

階層的な出力メッセージの平均化に基づく近似確率伝播法

小川原光一[†]

Approximate Belief Propagation by Hierarchical Averaging of Outgoing Messages

Koichi OGAWARA[†]

あらまし 本稿では、ノードの出力メッセージをそのノードの全出力メッセージの平均値で置き換え、さらに低解像度グラフから元のグラフへと階層的にメッセージを伝播することにより、画像に対して適用した場合に通常の確率伝播法と比較して計算時間を約 2 分の 1 から 3 分の 1 に、メモリ使用量を 40% に削減可能な近似確率伝播法を提案する。また、提案手法を CPU および GPU 上に実装し、Middlebury ステレオデータセットを使用して通常の確率伝播法と比較実験を行い、わずかな精度の低下と引き換えに計算時間とメモリ使用量の両面において性能が大きく改善されたことを示す。

キーワード 確率伝播法, ステレオ, GPU

1. はじめに

ステレオ視における視差計算 [1] や画像領域分割 [2] に代表される多くのビジョンの問題はマルコフ確率場によって定式化される。この問題は一般に NP-hard であることが知られているが [3], 確率伝播法 (Belief Propagation, BP) [4] や平均場近似 [5], グラフカット [6] などの大域的最適化法によって準最適解 (場合によっては最適解) を得ることができる。このとき、ステレオ視における視差計算や 3 個以上の領域分割を行う場合には、特殊な場合 [7], [8] を除くと一般に画素をノードとしたグラフに対する多値ラベル割り当て問題を解くことになり、確率伝播法もしくはグラフカットのいずれにおいても繰り返し法による収束計算が必要となるため高速処理には不向きであった。

確率伝播法では、ノード (画素) 数を N , ラベル数を L , 収束計算の繰り返し回数を T とすると、最大確率を与えるラベルを探索する max-product (min-sum) 法の計算量は $O(NL^2T)$ となる [4]。計算時間を削減する方法として、これまでに計算量がラベル数に対して線型なメッセージ計算法 [9], 低解像度画像から高解像度画像へと順に計算を行う階層的な確率伝播法 [9], [10], Graphics Processing Unit (GPU) の並

列処理機能を利用した高速化法 [11] などが提案されてきた。

一方 GPU や組み込み機器への実装を考えた場合には、計算時間の削減に加えてメモリ帯域のボトルネックや実装メモリ量の制約からメモリ使用量をいかに削減するかが課題となる。メモリ量を削減する方法については、これまでにメッセージの予測符号化法 [12], 画像のタイル分割に基づく計算法 [13], ラベル探索範囲の階層的な削減法 [14] などが提案されてきた。

しかし、上記の方法はいずれも確率伝播法のメッセージ計算式自体は変更していない。それに対して、本稿ではこの計算式自体を簡略化することによって、解の精度はわずかに低下するものの、画像に対して適用した場合に標準的な方法と比較して計算時間を約 2 分の 1 から約 3 分の 1 に、メモリ使用量を 40% に削減可能な近似確率伝播法を提案する。提案手法の特長は次の 2 点である。(1) ノードの出力メッセージをそのノードの全出力メッセージの平均値で置き換えることにより計算時間とメモリ使用量を大幅に削減する。(2) この平均化処理を低解像度グラフから元のグラフへと階層的に実行することにより近似に伴う精度の低下を抑制する。

なお、確率伝播法は木構造を持つグラフに対しては最適解を出力することが保証されているが、画像のように通常 4 連結グラフとして表現されるループを含むグラフに対しては、限定された解の範囲 [15] を除き収

[†]九州大学, 福岡県
Kyushu University, Japan

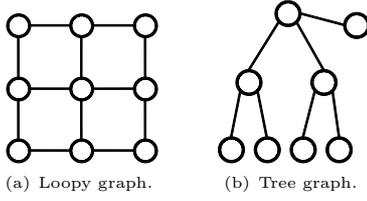


図 1 グラフ表現
Fig. 1 Graph representation.

束解の大域的な最適性は保証されていない．そのため，本稿ではステレオ視における視差計算のベンチマークとして標準的に使用されている Middlebury データセット [16] を使用し，実証的な観点から従来の確率伝播法に対する提案手法の評価を行う．また，提案手法は並列処理と親和性が高いため，CPU に加えて GPU への実装も行いその効果を検証した．

2. 確率伝播法

確率伝播法は，コンピュータビジョンやパターン認識への応用だけでなく，機械学習・データマイニング [10]・通信理論 [17] など非常に広範囲に用いられる有用な方法であるが，本稿ではビジョンの問題を例にとって議論を進める．ステレオ視における視差計算や画像の領域分割などの問題は，入力 $Y = (y_1, y_2, \dots, y_N)$ が与えられたときに画像の各画素にラベル $\{1, \dots, L\}$ を割り当てる問題として定式化される．ここで，図 1 (a) に示すように，画像の各画素をノードとみなし上下左右のノードとの間にエッジを持つ格子状の 4 連結グラフを考える．もし，あるノード i のラベル x_i が，対応する画素の入力値 y_i と隣接するノードのラベル $\{x_j | j \in N_e(i)\}$ のみから決定される場合には，これはマルコフ確率場と呼ばれ，最適なラベル割り当ては以下の式を最大化することによって得られる．

$$P(\mathbf{X}|\mathbf{Y}) = \prod_{k \leq N} \Phi(x_k, \mathbf{y}_k) \prod_{(i,j) \in A} \Psi(x_i, x_j) \quad (1)$$

ただし $\mathbf{X} = (x_1, x_2, \dots, x_N)$ であり， A は隣接するノード組の集合を， $N_e(i)$ はノード i に隣接するノードの集合を表す．また， $\Phi(x_k, \mathbf{y}_k)$ および $\Psi(x_i, x_j)$ はポテンシャル関数である．式 (1) の対数の負を取ると，

$$E(\mathbf{X}) = \sum_{i \leq N} D_i(x_i) + \sum_{(i,j) \in A} V(x_i, x_j) \quad (2)$$

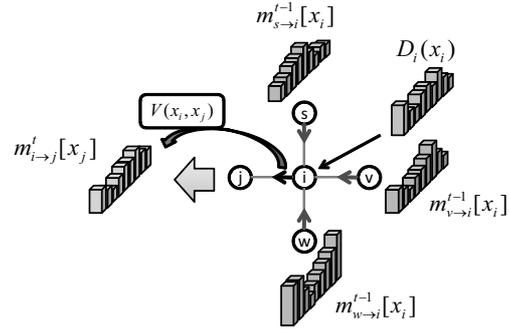


図 3 メッセージの計算
Fig. 3 Computation of messages.

と書き直すことができるため，この問題は評価関数 $E(\mathbf{X})$ を最小化する問題と等価であることが分かる．ここで， $D_i(x_i)$ は入力値 y_i を持つノード i にラベル x_i が割り当てられたときのコストを表し， $V(x_i, x_j)$ は隣接するノード i, j にそれぞれラベル x_i, x_j が割り当てられたときのペナルティを表す．通常，隣接するノードのラベルが異なる場合にペナルティを課すことから， $V(x_i, x_j)$ は平滑化項とも呼ばれる．

ここで，もしグラフが図 1 (b) に示すような木構造を持つ場合には，以下で説明する確率伝播法によって，有限回の繰り返し計算の後に式 (2) を最小化する大域的最適解 \mathbf{X}^* を得られることが知られている [4]．

確率伝播法では，図 3 に示すように各ノードから隣接するノードに対して以下で定義するメッセージを繰り返し伝播する．

t 回目の繰り返し計算においてノード i からノード j へ伝播するメッセージ $m_{i \rightarrow j}^t$ は，図 3 のヒストグラム表現に示すように要素数 L の配列として定義され，その x_j 番目の要素は以下の式によって計算される．

$$m_{i \rightarrow j}^t[x_j] = \min_{x_i} \left(V(x_i, x_j) + D_i(x_i) + \sum_{s \in N_e(i) \setminus j} m_{s \rightarrow i}^{t-1}[x_i] \right) \quad (3)$$

ただし，各要素の初期値は 0 とする．

メッセージ $m_{i \rightarrow j}^t$ は，グラフがループを含まない場合には，有限回の繰り返し計算の後，ノード j から見たノード i 以降の部分グラフ G' に含まれる全ノードラベルに関して式 (2) を周辺化 (marginalization) した結果に収束する．収束までの繰り返し回数は，ノード j と G' に含まれるノード群との最大距離に等しい．

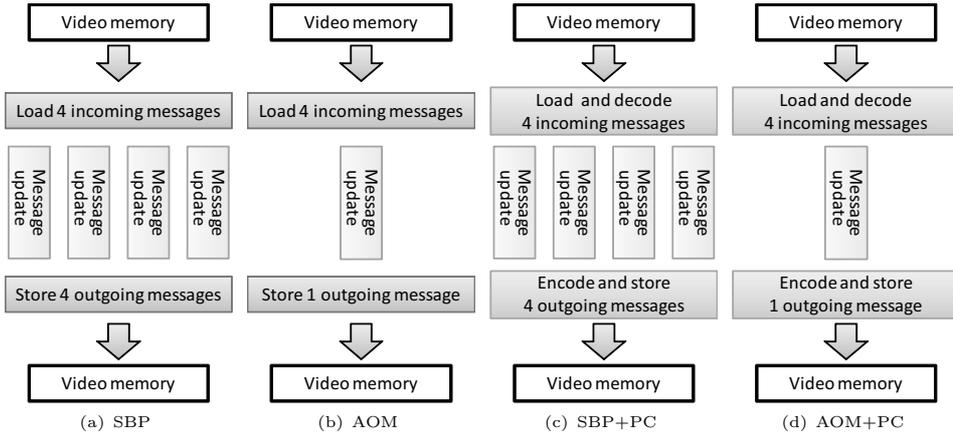


図 2 アルゴリズムの流れ：SBP は通常確率伝播法，AOM は出力メッセージの平均化に基づく近似確率伝播法，PC はメッセージの予測符号化法 (4.4.2 節) を表す。

Fig. 2 Flow of the algorithms: SBP means the Stanrad Belief Propagation algorithm, AOM means the approximated belief propagation algorithm by Averaging of Outgoing Messages and PC means Predictive Coding (Section 4.4.2).

ノード i に入力される全てのメッセージが収束した後，ラベルの推定値 x_i^* は以下のように計算される。

$$x_i^* = \underset{x_i}{\operatorname{argmin}} \left(D_i(x_i) + \sum_{j \in N_e(i)} m_{j \rightarrow i}^T[x_i] \right) \quad (4)$$

一方，図 1 (a) に示すように画像のようなループを含むグラフの場合には，確率伝播法によって大域的な最適解が得られる保証はないものの多くの場合に最適解に近い解を得られることが知られており [15], [18]，そのため確率伝播法は視差計算や領域分割など低レベルなビジョンの問題の標準的な解法の一つになっている。

3. 提案手法

3.1 出力メッセージの平均化

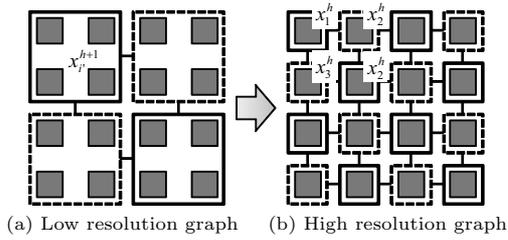
式 (3) の右辺第 3 項に示すとおり，あるノードの出力メッセージは，出力方向を除いた隣接ノードからの入力メッセージの和に基づき計算される。

提案手法では，この右辺第 3 項を式 (5) に示すように全隣接ノードからの入力メッセージの和と $\frac{\#N_e(i)-1}{\#N_e(i)}$ の積で置換する。ただし， $\#N_e(i)$ はノード i に隣接するノードの総数を表し，4 連結グラフの場合は 4 となる。

$$m_{i \rightarrow j}^t[x_j] = \min_{x_i} \left(V(x_i, x_j) + D_i(x_i) + \frac{\#N_e(i)-1}{\#N_e(i)} \sum_{s \in N_e(i)} m_{s \rightarrow i}^{t-1}[x_i] \right) \quad (5)$$

この新しいメッセージの計算式により，あるノードが出力する全てのメッセージは互いに等しくなる。そのため，図 2 (a) の”Message update” に示すように通常メッセージの計算は出力方向ごとに $\#N_e(i)$ 回計算する必要があるが，これが図 2 (b) に示すように 1 回に削減される。この計算処理の削減により，4 連結グラフの場合には後で示すように全体の計算時間が 2 分の 1 から 3 分の 1 程度に削減される。また，通常ノードごとに $\#N_e(i)$ 個の出力メッセージを保存する必要があるが，提案方法では出力メッセージが 1 つで済むため，式 (5) のデータコスト $D_i(x_i)$ の保存に必要なメモリ量が出力メッセージ 1 つを保存するために必要なメモリ量と等しいとすると，全体の使用メモリ量は $\frac{1+1}{1+\#N_e(i)}$ (4 連結グラフの場合は 40%) に削減される。

一般に式 (5) と式 (3) は一致しない。両者が一致するのは，例えば全入力メッセージの値が互いに等しい場合，つまり全出力メッセージの値が互いに等しい場合である。後でステレオ視の視差計算を例に実験的に示すが，一般的な画像に対する広義の画像領域分割問



(a) Low resolution graph (b) High resolution graph

図 4 階層的な平均化出力メッセージの伝播

Fig. 4 Hierarchical propagation of averaged outgoing messages.

題では、通常確率伝播法においてメッセージが収束したときに、多くのノードでは全出力メッセージが互いに等しくなっている。これが、式 (5) の単純な近似方法で良好な結果が得られる理由の 1 つである。

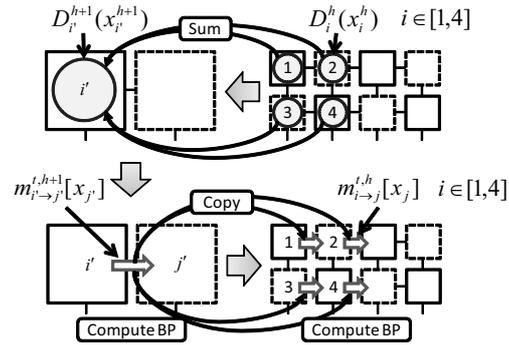
なお、最終的なラベルの推定値 x_i^* の計算方法は式 (4) と同じである。

また、平均場近似 [5] も、確率伝播法と同様に隣接ノードから受け取るメッセージに基づく並列メッセージ伝播法の 1 種とみなすことができる [19] が、提案するメッセージの平均化とは異なる思想に基づく方法である。提案手法では、隣接ノードから受け取るメッセージは通常確率伝播法と同じく要素数 L の配列であり、平均場近似のように要素の期待値を計算するわけではない。提案手法における平均化とは、あるノードから各隣接ノードに向けて独立に生成していた配列 (要素数は L) を、その平均である単一の配列 (要素数は L) で置き換えることを意味する。

3.2 階層的な平均化出力メッセージの伝播

メッセージの初期値を 0 として前節で述べた出力メッセージの平均化に基づく近似確率伝播法を計算すると、後で実験的に示すが、多くのノードでは最適解から外れた局所解にメッセージが収束する。

この問題を解決するために、階層的な確率伝播法 [9], [10] を用いる。この方法は、本来は長距離のメッセージ伝播を少ない繰り返し回数、つまり短時間で実現するための方法であるが、提案手法では主に平均化出力メッセージが局所解に陥ることを防ぐ目的で使用する。提案手法では、まず低解像度グラフ (図 4 (a)) においてメッセージが十分に収束するまで前節で述べた出力メッセージの平均化に基づく近似確率伝播法を計算する。次に、収束したメッセージを初期値として、高解像度グラフ (図 4 (b)) において、同様にメッセージが十分に収束するまで近似確率伝播法を計



(a) Low resolution graph (b) High resolution graph

図 5 階層間のメッセージ伝播

Fig. 5 Message propagation between hierarchies.

算する。これによって、高解像度グラフではより最適解に近い値を初期値としてメッセージが更新されていくため、局所解に陥る場合が少なくなる。

階層的な確率伝播法には、入力に依存せず一様な階層化を行う方法 [9] と、入力に依存した非一様な階層化を行うことによってさらに繰り返し数の低減を図る方法 [10] があるが、本研究では並列処理との親和性を考慮して一様な階層化を行う。

階層的な伝播法の手順は、図 5 の上段に示すように、まず

$$D_i^{h+1}(x_i^{h+1}) = \sum_{i \in [1,4]} D_i^h(x_i^h)$$

に従って低解像度グラフにおけるデータコストを順次計算する。ただし $h \in [1, H]$ は階層の番号を表し、 $h = 1$ は入力である高解像度グラフを、数字が 1 大きくなるにつれグラフの解像度が半分になることを意味する。次に、最も低解像度のグラフから

$$m_{i \rightarrow j}^{t,h}[x_j] = \min_{x_i} \left(V(x_i, x_j) + D_i(x_i) + \frac{\#N_e(i) - 1}{\#N_e(i)} \sum_{s \in N_e(i)} m_{s \rightarrow i}^{t-1,h}[x_i] \right)$$

に従って近似確率伝播法の計算を開始し、図 5 下段に示すようにある解像度のグラフで収束したメッセージを

$$m_{i \rightarrow j}^{t,h}[x_j^h] = m_{i' \rightarrow j'}^{t,h+1}[x_{j'}^{h+1}] \quad i \in [1,4], j' \in N_e(i')$$

に従い1段高い解像度のグラフの初期メッセージとして単純にコピーすることによって、順次解像度を上げながら近似確率伝播法を計算していく。なお、高解像度グラフ(階層 h)におけるメッセージの伝播先ノード j は、低解像度グラフ(階層 $h+1$)においてノード i' から見たノード j' と同じ方向に隣接するノードである。

また、後述する GPU を利用したノード単位での並列計算を念頭におき、変数への一貫性を持ったアクセスを保証するため、図4に示す実線で囲まれたノード群と点線で囲まれたノード群とで交互にメッセージを計算・伝播する方法を用いる。

4. 実験

実験では、ある対象を2台のカメラで撮影し、得られた2枚の画像を使って対象までの距離を画像の全画素について密に推定する問題を扱う。本章では、まず実験条件について述べた後、Middlebury データセット[16]を使用して提案手法と通常確率伝播法の比較実験を行い、実証的な観点から式(2)の評価関数 $E(X)$ の値と距離の逆数である視差の推定精度の2点について提案手法の評価を行う。

次に、提案手法が非常に大胆なメッセージ計算式の簡略化を行うにも関わらず良好な結果が得られる理由について、上記の実験結果の解析に基づき考察する。

最後に、提案手法を CPU に加えて GPU にも実装し、計算時間とメモリ使用量の削減効果について検証する。このとき、後述する計算量がラベル数に対して線型なメッセージ計算法とメッセージの予測符号化法を組み合わせることにより、特に GPU において提案手法の優位性が顕著に現れることを示す。

4.1 実験条件

本実験では、式(2)のデータコストを以下のように定義する。

$$D_i(x_i) = \min(|I_L(i) - I_R(i - x_i)|, \tau) \quad (6)$$

ただし、 $I_L(i)$ はステレオ対のうち左側画像の i 番目の画素値を表し、 $I_R(i - x_i)$ は右側画像の同じ画素から視差 x_i だけ水平に移動した $i - x_i$ 番目の画素値を表す。 τ は外れ値の影響を低減するための定数であり、本実験では $\tau = 30$ とした。

式(2)の平滑化項には隣接するラベルの差に線型な以下の関数を使用する。

$$V(x_i, x_j) = \min(c|x_i - x_j|, d) \quad (7)$$

ただし、 d は領域の輪郭などラベルが連続しない箇所において平滑化のペナルティを低減するための飽和項であり、本実験では $c = 14.0$, $d = 33.6$ とした。

4.2 評価関数の値と視差の推定精度の評価実験

本節では、Middlebury データセット[16]のうち Tsukuba, Venus, Teddy, Cones の4つを使用し、式(2)の評価関数 $E(X)$ の値と視差推定精度の2点について、提案手法と通常確率伝播法の比較を行う。なお、両手法とも階層的な計算法の有無に応じてそれぞれ実験を行い(計4通り)、階層的な計算法を使用する場合は階層数を4とした。以下、H# (Hierarchical method) は階層的な計算法とその階層数、SBP (Standard Belief Propagation) は標準的な確率伝播法、AOM (approximated belief propagation by Averaging of Outgoing Messages) は提案手法である出力メッセージの平均化に基づく近似確率伝播法を表す。また、参考までに、確率伝播法を使用せずに式(2)のデータコストのみを使用して画素ごとにラベルを推定した結果 (Without BP) も示す。

表1に、所定の総繰り返し回数 (# iterations) 後の評価関数の値(評価値)と、H4+SBP (総繰り返し回数80, 階層ごとの繰り返し回数20)の評価値を100%としたときの各評価値の比率を示す。標準的な確率伝播法(SBP)を使用した場合は、階層的な計算法(H)の有無によらず、総繰り返し回数80と320でほぼ同じ評価値(99%~101%)が得られていることから、総繰り返し回数80でほぼ収束しており、このときの評価値が準最適値であるとみなすことができる。一方、出力メッセージの平均化に基づく近似確率伝播法(AOM)を使用した場合は、階層的な計算法を使用する場合には評価値は101%~104%程度に納まるが、階層的な計算法を使用しない場合には113%~122%と準最適値から大きく外れた値になる。

図6に、TsukubaとVenusの2つのデータセットについて、総繰り返し回数を80とした場合の評価関数 $E(X)$ の変化をプロットしたグラフを示す。階層的な計算法を使用する場合は、階層が変わった直後に大きく評価値が下がり、また各階層では特に低解像度のグラフにおいて数回の繰り返しでほぼ評価値が収束していることが分かる。

表2に視差の推定誤差を示す。評価の方法は、[1]に

表 1 評価関数 $E(\mathbf{X})$ の値: H は階層的な計算法, SBP は標準的な確率伝播法, AOM は出力メッセージの平均化に基づく近似確率伝播法を表す.

Table 1 Value of evaluation function $E(\mathbf{X})$: H means Hierarchical method, SBP means the Standard Belief Propagation algorithm and AOM means the approximated belief propagation algorithm by Averaging of Outgoing Messages.

Algorithm	# iterations	Tsukuba		Venus		Teddy		Cones	
		$E(\mathbf{X})$	[%]	$E(\mathbf{X})$	[%]	$E(\mathbf{X})$	[%]	$E(\mathbf{X})$	[%]
H4+SBP	80	484814	100	875666	100	1370100	100	1674602	100
	320	482896	100	874329	100	1367032	100	1670682	100
H4+AOM (proposed)	80	505212	104	888761	101	1425441	104	1730117	103
	320	504464	104	888468	101	1423999	104	1726733	103
SBP	80	483941	100	878206	100	1388240	101	1686479	101
	320	482234	99	876619	100	1381769	101	1684669	101
AOM	80	574981	119	1064859	122	1643556	120	1905491	114
	320	560431	116	1035516	118	1628426	119	1893221	113
Without BP	0	4584873	946	8314305	949	8881040	648	8815741	526

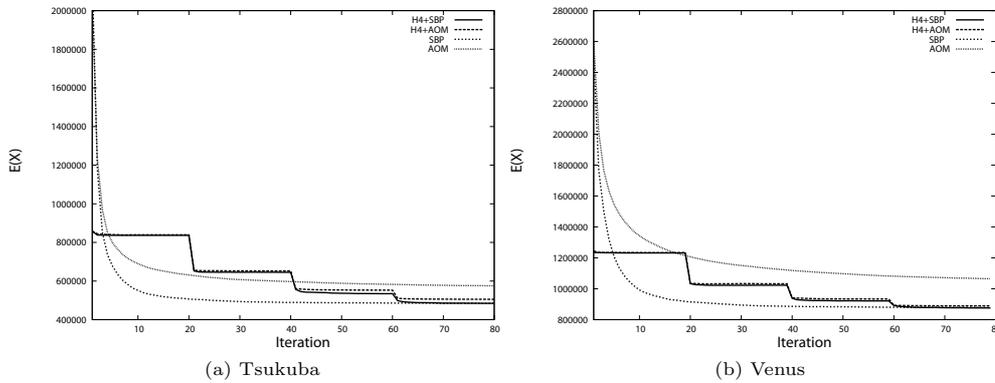


図 6 評価関数 $E(\mathbf{X})$ の値の変化: H は階層的な計算法, SBP は標準的な確率伝播法, AOM は出力メッセージの平均化に基づく近似確率伝播法を表す.

Fig. 6 Change in value of evaluation function $E(\mathbf{X})$: H means Hierarchical method, SBP means the Standard Belief Propagation algorithm and AOM means the approximated belief propagation algorithm by Averaging of Outgoing Messages.

表 2 視差の推定誤差: H は階層的な計算法, SBP は標準的な確率伝播法, AOM は出力メッセージの平均化に基づく近似確率伝播法を表す.

Table 2 Error in disparity estimation: H means Hierarchical method, SBP means the Standard Belief Propagation algorithm and AOM means the approximated belief propagation algorithm by Averaging of Outgoing Messages.

Algorithm	# iterations	Tsukuba		Venus		Teddy		Cones	
		nonocc [%]	all [%]						
H4+SBP	80	2.13	4.18	0.95	2.06	11.71	18.38	5.36	13.56
	320	2.11	4.17	0.99	2.10	11.62	18.28	5.28	13.47
H4+AOM (proposed)	80	2.39	4.51	1.62	2.79	12.11	19.06	5.73	14.21
	320	2.33	4.50	1.62	2.81	12.11	19.05	5.68	14.13
SBP	80	2.48	4.55	1.37	2.50	12.28	18.81	5.06	13.33
	320	2.18	4.24	1.35	2.46	12.26	18.82	5.09	13.37
AOM	80	4.47	6.59	8.58	9.92	16.45	23.63	10.13	19.26
	320	3.82	5.95	7.69	9.03	16.19	23.39	9.76	18.94
Without BP	0	47.02	48.17	66.32	66.87	71.34	74.23	69.62	72.95

ならって視差の真値と推定された視差の差が1より大きいものを推定誤りとみなし, 表の all に全画素に対

する誤り画素の比率 (%) を, 表の nonocc に隠蔽によって二枚の画像に対応点が存在しない画素を省いて

表 3 H4+SBP (総繰り返し回数 80) における最終出力メッセージの分散

Table 3 Variance of the final outgoing messages using H4+SBP (80 iterations in total).

	Tsukuba	Venus	Teddy	Cones
Ratio of white pixels [%]	91.7	91.9	84.6	83.7
Mean variance of black pixels	15.79	10.14	4.12	4.29
Mean variance of white pixels	0.00	0.00	0.00	0.00

計算した誤り画素の比率 (%) を示す。視差の推定誤差についても、表 1 と同様の傾向が見られる。すなわち、H4+SBP と SBP はだいたい同程度の推定誤差となり、H4+AOM は H4+SBP よりもわずかに悪い結果となるが、AOM はその他の結果と比較して大幅に悪い結果となる。

なお、評価値が小さいラベル割り当てが必ずしも視差の推定誤差を小さくするとは限らないが、全般的に両者の相関は高く、このことを利用して次節で解析を行う。また、図 7 に推定された視差画像を示す。他と比べて AOM の視差推定誤差が大きいことが見てとれる。

4.3 考察

本節では、階層的な計算法と出力メッセージの平均化法を組み合わせた場合 (H4+AOM) には評価値が標準的な確率伝播法による評価値の 101% ~ 104% 程度に納まり、出力メッセージの平均化法のみを使用した場合 (AOM) には評価値が大幅に低下する理由について考察する。

図 8 に、H4+SBP (総繰り返し回数 80) の最終状態において各ノードごとに 4 出力メッセージの分散を計算し、分散が 0.1 以上の画素を黒、それ以外を白で描画した結果を示す。また、表 3 に、図 8 の白画素の全画素に占める割合、黒画素の分散値の平均、白画素の分散値の平均を示す。これより、全ノードのうち 85 ~ 90% 程度のノードでは、通常確率伝播法において最終的にノードからの出力メッセージが全て等しくなっている (分散がほぼ 0) ことが分かる。また、図 7 (b) との比較により、出力メッセージが等しくないノードは主にラベルの値が変化する領域輪郭付近のノードであることが分かる。

一般に領域輪郭付近のノードの数は、領域内部にあり隣接するラベル値が等しいノードの数と比較して少なく、今回使用したデータセットでも同様の傾向があ

る。そのため、(1) 領域内部の多数のノードにおいて出力メッセージが最適値に収束し、かつ (2) 領域輪郭付近で大きくラベル値が変化しない場合には、領域輪郭付近の少数のノードは輪郭の両側から伝播される出力メッセージによって平滑化され、結果として真値に近い値に収束することが期待される。

このうち (2) は入力の条件でありアルゴリズム側では対処できないため、ここでは (1) の条件について考える。通常確率伝播法では、メッセージの初期値を 0 とし、収束するまで式 (3) に従いメッセージを更新する。ループを含むグラフの場合でも、式 (3) は局所的には正しい更新式であり、多くの場合に最適解に近い解に収束することが知られている。しかし、提案手法である式 (5) は、局所的に見ても正しい更新式ではなく、図 9(e) に示すように多くのノードで出力メッセージが準最適値から外れた局所解に陥ってしまう。

図 9 (a) は Venus データセットでの H4+SBP による視差の推定誤差を、図 9 (b)-(e) は同データセットでの H#+AOM による視差の推定誤差を、それぞれ誤って推定されたノードを黒で、正しく推定されたノードを白で示したものである。図中の variance は、図 8 (b) の白ノード、つまり本来は最終的に出力メッセージが等しくなるノードにおいて、最終階層 (元の解像度のグラフ) における初期メッセージと準最適最終出力メッセージ (互いに等しい) の差の 2 乗平均を計算したものである。すなわち、メッセージの初期値がどれだけ準最適なメッセージに近いを示す。また、図中の error ratio は、図 8 (b) の白ノードにおいて、最終的に誤って推定されたノードの割合を % で示す。

図 9 より、階層の数が増える程 variance の値が小さい、すなわちメッセージの初期値が準最適な最終メッセージに近くなり、同時に error ratio の値も小さくなる、すなわち誤って推定されたノードの数が少なくなることが分かる。つまり、階層的な計算法により、局所解の回避と準最適解への漸近を両立しながら平均化出力メッセージが上の階層に伝播されていき、この挙動が良好な推定結果に寄与していることが分かる。なお、いずれの場合も、最も低い解像度におけるメッセージの初期値は 0 とし、各階層における繰り返し回数は 20 としている。

4.4 計算時間およびメモリ使用量削減の評価実験
次に、CPU に加えて GPU に提案手法を実装し、計算時間およびメモリ使用量の削減効果について検証

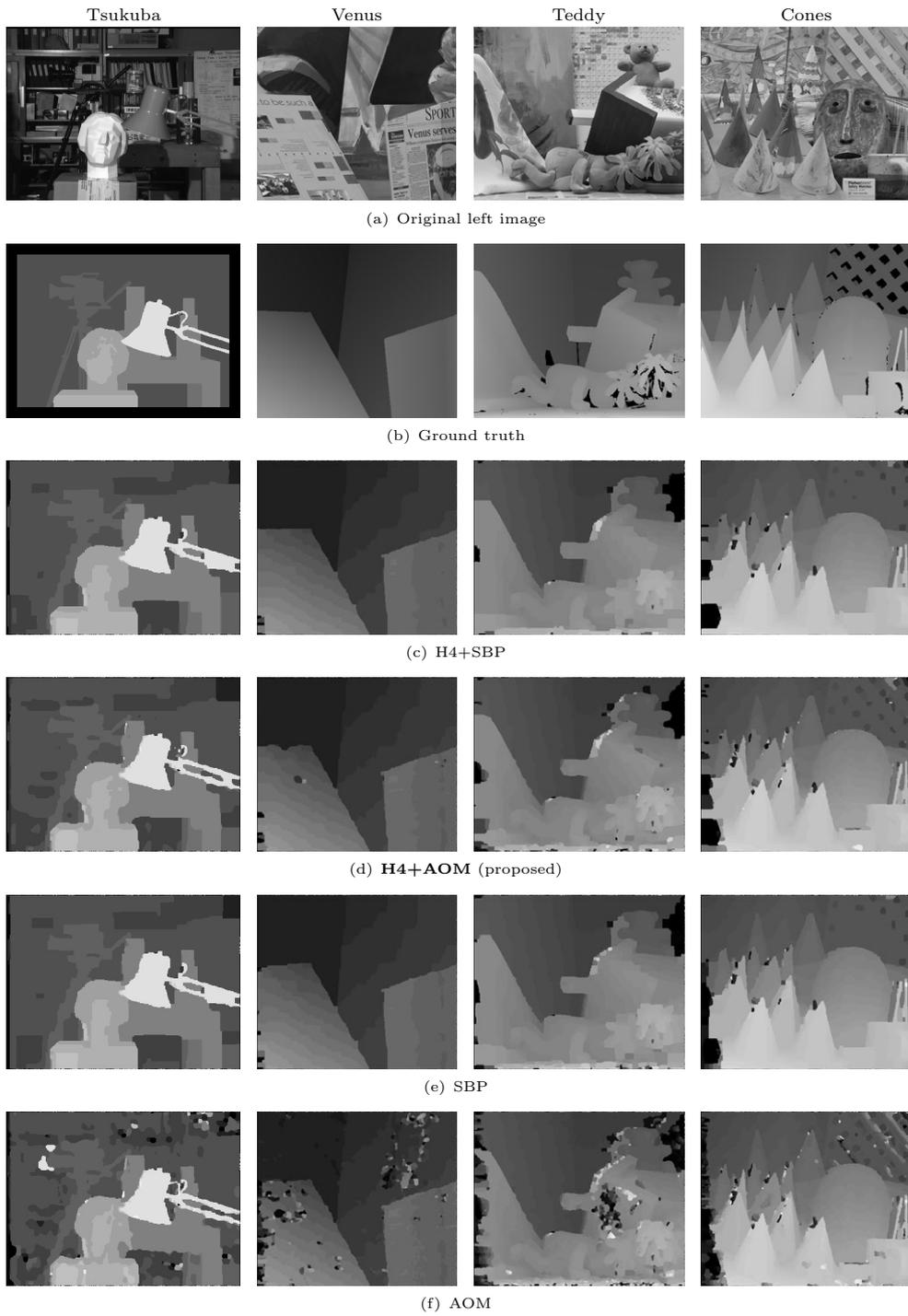


図 7 推定された視差画像
Fig.7 Estimated disparity map.

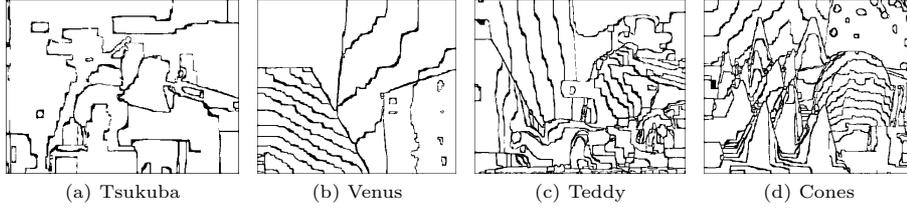


図 8 最終出力メッセージの分散：黒は分散が 0.1 より大きいノード，白はその他を表す．

Fig.8 Variance of the final outgoing messages: a black pixel means variance > 0.1 and a white pixel means otherwise.

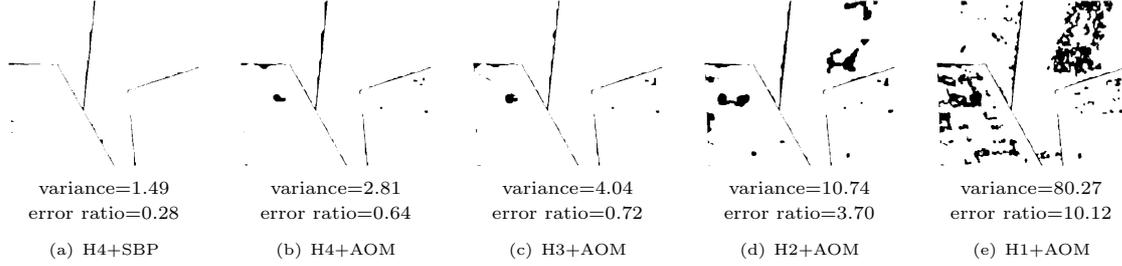


図 9 Venus データセットにおける誤推定ノード：黒は誤推定されたノード，白は正しく推定されたノードを表す．variance は最終階層における初期メッセージと準最適な最終メッセージの差の二乗平均を表し，error ratio は誤推定ノードの割合を表す．

Fig.9 Error in disparity estimation in Venus dataset: a black pixel means a wrongly estimated node and a white pixel means a correctly estimated node. Variance means the mean squared difference between the initial message at the final hierarchy and the optimal message. Error ratio means the ratio of the wrongly estimated nodes.

する．CPU 版は Visual Studio 2008 (C++) を用いて実装し，Xeon 3.0Ghz, Memory 2G の環境で評価を行った．また，GPU 版は CUDA [20] を用いて実装し，GPU には GeForce GTX285 を使用して評価を行った．

前節の結果より，出力メッセージの平均化に基づく近似確率伝播法では精度向上のために階層的な計算法の併用が欠かせないことが分かっているが，比較の観点から階層的な計算法の有無に応じてそれぞれ実験を行った．

また，提案手法を従来の高速化法と併用することによって計算時間やメモリ使用量をさらに削減できることを示す例として，計算量がラベル数に対して線型なメッセージ計算法とメッセージの予測符号化法を実装し，計算時間とメモリ使用量について評価実験を行った．なお，計算量がラベル数に対して線型なメッセージ計算法は常に使用することとし，4.2 節の 4 種類のアルゴリズムにメッセージの予測符号化法の有無を加えた計 8 種類のアルゴリズムの評価を行った．

4.4.1 計算量がラベル数に対して線型なメッセージ計算法 [9]

式 (3) のメッセージ計算では，ノード i から j に伝播される要素数 L のメッセージ $m_{i \rightarrow j}^t$ の各要素について，これを最小化する x_i を L 個のラベルの中から求めるため，その計算量は $O(L^2)$ となる．

式 (3) は以下のように書き換えることができるが，

$$m_{i \rightarrow j}^t[x_j] = \min_{x_i} (V(x_i, x_j) + h_{i \rightarrow j}^t[x_i]),$$

$$h_{i \rightarrow j}^t[x_i] = D_i(x_i) + \sum_{s \in N_e(i) \setminus j} m_{s \rightarrow i}^{t-1}[x_i] \quad (8)$$

$V(x_i, x_j)$ が式 (7) で定義される場合には，以下の手順によってメッセージを線型時間で計算することができる．ただし，以降では $m_{i \rightarrow j}^t[x_j]$, $h_{i \rightarrow j}^t[x_i]$ を $m[x_j]$, $h[x_i]$ と略す．

(1) 初期化

for x_j from 1 to L :

$$m[x_j] \leftarrow h[x_j]$$

(2) 前向き計算

for x_j from 2 to L :

$$m[x_j] \leftarrow \min(m[x_j], m[x_j - 1] + c)$$

(3) 後ろ向き計算

for x_j from $L - 1$ to 1 :

$$m[x_j] \leftarrow \min(m[x_j], m[x_j + 1] + c)$$

(4) 飽和处理

for x_j from 1 to L :

$$m[x_j] \leftarrow \min(m[x_j], \min_{x_i} h[x_i] + d)$$

これを提案手法である式 (5) に適用する場合には、式 (8) を以下のように書き換えればよい。

$$h_{i \rightarrow j}^t[x_i] = D_i(x_i) + \frac{\#N_e(i) - 1}{\#N_e(i)} \sum_{s \in N_e(i)} m_{s \rightarrow i}^{t-1}[x_i] \quad (9)$$

4.4.2 メッセージの予測符号化法 [12]

$V(x_i, x_j)$ が式 (7) で定義されるようにラベルの差に線型な関数である場合には、次の式が満足される。

$$|m[x_j + 1] - m[x_j]| \leq c \quad (10)$$

ここでメッセージの差を $pc(x_j) = m[x_j + 1] - m[x_j]$ と定義すると、 $m[x_j + 1]$ は $m[x_j]$ と $pc(x_j)$ から計算することができる。このとき、 $|pc(x_j)| \leq c$ が常に満足されるため、最大値が c となるように任意のビット数で $pc(x_j)$ を符号化することにより、 $m[x_j]$ の圧縮と量子化誤差の範囲での正確な復元が可能になる。[12] では $m[x_j]$ を 32bit の float 型で表現し、ビット数 4 で符号化することによって、全体としてメッセージ量を 8 分の 1 に削減している。この手法をメッセージの予測符号化法 (Predictive Coding, PC) と呼ぶ。

予測符号化を実行する場合は、確率伝播法にメッセージの符号化処理と復号処理が追加されるため、一連の処理の流れは図 2 (c) もしくは (d) のようになる。

4.4.3 実験結果

表 4 に、全てのアルゴリズムにおいて総繰り返し回数を 80 とし、4 つのデータセットに対して計 8 種類のアルゴリズムを CPU および GPU 上で実行したときの、確率伝播法の計算に要した計算時間と、H4+SBP (CPU) の計算時間を 100 % としたときの各計算時間の比率を示す。また、表の Mem に H4+SBP のメモリ使用量を 100 % としたときの各メモリ使用量の比率を表す。

データセットによってノード数 (N) とラベル数 (L)

が異なるため単純な比較はできないが、比率に着目するとどのデータセットの結果も同様の傾向を持つことが分かる。すなわち、GPU は CPU と比べて 20 ~ 50 倍程度速く、最も計算時間の少ない組み合わせは H4+AOM+PC (GPU) である。メモリ使用量については、H4+AOM+PC は AOM+PC に次いで 2 番目に少ないが、前節で示したように階層的な計算法を用いない AOM+PC は精度が低い。よって、最も優れたアルゴリズムは H4+AOM+PC (GPU) であるといえる。

ここで注意したいのは、CPU の場合はメッセージの予測符号化法 (PC) を併用することによりいづも計算時間が増大しているが、GPU の場合は出力メッセージの平均化 (AOM) と PC を併用することによって逆に計算時間が短くなっている点である。メッセージの予測符号化法を用いると、メッセージの符号化処理と復号処理が別途必要となるため、CPU においては計算時間が増加する。一方、GPU においても同様に計算時間が増加するはずであるが、出力メッセージの平均化を伴う場合にはコード量とメモリ使用量がさらに削減されるため、GPU のようにビデオメモリに対するアクセス遅延が大きく、かつレジスタや共有メモリなど利用可能な計算資源が少ない環境では、コード量増加に伴う PC の計算時間増加の効果を、利用可能な計算資源の増加に伴う PC の効率化の効果が上回ったと考えられる。この効果は、アルゴリズムを実装する際の最適化や GPU の持つ計算資源の量にも密接に依存する。今回はほぼ同じ C 言語のコードを CPU と GPU で使用しているが、最適化に関する話題は本稿の対象外であるためこれ以上は扱わない。なお、アルゴリズムが同じであれば CPU と GPU の結果が等しくなることは確認している。

5. まとめ

本稿では、確率伝播法において、ノードの出力メッセージをそのノードの全出力メッセージの平均値で置き換えることにより計算時間とメモリ使用量を大幅に削減し、同時にこの平均化処理を低解像度グラフから元のグラフへと階層的に実行することによって近似に伴う精度の低下を抑制する近似確率伝播法を提案した。

また、Middlebury データセットを使用して実証実験を行い、標準的な確率伝播法と比較して、わずかな精度の低下と引き換えに、計算時間とメモリ使用量を大幅に削減できることを示した。さらに、提案手法は

表 4 メモリ使用量と計算時間 [ms] の比較
Table 4 The required amount of memory and computational time [ms].

Algorithm	Mem [%]	Tsukuba (N=388x244, L=16)				Venus (N=434x383, L=20)				Teddy (N=450x375, L=60)				Cones (N=450x375, L=60)			
		CPU	GPU	CPU	GPU	CPU	GPU	CPU	GPU	CPU	GPU	CPU	GPU				
H4+SBP	100	5338	100	160	3.0	11873	100	299	2.5	42213	100	985	2.3	42672	100	1001	2.3
H4+SBP+PC	34	6633	124	277	5.2	13054	110	474	4.0	38731	92	1106	2.6	39276	92	1126	2.6
H4+AOM	44	2421	45	71	1.3	5570	47	126	1.1	16661	39	401	0.9	16656	39	411	1.0
H4+AOM+PC	27	2746	51	<u>63</u>	<u>1.2</u>	6122	52	<u>111</u>	<u>0.9</u>	18437	44	<u>284</u>	<u>0.7</u>	18428	43	<u>293</u>	<u>0.7</u>
SBP	94	16462	308	426	8.0	37405	315	822	6.9	130161	308	2773	6.6	130813	307	2799	6.6
SBP+PC	28	19934	373	733	13.7	38907	328	1283	10.8	115694	274	3067	7.3	115639	271	3095	7.3
AOM	38	7752	145	184	3.4	17249	145	340	2.9	51119	121	1138	2.7	51128	120	1159	2.7
AOM+PC	<u>21</u>	8255	155	155	2.9	18779	158	292	2.5	55804	132	784	1.9	56041	131	800	1.9

並列処理と親和性が高いことからこれを CPU に加えて GPU 上に実装し, CPU と比較して計算時間を大幅に削減できることを示した .

謝辞

本研究は, 科学研究費補助金若手 (B)(21700224) および科学技術総合推進費補助金「若手研究者の自立的な研究環境整備促進」の補助を受けている .

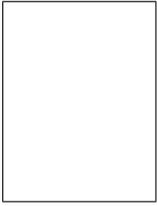
文 献

- [1] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International Journal of Computer Vision*, vol.47, no.1, pp.7–42, 2002.
- [2] X. Liu, O. Veksler, and J. Samarabandu, "Graph cut with ordering constraints on labels and its applications," *Proc. of IEEE Conference on Computer Vision and Pattern Recognition: CVPR*, pp.1–8, 2008.
- [3] S.E. Shimony, "Finding the maps for belief networks is np-hard," *Artificial Intelligence*, vol.68, no.2, pp.399–410, 1994.
- [4] W.T. Freeman, E.C. Pasztor, and O.T. Carmichael, "Learning low-level vision," *International Journal of Computer Vision*, vol.40, no.1, pp.25–47, 2000.
- [5] J. Zhang and G.G. Hanauer, "The application of mean field theory to image motion estimation," *IEEE Transactions on Image Processing*, vol.4, no.1, pp.19–33, 1995.
- [6] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.23, no.11, pp.1222–1239, 2001.
- [7] D. Schlesinger and B. Flach, "Transforming an arbitrary minsum problem into a binary one," *Technical Report TUD-FI06-01*, Dresden University of Technology, 2006.
- [8] V. Kolmogorov and C. Rother, "Minimizing non-submodular functions with graph cuts - a review," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol.29, no.7, pp.1274–1279, 2007.
- [9] P.F. Felzenszwalb and D.P. Huttenlocher, "Efficient

belief propagation for early vision," *International Journal of Computer Vision*, vol.70, no.1, pp.41–54, 2006.

- [10] L. Xiong, F. Wang, and C. Zhang, "Multilevel belief propagation for fast inference on markov random fields," *Proc. of IEEE International Conference on Data Mining*, pp.371–380, 2007.
- [11] Q. Yang, L. Wang, R. Yang, S. Wang, M. Liao, and D. Nister, "Real-time global stereo matching using hierarchical belief propagation," *Proc. of BMVC*, pp.989–998, 2006.
- [12] T. Yu, R.-S. Lin, B. Super, and B. Tang, "Efficient message representations for belief propagation," *Proc. of ICCV*, pp.1–8, 2007.
- [13] C.-K. Liang, C.-C. Cheng, Y.-C. Lai, L.-G. Chen, and H.H. Chen, "Hardware-efficient belief propagation," *Proc. of CVPR*, pp.80–87, 2009.
- [14] Q. Yang, L. Wang, and N. Ahuja, "A constant-space belief propagation algorithm for stereo matching," *Proc. of CVPR*, pp.1–8, 2010.
- [15] Y. Weiss and W.T. Freeman, "On the optimality of solutions of the max-product belief-propagation algorithm in arbitrary graphs," *IEEE Trans. on Information Theory*, vol.47, no.2, pp.736–744, 2001.
- [16] "http://vision.middlebury.edu/stereo/".
- [17] Y. Kabashima, "A cdma multiuser detection algorithm on the basis of belief propagation," *Journal of Physics A: Mathematical and General*, vol.36, no.43, pp.11111–11121, 2003.
- [18] J. Mooij and H. Kappen, "Sufficient conditions for convergence of loopy belief propagation," *Proc. of the Twenty-First Conference on Uncertainty in Artificial Intelligence (UAI-05)*, pp.396–403, 2005.
- [19] Y. Weiss, "Comparing the mean field method and belief propagation for approximate inference in mrfs," *Advanced Mean Field Methods - Theory and Practice*, pp.229–240, MIT Press, 2001.
- [20] NVIDIA Corporation, "NVIDIA CUDA programming guide," version 2.3 edition, 2009.

(平成 xx 年 xx 月 xx 日受付)



小川原光一（正員）

2002年東京大学大学院工学系研究科電子情報工学専攻博士課程修了，博士（工学）．科学技術振興機構博士研究員，東京大学生産技術研究所特任助手を経て，2006年12月より九州大学特任准教授となり，現在に至る．コンピュータビジョン・知能ロボット

の研究に従事．2007年IEEE/ICRA'07 Best Vision Paper Awardを受賞．IEEEの会員．

Abstract This paper presents an approximate belief propagation algorithm that replaces outgoing messages from a node with the averaged outgoing message and propagates messages from a low resolution graph to the original graph hierarchically. The proposed method reduces the computational time by half or two-thirds and reduces the required amount of memory by 60% compared with the standard belief propagation algorithm when applied to an image. The proposed method was implemented on CPU and GPU, and was evaluated against Middlebury stereo benchmark dataset in comparison with the standard belief propagation algorithm. It is shown that the proposed method outperforms the other in terms of both the computational time and the required amount of memory with minor loss of accuracy.

Key words belief propagation, stereo, GPU