

ハードウェア指向ステレオマッチングアルゴリズムのアーキテクチャとそのFPGA実装

北海道大学 大学院情報科学研究科*, 慶應義塾大学 理工学部 電子工学科†

松山 健人*, 真田 祐樹*, 大畑 克樹†, 大垣 哲郎†, 大平 貴徳*, 築田 聡史*, 五十嵐 正樹*, 池辺 将之*, 浅井 哲也*, 本村 真人*, 黒田 忠広†

VDECデザイナーズフォーラム2013

研究背景

～「見る」ための3Dから「使う」ための3Dへ～

見る: 3Dテレビ, 3DS

処理: Kinect, Light Field Camera



Kinectでの深度推定例



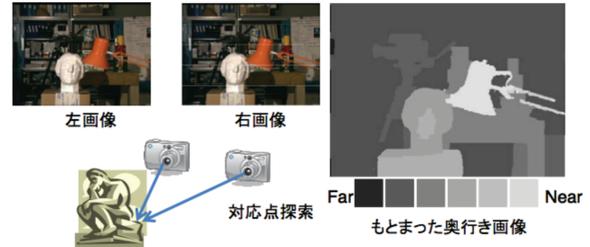
Light Field Camera

処理内容: 背景置換, AR, 物体認識補助, リフォーカス

→これらを実現する技術の一つが「ステレオマッチング」

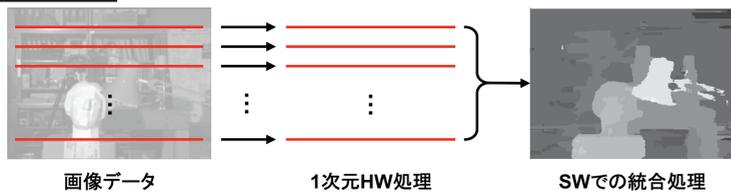
ステレオマッチングとは?

※ステレオ画像から深度を推定する技術



使用するハードウェア、アルゴリズムにより、精度、速度等は大きく変化
→傾向としては精度・速度が両立せず

研究目的



ステレオマッチングをHWとSWで分割して精度・速度を同時に得る
→ステレオマッチングではカメラを左右に並べるため、横方向の処理結果が重要
→HW部では以下のような高速化のための改良ができると考えられる

- ・SWで2次元処理を行い、HW処理を1次元化
- ・メモリを節約するためにコスト算出にグレースケールを使用
- ・多並列化による高速化の実現(1行読めば処理開始可能)

コスト補正部(Guided Filter)の1次元化



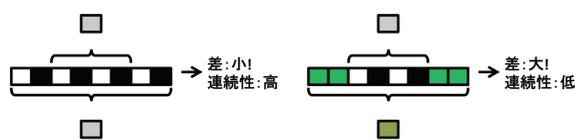
※精度は落ちる

- コスト補正部に
- ・1次元処理を採用→100行分以上のラインバッファを削減
 - ・グレースケールを採用→さらなるメモリの節約
- 並列型ステレオマッチングが実現可能に

フィルタ半径設定による精度向上

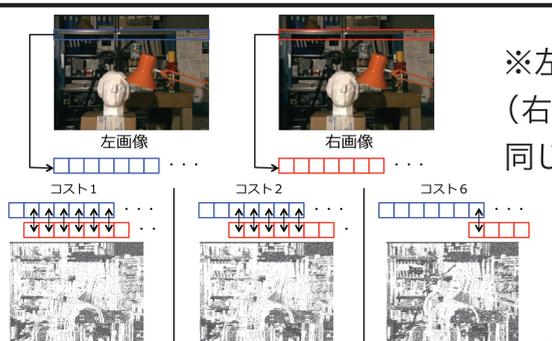
1次元処理ならば行ごとにフィルタ半径を設定可能

→1次元化の際、深度マップの精度を犠牲にしたのをリカバー



1. 入力画像の一部が横方向に同じテクスチャを持っているか否かでフィルタサイズ設定
2. 半径の違う平滑化フィルタをかけた結果の差分からテクスチャの連続性が判明
3. 1行毎に連続性の傾向を出し、それに応じた半径を設定

並列型ステレオマッチングアーキテクチャ



※左視点基準の場合(右視点基準でも同じことを行う)

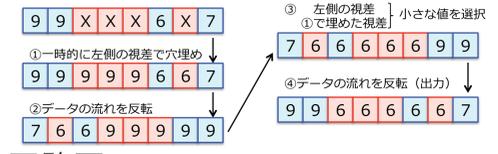
深度別のコスト算出を並行に処理→遅延器を用意し実現
コストの値が小さいほど正確な視差d→LTA(Loser Takes All)で深度決定

オクルージョン(左右の整合性)チェック

左右の視差画像の不一致箇所(エラー)を検出



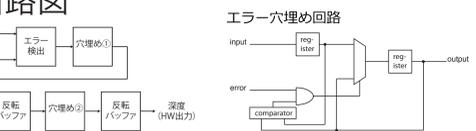
欠損の穴埋め(右重視と左重視を考慮しながら)



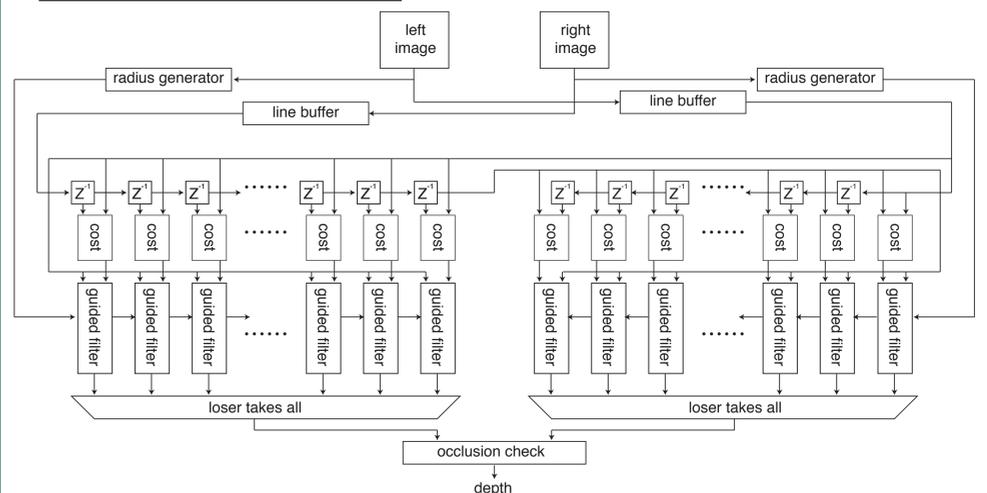
回路図

エラーの大きさ・頻度は不定

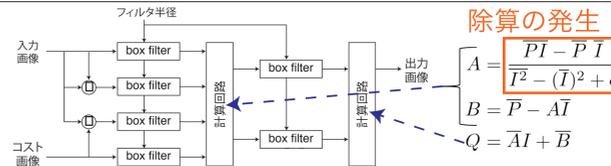
→ハードウェア化には工夫が必要



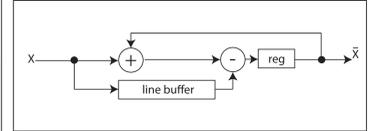
全体回路図(まとめ)



Guided Filter



Box Filter(Line Buffer)



実装条件

FPGA: Startix II(ALTERA)
ボード: PowerMedusa MU200SX II(MMS)
論理合成ツール: Quartus II(ALTERA)



- ・実装における変更点
- 1. 視差の精度: 16深度→8深度(FPGAのリソース不足)
→原因: Guided Filter 内の除算器
- 2. 入力画像の縮小(視差の精度に合わせて)

実装結果

入力	出力 (8深度)	参考 (16深度)	SWによるエラー修正をした最終結果

上下方向1次元修正後 メディアンフィルタ

入力画像 192 × 144 (24bit RGB)
出力画像 192 × 144(3bit)
FPGA CLK 20MHz
FPS 720、処理時間 14msec (16深度はSWシミュレーション結果)

FPGAでの実装に成功