

## カラー情報を用いたデプスデータの Hole 除去

日大生産工(院) ○石井 博樹  
日大生産工 目黒 光彦

### 1 まえがき

奥行きデータを安価に取得することができるデバイス「Kinect」がマイクロソフト社から発売されている。Kinect自体は室内ゲーム用で、モーションキャプチャを目的としているが、マイクロソフト社がSDKを公開したことにより、容易にデプスデータを取得できるようになったことから、Kinectを用いた研究が活発に行われている[1]-[3]。

Kinectによって取得出来るデータは一般的なカラー情報と、奥行きを表すデプスデータである。Kinectは、デプスマップの取得に赤外線プロジェクトから幾何学的パターンを照射し、赤外線カメラで撮影した幾何学的パターンのゆがみ具合を計算して奥行きを求めている。赤外線を利用していることから、主に鏡や金属などの鏡面反射の物体、エッジ部分、Kinectから極端に近い、あるいは遠い部分などのデータが欠損することがある。

そこで、本研究ではカラー情報とデプスデータを組み合わせて、欠落したデプスデータを補間する手法を提案する。注目位置の周辺の物体のカラー情報は似かよっているという仮定を元に、同じ位置にありかつ似たような色の物体周辺のデプスデータは似かよっているという仮定を利用する。欠落している画素を中心にし、その周辺のカラー情報とデプスデータを使用する。注目画素と参照画素の距離が近いほど、デプスデータの値も近い場合が多いため重みを大きくする。同様に、注目画素と参照画素のカラー情報の色合いが近いほど、重みを大きくする。この二つの重みを元にしたデプスデータにおける補間処理により、欠損した部分を埋めるHole filterを提案する。

### 2 従来の研究

Kinectのデプスデータの補間や修正する手法は、いくつか考案されている。

A. Bapat[1]らは、カラー情報を元にした、デプスデータのメディアンフィルタを提案した。カラー情報のRGB各チャンネルごとに、注目画素と参照画素の差分を求め、閾値以下であればメディアンフィルタを処理した後の参照画素のデプスデータで、注目画素のデプスデータを埋め合わせる。この処理を何回か繰り返したあとに、バイラテラルフィルタとNon Local Meansでノイズ除去を行い最終的な結果が出力される。

S. Michael[2]らは、デプスデータに発生するエッジのジグザク部分に注目した。デプスデータで取得出来るエッジは直線になってはいないことが多いので、カラー情報から信頼のおける画素を使用し、デプスデータのエッジを改善するためのガイドとして使用される。

K. R. Vijayanagar[3]らは、デプスデータの欠損はエッジと鏡面、光沢などの二種類の原因があることを想定し、自動的に二種類を分離した。その後それぞれの場所に適した補間アルゴリズムを適用するようにした。これは、単純なガウシアンフィルタより優れている。また、時間フィルタリングにより安定した補間を行うことが出来るようになっている。

### 3 提案手法

#### 3.1 CIE $L^*a^*b^*$ 色空間

カラー情報は通常、隣接画素同士ではそれほど大きく色の変化はない。この特徴を利用して、補間箇所の画素と、補間に利用する画素同士の色の近さを計算して、デプスデータを補間する。

色同士の近似度を計算するにはCIE  $L^*a^*b^*$ 色空間に変換する必要がある。通常Kinectで撮影されたカラー情報はRGB色空間である。これは、赤と緑と青の三原色を混ぜ合わせて表現

---

Hole filter of depth data using the color information

Hiroki ISHII and Mitsuhiko MEGURO

する加法混合の一種である。現在のデジタル画像では広く使われている一方、色の近似度を計算するには向かない。そこで、CIE L\*a\*b\*色空間に変換する。CIE L\*a\*b\*色空間は明度と二つの補色次元で構成されている均等色空間として設計されており、ユークリッド距離を計算することにより色の近似度を計算することができる。変換式を式(1)～(5)に示す。

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.412453 & 0.357580 & 0.180423 \\ 0.212671 & 0.715160 & 0.072169 \\ 0.019334 & 0.119193 & 0.950227 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (1)$$

$$L^* = 116f\left(\frac{Y}{Y_n}\right) - 16, \quad (2)$$

$$a^* = 500\left(f\left(\frac{X}{X_n}\right) - f\left(\frac{Y}{Y_n}\right)\right), \quad (3)$$

$$b^* = 200\left(f\left(\frac{Y}{Y_n}\right) - f\left(\frac{Z}{Z_n}\right)\right), \quad (4)$$

$$f(t) = \begin{cases} t^{\frac{1}{3}}, & \text{for } t > \left(\frac{24}{116}\right)^3 \\ \frac{841}{108}t + \frac{16}{116}, & \text{for } t \leq \left(\frac{24}{116}\right)^3 \end{cases} \quad (5)$$

なお、式(2)、(3)、(4)において $X_n = 95.05$ 、 $Y_n = 100$ 、 $Z_n = 108.91$ とする。これは色温度6,500Kの昼光色を想定した標準光源D65のXYZ値である。

### 3. 2 Kinectの仕様

Kinectで取得出来るカラー情報を図1に、デプスデータを図2に示す。なおデプスデータで取得出来る値は0から10000になるため青いほど0、赤いほど10000のデータになるように擬似カラーに変換し、画像で見やすくしている。

図2の青の濃い部分がデータの取得の出来ない領域である。Kinectでは奥行の取得に赤外線を利用していることから、主に鏡や金属などの反射しやすい物体、エッジ部分、Kinectから極端に近い、あるいは遠い部分のデータが取得出来ないことがある。

Kinectはカラー情報用のRGBカメラとデプスデータ用の赤外線カメラは物理的に違う位置に設置されているため、単純に重ね合わせることは出来ない。しかし、SDK上にキャリブ

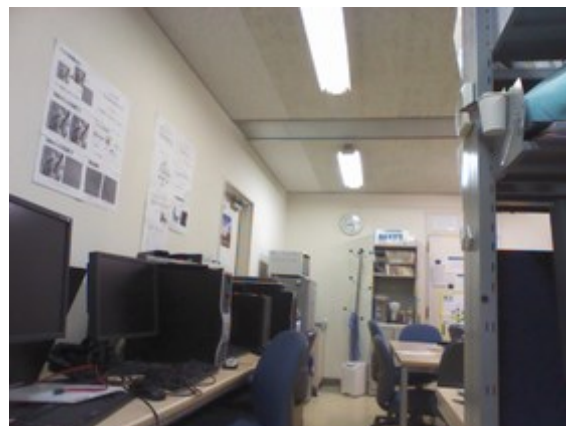


図 1. カラー画像

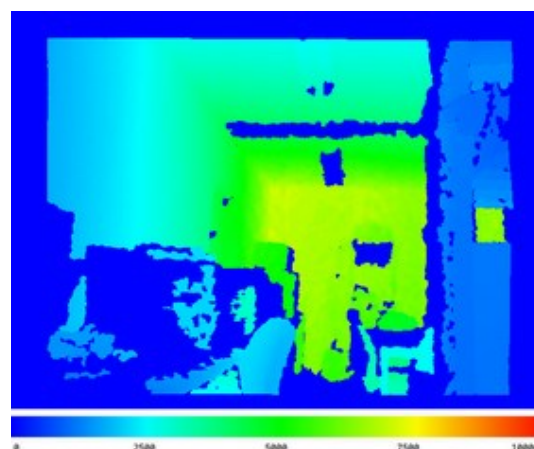


図 2. デプスデータ

レーション情報を元にした位置合わせを行うAPIが用意されている。このAPIは十分な精度ではあるが、位置合わせ影響により上下左右のデプスデータが欠落する。Kinectでは取得出来ないデータは全て0として処理されている。

### 3. 3 デプスデータの補間

デプスデータの補間は、欠損部分周辺にある有用なデプスデータから推測する。推測方法は同じ位置にあるカラー情報と近似度が高い部分のデプスデータの比重を多くする。本研究ではバイラテラルフィルタの一部を改変した、Hole filterを作成する。Hole filterの式を式(6)～(9)に示す。

$$g(i, j) = \frac{\sum_{n=-w}^w \sum_{m=-w}^w f_d(i+m, j+n) * P * N}{\sum_{n=-w}^w \sum_{m=-w}^w P * N}, \quad (6)$$

$$P = \exp\left(-\frac{m^2 + n^2}{2\sigma_1^2}\right), \quad (7)$$

$$N = \exp\left(-\frac{\sqrt{\Delta E}}{2\sigma_2}\right), \quad (8)$$

$$\begin{aligned} \Delta E = & (f_L^*(i, j) - f_L^*(i + m, j + n))^2 \\ & + (f_a^*(i, j) \\ & - f_a^*(i + m, j + n))^2 \\ & + (f_b^*(i, j) \\ & - f_b^*(i + m, j + n))^2. \end{aligned} \quad (9)$$

式(6)の $f_a(i + m, j + n)$ は参照画素のデプスデータの画素の値である。 $(2W + 1) * (2W + 1)$ のフィルタのサイズであり、推測に利用する領域の範囲である。式(7)の $P$ は、注目画素から参照画素の距離に対して重み付けをしている。注目画素から近い場合は重く、遠い場合は軽くする。 $\sigma_1$ は標準偏差である。式(8)の $N$ は、注目画素と参照画素の色差を計算し重み付けをしている。注目画素と色差が近い場合は重く、遠い場合は軽くする。式(9)の $\Delta E$ は $L^*a^*b^*$ のユークリッド距離で求められた色差を計算している。 $\sigma_2$ は標準偏差である。最終的に $g(i, j)$ を出力として得る。

基本的にデプスデータの欠損はある程度まともな発生し、1画素だけ抜けることはまれである。そのため、上記の式をそのまま適用した場合、値があるデプスデータが注目画素の周辺に数個しかない状況があり、品質が大きく低下する。そこで、注目画素の周辺に値があるデプスデータが数個しか無い場合は、先に周辺のデータを補間してから処理をするようにしている。

#### 4 結果

二種類の状況で実験を行った。一つ目の状況の結果を図3～図6に示す。図3は入力されたカラー情報、図4は入力されたデプスデータ、図5は補間後のデプスデータである。図4に見られた、デプスデータの欠損部分(画像では濃い青の部分)が、図5では適切に補間されていることが分かる。図3のカラー情報と図5の補間後のデプスデータをアルファブレンドした結果を図6に示す。同様に別の画像で実験した結果が、図7～図10に示す。

#### 5 まとめ

本研究では欠落したデプスデータを補間する新しい手法を提案した。カラー情報の注目画

素の周辺は似たような色を持っている性質を利用し、デプスデータの補間がなされた。今後の課題として、補完効果の定量的な評価及び特定の状況下での品質向上が上げられる。

#### 参考文献

- [1] A. Bapat, A. Ravi, and S. Raman, "An iterative, non-local approach for restoring depth maps in RGB-D images," *2015 Twenty First Natl. Conf. Commun.*, pp. 1–6, Feb. 2015.
- [2] M. Schmeing and J. Xiaoqi, "Color Segmentation Based Depth Image Filtering," *Adv. Depth Images Anal. Appl.*, vol. 7854, pp. 68–77, Nov. 2012.
- [3] K. R. Vijayanagar, M. Loghman, and J. Kim, "Real-Time Refinement of Kinect Depth Maps using Multi-Resolution Anisotropic Diffusion," *Mob. Networks Appl.*, vol. 19, no. 3, pp. 414–425, Sept. 2013.

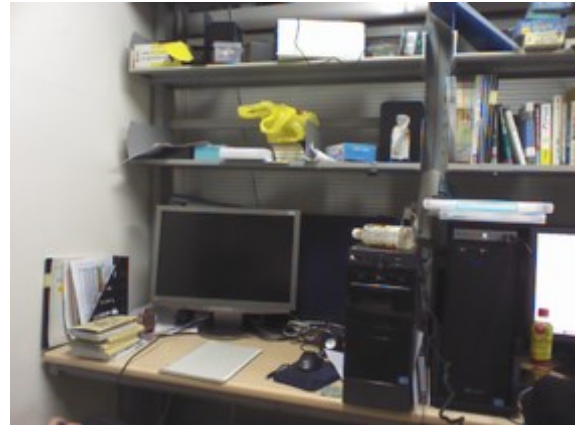


図 3. カラー情報

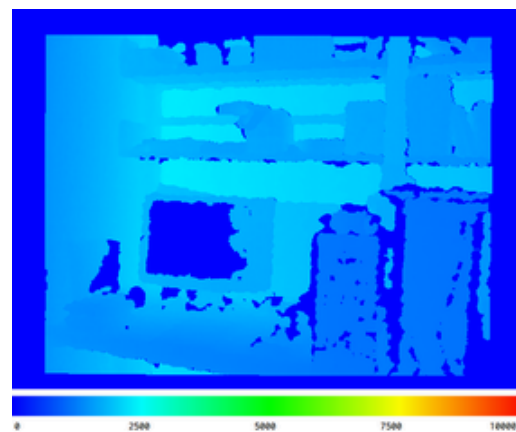


図 4. 補間前のデプスデータ

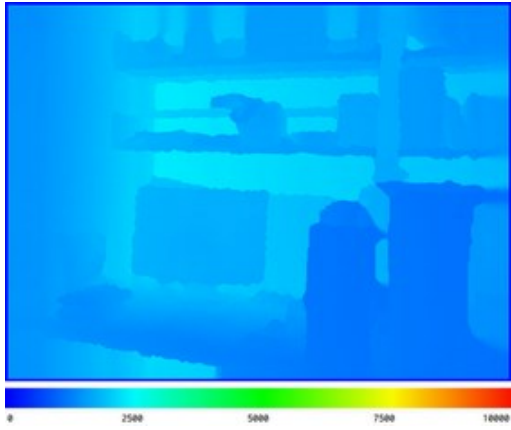


図 5. 補間後のデプスデータ

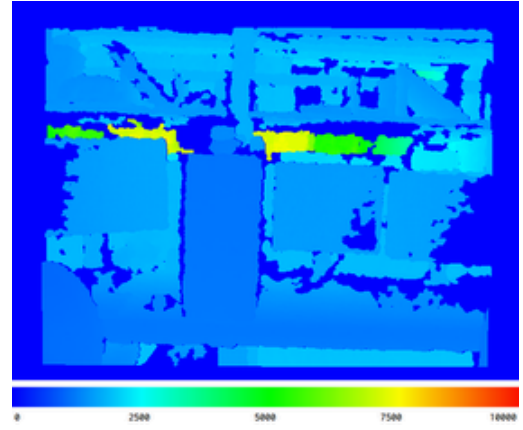


図 8. 補間前のデプスデータ



図 6. カラー情報とデプスデータの合成

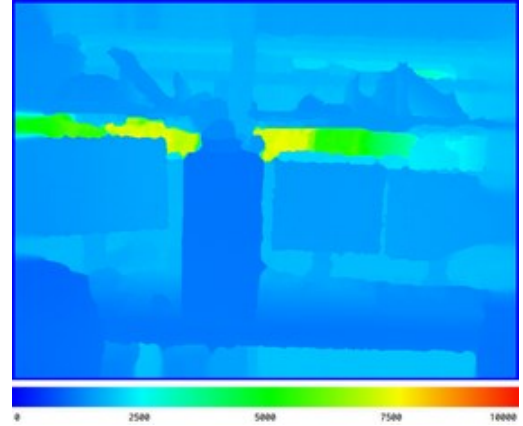


図 9. 補間後のデプスデータ



図 7. カラー情報

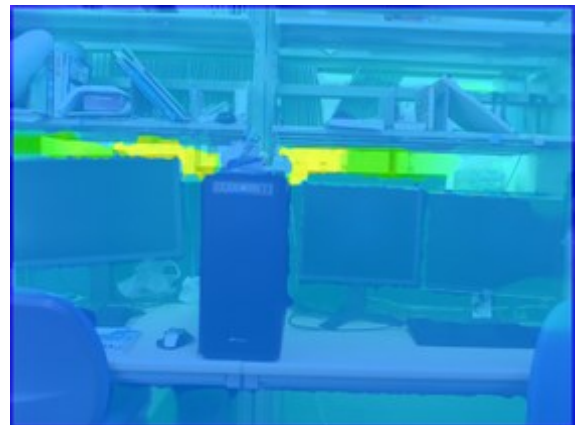


図 10. カラー情報とデプスデータの合成