



# 遠隔地の視覚的環境を観測者の 周囲に再現する実験システムの 開発と UZUME 計画への応用

---

床井浩平 (和歌山大学)

大山英明 (産総研)

河野功 (JAXA)



# はじめに

## この研究の立場

# 研究目的

- 遠隔地にあるものを目の前にあるように見せる
- 目の前に見えるものを操作して遠隔地に関与する

視覚の伝送



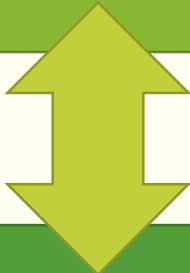
遠隔操作

# 遠隔操作型代理科学者 (ERSA) の実現

- テレイグジスタンス (telexistence) 技術を利用する
  - バーチャルリアリティの一分野
  - 遠隔地にある人や物を身近にあるように感じさせる
  - その環境下で遠隔操作などをリアルタイムに行う
- テレロボティクス (telerobotics) 技術と組み合わせる
  - テレイグジスタンスロボット
  - リアルに再現された遠隔地の環境下で操縦する
  - 現地で作業しているような感覚を与える

# テレイグジスタンスロボットの二つの方向

操縦者の周囲に現地の環境を再現する



対象が操縦者の手元にあるように見せる

地球上

# 本研究の目標

自分がそこに行く



相手を連れてくる

# 相手が目の前にいるように感じさせる



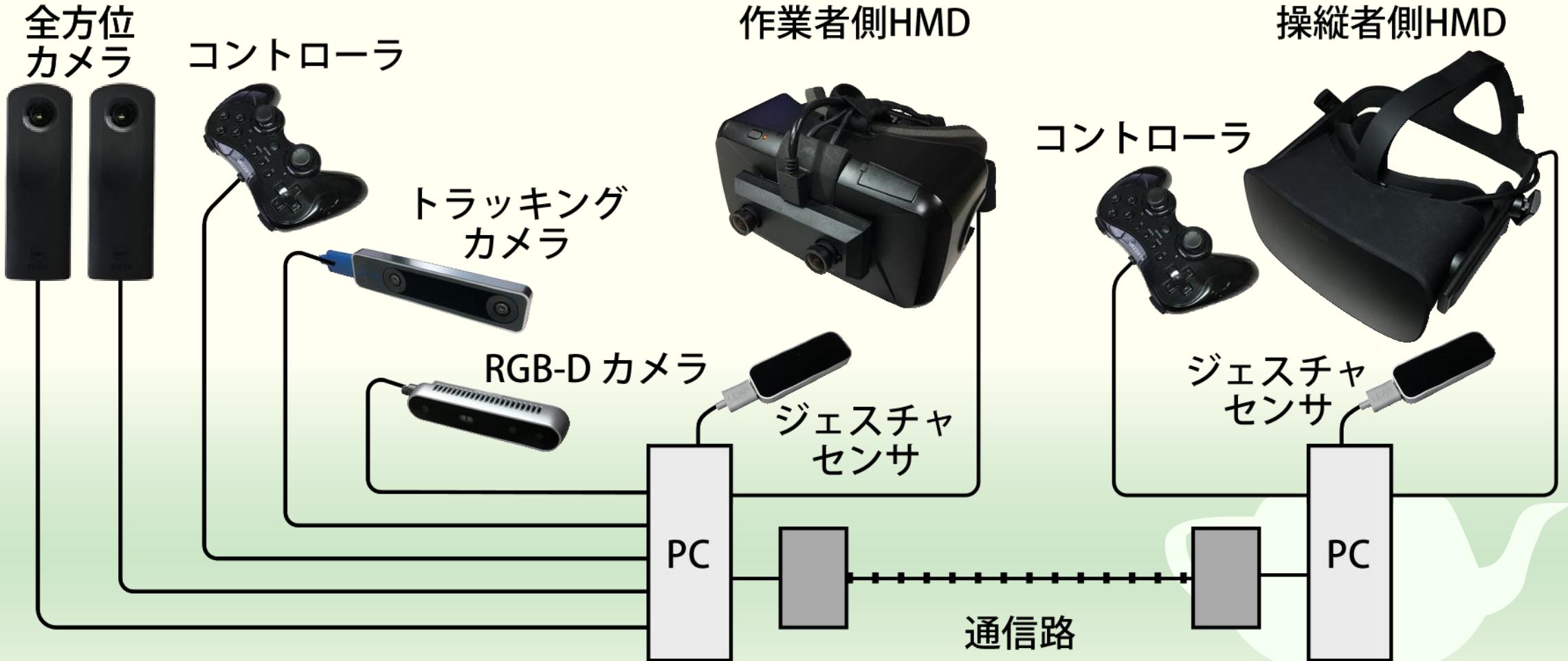
栗田滋弘, “没入型ディスプレイを用いた臨場感通信に関する研究,” 2001年度卒業研究



# 遠隔地の視覚的環境を観測者の 周囲に再現する実験システム

TelExistence Display System (TED)

# システム概要



# システム構成例



# システムの機能

- 映像の入出力に多様な設定が可能
  - 入力側
    - 単眼カメラ, ステレオカメラ, 魚眼カメラ, ステレオ魚眼カメラ, 全方位カメラ, 全方位ステレオカメラ, RGB-D センサ
  - 表示側
    - 単眼ディスプレイ, 立体視ディスプレイ (サイドバイサイド, トップアンドボトム), HMD (Oculus Rift DK2 / CV1 / S)
- 3D モデルの重畠表示
  - Alias OBJ ファイルの読み込みと JSON によるシーン記述
  - ジェスチャセンサ (Leap Motion) による制御

# シーンの記述

- JSON によるグラフ形式
- Alias OBJ 形式の形状データの読み込み
- 共有メモリを介して外部プログラムで取得した配置情報を参照可能
- 外部インターフェース
- Leap Motion
- トランкиングカメラ

```
{  
  "position": [ x, y, z ],  
  "rotation": [ x, y, z, angle ],  
  "scale": [ x, y, z ],  
  "model": "形状データファイル名",  
  "controller": 变換行列番号,  
  "children": [  
    シーングラフのノード,  
    :  
    シーングラフのノード  
  ]  
}
```

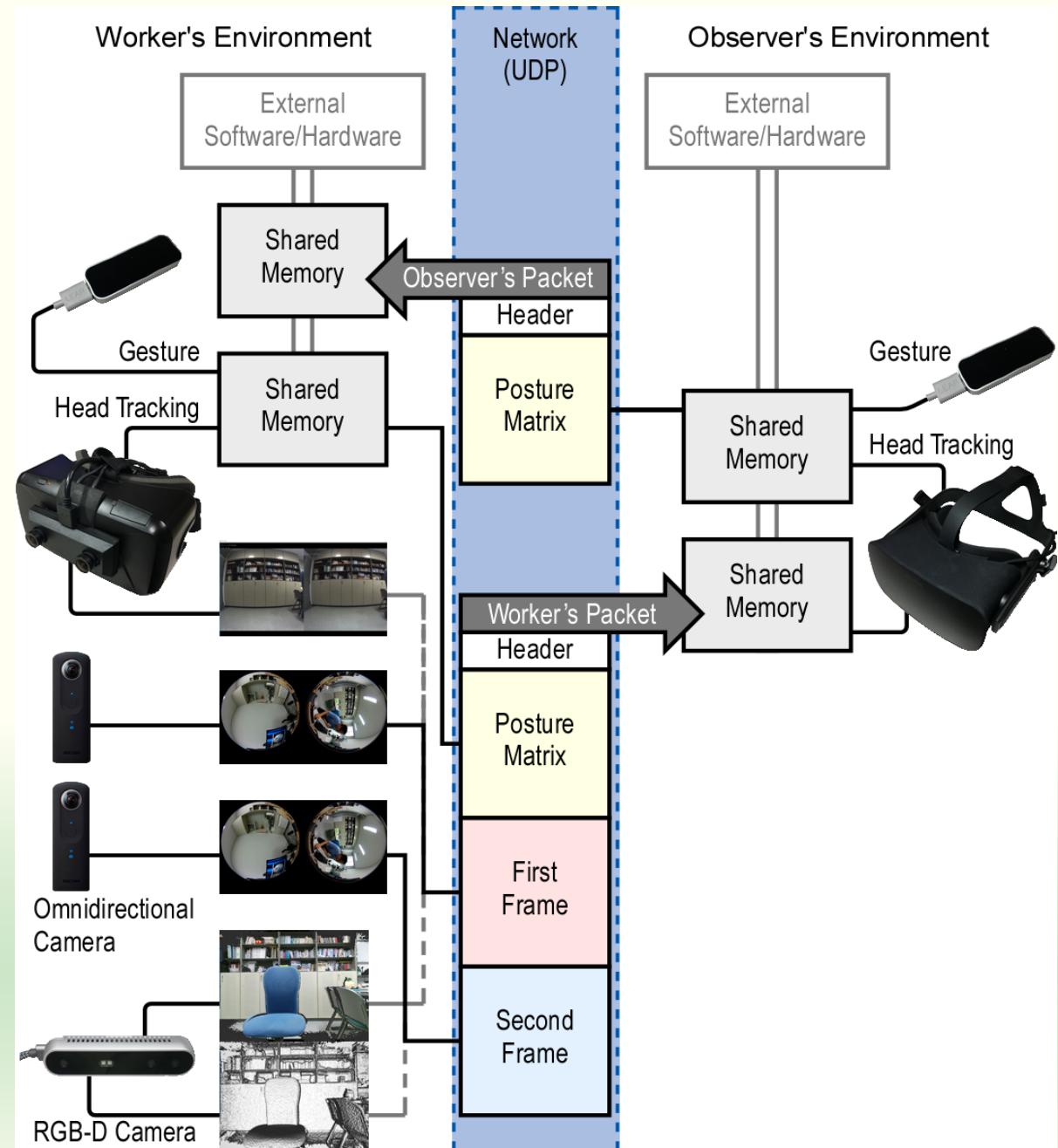
# 仮想物体の重畠



# ネットワーク機能

- 作業者側の環境映像を操縦者に**伝送**
- 作業者と操縦者のヘッドトラッキング**情報を交換**
- 作業者と操縦者のジェスチャ (手の姿勢) を**交換**
  - 互いの姿勢を互いの視野に重畠表示
- 作業者側の周囲の**距離情報を操縦者側に伝送** (未了)
  - RGB-D カメラ及び全方位ステレオカメラによる取得
  - 現在システムに統合中
- UDP によるリアルタイム/ステートレスなデータ転送

# 通信手順



# 視界の共有



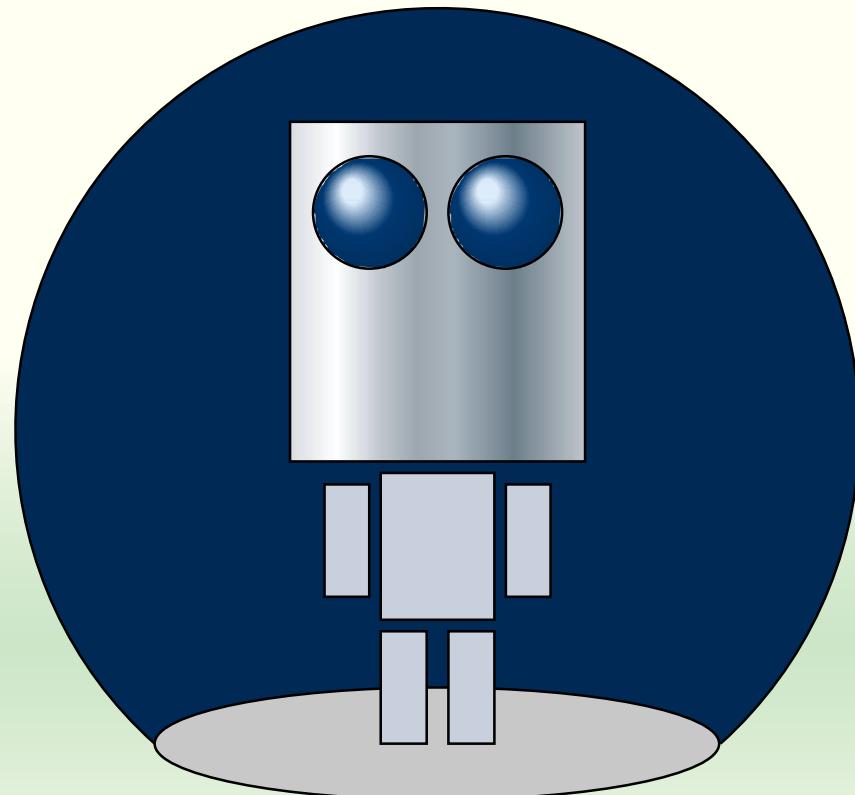


# UZUME 計画への応用

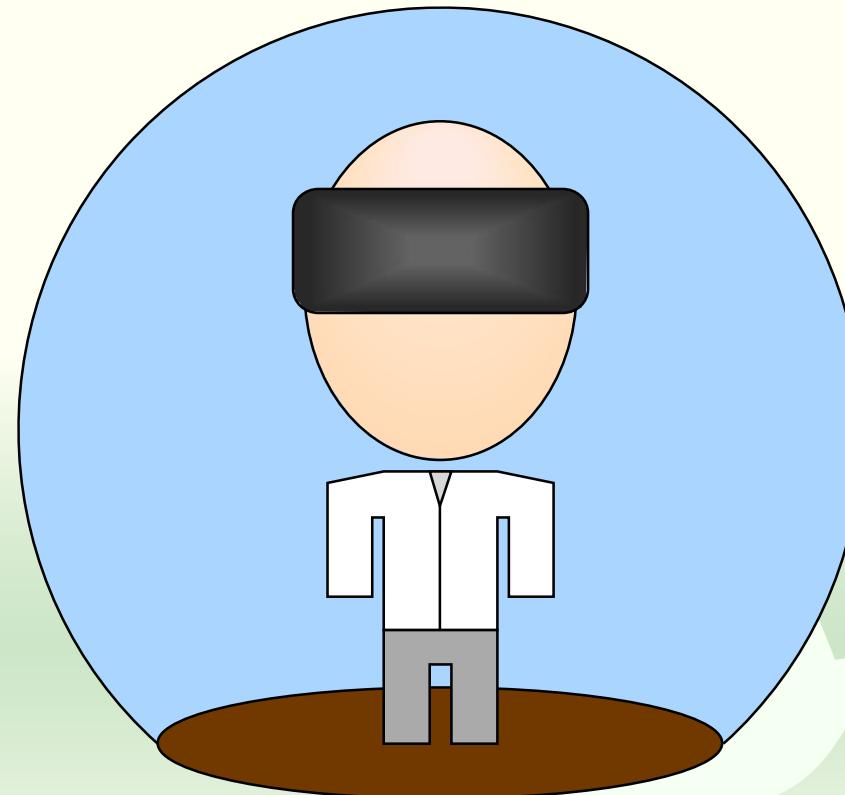
環境が大きく異なる遠隔地の場合

# 作業者と操縦者の環境は異なる

作業者（遠隔操作型代理科学者）



■ 操縦者（地上の科学者）



# 遠隔環境の再現において考慮すべき点

- そもそも遠隔環境は人間に知覚が容易(可能)なのか
  - 大半の科学者は月面や宇宙環境における経験を持たない
- 明度環境の不一致
  - コントラスト・ダイナミックレンジが異なる
- 色彩環境の不一致
  - マルチスペクトル画像の取り扱い
- スケールも異なる

月面の映像はコントラストが非常に高い

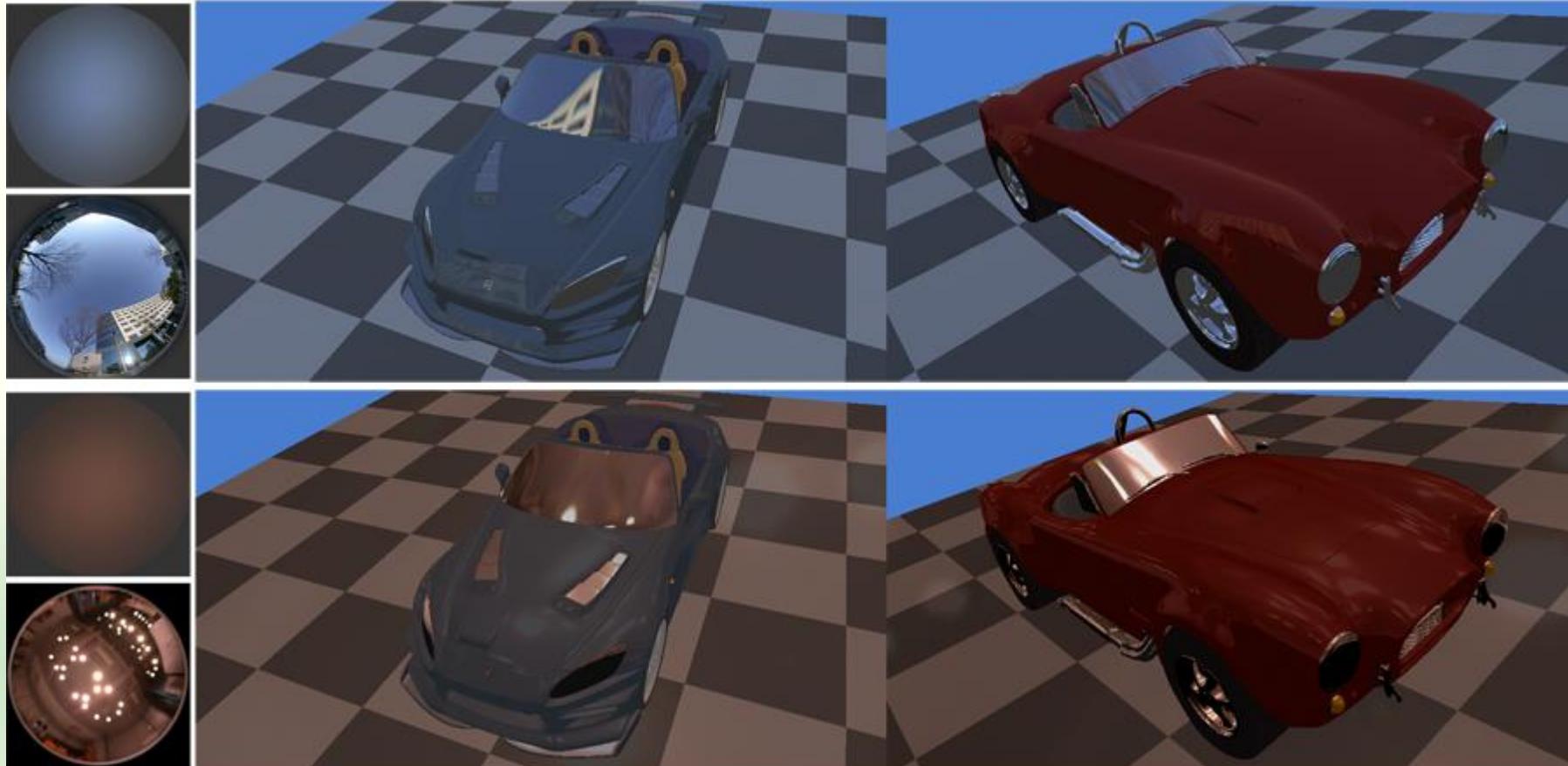


# 科学者に適切な情報提示を行う

- 再現する遠隔環境を**地上の環境に合わせる**
  - 地上の環境との合成
  - 放射照度マッピング
  - リライティング(再照明)
- 提示すべき情報を**抽出・強調・加工**する
  - 疑似カラー・イラストレーション化
  - トーンマッピング
- 科学者との議論が必要



# 放射照度マッピング



# 制限事項

- 伝送帯域が狭い
  - 1Mbps も確保できない?
    - 電波が弱い(遠い, 電源容量が制限されている, 等)
    - これは将来技術的に解消できる可能性がある
- 伝送遅延が大きい
  - これを解消する方法は現時点では現実には存在しない
    - 光(電波)でも月まで約1.3秒, 火星まで約4分28秒
    - TCP の3ウェイハンドシェークのようなものは使い物にならない
- 対話的な操作が困難

# 視点の追従性の改善

- 操縦者の視点は随意的に移動する
  - 体が移動しなければ移動量は相対的に小さい
- ロボットの視点は固定されている
  - カメラはロボットヘッドに固定されている
  - 映像に伝送遅延が存在する



ロボットの視点を固定したまま  
操縦者が任意の視点から見られるようにする

# 広角(魚眼)画像を用いた視野の安定化

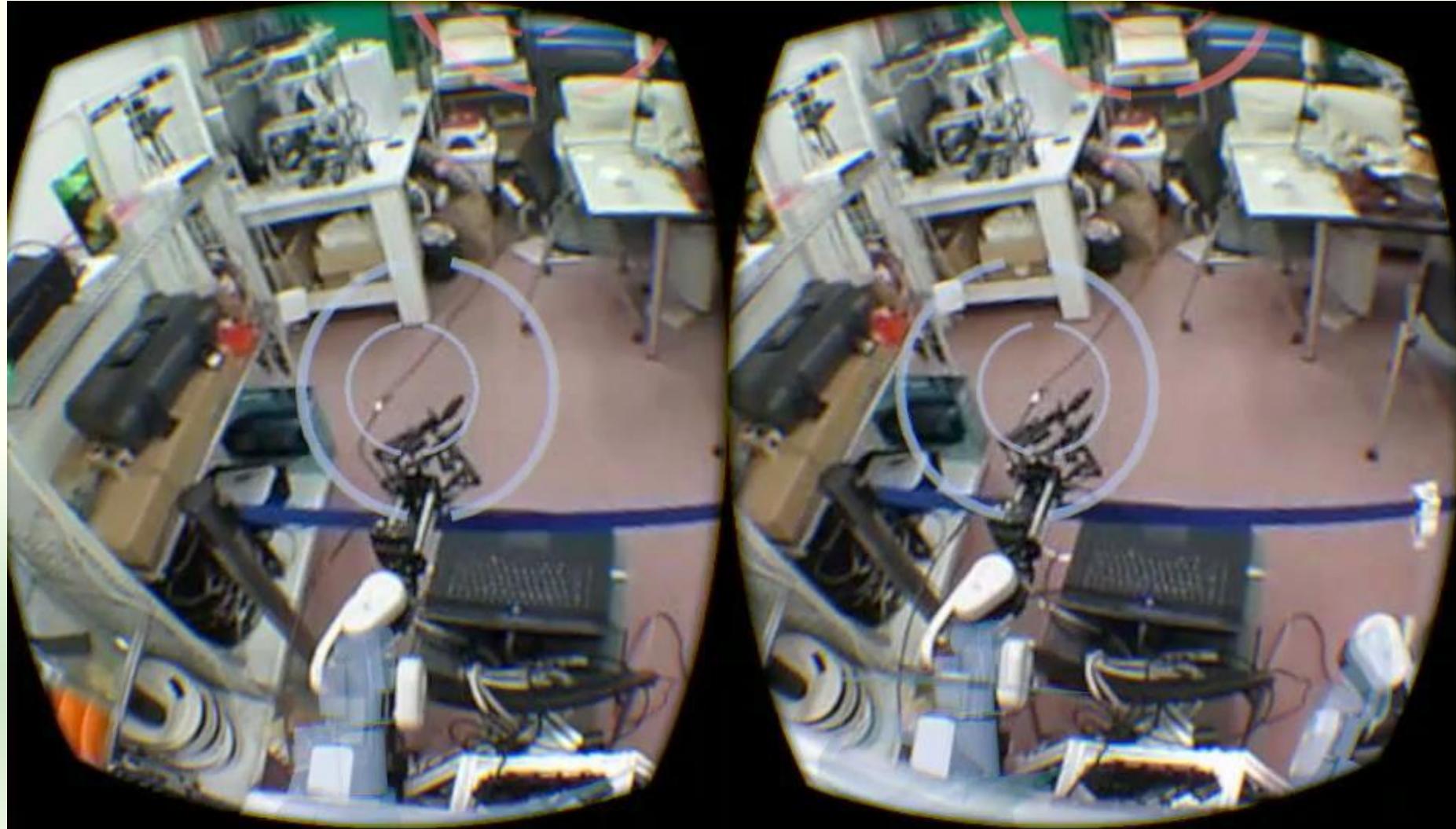


大山 他, “TelExistence Display System (TED): テレイグジスタンス／遠隔行動誘導／体験共有のための AR 表示オープンソースソフトウェア,” 計測自動制御学会システムインテグレーション部門講演会, Vol. 18, P. 2C2-06, 2017

# 遅れのあるテレイグジスタンスロボット



# 予測CGと遅れて動作する実ロボット



# 遠景と近景に分ける

## ■ 遠景

- ロボットの作業空間における自己位置の把握
  - 環境に対する移動量は相対的に小さい
  - 深度のみを利用し occlusion は反映しない

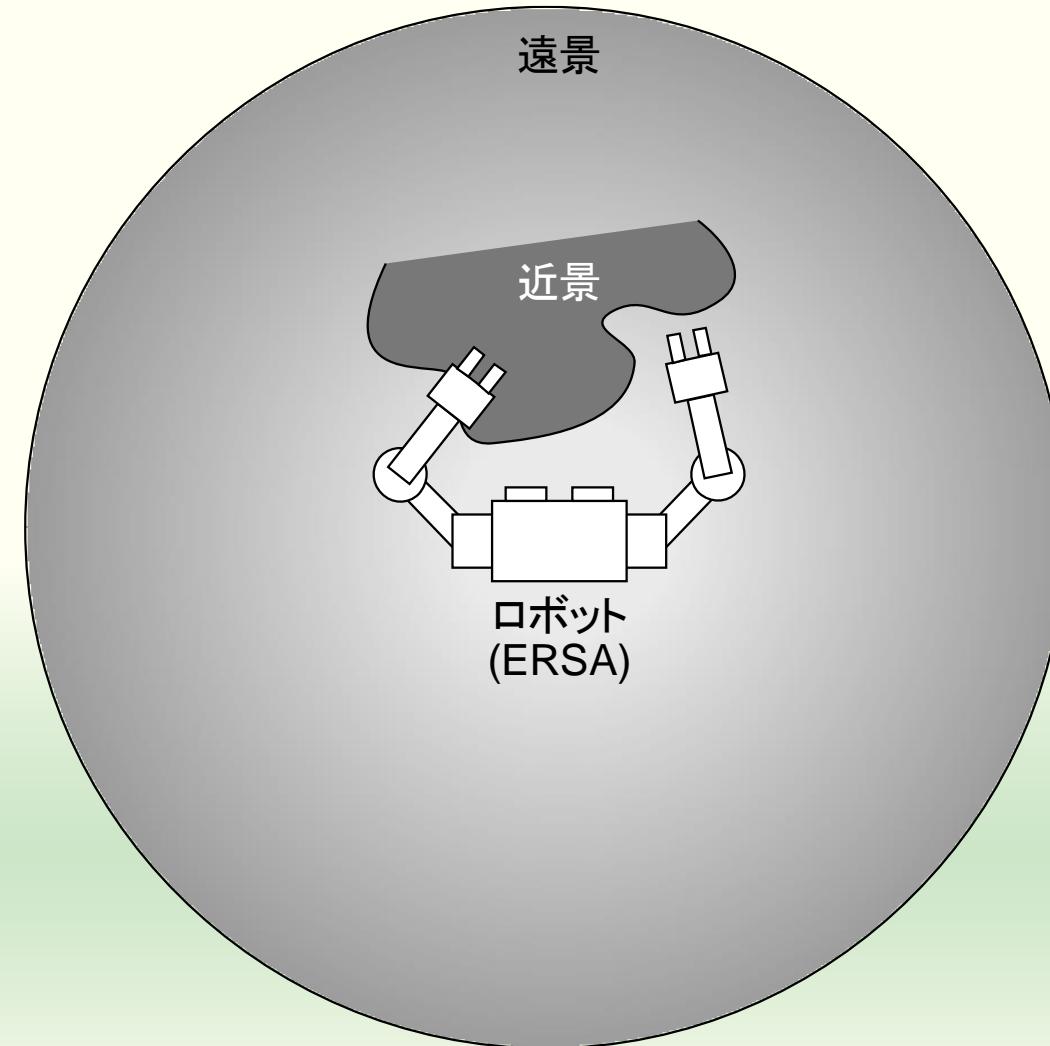
## ■ 近景

- 探査対象の詳細形状の把握
  - 覗き込むようなことがある
  - 3次元形状を計測する

## ■ 遠景に近景を重畠して表示する



# 遠隔操作型代理科学者の作業空間



# 遠隔環境の再現手法

## ■ 遠景の全方位(全天球)画像

- あらかじめ遠隔地の環境映像を操縦者側に蓄積しておく
- 遠隔地の環境は短時間に大きく変化しないものと仮定

## ■ 遠景と近景の距離画像の利用

- 遠景は全方位ステレオ計測により取得する
- 近景は RGB-D センサ(ステレオまたは TOF)により取得する



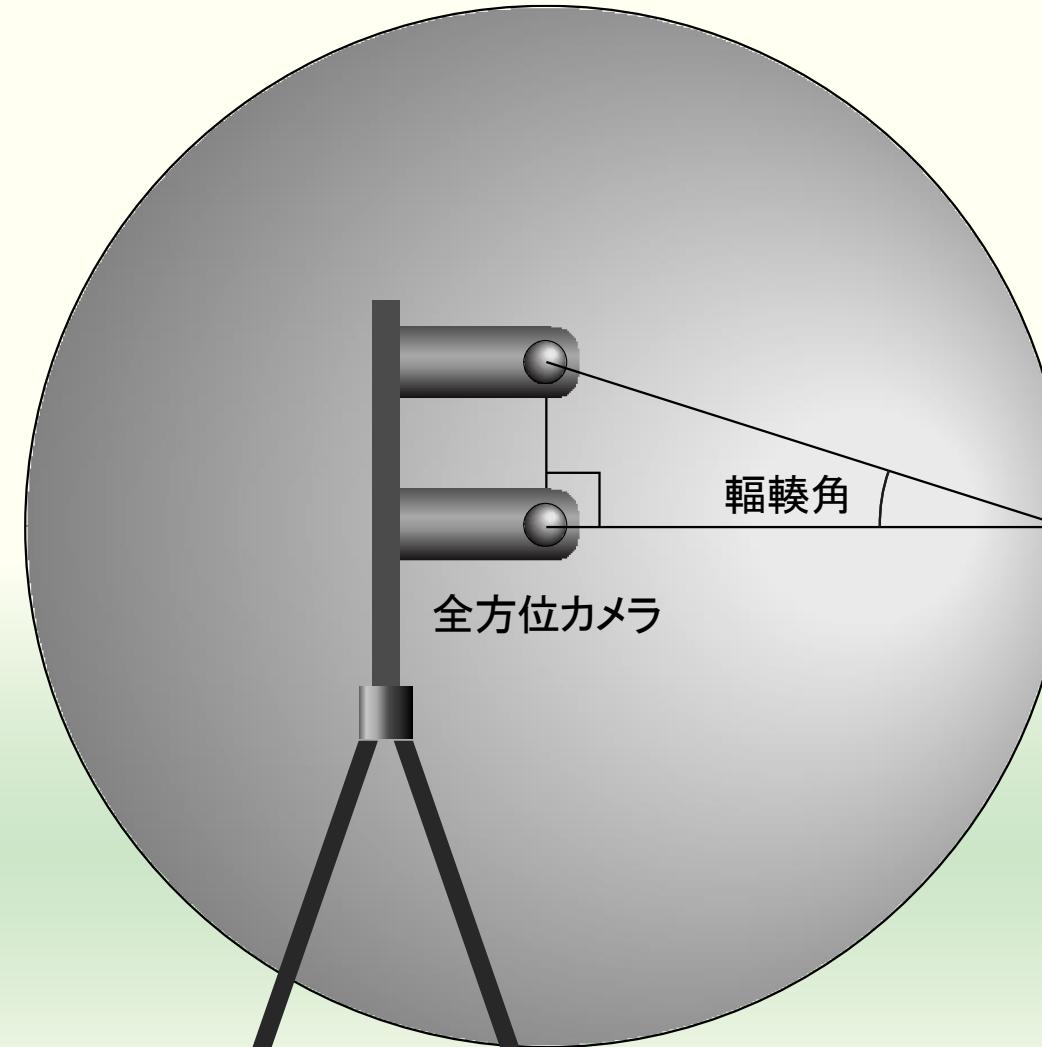
# 宇宙空間で使用可能な小型全天球カメラ

- JAXAとRICOHの共同開発
- 宇宙ステーション補給機  
「こうのとり」8号機で  
ISSに運搬
- 「きぼう」日本実験棟の  
船外実験プラットフォー  
ムから撮影予定
- 2019年8月28日発表
- RICOH THETA Sベース
- THETA Sは本研究で使用

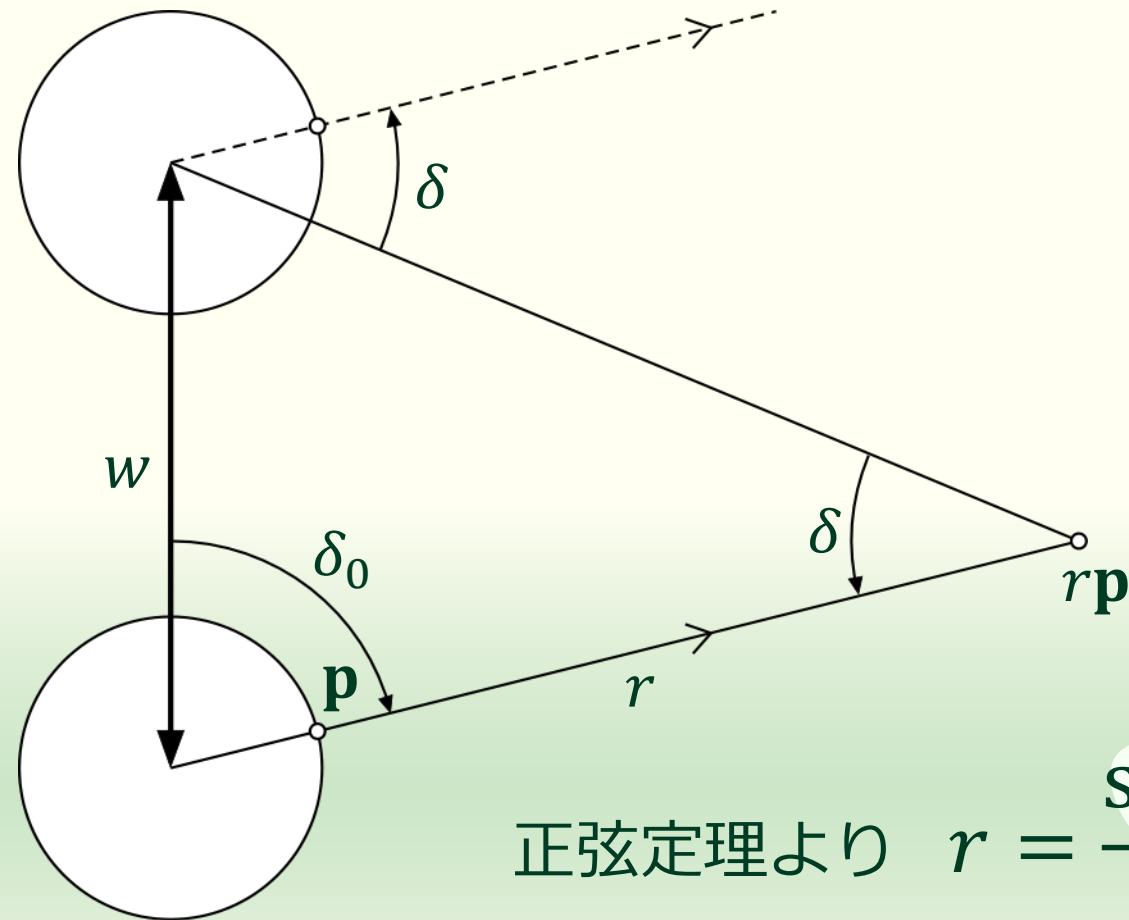


©JAXA

# 遠景の距離の取得

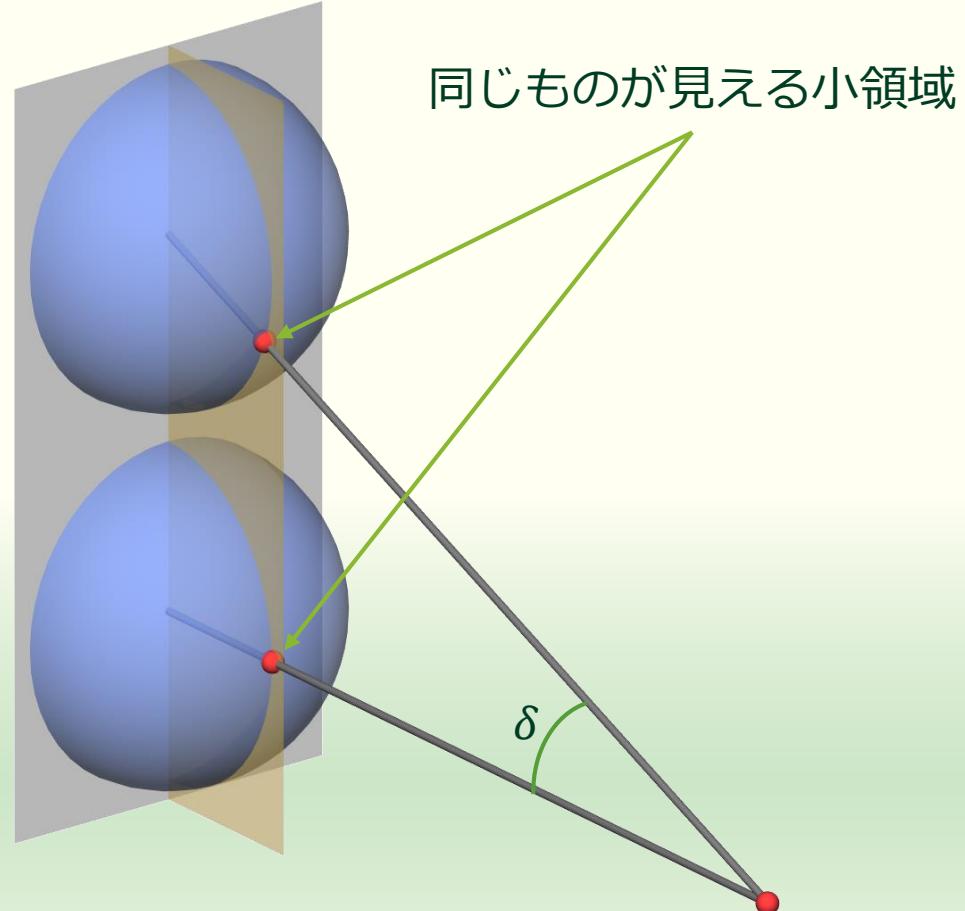


# 等距離射影魚眼レンズによる輻輳角と位置

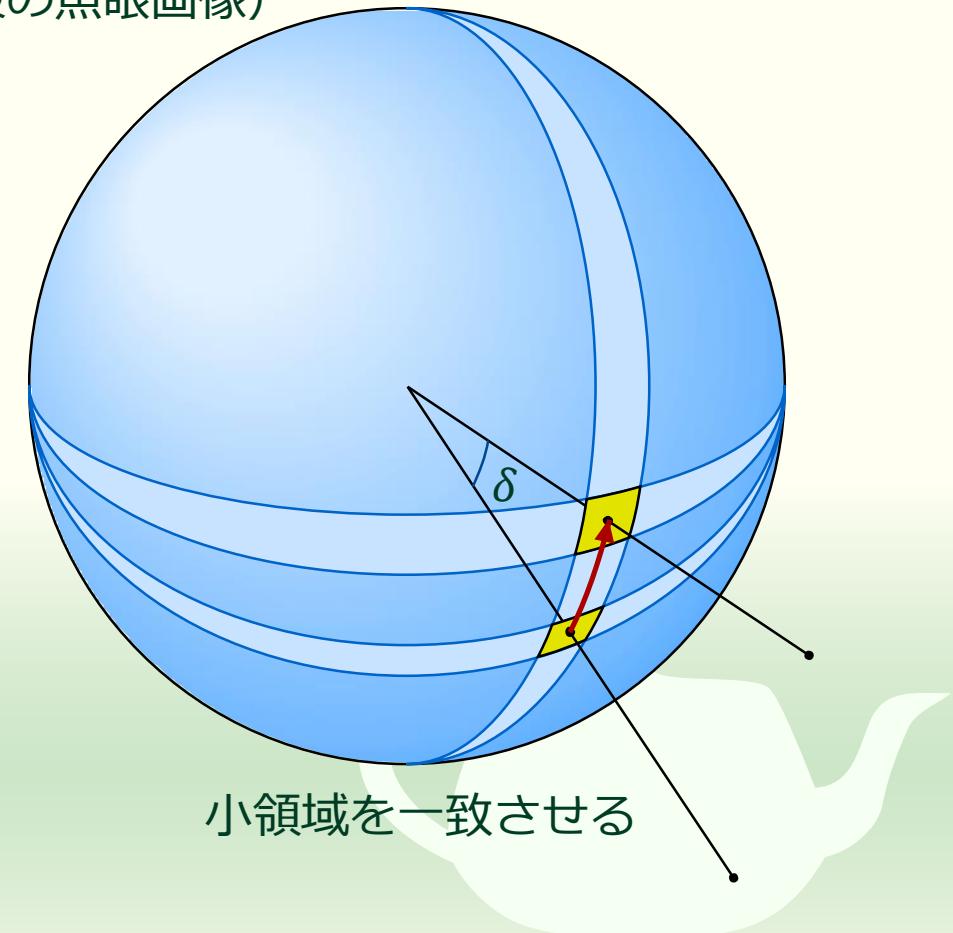


正弦定理より  $r = \frac{\sin(\delta + \delta_0)}{\sin \delta} w$

# ステレオ魚眼画像による輻輳角の取得



(上段の魚眼画像)

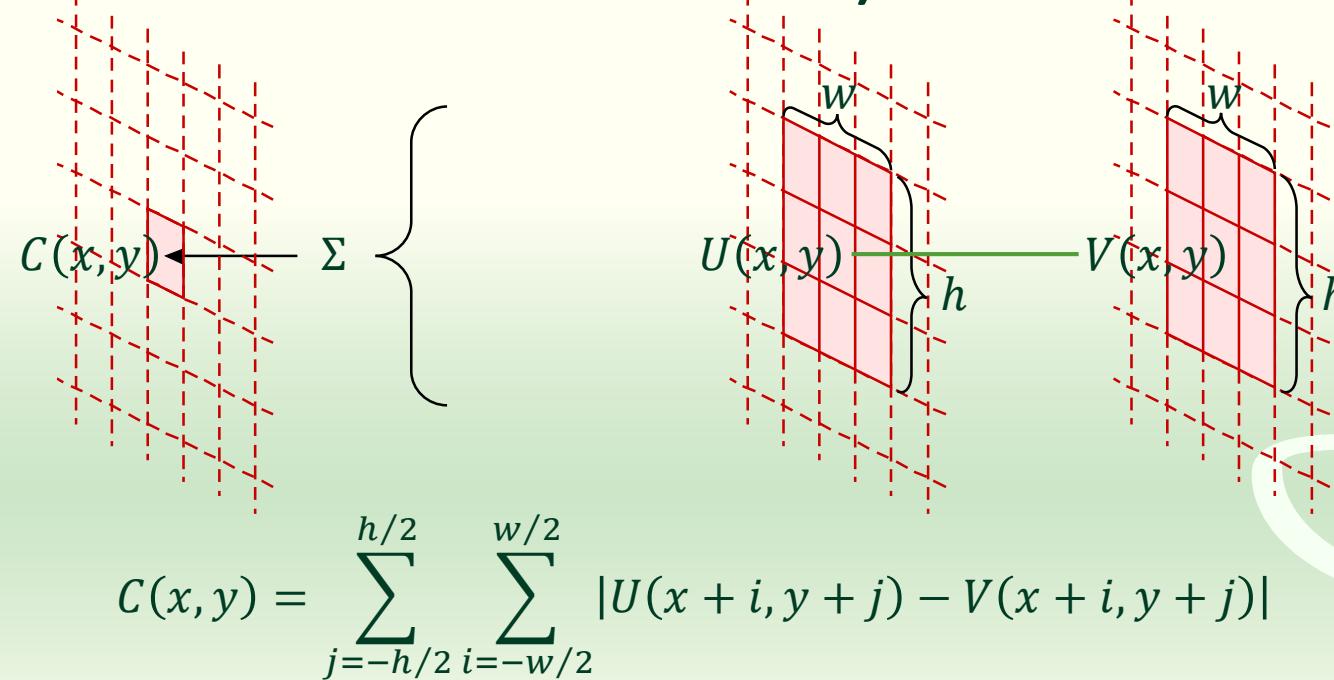


# 魚眼画像の変形

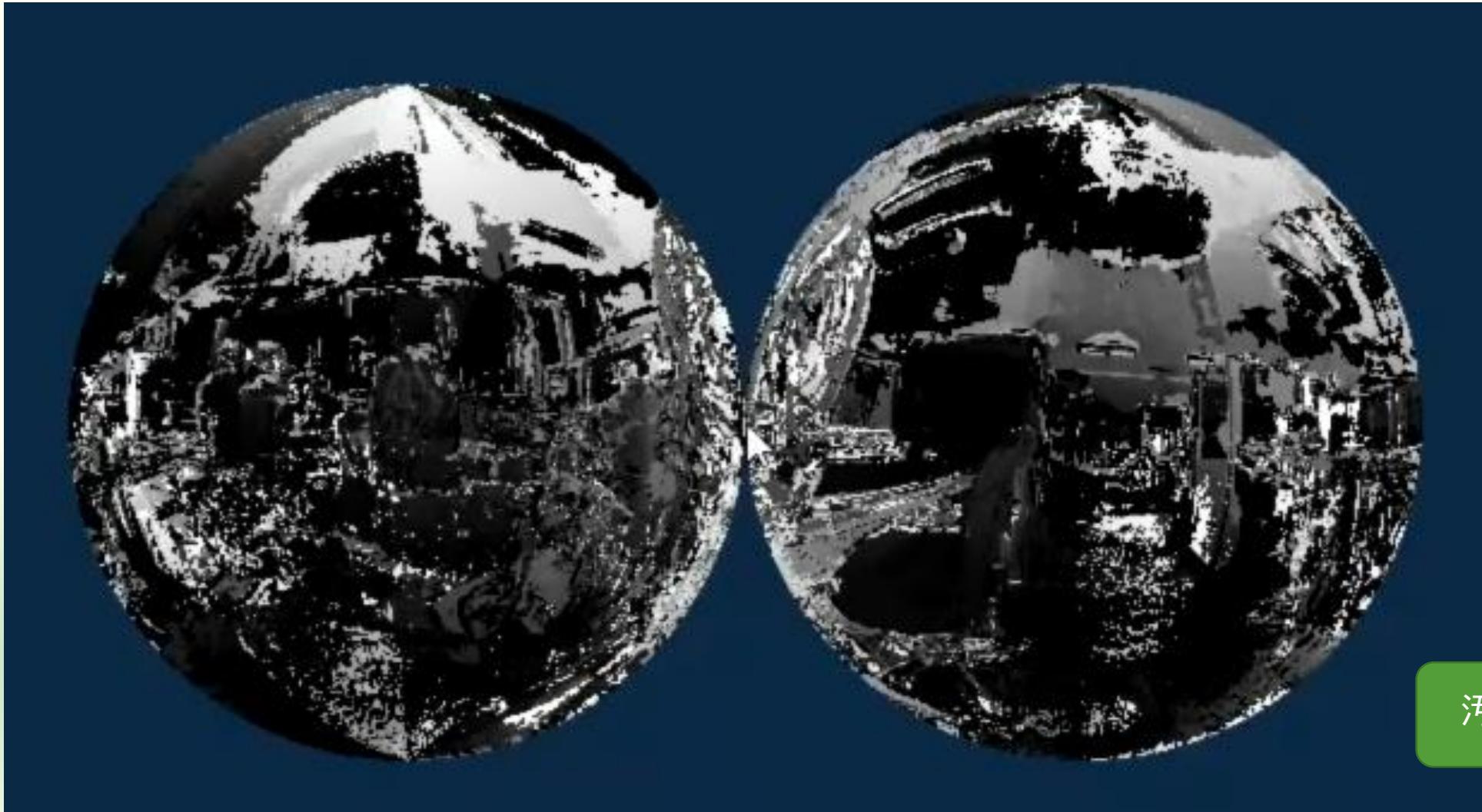


# マッチングコスト

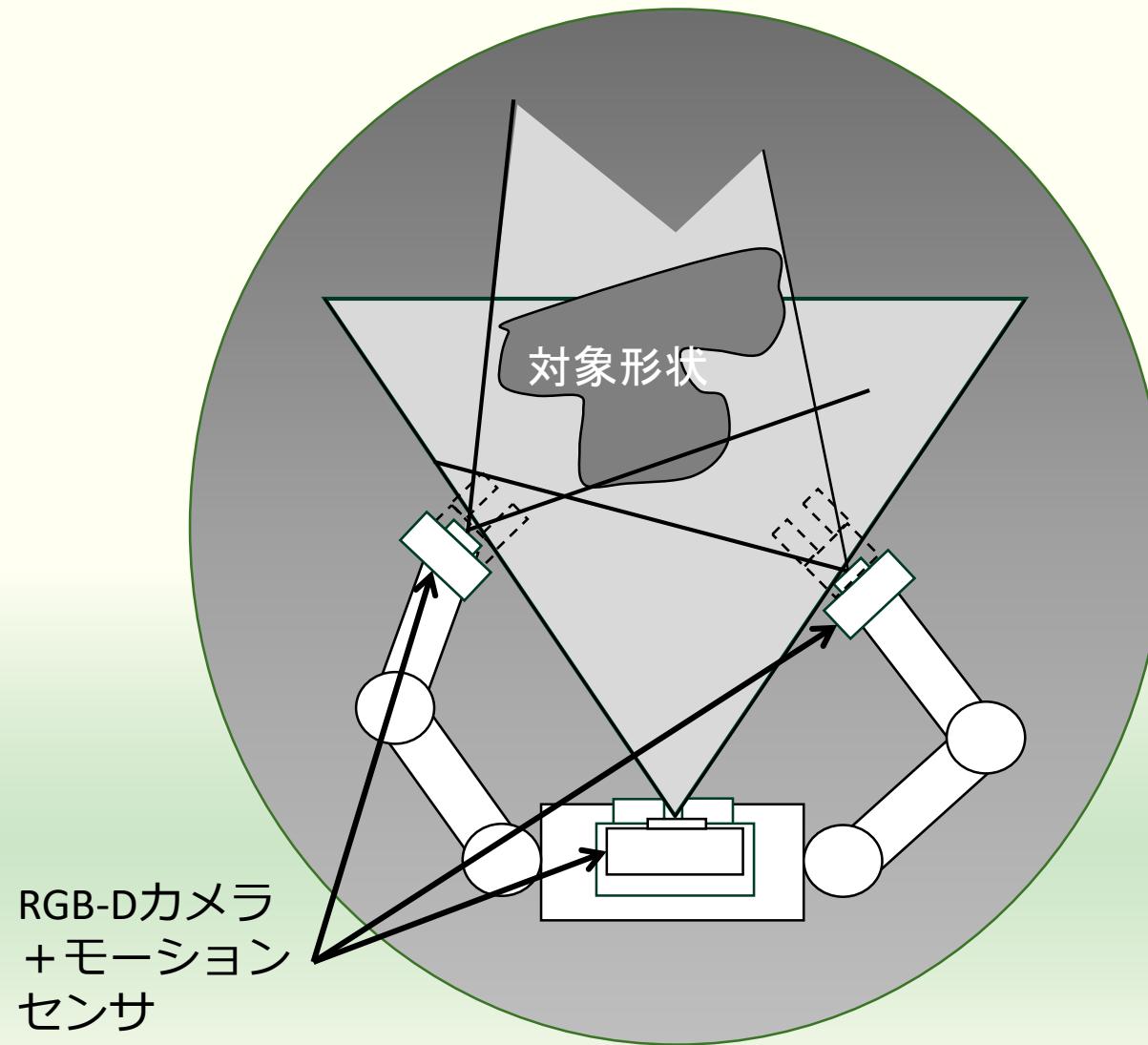
- 二つの領域が似ているほど低い
- 全く同じなら 0
- SAD (Sum of Absolute Difference) の場合


$$C(x, y) = \sum_{j=-h/2}^{h/2} \sum_{i=-w/2}^{w/2} |U(x + i, y + j) - V(x + i, y + j)|$$

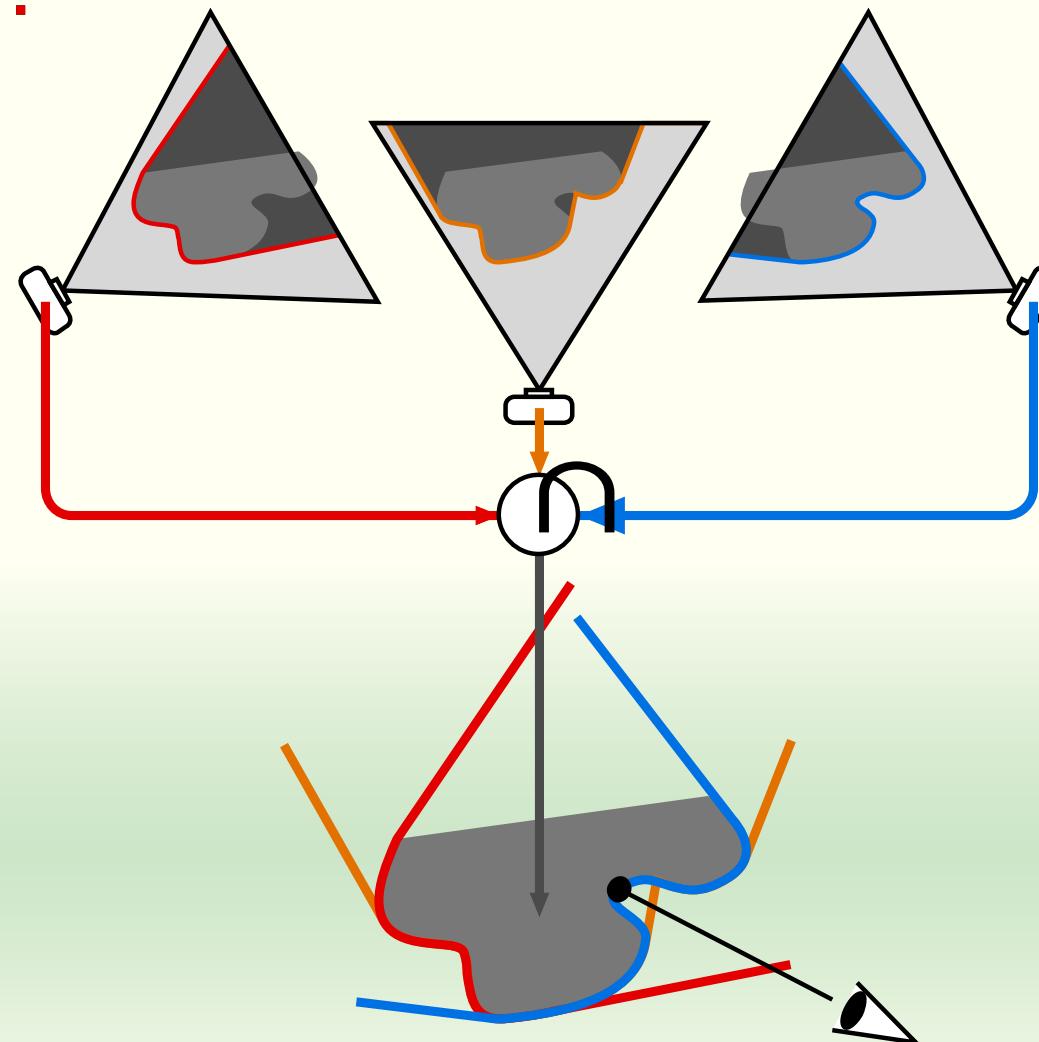
# ブロックマッチングによる視差画像



# 近景の取得



# デプスマップ融合



# デプスマップ融合による形状再構成

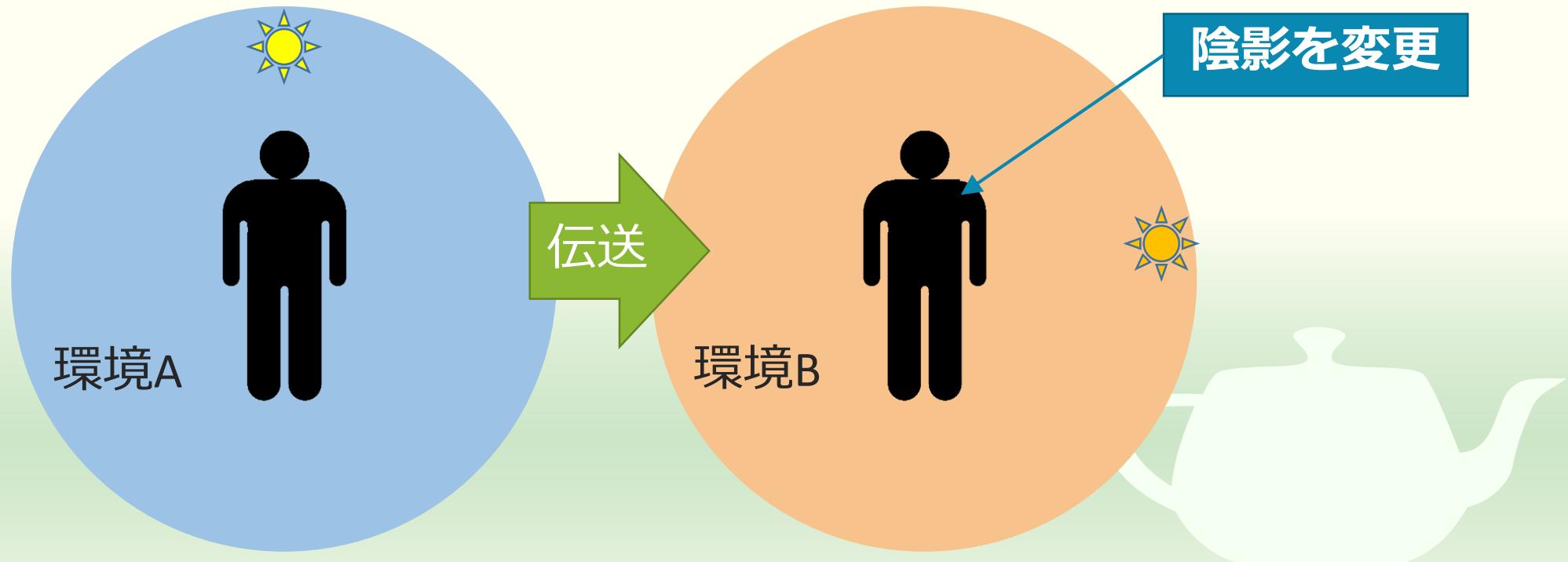


汚い

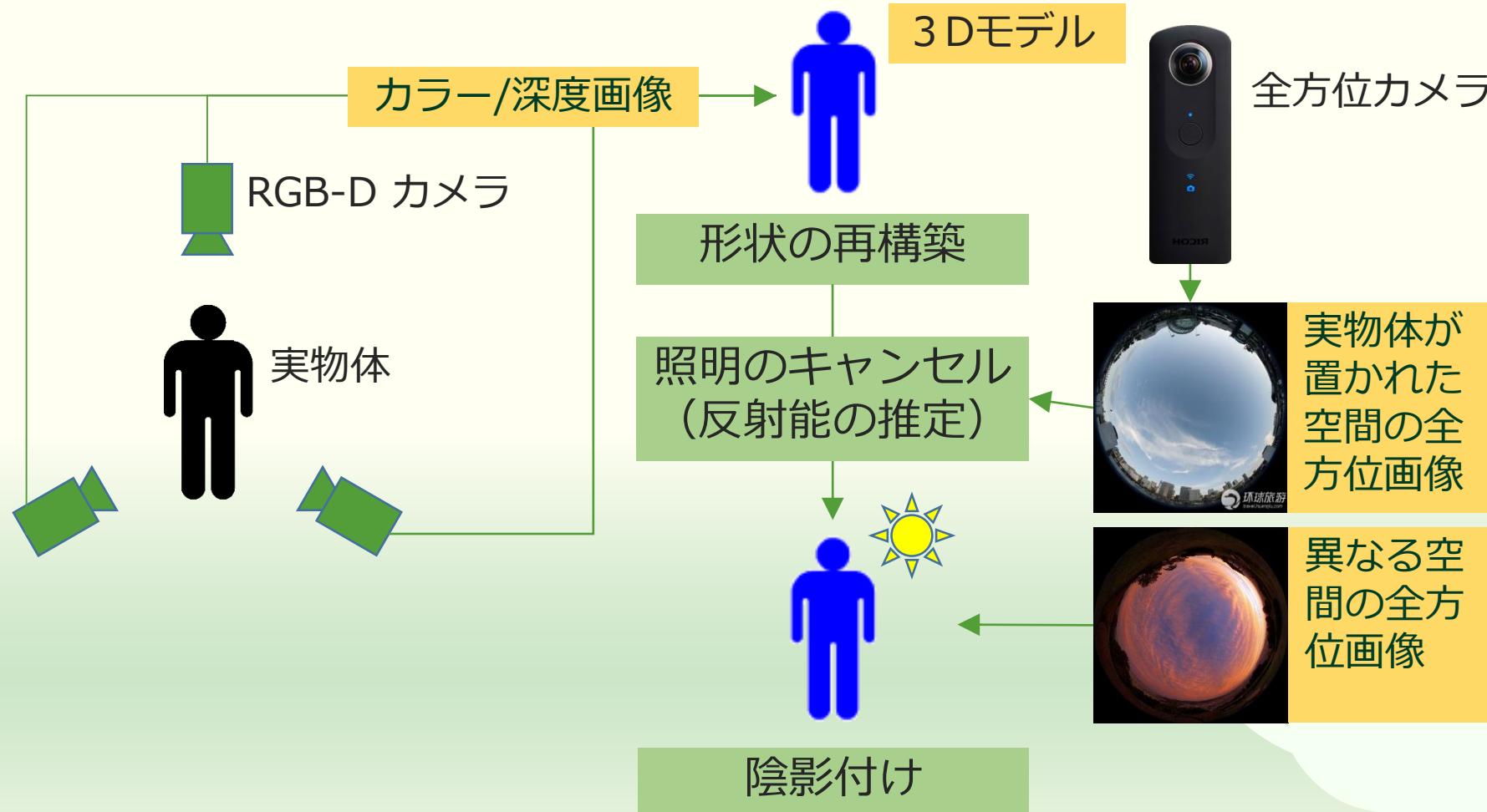
森正樹, 床井浩平. "深度センサを用いて再構成した全周形状の立体視 HMD による観測 (現実世界と融合する立体映像および立体映像技術一般)." 映像情報メディア学会技術報告 38.24. 一般社団法人 映像情報メディア学会, 2014.

# 光源環境の再現

- 光源環境に応じて実物体の陰影を変更する
  - 青空の下に居ながら、夕日を浴びているような陰影



# 再照明



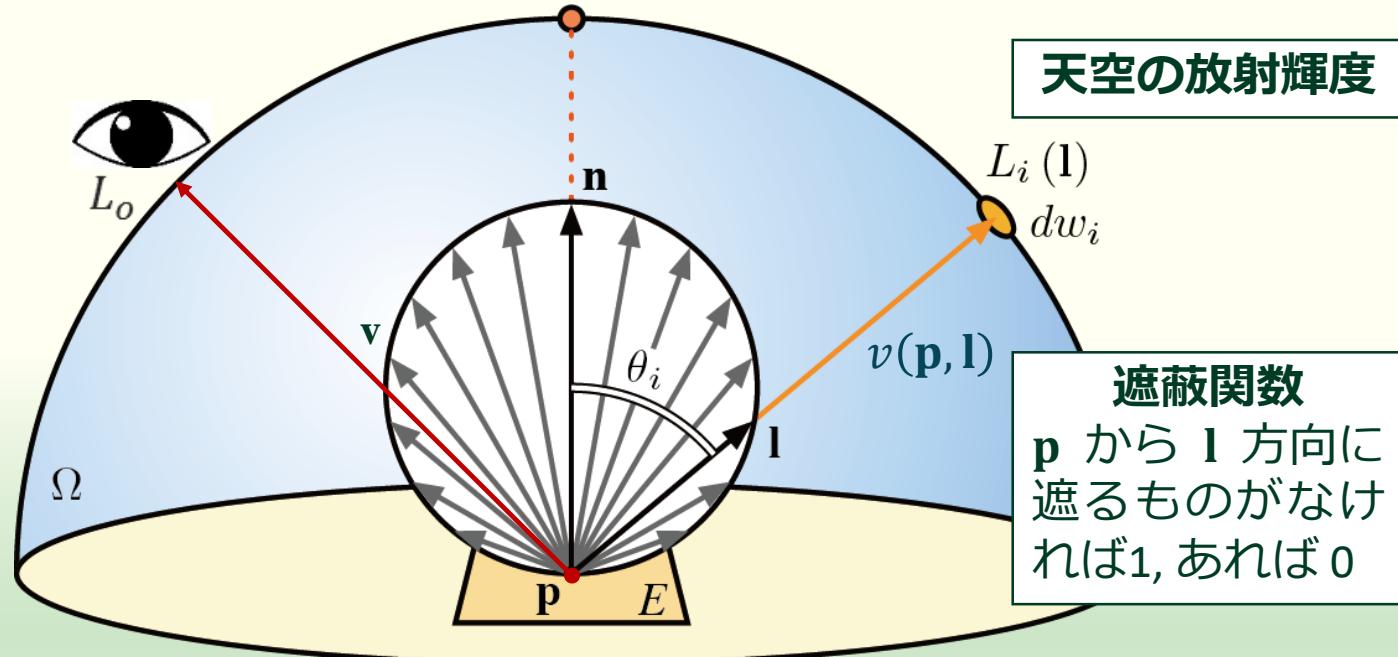
# 反射能の推定

$\mathbf{p}$  から  $\mathbf{v}$  方向  
への放射輝度

$$L_o(\mathbf{p}, \mathbf{v}) = \int_{\Omega} f(\mathbf{l}, \mathbf{v}) \nu(\mathbf{p}, \mathbf{l}) L_i(\mathbf{l}) \cos \theta_i d\omega_i$$

完全拡散反射面と仮定

$$L_o(\mathbf{v}) \cong \rho E$$



$\mathbf{p}$  の放射照度

$$E(\mathbf{p}) = \frac{1}{\pi} \int_{\Omega} L_i(\mathbf{l}) \nu(\mathbf{p}, \mathbf{l}) \cos \theta_i d\omega_i$$

モンテカルロ法で近似

$$E \cong \frac{1}{N} \sum_{i=1}^N L_i(\mathbf{l}(\theta_i))$$

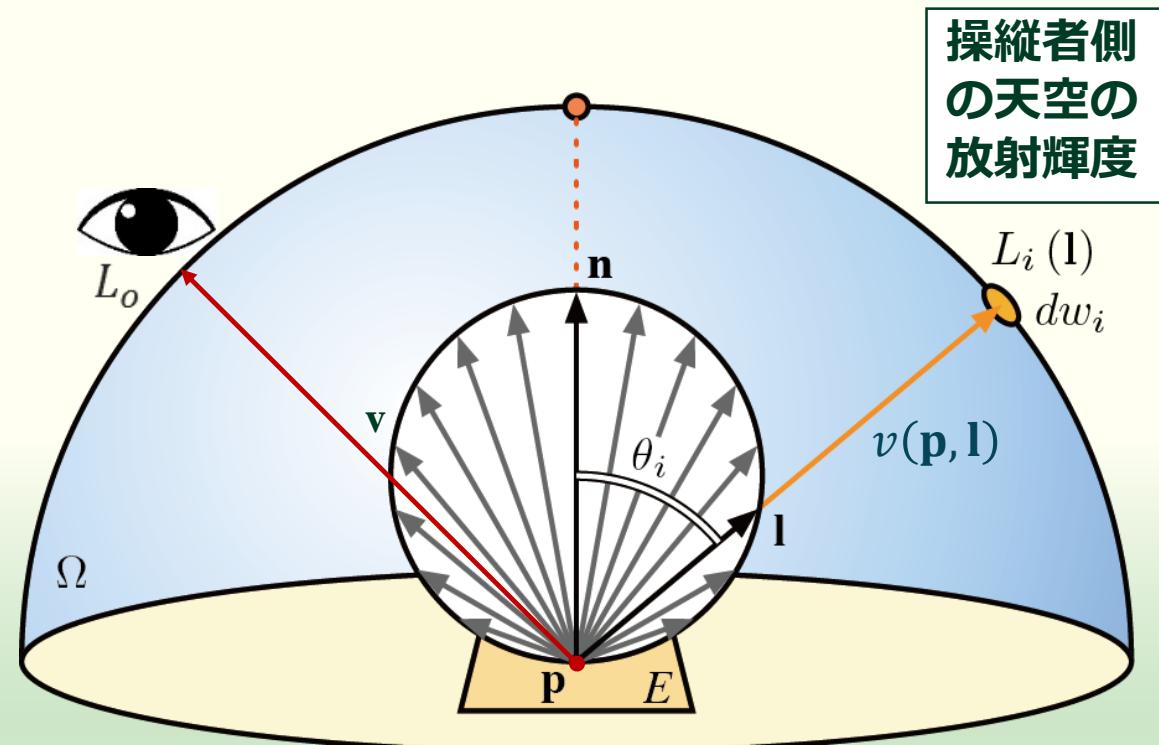
RGB-Dカメラの  
カラー画像

反射能

$$\rho \cong \frac{L_o(\mathbf{v})}{E}$$

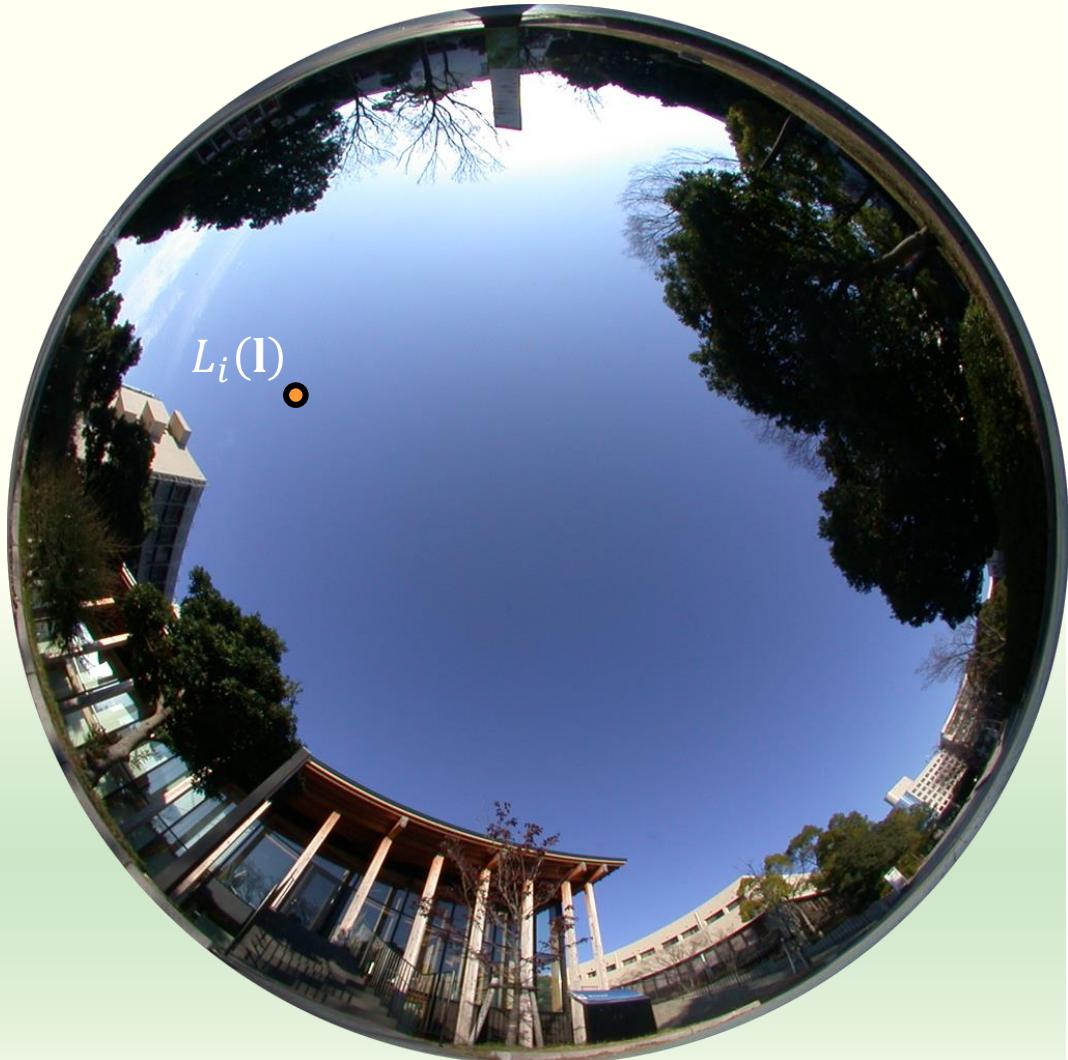
全方位カメラの  
天空画像

# 環境マップを使った操縦者側の照明



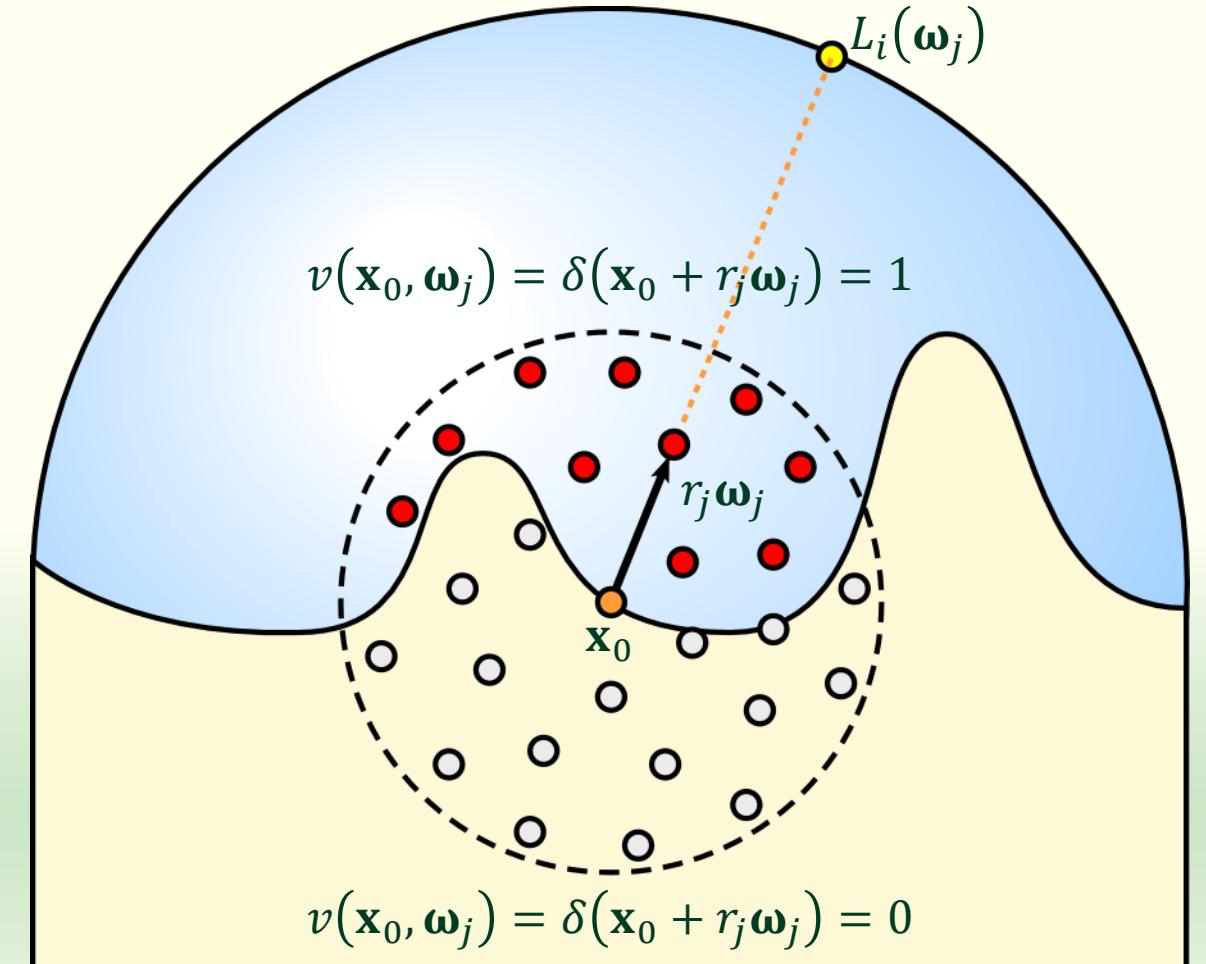
**p の放射照度**

$$E(p) = \frac{1}{\pi} \int_{\Omega} L_i(l) \nu(p, l) \cos \theta_i d\omega_i$$

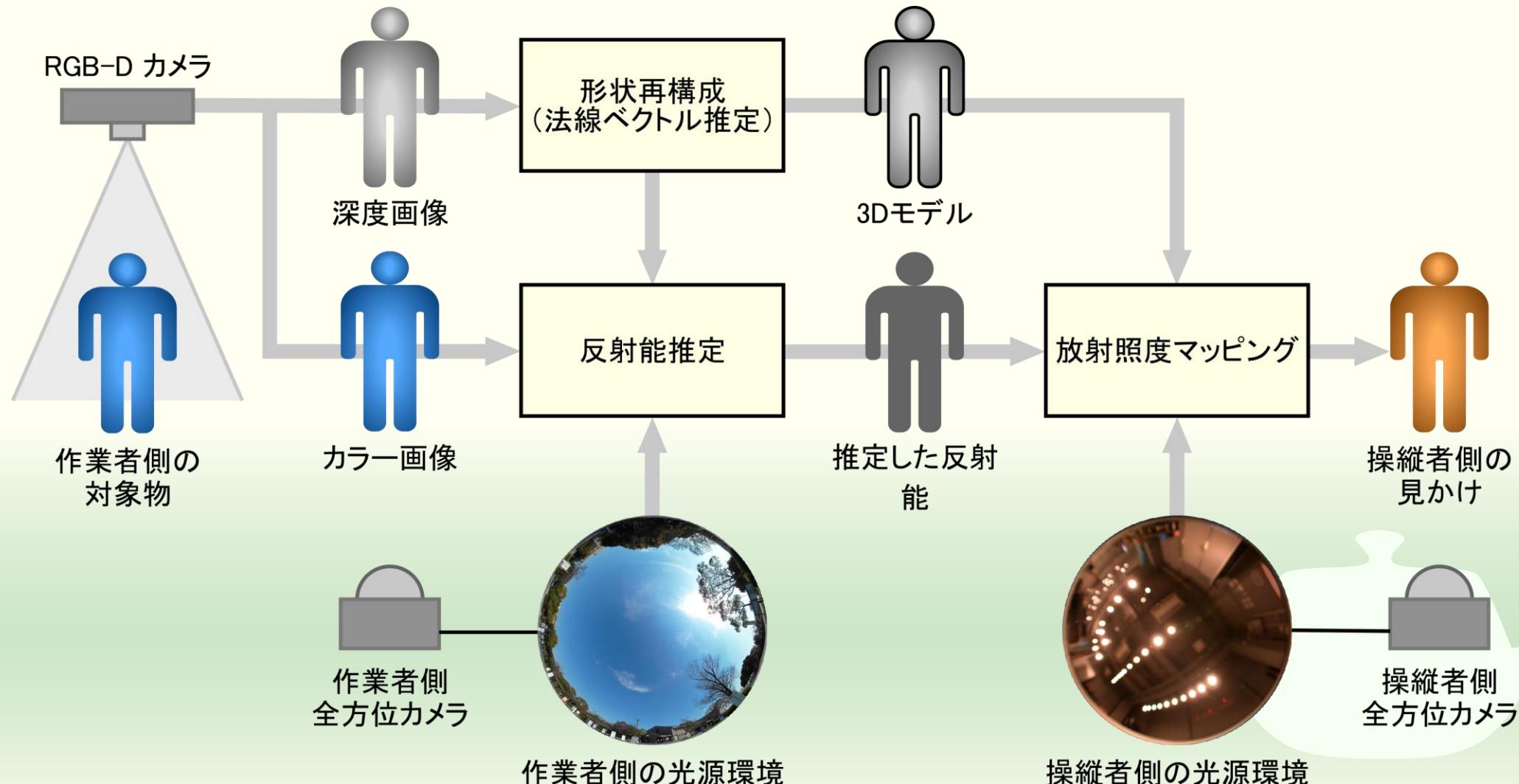


# 環境遮蔽を考慮した放射照度マッピング

- 画素  $\mathbf{x}_0$  の周りに球状に点  $\mathbf{x}_0 + r_j \omega_j$  を散布する
- 散布した点に対してデプステストを行う
- デプステストに成功した点の方向  $\omega_j$  の天空の放射輝度  $L_i(\omega_j)$  に、その方向の余弦  $\cos \theta_i$  をかけて合計する



# 光源環境の置き換え手順



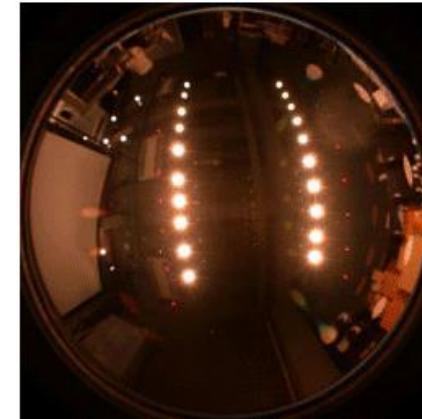
# 実験

- 2台の深度カメラで物体を撮影
- 異なる環境の天空画像による陰影の変化を確認

対象 椅子、人



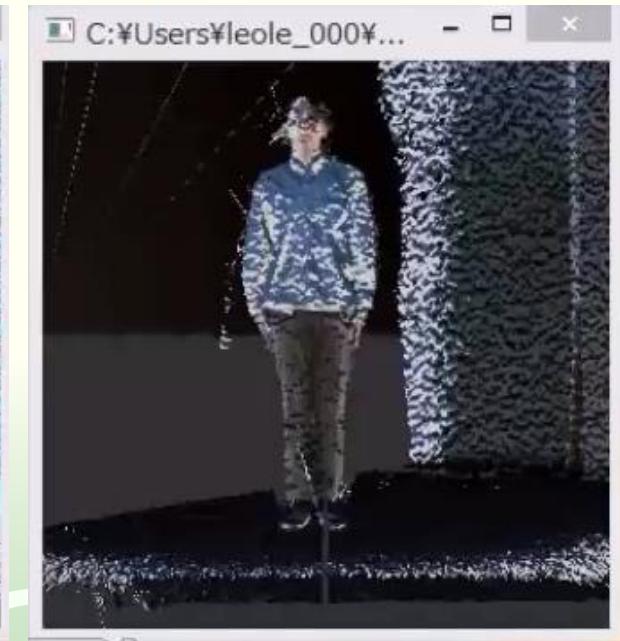
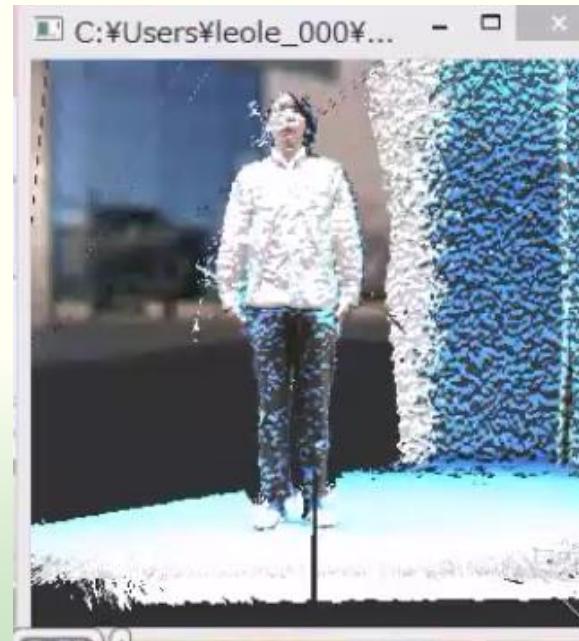
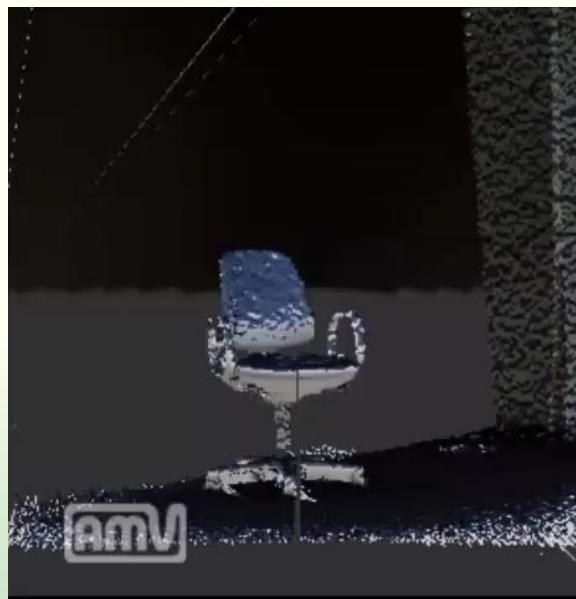
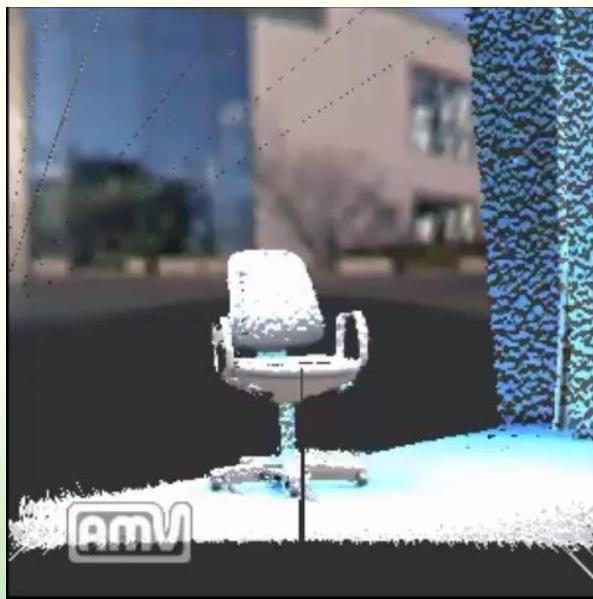
撮影環境 屋内



合成先環境 屋外 (昼) 屋外 (夜)



# 実験結果



汚い



# おわりに

## 現状を振り返って

# 多様な要素技術の統合が課題

- 形状再構成（モデリング）
  - 全周距離計測（遠景），デプスマップ融合（近景）
- 再照明（レンダリング）
  - 放射照度計測，反射能推定，放射照度マッピング
- データ伝送（通信）
  - 符号化，エラーリカバリ，保存と再生
- モーションキャプチャ（アニメーション）
  - ロボット操作，動作シミュレーション
- 自由視点観測（AR/VR）



# 今後の計画

- キレイにする
  - 距離測定のノイズ除去・境界判定の精度向上と安定化
  - バイラテラルフィルタ・カルマンフィルタ・SGM の導入
- 映像信用コードックの変更
  - WebRTC などの利用 (現在は Motion JPEG ベース)
  - トラッキング情報などとの同期
  - 部分更新・注視領域 (ROI) の詳細化
- シーン記述の変更
  - 同じ JSON ベースの glTF (GL Transmission Format) に合わせる



# 今後の計画

## ■ 複合センサシステムの開発

- RGB-Dカメラとモーションセンサを組み合わせた複合センサシステムの開発（単体でリアルタイムSLAM可能）
- ステレオカメラ、RGB-Dカメラ、モーションセンサを組み合わせた複合センサシステムの開発（RGB-Dカメラの計測不能領域の補填）



# プラットフォームの変更

- Windows 上で実装中
  - Windows 特有の機能は使っていない
- ROS への対応
  - ロボットを相手にすることを考えるなら...
- Linux / Android への移行
  - 低消費電力なデバイスへの対応
  - Oculus Quest などへの対応

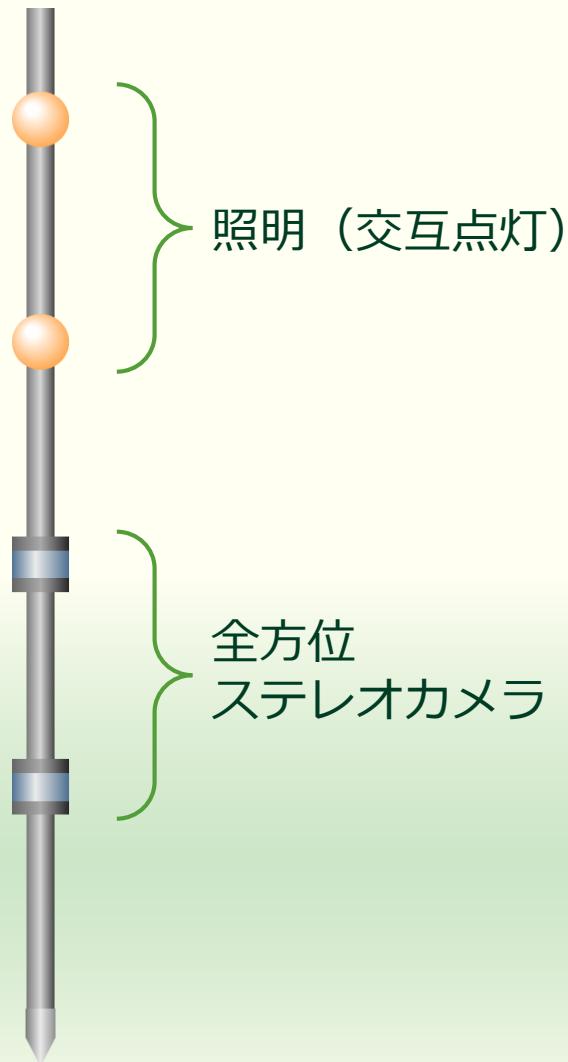


# 遠隔作業の実現

- 遠隔環境の精密な取得と再現
  - 視覚環境としてだけでなく作業シミュレーションのため
- 物理シミュレーションによる作業予測
  - 操縦者の操作による結果の予測映像を重畠表示する
- 成功した操作コマンドのバッチ送信
  - 実際の結果との比較をもとにリカバリする
- 高度な自律性をもったロボットが必要
  - 失敗を検知してリカバリできる
  - 遠隔地の状況や環境変化に対応できる



# ロボットの代わりの棒状プローブ



- 穴の中は暗い
  - おそらく照明が必要
- 全方位ステレオカメラ
  - ステレオ計測による距離の取得
- 複数の照明
  - 照度差ステレオによる勾配の取得
- 穴が巨大だと難しい?

# まとめ

- 遠隔地の視覚的環境を観測者の周囲に再現
  - 遠隔地のものや人を目の前に連れてくる実験システム
- UZUME 計画への応用
  - 遠隔地の遠景の取得
  - 遠隔地の近景の取得
  - 光源の置き換え
  - システムの統合は未完
- ソースコードは GitHub にあります
  - <https://github.com/tokoik/ted/>





# おわり

ありがとうございました

# 通信路の伝送遅延への対処

- あらかじめ伝送された観測対象の測定データとその環境データをもとに**手元にあるもの**として観測する
  - その手段の一つとしての光源環境の置き換え
- あらかじめ伝送されたロボットの周囲の環境映像から操縦者の視野の映像を切り出して表示する
- 操縦者の頭部の動き（視線の移動）に伴う表示の遅れを見かけ上解消することができる



# パノラマステレオ

- ステレオカメラをパンチルトヘッドで回転してもパノラマステレオ映像は生成できない
  - カメラが回転軸からオフセットしているために全方位画像を生成する際に画像がぶれる
  - 全方位画像上ではエピポーラ平面が一意に定まらない
- 
- 深度画像を用いる
  - 探査対象の形状測定は必要
  - ステレオカメラ, 単眼カメラ, 深度センサ等を併用

# 全方位画像を用いた場合の問題

- データ量が大きい
  - 注視方向以外も伝送される
- 注視方向の解像度が低い
  - 全方位画像の一部を切り取って拡大している



# 視野に重畠する仮想物体

- 世界座標系に配置する物体
  - ロボット, 対象物等
- 操縦者の頭の動きに追従する物体
  - 操縦者の視線マーカーなど
- ロボットのカメラの動きに追従する物体
  - ロボットの視線マーカーなど



# 今後の課題

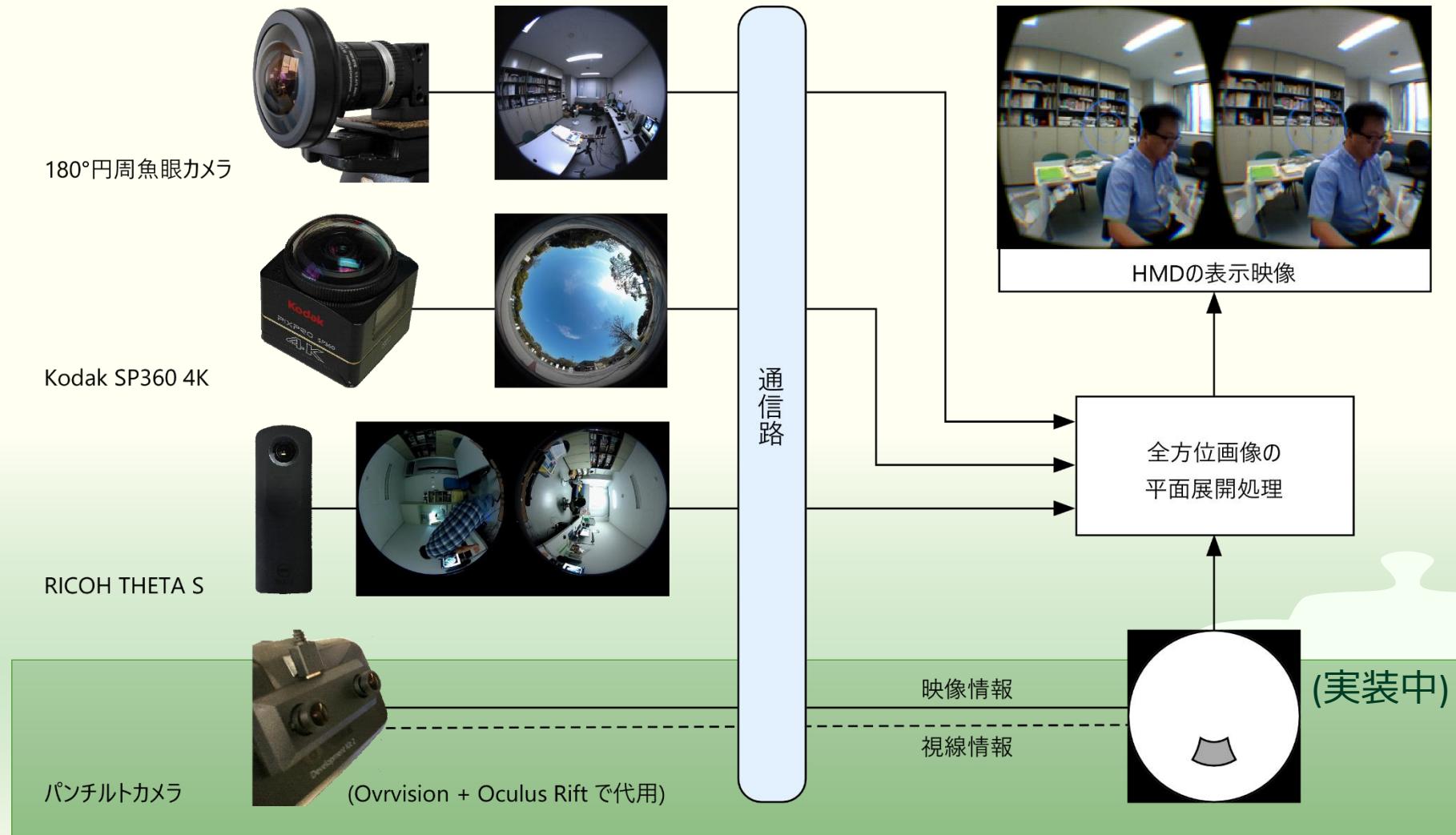
- この目的に適した環境映像の伝送手法の開発
- 環境の深度情報の取得と応用
  - パノラマステレオの実現
  - 探査対象との干渉を考慮した重畳表示
- 実際のロボットや操縦者のジェスチャーセンサーをこのシステムと接続
  - システムの操作性やロボットによる作業性の評価
  - このシステムによる遠隔ロボット制御フレームワークの検討



# 提案手法

遠隔環境の取得

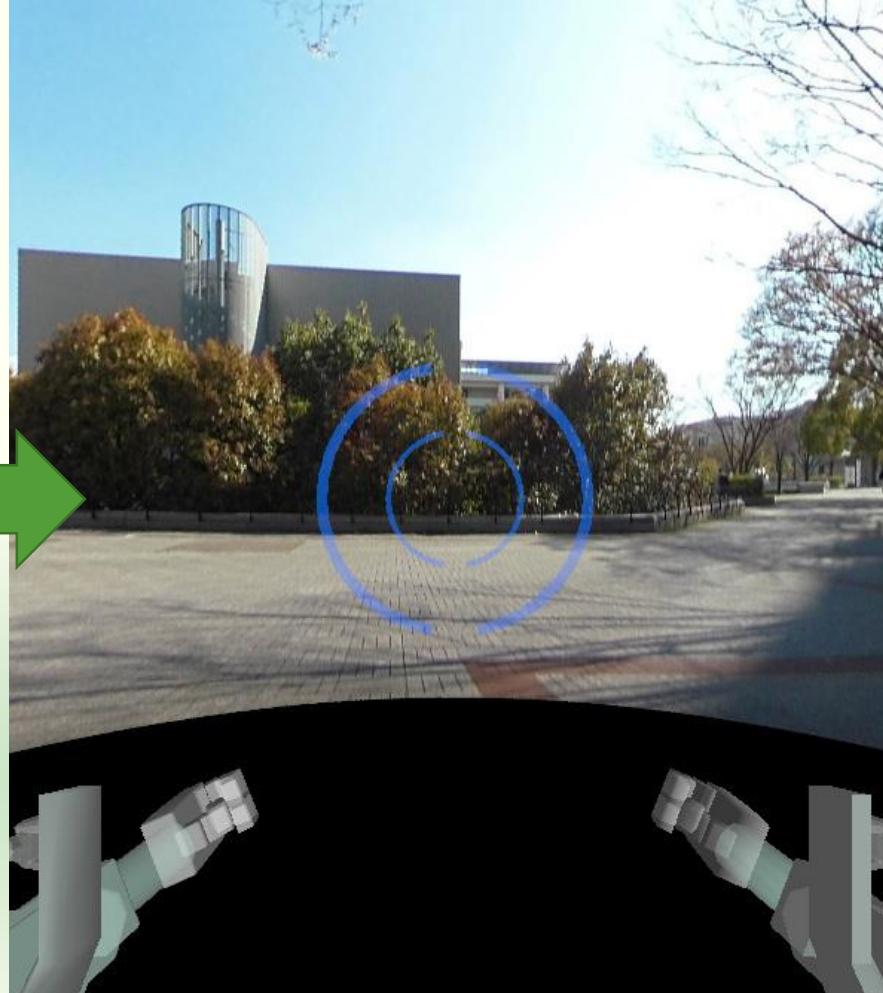
# 全方位画像の取得と再生



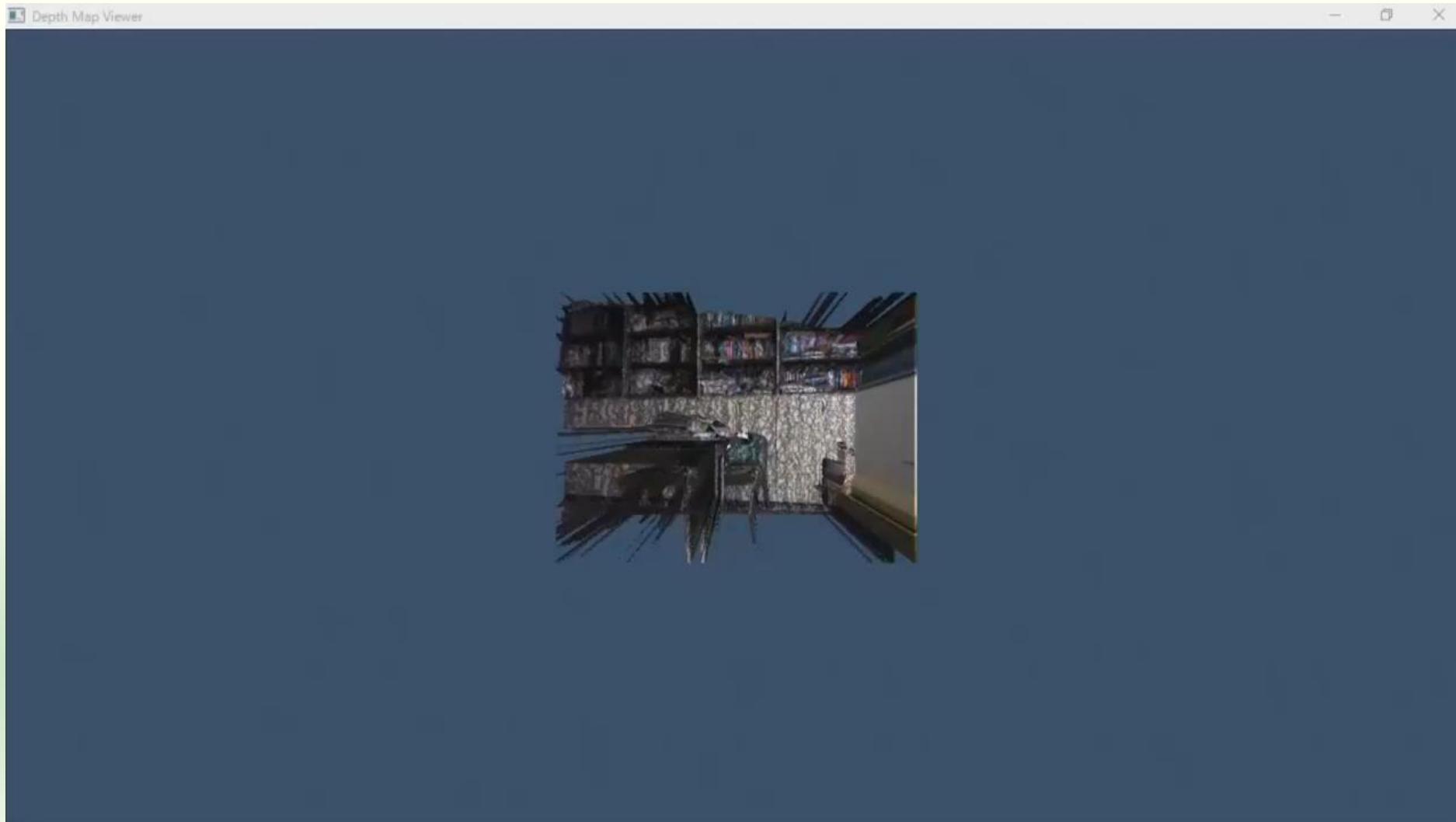
# 全方位画像の平面展開処理



Kodak PIXPRO SP360 4K (画角220°)



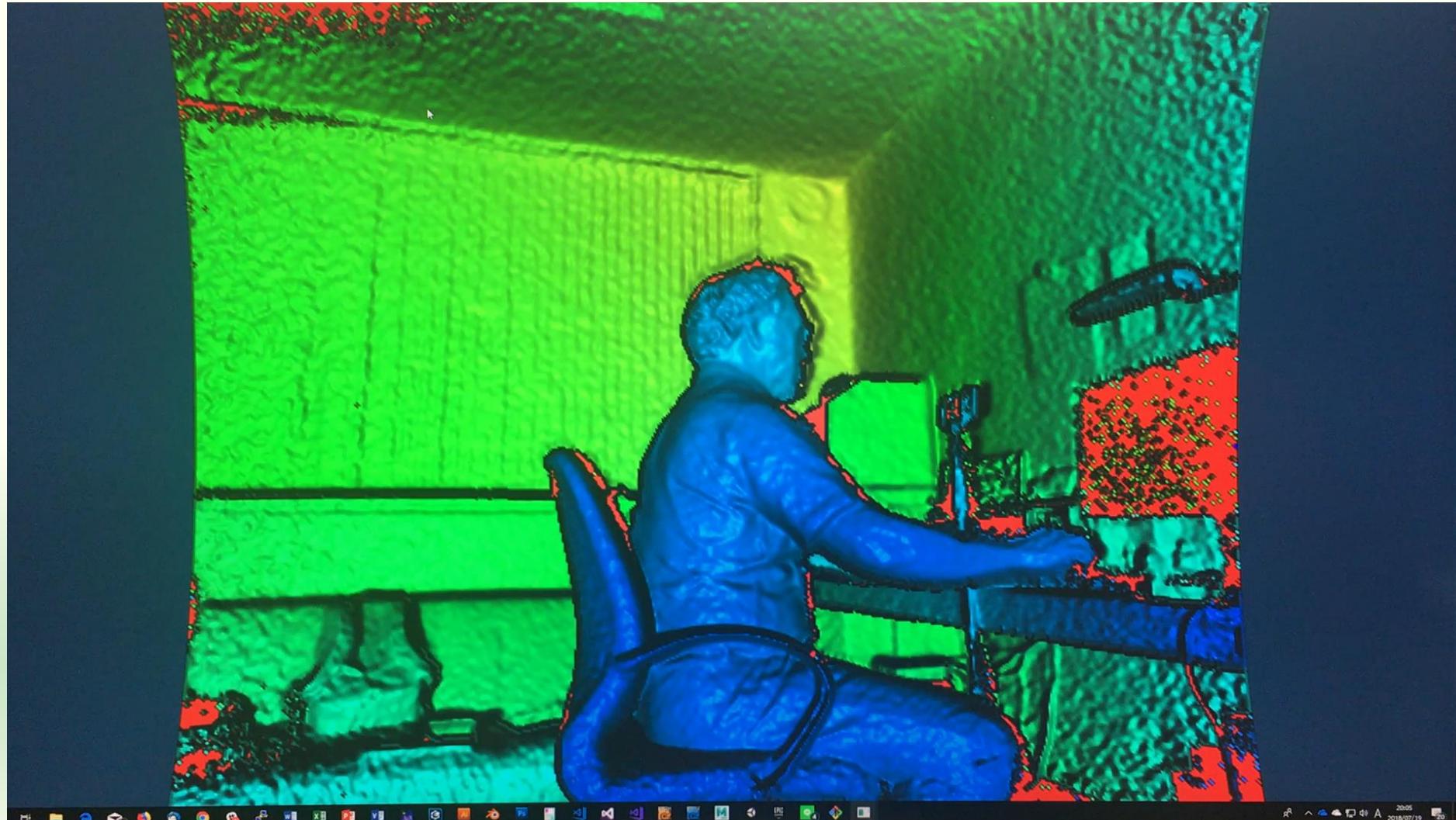
# 深度センサ



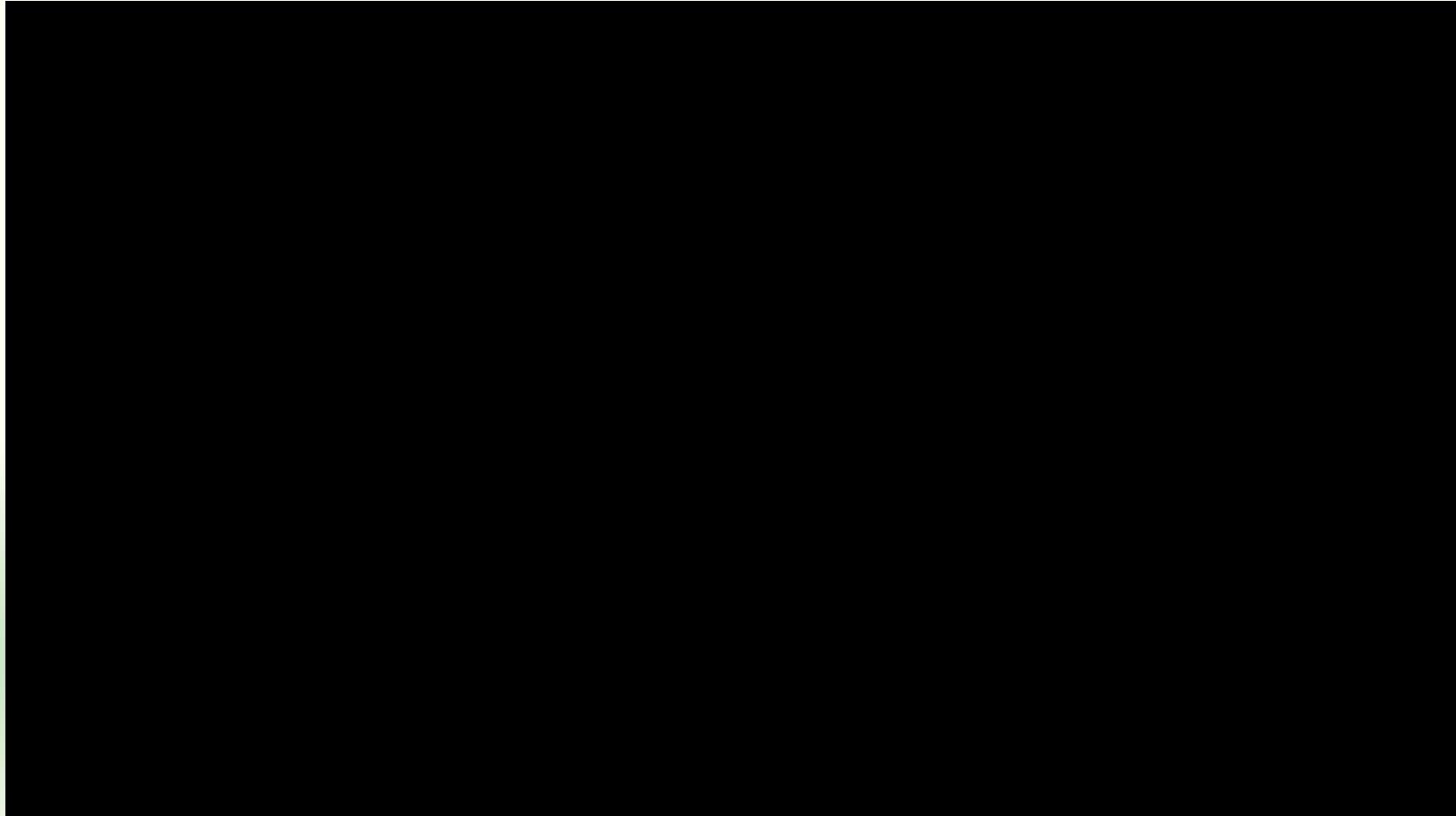
# 深度画像の平滑化



# 距離情報のフィルタリング



# 人体全周プロジェクションマッピング



# Leap Motion コントローラ



# 姿勢情報の伝送



# 反射率の推定

$$\text{反射率} = \frac{\text{反射した光}}{\text{入射した光}}$$

- ・鏡面反射
- ・拡散反射



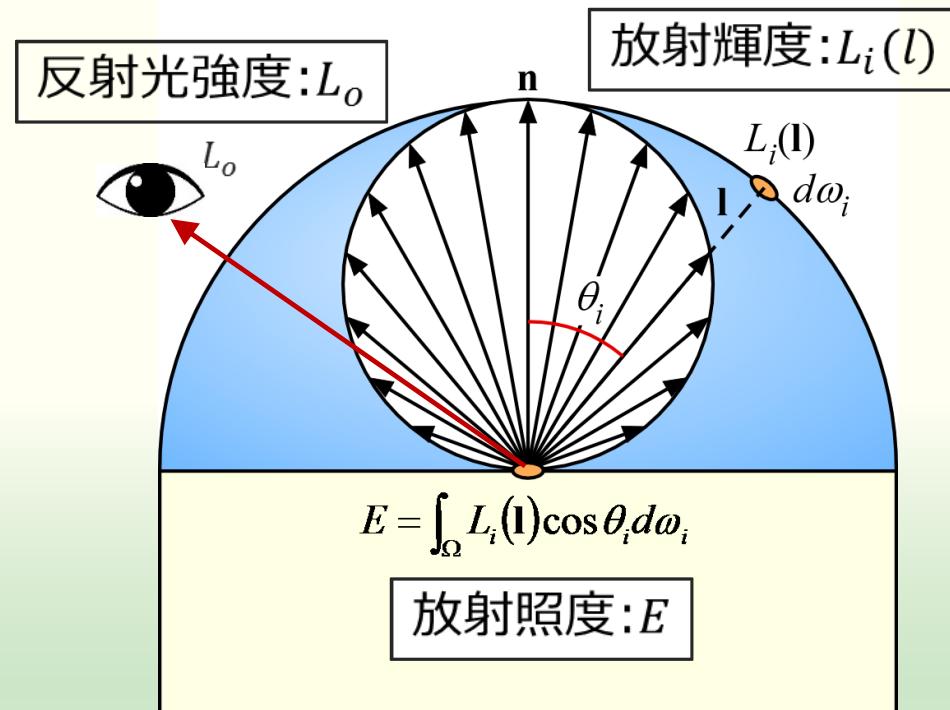
RGB-D カメラのカラー  
画像

・放射照度



全方位画像

# 放射照度の見積もり

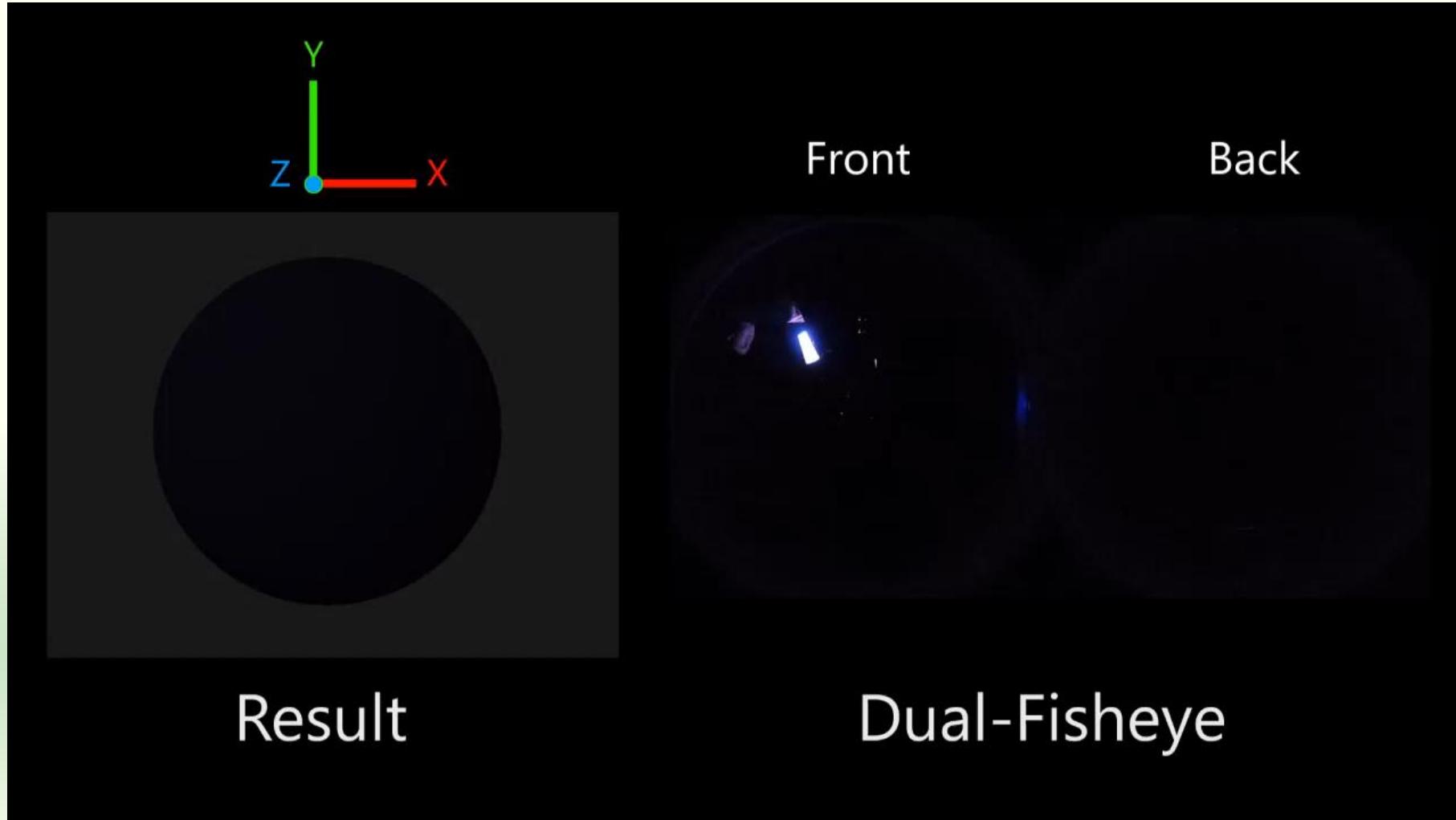


天空画像より放射照度  $E$ 、  
カラー画像より反射光強度  $L_o$   
が算出できる。

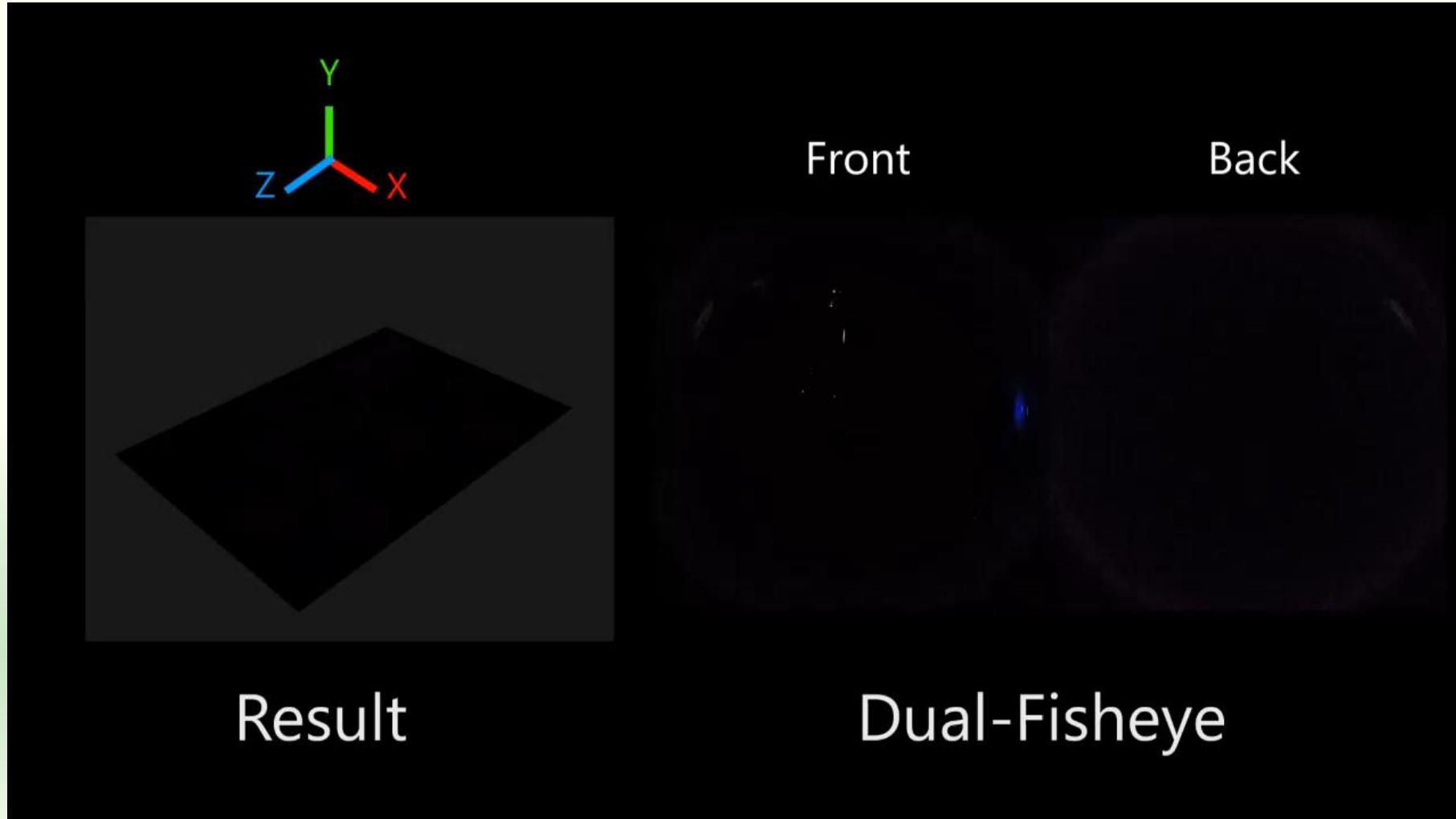
鏡面反射を持たなければ、

反射率 :  $f = L_o/E$

# 放射照度計算のリアルタイム化



# 放射照度計算のリアルタイム化



# 色再現性の検証

全周画像から陰影付けしたCGモデル



■ 同じ光源下で撮影した実物体

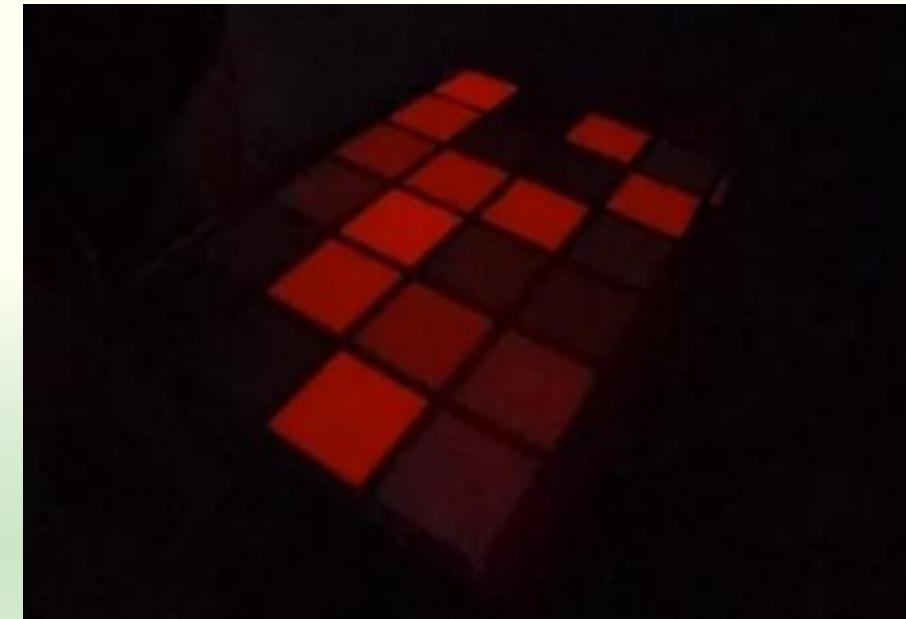


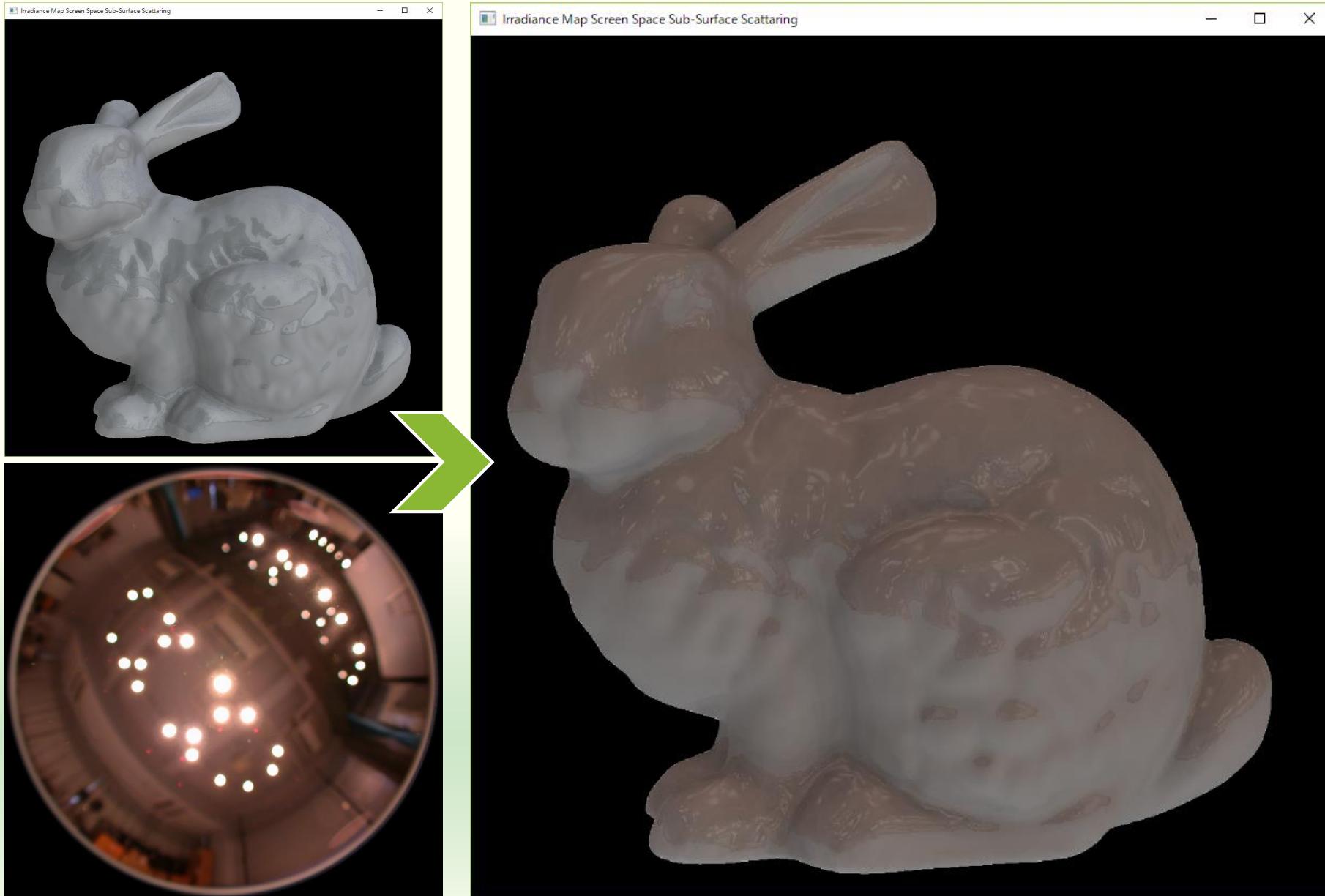
# 色再現性の検証

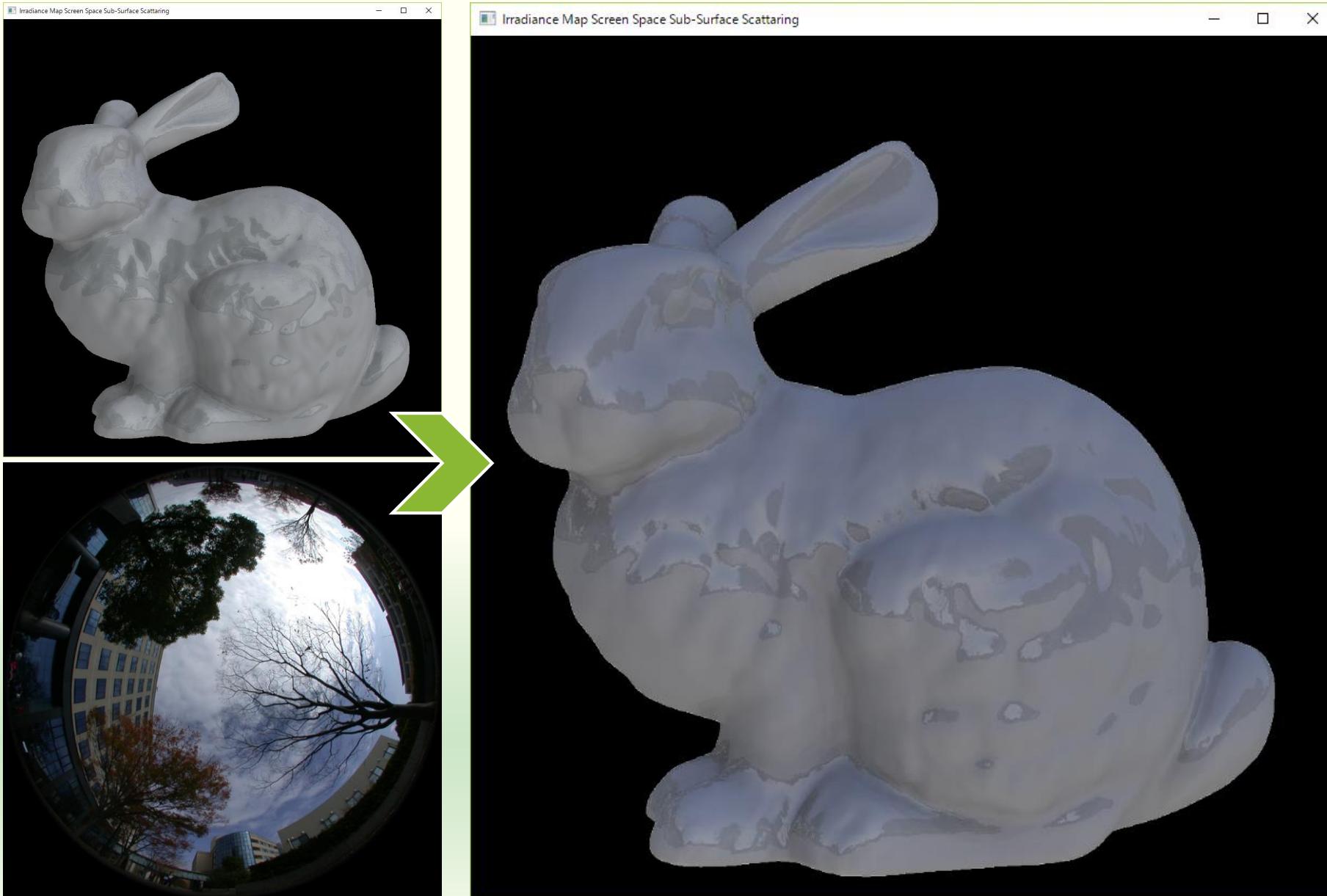
全周画像から陰影付けしたCGモデル



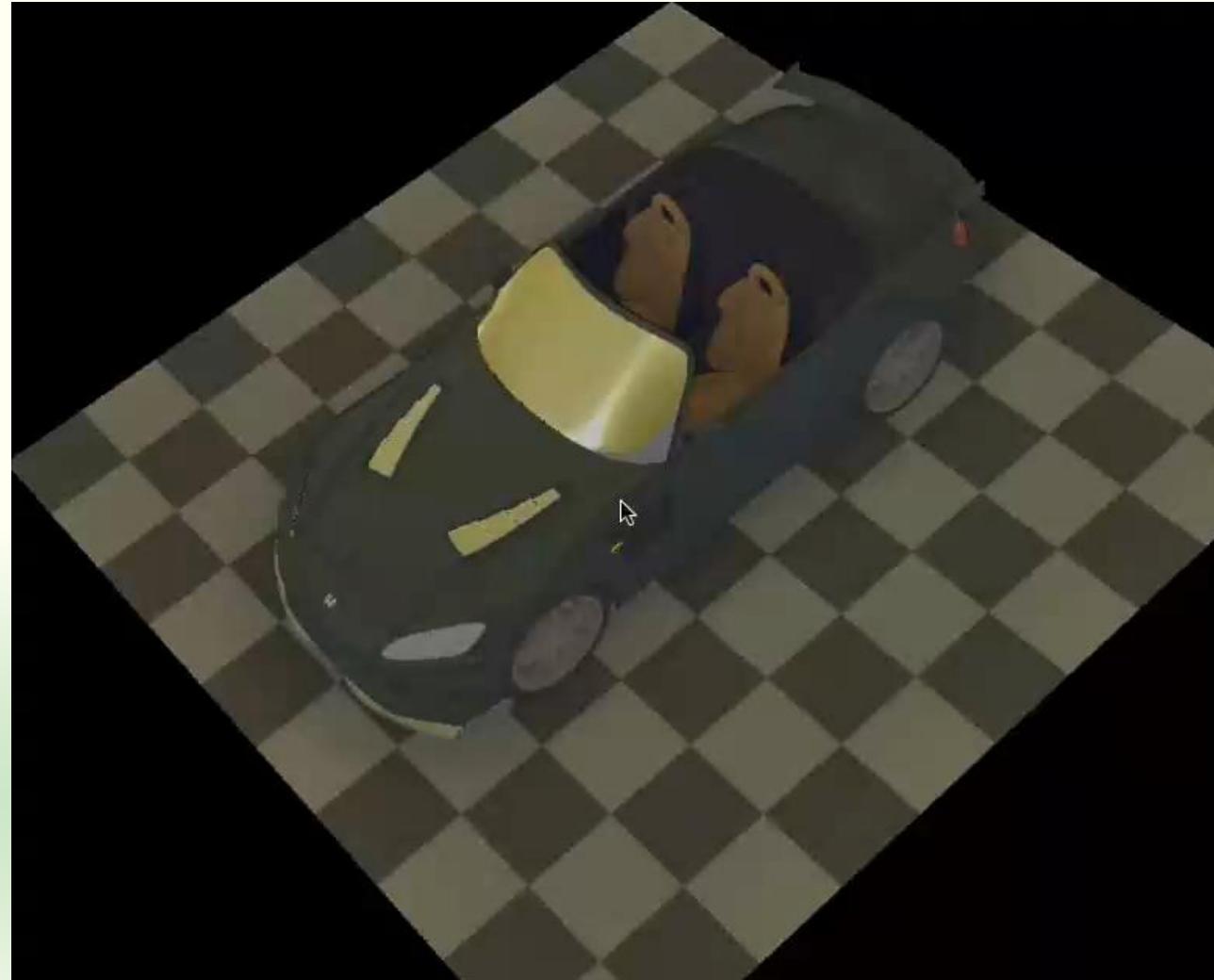
■ 同じ光源下で撮影した実物体



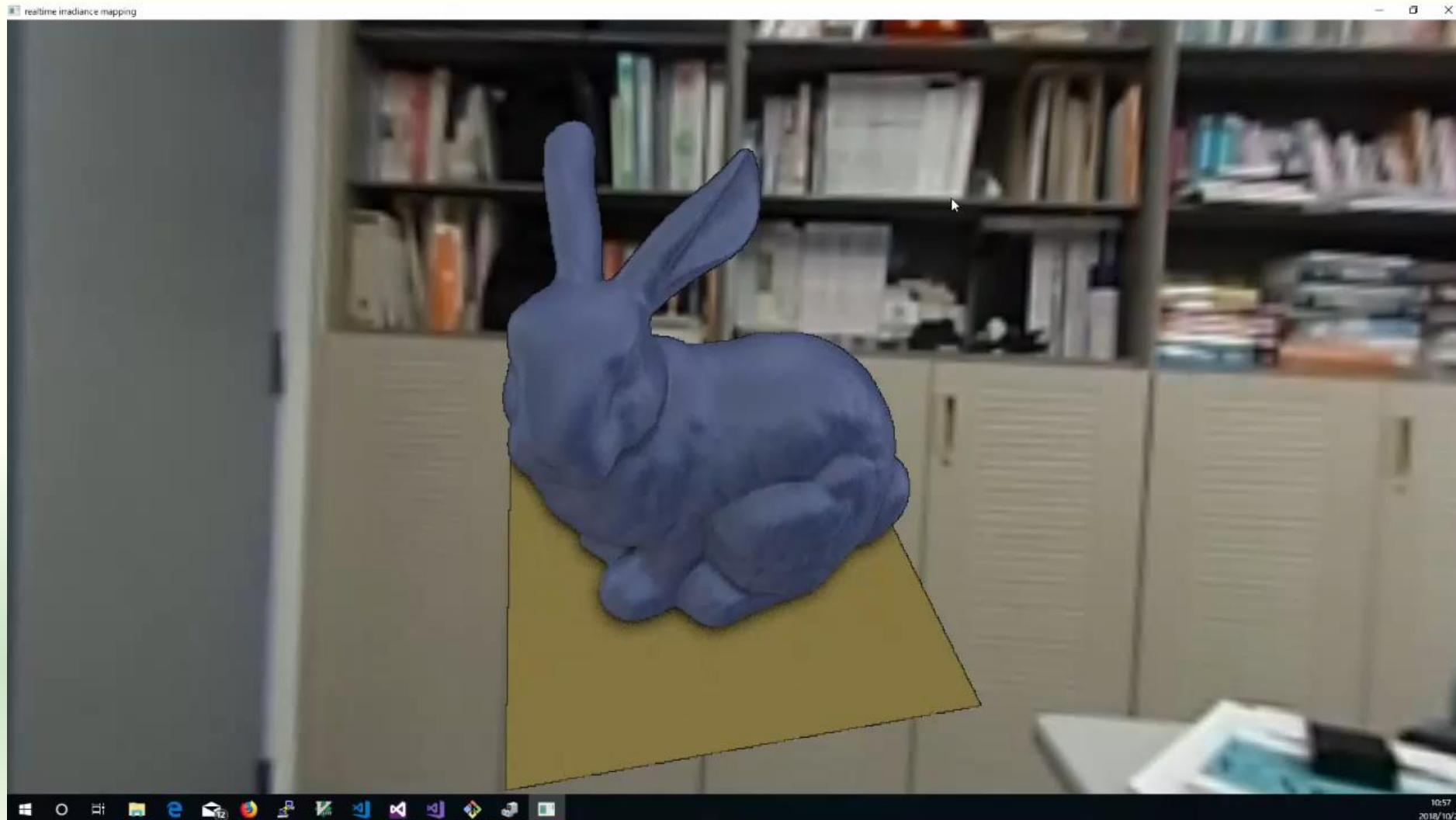




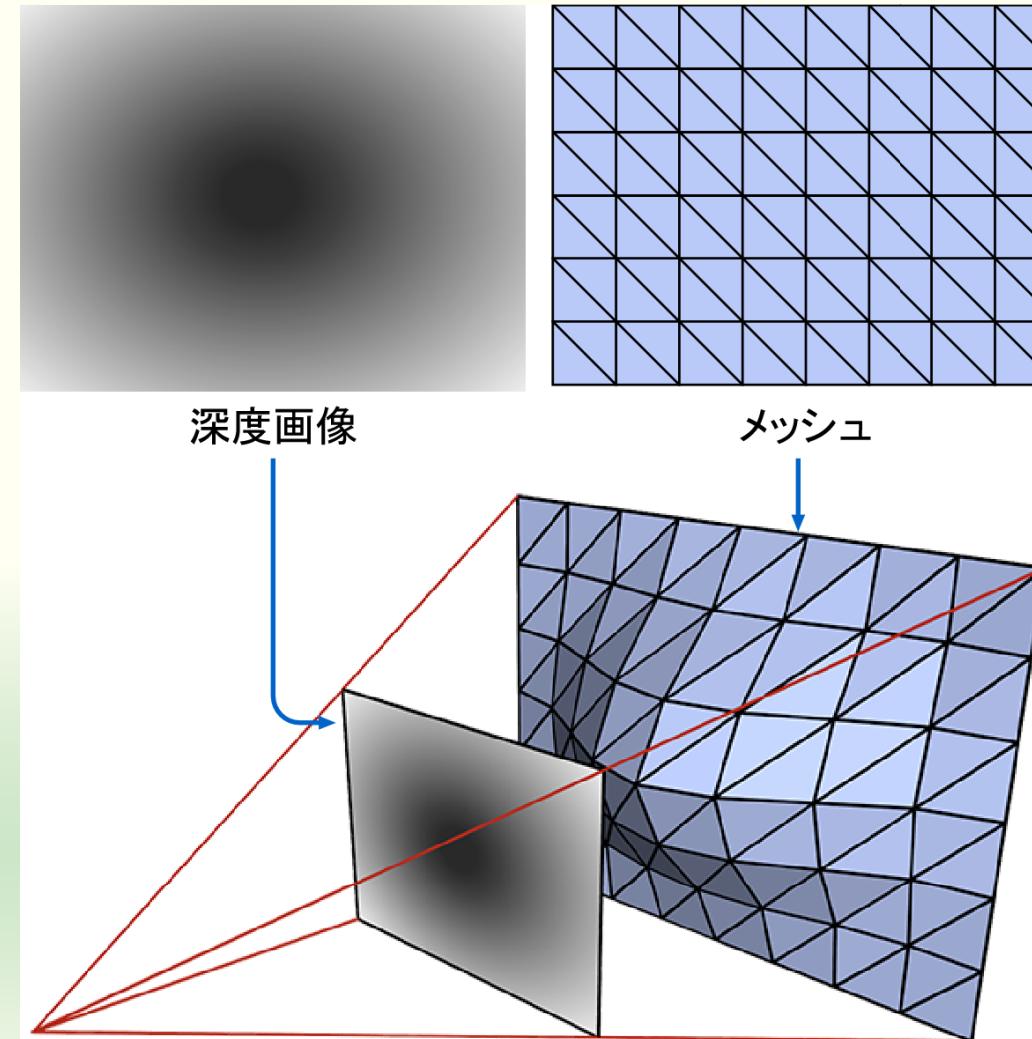
# 光源環境の動的変化



# リアルタイム放射照度マッピング



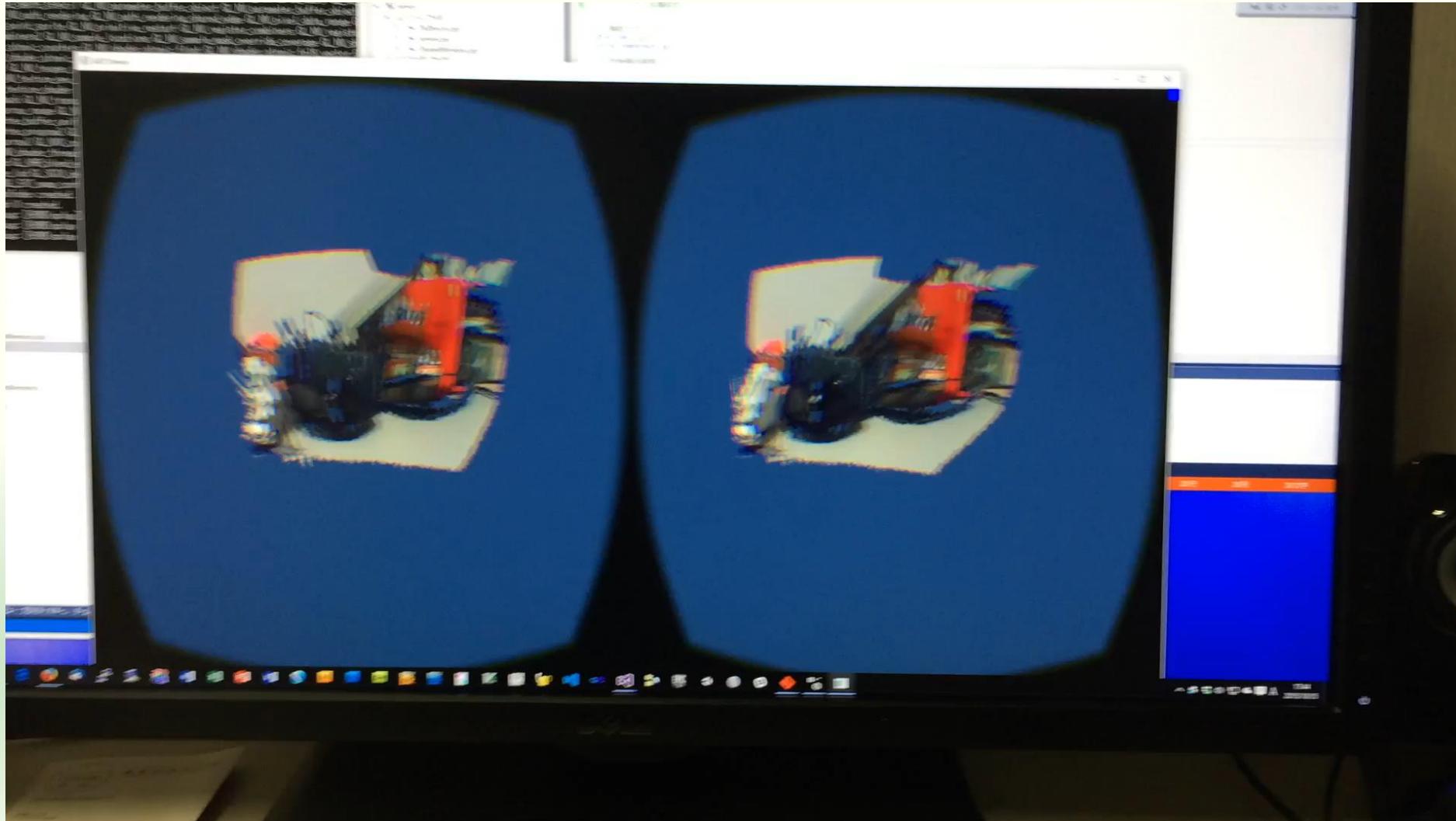
# デプスマップによる変位マッピング



# デプスマップを使った自由視点表示



# デプスマップを用いた自由視点観測





# 実施状況

## 現状報告

# 遠隔環境の取得と伝送

## ■ 光源環境と視界

- 全方位カメラを使用して取得した光源環境を操縦者側に伝送し任意の方向の視界の切り出しが完了
- 全方位カメラからの映像取得および伝送の際の符号化方式の改善が課題

## ■ 近景と遠景の三次元情報

- RGB-Dカメラによる近景の取得(形状再構成)は完了
- 全方位カメラによる遠景の距離情報の取得は実装中
- 三次元情報の伝送についても実装中

# 操縦者環境の取得と伝送

- 姿勢情報（ロボットの制御）
  - HMD のヘッドトラッキングおよび Leap Motion による操縦者の頭部と両手の姿勢の取得と伝送は完了
  - 操縦者と作業者双方の姿勢を交換することも可能
- 光源環境と視界（遠隔行動制御への応用）
  - 操縦者側の光源環境や視界を作業者側に伝送することは当初想定していなかったので未完



# 遠隔環境の再現

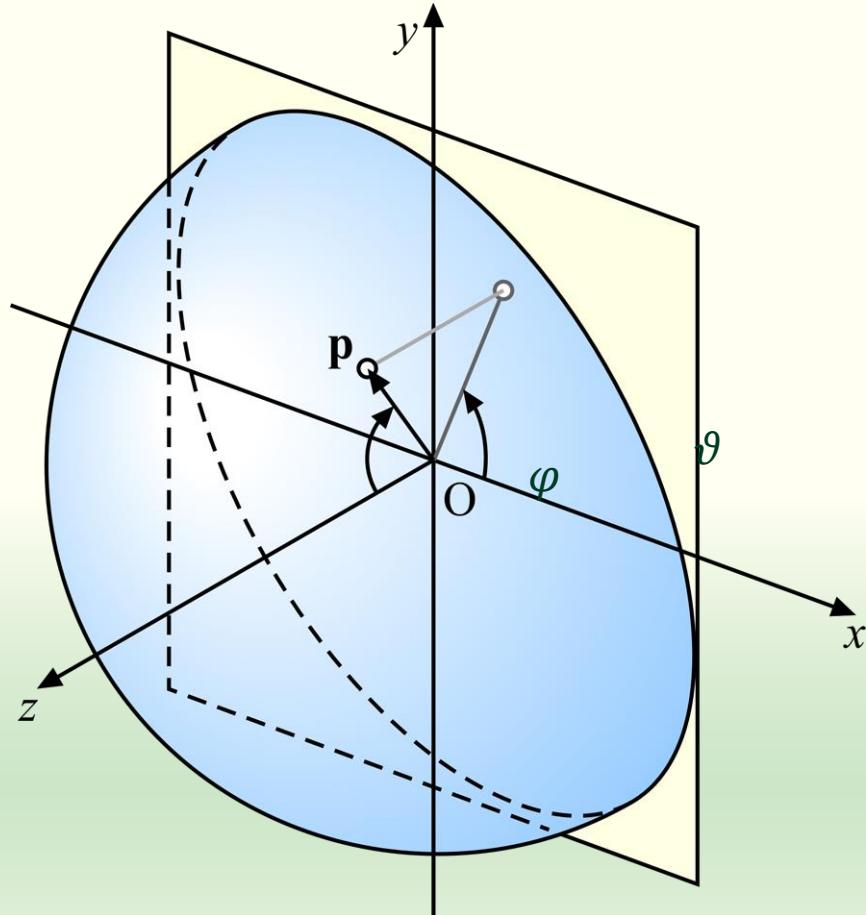
## ■ 再照明

- 再照明の手法についてはほぼ完成
- システムへの実装は三次元形状の伝送が完了次第
- 取得した距離情報のフィルタリング手法の改善中
- RGB-Dカメラ 자체を更新中

## ■ 自由視点観測

- 全方位カメラによる遠景の距離情報の取得および形状データの伝送の実装が完了していないため未完

# 等距離射影方式の魚眼レンズの撮影空間



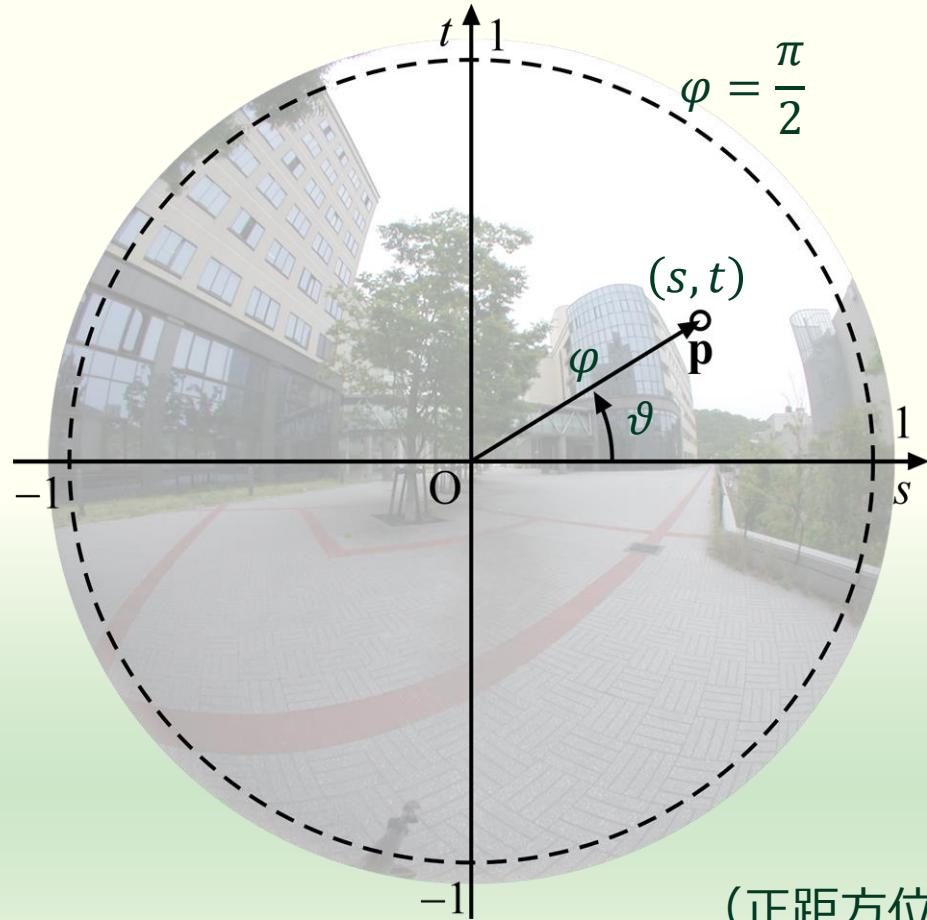
$\mathbf{p} = (x, y, z)$  として

$$\vartheta = \tan^{-1} \frac{y}{x}$$

$$\varphi = \cos^{-1} z$$



# 魚眼画像上の位置



$$s = \frac{2\varphi}{\pi} \cos \vartheta = \frac{2x \cos^{-1} z}{\pi \sqrt{x^2 + y^2}}$$

$$t = \frac{2\varphi}{\pi} \sin \vartheta = \frac{2y \cos^{-1} z}{\pi \sqrt{x^2 + y^2}}$$

# 二重魚眼画像の場合

$z \geq 0$  のとき



$$s = \frac{2x \cos^{-1} z}{\pi \sqrt{x^2 + y^2}}$$

$$t = \frac{2y \cos^{-1} z}{\pi \sqrt{x^2 + y^2}}$$

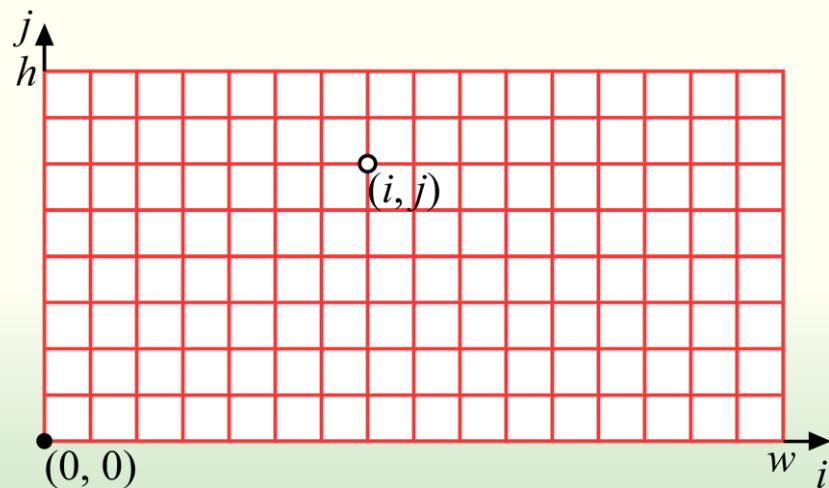
$z < 0$  のとき



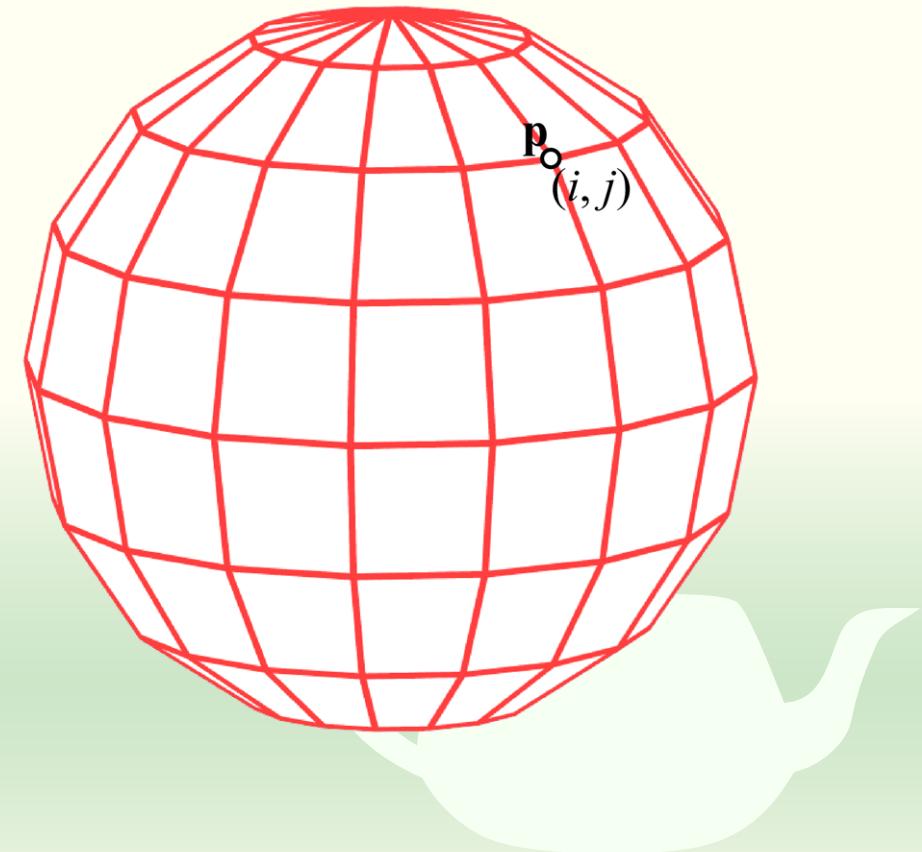
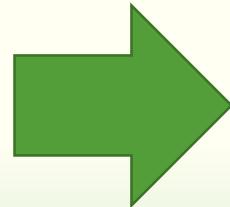
$$s = \frac{-2x \cos^{-1}(-z)}{\pi \sqrt{x^2 + y^2}}$$

$$t = \frac{2y \cos^{-1}(-z)}{\pi \sqrt{x^2 + y^2}}$$

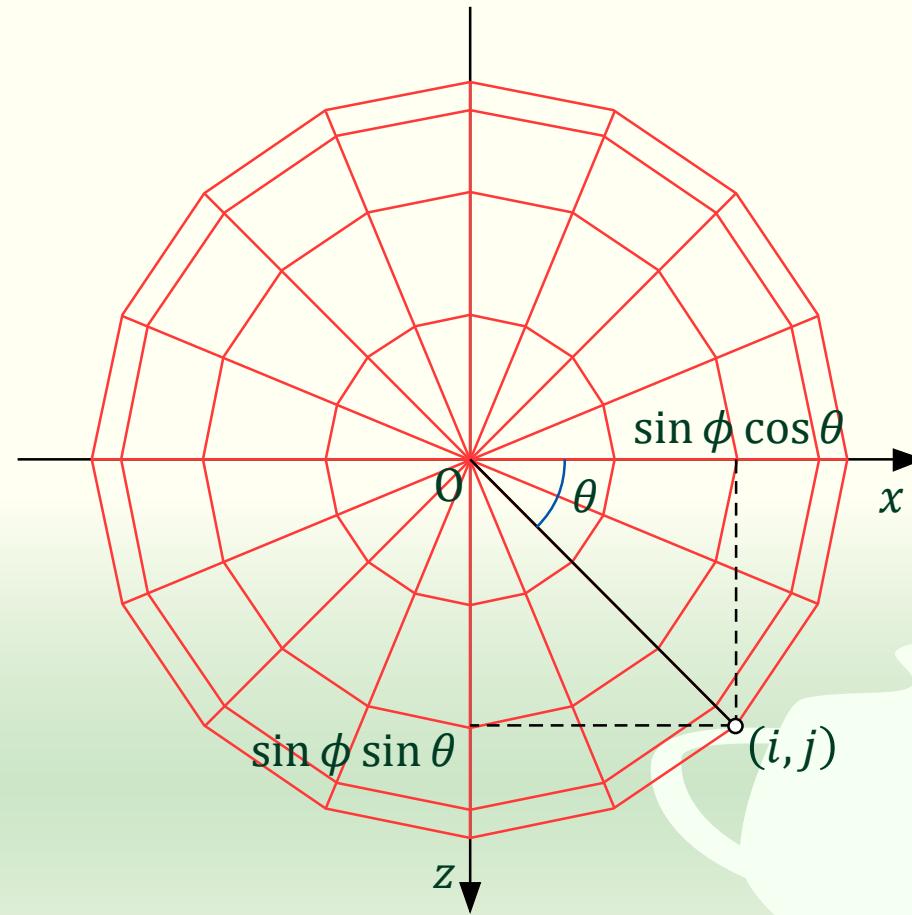
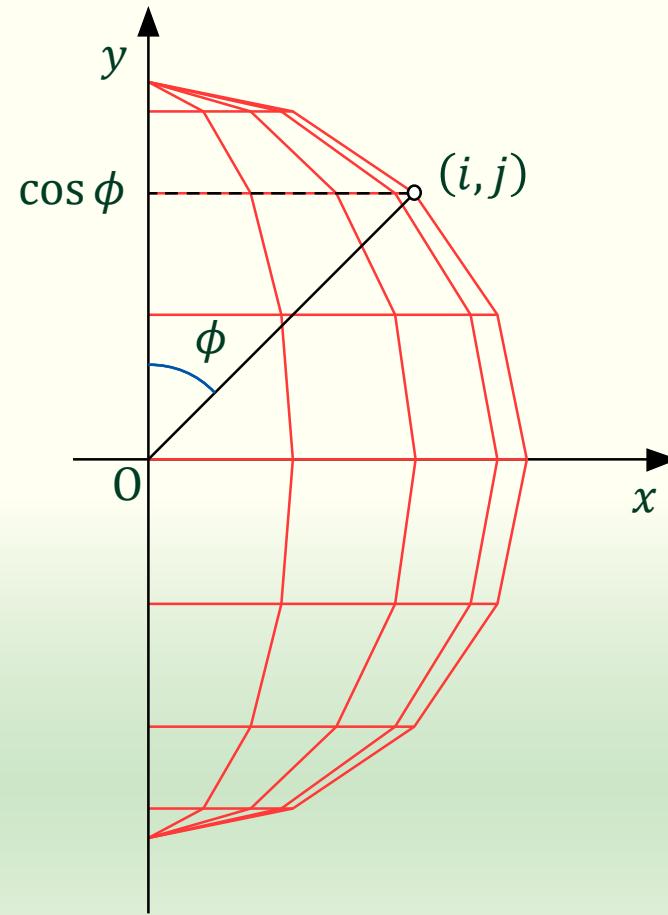
# 球状のメッシュ



(正距円筒図法)



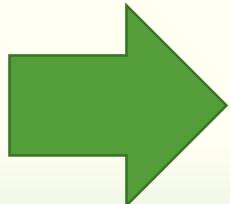
# 格子点の位置



# 格子点の位置

$$\theta = \frac{2\pi i}{w}$$

$$\phi = \frac{\pi j}{h}$$

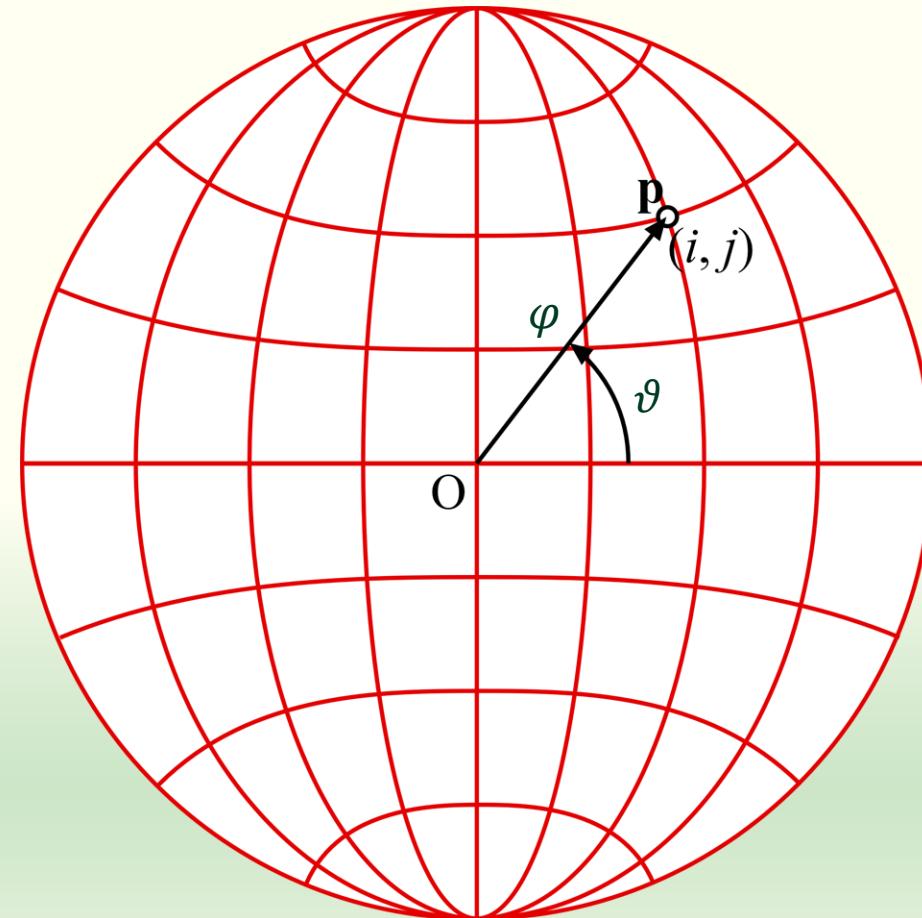


$$x = \sin \phi \cos \theta$$

$$y = \cos \phi$$

$$z = \sin \phi \sin \theta$$

# 魚眼画像上のメッシュ



# 緯度 $\phi$ を $\delta$ ずらす

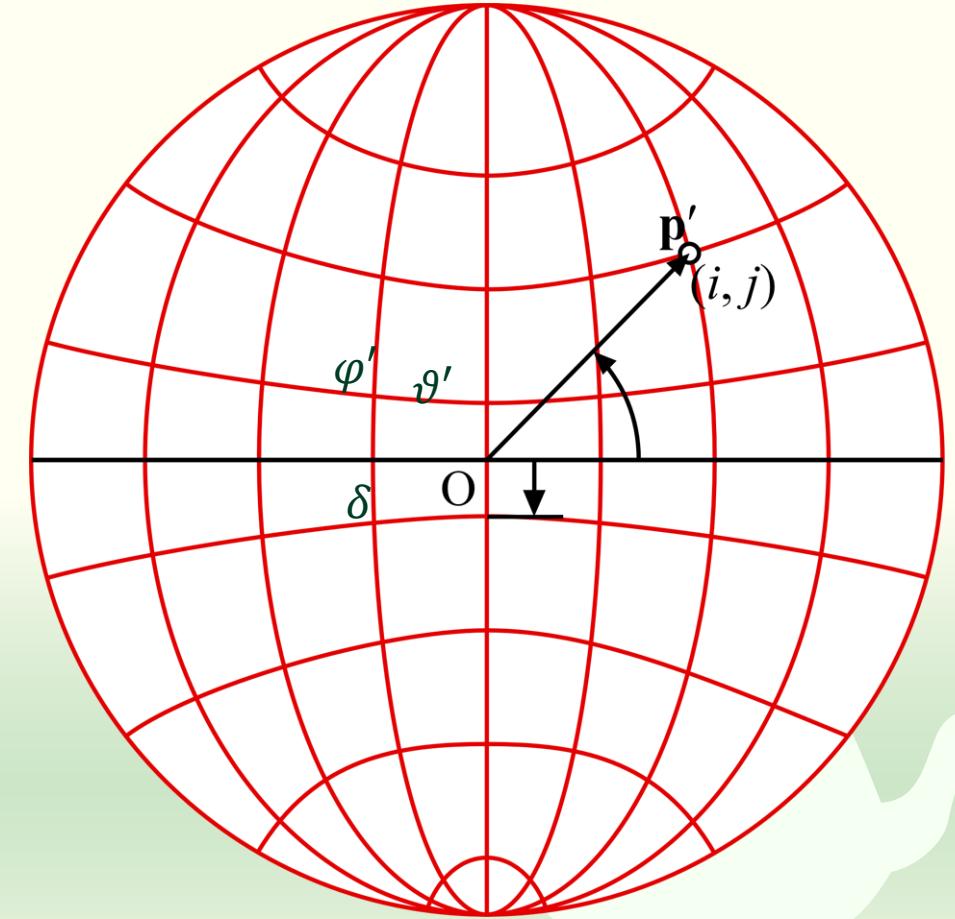
$$x' = \sin(\phi - \delta) \cos \theta$$

$$y' = \cos(\phi - \delta)$$

$$z' = \sin(\phi - \delta) \sin \theta$$

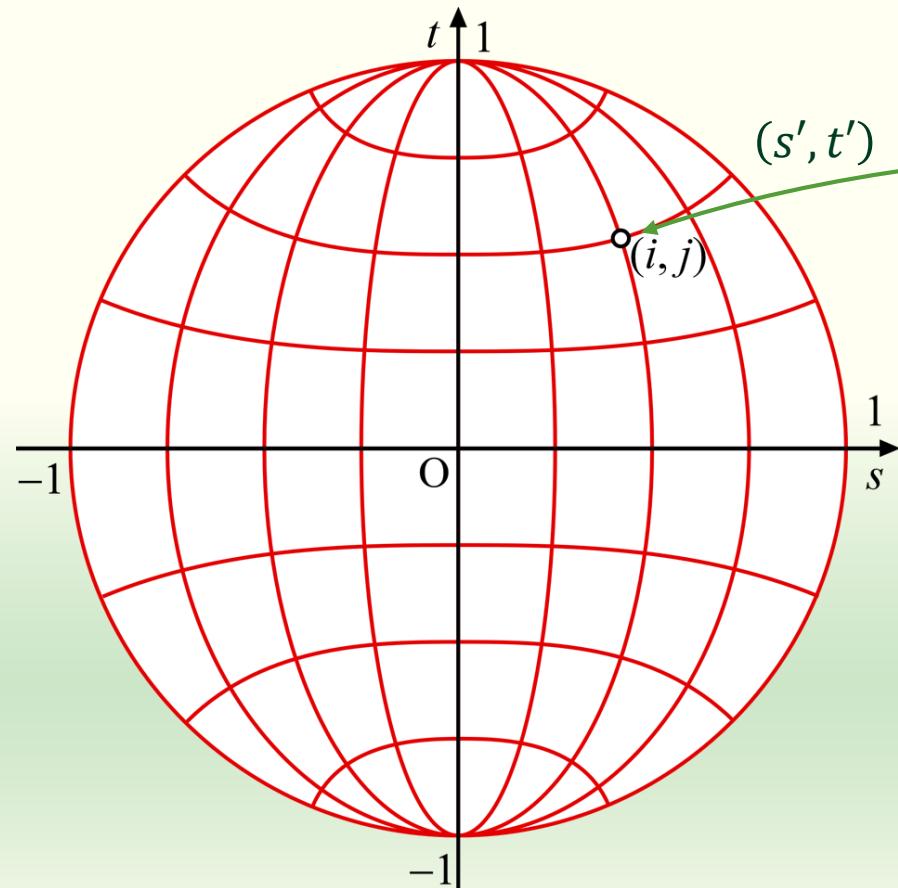
$$s' = \frac{2\varphi'}{\pi} \cos \vartheta' = \frac{2x' \cos^{-1} z'}{\pi \sqrt{x'^2 + y'^2}}$$

$$t' = \frac{2\varphi'}{\pi} \sin \vartheta' = \frac{2y' \cos^{-1} z'}{\pi \sqrt{x'^2 + y'^2}}$$

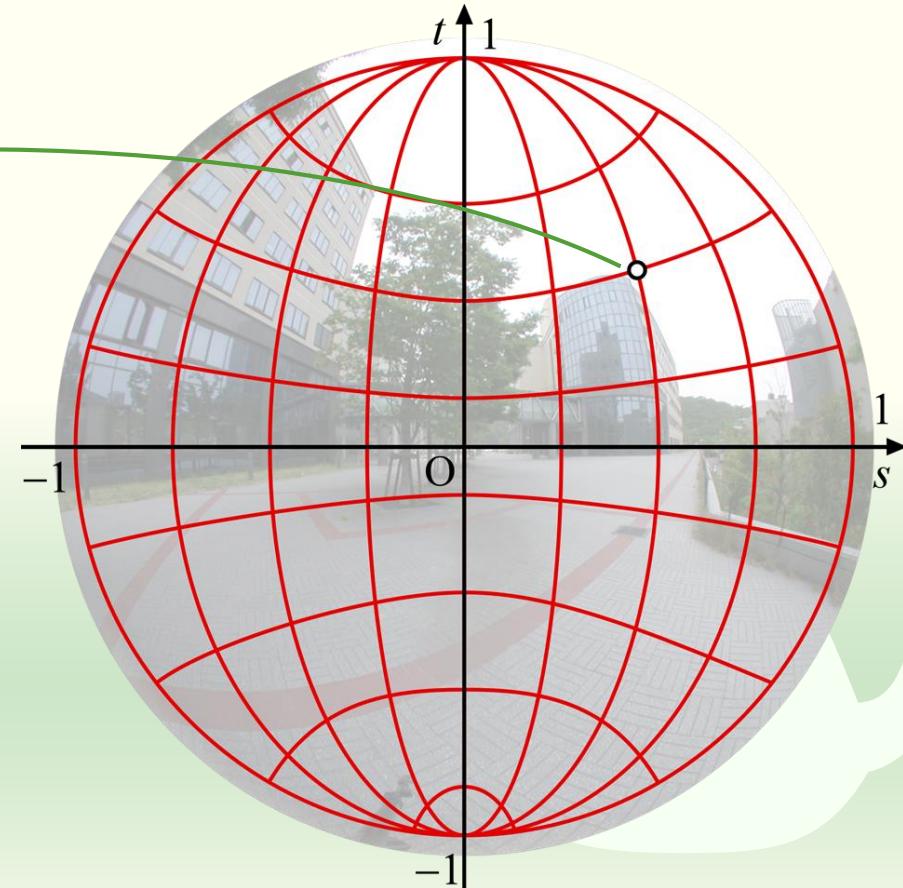


# テクスチャマッピングによる変形

描画するメッシュ



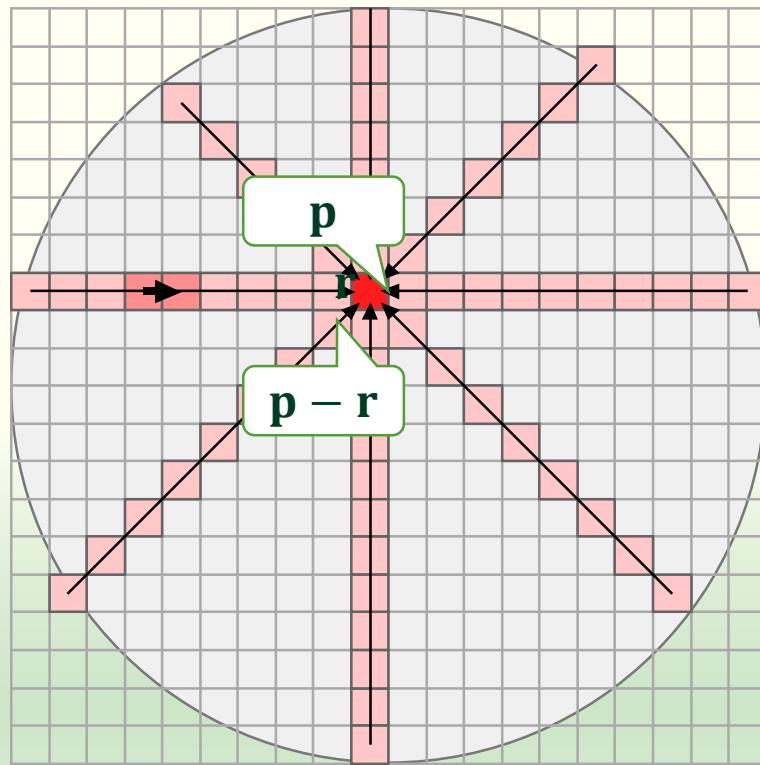
■ マッピングするテクスチャ



# 視差の連續性を考慮したマッチングコスト

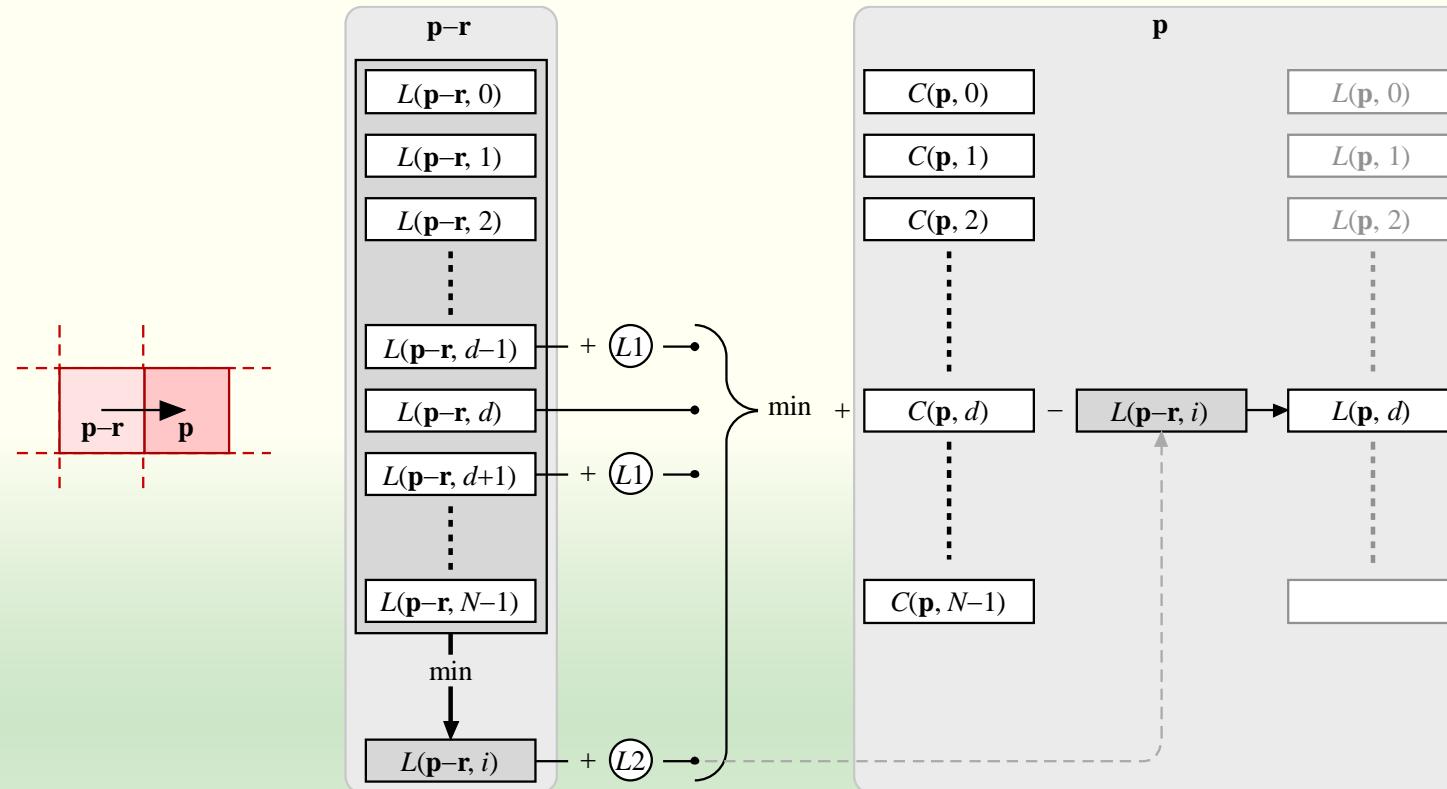
- 視差が  $d$  の時のマッチングコストのペナルティ
  - 直前に処理した画素の視差  $d$  の時のコスト
    - ペナルティなし
  - 直前に処理した画素の視差  $d+1$  の時のコスト
    - ペナルティ  $L1$  を加える
  - 直前に処理した画素の視差  $d-1$  の時のコスト
    - ペナルティ  $L1$  を加える
  - 直前に処理した画素の最小コスト
    - ペナルティ  $L2$  を加える
- 1~4のコストとペナルティーの和のうち最小のものに現在の画素のマッチングコストを加える

# セミグローバルマッチング



$$\begin{aligned} L(\mathbf{p}, d) &= C(\mathbf{p}, d) \\ &+ \min \{ \\ &L(\mathbf{p} - \mathbf{r}, d), \\ &L(\mathbf{p} - \mathbf{r}, d + 1) + L1, \\ &L(\mathbf{p} - \mathbf{r}, d - 1) + L1, \\ &\min_i L(\mathbf{p} - \mathbf{r}, i) + L2 \\ &\} - \min_i L(\mathbf{p} - \mathbf{r}, i) \end{aligned}$$

# セミグローバルマッチングの実装



# 広角(魚眼)画像を用いた視野の安定化



# 近景の取得

