[山梨大学工学部研究報告 第42号 1991年12月]

論　　文

ＤＳＰによる輪郭抽出

小林謙一*, 稲葉政信*, 近藤芳人*, 大木　真*, 橋口住久*

（平成３年８月31日受理）

Extraction of Outlines of Pictures

by Kenichi KOBAYASHI*, Masanobu INABA*, Yoshihito KONDO*, Makoto OHKI*
and Sumihisa HASHIGUCHI*

Abstract

We trially manufactured a picture processing board extracting the outlines of pictures. A picture of 512×512 pixels is horizontally divided into five blocks, and each of which is smoothed and differentiated by 3×3 square masks. The mask operations are applied in parallel by using one DSP for each block. The total processing time of 2.12s mainly consists of the time necessary to transfer the picture data between memories.

１．はじめに

生産工程において画像処理を行うときには実時間処理に近い高速の処理が要求されるが，画像のデータ量が多いので，単純な画像処理であっても，実時間で行うことは容易でない．

ここでは，生産ラインにおける検査等に応用することを目標として，画像の輪郭抽出処理を行うハードウエアを試作し，処理時間について検討した．

２．処理システム

図２－１の輪郭抽出システムは，入力部，信号処理部，出力部で構成される．

入力部では，画素数５１２×５１２の画像をカメラから取り込み，各画素の輝度を８ビットにＡＤ変換する．信号処理部では，ＡＤ変換されたディジタル信号の平滑化と微分処理を行って，輪郭を抽出する．出力部では，輪郭抽出データをＤＡ変換してディスプレイに出力する．

* 電子情報工学科，Department of Electrical Engineering & Computer Science

この輪郭抽出システムについて，輪郭抽出のアルゴリズム，ハードウェアの構成，システムの処理時間について検討した．

３．抽出アルゴリズム

３．１　輪郭抽出

画像の輪郭を抽出するには，まず，元の画像を平滑化してノイズを除いてから，水平方向と垂直方向に微分して輪郭を生成する．平滑化と微分とは，それぞれ，図３－１のように３×３のマスク処理とし，画素ｆとマスク要素Ｍの畳み込みの演算

$$h(i,j)=\sum_{k=-1}^{1}\sum_{m=-1}^{1} f(i+k,j+m)\times M(k+1,m+1) \quad (3\text{-}1)$$

を行う．

すなわち，図３－１において，画像中の任意の３×３画素（f_{00}～f_{22}）からなる正方ブロックと３×３の処理マスク（M_{00}～M_{22}）との積和h_{11}を被処理ブロックの中央（f_{11}の位置）の画素の輝度とする．

３．２　平滑化マスク

元画像に含まれるノイズを低減するために平滑化を行う．平滑化は，１ブロック内の輝度の平均値をとる操作

$$h(i,j)=\frac{1}{3^2}\sum_{x=-1}^{1}\sum_{y=-1}^{1}f(i+x,j+y) \qquad (3\text{-}2)$$

であり，平滑化マスクは

$$\begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix}$$

である．

３．３　微分マスク

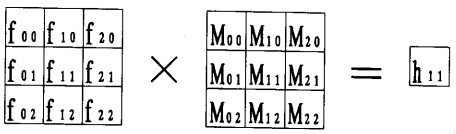
エッジや線を検出するために微分を行う．画像は離散値をとるので微分の代わりに差分を用いる．微分は方向性を持たない２次元ラプラシアン

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \qquad (3\text{-}3)$$

入力部　信号処理部　出力部

図２－１　画像の輪郭抽出システム

Fig.2-1　Blockdiagram of the system for extraction of the outlines of pictures



被処理ブロック　処理マスク　処理結果

図３－１　マスク処理

Fig.3-1　Mask operation

を用いて行う．

微分の代わりに差分

$$\Delta_x^2 f = f(i+1,j)+f(i-1,j)-2f(i,j) \qquad (3\text{-}4)$$

$$\Delta_y^2 f = f(i,j+1)+f(i,j-1)-2f(i,j) \qquad (3\text{-}5)$$

を用いると，ラプラシアンは

$$\nabla^2 f = f(i+1,j) + f(i-1,j) + f(i,j+1) + f(i,j-1) - 4f(i,j) \qquad (3\text{-}6)$$

とかける．したがって，微分マスクは

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

となる．なお，このマトリックスの符号は式(3-6)の係数とは逆符号にしてあるが微分の効果は同じである．

３．４　マスク処理

図３－２は，（ｍ×ｎ）画素の対象画像にマスク処理を加える手順である．まず，左上の３×３の画素｛（０，０）～（２，２）｝にマスク処理を行い，その結果を（１，１）に書き込む．次に，マスクを１画素右に移動して｛（１，０）～（３，２）｝にマスク処理を行い，結果を（２，１）に書き込む．マスクを右に１画素づつ移動させｍ番目の列のマスク処理が終わるまでこの操作を（ｍ－２）回行う．次に，マスクを縦方向に１ライン下にさげ左端から右方向に同様の処理を行う．このような操作をｎライン目まで行う．これらの処理を行うとマスク処理されたデータは，元の画像の上下左右の各１画素分ずつが失われ，得られる画像は，（ｍ－２）×（ｎ－２）画素となる．

平滑マスクと微分マスクを結合すると５×５の連結マスクが得られる．５×５画素の１ブロックに５×５の連結マスク処理を加えるときの演算回数は，乗算２５回，累算２５回である．５１２×５１２画素の１フレーム全体にわたる演算回数は，乗算と累算それぞれについて，

$$25\times(512-4)^2=6,451,600 \text{ 回}$$

である．一方，３×３のマスク処理を２回行う場合には，乗算と累算それぞれについて，

$$9\times(512-4)^2+9\times(510-2)^2=4,663,476 \text{ 回}$$

であり，連結マスクを使うより演算回数が少ないので

４．２　各ユニットの構成

各ユニットは，図４－１に示すように，ＣＰＵ周辺回路とＤＳＰ周辺回路とからなる．

４．２．１　ＣＰＵ周辺回路

ＣＰＵ周辺回路は，ＣＰＵ，ＲＯＭ，ＲＡＭ，ＡＤ，ＤＡ，ＦＭ１，ＦＭ２で構成される．

ＣＰＵは，システム全体の制御，データ転送を行う．ＲＯＭには，ＣＰＵのベクタ，ＣＰＵプログラムを書き込んである．ＲＡＭは，各種ベクタ，ワークエリア用である．ＦＭ１の容量としては，

５１２×５１２＝２６２，１４４（バイト）

必要であるので，１３１，０７２×８ビットのＳＲＡＭを２個用いた．

一方，マスク処理により元画像の上下左右２ラインづつが失われるので，ＦＭ２の容量としては，

５０８×５０８＝２５８，０６４（バイト）

あればよい．ＦＭ１と同様に１３１，０７２×８ビットのＳＲＡＭを２個用いる．

４．２．２　ＤＳＰ周辺回路

ＤＳＰ周辺回路は，ＤＳＰ，ＬＭ，ＲＯＭより構成される．ＬＭ１～４には，

５１２×１０６＝５４，２７２（バイト）

を格納する．またＬＭ５には，

５１２×１０４＝５３，２４８（バイト）

を格納する．ＬＭには，６４ｋバイトのＳＲＡＭを１個ずつ使用する．ＤＳＰとのデータのやりとりを速くするために，ＬＭは高速であることが必要である．ＲＯＭには，ＤＳＰのプログラムおよびマスクデータを格納する．

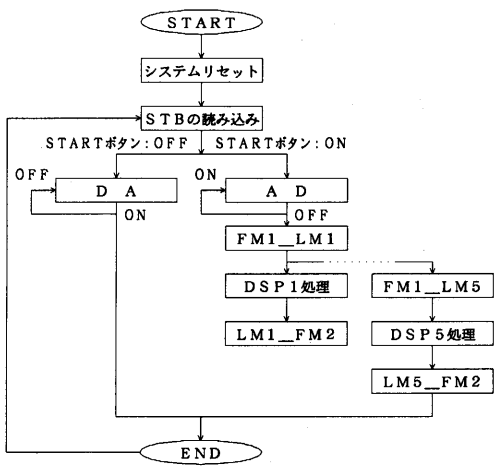


図４－２　マスク処理のフロチャート

Fig.4-2 Flowchart of mask operation

５．処理時間

５．１　試作システムの処理時間

図５－１は，各ＤＳＰにおいて輪郭抽出処理に要する時間である．全処理時間Ｔは，ＦＭ１からＬＭ１へのデータ転送開始時から，ＬＭ５からＦＭ２への転送が完了するまでの時間であり

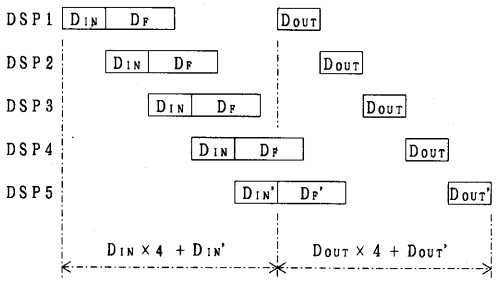


図５－１　処理時間

Fig.5-1 Processing time

表５－１　処理時間（結果）

Table 5-1 Processing time (result)

データ転送時間（FM1→LM1～4）	DIN	224ms
データ転送時間（FM1→LM5）	DIN'	220ms
データ転送時間（LM1～4→FM2）	DOUT	202ms
データ転送時間（LM5→FM2）	DOUT'	198ms
輪郭抽出時間　（DSP1～4）	DF	334ms
輪郭抽出時間　（DSP5）	DF'	326ms
全体の処理時間	T	2.12s

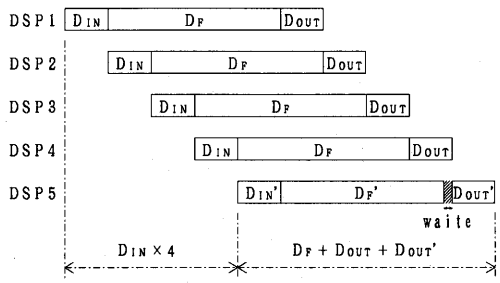


図５－２　改良システムの処理時間

Fig.5-2 Processing time of the improved version

高速処理ができることになる．したがって，ここでは３×３のマスク処理を２回行う方を採用した．

いずれの方法をとっても，最終的に得られる画素数は，上下左右各２画素分ずつが失われる．

３．５　並列処理

計算時間を短縮するために積和演算を高速に行うＤＳＰを用いる．周辺機器等の要求から，ＤＳＰとしてモトローラＤＳＰ５６００１を用いた．ＤＳＰ５６００１のメモリ容量は６５，４７２バイトであり，５１２×５１２画素分のデータを一度に取り込むことができないので，１フレームを５つのブロックに分割し，それぞれを５個のＤＳＰによって並列処理する．図３－３のように，ＤＳＰ１～４に５１２画素×１０６ライン分，ＤＳＰ５に５１２画素×１０４ライン分のデータを担当させる．ここで，４ラインづつ重複しているのは，平滑マスク処理と微分マスク処理によって上下２ラインづつ画像が失われるのを補うためである．

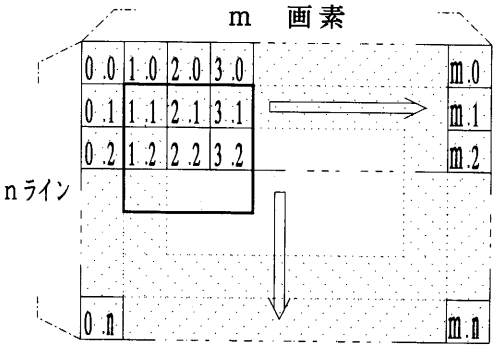


図３－２　マスク処理の進み方

Fig.3-2　Procedure of mask operation

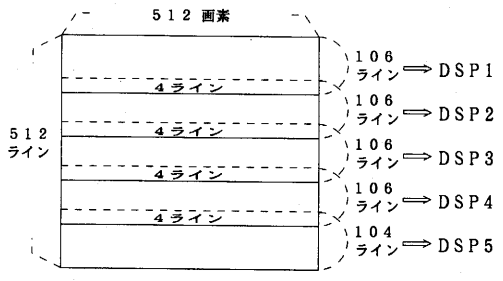


図３－３　画像データの分け方

Fig.3-3　Block division of an image

４．ハードウェア

４．１　信号処理部の構成

図４－１は信号処理部のブロック図，図４－２は処理の流れである．

図４－１においてＣＰＵ（ＭＣ６８０００，８Ｍｚ）はハードウェアの管理を行う．ＤＳＰ１～５（ＤＳＰ５６００１，２０．５Ｍｚ）はマスク処理を行う．データバスは，ＣＰＵ側が１６ビットバス，ＤＳＰ側が８ビットバスである．ここで，ＣＰＵ側が１６ビットバスとしてあるのは，ＡＤからＦＭ１へとＦＭ２からＤＡへのデータ転送を高速かつ確実に行うためであり，ＤＳＰ側が８ビットバスとしてあるのは，画像データが８ビットであるためである．

カメラからの画像信号をＡＤコンバータで５１２×５１２画素８ビットに量子化し，フレームメモリＦＭ１に書き込む．ＦＭ１の画像データを図３－３のように５つに分割し，バススイッチ（ＳＷ１～５）を経由してＤＳＰの外部メモリ（ローカルメモリＬＭ１～５）に転送する．ローカルメモリのデータに各ＤＳＰにより平滑マスク演算と微分マスク演算を行って輪郭抽出し，結果をＬＭ１～５に記録する．バススイッチＳＷ１～５を経由してＬＭ１～５の処理済データをフレームメモリＦＭ２に転送する．ＦＭ２のデータをＤＡ変換して出力し，ディスプレイに表示する．

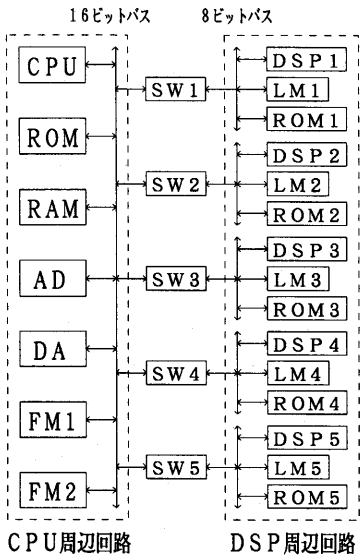


図４－１　信号処理部のブロック図

Fig.4-1　Blockdiagram of the processing unit

$$T = 4D_{IN} + D_{IN'} + 4D_{OUT} + D_{OUT'} \quad (4\text{-}1)$$

である．各処理時間は表５－１の通りであるので，全体の処理時間Ｔ＝２．１２ｓとなる．

５．２　処理時間の短縮

試作システムでは，デバッグ等の容易さを考慮してＣＰＵが８ＭＨｚ，ＤＳＰが２０．５ＭＨｚとクロック周波数を低めに選んである．

試作システムの問題点は，ＣＰＵ側が１６ビットバス，ＤＳＰ側が８ビットバスとしてあるのでデータ転送がスムースではなく，ＣＰＵに負担がかかりすぎていることである．そのため，輪郭抽出処理よりもデータ転送に時間がかかっており，全体の効率が悪くなっている．効率改善のためには，ＤＳＰ側のバスを１６ビットにしてＣＰＵの負担を軽くする必要がある．図５－２のようにデータ転送時間と輪郭抽出時間とのバランスがとれるようにすれば，効率の良いシステムとなり全体の処理時間も短縮できる．また，ＣＰＵにＭＣ６８０００（１２ＭＨｚ），ＤＳＰにＤＳＰ５６００１（２７ＭＨｚ）を使用すれば，処理時間はさらに短縮する．

６．おわりに

画像の輪郭抽出処理を行うために，複数のＤＳＰを用いた輪郭抽出ボードの試作を行った．

ＣＰＵにＭＣ６８０００（８ＭＨｚ），ＤＳＰにＤＳＰ５６００１（２０．５ＭＨｚ）を用いたときの処理時間は，２．１２ｓであった．しかし，生産工程において画像処理を行う場合には，実時間（約３００ｍｓ）で輪郭抽出処理を行う必要がある．そこで，ＤＳＰ側のバスを１６ビットバスにしてデータ転送を高速に行い，クロック周波数の高いＣＰＵとＤＳＰを用いて全体の処理能力を向上さると処理時間がさらに短縮できる．また，実時間処理実現のためにＡＤ，ＤＡとＬＭとのインターフェースをダイレクトに行える構成を検討中である．

参考文献

１）近藤芳人ほか:"ＤＳＰを用いた画像の輪郭抽出", 電子情報通信学会春期全国大会, D-522, (1990)

２）稲葉政信ほか:"ＤＳＰによる画像の輪郭抽出", 電子情報通信学会春期全国大会, D-614, (1991)

３）M68000 FAMILY REFERENCE, Motorola, Inc., (1988)

４）M68000　マイクロプロセッサ　ユーザーズ・マニュアル, ＣＱ出版社, (1989)

５）細田　誠:"68000 系ハードウェア設計ガイド", PP. 8-55, ＣＱ出版社, (1990)

６）DSP56000/DSP56001 Digital Signal Processor User's Manual, REV. 1, Motorola, Inc.

７）仁戸田一之，藤原淑恵:"DSP56000/56001の概要と応用", 別冊インターフェース　ＤＳＰを使いこなす, PP. 181-189, ＣＱ出版社, (1989)