PCIバス転送型フレーム・グラバを用いた 画像取り込みの実際*

榊原 潤**

Digital Image Acquisition by PCI Bus Frame Grabbers

Jun SAKAKIBARA

1. はじめに

コンピュータへの画像の取り込みは、今日では可視化計測を行うに当たって欠くことのできないプロセスであり、なるべく大量の画像を高速かつ高品位に獲得することが望まれる。そのための画像取り込み装置として、数年前までは単体の画像処理装置なるものが市場に多く見られた。しかし、近年のパソコン(PC)の高性能化に伴って、画像取り込み(フレーム・グラバ)ボードを拡張スロットに装着したPCで画像を取得し、自作あるいは既成の多種多様なソフトウェアをPC上で実行して画像処理を施す方法が最近の主流となっている。これは、プログラムの改造や更新が容易であり、単体の画像処理装置に比べて自由度が高いためである。また、CPUとしてPentiumを搭載したPCであれば従来の画像処理専用ハードウェアにまさる性能を有することもその理由であろう。本稿ではフレーム・グラバ・ボード、特に最近のPCIバス転送型のボードを用いてPCに画像を取り込む方法を具体例を交えて解説する。さらに最新のプログレッシブ・スキャン・カメラによる高速画像取り込みの方法についても触れる。

2. PCIバス転送型フレーム・グラバのハードウェア

2.1 PCIバスのアーキテクチャ

PCIバス転送型フレーム・グラバについて述べる前に、PCIバス自体について説明しておく。PCI (Peripheral Component Interconnect) バスは、PC内部の各デバイス間の超高速データ転送を実現するための標準規格であり、1991年10月にその最初の素案が提示されたのち、業界各社の共同作業によって標準規格の策定が行われ、1994年6月までに第2.1版のほぼ完全な仕様が完成している¹⁾。1995年前後にインテル社のPentiumプロセッサがPC用CPUの主流として普及したのを契機に、PCIバスはPCを構成する中核のバスとして完全にその地位を確保した。PCIバスはCPU規格に依存しないた

め汎用性が高く将来のCPUアーキテクチャにも対応できる。さらに、従来型PCの標準バスであるISAバスのバス幅が8ビットを許容する16ビットであるのに対し、PCIバスは完全に32ビットであり、ISAバスで必要であったバス幅の切り替えなどの複雑な動作が簡略化された。このため、最大で132 Mバイト/秒の転送レートを可能とする (ISAバスでは8 Mバイト/秒)。

Fig. 1にPCIバスをベースとしたPCのシステム構成図を示す。PCIバス・システムで中核となるのが図中央にあるPCIチップで、PCIバスの制御を行うと同時にCPUバスであるホスト・バスとPCIバスのインターフェースの役割も果たしている。メイン・メモリもPCIチップにより管理されており、フレーム・グラバなどの

Fig. 1 PCIバスをベースにしたPCシステムの内部構成。CPUはPCIチップを介してメイン・メモリやフレーム・グラバなどの周辺機器とアクセスする。

* 原稿受付 1997年10月1日

** 正会員 筑波大学構造工学会

PCI デバイス間やメイン・メモリ間のデータ転送は PCI チップによって行われる。このように、プログラム命令やデータの格納されているメイン・メモリと PCI デバイスは直結していないため、PCI デバイス間で転送を行いながら CPU は他の命令を実行することも可能である。なお、従来の ISA バスは PCI バスからブリッジ (PIIX) を介して接続されており、従来通り使用することができる。

2.2 PCIバス転送型フレーム・グラバ

従来のフレーム・グラバ・ボードは、そのボード上に画像を記憶するためのメモリ (オン・ボード・メモリ) を搭載してあるものが一般的であり、カメラなどから入力される画像は一度オン・ボード・メモリに取り込まれたのち、PC 側から ISA バスなどの I/O を経由してアクセスすることとなる (Fig. 2 参照)。ISA バスの転送速度は低く、リアル・タイムで画像をコンピュータへ転送してディスプレイ表示することは不可能であるため、入力された画像は D/A コンバータを通して TV モニターで表示する。また、可視化計測を行う場合、物理現象の時間変化を捉えるために連続した大量の画像を取り込むことがしばしば要求されるが、一度に連続して取り込める枚数はオン・ボード・メモリの容量に依存するため、その枚数を増やしたければ搭載メモリの多いボードを選ぶことになる。しかし、こうしたボードは近年の DRAM の値段に比してかなり高額なものとなることが多い。そのため、連続画像取り込みを行う場合には画像を一旦 VTR やレーザー・ディスクなどの媒体に記録した後、再生画像を逐次コンピュータへ取り込むという方法が一般的であった²⁾。

Fig. 2 オン・ボード・メモリ内蔵フレーム・グラバと PCI バス転送型フレーム・グラバの構成。PCI バス転送型は オン・ボード・メモリが不要なため、回路構成はシンプルである。

ところが、最近では PC のホスト・メモリの価格が急激に低下しており (128 MB で 8 万円程度)、また、PCI バスによるデータの超高速転送が可能になってきたため、入力した画像を A/D 変換した後、直ちに PC のホスト・メモリに転送する、いわゆる PCI バス転送型フレーム・グラバが各社から発売されはじめている。これは 1 KB 程度のバッファ・メモリを有するものの、基本的にオン・ボード・メモリを持たないため、回路構成が単純・安価であり、大量の連続画像取り込みも PC のホスト・メモリさえ増設すれば極めて容易に行うことができる。さらに、入力された画像は PCI バスを経由してグラフィック・カードへ転送することができるため、それを PC のモニター上にリアルタイムで表示することが可能である。

PCI バス転送型フレーム・グラバを用いて大量の画像を取り込むことの利点を挙げるとすれば、(1)取り込んだ後の画像のアクセスが容易であること。これは VTR やレーザーディスクなどの外部記憶システムに画像が格納される場合に比べて、画像が直接メインメモリに格納されるので各画像のフレーム管理や処理が容易である。(2)メカニカルな部分がないのでタイミング・ジッタが少ない。VTR を用いて大量の画像を記録する方法では水平走査線の開始位置が回転ヘッドの回転ムラに影響されるが、フレーム・グラバに画像を直接入力する場合にはそうした影響を排除できる。(3)画質は最良。VTR などの記録メディアを通さずして取り込めるため画質は極めて良好である。(4)オン・ボード・メモリを持たないため安価である。(5)後述するプログレッシブ・スキャン・カメラのように複数チャンネルの出力をもつカメラを接続することができ、タイミング・ジッタなしで同時取り込みが可能である。一方、欠点は(1)長時間連続録画は困難。一般的な PC ではメイン・メモリを最大で 256 MB までしか実装できないため、連続記録可能な画像は 800 枚程度*であり、30 フレーム/毎秒のフレーム・レートで約 27 秒間しか録画できない。(2)最大転送速度が PCI チップに依存する。後述のように転送レートが PCI チップに依存するため、適当な PCI チップを搭載した PC システムを購入する必要がある。

Table 1 に現在 (1997 年 9 月) 入手可能な PCI バス転送型フレーム・グラバの一部を示す。

2.3 PCIチップ・セット

上述の PCI チップ・セットとしては様々なものが市場に存在している (Table 2) が、これらすべてが十分な性能を有するとは限らない。Pentium 用の Orion チップセットは当初予定されていた性能が保証されず、10 MB/秒程度の転送速度しか実現されていない。Mercury や Neptune は Pentium 100 MHz 以前にしばしば用いられてきたが、一部の画像取り込みボードは完全な動作が保証されず、最近では Triton II (430HX) が一般的であ

* 画像の大きさを 640 × 480 ピクセルとし、OS に必要なメモリ領域 (25 Mb) を差し引いた場合

Table 1 入手可能な PCI 転送型フレーム・グラバの一部

型番	製造元	画像サイズ	同時入力チャネル数	定価	国内問い合わせ先	備考
DT3155	Data Translation	640x480 8ビット	1ch RS-170	198,000 円	(株)リンクス・コーポレーション TEL:03-5489-3871	
DT3154	"	640x480 24ビット	3ch (RGB) RS-170	未定	"	
IC/PCI	Imaging Technology	640x480 24ビット	3ch (R,G,B) ^{注1)} RS-170	630,000 円	伊藤忠テクノサービス(株) TEL:03-3419-9193	オン・ボード・メモリ付き
GENESIS-LC	Matrox 社	64Kx64K (最大)	4ch, RS-170 その他	630,000 円	住金制御エンジニアリング(株) TEL:03-3816-8611	オン・ボード・メモリ、外部表示機能付き
METEOR/RGB	"	640x480 24bit	3ch (R,G,B) ^{注1)} RS-170	148,000 円	"	
PCAPCI	カノプス(株)	640x480 24ビット	1ch(コンボ・ジット) NTSC	98,000 円	カノプス(株) TEL:078-992-6830	
FDM-PCI II	(株)フォトン	640x480 24ビット	1ch(コンボ・ジット) ^{注1)} NTSC	98,000 円	(株)フォトン TEL:03-3486-3480	

注1) VTR 出力には同期できない。

Table 2 PCI チップ・セットの一覧

適合 CPU	PCI チップ・セット型番	別名	備考 (最大使用可能メモリ量等)
i80486 用	425EX/426EX	Aries	
	423TX/424ZX/378SB	Saturn II	
Pentium 用	430LX	Mercury	
	430NX	Neptune	DualCPU
	430FX	Triton	128MB
	430VX	Triton VX	128MB
	430HX	Triton II	512MB, DualCPU
	430TX	通称 TX	256MB
Pentium Pro	450KX	Orion	
	450GX		Quad CPU
	440FX	Natoma	1024MB, DualCPU
Pentium Pro, Pentium II 用	440LX		1024MB, DualCPU AGP 対応 ²⁾
	440BX (未発売)		

²⁾AGP は、Accelerated Graphic Bus で、グラフィックボードを PCI バスに挿すのではなく、専用の AGP バスに挿すことによって、PCI バスの帯域 (133 MB/sec; 32 bit PCI) をあけようというもので、AGP は現在 266 MB/sec, 2 倍, 4 倍モードでは 532 MB/sec, 1064 MB/sec の帯域を確保する。このバスはチップから直接出ている。

る。Pentium Pro 用の 450 KX, GX は第 1 世代のチップセットであり、特に GX は 4-CPU, 4-way メモリーインターリーブをサポートするが、チップ数が多い(6~8 個)ためにスループットが悪く一般的には広まっていない。PCI バス転送型画像取り込みボードを用いるときは、CPU が Pentium の場合には Triton を、Pentium Pro の場合には Natoma を用いた PC を選択する必要がある。

3. 一般的 (RS-170 方式) な白黒 CCD カメラを用いた画像取り込み

ここでは、家庭用テレビのビデオ信号規格として最も普及している NTSC 規格のモノクロ版である RS-170 規格に準拠した白黒 CCD カメラ SONY XC-75 を Data Translation 社製フレーム・グラバ DT 3155 に接続し、画像の取り込みを行う方法をそのソフトウェアを中心に解説する。

3.1 ハードウェア構成

使用したハードウェアの一覧を Table 3 に示す。各機器の結線は至極単純で、CCD カメラの映像出力をフレーム・グラバへ直接入力した。PC は各部品を個別に購入して組み立てたものである。なお、ここで用いた SCSI インターフェース (AHA 1510 A, Adaptec) はフレーム・グラバ DT 3155 と共用することができなかった。おそらく、SCSI インターフェースがプラグアンドプレイ対応でない旧式のものであったためと思われる。

3.2 ソフトウェア構成

フレーム・グラバ制御用のソフトウェアとして既成品も販売されているが、より自由度が高く特殊な用途にも応用できるように、ここでは C 言語によって画像取り込みプログラムを開発することを考える。Data Translation ではユーザが独自の画像取り込み用ソフトウェアを開発できるように、フレーム・グラバ用のソフトウェ

Table 3 使用ハードウェア一覧

品名	型番	製造元	問い合わせ先	備考
フレーム・グラバ	DT3155	Data Translation	㈱リンクス・コーポレーション TEL:03-5489-3871	
CCD カメラ	XC-75	SONY	東通産業㈱	
マザー・ボード(PC)	P/I-P6NP5	ASUS	㈱ソートップ	PCI チップ:440FX メイン・メモリ:64MB
CPU	PentiumPro, 150MHz	Intel	〃	
SCSIインターフェース	AHA-1510A	Adaptec	〃	

開発キット (FG SDK) をライブラリの形で供給している。ここで紹介するプログラムは Data Translation FG SDK にサンプルとして添付されているソース・コードの ack2hst.c を改造したものであり、主要ソース・コード (Table 4) は著者のホーム・ページ (<http://www.kz.tsu-kuba.ac.jp/~sakakiba/dt3155.html>) に公開してある。なお、OS は Microsoft Windows NT 4.0, コンパイラは Microsoft Visual C/C++ 4.0 を使用した。

メイン・プログラムである `ack2hst.c` には Windows アプリケーションとして機能するために `WinMain()` や `MainWndProc()` などのメッセージ・ループに加えて、フレーム・グラバの初期化サブルーチン `InitDevice()` が記述されている。`InitDevice()` では `OIImgOpenDevice()` によってフレーム・グラバのデバイス・ドライバをオープンし、ハンドルを得る。つぎに `OIImgQueryDeviceCaps()`、`OIFgQueryInputCaps()` などの関数でフレーム・グラバが実行できる機能についての情報を得る。これは、Data Translation が供給している他のフレーム・グラバを同一のソフトウェアで等価に操作するためである。さらに、入力用ロック・アップ・テーブルを初期化するために `OIFgWriteInputLUT()`、`OIFgSetInputControlValue()` を呼んで初期化を終了する。

画像の取り込みを行う部分は別ファイルの `player.c` に記述してある。この中で、`PlayerEnter()` は `OIFgAllocateBuiltInFrame()` を呼ぶことで、外部から入力された画像がストアされる領域（フレーム領域）をデバイス・メモリー上に確保する。デバイス・メモリーは DT 3155 デバイス・ドライバが PC 起動時に自動的に確保する領域であり、また、フレーム領域は PC 起動後にユーザがデバイス・メモリー上に確保するものである。Fig. 3 にデバイス・メモリー、フレーム領域、ユーザ・アプリケーションのデータ領域、およびそれらの間でのデータ転送に用いる関数名を示す。フレーム・グラバによってデジタイズされた画像はフレーム領域に転送され、これをユーザ・データ・エリアへ再度 `memcpy()` などを用いて転送して画像処理やファイル・セーブなどを行う。`PlayerEnter()` ではフレーム領域を 1 フレーム (640 × 480 ピクセル) のみ確保したのち、`GlobalAlloc()` によってユーザ・データ・エリアに複数枚分の画像をストアする部分を確保する。実際に画像を取り込む直前に呼び出

Table 4 DT 3155 による連続画像取り込みのための主要プログラム群

ソース・ファイル名	内 容
ACQ2HST.C	メイン・プログラム、デバイス初期化等
player.c	画像取り込み、表示
save.c	画像のファイルへのロード、セーブ
pref.obj	セットアップファイルのロード、セーブ
jprog.c	各種サブルーチン
PASSTHRU.C	入力画像のリアルタイム表示
FRAMESIZ.C	フレーム・グラバの各種設定値の変更等
A2D.C	
VIDEOSEL.C	
SYNCSNT.C	
VIDEOSIZ.C	
PIXELCLK.C	
INP_SRC.C	
INP_FILT.C	
TIMEOUT.C	
FRAMETYP.C	
UTILS.C	
ACQ2HST.res	リソース・ファイル

Fig. 3 ホスト・メモリにおけるデバイス・メモリと画像の転送されるメモリ領域(フレーム)の配置.

される RecordImages() では、OIFgAsyncAcquireFrameToDevice() によってフレーム領域への画像取り込みが指示された直後に、前回にフレーム領域へ取り込まれた画像を memcpy() を用いてユーザ・データ・エリアへ転送する。memcpy() によるデータ転送レートは 50 MB/s 程度であり、画像取り込み転送レートの 10 MB/s に比べて高速である。そのため、新しい画像がフレーム領域へ流れ込んで来るときには既に memcpy() によって前回の画像はユーザ・データ・エリアへ転送されている。この作業を 1/30 秒ごとに繰り返すことで時間的に連続した画像をユーザ・データ・メモリへ転送することが可能となる。

原理的にはフレーム・グラバから送られてくる画像を直接ユーザ・データ・メモリへ転送することも可能ではあるが、そのための関数は Data Translation FG SDK には用意されていない。

3.3 ビデオ信号について

実際の画像取り込み例を紹介する前に、NTSC あるいはその白黒版の RS-170 映像信号のタイミングについてふれておく。カメラの撮像素子である CCD は光を電荷に変換する素子（画素）が Fig. 4 の様に昇目状に配置されたものであり、蓄積された電荷は一定期間間隔で映像信号として出力される（Fig. 5）。ここで、NTSC（あるいは RS-170）では偶数ラインと奇数ラインに蓄積された電荷の出力は別々に行われる（インターレース）。すなわち、Fig. 5 において(1)の期間（1/30 秒間）に偶数ラインの画素群に蓄えられた（すなわち被写体を見た）電荷は、蓄積終了後 1/60 秒間かけて(1)' の期間に映像信号としてカメラからフレーム・グラバへ出力される。このとき、奇数ラインの画素群は電荷蓄積を行っており(2)、偶数ラインの映像転送が終了した時点で、こんどは奇数ラインの電荷が転送される(2)'。こうして送られてくる映像信号はフレーム・グラバによって偶数、奇数ラインを識別されてメモリ上へ格納される。このように、偶数ラインと奇数ラインでは被写体を見はじめる時間が異なることを知っておく必要がある。なお、偶数ラインあるいは奇数ラインのみで構成される画像をフィールド画像、両者全体で構成される画像をフレーム画像と称し、電荷蓄積時間が上記のように 1/30 秒間の場合を「フレーム蓄積モード」、一方、各フィールドの後半 1/60 秒間のみに蓄積を行う場合を「フィールド蓄積モード」と一般的に呼ぶ。

3.4 画像の取り込み例

それでは、実際に画像の取り込みを行った例を紹介しよう。Fig. 6 に示すように、黒色の点を描いた白色紙を等速回転するレコード・プレーヤにのせ、日光を照明として撮影した。このとき、CCD カメラはフレーム蓄積モードで電子シャッターは無しである。Fig. 7 (a)(b)は連続する 2 フレームの画像である。画像下側に回転中心があるので点は左右に“おれ”ている。Fig. 7 (b)の“おれ”ている部分の拡大図を Fig. 8 に示す。前節で述べたように、奇数ラインと偶数ラインで“見はじめる時間”が異なるため、2 時刻の画像が交互に合わり縞状の画像となる。また、中央付近では奇数ラインと偶数ラインの“おれた直線”が互いにオーバーラップしている。この部分は Fig. 5 において(1)と(2)が重複している部分に相当する。

つぎに、Fig. 7 に示した画像を偶数ラインと奇数ラインを別々の画像（フィールド画像）に分離し、縦方向を補間したものを Fig. 9 に示す。縦方向が補間されているため、画像の解像度はフレーム画像（640×480 ピクセル）の半分（640×240 ピクセル）である。Fig. 9 (a)(c)は偶数、Fig. 9 (b)(d)は奇数フィールドである。(a)から(d)に

Fig. 4 CCD 素子の画素配列。

Fig. 5 電荷蓄積期間（被写体を見ている期間）と蓄積した電荷を映像信号として転送する期間。奇数ラインと偶数ラインではそれぞれ被写体を“見はじめる”時間が異なる（インターレース方式）。

照明（太陽光、ちらずきが無くても良い）

Fig. 6 「画像取り込み例」のための撮影装置。黒い点を描いた白色紙を回転円盤（レコード・プレーヤ）にのせ、連続光（太陽光）を照明として CCD カメラ（フレーム蓄積モード、電子シャッターなし）で撮影、フレーム・グラバへ取り込んだ。

かけて点が左側へ移動し、各点は互いにその幅の 1/2 ずつオーバーラップしている。このオーバーラップ部分があるかないかを見ることで画像が完全に連続的に取り込まれたことを確認することができる。

ここで示した画像は連続的に撮影されているが、場合によってはフィールドが飛び連続性が失われることがある。これは Windows NT が複数のプロセスを時分割で

Fig. 7 回転する黒色点を通常の CCD カメラで撮影して得た連続する 2 つのフレーム画像(a)(b). 画像下方に回転中心がある.

Fig. 8 Fig. 7 (b) 枠線内の拡大図 (縦方向に伸張).

Fig. 9 Fig. 7 の画像を偶数・奇数フィールドに分割, 縦方向を補間した画像. よって, 縦方向の解像度は $1/2$ に低減している.

実行させるため、画像取り込みのためのプロセス以外のプロセスがディスク・アクセスや重い負荷のジョブを実行したときに、画像を連続的に取り込めなくなるからである。この解決法として、画像取り込みのプログラムの優先順位を最大にしておくことが挙げられる。Windows NT のタスクバーからタスク・マネージャを開き、画像取り込みプロセスを選択してその優先順位を上げれば、そのプロセスのみが最優先で実行される。

4. プログレッシブ・スキャン・カメラを用いた画像取り込み

ここでは、最近注目されている 60 フレーム/秒で撮影可能な SONY のプログレッシブ・スキャン・カメラ XC-7500 による画像の取り込みについて解説する。

4.1 プログレッシブ・スキャン・カメラとは

前章で使った通常の RS-170 準拠の CCD カメラは、Fig. 5 に示したように奇数ラインと偶数ラインで電荷蓄積開始時間が異なり、Fig. 7 で見られるようにそのフレーム画像は常に“ブレ”たものになってしまう。また、フィールド画像のみを使えばその垂直方向の解像度がフレーム画像の半分であり、これは画像処理、特に PIV や PTV などのサブ・ピクセル精度を要求される“粒子位置計測”に通常の CCD カメラを用いるときには大きな問題となる。これを解決してくれるのがここで紹介するプログレッシブ・スキャン・カメラである。

プログレッシブ・スキャン・カメラは前章で使った通常の RS-170 準拠の CCD カメラと画素配列は同様であるが、蓄積した電荷の出力が 2 系統あり、通常のカメラの 2 倍の画像転送レート、すなわち 60 フレーム/秒でノン・インターレース画像を取り出すことができる。Fig. 10 はその電荷蓄積期間と出力期間を時系列に示したものである。ここで、期間 (0) に偶数ラインに蓄積(被写体を見た)された電荷は映像出力 1 へ (0)' と出力され、また、同時に(期間(1)に) 奇数ラインに蓄積された電荷は映像出力 2 へ (1)' と出力される。その次の期間(2)および(3)に蓄積された電荷は期間 (0) (1) のものとは異なり、映像出力 2 および 1 へ出力される。これは各映像出力に偶数と奇数フィールドを交互に出力させることで、各映像信号をあたかも通常の RS-170 映像信号と同様に見せかけるためであり、事実、テレビ・モニターや VTR などに入力させることが可能である。得ら

Fig. 10 プログレッシブ・スキャン・カメラの電荷蓄積および映像転送期間。

れる画像は 1/60 秒ごとのフレーム画像であり、その電荷蓄積開始時間も画像全体のすべての点で同一である。

4.2 ハードウェア構成

XC-7500 の出力は 2 系統あるため、自ずとフレーム・グラバも 2 チャンネル必要となるわけであるが、ここでは前章で紹介したフレーム・グラバ DT 3155 を 2 台同時に使用した。なお、同様のことを RGB 入力のカラースタイル・フレーム・グラバ (例えば IC/PCI (Imaging Technology) など) を用いても実現可能である。使用したハードウェアは、CCD カメラとして XC-7500 (SONY) を用いたことおよびフレーム・グラバ DT 3155 を 2 台用いたこと以外は、前章の表に挙げたものと同様である。

4.3 ソフトウェア構成

ソフトウェア構成は前章の ack2hst.c と似たものであり、ack2hst2.c がメイン・プログラムで、player2.c が実際の画像取り込みを行う部分が記述されているファイルである。ack2hst2.c の InitDevice() では 2 枚のフレーム・グラバのデバイスをそれぞれ初期化している。player2.c の PlayerEnter() では OIFgAllocateBuiltInFrame() を呼ぶことでフレーム領域の確保を行っている。これは前章の player.c と似ているが、実際は少し方法が異なっている。それは、前章の player.c ではフレーム領域を 1 フレームのみ確保して、連続的にそのフレーム領域へ取り込まれてくる画像を memcpy() でユーザが GlobalAlloc() によって別に用意した領域へ転送したのに対し、player2.c ではフレーム領域を一度に取り込みたい画像の枚数分用意しておき、ユーザが別に領域を用意しない点である。これは、2 台の DT 3155 から同時に画像を取り込もうとすると memcpy() を用いて画像の転送を並行して行おうとすると転送処理が間に合わず、完全に連続的な取り込みができなかったためである。そのため、インストールされているホスト・メモリ一杯に画像を取り込みたいときには、その分の領域をデバイス領域として PC 起動時からデバイス・ドライバに占有させておく必要がある。この領域は PC 起動後に容易に変更できないため、画像取り込みを行わないとき他のアプリケーションにメモリを解放することができない。この点は不便であり、前章でも述べたようにフレーム・グラバから送られてくる画像を直接ユーザ・データ・メモリへ転送することが可能なライブラリ関数がいずれ用意されることを期待するに留まる。player2.c の RecordImages() では、OIFgAsyncAcquireMultipleToDevice() によってフレーム領域への連続的な画像取り込みがそれぞれのフレーム・グラバに対して指示された後、OIFgIsAsyncAcquireJobDone() によって転送の終了が監視される。

4.4 画像の取り込み

3.4 節と同様に等速回転する黒色点を撮影した結果を Fig. 11 (a)-(d) に示す。なお、2 台のフレーム・グラバによって得られた各フィールド画像はソフトウェア的に合

Fig. 11 プログレッシブ・スキャン・カメラで撮影した連続する4フレームの画像(a)-(d). 被写体は Fig. 9 (Fig. 7) のものと同一.

成してフレーム画像として再構築した。電荷蓄積時間が1/60秒であるため、通常のCCDカメラで撮影した場合 (Fig.7) に比べて“ブレ”幅が1/2である。偶数および奇数フィールドへ同時に電荷が蓄積されるため両フィールドのずれは無く、縦方向の補間を行う必要がないので解像度の低減も無い。

5. おわりに

PCIバス転送型フレーム・グラバによる画像の取り込みについて Data Translation DT 3155 とその具体的なプログラムなどを中心に解説した。今後はPCのホスト・メモリがさらに低価格・大容量化し、連続的に取り込むことのできる画像枚数も今以上に増加すると考えら

れ、フレーム・グラバ+PCシステムが従来のVTRと全く同様に使われるような時代も近いと期待される。

最後に、PCIバスに関する詳細な情報とハードウェアの一部を御提供いただいた慶應義塾大学理工学部システムデザイン工学科の前田昌信教授、菱田公一教授および同大学院生の安部元幸氏、川口達也氏に謝意を表する。

参考文献

- 1) 滝 誠一, “PCIバスの概要”, Interface, 3月号, pp. 100-116, 1997
- 2) 榊原 潤, 菱田公一, 前田昌信, “画像処理による流動場の速度・温度同時計測法”, Thermal Science & Engineering, 2巻, 4号, pp. 1-9, 1994