

輝度変化にロバストな高速オプティカルフローアルゴリズムのFPGA実装

著者	深山 正幸, 田村 賢一, 松田 吉雄
著者別表示	Miyama Masayuki, Tamura Kenichi, Matsuda Yoshio
雑誌名	画像電子学会誌
巻	40
号	1
ページ	191-199
発行年	2011-01
URL	http://doi.org/10.24517/00049538

doi: 10.11371/iieej.40.191



輝度変化にロバストな高速オプティカルフローアルゴリズムのFPGA実装

深山正幸[†](正会員) 田村賢一[†] 松田吉雄[†][†] 金沢大学大学院自然科学研究科

あらまし オプティカルフローは動画における連続する2フレーム間の画素毎の動きベクトルである。本論文は輝度変化にロバストで高速なオプティカルフロー推定アルゴリズムを提案する。輝度変化量を差分平均により求め、時間輝度勾配を補正し、ロバスト化を図る。初期値設定法、階層処理法、ウィンドウサイズの最適化により処理の高精度化と高速化を図る。シミュレーションの結果、輝度変化を加えた場合の平均角度誤差は18.65°から4.29°に大きく改善され、ロバスト性を確認した。輝度変化を加えず精度を同程度とした場合のCPU実行時間は従来の15%となり、高速性を確認した。さらに本アルゴリズムに基づく専用回路構成を提案し、FPGA実装する。提案回路は従来回路より輝度変化にロバストな上に、同程度の精度とスループット性能を仮定したとき、動作周波数が20%、面積が45%未満、電力が9%未満に削減される。本回路は実時間画像認識や画像再構成に応用可能である。

キーワード：オプティカルフロー、輝度変化、ロバスト性、実時間、FPGA

Summary An optical flow is defined as a motion vector for each pixel between successive two images in a video. This paper proposes a fast optical flow algorithm with robustness for luminance change. The algorithm includes the luminance change estimation, initial vector decision, fast hierarchical processing, and window size optimization. These methods result in the robustness for the luminance change, accurate flow estimation, and low computational costs. Simulation results show the flow accuracy improves from 18.65° to 4.29°, when giving luminance change. The CPU execution time reduces to 15% compared to the conventional with the same accuracy. In addition, we propose a dedicated architecture based on this algorithm and implement it on an FPGA. Experimental results show the new architecture decreases the operating frequency to 20%, the circuit area to 45%, and the power consumption to 9%. This circuit is applicable to real-time image recognition and image reconstruction.

Key words: optical flow, luminance change, robustness, real-time, FPGA

1. はじめに

オプティカルフローは動画における連続する2フレーム間の画素毎の動きベクトルである。オプティカルフローを用いることにより、フレーム内の物体の動きやカメラの動きを検出することが可能となる。オプティカルフローは車載、ロボット、監視などの応用分野における様々な物体を対象とする画像認識や、超解像やフレームレートアップコンバージョンなどの画像処理に応用で

きる^{1),2)}。

オプティカルフローを求める代表的なアルゴリズムとして、Horn and Schunk(HS)やLucas and Kanade(LK)の方法が知られている^{3),4)}。これらはいずれも「同一画素の輝度値はフレーム間で変化しない」という輝度保存性を仮定している。特別な配慮のない通常の動画では日射量の変化などの撮影環境の変化や蛍光灯のちらつきなどのノイズにより輝度保存性が容易に破綻する。フレーム間で輝度保存性が成立しないとフローの推定精度が大きく劣化する。そこで輝度変化にロバストなオプティカルフロー推定アルゴリズムが求められている。

一方、オプティカルフローは画素単位の処理であるため演算量が非常に大きい。HSアルゴリズムは「フロー

“FPGA Implementation of Fast Optical Flow Algorithm with Robustness for Luminance Change” by Masayuki MIYAMA, (Member), Kenichi TAMURA and Yoshio MATSUDA, (Graduate School of Natural Science and Technology, Kanazawa University).

の変化は空間的に滑らかである」という仮定に基づいており、任意の画素と隣接する画素のフローの差分が小さくなるまで推定を何回も繰り返す必要がある¹⁾。これに対して、LK アルゴリズムは画素毎に独立にフローを推定し、収束までの繰り返し回数が比較的少ない⁵⁾。一般に画像認識は離散した少数の特徴点のみをフロー推定対象とするため、特徴点のフローだけを推定できる LK アルゴリズムが周囲のフローの推定を必要とする HS アルゴリズムより演算量の面で有利である⁶⁾。この場合でもフレームの解像度が高くなれば特徴点も多くなり演算量が增大する。また、超解像やフレームレートアップコンバージョンに適用するには基本的にフレーム内の全画素のフローが必要である。多数の画素に対する汎用プロセッサを用いたソフトウェアによるフロー推定の実時間処理は演算量が大いいため困難である。VGA 30 fps 程度の解像度を持つ動画像に対して全画素のフロー推定を実時間処理するには高速アルゴリズムに基づく専用ハードウェアが必要である。

そこで本研究では、LK アルゴリズムに基づく輝度変化にロバストで高速なオプティカルフロー推定アルゴリズムを提案する。輝度変化量を差分平均により求め、時間輝度勾配を補正し、ロバスト化を図る。初期値設定法、階層処理法、ウィンドウサイズの最適化により処理の高精度化と高速化を図る。シミュレーションの結果、輝度変化を加えた場合の平均角度誤差は 18.65° から 4.29° に大きく改善され、提案アルゴリズムのロバスト性を確認した。輝度変化を加えず精度を同程度とした場合の CPU 実行時間は従来の 15% となり、高速性を確認した。差分平均による輝度変化への対応と初期値設定と階層処理は動画像符号化の動き検出に採用されている方法である。今回これら三つの方法の組み合わせとウィンドウサイズの最適化を LK アルゴリズムに基づくオプティカルフロー推定に新たに適用し、輝度変化へのロバスト性と低演算量と高いフロー精度を同時に達成した。さらに本アルゴリズムに基づく専用回路構成を提案し、FPGA 実装する。提案回路は従来回路より輝度変化にロバストな上に、同程度の精度とスループット性能を仮定したとき、動作周波数が 20% に削減される。このとき同一の半導体技術を仮定すると、回路面積が 45% 未満、電力が 9% 未満に削減される。

次節以降の本論文の構成は以下のとおりである。第 2 節では LK アルゴリズムを拡張した PLK アルゴリズムについて説明する。第 3 節では輝度変化にロバストで高速なアルゴリズムを提案し、シミュレーション結果によ

り定量的効果を示す。第 4 節では本アルゴリズムに基づく専用回路構成を提案する。第 5 節では FPGA 実装結果により効果を定量化し、第 6 節で本論文をまとめる。

2. PLK アルゴリズム

Pyramidal Lucas and Kanade (PLK) アルゴリズムは LK アルゴリズム⁴⁾に増分推定と階層化を導入し、精度改善と大きな動きへの対応を図っている⁷⁾。PLK アルゴリズムにおいて、オプティカルフロー $\mathbf{v} = [v_x \ v_y]^T$ は式 (1) の誤差関数 E を最小化する動きベクトルとして定義される。

$$E = \sum_{(x,y) \in \Omega} (I(x,y) - J(x+v_x, y+v_y))^2 \quad (1)$$

$I(x,y)$ と $J(x,y)$ はそれぞれ連続する現フレームと次フレームの座標 (x,y) における輝度値であり、 Ω はフロー推定対象の画素を中心とする小さな正方形の画像 (ウィンドウ) を表す。ここでウィンドウ辺の画素数をウィンドウサイズ W と定義する。 I フレーム、 J フレームでのウィンドウをそれぞれウィンドウ A 、ウィンドウ B と呼ぶ。式 (1) は「同一画素の輝度値がフレーム間で変化しない」という輝度保存性と、「ウィンドウ内の全画素の動きが同一である」という局所的な動きの同一性の二つの仮定に基づいている。輝度保存性は以下の式 (2) で表される。

$$I(x,y) - J(x+v_x, y+v_y) = 0 \quad (2)$$

式 (1) の右辺第 2 項をテイラー展開し、1 次までの項で線形近似する。そして最小二乗法により式 (1) の E を最小化する \mathbf{v} を求めることで式 (3) が導かれる。

$$\begin{aligned} \mathbf{v} &= \mathbf{G}^{-1} \mathbf{b} \\ \mathbf{G} &= \sum_{(x,y) \in \Omega} \begin{bmatrix} I_x^2(x,y) & I_x(x,y)I_y(x,y) \\ I_x(x,y)I_y(x,y) & I_y^2(x,y) \end{bmatrix} \\ \mathbf{b} &= \sum_{(x,y) \in \Omega} \begin{bmatrix} I_x(x,y)I_t(x,y) \\ I_y(x,y)I_t(x,y) \end{bmatrix} \end{aligned} \quad (3)$$

ここで、 I_x, I_y, I_t はそれぞれ x 軸 (画像空間横軸)、 y 軸 (画像空間縦軸)、 t 軸 (時間軸) 方向の座標 (x,y) における輝度勾配であり、 \mathbf{G} と \mathbf{b} をそれぞれ空間輝度勾配行列、ミスマッチベクトルと呼ぶ。空間輝度勾配 I_x, I_y と時間輝度勾配 I_t は実際には隣接画素間の輝度値の差分により求められる。

図 1 に PLK アルゴリズムのフローチャートを示す。最初にフロー推定対象画素を中心としてガウシアンフィルタと 2:1 サブサンプリングを繰り返し適用して原画像から階層画像を作成する。そして注目階層におけるウィ

1) HS アルゴリズムの漸化式に SOR 法を適用して解の収束を高速化する研究が行われている。

ンドウ A の輝度値から I_x, I_y が、ウィンドウ A とウィンドウ B の輝度値から I_t が計算される。次に、式 (3) より G と b が計算され、最後に v が計算される。前回までに得られたフローに今回推定された v を増分として加算し、次回のためにウィンドウ B を v だけ移動する。これらの処理はレベル L からレベル 1(原画像) まで階層毎に K 回繰り返される。なお、階層を 1 つ降りる際にはフローを 2 倍する。

これまでにいくつかのオプティカルフロー推定専用プロセッサが提案されている⁸⁾と⁹⁾はLK アルゴリズムに基づくオプティカルフロープロセッサを提案し、FPGA 実装しているが、いずれも階層処理を行っていないため大きな動きに対応できず、輝度変化への対応も行っていない。本研究の基になった¹⁰⁾はPLK アルゴリズムに基づき階層処理を導入しているため大きな動きに対応できる。しかし輝度変化への対応が行われておらず、フロー推定精度向上のためにウィンドウサイズ W を大きくとっているため動作周波数と回路規模が大きい。

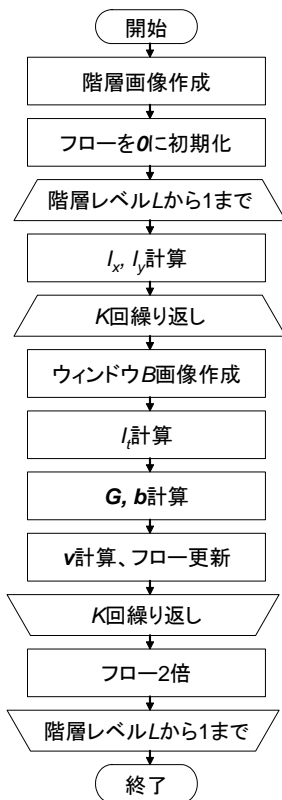


図1 PLK アルゴリズムのフローチャート
Fig. 1 Flowchart of PLK algorithm

3. 輝度変化にロバストな高速オプティカルフローアルゴリズム

3.1 輝度変化へのロバスト化

3.1.1 最小二乗法による解法

PLK アルゴリズムは対応する画素の輝度値が移動の前後で変化しないという式 (2) で表される輝度保存性を仮定している。しかし日射量の変化などの撮影環境の変化や蛍光灯のちらつきなどの雑音により輝度保存性は破綻し、フロー精度が劣化する。これまでに動画像符号化の動き検出において輝度変化への対応が研究されている¹¹⁾。オプティカルフロー推定においても理論的なアルゴリズムの研究が行われている¹²⁾。ここでは¹³⁾で用いられており式 (4) で表される輝度変化を考慮した誤差関数 E' を導入する。

$$E' = \sum_{(x,y) \in \Omega} (I(x,y) - J(x+v_x, y+v_y) + \xi)^2 \quad (4)$$

ここで ξ はウィンドウ内の大域的な輝度変化の値を表す。誤差関数 E と同様に E' を最小化する $v' = [v_x \ v_y \ \xi]^T$ を求めると式 (5) を得る。ただし、 I_x, I_y, I_t の後ろにつく (x, y) を省略した。

$$v' = G'^{-1} b' \quad (5)$$

$$G' = \sum_{(x,y) \in \Omega} \begin{bmatrix} I_x^2 & I_x I_y & I_x \\ I_x I_y & I_y^2 & I_y \\ I_x & I_y & 1 \end{bmatrix}$$

$$b' = \sum_{(x,y) \in \Omega} \begin{bmatrix} I_x I_t \\ I_y I_t \\ I_t \end{bmatrix}$$

3.1.2 差分平均による解法

式 (4) の誤差関数 E' はフレーム間でウィンドウ中の全画素の輝度値がそれぞれ ξ だけ変化していることを仮定している。この仮定の下で v が真のフローに近ければ ξ は各ウィンドウの対応する画素輝度値の差分平均で近似できると考えられる。よって ξ を以下の式 (6) で求める。

$$\xi = \frac{1}{W^2} \sum_{(x,y) \in \Omega} (I(x,y) - J(x+v_x, y+v_y)) \quad (6)$$

式 (6) で求められた ξ を用いてウィンドウ B の輝度値を補正後、式 (3) により v を求める。この解法は最小二乗法による解法と比較して行列 G のサイズが小さいため演算量が少ない。この方法は¹¹⁾で提案されている方法とほぼ同じであるが、実装のし易さを考慮して、二つのウィンドウの輝度平均の差分を取るのではなく、対応画素の輝度差分の平均を取っている。

3.2 高精度化と高速化

3.2.1 初期値

PLK アルゴリズムは最上位層のフロー推定の初期値を 0 とする．このとき輝度変化が乏しい領域では原理上フローが正しく推定できない．そこで動画像符号化の動き検出で採用されている初期値設定法¹⁴⁾を導入する．最上位層におけるフロー推定の初期値の設定法を図 2 に示す．上，右上，左に隣接する画素のフローを初期値候補とし，3 つのフローが予め決められた角度 θ の範囲内であれば対象画素は一つの動きを持つ領域内にあると判断し，それらの中間値を初期値として採用する．3 つのフローが角度 θ の範囲に収まらない場合，二つ以上の異なる動きを持つ領域の境界にあると判断し，初期値を 0 とする．一般に領域の境界は輝度変化が顕著であり初期値を 0 としてもフローを比較的正しく推定できる．

3.2.2 階層処理

PLK アルゴリズムは画素毎に上位層のフロー計算を別々に行うため，最下位層の全画素についてフローを計算する場合，非効率である．そこで動画像符号化の動き検出で採用されている階層処理法¹⁵⁾を導入する．ただしウィンドウサイズは全階層同一で，階層画像作成用のガウシアン・フィルタのカーネル・サイズは 3×3 である．この方法を図 3 に示す．これは上位層の 1 つの画素に対して得られたフローを下位層の対応する 4 つの画素のフロー推定における共通する初期値として用いる．このとき上位層の 1 つの画素は下位層の 4 つの画素の中心に位置するものとして，上位層の輝度値を下位層の 4 つの輝度値の線形補間により作成する．

階層数を L ，最下位層の画素数を N としたとき，PLK アルゴリズムの演算量は LN に比例し，提案法の演算量

は hN に比例する．ただし階層数 L に応じた係数 h は式 (7) で表される．

$$h = \sum_{i=1}^L \frac{1}{2^{2i-2}} \quad (7)$$

例えば $L = 3$ のとき PLK アルゴリズムの演算量は $3N$ であり，提案法の演算量は $1.3125N$ であり，演算量が約 44% に削減される．前述した初期値設定法と組み合わせた場合，輝度変化の乏しい領域において周辺の正しいフローを初期値として領域内の下位層に伝播させることにより精度が向上する．高精度化によりフロー推定の繰り返し回数を削減でき，さらに演算量を削減できる．

3.2.3 ウィンドウサイズ

一般にウィンドウサイズ W を大きくすると「ウィンドウ内の全画素の動きが同一である」という局所的な動きの同一性の仮定による制約が強くなりノイズによる誤推定が少なくなるためフローの推定精度が高くなる．一方で W の増大は演算量の増加を招く．前述の初期値設定法と階層処理法の組み合わせによるフロー推定の高精度化により精度を維持したまま W を小さくできる． W の縮小により W の 2 乗に比例して G と b の計算に関する演算量を削減できる．例えば W を 11 から 5 とすれば G と b の計算に関する演算量を約 9% に削減できる．

3.3 シミュレーション結果

提案法の効果を確認するためにシミュレーションを行った．使用した計算機の実環境は以下のとおりである．

- 計算機: emachines J2926
- CPU: Intel Celeron D 335 2.80GHz
- Memory: 2GB
- OS: Microsoft Windows 7
- コンパイラ: Microsoft Visual C++ .NET 2002
- 最適化オプション: 未使用

使用した 3 種類のテストシーケンスと各コレクトフローを図 4 に示す¹⁶⁾．Translating Tree(Trans.) と Diverging Tree(Diver.) は同じ木の画像であるが，Trans. の動きが右方向の平行移動に対して，Diver. の動きは中心から外側への発散（中心へのズーム）である．輝度変化へのロバスト性を評価するため，以下の式に従い Trans. と Diver. の J フレーム（次フレーム）の各画素に輝度変化を与えた．

$$J' = \begin{cases} J + \frac{J}{255} \times B & (-255 \leq B < 0) \\ J + \frac{255-J}{255} \times B & (0 \leq B \leq 255) \end{cases} \quad (8)$$

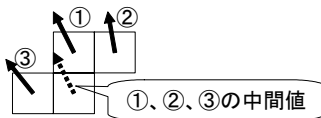


図 2 初期値設定法
Fig. 2 Initialization Method

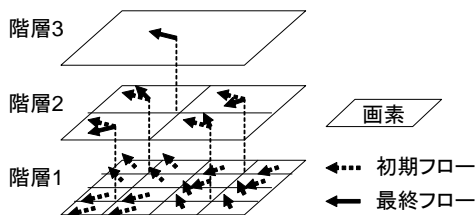


図 3 階層処理
Fig. 3 Hierarchical Method

表 1 比較対象の手法
Table 1 methods under comparison

名前	W	L	K	輝度変化	初期値	階層処理
W11	11	2(3)	1	なし	なし	なし
W5	5	2(3)	1	なし	なし	なし
LS	5	2(3)	1	最小二乗法	なし	なし
MD	5	2(3)	1	差分平均	なし	なし
LS+	5	2(3)	1	最小二乗法	あり	あり
MD+	5	2(3)	1	差分平均	あり	あり

ここで J は元の輝度値, J' は変化後の輝度値, B は輝度変化量を生成するためのパラメタ (以後輝度変化量 B) であり, フレーム内で共通である. ただし J と J' の後ろにつく (x, y) は省略した. オーバーフローとアンダーフローが防止され, 輝度変化量が均一ではなく, 輝度値の相対的な関係が保存されることに注意せよ. Yosemite (Yosem.) の動きは比較的複雑である. 上部の雲の右方向への移動に伴い雲の厚さの違いにより差し込む日光が変化し, この部分の輝度値が変化する. Yosem. は元々輝度値が変化しているため, 輝度変化を加えなかった. これらはコンピュータ制御で生成された動画であり, 各画素の正確なフローが既知である.

比較対象となる 6 つの手法を表 1 に示す. W, L, K はそれぞれウィンドウサイズ, 階層数, 階層毎の繰り返し推定回数を表す. W11 と W5 はオリジナルの PLK アルゴリズムであり, W がそれぞれ 11 と 5 である. W11 以外の W は 5 である. シーケンス毎の動きの大きさに応じて Trans. と Diver. の階層数は 2 とし, Yosem. の階層数は 3 とした. K はすべて 1 である. LS は最小二乗法で輝度変化を求めるアルゴリズムである. MD は差分平均で輝度変化を求めるアルゴリズムである. LS+ と MD+ はそれぞれ LS と MD に初期値設定と階層処理による高精度化と高速化を施したアルゴリズムである. LS+ と MD+ の初期値決定の角度 θ を 90° とした.

Trans. と Diver. に対して 6 つの手法に関する輝度変化量 B とフロー推定精度の関係を図 5 に示す. 横軸は輝度変化量, 縦軸は精度を表す. 精度の指標としてオプティカルフローの研究でよく用いられている平均角度誤差 (MAE: Mean Angle Error) を採用する. MAE は次式で表される.

$$MAE = \frac{1}{N} \sum_x \sum_y \left(\arccos \frac{\mathbf{v}_c \cdot \mathbf{v}_e}{\|\mathbf{v}_c\| \|\mathbf{v}_e\|} \right) \quad (9)$$

ここで $N, \mathbf{v}_c, \mathbf{v}_e$ はそれぞれ推定されたフローの総数, 正確なフロー, 対応する推定されたフローを表す. \cdot は内積を表し, $\|\cdot\|$ はノルムを表す. 従来法である W11 と W5 は輝度変化に対して精度が大きく劣化する. 提案法である LS, MD, LS+, MD+ は輝度変化に対して精度

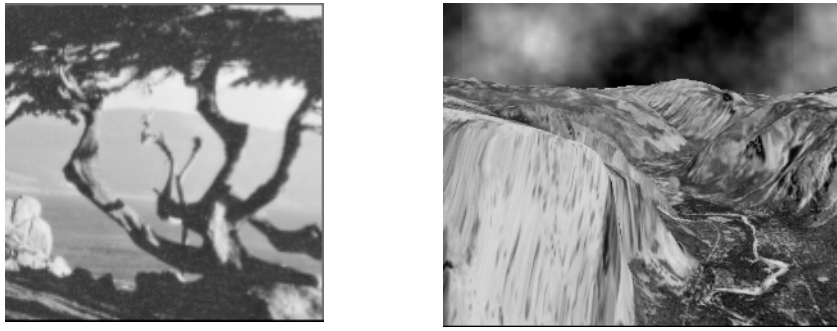
の劣化が少なく, ロバストであるといえる. また MD+ は LS や MD より精度が改善されており, LS+ と同等である. W5 と MD+ により推定されたフローをそれぞれ図 6 と図 7 に示す. このとき輝度変化量 B はどちらも +10 である. 輝度変化により W5 ではフローが激しく乱れるが, MD+ ではフローが大きく乱れることなく比較的正しく求められている.

次に Yosem. に関して 6 つの手法に対するプログラム実行時間とフロー推定精度の関係を図 8 に示す. 棒グラフは 1 フレームの平均実行時間を表し, 折れ線グラフは精度を表す. MD は LS より CPU 実行時間は短い, フロー精度が大きく劣化している. MD は差分平均で輝度変化を求めるので, 推定の初期段階においてフローが不正確なとき輝度変化を正しく求められず時間輝度勾配を正しく補正できないためフロー精度が悪い. LS は最小二乗法により輝度変化を求めるので, 初期フローの精度に大きく影響されず, フロー精度が比較的高い. LS+ の CPU 実行時間は LS の約半分であり, フロー精度も良い. MD+ の CPU 実行時間は MD の半分未満であり, フロー精度も大幅に良い. MD+ は初期値設定や階層処理により初期段階からフロー精度が高く差分平均でも輝度変化が正しく求められフロー精度が MD より大きく向上したと考えられる. MD+ は LS+ より CPU 実行時間が短く, フロー精度も良い. LS+ が輝度変化とフローを同時に求めるのに対して, MD+ は輝度変化を求めて時間輝度勾配を先に補正してからフローを求めるので, LS+ よりフロー推定の収束が少し速くなり, 繰り返し回数を同じにしたとき精度が少し向上すると考えられる. 従来 W5 に対して提案する MD+ は CPU 実行時間が約 47% に減少し, しかも精度が 3.4 度改善する. Yosemite に対する W5 と MD+ によるフローをそれぞれ図 6 と図 7 に示す. 上部の雲や左下の山のフローが改善していることが分かる. W11 に対して MD+ は CPU 実行時間が約 15% に減少し, しかも精度は同等である.

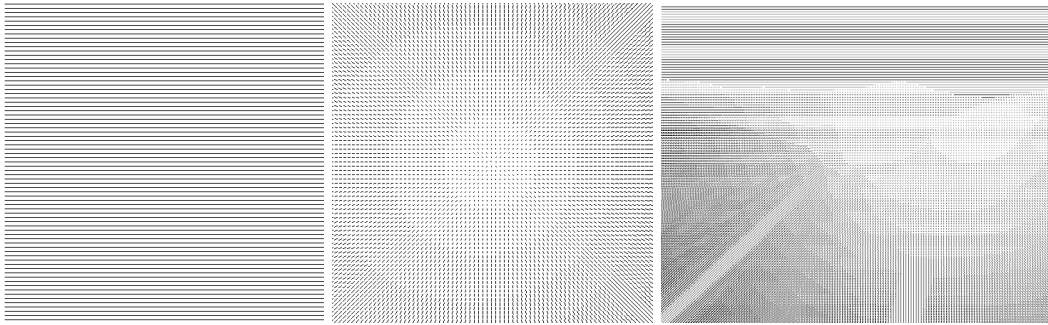
図 9 にカメラ入力画像の処理結果を示す. ここでは蛍光灯で照らされた室内を人物が左から右に移動している. 輝度変化により W5 ではフローが激しく乱れるのに対し, MD+ ではフローの乱れが抑えられていることを確認できる.

4. 回路構成

¹⁰⁾で提案された構成を基に, 輝度値の差分平均による輝度変化へのロバスト化と初期値設定と階層処理による高精度化・高速化を図った, オプティカルフロープロセッサのブロック図を図 10 に示す. 本プロセッサは階層画像作成部 (PIC), 空間輝度勾配行列算出部 (SGM), ウィン



(a) Test Images (Tree, Yosemite)



(b) Correct Flows (Translating Tree, Diverging Tree, Yosemite)

図 4 テストシーケンス
Fig. 4 Test Sequences

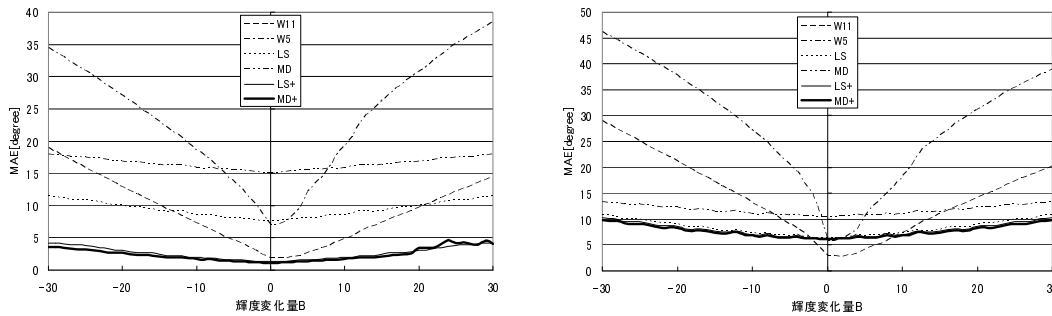


図 5 輝度変化量とフロー推定精度 (左: Trans., 右: Diver.)

Fig. 5 Relation between Luminance Change and Flow Accuracy (Left:Trans., Right:Diver.)

ドウ B 作成部 (WBC), 時間輝度勾配算出部 (TG), ミスマッチベクトル算出部 (MMV), オプティカルフロー算出部 (OPF), シーケンサ (SEQ), アドレスジェネレータ (AG), 各種画像メモリ, 初期値保存用 FIFO (FIFO) から構成される¹⁰⁾との大きな違いは次の 2 点である。第 1 は階層処理の方法が変わったのでウィンドウ A の画像を作成する回路が不要になり、省かれている。第 2 は TG で時間輝度勾配の他に輝度変化量を計算することである。TG は¹¹⁾で提案されているように二つのウィンドウの輝

度平均の差分を取るのではなく、対応画素の輝度差分の平均を取ることで輝度変化量を求めている。これは元々の計算対象である時間輝度勾配を再利用し、回路規模を削減するためである。

PIC は 2 : 1 サブサンプリングとガウシアンフィルタリングを行い、階層画像を作成する回路である。

SGM は空間輝度勾配行列 G を算出する回路である。ウィンドウ A の画素毎に空間輝度勾配 I_x, I_y を求め、空間輝度勾配行列の各要素の値を計算する。要素毎に全

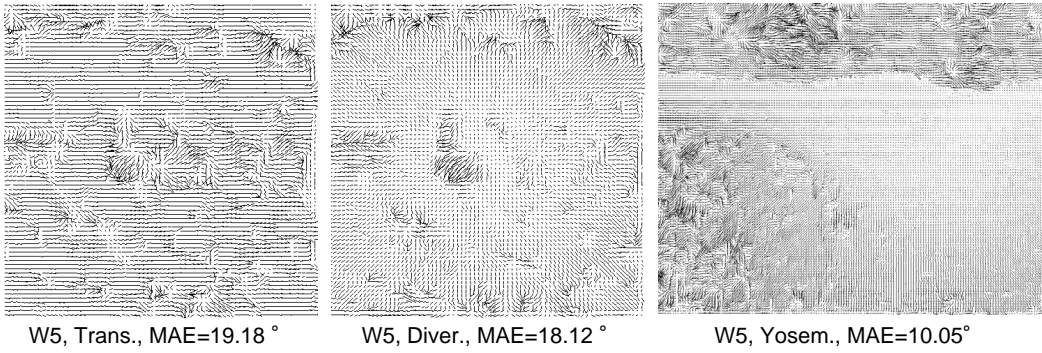


図 6 従来法によるフロー (左: Trans., 中央: Diver., 右: Yosem.)
 Fig.6 Flows by Conventional Method (Left:Trans., Center:Diver., Right:Yosem.)

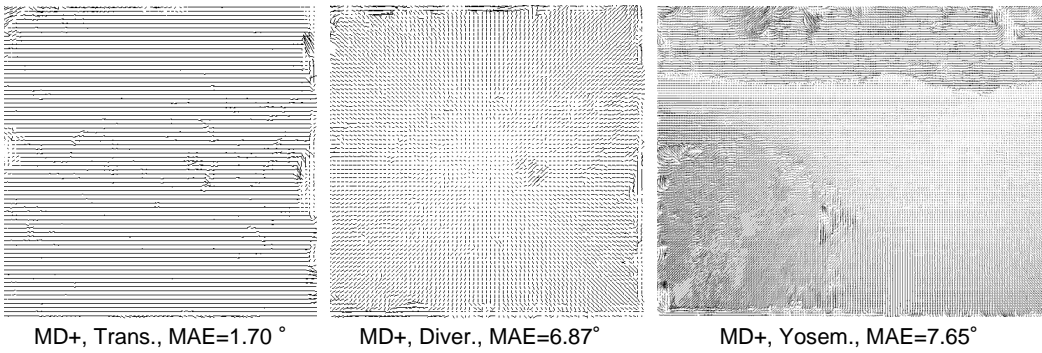


図 7 提案法によるフロー (左: Trans., 中央: Diver., 右: Yosem.)
 Fig.7 Flows by Proposed Method (Left:Trans., Center:Diver., Right:Yosem.)



図 9 カメラ入力画像に対する従来法および提案法によるフロー (左: カメラ入力画像, 中央: 従来法フロー, 右: 提案法フロー)
 Fig.9 Camera Captured Image and Flows by Conventional and Proposed (Left:Camera Captured Image, Center:Conventional Flow, Right:Proposed Flow)

画素分の値を合計することで空間輝度勾配行列 G が導出される。

WBC はウィンドウ B の画像を作成する回路である。最新のオプティカルフローを用いて、 J フレームの階層画像から移動先の小数画素の輝度値を双一次内挿により計算し、ウィンドウ B を作成する。

TG は画素毎の輝度変化と、それらの平均値を算出する回路である。TG ではウィンドウ A とウィンドウ B を用いて対応する画素の時間輝度勾配 (輝度値の差分) I_t を計算する。さらに、ウィンドウ内の全画素について差分を合計し、全画素数で割って平均 ξ を計算する。

MMV はミスマッチベクトル b を算出する回路であ

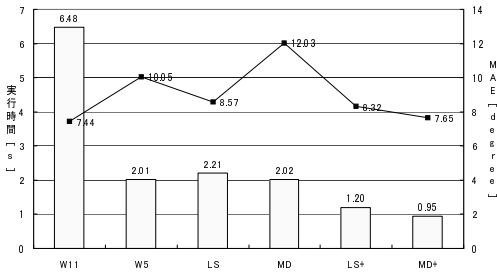


図 8 プログラム実行時間とフロー推定精度

Fig. 8 Relation between Computing Time and Flow Accuracy

最初に TG で求められた各画素の I_t を ξ で補正する。そして画素毎に SGM で求められた I_x, I_y と補正された I_t からミスマッチベクトルの各要素の値を計算する。そして要素毎に全画素分の値を合計することでミスマッチベクトル \mathbf{b} が導出される。

SGM, WBC, TG, MMV の各ブロックにおいて、1 サイクルでウィンドウ内の 1 画素を処理する演算器をウィンドウサイズ W と同じ数だけ配置し、これを 1 ステージとしてパイプライン的に動作させることで、1 サイクルでウィンドウ 1 行分の計算を可能としている。従って一組の \mathbf{G} と \mathbf{b} が W サイクルで計算できる。

OPF はオプティカルフローを算出する回路である。OPF では SGM, MMV で計算した \mathbf{G} と \mathbf{b} を用いて式 (3) を解き、増分オプティカルフローを計算する。算出された増分オプティカルフローを、前回までの計算で得られたオプティカルフローに加算し、これを更新する。他のブロックと計算速度を合わせるために、OPF は 1 個の増分オプティカルフローを W サイクルで計算する。従って OPF に含まれる除算器は低速でよく、引き戻し法に

基づくシンプルな構成でよい。

制御部は SEQ と AG から構成される。SEQ はフレーム毎の各画素についてフロー計算の処理順序を制御する。最上位層における初期値の決定も行う。AG は処理対象のデータを各種画像メモリから読み出すためのアドレスとオプティカルフローを外部メモリに書き込むためのアドレスを生成する。

本プロセッサは 1 画素のフロー推定における 1 回の繰り返し計算を W サイクルで実行する。式 (7) で表される階層数 L に応じた係数を h 、各階層における繰り返し回数を K 、1 フレームの (最下位層の) 画素数を N 、フレームレートを R (フレーム/s) としたとき、本プロセッサの実時間処理に必要なおよその動作周波数 F は以下の式で表される。

$$F = hNKWR \quad (10)$$

例えば W を 11 から 5 に削減すると動作周波数を約 45% に削減できる。これを前述の階層処理と組み合わせると動作周波数を約 20% に削減できる。さらに同一の半導体技術を仮定すると、メモリ面積は W の 2 乗、ロジック面積は W に比例して削減される。消費電力は動作周波数と回路面積の積に比例するため¹⁰⁾で提案された W11 に基づく従来の回路構成法と比較して MD+ に基づく提案法により消費電力を少なくとも 9% 未満に削減できると考えられる。しかもフロー精度は同程度である。

5. FPGA 実装

表 1 の 6 つの手法の内、W5 と LS と MD+ について FPGA 実装した。使用した FPGA は以下のとおりである。

- FPGA ボード: Celoxica RC2000
- FPGA: Xilinx Virtex-II XC2V6000
- LUT 数: 67,584
- 18Kb RAM 数: 144
- 18 × 18 ビット乗算器数: 144

実装結果を表 2 に示す。スループットはパラメタを ($L = 3, W = 5, K = 1$)、動作周波数を 30MHz と仮定している。MD+ はフロー精度が高いうえに回路規模が小さくスループット性能が高い。VGA 30 fps の実時間処理に必要な MD+ の動作周波数は約 62MHz である。これは¹⁰⁾の W11 に基づく ASIC 実装結果の約 19% であり、大幅な消費電力削減の可能性を意味する。

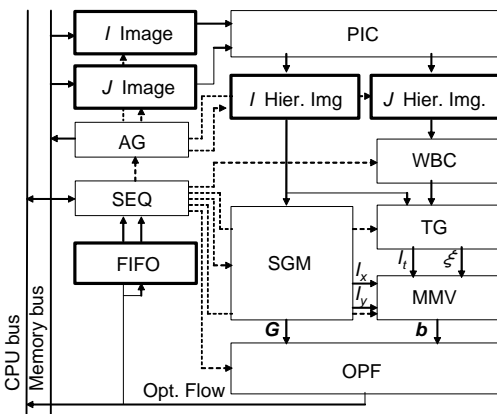


図 10 オプティカルフロープロセッサのブロック図

Fig. 10 Block Diagram of Optical Flow Processor

表 2 FPGA 実装結果
Table 2 FPGA Implementation Results

	W5	LS	MD+
LUT 使用数	52,697	87,978	35,598
RAM 使用数	123	123	96
乗算器使用数	138	143	108
最大動作周波数 (MHz)	32.18	30.23	32.54
スループット @30MHz(画素/s)	2,310k	2,310k	4,516k

6. ま と め

本研究では, PLK アルゴリズムに基づく輝度変化にロバストで高速なオプティカルフロー推定アルゴリズムを提案した. 最小二乗法より簡単な差分平均により輝度変化量を求め, 時間輝度勾配を補正し, ロバスト化を図った. 初期値設定法, 階層処理法, ウィンドウサイズの最適化により処理の高精度化と高速化を図った. シミュレーションの結果, 輝度変化量 B を +10 としたときの平均角度誤差は 18.65° から 4.29° に大きく改善され, これらの手法を組み合わせた提案アルゴリズムのロバスト性を確認した. 輝度変化を加えず精度を同程度とした場合の CPU 実行時間は従来の 15% となり, 高速性を確認した. さらに本アルゴリズムに基づく専用回路構成を提案し, FPGA 実装した. 提案回路は従来回路より輝度変化にロバストな上に, 同程度の精度とスループット性能を仮定したとき, 動作周波数が 20% に削減される. 同一の半導体技術を仮定すれば, 面積が 45% 未満, 電力が 9% 未満に削減されると考えられる. 本プロセッサの実時間画像認識や画像再構成への応用は今後の課題である.

謝 辞

本研究は東京大学大規模集積システム設計教育研究センターを通し, メンター株式会社の協力で行われたものである. 本研究に貢献した石原一氏に感謝する.

参 考 文 献

- 1) David A. Forsyth and Jean Ponce, "Computer Vision A Modern Approach," Person Education Inc., USA, 2003.
- 2) Al Bovik, "Handbook of image and video processing," Elsevier Academic Press, USA and UK, 2005.
- 3) B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artif. Intell.*, vol.17, pp.185-204, 1981.
- 4) B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," In *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, 1981.
- 5) J. L. Barron, D. J. Fleet, and S. S. Beauchemin, "Performance of optical flow techniques," *Int. J. Comput. Vis.*, vol.12, no.1, pp.43-77, 1994.
- 6) Carlo Tomasi and Takeo Kanade, "Detection and Track-

ing of Point Features," Technical Report CMU-CS-91-132, April 1991.

- 7) J. Y. Bouguet, "Pyramidal Implementation of the Lucas Kanade feature tracker description of the algorithm," Intel Corporation, Microprocessor Research Labs, OpenCV Documents, 1999.
- 8) Javier Diaz, Eduardo Ros, Francisco Pelayo, Eva M. Ortigosa, and Sonia Mota, "FPGA-Based Real-Time Optical-Flow System," *IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY*, VOL.16, NO.2, pp.274-279, February 2006.
- 9) Venkataraman Mahalingam, Koustav Bhattacharya, Nagarajan Ranganathan, Hari Chakravarthula, Robin Roberson Murphy, and Kevin Sheldon Pratt, "A VLSI Architecture and Algorithm for Lucas-Kanade-Based Optical Flow Computation," *IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS*, VOL.18, NO.1, pp.29-38, January 2010.
- 10) Y. Murachi, Y. Fukuyama, R. Yamamoto, J. Miyakoshi, H. Kawaguchi, H. Ishihara, M. Miyama, Y. Matsuda, and M. Yoshimoto, "A VGA 30-fps Realtime Optical-Flow Processor Core for Moving Picture Recognition," *IEICE TRANS. ELECTRON.*, VOL.E91-C, NO.4, pp.457-464, April 2008.
- 11) 上倉 一人, 渡辺 裕, 小林 直樹, 一之瀬 進, 安田 浩, "演算量低減を考慮したグローバル動き・輝度変化補償動画像符号化", *電子情報通信学会論文誌 B*, Vol.J82-B, No.9, pp.1676-1688, 1999 年 9 月.
- 12) Yeon-Ho Kima, Aleix M. Martinez, Avi C. Kak, "Robust motion estimation under varying illumination," *Image and Vision Computing* 23 (2005), pp.365-375, 2005.
- 13) J. M. ODOBEZ and P. BOUTHEMY, "Robust Multiresolution Estimation of Parametric Motion Models," *JOURNAL OF VISUAL COMMUNICATION AND IMAGE REPRESENTATION*, Vol.6, No.4, December, pp.348-365, 1995.
- 14) P. LIU, L. JIANG, H. NAKAYAMA, T. YOSHITAKE, H. KOMAZAKI, Y. WATANABE, H. ARAKI, K. MORIOKA, S. LEE, H. KUBOSAWA, Y. OTOBE, "A Low-Power MPEG-4 Codec LSI for Mobile Video Application," *IEICE TRANS. ELECTRON.*, VOL.E86-C, NO.4, pp652-660, APRIL 2003.
- 15) Kwon Moon Nam, Joon-Seek Kim, Rae-Hong Park, and Young Serk Shim, "A Fast Hierarchical Motion Vector Estimation Algorithm Using Mean Pyramid," *IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY*, VOL. 5, NO. 4, pp.344-351, AUGUST 1995.
- 16) ftp://ftp.csd.uwo.ca/pub/vision, John Barron the University of Western Ontario Department of Computer Science.

(2010 年 5 月 26 日受付)

(2010 年 10 月 4 日再受付)

深 山 正 幸 (正会員)



1988 筑波大学第三学群情報学類卒．
1988 (株)PFU 入社．1995 北陸
先端科学技術大学院大学情報科学研
究科博士前期課程了．1996 イノテッ
ク(株)入社．2000 金沢大学工学部
電気電子システム工学科助手．2004
金沢大学大学院自然科学研究科博士
(工学)．2006 金沢大学大学院自然
科学研究科講師(現職)．実時間画
像処理 VLSI のアルゴリズムとアー
キテクチャに興味を持つ．

田 村 賢 一



2009 金沢大学工学部電気電子シス
テム工学科卒．現在，同大学院自然
科学研究科博士前期課程在学中．主
として，オプティカルフロー検出用
LSI アーキテクチャの研究に従事．

松 田 吉 雄



昭 52 阪大・理・物理卒，昭 54 同大
大学院工学研究科応用物理学専攻修士
課程修了．昭和 58 同博士課程修了．
昭 60 三菱電機(株)入社．DRAM，
システム LSI，高周波化合物半導体
の研究・開発を経て平 17 年退職．平
17 より金沢大学大学院自然科学研
究科教授．現在，システム LSI 特に
画像処理関連 LSI の研究に興味を持
つ．工博．