

再帰的計算に基づく3眼ステレオマッチングの VLSIアーキテクチャ

丹治 慶太 張山 昌論 亀山 充隆

東北大学大学院情報科学研究科 〒980-8579 宮城県仙台市青葉区荒巻字青葉 6-6-05

E-mail: tanji@kameyama.ecei.tohoku.ac.jp, {hariyama, kameyama}@ecei.tohoku.ac.jp

あらまし 3次元計測のためのステレオビジョンでは、画像間の対応を高速かつ高信頼に求めることが必要となる。対応点探索の手法として、SAD(Sum of Absolute Differences) 演算に基づく方法がある。SAD 演算に基づく対応点探索では、SAD 演算が膨大な回数行われるため計算量の減少が重要となる。本稿では、SAD 間の共通の中間結果に着目した再帰的 SAD を提案する。また、一般的な2眼ステレオビジョンでは、オクルージョンなどにおいて誤差が生じる問題がある。この問題に対し、基線の異なる3台のカメラ対を用いることで信頼性の向上を図る。3眼ステレオビジョンでは計算量が増加する問題があるが、提案する再帰的 SAD アルゴリズムを用いることで2眼ステレオビジョンから計算量の増加をわずかとできることを明らかにする。さらに、提案する3眼ステレオビジョンアルゴリズムのVLSI化を指向したプロセッサアーキテクチャを示す。

キーワード SAD(Sum of Absolute Differences), ステレオビジョン, スケジューリング

Architecture of a Stereo Matching VLSI Based on Recursive Computation

Keita TANJI, Masanori HARIYAMA, and Michitaka KAMEYAMA

Graduate School of Information Sciences, Tohoku University
6-6-05 Aoba, Aramaki, Aoba, Sendai, Miyagi, 980-8579 Japan

E-mail: tanji@kameyama.ecei.tohoku.ac.jp, {hariyama, kameyama}@ecei.tohoku.ac.jp

Abstract This paper presents a processor architecture for high-speed and reliable trinocular stereo matching based on recursive computation of SAD (Sum of Absolute Differences) computation. The major disadvantage of the trinocular stereo matching is its large computational amount that is double of binocular stereo matching in typical approaches. This is because two SAD results are required to find the depth of a single pixel. To solve this problem, the intermediate results to merge the two SAD results are fully exploited. Moreover, the interconnect-aware scheduling is proposed to obtain highly-parallel architecture.

Key words SAD(Sum of Absolute Differences), stereo vision, scheduling

1. ま え が き

知能ロボットや、高安全自動車などをリアルワールドで実現するためには、高速かつ高信頼に外部環境を認識する必要がある[1]。外部環境認識のための3次元計測の典型的な方法として、2台のカメラを用いて取得された画像間の対応点探索に基づくステレオビジョンがある。対応点探索の手法として、2枚の画像上にウィンドウを設け、ウィンドウ間で輝度値の絶対値差の和をとるSAD(Sum of Absolute Differences) 演算に基づく方法がよく知られている。SAD 演算に基づく方法では、SAD 演

算の計算量が膨大となる。また、一般的な2眼ステレオビジョンでは、エッジラインと平行となる類似領域や、オクルージョンにおいて誤差が生じる。

これらの問題を解決するために本稿では、計算量の少ない3眼ステレオマッチングとそのVLSIアーキテクチャを提案する。SADの中間結果を再帰的に利用し計算量を削減するアルゴリズムを提案するとともに、アルゴリズムが規則性・並列性を有することを述べ、VLSI化に適することを示す。本稿では従来多く行われている2眼ステレオビジョンから、カメラを1台増やすことで高信頼化を図った3眼ステレオビジョンを行う。基

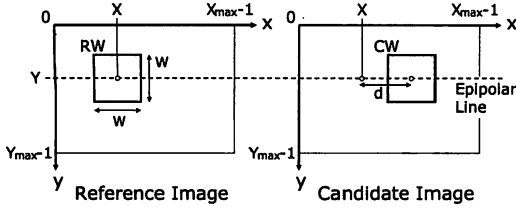


図 1: SAD 演算による 2 眼ステレオビジョン

線方向の異なるカメラ配置をとり信頼性が向上されることを示す。3 眼ステレオビジョンでは、情報量の増加とともに計算量が 2 倍となる問題に対し、提案する再帰的演算に基づく SAD アルゴリズムを用いることで、2 眼ステレオビジョンから計算量の増加をわずかとする。また、計算時間最小化を指向したスケジューリングに基づくプロセッサアーキテクチャを述べる。

2. 3 眼ステレオマッチングアルゴリズム

2.1 基本的な 2 眼ステレオマッチングアルゴリズム

ステレオビジョンは、図 1 に示すように、2 枚の画像間の対応付けを行った後に、三角測量を行うことにより、物体の 3 次元座標を求める処理である。対応点探索の典型的な手法として、SAD 演算を用いた方法がよく用いられる。SAD 演算に基づく対応点探索では、参照画像上の任意の参照点 (X_R, Y_R) を中心とするサイズ $W \times W$ の参照ウィンドウを設定する。2 台のカメラの光軸を並行にする典型的な配置を用いると、対応点は候補画像のエピポーラライン $Y = Y_R$ 上に存在することが保証される。そこで、候補画像上に、エピポーラライン上の候補点 $(X_R + d, Y_R)$ を中心とするサイズ $W \times W$ の候補ウィンドウを設定する。参照点と候補点の X 座標の差 d を視差と呼ぶ。このとき、参照点と候補点に対する SAD 演算は次式で与えられる。

$$SAD(X_R, Y_R, d) = \sum_{j=-\frac{W-1}{2}}^{\frac{W-1}{2}} \sum_{i=-\frac{W-1}{2}}^{\frac{W-1}{2}} AD(X_R + i, Y_R + j, d) \quad (1)$$

$$AD(X_R + i, Y_R + j, d) = |I_R(X_R + i, Y_R + j) - I_C(X_L + i + d, Y_L + j)| \quad (2)$$

ここで I_R, I_C はそれぞれ、参照画像と候補画像の輝度値である。SAD が小さいほどウィンドウ内の画像が類似していることを意味する。そこで、参照点とエピポーラライン上のすべての候補点に対して SAD を計算し、SAD 値が最小となる点を対応点と決定することにする。対応点探索においては、SAD 演算を膨大な数のウィンドウに対して行う必要があり、計算量が膨大となる。

2.2 共通の中間結果に着目した計算量削減

SAD 演算間での共通な中間結果を再帰的に利用することにより、計算量を削減することが可能である [2]。図 2 に、 3×3 のウィンドウサイズの例を示す。参照点 $(1, 1)$ を中心とする参照ウィンドウ RW_1 と、候補点 $(1, 1)$ を中心とする候補ウィン

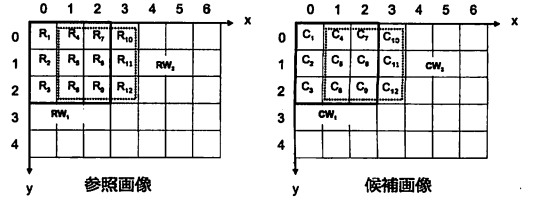


図 2: 共通の中間結果を有する SAD

ドウ CW_1 に対する SAD 演算 SAD_1 は、式 (1) を用いて、次のように与えられる。

$$SAD_1 = |R_1 - C_1| + |R_2 - C_2| + |R_3 - C_3| + |R_4 - C_4| + |R_5 - C_5| + |R_6 - C_6| + |R_7 - C_7| + |R_8 - C_8| + |R_9 - C_9| \quad (3)$$

同様に、参照点 $(2, 1)$ を中心とする参照ウィンドウ RW_2 と、候補点 $(2, 1)$ を中心とする候補ウィンドウ CW_2 の SAD 演算 SAD_2 は次式で与えられる。

$$SAD_2 = |R_4 - C_4| + |R_5 - C_5| + |R_6 - C_6| + |R_7 - C_7| + |R_8 - C_8| + |R_9 - C_9| + |R_{10} - C_{10}| + |R_{11} - C_{11}| + |R_{12} - C_{12}| \quad (4)$$

式 (4) と式 (5) には共通項 $\sum_{i=4}^9 (R_i - C_i)$ が存在する。したがって、 SAD_1 で共通項を演算した結果 (中間結果) を SAD_2 の演算の際に再利用すると計算量を削減できる。共通の演算が存在するのは、ウィンドウ間に重なりがあり、かつ各 SAD 演算の視差が等しいときとなる。視差が異なる SAD 演算間では、AD 演算への入力の組み合わせが異なるため共通の中間結果が存在しない。

2.3 計算量最小化を指向した VLSI 向きステレオマッチングアルゴリズム

視差 d を有する SAD の集合を S_d で表すことにする。 S_d は次式で与えられる。

$$S_d = \{SAD(X, Y, d) \mid 0 \leq X < X_{max}, 0 \leq Y < Y_{max}, 0 \leq d < d_{max}\} \quad (5)$$

各 SAD 演算は必ず S_d に含まれるため、各 SAD 演算に対して再帰的計算が適用できる。

2.2 節より、視差が異なる SAD には共通の中間結果がない、すなわち、データ依存性がない。そのため、視差が異なる SAD の集合 $S_0, S_1, \dots, S_{d_{max}}$ の要素はそれぞれ独立しており並列性が存在する。以下では、共通の中間結果を有する、すなわちデータ依存関係を有する SAD の集合 S_d に対して、計算量最小化を指向したアルゴリズムを説明する。

$SAD(X, Y, d)$ を計算するための中間結果として、参照ウィンドウと候補ウィンドウの右端の列の画素に対する部分的な SAD (以下、SLICE と呼ぶ) を定義する。

$$SLICE(X, Y, d)$$

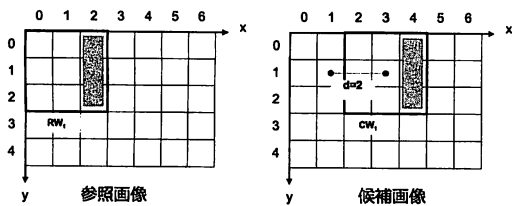


図 3: SLICE(1,1,2)

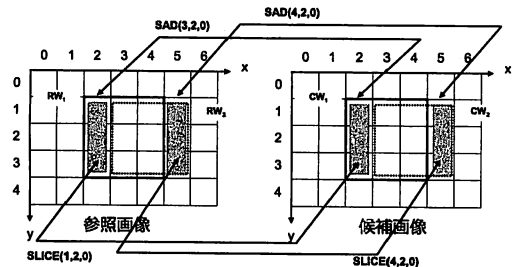


図 4: SAD 間における中間結果の再利用

$$= \sum_{j=-\frac{W-1}{2}}^{\frac{W-1}{2}} AD\left(X + \frac{W-1}{2}, Y + j, d\right) \quad (6)$$

図 3 は $W = 3, d = 2$ の場合の SLICE の例である。SAD(1,1,2) の計算に用いられる SLICE 部分を灰色で示している。

図 4 のようなウィンドウ RW_2 と CW_2 があるとき、その SAD 演算を行うことを考える。その際、すでに SLICE(1,2,0) と RW_1 と CW_1 の SAD 演算結果が求められているとすると、対象ウィンドウの SAD 演算の結果は以下の式で求めることができる。

$$\begin{aligned} SAD(4,2,0) \\ = SLICE(4,2,0) + SAD(3,2,0) - SLICE(1,2,0) \end{aligned} \quad (7)$$

さらに、図 5 の SLICE(4,2,0) を求めることを考える。その際、すでに SLICE(4,1,0) と $AD(5,0,0)$ の演算結果が求められているとすると、SLICE(4,2,0) の演算結果は以下の式で求めることができる。

$$\begin{aligned} SLICE(4,2,0) \\ = AD(5,3,0) + SLICE(4,1,0) - AD(5,0,0) \end{aligned} \quad (8)$$

以上、実例を用いて、中間結果を利用した SAD 演算の説明を行った。それらを一般化した式を式 (9) に示す。

$$\begin{aligned} SAD(X, Y, d) = SAD(X-1, Y, d) \\ - SLICE(X-W, Y, d) \end{aligned}$$

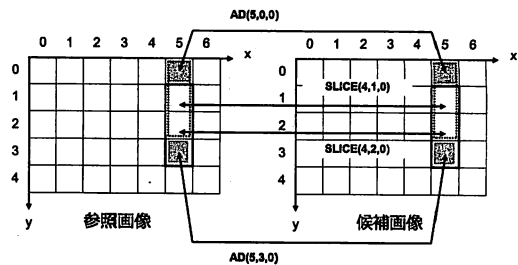


図 5: SLICE 間における中間結果の再利用

$$\begin{aligned} + SLICE(X, Y-1, d) \\ - AD\left(X + \frac{W-1}{2}, Y - \frac{W-1}{2} - 1, d\right) \\ + AD\left(X + \frac{W-1}{2}, Y + \frac{W-1}{2}, d\right) \end{aligned} \quad (9)$$

このような中間結果を利用した演算の利点は、計算量を削減できることである。すなわち、過去の SAD 演算の結果もしくはその一部を保存することにより、計算量の少ない効率的な SAD 演算を行うことが可能である。

2.4 3 眼ステレオマッチング

従来の 2 台のカメラを用いる 2 眼ステレオビジョンでは、オクルージョン（一方のカメラからしか見えない領域）が多く、誤対応の原因となる。本稿では図 6 のように 3 台のカメラを配置し、オクルージョンの減少を図った。

3 眼ステレオビジョンでは、左右対のカメラでのステレオマッチング、上下対のカメラでのステレオマッチングの結果を合わせて評価することで、より明瞭な SAD の最小値を検出し信頼性を高める。評価方法として、2 つの SAD の結果から真の最小値を検出するという方法が提案されている [3]。しかしながら、図 7 における左右対応付けのような場合、座標 $X \sim X'$ において SAD の極小値が多数あがってしまうため、誤対応となる可能性がある。本稿では、同一視差における SAD の和を評価関

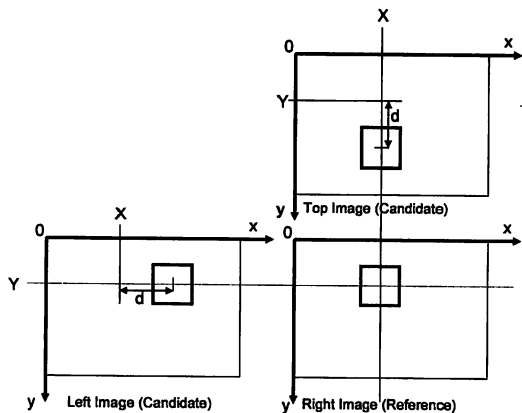


図 6: 3 眼ステレオビジョン

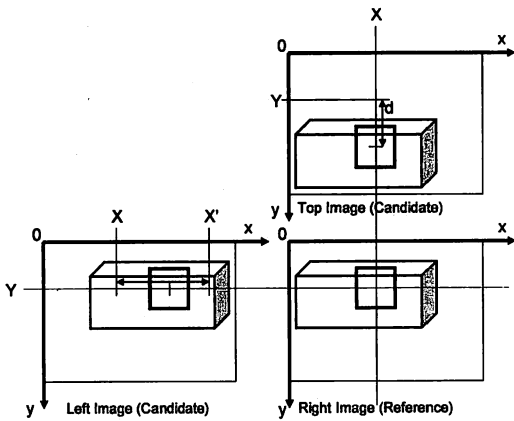


図 7: エピポーララインと平行な類似部分の対応付け

数とする。SAD の和で評価する場合、図 7 のように、左右の対応付けでは、輝度パターンが類似し SAD の極小部分が多くなる場合、上下対応付けにおいて SAD が単一の極小値をとることで、SAD の和の最小値はより明確となる。

3 眼ステレオビジョンは左右、上下それぞれのステレオマッチングを行うため 2 眼ステレオビジョンと比較して計算量が 2 倍になる。左右、上下のウィンドウ対に対する SAD の和 (以下 SAD2 とする) は次式で与えられる。

$$\begin{aligned}
 & SAD2(X, Y, d) \\
 = & \sum_{j=-\frac{W-1}{2}}^{\frac{W-1}{2}} \sum_{i=-\frac{W-1}{2}}^{\frac{W-1}{2}} |I_R(X+i, Y+j) \\
 & \quad - I_L(X+i+d, Y+j)| \\
 + & \sum_{j=-\frac{W-1}{2}}^{\frac{W-1}{2}} \sum_{i=-\frac{W-1}{2}}^{\frac{W-1}{2}} |I_R(X+i, Y+j) \\
 & \quad - I_T(X+i, Y+j+d)| \quad (10)
 \end{aligned}$$

ここで、 I_L, I_T はそれぞれ左候補画像、上候補画像の輝度値である。式 (10) は以下のように変形でき、式 (1) と対応する。

$$\begin{aligned}
 & SAD2(X, Y, d) \\
 = & \sum_{j=-\frac{W-1}{2}}^{\frac{W-1}{2}} \sum_{i=-\frac{W-1}{2}}^{\frac{W-1}{2}} AD2(X+i, Y+j, d) \quad (11)
 \end{aligned}$$

ここで、

$$\begin{aligned}
 & AD2(X, Y, d) \\
 = & |I_R(X+i, Y+j) - I_L(X+i+d, Y+j)| \\
 & + |I_R(X+i, Y+j) - I_T(X+i, Y+j+d)| \quad (12)
 \end{aligned}$$

式 (12) のように、1 画素の輝度の絶対値差の和を先にとることで、中間結果の生成過程を 2 眼ステレオマッチングと同様のも

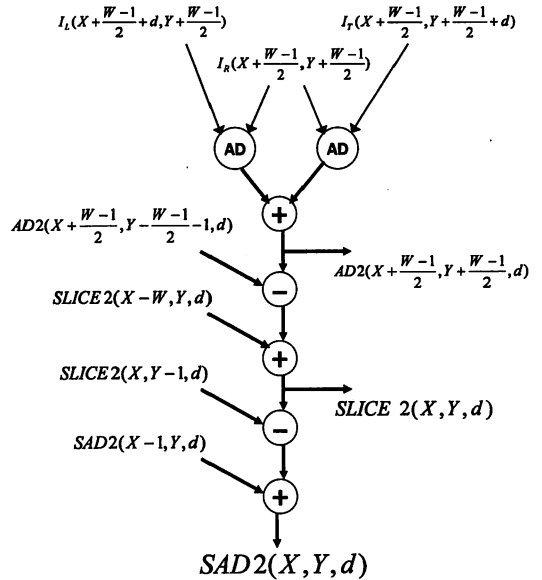


図 8: SAD2 計算の DFG

のとした再帰的 SAD 計算を適用する。式 (9) と対応するように書き換えると次式ようになる。

$$\begin{aligned}
 & SAD2(X, Y, d) \\
 = & SAD2(X-1, Y, d) - SLICE2(X-W, Y, d) \\
 & + SLICE2(X, Y-1, d) \\
 & - AD2(X+\frac{W-1}{2}, Y-\frac{W-1}{2}-1, d) \\
 & + AD2(X+\frac{W-1}{2}, Y+\frac{W-1}{2}-1, d) \quad (13)
 \end{aligned}$$

ここで

$$\begin{aligned}
 & SLICE(X, Y, d) \\
 = & \sum_{j=-\frac{W-1}{2}}^{\frac{W-1}{2}} AD2(X+\frac{W-1}{2}, Y+j, d) \quad (14)
 \end{aligned}$$

図 8 に SAD2 計算の DFG(Data Flow Graph) を示す。この DFG のように、SAD 計算一回は AD 演算 2 回、加減算 5 回で行われる。提案手法と、従来の 3 眼ステレオビジョンの手法の計算量の比較を表 1 に示す。提案手法と従来の手法において、SAD 演算回数は同じである。AD 演算 1 回の計算量は、ほぼ加算 1 回と同等とみなすことができ、従来の方式に対する提案方式の計算量の割合は、表 1 より、次式で与えられる。

$$\frac{7}{4W^2-1} \times 100 \quad (15)$$

表 1 計算量の評価

	Conventional	Proposed
Number of ADs per SAD	$2W^2$	2
Number of additions per SAD	$2(W^2-1)+1$	5

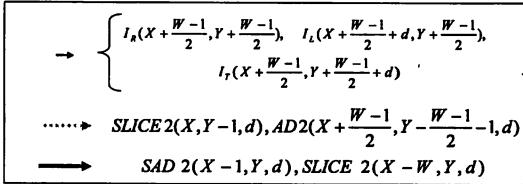
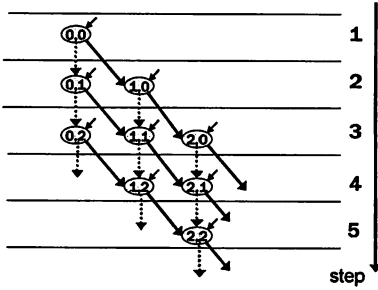


図 9: ASAP スケジューリングに基づく DFG

したがって、典型的なウィンドウサイズとして、 $W = 15$ である場合、全体の SAD 演算の計算量は、従来の 0.78 % 以下となる。

AD 演算の和を先にとることで、1 画素あたりの SAD 演算の計算量は 2 眼ステレオビジョンから、加算 2 回のみ増加で 3 眼ステレオビジョンの SAD を求めることができる。再帰的演算に基づく SAD 計算で左右、上下それぞれのステレオマッチングを行う場合、演算回数は 2 眼ステレオビジョンの 2 倍と加算 1 回となり 11 回となる。提案手法では演算回数は 7 回となり、64 % 以下の計算量となる。

3. プロセッサアーキテクチャ

以下では、ステレオビジョン VLSI プロセッサの主な構成要素であるステレオマッチングについて、提案アルゴリズムの演算部に関する概要を述べる。2.3 節で述べたように、同一視差の SAD 集合 S_d 間では並列性が存在する。よって S_d はそれぞれ並列に演算部で計算されるため、1 つの S_d を高速に処理することが重要となる。

演算部は複数個の処理要素 (PE) から構成される。 S_d の処理時間最小化を行うためには、 S_d 中の全 SAD をできるだけ早く計算する必要がある。SAD 演算間では中間結果の転送が必要であり、各 PE は中間結果を記憶するためのレジスタファイルおよび、式 (13) にしたがって SAD 演算を行うための演算器 (FU) から構成される。そこで、演算に必要なデータがそろった SAD 演算ノードをできるだけ早いステップで実行する ASAP (As Soon As Possible) スケジューリングを行う。これを図 9 に示す。ASAP スケジューリングの処理ステップは、実行可能な最小処理ステップ数であることが知られている。

図 10 に PE 内の演算ユニット (FU) の構成を示す。AD 演算は、全加算器で減算を行った後、各ビットの出力と最上位の符号ビットとの排他的論理和 (ExOR) をとることで実現される。

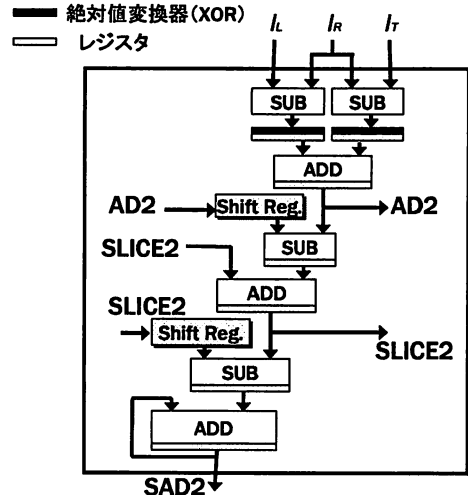


図 10: 演算ユニットの構成

演算部ではまず、左右方向 AD 演算、上下方向 AD 演算を並列に行う。その 2 つの加算と 4 個の中間結果を用いて加減算を行い、左右 SAD、上下 SAD の和を計算する。また、同時にそれぞれの過程で中間結果の生成を行う。各演算器間には、レジスタを挿入し演算器レベルでパイプラインを活用する。左右方向、上下方向 AD 演算を並列に行うことで、計算時間は 2 眼処理の場合と比較して加算 1 回分のみの増加となる。

4. むすび

SAD 演算の共通な中間結果に着目し、計算量の最小化を指向した VLSI 向きステレオマッチングアルゴリズムを提案した。また、3 台のカメラを逆 L の字型に配置することで従来の 2 眼ステレオビジョンで問題となる点を解消し、信頼性を向上できることを示した。さらに、提案した再帰的 SAD 計算アルゴリズムを用いることで、2 眼ステレオビジョンから計算量の増加をわずかとできることを示した。また、そのプロセッサアーキテクチャ概要を示した。

3 眼ステレオビジョンによる計算量の増加は解消されたものの、情報量の多さは変わらない。これにより、メモリ使用領域の増加は避けられない。今後は、最適なメモリアロケーションについて考察する必要がある。

参考文献

- [1] 亀山充隆, 藤岡与周, “ロボット用 VLSI プロセッサシステム”, 日本ロボット学会誌, vol.14, no.1, pp.22-25, 1996.
- [2] 三浦清志, 張山昌論, 亀山充隆 “再帰的計算に基づくステレオマッチングと VLSI 化”, 電気情報通信学会論文誌, No.8 pp.752-759, 2003.
- [3] 横山直人, 張山昌論, 亀山充隆 “最適スケジューリングに基づく 3 眼ステレオビジョン VLSI プロセッサの構成”, 電子情報通信学会技術研究報告, ICD2006-153, pp.55-60, 2006.