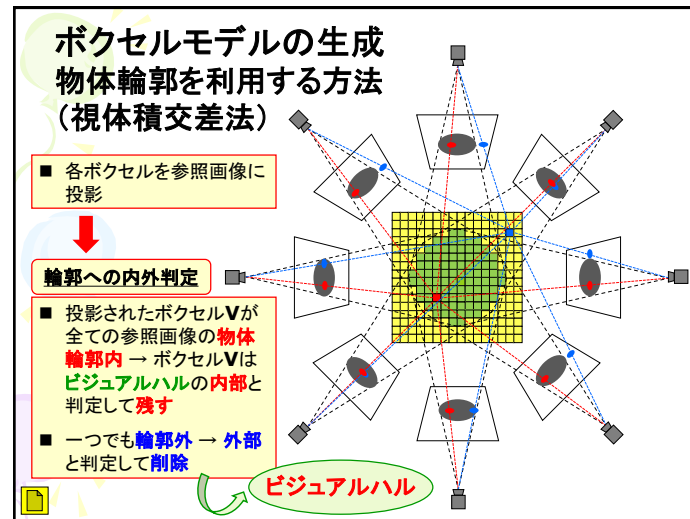
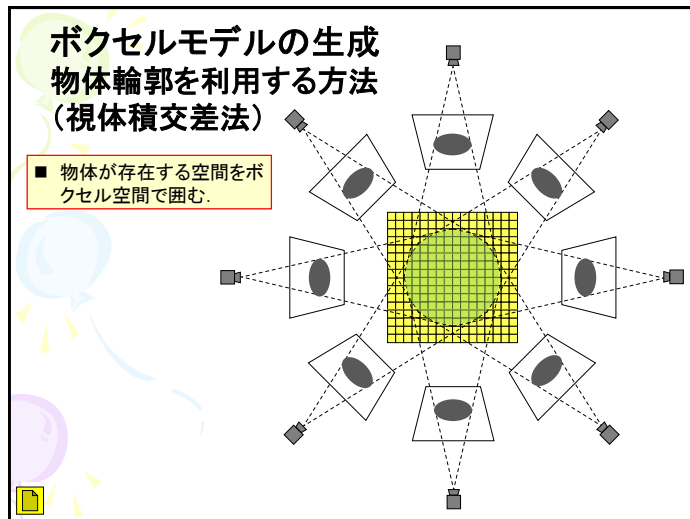
- 各実写カメラの参照画像上の**物体輪郭(シルエット)**による**視錐体**の共通領域として**ビジュアルハル**を構築する。
- ボクセルベース手法 → 3次元形状モデルの構築
- ポリゴンベース手法 → 3次元形状モデルの構築
- イメージベース手法 → 自由視点画像の生成

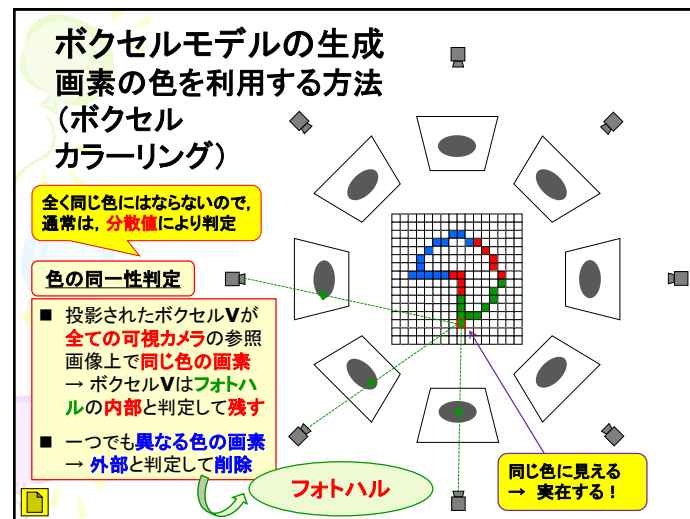
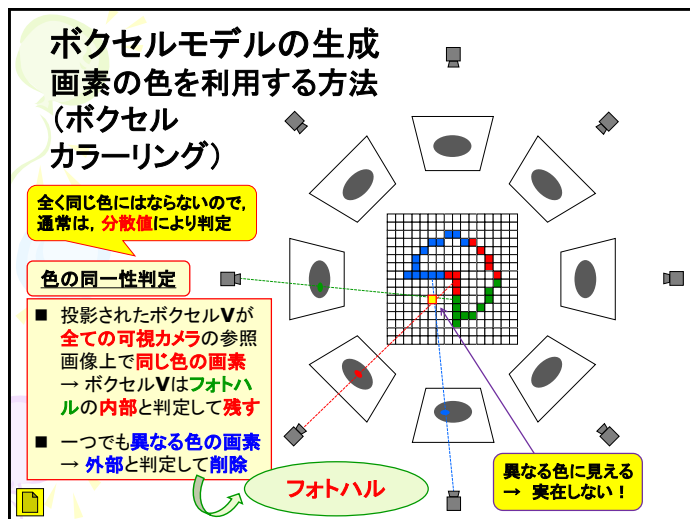
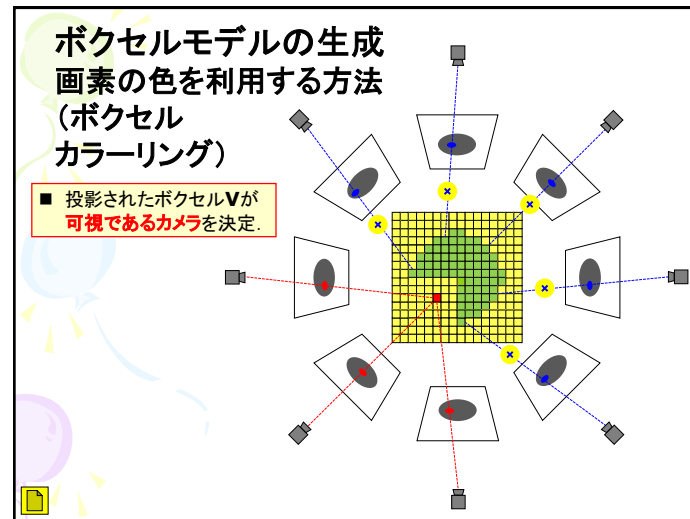
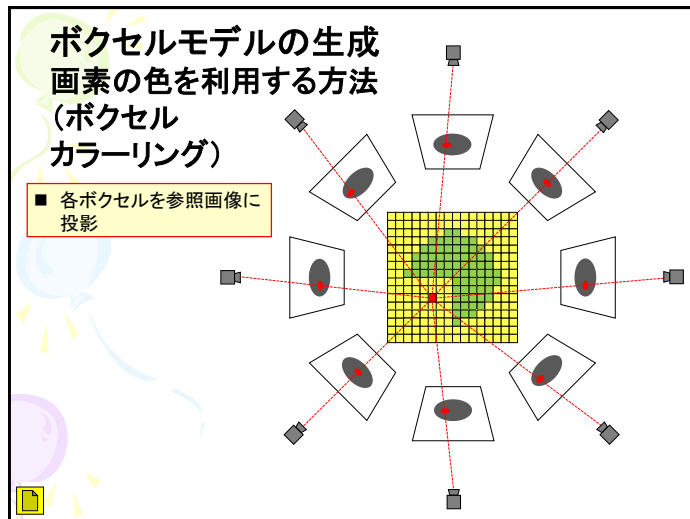


ボクセルモデルの生成 物体輪郭を利用する方法 (視体積交差法)

- 参照画像上の**物体輪郭**を利用して、**ビジュアルハル**のボクセルモデルを生成。

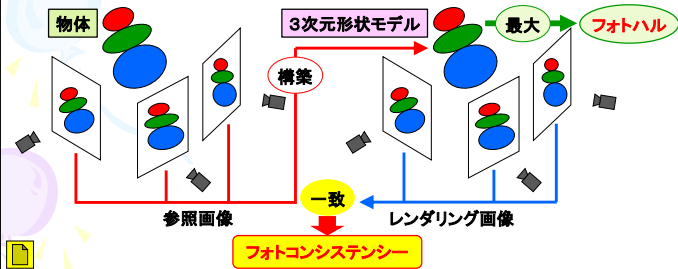






ボクセルモデルの生成 画素の色を利用する方法 (ボクセルカラーリング)

- ビジュアルハルと同様, **フォトハル**も近似形状(正確な物体形状ではない)
- 各実写カメラの位置から**3次元形状モデル**をレンダリングした画像が各参照画像と一致 → **フォトコンシステンシー(photo-consistency)**
- **フォトハル**は**フォトコンシステント**な形状モデルのうちで**最大**(全てのフォトコンシステントな形状モデルを内包する)

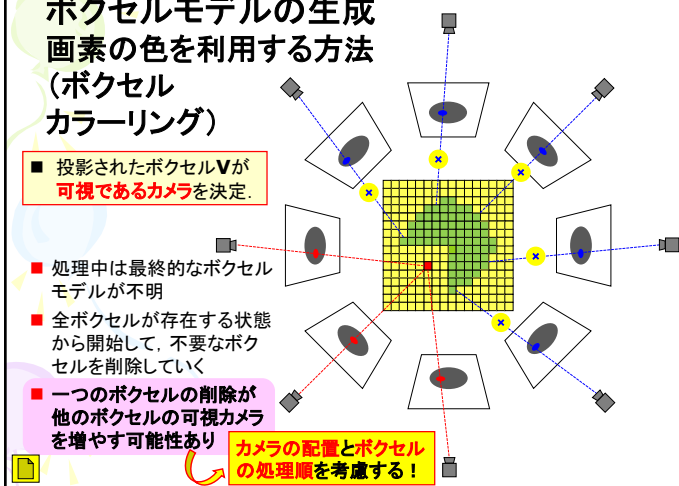


ボクセルモデルの生成 画素の色を利用する方法 (ボクセルカラーリング)

- 投影されたボクセルVが**可視であるカメラ**を決定.

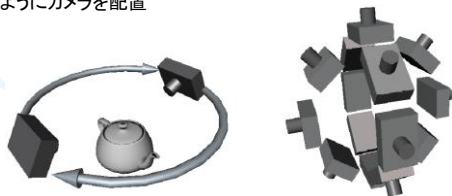
- 処理中は最終的なボクセルモデルが不明
- 全ボクセルが存在する状態から開始して, 不要なボクセルを削除していく
- 一つのボクセルの削除が他のボクセルの可視カメラを増やす可能性あり

カメラの配置とボクセルの処理順を考慮する!



ボクセルカラーリング カメラ配置に制約を与える場合

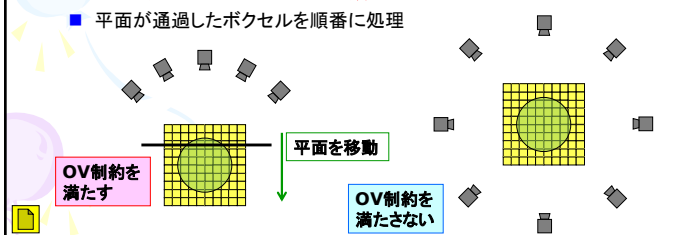
- **Voxel Coloring**
- S. Seitz, C. Dyer, "Photorealistic Scene Reconstruction by Voxel Coloring", 1997
- **Ordinary visibility constraint (OV制約)**
全カメラに対して**near-to-far (近くから遠くへ)**の順序で**全ボクセル**が処理されるようにカメラを配置



S. Seitz, C. Dyer, "Photorealistic Scene Reconstruction by Voxel Coloring", 1997

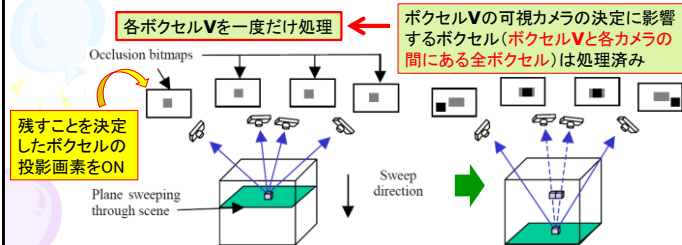
ボクセルカラーリング カメラ配置に制約を与える場合

- **Ordinary visibility constraint (OV制約)**
全カメラに対して**near-to-far (近くから遠くへ)**の順序で**全ボクセル**が処理されるようにカメラを配置
- OV制約を満たす典型的なカメラ配置とボクセル処理順
 - シーンの一方向の側にだけ全カメラを配置
 - 全カメラに対して**near-to-far**順序で平面を移動 (**平面スワイプ**)
 - 平面が通過したボクセルを順番に処理



ボクセルカラーリング カメラ配置に制約を与える場合

- OV制約を満たす典型的なカメラ配置とボクセル処理順
- シーンの一部の側にだけ全カメラを配置
- 全カメラに対して**near-to-far**順序で平面を移動(平面スイープ)
- 平面が通過したボクセルを順番に処理



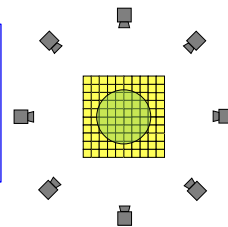
G. Slabaugh, B. Culbertson, T. Malzbender, R. Schafer, "A Survey of Methods for Volumetric Scene Reconstruction from Photographs", 2001

ボクセルカラーリング 任意のカメラ配置を可能とする場合

- カメラ配置にOV制約が無い場合、全カメラに対して**near-to-far**順序で**全ボクセルを処理できない**
- 一つのボクセルの削除が、**処理済みボクセルの可視カメラを増やす**可能性あり → **処理された際の色の同一性判定で残されたボクセルが、新たな可視カメラを含めた同一性判定により、実際には削除されるべきボクセルであると判定される可能性あり**

任意のカメラ配置のアルゴリズム

- 適当な順序でボクセルを処理
- **削除されるボクセルが一つでも生じた場合**, 残っているボクセルに対して、「**可視カメラの決定**」と「**色の同一性判定**」をやり直す
- 削除されるボクセルが生じなくなった時点で処理を終了



ボクセルカラーリング 任意のカメラ配置を可能とする場合

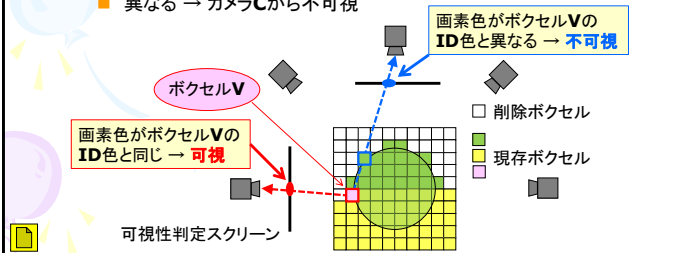
- **Space Carving**
- K. N. Kutulakos, S. M. Seitz, "What Do N Photographs Tell Us about 3D Shape?", 1998
- K. N. Kutulakos, "Approximate N-View Stereo", 2000
- K. N. Kutulakos, S. M. Seitz, "A Theory of Shape by Space Carving", 2000
- 方向を変えて**複数の平面スイープ**を実行(例えば, x, y, z軸の正負方向).
- **平面が通過した後方のカメラ**だけを可視カメラの候補とする.
→ それらのカメラに対して**near-to-far**順序の**平面スイープ**になる.
- カメラ校正等の誤差の影響を軽減するため, **色の同一性判定を改善**
各可視カメラの参照画像上へのボクセルの投影点を中心として**半径rの円領域**に同じ色の画素があるか.

ボクセルカラーリング 任意のカメラ配置を可能とする場合

- **Generalized Voxel Coloring**
- W. B. Culbertson, T. Malzbender, G. Slabaugh, "Generalized Voxel Coloring", 1999
- カメラごとに**可視性判定スクリーン**を用意.
- 各ボクセルに**可視性判定のための固有のID色**を与える.
- **現在残っているボクセル(現存ボクセル)**について次の処理を行う.
 - 各カメラの可視性判定スクリーンに全現存ボクセルを各**ID色**で描画(ハードウェアによる**Zバッファレンダリング**などで効率化)
 - 各ボクセル**V**を各カメラ**C**の可視性判定スクリーンに投影
 - 投影点の画素色がボクセル**V**の**ID色**と同じ → カメラ**C**から可視
 - 異なる → カメラ**C**から不可視

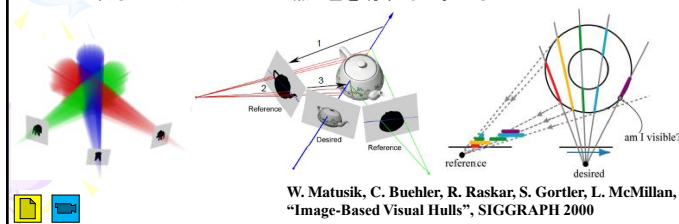
ボクセルカラーリング 任意のカメラ配置を可能とする場合

- 現在残っているボクセル(現存ボクセル)について次の処理を行う。
 - 各カメラの可視性判定スクリーンに全現存ボクセルを各ID色で描画(ハードウェアによるZバッファレンダリングなどで効率化)
 - 各ボクセルVを各カメラCの可視性判定スクリーンに投影
 - 投影点の画素色がボクセルVのID色と同じ → カメラCから可視
 - 異なる → カメラCから不可視



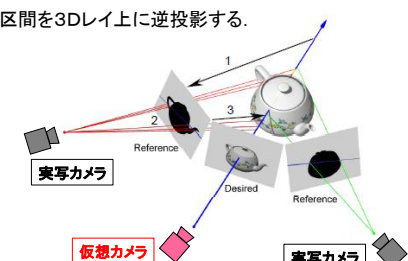
物体輪郭を利用する方法 イメージベース手法

- W. Matusik, C. Buehler, R. Raskar, S. Gortler, L. McMillan, "Image-Based Visual Hulls", SIGGRAPH 2000
 - 複数のカメラ映像からリアルタイムで自由視点映像を生成する。
 - 各カメラ映像の物体輪郭からビジュアルハルを定義する。
 - ビジュアルハルの3次元形状モデルは生成せず、自由視点画像の各画素に映るビジュアルハル上の点と色を効率的に求める。



物体輪郭を利用する方法 イメージベース手法

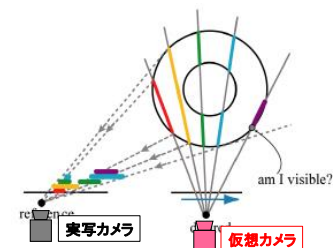
- 自由視点画像の各画素に対するビジュアルハル上の最前点の決定
 - ① 自由視点画像の注目画素に対する3Dレイを参照画像上に投影する。
 - ② 参照画像上に投影された2Dレイが物体輪郭の内部に入る区間を求める。
 - ③ 2Dレイの物体輪郭内部区間を3Dレイ上に逆投影する。
 - ④ 上記の処理を全参照画像について行い、3Dレイ上で全ての物体輪郭内部区間の共通区間(ビジュアルハルの内部区間)を求める。その区間上で最も仮想カメラに近い点が注目画素に対するビジュアルハル上の最前点となる。



W. Matusik, C. Buehler, R. Raskar, S. Gortler, L. McMillan, "Image-Based Visual Hulls", SIGGRAPH 2000

物体輪郭を利用する方法 イメージベース手法

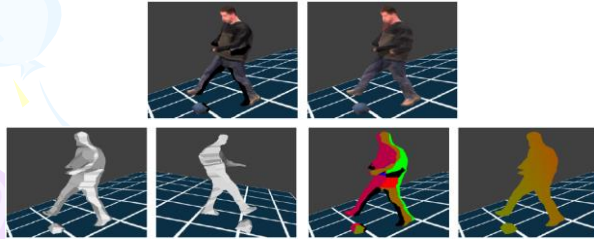
- ビジュアルハル上の最前点の色の決定
 - ① 各参照画像に対して、自由視点画像の各3Dレイのビジュアルハル内部区間を実写カメラに対してnear-to-far(近くから遠くへ)順序で投影する。
 - ② 参照画像上に投影された各ビジュアルハル内部区間の重なりから、自由視点画像の各画素に対するビジュアルハル上の最前点が参照画像上で可視かどうかを決定する。
 - ③ 各最前点について、可視である参照画像上の色を用いて、自由視点画像上の色を決定する。



W. Matusik, C. Buehler, R. Raskar, S. Gortler, L. McMillan, "Image-Based Visual Hulls", SIGGRAPH 2000

物体輪郭を利用する方法 ポリゴンベース手法

- W. Matusik, C. Buehler, L. McMillan, "Polyhedral Visual Hulls for Real-Time Rendering", 2001
- 複数のカメラ映像から、ポリゴンモデルによってビジュアルハルを構築し、自由視点映像をリアルタイムで表示する。



W. Matusik, C. Buehler, L. McMillan, "Polyhedral Visual Hulls for Real-Time Rendering", 2001

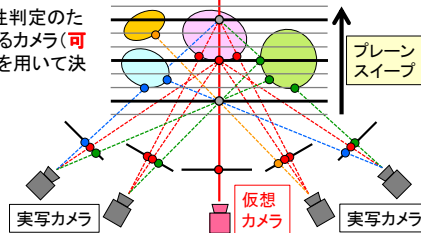


イメージ・ビデオベースCG技術の分類

出力		3次元形状モデル			2.5次元モデル	仮想カメラからの自由視点画像
		ボクセル	ポリゴン	点群	仮想カメラからのデプス画像	
参照画像からの入力						
物体輪郭(シルエット)		視積交差法	Polyhedral visual hull			Image-based visual hull
画素	色	ボクセルカラーリング			プレーンスweep ステレオマッチング	ビュー補間 ライトフィールド ライトフィールドレンダリング, ルミグラフィ フォーカルプレーン手法
	デプス			デプスハル	層化デプス画像(LDI)	ポイントスプラッティング

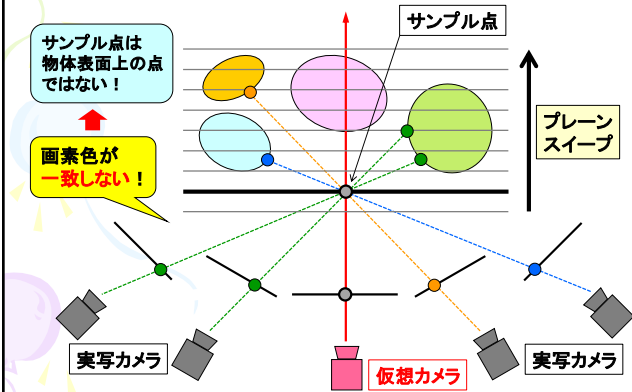
プレーンスweep

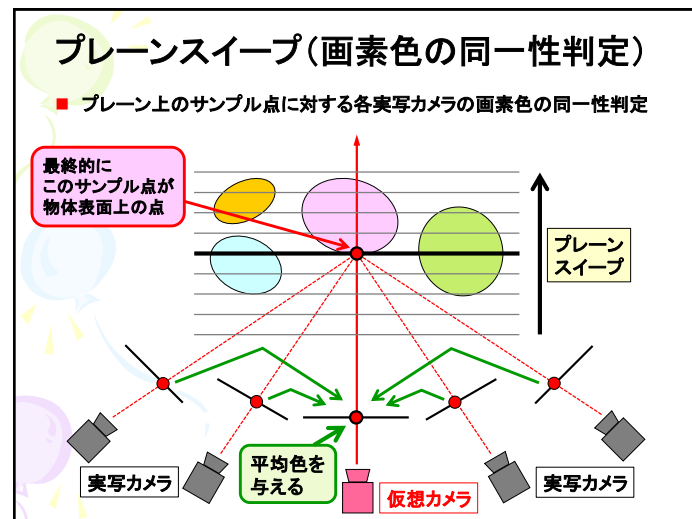
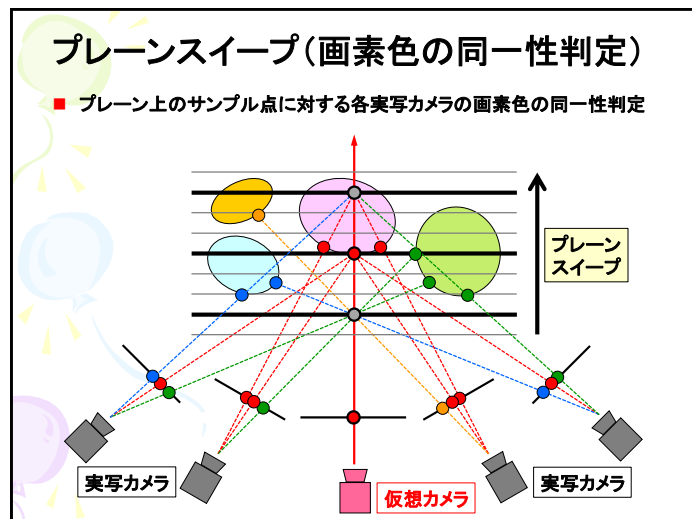
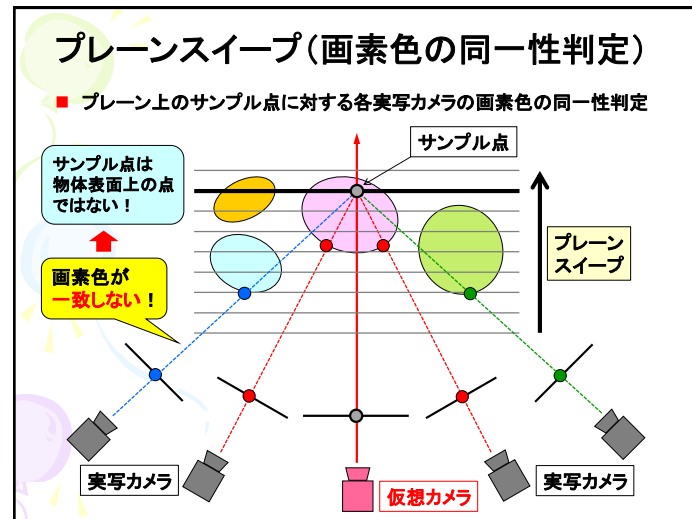
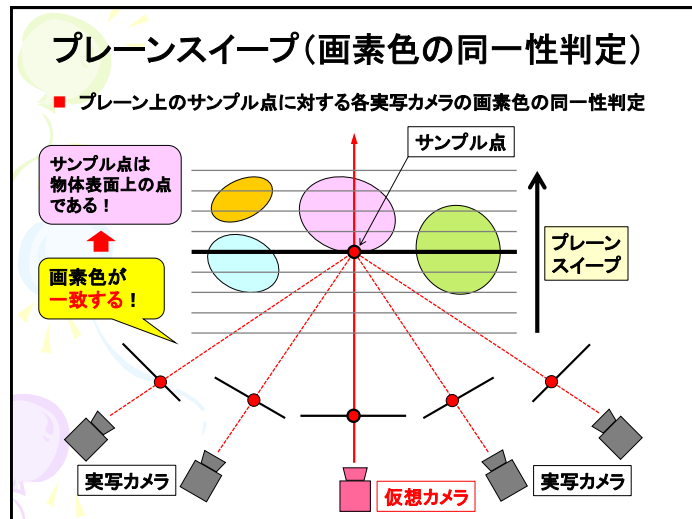
- 仮想カメラのスクリーン(自由視点画像)に平行な**プレーン(平面)**を **near-to-far(近から遠くへ)**の順序で**sweep(移動)**させ、自由視点画像の各画素に映る**物体表面点の色とデプス(奥行き)**を求める。
- プレーン位置(デプス)は離散的に等間隔に配置する。
- 自由視点画像の各画素に投影される**プレーン上のサンプル点**に対して、各実写カメラのスクリーン(参照画像)上への投影位置における**画素色の同一性判定**により**物体表面点**かどうかを決定。
- 正確な画素色の同一性判定のため、サンプル点が見えるカメラ(**可視カメラ**)を**可視マップ**を用いて決定する。



プレーンスweep(画素色の同一性判定)

- プレーン上のサンプル点に対する各実写カメラの画素色の同一性判定

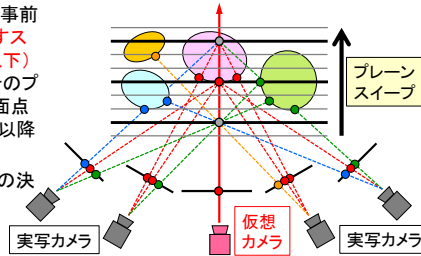




プレーンスイープ(画素色の同一性判定)

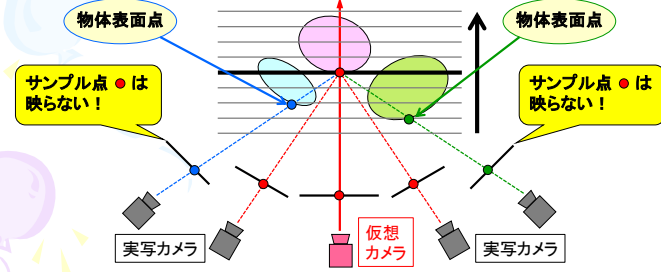
- プレーン上のサンプル点に対する各実写カメラの画素色の同一性判定
- **画素色の同一性を計るスコア**として、各実写カメラの画素色の**分散値**を用いることが多い。分散値:小 → 同一性スコア:高
- **物体表面点の決定方法**には、次のいずれかが用いられる。

- ① 全てのプレーン位置でスコアを求め、**最高スコア(最小分散値)**となるプレーン位置に物体表面点があると判定する。
- ② プレーンスイープ中に事前に決めた**閾値を満たすスコア(分散値が閾値以下)**が得られた時点で、そのプレーン位置に物体表面点があると判定し、それ以降は処理しない。
← 効率的だが、閾値の決定が難しい。



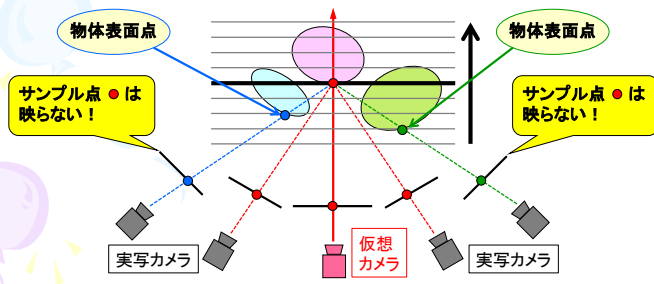
プレーンスイープ(可視カメラの決定)

- **可視マップを用いた可視カメラの決定**
- 特に、実写カメラどうしの距離が大きな場合、カメラごとに、他の物体や(その物体自身の)他の部分で隠れて見えない物体表面点が生じる。
- プレーン上のサンプル点に対する画素色の同一性判定の際、その**サンプル点が見える実写カメラ(可視カメラ)**の画素色だけを使う必要あり。



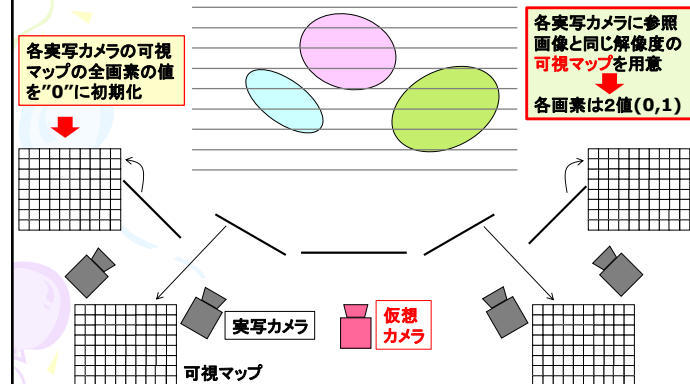
プレーンスイープ(可視カメラの決定)

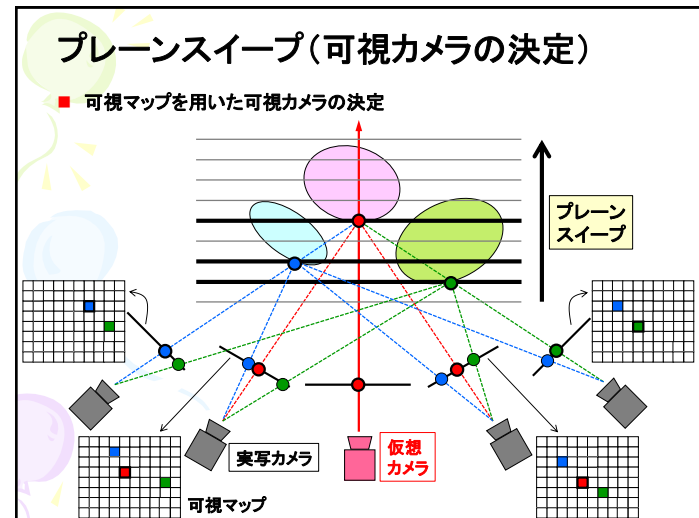
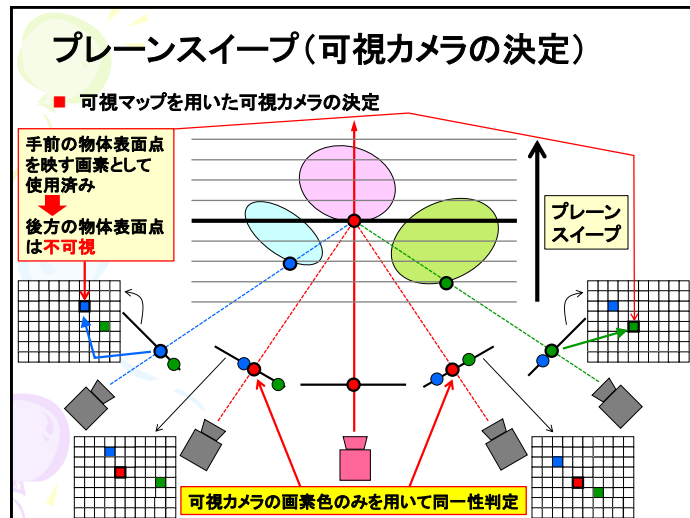
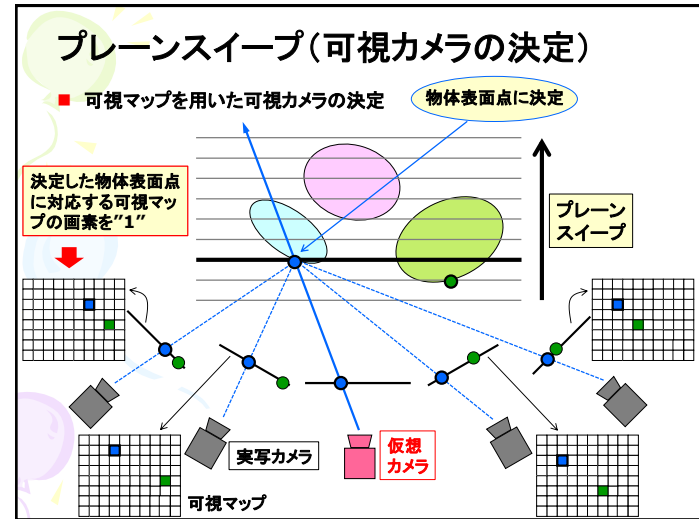
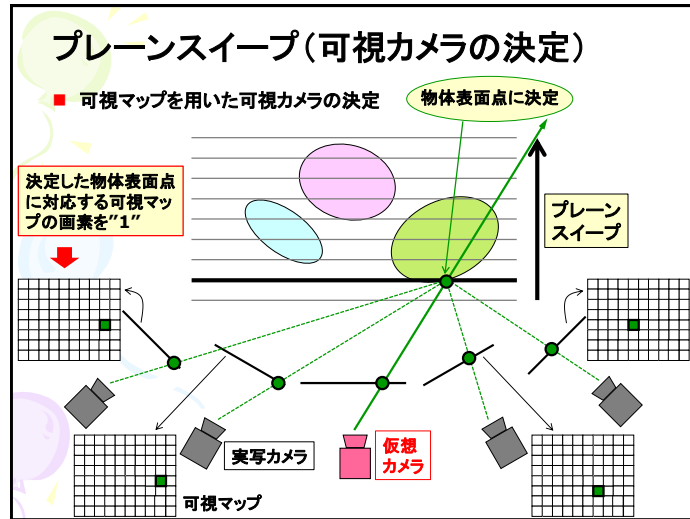
- **可視マップを用いた可視カメラの決定**
- プレーンのスイープ中に**物体表面点**を決定していく必要あり。
→ 物体表面点の決定方法として、**前述の①は使えない**、→ ②を使う。
- サンプル点の可視カメラを決定するため**可視マップ(visibility map, occlusion map)**を用いる。



プレーンスイープ(可視カメラの決定)

- **可視マップを用いた可視カメラの決定**





プレーンスweep(問題点)

■ 参照画像の離散的画素による問題

- サンプル点を投影した実写カメラのスクリーン(参照画像)上の投影位置が画素の中心からずれるため、同一性判定に用いる画素色が不正確になる。

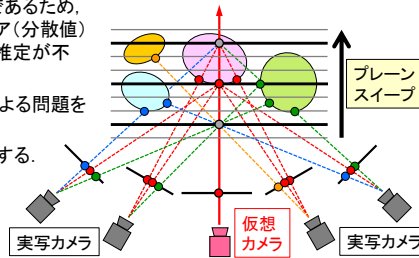
■ 離散的プレーン位置による問題

- 物体表面点が正確にプレーン上に位置することは稀であるため、画素色の同一性スコア(分散値)による物体表面点の推定が不正確になる。

- 上記の離散的画素による問題をより大きくする。

- プレーン間隔を小さくする。

- 上記の問題は軽減
- 処理時間は増大



プレーンスweep

“Undersampled Light Field Rendering by a Plane Sweep”

- Y. Liu, G. Chen, N. Max, C. Hofsetz, P. McGuinness, “Undersampled Light Field Rendering by a Plane Sweep”, Computer Graphics Forum, vol.25, no.2, 2006.

- サンプル点の投影のためのホモグラフィ行列の利用
- 物体表面点の決定のためのZバッファの利用
- 計算量の削減のためのビジュアルハルの利用
- 同一性スコアのためのテンプレートの利用
- 同一性スコアの計算方法

Y. Liu, G. Chen, N. Max, C. Hofsetz, P. McGuinness, “Undersampled Light Field Rendering by a Plane Sweep”, 2006

プレーンスweep

“Undersampled Light Field Rendering by a Plane Sweep”

■ サンプル点の投影のためのホモグラフィ行列の利用

- ホモグラフィ行列 H により、仮想カメラの自由視点画像の画素 P_t と実写カメラの参照画像の画素 P_s の2次元座標値を直接的に対応付ける。
- プレーン上のサンプル点の3次元座標値を扱う必要がない。
- 行列 H は、仮想カメラと実写カメラのスクリーン、ならびに、プレーンの位置関係から決定される。

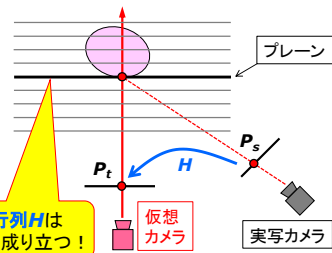
$$P_t[x_t, y_t], P_s[x_s, y_s]$$

$$\vec{P}_t = H \vec{P}_s$$

$$\begin{bmatrix} x_t' \\ y_t' \\ w_t \end{bmatrix} = \begin{bmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} x_s \\ y_s \\ 1 \end{bmatrix}$$

$$x_t = \frac{x_t'}{w_t}, y_t = \frac{y_t'}{w_t}$$

ホモグラフィ行列 H は平面に対して成り立つ!



プレーンスweep

“Undersampled Light Field Rendering by a Plane Sweep”

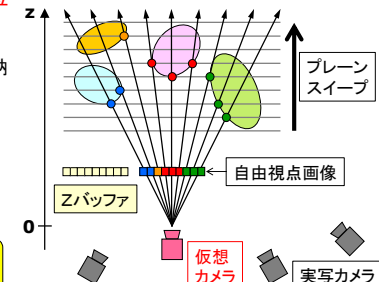
■ 物体表面点の決定のためのZバッファの利用

- 通常のZバッファレンダリングでは、各物体の処理ごとに、Zバッファの各画素に最も近い物体表面点までのZ値(最小Z値)を格納するように更新。

- この手法では、各プレーン位置の処理ごとに、Zバッファの各画素に同一性スコアの最高値(最小分散値)を格納するように更新。

最終的に、Zバッファの各画素について、最高スコア(最小分散値)となったプレーン位置に物体表面点があるものと判定。

Zバッファレンダリングはハードウェアにより高速実行できる!



プレーンスweep

“Undersampled Light Field Rendering by a Plane Sweep”

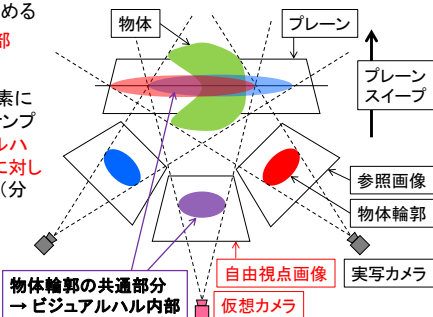
■ 計算量の削減のためのビジュアルハルの利用

- 各実写カメラの参照画像上の物体輪郭をプレーン上に投影し、共通部分を求める

→ **ビジュアルハル内部**
→ 物体を内包

- 自由視点画像の各画素に対するプレーン上のサンプル点のうち、**ビジュアルハル内部のサンプル点に対してのみ**、同一性スコア(分散値)を計算

→ 計算量の削減



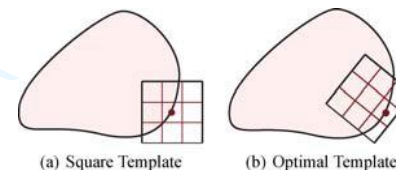
プレーンスweep

“Undersampled Light Field Rendering by a Plane Sweep”

■ 同一性スコアためのテンプレートの利用

- 正確に画素色の同一性を判定するスコアを計算するため、各実写カメラの参照画像上で、プレーン上のサンプル点が投影される画素だけでなく、その周囲の画素も含めた**テンプレート**を用いて、**複数の画素色が2次元的に並ぶパターン**の類似度を計るスコアを計算する。

- 物体の境界部分など、急にデプスが変化する部分に対応するため、**形と大きさの異なる複数のテンプレート**を用意する。



(a) Square Template

(b) Optimal Template

Y. Liu, G. Chen, N. Max, C. Hofsetz, P. McGuinness, “Undersampled Light Field Rendering by a Plane Sweep”, 2006

プレーンスweep

“Undersampled Light Field Rendering by a Plane Sweep”

■ 同一性スコアの計算方法

- 参照画像ごとの単一画素を用いる場合

- n 個の画像 $I_i (i = 1, \dots, n)$ の画素色 c_i の分散値

$$\bar{c} = \frac{1}{n} \sum_{i=1}^n c_i, \quad \text{var}[c_i] = \frac{1}{n} \sum_{i=1}^n |c_i - \bar{c}|^2$$

画素色がグレースケールの場合

$$c_i = c_i, \quad \bar{c} = \bar{c}$$

$$|c_i - \bar{c}|^2 = (c_i - \bar{c})^2$$

画素色がカラーの場合

$$c_i = [r_i, g_i, b_i], \quad \bar{c} = [\bar{r}, \bar{g}, \bar{b}]$$

$$|c_i - \bar{c}|^2 = (r_i - \bar{r})^2 + (g_i - \bar{g})^2 + (b_i - \bar{b})^2$$

プレーンスweep

“Undersampled Light Field Rendering by a Plane Sweep”

■ 同一性スコアの計算方法

- 参照画像ごとのテンプレート(複数画素)を用いる場合

- 2つの画像 I_1, I_2 間でのスコア

画像 I_1, I_2 上のテンプレート内の m 個の画素色 $c_{1,k}, c_{2,k} (k = 1, \dots, m)$

画素色がグレースケールの場合

$$c_{1,k} = c_{1,k}, \quad c_{2,k} = c_{2,k}$$

$$|c_{1,k}| = |c_{1,k}|, \quad |c_{2,k}| = |c_{2,k}|, \quad |c_{1,k} - c_{2,k}| = |c_{1,k} - c_{2,k}|$$

$$|c_{1,k}|^2 = (c_{1,k})^2, \quad |c_{2,k}|^2 = (c_{2,k})^2, \quad |c_{1,k} - c_{2,k}|^2 = (c_{1,k} - c_{2,k})^2$$

$$t_1 = [c_{1,1}, c_{1,2}, \dots, c_{1,m}] = [c_{1,1}, c_{1,2}, \dots, c_{1,m}]$$

$$t_2 = [c_{2,1}, c_{2,2}, \dots, c_{2,m}] = [c_{2,1}, c_{2,2}, \dots, c_{2,m}]$$

プレーンスweep

“Undersampled Light Field Rendering by a Plane Sweep”

- 同一性スコアの計算方法
 - 参照画像ごとのテンプレート(複数画素)を用いる場合
 - 2つの画像 I_1, I_2 の間でのスコア
- 画像 I_1, I_2 上のテンプレート内の m 個の画素色 $c_{1,k}, c_{2,k}(k = 1, \dots, m)$

画素色がカラーの場合

$$c_{1,k} = [r_{1,k}, g_{1,k}, b_{1,k}], \quad c_{2,k} = [r_{2,k}, g_{2,k}, b_{2,k}]$$

$$|c_{1,k}| = |r_{1,k}| + |g_{1,k}| + |b_{1,k}|, \quad |c_{2,k}| = |r_{2,k}| + |g_{2,k}| + |b_{2,k}|$$

$$|c_{1,k} - c_{2,k}| = |r_{1,k} - r_{2,k}| + |g_{1,k} - g_{2,k}| + |b_{1,k} - b_{2,k}|$$

$$|c_{1,k}|^2 = (r_{1,k})^2 + (g_{1,k})^2 + (b_{1,k})^2, \quad |c_{2,k}|^2 = (r_{2,k})^2 + (g_{2,k})^2 + (b_{2,k})^2$$

$$|c_{1,k} - c_{2,k}|^2 = (r_{1,k} - r_{2,k})^2 + (g_{1,k} - g_{2,k})^2 + (b_{1,k} - b_{2,k})^2$$

$$t_1 = [c_{1,1}, c_{1,2}, \dots, c_{1,m}] = [r_{1,1}, g_{1,1}, b_{1,1}, r_{1,2}, g_{1,2}, b_{1,2}, \dots, r_{1,m}, g_{1,m}, b_{1,m}]$$

$$t_2 = [c_{2,1}, c_{2,2}, \dots, c_{2,m}] = [r_{2,1}, g_{2,1}, b_{2,1}, r_{2,2}, g_{2,2}, b_{2,2}, \dots, r_{2,m}, g_{2,m}, b_{2,m}]$$

プレーンスweep

“Undersampled Light Field Rendering by a Plane Sweep”

- 同一性スコアの計算方法
- 参照画像ごとのテンプレート(複数画素)を用いる場合
 - 2つの画像 I_1, I_2 の間でのスコア

The sum of absolute difference (SAD) ← マンハッタン距離

$$f_{SAD}(t_1, t_2) = \sum_{k=1}^m |c_{1,k} - c_{2,k}| = |t_1 - t_2|$$

The sum of squared distance (SSD) ← ユークリッド距離の2乗

$$f_{SSD}(t_1, t_2) = \sum_{k=1}^m |c_{1,k} - c_{2,k}|^2 = |t_1 - t_2|^2$$

The normalized cross-correlation (NCC) ← 正規化された内積

$$f_{NCC}(t_1, t_2) = \frac{\sum_{k=1}^m c_{1,k} \cdot c_{2,k}}{\sqrt{\sum_{k=1}^m |c_{1,k}|^2} \sqrt{\sum_{k=1}^m |c_{2,k}|^2}} = \frac{t_1 \cdot t_2}{|t_1| |t_2|} = \cos \theta$$

プレーンスweep

“Undersampled Light Field Rendering by a Plane Sweep”

- 同一性スコアの計算方法
- 参照画像ごとのテンプレート(複数画素)を用いる場合
 - 2つの画像 I_1, I_2 の間でのスコア

The zero-mean normalized cross-correlation (ZNCC) ← 零平均の正規化内積

$$f_{ZNCC}(t_1, t_2) = \frac{\sum_{k=1}^m (c_{1,k} - \bar{c}_1) \cdot (c_{2,k} - \bar{c}_2)}{\sqrt{\sum_{k=1}^m |c_{1,k} - \bar{c}_1|^2} \sqrt{\sum_{k=1}^m |c_{2,k} - \bar{c}_2|^2}}$$

$$= \frac{(t_1 - \bar{t}_1) \cdot (t_2 - \bar{t}_2)}{|t_1 - \bar{t}_1| |t_2 - \bar{t}_2|} = \cos \theta$$

画素色がグレースケールの場合

$$\bar{c}_1 = \bar{c}_1 = \frac{1}{m} \sum_{k=1}^m c_{1,k}, \quad \bar{c}_2 = \bar{c}_2 = \frac{1}{m} \sum_{k=1}^m c_{2,k}$$

$$\bar{t}_1 = [\bar{c}_1, \bar{c}_1, \dots, \bar{c}_1] = [\bar{c}_1, \bar{c}_1, \dots, \bar{c}_1], \quad \bar{t}_2 = [\bar{c}_2, \bar{c}_2, \dots, \bar{c}_2] = [\bar{c}_2, \bar{c}_2, \dots, \bar{c}_2]$$

プレーンスweep

“Undersampled Light Field Rendering by a Plane Sweep”

- 同一性スコアの計算方法
- 参照画像ごとのテンプレート(複数画素)を用いる場合
 - 2つの画像 I_1, I_2 の間でのスコア

画素色がカラーの場合

$$\bar{c}_1 = [\bar{r}_1, \bar{g}_1, \bar{b}_1] = \frac{1}{m} \sum_{k=1}^m c_{1,k} = \frac{1}{m} \sum_{k=1}^m [r_{1,k}, g_{1,k}, b_{1,k}]$$

$$= \left[\frac{1}{m} \sum_{k=1}^m r_{1,k}, \frac{1}{m} \sum_{k=1}^m g_{1,k}, \frac{1}{m} \sum_{k=1}^m b_{1,k} \right]$$

$$\bar{c}_2 = [\bar{r}_2, \bar{g}_2, \bar{b}_2] = \frac{1}{m} \sum_{k=1}^m c_{2,k} = \frac{1}{m} \sum_{k=1}^m [r_{2,k}, g_{2,k}, b_{2,k}]$$

$$= \left[\frac{1}{m} \sum_{k=1}^m r_{2,k}, \frac{1}{m} \sum_{k=1}^m g_{2,k}, \frac{1}{m} \sum_{k=1}^m b_{2,k} \right]$$

$$\bar{t}_1 = [\bar{c}_1, \bar{c}_1, \dots, \bar{c}_1] = [\bar{r}_1, \bar{g}_1, \bar{b}_1, \bar{r}_1, \bar{g}_1, \bar{b}_1, \dots, \bar{r}_1, \bar{g}_1, \bar{b}_1]$$

$$\bar{t}_2 = [\bar{c}_2, \bar{c}_2, \dots, \bar{c}_2] = [\bar{r}_2, \bar{g}_2, \bar{b}_2, \bar{r}_2, \bar{g}_2, \bar{b}_2, \dots, \bar{r}_2, \bar{g}_2, \bar{b}_2]$$

プレーンスweep

“Undersampled Light Field Rendering by a Plane Sweep”

- 同一性スコアの計算方法
- 参照画像ごとのテンプレート(複数画素)を用いる場合
 - n 個の画像 $t_i (i = 1, \dots, n)$ の間でのスコア

$$t_{ref} = \frac{1}{n} \sum_{i=1}^n t_i$$

全参照画像の
テンプレートの平均

$$f_{mSAD} = \sum_{i=1}^n f_{SAD}(t_i, t_{ref})$$

$$f_{mSSD} = \sum_{i=1}^n f_{SSD}(t_i, t_{ref})$$

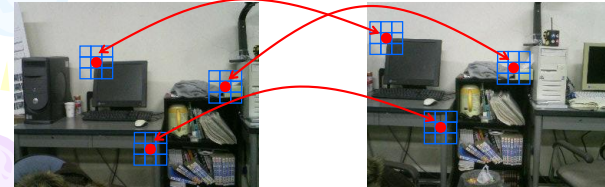
$$f_{mNCC} = \sum_{i=1}^n f_{NCC}(t_i, t_{ref})$$

$$f_{mZNCC} = \sum_{i=1}^n f_{ZNCC}(t_i, t_{ref})$$



ステレオマッチング

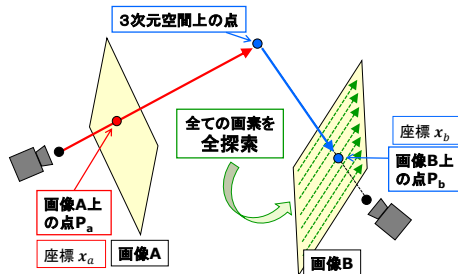
- 2つの画像間でシーン中の同じ部分を映している画素を対応付ける。
→ カメラ校正ができていれば、三角測量の原理で、画像の**画素ごとのデプス(デプス画像)**が得られる。
- 通常、**目的とする画素を中心としたテンプレート**を用いて、近傍の画素領域が最も類似する色パターンを持つ画素どうしを対応付ける。



上図では、説明の都合上、テンプレートを大きく描いている。

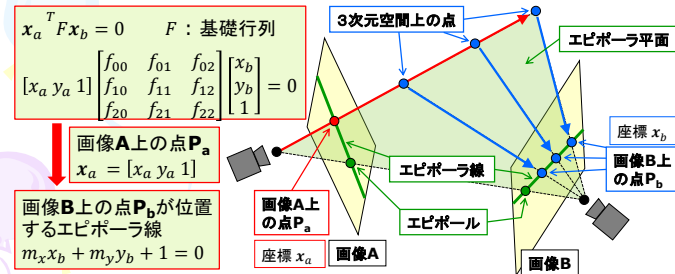
ステレオマッチング

- 一辺の画素数が $O(N)$ の画像A, Bについて、画像Aの画素 P_a に画像Bの画素 P_b を対応付ける。
- シンプルに全探索を行う方法
画像Aの1つの画素を画像Bの**全ての画素**とテンプレート比較する。この場合、画像Bの $O(N^2)$ 個の画素と比較が必要。画像Aの全画素を対応付けるためには $O(N^4)$ の計算量が必要。



ステレオマッチング

- 一辺の画素数が $O(N)$ の画像A, Bについて、画像Aの画素 P_a に画像Bの画素 P_b を対応付ける。
- エピポーラ幾何を利用して効率的に探索を行う方法
画像Aの1つの画素を画像Bの**エピポーラ線上の画素**とテンプレート比較する。この場合、画像Bの $O(N)$ 個の画素と比較が必要。画像Aの全画素を対応付けるためには $O(N^3)$ の計算量が必要。



$$x_a^T F x_b = 0 \quad F: \text{基礎行列}$$

$$\begin{bmatrix} f_{00} & f_{01} & f_{02} \\ f_{10} & f_{11} & f_{12} \\ f_{20} & f_{21} & f_{22} \end{bmatrix} \begin{bmatrix} x_b \\ y_b \\ 1 \end{bmatrix} = 0$$

$$\text{画像A上の点 } P_a \\ x_a = [x_a \ y_a \ 1]$$

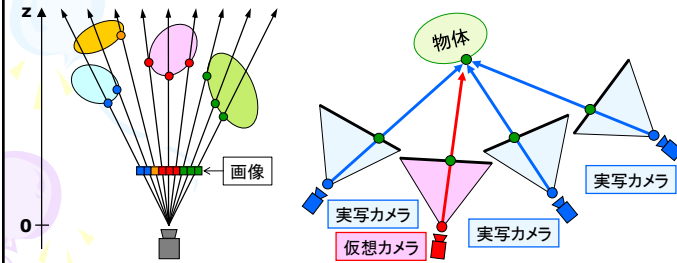
$$\text{画像B上の点 } P_b \text{ が位置} \\ \text{するエピポーラ線} \\ m_x x_b + m_y y_b + 1 = 0$$

イメージ・ビデオベースCG技術の分類

出力	3次元形状モデル			2.5次元モデル	仮想カメラからの自由視点画像
	ボクセル	ポリゴン	点群	仮想カメラからのデプス画像	
参照画像からの入力					
物体輪郭(シルエット)	視体積交差法	Polyhedral visual hull			Image-based visual hull
画素色	ボクセルカラーリング			プレーンスイープ ステレオマッチング	ビュー補間 ライトフィールド ライトフィールド ドレンダリング ルミグラフ フォーカル プレーン手法
デプス			デプスハル	層化デプス画像(LDI)	ポイントスプラッティング

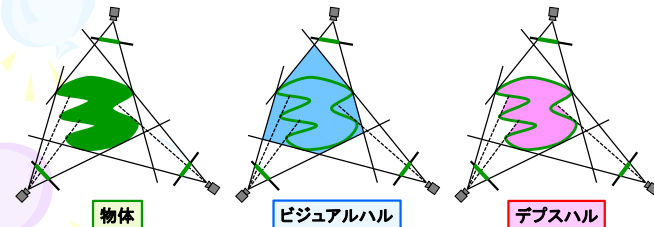
画素のデプスを利用する方法

- 画素ごとにカメラ視点から物体表面までの**デプス(奥行き, Z値)**を持つ**デプス画像**を利用する。
 - 参照画像に対するデプス画像
 - 自由視点画像に対するデプス画像
- 物体表面を構成する**3次元点群**を扱う。



デプスハル(Depth Hull)

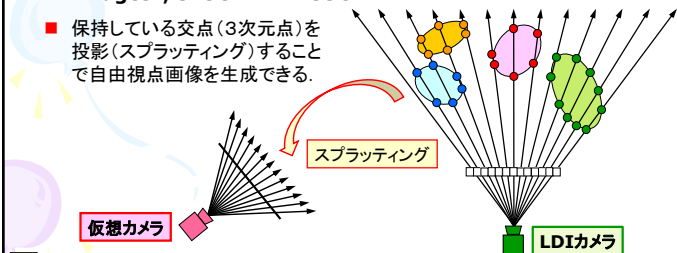
- **デプスハル(depth hull)**は、参照画像に付加された**デプス画像**を用いることで、物体輪郭のみから構築される**ビジュアルハル(visual hull)**に比べて、物体の3次元形状の近似精度を高めたモデルである。
- A. Bogomjakov, C. Gotsman, "Reduced Depth and Visual Hulls of Complex 3D Scenes", Computer Graphics Forum, Vol.27(2), 2008.



A. Bogomjakov, C. Gotsman, "Reduced Depth and Visual Hulls of Complex 3D Scenes", 2008

層化デプス画像(Layered Depth Image)

- **Zバッファ**は、画素ごとのレイに対して、カメラに最も近い物体表面との交点を保持するデータ構造。
- **層化デプス画像(Layered Depth Image : LDI)**は、画素ごとのレイに対して、全ての物体表面との交点を保持するデータ構造。
- J. Shade, S. Gortler, L. He, R. Szeliski, "Layered Depth Images", SIGGRAPH 1998.
- 保持している交点(3次元点)を投影(スプラッティング)することで自由視点画像を生成できる。



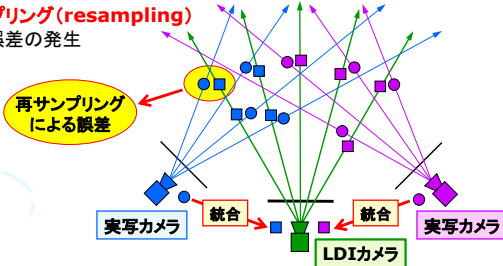
J. Shade, S. Gortler, L. He, R. Szeliski, "Layered Depth Images", 1998

層化デプス画像(Layered Depth Image)

LDIの生成方法

- 仮想世界では、形状モデルに対して、レイトレーシング法のような方法で、画素ごとのレイに対して全ての物体表面との交点が計算できる。
- 実世界では、複数の実写カメラに対するデプス画像(画素ごとのカメラに最も近い物体表面までのデプス)を一つのLDIに統合する。

→ 再サンプリング(resampling)による誤差の発生



J. Shade, S. Gortler, L. He, R. Szeliski, "Layered Depth Images", 1998

ポイントスプラッティング

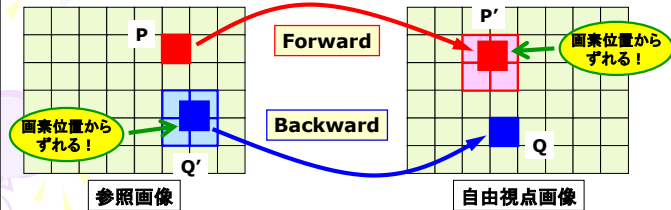
- 参照画像と自由視点画像の間で画素位置を対応付ける。

Forward mapping (splatting)

参照画像の画素Pを基準とし、画素Pを自由視点画像上に投影する。投影された位置にある(複数の)画素P'に画素Pの色を分配する。

Backward mapping

自由視点画像の画素Qを基準とし、画素Qの位置に投影される参照画像の(複数の)画素Q'を求める。画素Q'の色から画素Qの色を求める。



イメージ・ビデオベースCG技術の分類

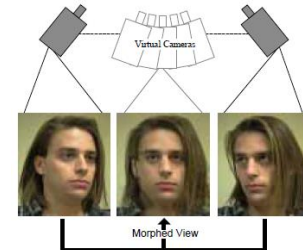
出力	3次元形状モデル			2.5次元モデル	仮想カメラからの自由視点画像
	ボクセル	ポリゴン	点群	仮想カメラからのデプス画像	
参照画像からの入力					
物体輪郭(シルエット)	視体積交差法	Polyhedral visual hull			Image-based visual hull
画素色	ボクセルカラーリング			プレーンスイープ ステレオマッチング	ビュー補間 ライトフィールド ライトフィールドレンダリング、ルミグラフ フォーカルプレーン手法
	デプス		デプスハル	層化デプス画像(LDI)	ポイントスプラッティング

ビュー補間

- ビュー補間(view interpolation)は、複数の参照画像の画素の位置と色を補間することで、自由視点画像の各画素の適切な色を決定する。

- 実写カメラと仮想カメラの相対的な位置関係などを考慮して、参照画像と自由視点画像の間の画素位置の対応関係を求める。

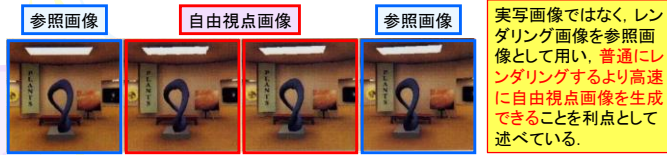
- 画素のマッピング(forward mapping, backward mapping)やスプラッティング(splatting)などにより、参照画像の画素色を投影して自由視点画像の画素色を決定する。



S. Seitz, C. Dyer, "View Morphing", 1996

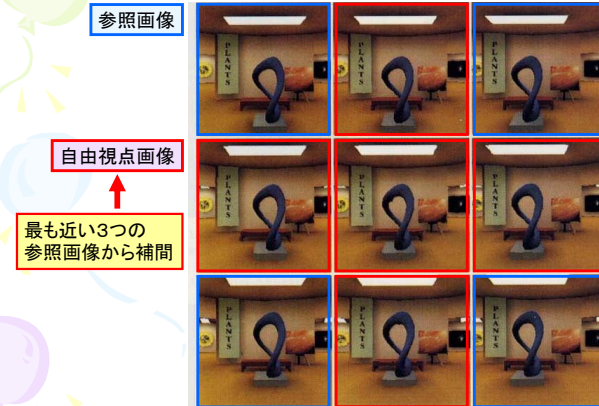
View Interpolation for Image Synthesis

- S. E. Chen, L. Williams, "View Interpolation for Image Synthesis", SIGGRAPH 1993
- 実写カメラ間の相対位置, ならびに, 各参照画像の画素ごとのデプス(デプス画像)が既知とする.
- 参照画像の各画素の3次元点を求め, 自動的に参照画像間の画素の対応付けを行う.
- 参照画像間の画素の変位を補間し, その画素(色)を自由視点画像上に **forward mapping** で投影する.



S. E. Chen, L. Williams, "View Interpolation for Image Synthesis", 1993

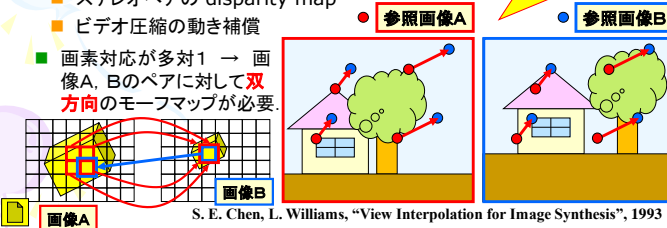
View Interpolation for Image Synthesis



S. E. Chen, L. Williams, "View Interpolation for Image Synthesis", 1993

View Interpolation for Image Synthesis

- 参照画像間の画素の対応付け
- モーフマップ
- 参照画像Aから参照画像Bへの **forward mapping** をあらわす.
- 参照画像Aの各画素の**3Dオフセットベクトル**(参照画像Bの対応画素への変位ベクトル)を登録する.
- 以下に類似する.
 - オプティカルフローのオフセットベクトル場
 - ステレオペアの disparity map
 - ビデオ圧縮の動き補償
- 画素対応が多対1 → 画像A, Bのペアに対して**双方向**のモーフマップが必要.



S. E. Chen, L. Williams, "View Interpolation for Image Synthesis", 1993

View Interpolation for Image Synthesis

- 3Dオフセットベクトル

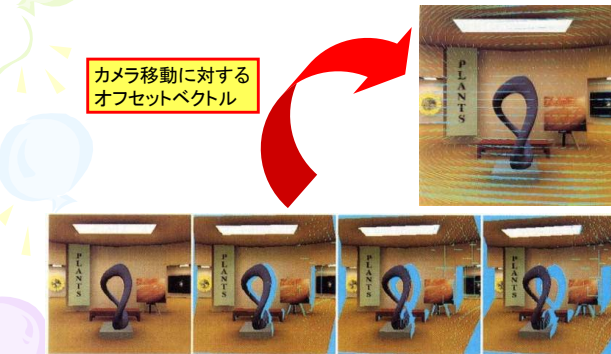


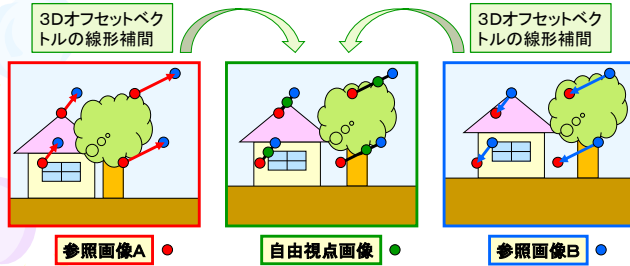
Fig. 3 A source image viewed from a camera rotated to the right.

S. E. Chen, L. Williams, "View Interpolation for Image Synthesis", 1993

View Interpolation for Image Synthesis

■ 3Dオフセットベクトルの補間

- 参照画像ペアの3Dオフセットベクトルを線形補間し、補間されたベクトル分だけ参照画像画素を移動して自由視点画像上にマップする。
→ 通常の透視投影変換(同次座標による 4×4 行列の積で計算量が大)による3次元空間上での計算による画素の移動を効率的に近似する。



S. E. Chen, L. Williams, "View Interpolation for Image Synthesis", 1993

View Interpolation for Image Synthesis

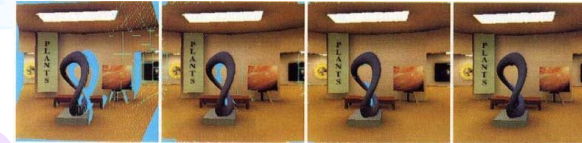
■ 自由視点画像の合成

- マップされた参照画像画素の重なり

視点移動で画像上の領域が縮むために発生。異なる参照画像画素が自由視点画像上の同じ画素位置にマップされた場合、Zバッファで前後関係を判定。

- 参照画像画素がマップされない穴領域の発生

視点移動による領域の広がりやオクルージョンのために発生。周囲の画素の色により補間する、または、複数の参照画像からのマップで埋める。



1つの参照画像による穴領域

2つの参照画像による穴領域

2つの参照画像(視点に近い)による穴領域

周囲の画素の色による補間

S. E. Chen, L. Williams, "View Interpolation for Image Synthesis", 1993

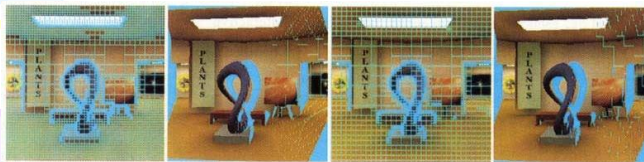
View Interpolation for Image Synthesis

■ 最適化

- モーフマップからブロックリストを生成する。

参照画像上の近隣の画素のうち、ペアとなる参照画像上の近くにマップされる(3Dオフセットベクトルが近い)ものをブロックにまとめ、一度にマップを行う。

- 4分木によるブロック圧縮



S. E. Chen, L. Williams, "View Interpolation for Image Synthesis", 1993

View Interpolation for Image Synthesis

■ 実装

複数の参照画像間関係をグラフ構造化する。

- ノード:

参照(色)画像、デプス画像、カメラの位置と向き

- 双方向アーク:

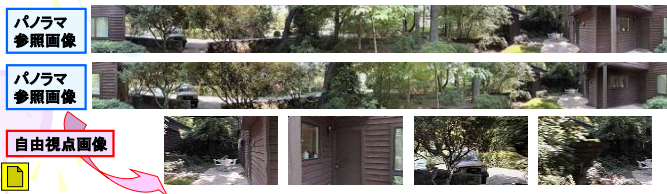
カメラ位置が近いノードを結び、ブロックリスト



S. E. Chen, L. Williams, "View Interpolation for Image Synthesis", 1993

Plenoptic Modeling: An Image-Based Rendering System

- L. McMillan, G. Bishop, "Plenoptic Modeling: An Image-Based Rendering System", SIGGRAPH 1995
- 実世界を記述する **plenoptic function** の実現を目指す。
- 複数の実写カメラ位置で **円柱投影** による **パノラマ参照画像** を生成する。
- **円柱エピポーラ幾何** により、異なる実写カメラ位置のパノラマ参照画像間の画素の対応関係を求める。
- パノラマ参照画像と画素の対応関係から、**参照画像画素の3次元点** をマップして **自由視点画像** を生成する。



Plenoptic Modeling: An Image-Based Rendering System

■ Plenoptic function

- 実世界の全てのシーンを表現する光の情報
- 光の輝度 p を次式で表現する。

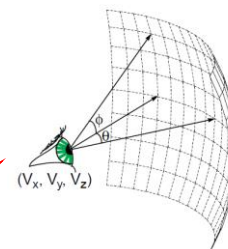
$$p = P(\theta, \phi, \lambda, V_x, V_y, V_z, t)$$

θ, ϕ : 視線方向

λ : 光の波長

V_x, V_y, V_z : 視点位置

t : 時間



これが分かれば、
実世界の全てのシーンが
CGで再現できる！



L. McMillan, G. Bishop, "Plenoptic Modeling: An Image-Based Rendering System", 1995

Plenoptic Modeling: An Image-Based Rendering System

- 円柱投影によるパノラマ参照画像の生成
- カメラ位置ごとに、ビデオカメラを **パン** (水平方向回転) し、**連続パン画像** を撮影する。
- パン画像間で幾つかの対応点を制約として与え、カメラの内部パラメータと各パン画像のパン角度を最適化問題の解として求め、カメラ視点位置を中心とした **円柱投影 (cylindrical projection)** を求める。
- 円柱投影により全パン画像から **パノラマ参照画像** を生成する。



L. McMillan, G. Bishop, "Plenoptic Modeling: An Image-Based Rendering System", 1995

Plenoptic Modeling: An Image-Based Rendering System

- 異なるカメラ視点の相対位置、ならびに、パノラマ参照画像上の画素の対応関係の決定
- 異なるカメラ視点のパノラマ参照画像上で共通に見える対応点を制約として与え、最適化問題を解くことにより、**カメラ視点の相対位置 (円柱投影の相対関係)** を求める。
- **円柱エピポーラ幾何 (cylindrical epipolar geometry)** により、異なるカメラ視点のパノラマ参照画像上の **画素の対応関係 (image flow field)** を **disparity 画像** として求める。



L. McMillan, G. Bishop, "Plenoptic Modeling: An Image-Based Rendering System", 1995

Plenoptic Modeling: An Image-Based Rendering System

- 平面エピポーラ幾何

3次元空間上の点
エビポーラ平面
画像A上の点 P_a
エビポーラ直線
画像A
エビポール
平面スクリーン
画像B
画像B上の点 P_b

L. McMillan, G. Bishop, "Plenoptic Modeling: An Image-Based Rendering System", 1995

Plenoptic Modeling: An Image-Based Rendering System

- 円柱エピポーラ幾何

3次元空間上の点
エビポーラ平面
画像A上の点 P_a
エビポーラ曲線
画像A
エビポール
円柱スクリーン
画像B
画像B上の点 P_b

L. McMillan, G. Bishop, "Plenoptic Modeling: An Image-Based Rendering System", 1995

Plenoptic Modeling: An Image-Based Rendering System

- 自由視点画像の生成
- パノラマ参照画像と画素の対応関係 (disparity画像) から、参照画像画素の3次元点を **forward mapping** で投影して自由視点画像を生成。
- 投影される3次元点の自由視点画像上での **重なり** の前後判定は、**マクミランの順序付けアルゴリズム** で効率的に解決する。実写カメラと仮想カメラの位置関係から、デプス (奥行き、Z値) によるソーティングをすることなく、正しく **back-to-front** (後ろから前へ) の順序で投影する。

パノラマ参照画像
パノラマ参照画像
自由視点画像

L. McMillan, G. Bishop, "Plenoptic Modeling: An Image-Based Rendering System", 1995

View Morphing

- S. Seitz, C. Dyer, "View Morphing", SIGGRAPH 1996
- 異なる視点位置から撮影された2つの画像に対して、歪みのない自然なモーフィングを行なう。

Morphed View

Figure 4: View Morphing in Three Steps. (1) Original images I_0 and I_1 are prewarped to form parallel views I_0' and I_1' . (2) I_0' is produced by morphing (interpolating) the prewarped images. (3) I_0' is postwarped to form I_0'' .

S. Seitz, C. Dyer, "View Morphing", 1996

View Morphing

Figure 6: View Morphing Procedure: A set of features (yellow lines) is selected in original images Z_0 and Z_1 . Using these features, the images are automatically prewarped to produce \hat{Z}_0 and \hat{Z}_1 . The prewarped images are morphed to create a sequence of in-between images, the middle of which, $\hat{Z}_{0.5}$, is shown at top-center. $\hat{Z}_{0.5}$ is interactively postwarped by selecting a quadrilateral region (marked red) and specifying its desired configuration, $Q_{0.5}$, in $\hat{Z}_{0.5}$. The postwarps for other in-between images are determined by interpolating the quadrilaterals (bottom).

S. Seitz, C. Dyer, "View Morphing", 1996

View Morphing

■ 実験例

Figure 7: Facial View Morphs. Top: morph between two views of the same person. Bottom: morph between views of two different people. In each case, view morphing captures the change in facial pose between original images Z_0 and Z_1 , conveying a natural 3D rotation.

S. Seitz, C. Dyer, "View Morphing", 1996

View Morphing

■ 実験例

Figure 9: Mona Lisa View Morph. Morphed view (center) is halfway between original image (left) and its reflection (right).

S. Seitz, C. Dyer, "View Morphing", 1996

High-quality Video View Interpolation using a Layered Representation

■ C. Zitnick, S. Kang, M. Uyttendaele, S. Winder, R. Szeliski, "High-quality Video View Interpolation using a Layered Representation", SIGGRAPH 2004

■ 複数のビデオ映像をスムーズに補間することで、高品質な自由視点映像を生成する。

Figure 1: A video view interpolation example: (a,c) synchronized frames from two different input cameras and (b) a virtual interpolated view. (d) A depth-matted object from earlier in the sequence is inserted into the video.

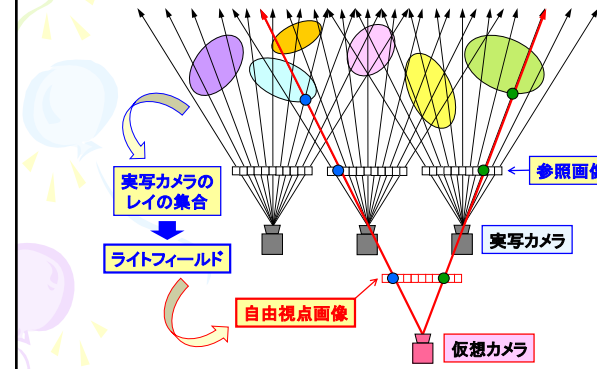
C. Zitnick, S. Kang, M. Uyttendaele, S. Winder, R. Szeliski, "High-quality Video View Interpolation using a Layered Representation", 2004

イメージ・ビデオベースCG技術の分類

出力	3次元形状モデル			2.5次元モデル	仮想カメラからの自由視点画像
参照画像からの入力	ボクセル	ポリゴン	点群	仮想カメラからのデプス画像	
物体輪郭(シルエット)	視体積交差法	Polyhedral visual hull			Image-based visual hull
画素色	ボクセルカラーリング			プレーンスイープ ステレオマッチング	ビュー補間 ライトフィールド ライトフィールドレンダリング ルミグラフ フォーカルプレーン手法
デプス			デプスハル	層化デプス画像(LDI)	ポイントスプラッチング

ライトフィールドを利用する方法

- 各実写カメラの参照画像の画素ごとのレイ(光線)
 - ライトフィールド(光の場)の構築 → 自由視点画像の生成

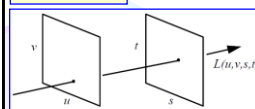


ライトフィールドレンダリング, ルミグラフ

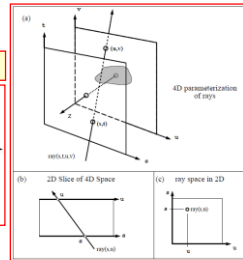
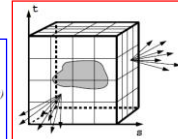
- M. Levoy, P. Hanrahan, "Light Field Rendering", SIGGRAPH 1996
- S. Gortler, R. Grzeszczuk, R. Szeliski, M. Cohen, "The Lumigraph", SIGGRAPH 1996
- 多数の参照画像から4変数関数(Plenoptic function)としてライトフィールドを構築し, その関数値を補間することで自由視点画像を生成.

3次元空間上のレイを
2平面上の4パラメータで規定

Light Field



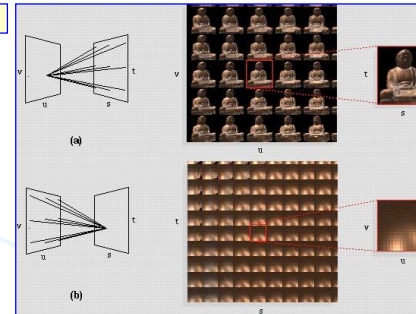
Lumigraph



ライトフィールドレンダリング, ルミグラフ

- 離散的 plenoptic function
 - 離散パラメータ値のレイに対する関数値(色値)

Light Field



M. Levoy, P. Hanrahan, "Light Field Rendering", 1996

ライトフィールドレンダリング, ルミグラフ

■ 離散的 plenoptic function

→ 離散パラメータ値のレイに対する関数値(色値)

Lumigraph

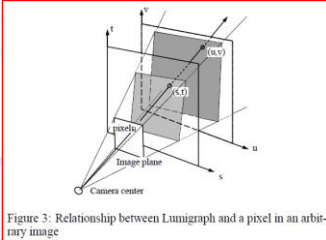


Figure 3: Relationship between Lumigraph and a pixel in an arbitrary image

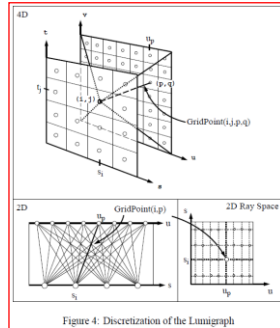


Figure 4: Discretization of the Lumigraph

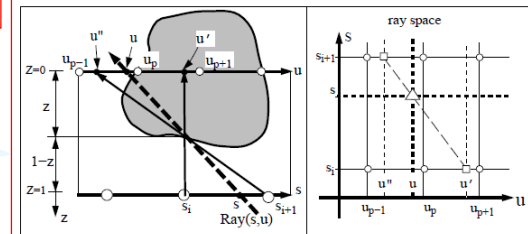
S. Gortler, R. Grzeszczuk, R. Szeliski, M. Cohen, "The Lumigraph", 1996

ライトフィールドレンダリング, ルミグラフ

■ 自由視点画像の生成

- 自由視点画像の各画素へのレイを与えるパラメータについて、**離散パラメータ値に対する関数値(色値)を補間して画素の関数値(色値)を算出**。
- 物体表面までのデプス情報などの**幾何情報(プロキシ)**を利用することで、自由視点画像の品質を高める。

Lumigraph



S. Gortler, R. Grzeszczuk, R. Szeliski, M. Cohen, "The Lumigraph", 1996

Figure 6: Depth correction of rays

ライトフィールドレンダリング, ルミグラフ

■ 幾何情報(プロキシ)の生成

視体積交差法により、参照画像の物体輪郭からビジュアルハルのボクセルモデルを生成し、3次元空間上の物体領域を近似的に求める。

Lumigraph



Figure 12: Segmented image plus volume construction

S. Gortler, R. Grzeszczuk, R. Szeliski, M. Cohen, "The Lumigraph", 1996

ライトフィールドレンダリング, ルミグラフ

■ 参照画像の撮影環境

Light Field

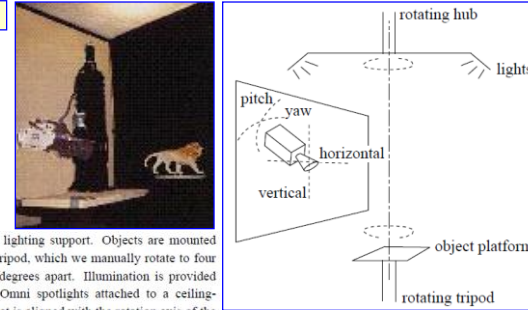


Figure 10: Object and lighting support. Objects are mounted on a Bogen fluid-head tripod, which we manually rotate to four orientations spaced 90 degrees apart. Illumination is provided by two 600W Lowell Omni spotlights attached to a ceiling-mounted rotating hub that is aligned with the rotation axis of the tripod. A stationary 6' x 6' diffuser panel is hung between the spotlights and the gantry, and the entire apparatus is enclosed in black velvet to eliminate stray light.

M. Levoy, P. Hanrahan, "Light Field Rendering", 1996

ライトフィールドレンダリング, ルミグラフ

- 参照画像の撮影環境

Lumigraph

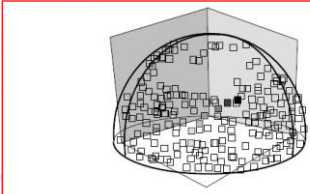


Figure 11: The user interface for the image capture stage displays the current and previous camera positions on a viewing sphere. The goal of the user is to "paint" the sphere.

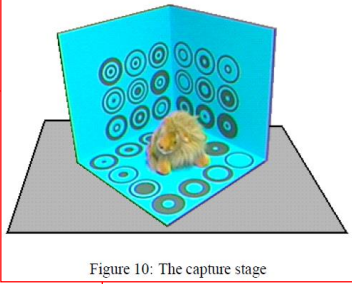


Figure 10: The capture stage

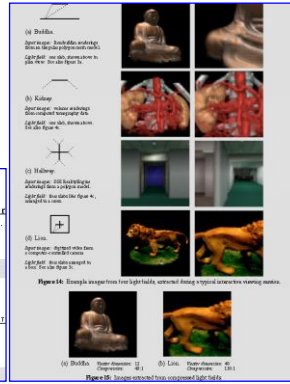
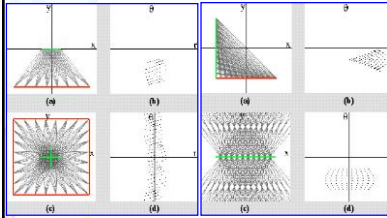
S. Gortler, R. Grzeszczuk, R. Szeliski, M. Cohen, "The Lumigraph", 1996

ライトフィールドレンダリング, ルミグラフ

- 自由視点画像の例

2平面の配置が可変

Light Field



M. Levoy, P. Hanrahan, "Light Field Rendering", 1996

ライトフィールドレンダリング, ルミグラフ

- 自由視点画像(ステレオ画像)の例

Lumigraph

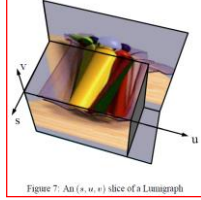
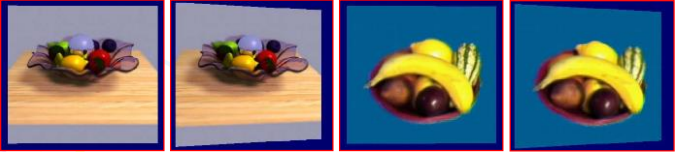


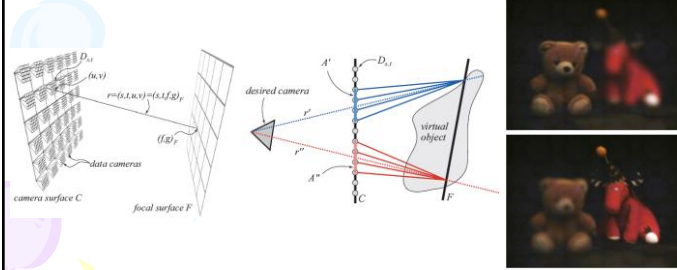
Figure 7: An (s, u, v) slice of a Lumigraph



S. Gortler, R. Grzeszczuk, R. Szeliski, M. Cohen, "The Lumigraph", 1996

ライトフィールドを利用した仮想カメラ

- A. Isaksen, L. McMillan, S. Gortler, "Dynamically Reparameterized Light Fields", SIGGRAPH 2000
- 複数の参照画像からライトフィールドを構築し, 自由視点位置, レンズ口径, フォーカスなどを自由に変更できる仮想カメラを実現する.



A. Isaksen, L. McMillan, S. Gortler, "Dynamically Reparameterized Light Fields", 2000

カメラアレイによる多様な映像の生成

- B. Wilburn, N. Joshi, V. Vaish, E. Talvala, E. Antunez, A. Barth, A. Adams, M. Horowitz, M. Levoy, "High Performance Imaging Using Large Camera Arrays", SIGGRAPH 2005

- 多数のカメラによりカメラアレイを構成し、ハイダイナミックレンジ、高解像、高速撮影を実現し、自由視点位置、レンズ口径、フォーカスなどを自由に変更することで、多様な映像を生成する。



B. Wilburn, N. Joshi, V. Vaish, E. Talvala, E. Antunez, A. Barth, A. Adams, M. Horowitz, M. Levoy, "High Performance Imaging Using Large Camera Arrays", 2005



その他

オパシティハル(Opacity Hull)

- ビジュアルハルの表面上に不透明度(α 値)と反射光の場を与えた**オパシティハル(opacity hull)**を構築し、半透明物体を任意の光環境で照らしたシーンを任意の視点から見た自由視点画像を生成する。

- W. Matusik, H. Pfister, A. Ngan, P. Beardsley, R. Ziegler, L. McMillan, "Image-Based 3D Photography using Opacity Hulls", SIGGRAPH 2002.

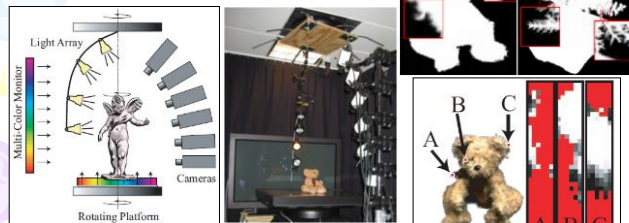
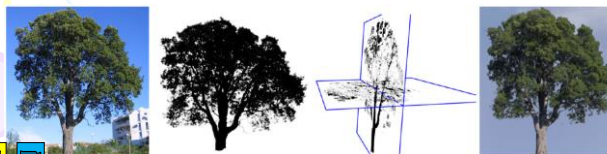


Figure 4. Observed alpha values for points on the opacity hull. Red color indicates another camera view.



半透明ボクセルモデル

- ボクセルモデルに不透明度(α 値)を与えて半透明物体を表現。
- 対象物体を周囲の複数の場所から撮影した参照画像を用い、それらに α マッピングを適用して α マツト画像を生成。生成された複数の(2次元) α マツト画像を利用して(3次元)ボクセルの α 値を推定することで半透明ボクセルモデルを生成。
- 樹木を対象とし、細かな葉や枝が背景とともに1つの画素やボクセルの内部に含まれる微細な形状を α 値を用いた半透明ボクセルモデルで表現。
- A. Reche, I. Martin, G. Drettakis, "Volumetric Reconstruction and Interactive Rendering of Trees from Photographs", SIGGRAPH 2004



複数のカメラとプロジェクタによる3DTV

- 複数のカメラとプロジェクタにより、カメラ視点の異なる複数のビデオ映像を撮影、転送、再生することで、立体視を可能とした3Dテレビのシステム。
- **W. Matusik, H. Pfister, "3D TV: A Scalable System for Real-Time Acquisition, Transmission, and Autostereoscopic Display of Dynamic Scenes", SIGGRAPH 2004**



Figure 1: 3D TV system. Left (top to bottom): Array of 16 cameras and projectors. Middle: Rear-projection 3D display with double-lenticular screen. Right: Front-projection 3D display with single-lenticular screen.

W. Matusik, H. Pfister, "3D TV: A Scalable System for Real-Time Acquisition, Transmission, and Autostereoscopic Display of Dynamic Scenes", 2004

