

研究速報

Multi-Pass Dynamic Programming による  
光線空間補間

福嶋 慶繁<sup>†a)</sup> 圓道 知博<sup>†</sup>(正員)  
藤井 俊彰<sup>†</sup>(正員) 谷本 正幸<sup>†</sup>(正員:フェロー)

Ray-Space Interpolation Using Multi-Pass Dynamic Programming

Norishige FUKUSHIMA<sup>†a)</sup>, Nonmember, Tomohiro YENDO<sup>†</sup>,  
Toshiaki FUJII<sup>†</sup>, Members, and  
Masayuki TANIMOTO<sup>†</sup>, Fellow

<sup>†</sup> 名古屋大学大学院工学研究科, 名古屋市

Graduate School of Engineering, Nagoya University, Furo-cho, Chikusa-ku, Nagoya-shi, 464-8603 Japan

a) E-mail: fukushima@tanimoto.nuee.nagoya-u.ac.jp

あらまし 本論文では, 多視点画像から視点依存視差推定し, その最適化を行う. 従来では困難であった, ノイズが多い領域や, テクスチャの少ない領域で推定可能なように, Dynamic Programming を拡張した, Multi-Pass Dynamic Programming を提案する.

キーワード 光線空間法, Image-Based Rendering, Multi-Pass Dynamic Programming, 視点依存視差推定

1. まえがき

Image-Based Rendering (IBR) とは, 撮影画像を直接使い, 様々な映像を描画する手法である. IBR は, ポリゴンを用いるコンピュータグラフィックスによる手法に比べ, 容易に写実的な映像生成が可能である.

本論文は, IBR である光線空間法 [1] を使い, 自由視点画像生成を行うことを目的としている. 光線空間法による自由視点画像生成を行うためには, Plenoptic Sampling [2] を満たす高密度な光線情報が必要となるが, この情報の取得は困難である. たとえ, 大規模カメラアレー (100 台程度, 図 1 参照) を用いても十分な光線取得は不可能である. つまり, 自由視点画像を出力するためには, 不足光線を補間することが必須となる.

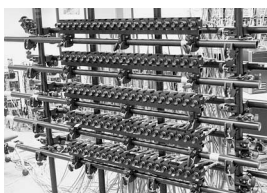


図 1 カメラアレー  
Fig. 1 Camera array.

この際, 三次元シーンの完全復元を目指すとは負荷が高く, また, その精度には疑問が残る. 本論文では, 三次元モデルをかえさず, 直接的に仮想視点の生成を目指す. これは, 視点依存視差推定を正確かつ高速に行うことがかぎとなる [3]. 本論文は, その視差推定を, 被写体の奥行が滑らかに変化するという拘束条件のもとで, 推定を高速に行う手法を提案する.

2. 自由視点画像合成と視点依存視差画像

2.1 視点依存視差画推定

直接的に自由視点画像を生成するには, 生成視点を通過する光線群とその光源奥行に対応する画像中の点が分かればよい. 設置した二次元配置カメラアレーを真上から見たものである, 図 2 中の  $Ray_A$  を補間するとき, 仮想視点  $V_{x,y,z}$  を通過し,  $(u, v)$  の方向をもつ補間対象の光線は, カメラアレー上に張られる光線空間の参照平面座標  $(X, Y)$  を以下の式で通過する.

$$X = x + uz, Y = y + vz$$

実カメラがその座標中整数値にあるとき, 通過光線とその近傍カメラ LU, LD, RU, RD の座標を, 内分比  $\alpha, \beta$ ,  $(0 \leq \alpha, \beta < 1)$  で表すと,

$$LU_{u,v}: ([X], [Y]), RU_{u,v}: ([X+1], [Y])$$

$$LD_{u,v}: ([X], [Y+1]), RD_{u,v}: ([X+1], [Y+1])$$

$$V_{x,y,z}(u, v): ([X] + \alpha, [Y] + \beta)$$

となる (図 3). 次に, 光源奥行と近傍カメラの画像対応をつけるために, 図 4 のように奥行を視差  $d$  により関係付ける. 視差は  $d = \frac{L}{z}$  で与えられ (L はカメラ間隔), 内分関係を用いれば, 各画像対応は以下となる. (ただし,  $\bar{\alpha} = 1.0 - \alpha, \bar{\beta} = 1.0 - \beta$ )

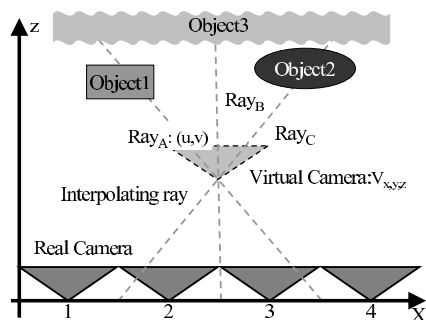


図 2 視点依存視差推定  
Fig. 2 View dependent disparity estimation.

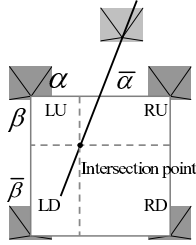


図 3 内 分  
Fig. 3 Intersection.

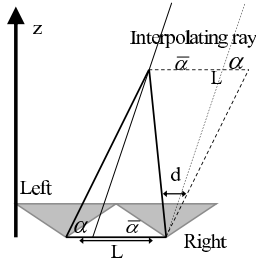


図 4 視 差  
Fig. 4 Disparity.

$$LU_{u,v}(d):(u+\alpha d, v+\beta d), RU_{u,v}(d):(u-\bar{\alpha}d, v+\beta d)$$

$$LD_{u,v}(d):(u+\alpha d, v-\bar{\beta}d), RD_{u,v}(d):(u-\bar{\alpha}d, v-\bar{\beta}d)$$

視差  $d$  を与えた時、近傍カメラ中の対応光線群が、

$$LU_{u,v}(d) \simeq RU_{u,v}(d) \simeq LD_{u,v}(d) \simeq RD_{u,v}(d)$$

となる Lambert 反射特性を満たせば、参照平面上での距離に応じて線形補間で光線補間ができる。この視差決定の詳細は 2.2 以降に説明する。

$$V_{x,y,z}(u, v) = \bar{\alpha}\beta LU + \alpha\bar{\beta}RU + \bar{\alpha}\beta LD + \alpha\bar{\beta}RD$$

上記の、1) 生成視点決定、2) 必要近傍光線群の計算、3) 対応点からの生成視点の視差計算、という流れで推定される仮想視点中のシーン構造は、生成視点ごとに結果が異なるため、視点依存視差推定と呼ぶ。これらの処理を生成視点中の光線  $Ray_{B,C}$  など全光線で行うと、視点依存視差画像  $VDD(u, v)$  (View dependent disparity map) が生成される。このようにして、視点ごとに視差推定を伴い、直接的に画像生成を行う。

## 2.2 SAD による素朴な手法の問題点

視点依存視差画像を推定する時、Lambert 反射を満たす視差を、輝度の SAD (Sum of Absolute Difference) による一致度  $E_D$  で毎 pixel  $(x, y)$  において測る。補間光線の近傍カメラを通る光線を

$Ray_{p,q} \in Ray = \{LU, LD, RU, RD\}$  ( $p \neq q$ ) とし全組合せの和をとる。

$$E_D^{x,y}(d) = \sum_{Ray_{p,q} \in Ray} |Ray_p^{x,y}(d) - Ray_q^{x,y}(d)|$$

$$VDD(x, y) = \operatorname{argmin}_d E_D^{x,y}(d)$$

しかし、この条件だけでは、多くの誤対応がおきる(推定結果の図 8 左上を参照)。これは、テクスチャの薄い領域など、輝度の条件だけでは対応しきれないためである。そこで、本論文は、滑らかな視差画像が生成されるように、輝度に加え、周囲の奥行状態との相違を拘束条件として視差を導出する。つまり、画像全体の一致度に加え、その生成視差画像の微分値も同時にエネルギー  $E_I$  として最小化する。 $\lambda$  は両コストのバランス定数である。

$$E_I = \sum_x \sum_y (E_D^{x,y} + \lambda \|\nabla VDD(x, y)\|)$$

## 2.3 Dynamic Programming による視差推定

このエネルギー  $E_I$  のモデルは Morkov Random Field [5] と呼ばれ、その最小化は NP 問題である。しかし、微分を 1 方向に限定すると二次元画像中の最適化となり、Dynamic Programming (DP) により効率的に解ける [4]。本章中の DP も [4] と同様に、左画素の視差と自分との差分  $E_S$  と、自身の画素の一致度  $E_D$  の和をエネルギー関数  $E$  として総和を最小化する。しかし文献 [4] はステレオ画像上の視差推定であり、本手法は、視点依存視差画像の視差推定である。そこで、画像ライン  $y$  ごとのエネルギー関数  $E_{line}^y$  を最小化する視差  $d(x)$  の組合せを視点依存視差画像とする。 $E_S$  中の  $\gamma$  は大きな差がある際の打ち切り値である。

$$E^{x,y}(d(x)) = E_D^{x,y}(d(x)) + \lambda E_S^{x,y}(d(x))$$

$$E_S = \min(|d(x-1) - d(x)|, \gamma)$$

$$E_{line}^y = \min_x (E^{x,y}(d(x)))$$

$$VDD^y(x) = \operatorname{argmin}_{d(x)} E_{line}^y$$

$E_{line}$  の最小値は、通常は視差の組合せを総当りで求めるしかないが、DP を用いれば、効率的に解ける。まず、画像中  $x$  列目までのエネルギー関数の総和を各視差  $d$  ごとに表すと  $F_x(d) = \min \sum_0^x (E^{x,y})$  であり、 $x$  が終端なら最小化対象の  $E_{line}$  と同じとなる。最適性の原理から、全体が最適であるとき、その部分も最適

でなければならないため、 $F_x$  は直前の最適解の  $F_{x-1}$  とその解を選択する直前の奥行  $d_{pre}$  との漸化式で表される。x 列目の視差を  $d = d(x)$  とすると、

$$F_x(d) = \min(F_{x-1}(d_{pre}) + E^{x,y}(d))$$

求めるものは、この  $F$  を最小とする視差の組合せであるので、左画素の奥行  $d_{pre}$  をすべての奥行状態において  $x = 1, 2, \dots$  と順次計算し  $pd_x(d)$  へ格納する。そして、終端まで演算し、終端で最小値をもつ視差から、 $pd(x)$  をトラックバックすることで視差  $VDD$  の組合せを選択する。

$$pd_x(d) = \underset{d_{pre}}{\operatorname{argmin}}(F_{x-1}(d_{pre}) + E^{x,y}(d))$$

図 5 は、この漸化式を模式化したもので各 Node は  $E_D$  のコストをもち、異なる視差状態へ遷移をする edge には相応のコスト  $E_S$  を与える。また表 1 は各々の視差状態における、累積最小コスト  $F$  と直前の最適視差を記憶したものである。上記の問題はこのネットワークの最小経路を導出する問題と等価である。まず、初期値  $F_1$  に一致度  $E_D^1$  を各奥行ごとに代入する。次に  $F_2$  のすべての奥行状態に対し、 $F_1$  と edge の和が最小となる直前の pass を決定・記憶し、自身の一致度  $E_D^2$  を足し合わせ  $F_2$  とする。この処理を終端まで行い、最小値をもつノードを見つけ、それらの pass がつながる経路を、最小コストを与える視差の組とす

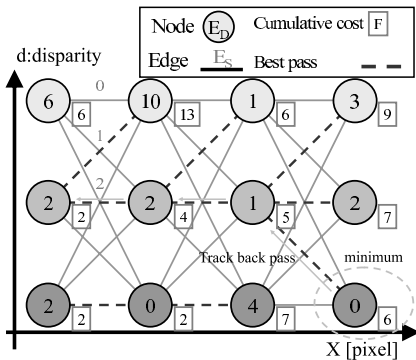


図 5 Dynamic Programming のネットワーク  
Fig. 5 The network of Dynamic Programming.

表 1  $F_x$  とその直前の最小視差  
Table 1  $F_x$  and previous minimam disparity.

	pd: $F_1(d)$	pd: $F_2(d)$	pd: $F_3(d)$	pd: $F_4(d)$
$d = 0$	N/A : 6	1 : 13	1 : 6	1 : 9
$d = 1$	N/A : 2	1 : 4	1 : 5	1 : 7
$d = 2$	N/A : 2	0 : 2	0 : 7	1 : 6

る。図 8 の中上は推定視差画像である。DP は二次元画像の最適化にとどまるため、行間が飛び飛びであることが分かる。次節はこれを解決する手法を述べる。

### 2.4 Multi-Pass Dynamic Programming

行間の不連続性の問題を解決するためには、エネルギー関数  $E_I$  において上下左右の微分値を考慮する必要があるが、DP では二次元画像の最適化が限界である。そのため、本論文では、他方向からの視差の滑らかさを伝搬させた一致度  $E_D$  を新たに生成し DP を適応する。ここでは DP を順方向、逆方向、縦方向の 3 方向から行い、エネルギー最小化する。

さて、先の DP で求めた各々の行  $y$  の部分最小コスト  $F_x^y(d)$  を一致度  $E_D^{x,y}(d)$  とし、各々独立に最小値を選択すると、たとえトラックバック経路を記憶しなくとも、ほぼ DP と同様な準最適解の視差が得られる。

$$VDD(x, y) = \underset{d}{\operatorname{argmin}} F_x^y(d(x))$$

これは、DP のコスト算出時にイレギュラーな経路に大きなコストが計上されるためである。図 6 は準最適解、最適解の視差を示したものである。つまり、1 度目の DP の結果である  $F$  に対し、別方向からの 2 度目の DP を適応すると、行間の不連続性を解決することが可能である。しかし  $F$  は、 $x$  が進行すると累積的に増加する(図 7)。対して、SAD の一致度はほぼ一定値である。そのため、 $F$  をコストとして用いるには正規化の必要がある。ここでは、単純に各  $x$  ごとに正規化するのではなく、DP を順方向に対し、逆向きの進行方向(つまり  $x, x + 1$  の漸化式を逆方向から解く)から適応したものを足し合わせることで正規化する。図 7 は  $y$  行目の順方向のコスト  $F_{x+}^y$ 、逆方向のコスト  $F_{x-}^y$ 、その和  $F_{x_{total}}^y$  を示したものである。

$$E_D^{x,y}(d) = F_{x_{total}}^y(d) = F_{x+}^y(d) + F_{x-}^y(d)$$

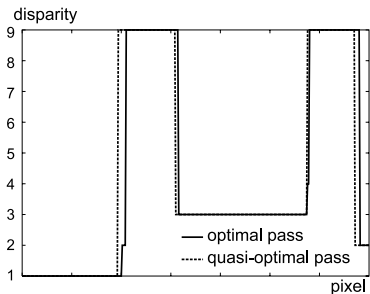


図 6 最適解と準最適解の視差  
Fig. 6 Optimal and quasi-optimal disparity.

このように、順方向と逆方向からの DP 結果の和をとると、正規化すると同時に左右の影響を考慮した一致度が生成可能となる。この正規化したコスト  $F_{x_{total}}^y$  に対して、最終的に縦方向から DP を適応する。

縦方向の漸化式を、 $x$  行目の  $y$  列目までの縦の部分エネルギー  $G_x^y(d(y))$ 、直前の視差を  $d_{pre} = d_x(y-1)$ 、現在の視差を  $d = d_x(y)$  を用いると

$$E^{x,y}(d_x(y)) = F_{x_{total}}^y + \lambda E_S^{x,y}(d_x(y))$$

$$E_S = \min(|d_x(y-1) - d_x(y)|, \gamma)$$

$$E_{line}^x = \min \sum_y (E^{x,y}(d_x(y)))$$

$$VDD^x(y) = \operatorname{argmin}_{d_x(y)} E_{line}^y$$

$$G_y^x(d) = \min(G_{y-1}^x(d_{pre}) + E^{x,y}(d))$$

$$pd_y(d) = \operatorname{argmin}_{d_{pre}} (G_{y-1}^x(d_{pre}) + E^{x,y}(d))$$

となり、縦方向にたどることで最小のコストを与える

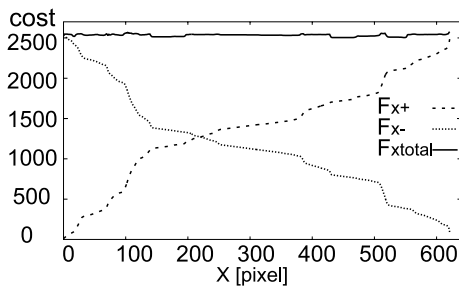


図 7 順方向、逆方向の和をとり DP コストを正規化  
Fig. 7 Normalization for DP cost.

奥行が求まる。このように多方向の DP を統合する、Multi-Pass DP により、行間の相違が少ない推定が可能となる。この三つの多方向への DP を本論文では Multi-Pass Dynamic Programming (MPDP) と呼ぶ。

### 3. 実験及び結果

SAD, DP, MPDP の各手法において計算時間と生成画像の精度の評価実験を行った。各パラメータは  $\gamma = 2, \lambda = 4$  とした。その際、CG により、PSNR の画質評価を行うと同時に、実写画像での確認を行った。入力画像の解像度はどちらも  $640 \times 480$  である。図 9 は CG によりレンダリングした画像群を用い、各手法により補間した画像を CG による生成画像を真値として PSNR 評価したものである。このとき、カメラ間隔を広げることで、最大視差を増やし、補間の頑強さ・正確さを測定した。グラフから、MPDP が最も精度良く合成可能なことが分かる。また、実写画像に用い

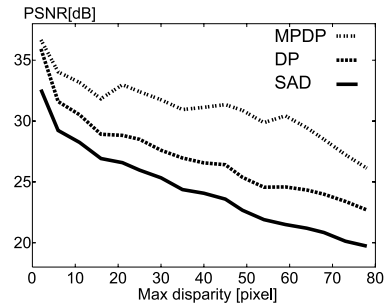


図 9 各手法の PSNR 評価  
Fig. 9 Evaluation of each method by PSNR.

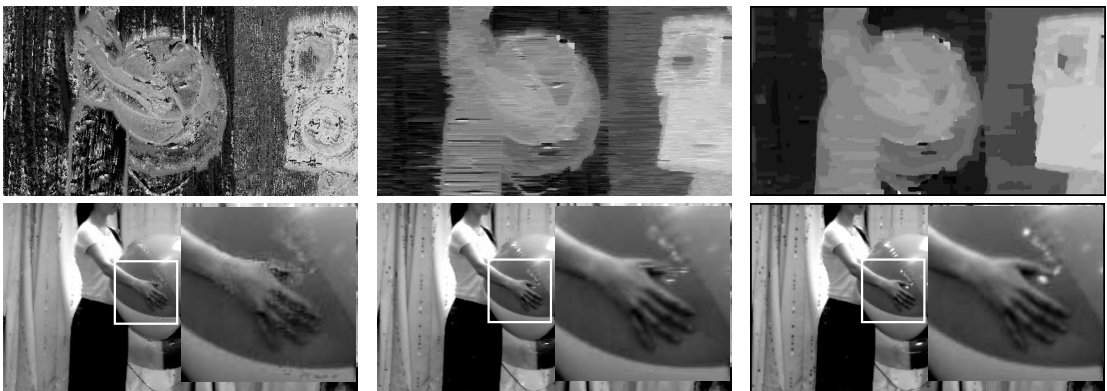


図 8 実写画像による自由視点画像と視点依存視差画像：左から SAD, DP, MPDP による手法：上) 視点依存視差画像：下) 自由視点画像  
Fig. 8 Free viewpoint image and view dependent disparity map (left to right) SAD, DP, MPDP.

たカメラアレーは  $20 \times 5$  (横:縦) 台であり, カメラ間隔は 5 cm, その最大視差は 20 pixel であった. 図 8 は各手法により生成された自由視点画像と, その視点依存視差画像である. MPDP による視差画像が最も滑らかであり, また生成画像のノイズも少ない.

これらの手法計算時間は最大視差 20 pixel のとき, SAD 210 ms, DP 525 ms, MPDP 971 ms となった. これらの計算をピラミッド構造で高速化計算すると [3], SAD 90 ms, DP 121 ms, MPDP 191 ms となり, 準ビデオレートに達することが可能であった.

#### 4. む す び

生成画像の画質向上のため, Dynamic Programming を様々な方向から適応する Multi-Pass Dynamic Programming により, 視点依存視差推定をより正確に行う手法を提案した. 結果, 191 ms で, より正確な自由視点映像が生成可能となった. 今後は, 物体の隠

れに対応することや, GPU による実装が挙げられる.

#### 文 献

- [1] T. Fujii, T. Kimoto, and M. Tanimoto, "Ray space coding for 3D visual communication," Proc. PCS'96, vol.2, pp.447-451, March 1996.
  - [2] J.-X. Chai, S.-C. Chan, H.-Y. Shum, and X. Tong, "Plenoptic sampling," Proc. SIGGRAPH'00, pp.307-318, 2000.
  - [3] N. Fukushima, T. Yendo, T. Fujii, and M. Tanimoto, "Real-time arbitrary view interpolation and rendering system using Ray-Space," Proc. SPIE, vol.6016, pp.250-261, Oct. 2005.
  - [4] Y. Ohta and T. Kanade, "Stereo by intra- and inter-scanline search using dynamic programming," IEEE Trans. Pattern Anal. Mach. Intell., vol.7, no.2, pp.139-154, 1985.
  - [5] S. Li, Markov Random Field Modeling in Computer Vision, Springer Verlag, 1995.
- (平成 18 年 11 月 30 日受付, 19 年 2 月 16 日再受付)