

〔非公開〕

TR-C-0028

モアレ縞生成及び3次元形状
合成システム取扱説明書

肥塚 隆
TAKASHI KOEZUKA

1989. 3. 13.

ATR通信システム研究所

モアレ縞生成及び3次元形状合成システム

取扱説明書

1989年 3月

A T R 通信システム研究所

知能処理研究室

目 次

1. 目的と概要
2. 実験システムの構成
3. モアレ縞生成のための準備
4. 光学系の調整方法
5. 実験プログラムの概略機能
6. 実験プログラムの主な仕様
(モアレ縞の生成)
7. 実験プログラムの主な仕様
(モアレ縞からの形状合成)

A P P E N D I X

1. 目的と概要

当所において開発したモアレ縞の生成及びモアレ縞からの3次元形状合成の実験システムの機能ならびに使用方法について述べる。開発技術の継承ならびに新研究所への移転や研究者の入替えが生じた場合の迅速な対応が可能ないようにしておくことが主たる目的である。主として前半は実験システムのハードウェアについて、後半はソフトウェアについて記す。

本システムは変形格子像をカメラで入力し、デジタル画像処理によって等高線モアレ縞を生成させ、細線化・次数判定の後3次元形状をサーフェイスモデルで表現するものである。モアレ縞の絶対次数の判定については、光切断法による方法を採用している。またこのサーフェイスモデルを用いてテクスチャマッピング、ステレオ画像対の生成・表示、アニメーションの作成を行うこともできる。

開発したソフトウェアは大別すると、

- (1) 3次元形状自動入力システム・・・モアレ縞の生成等
- (2) 3次元画像処理システム・・・モアレ縞からの形状合成等

から成っている。

なお、本書において使用する用語ならびに記号等を以下のように定める。

表1-1 本書において使用する用語ならびに記号

用語	記号	用語	記号
格子面～基準面間距離	d	カメラ画素の横方向の大きさ	r_x
格子面～回転軸間距離	L	カメラ画素の縦方向の大きさ	r_y
z 軸～プロジェクトの光軸間距離	L_1	カメラの撮影倍率	MAG
z 軸～カメラの光軸間距離	L_2	サンプリングピクセル数	m_1
プロジェクトのレンズの焦点距離	f_g	レンズ中心～格子面間距離	a
カメラレンズの焦点距離	f_c	レンズ中心～カメラ面間距離	a'
カメラふれ角度	β		
基準格子ピッチ	P_0		
サンプリング格子ピッチ	P_s		

2. 実験システムの構成

2-1. 実験システムのハードウェア構成

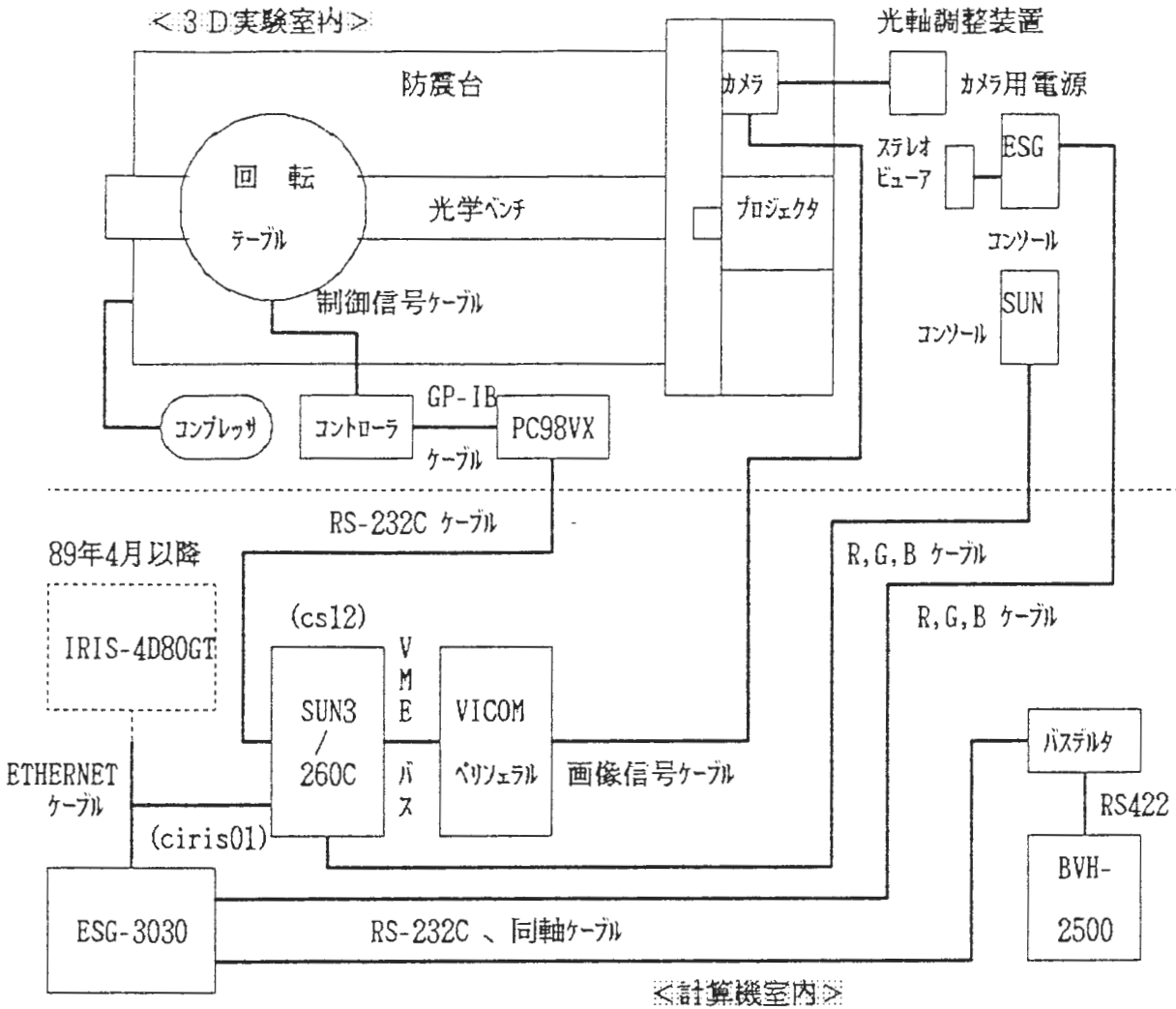


図 2-1 モアレ縞生成及び形状合成実験システムのハードウェア構成

2-2. 各装置の主な仕様と機能

(1) 光軸調整装置 (フォトリクス製)

スライドプロジェクタとカメラとの光軸の間隔や角度を調整するための機器。光軸間隔はミリ単位、カメラの振れ角度は0.1°単位で調整可能。またカメラのあおり角度を調整できるようにカメラ置き台にあおり機構を備えている。

(2) 光学ベンチ (フォトリクス製)

回転テーブルを所定の位置に位置決めするための装置。ミリ単位で調整可能。軸方向は光軸調整装置の軸に対して垂直になるように設定している。

(3)回転テーブル（フォトニクス製）

物体を上へのせ支持するためのテーブル。回転角度の設定精度は 0.01° 、垂直方向移動の設定精度は 0.1 ミクロン、テーブルのあおり角度の設定精度は 0.1° である。

(4)防震台（ヘルツ工業製）

光軸調整装置・光学ベンチを安定的に保持するため空気バネで両者を支えるための装置。

(5)コンプレッサ（日立製）

防震台に圧搾空気を供給する装置。

(6)スライドプロジェクタ（ELMO製）

基準格子を物体上に投影するための光源。特に格子を光軸のまわりに傾ける機構をもっていてカメラの画素の縦方向の配列との平行性を調節可能。

(7)カメラ（KODAK製）

画素数 1340×1120 の高解像度CCDカメラ。VICOMに対して直接デジタルの画像信号（モノクロ）を伝送する。

(8)カメラ用電源（KODAK製）

上記カメラ専用の電源ユニット。

(9)テーブルコントローラ（フォトニクス製）

回転テーブルの回転角度・垂直方向の移動量・あおり角度を設定制御するためのコントローラ。

(10)PC98VX（日本電気製）

ホスト計算機（CS12）からの制御軸名・設定値を受信しテーブルコントローラに送信するとともに、設定値のチェックを行いその結果をホスト計算機に返す。

(11)SUN3/260C（SUN MICROSYSTEMS社製）

ホスト計算機として、回転テーブル制御設定値の送信、画像処理装置（VICOM）の制御、を行うと共にモアレ縞チェーンファイル・オブジェクトファイルの生成、管理並びにプログラム開発を行う。主メモリ 32 MB、FPA、グラフィックスボード、 280 MBディスク等のオプションを搭載している。

(12)VICOM-ペリフェラル

カメラ入力、フィルタリング等の画像処理、画像データファイルの入出力を行う。

モアレ縞の生成はこの装置にて行う。1024系ディスプレイシステム(VIDIS)を装備している。

(3)ESG-3030 (SILICON GRAPHICS社製)

モアレ縞チェーンファイルの表示、オブジェクトファイルの生成・表示、テクスチャマッピング、ステレオ画像の表示等を行うグラフィックエンジン。FPA、170MBディスク、GENLOCK等のオプションを搭載している。またテクスチャマッピング等CG画像生成・表示・録画のための WAVEFRONT SOFTWARE をインストールしている。なお図には記していないが、グラフィックエンジン・VTRを同一の信号で同期させるための同期信号発生器(TSG-170A、SONY製)からの信号を入力している(以下の機器にも該当する)。またCRTのR、G、B、SYNCの信号はNTSCエンコーダ(FAROUJA製)に入力されている。

(4)IRIS-4D80GT (SILICON GRAPHICS社製)

モアレ縞チェーンファイルの表示、オブジェクトファイルの生成・表示、テクスチャマッピング、ステレオ画像の表示等を行うグラフィックエンジン。380MBディスク、GENLOCK等のオプションを搭載している。またテクスチャマッピング等CG画像生成・表示・録画のための WAVEFRONT SOFTWARE をインストールしている。

(5)バスデルタ (LYON LAMB 社製)

グラフィックエンジンからの制御信号を受けて画像録画用VTRを制御するためのVTRコントローラ。コマ録り・タイトルマット生成・タイムコードの設定等を行う。

(6)BVH-2500 (SONY製)

グラフィックエンジンで表示している画像を録画するための1インチVTR。

(7)ステレオビューア (理経製)

ステレオ画像対を時分割的に表示し、液晶メガネを同期させて立体視させるための装置。

2-3. 主な装置の処理内容及びデータの流れ

表2-1 処理内容及びデータの流れ

SUN3/260C+VICOMペリフェラル	ESG-3030	PC98VX
<p>①回転テーブル制御設定値送受信 (SUN) ←</p> <p>②カメラ入力 (VICOM) CCDカメラのデジタル映像信号の読込 画像メモリへの書込み</p> <p>③モアレ縞の生成 (VICOM) フィルタリング処理 (コマンドの実行) 画像データの入出力・表示</p> <p>④モアレ縞のトレース・細線化 (SUN)</p> <p>⑤モアレ縞の次数判定 (SUN)</p> <p>⑥モアレ縞チェーンファイルの作成 (SUN) ←</p> <p>⑦モアレ縞からのサーフェイスモデルの生成 (SUN)</p>	<p>ftpでファイル転送 (バイナリモード)</p> <p>①テキストチャデータの生成 ・表示</p> <p>ftpでファイル転送</p> <p>②モアレ縞からのサーフェ イスモデルの生成</p> <p>③サーフェイスモデルの表 示・操作</p> <p>④テキストチャマッピング等 のCG画像処理・表示</p> <p>⑤VTRへの録画</p>	<p>→ 設定値受信 エラー チェック 回転テーブ ル制御 受信値送信</p>

3. モアレ縞生成のための準備

3-1. VICOMペリフェラルの立ち上げ

- (1) CRT、INTERACTIVE DEVICEの電源ON (3D実験室)。
- (2) 本体背面のブレーカをON (以下コンピュータ室2)。
- (3) 本体前面のキーを中央に回す。
- (4) 本体前面の緑のボタンを押す。点灯を確認する。
- (5) 本体前面のキーを右まで回す。
- (6) VIDIS (1024系ディスプレイシステム) の電源をON。

3-2. SUN3/260Cの立ち上げ

- (1) CRTの電源をON (3D実験室)。
- (2) 本体及びディスクの電源をON (コンピュータ室2)。

注意 UNIXを起動した後、VICOMのドライバをロードしている途中でシステムがダウンしたり、HANGUPすることがある (特にその日最初に立ち上げる時) ので、その場合は一旦すべての機器の電源をOFFとし3-1、3-2の操作を初めから行うこと。

3-3. 回転テーブルの位置の調節

- (1) 撮影対象物の大きさ、モアレ縞の分解能にあわせて対象物をのせる回転テーブルの格子面 (基準面) からの距離を決定する。
- (2) 回転テーブルを光学ベンチ上をスライドさせて上記距離の位置に移動させ固定する。またその時の回転中心の位置を基準位置 (L及びd) とするので目盛りを讀んでおくこと。回転軸の位置は垂直移動軸の中心に一致し光学ベンチに平行な方向に見て端から120ミリにある。

3-4. 基準格子の選定

- (1) 基準格子としては、格子ピッチが①30ミクロン・②40ミクロン・③50ミクロンの3種類があるので、対象物の大きさや位置、カメラレンズの焦点距離等から適当なものを選定する。
- (2) プロジェクタのレンズ (70~120ミリズーム) は常に焦点距離を最小の位置

にしておくこと（モアレ縞の分解能を最大にするため）。

3-5. カメラレンズの選定

- (1)撮影対象物の大きさ、モアレ縞の分解能にあわせて適当に選定する。用意してあるレンズとしては、焦点距離で①55ミリ（標準レンズ）・②35～105ミリ・③100～300ミリの3種類がある。目安としては、ウサギの置物ならば②または③を、ヴィーナスの胸像程度の大きさであれば①を用いればよい。
- (2)選定したレンズをカメラにマウントする。

3-6. カメラの姿勢の調節

- (1)回転テーブル上に倍率測定用スクリーンを配置する。
- (2)カメラとプロジェクタとの光軸間隔を所望の値にセットする。
- (3)カメラの電源ユニットの電源をON（シャッタの機械音がする）。
- (4)VICOMのcimを立ち上げ、プログラム'vertchck.vc'を実行する。
- (5)レンズのピントを調節する。
- (6)倍率測定用スクリーン上の縦方向の目盛りが画面の中央（赤い線で表示）にくるようにカメラ置き台のマイクロメータを調節する。
- (7)倍率測定用スクリーン上の縦方向の目盛りが赤い線と一致するようにつまみを回してカメラのあおり角度を調節する（写真3-1）。

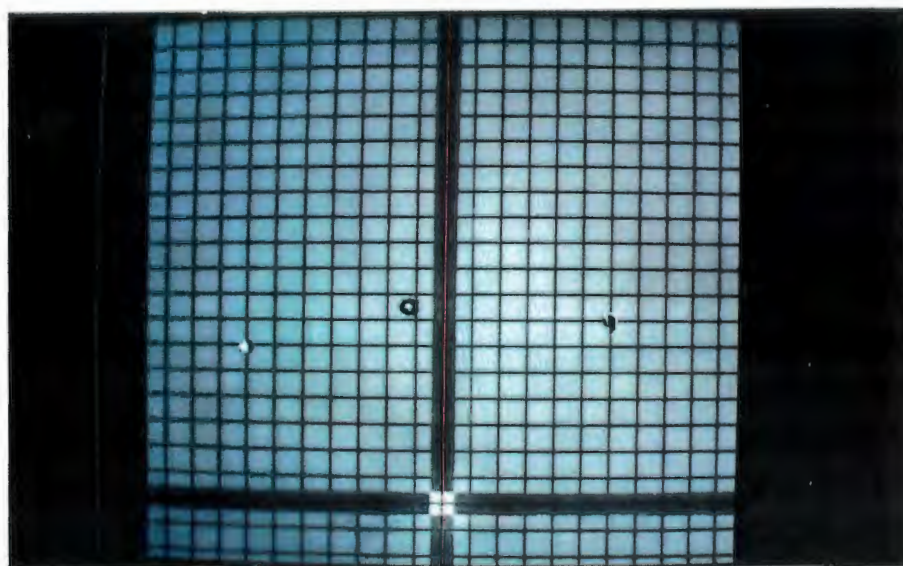


写真3-1 カメラのあおり角度の調節

3-7. カメラレンズの倍率の測定

(1)倍率測定用スクリーンをカメラで撮影した画像から、目盛り上の適当な長さ x (例えば100ミリ) が画像メモリ上の何ピクセルに相当するか (p) を測定する。

c i mレベルで# p r iを使用する (写真3-2)。

(2)カメラレンズの倍率 (MAG) を以下の式により求める。

$$\text{MAG} = \frac{x}{p}$$

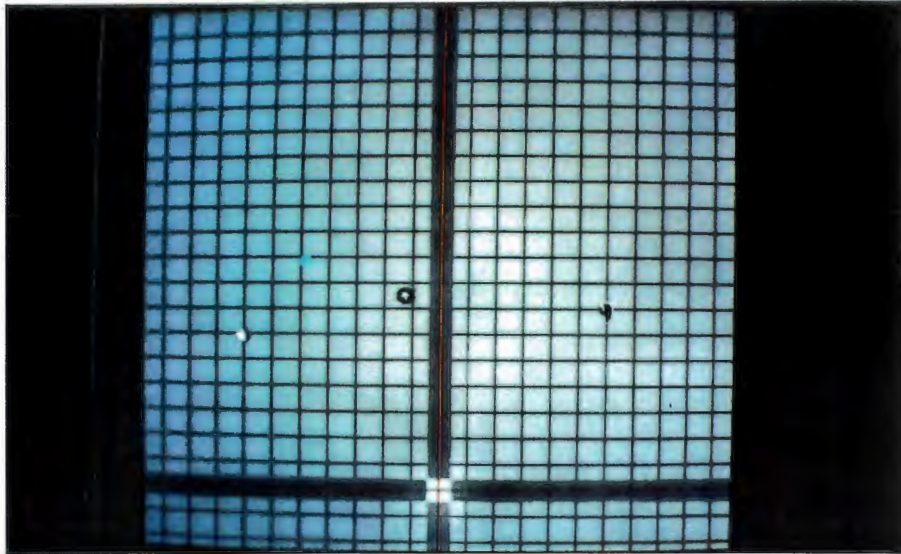


写真3-2 カメラレンズの倍率の測定

3-8. カメラレンズの焦点距離の計算

(1)3-7. で測定したMAGを用いてカメラレンズが理想的な単レンズであると見なしてカメラレンズの焦点距離 a' を以下の式により求める。

$$a' = \frac{r_x}{\text{MAG} + r_x} (L + r_r)$$

3-9. カメラふれ角度の計算

(1)カメラ光軸と光学ベンチの基準軸との間の角度 β を以下の式により求める。

$$\beta = \arctan \frac{l}{L + r_r} \quad \text{単位は}^\circ \text{。}$$

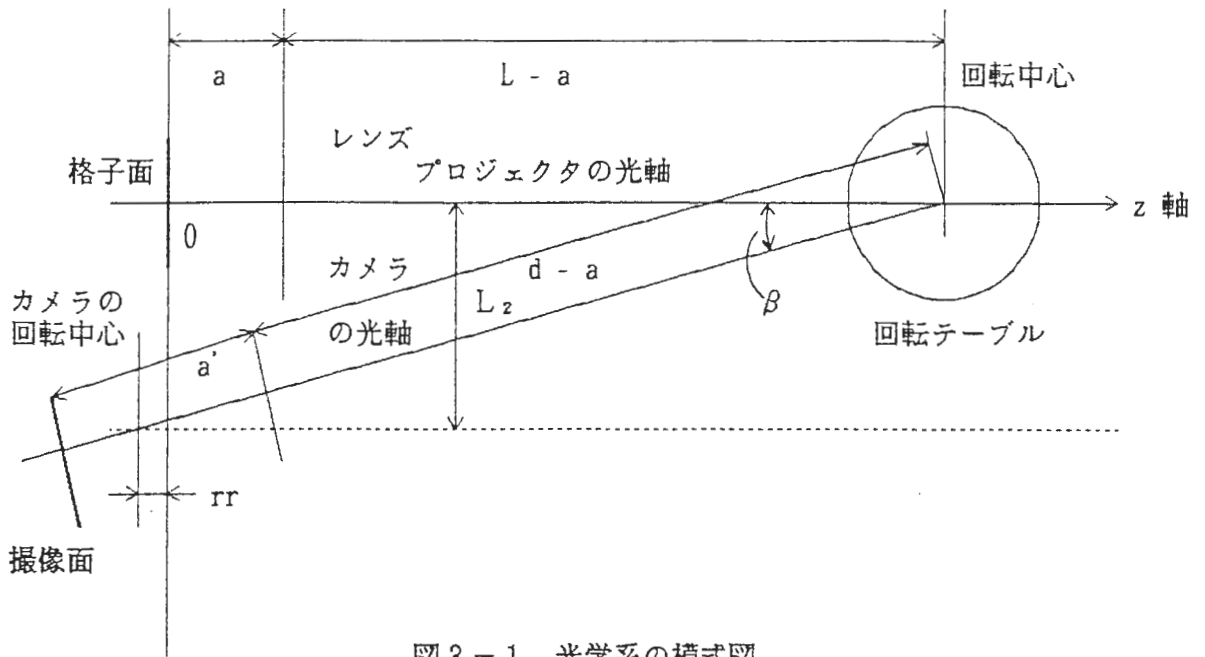


図3-1 光学系の模式図

格子面は光学ベンチの目盛り上で $2810 + 148.5 = 2958.5 \text{ mm}$ のところに相当する。回転中心は垂直テーブルの中心であり、格子面に近い側のエッジから 120 mm の位置にあるので、そのエッジの位置が光学ベンチの目盛り上で D であるとすると、

$$L = 2958.5 - D + 120 \text{ mm}$$

として求められる。

4. 光学系の調整方法

3. の準備が終わった後、以下の要領で光学系の調整を行う。

- (1) 回転テーブル上に倍率測定用スクリーンを配置する。
- (2) 倍率測定用スクリーン上でカメラ視野に入る部分に白画用紙を貼り付ける。
- (3) プロジェクタ、カメラ電源ユニット、PC98VX、テーブルコントローラの電源をON。
- (4) 倍率測定用スクリーンがプロジェクタの光軸に垂直になる位置を基準として、カメラふれ角度に等しい角度だけ回転テーブルを回転させスクリーン面がカメラの光軸に対して垂直になるようにする。
- (5) プロジェクタのトレイに基準格子をセットする。
- (6) カメラで白画用紙に投影した格子の画像を撮影する（プログラムは 'vertchck. vc' を使用する）。縦方向の格子が画面中央の赤い線に平行になるようにプロジェクタ本体前面のスライド傾斜用ツマミを調節する（写真4-1）。

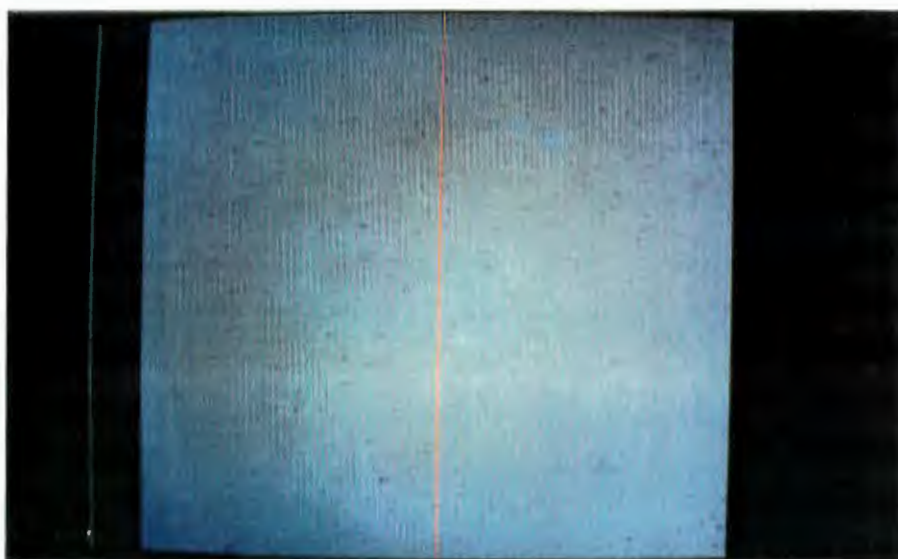


写真4-1 格子の傾きの調節

- (7) 撮影した画像（イメージメモリの1に格納されている）を入力としてモアレ縞を生成させる（プログラムはVICOMの 'moire6. vc' を使用する）。
- (8) モアレ縞のパターンが左右対称ならば調整終了（写真4-2）。
- (9) モアレ縞のパターンが画面の左から右にかけて赤、青、緑の順に現れる場合（写真4-3）は、カメラレンズのズームを小さくする。その後(6)の操作を繰り返す。
- (10) モアレ縞のパターンが画面の左から右にかけて緑、青、赤の順に現れている場合（写真4-4）は、カメラレンズのズームを大きくする。その後(6)の操作を繰り返す。

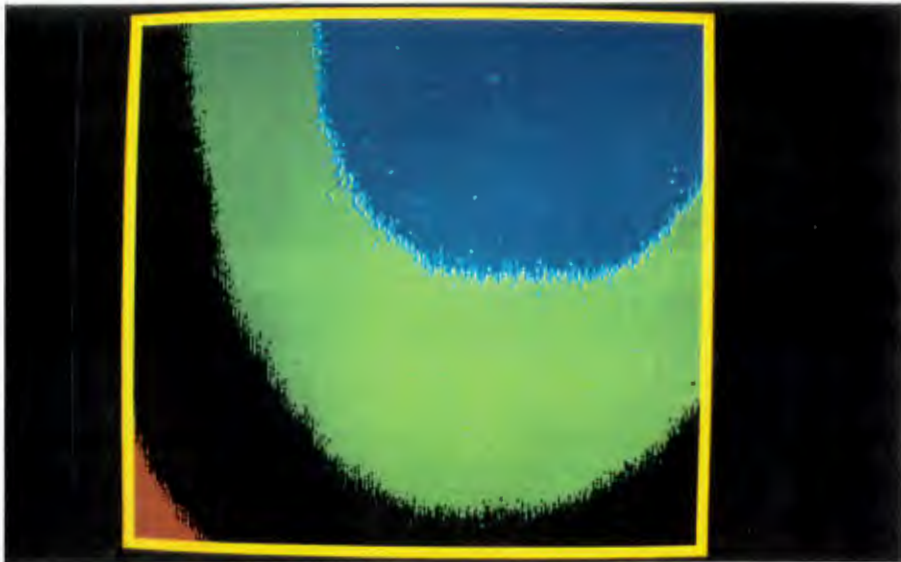


写真4-2 平面のモアレ縞のパターン（調整完了）



写真4-3 平面のモアレ縞のパターン（レンズのズームがやや大きい）



写真4-4 平面のモアレ縞のパターン（レンズのズームがやや小さい）

カメラが画像を撮影する場合の注意事項

以上の調整が終われば、対象物体を回転テーブルにのせテーブルの回転角度や画像の枚数等を適当に決めパラメータファイルを作成して次章以下に述べるプログラムを実行させれば良いが、以下に若干の注意事項を掲げる。

- ①カメラは機械式のシャッターを用いているので画像をとる時以外はなるべく電源をOFFにしておくことが望ましい。
- ②カメラ側でコントラストやゲインを調整できないので、通常はNDフィルタをカメラレンズ前面に装着すると共にレンズの絞りを調整して、できるだけ変形格子像のコントラストを高め、また焦点深度を深くとるようにすること。
- ③人間の顔やコップ等の光る対象物の画像を撮影する場合は、プロジェクタ・カメラ共にレンズ前面に偏光フィルタを装着して光る部分を抑えるようにすること。カメラで撮影した画像において光る部分を最も小さくするようにカメラ側の偏光フィルタをレンズの回りに回転させて調整すること。
- ④変形格子像を撮影する場合は室内をなるべく暗くしておくこと。
- ⑤テクスチャ画像をカメラを用いて撮影する場合は室内の照明の状態に注意すること。特に、人間の顔のテクスチャをとる場合スポットライトを用いて照明しないと、目の瞳や眉毛など暗い部分の輝度レベルが0になってしまいテクスチャマッピングした結果の画像に孔が開いたようになってしまうので注意すること。

5. 実験プログラムの概略機能

プログラムの一覧を以下にまとめる。

表5-1 実験プログラムの一覧表(1)

目的	名称	概略機能	マシン、言語
モアレ縞の生成	moire	①撮影条件、設定データの入力・変更 ②回転ステージの制御 ③カメラからの画像入力 ④モアレ縞の生成 ⑤モアレ縞画像データの出力 ⑥モアレ縞チェーン符号ファイルの作成・出力	SUN3/260C SUN-FORTRAN UNIX-C VICOM ライブラリ
	trace	①モアレ縞の細線化(通常) ②モアレ縞の細線化(4倍密度) ③細線化画像データの出力	SUN3/260C SUN-FORTRAN UNIX-C
	calib	①撮影条件、設定データの入力 ②回転ステージの制御 ③カメラからの画像入力 ④光切断法のための変換マトリックス同定	SUN3/260C SUN-FORTRAN UNIX-C VICOM ライブラリ
モアレ縞からの形状合成	menu 及び menu_4	①メニュー形式で以下に記すプログラムの選択・実行を行う。 *で注を付していないものは両方でサポート。	ESG-3030 UNIX-C グラフィックスライブラリ
	moire1	①モアレ縞チェーン符号ファイルを入力とし画像データを横スキャンすることにより節点を発生させ三角パッチを生成させる。 ②生成させた三角パッチを WAVEFRONTフォーマットのオブジェクトファイルとして出力する。 ③画像データ、節点、三角パッチを表示する。	ESG-3030 UNIX-C グラフィックスライブラリ
	moire2 *menuで選択	①モアレ縞チェーン符号ファイルを入力としモアレ縞を折れ線近似することにより節点を発生させ三角パッチを生成させる。 ②生成させた三角パッチを WAVEFRONTフォーマットのオブジェクトファイルとして出力する。 ③画像データ、節点、三角パッチを表示する。	ESG-3030 UNIX-C グラフィックスライブラリ
	moire2_4 *menu_4で選択	①4倍密度で細線化したモアレ縞チェーン符号ファイルを入力としモアレ縞を折れ線近似することにより節点を発生させ三角パッチを生成させる。 ②生成させた三角パッチを WAVEFRONTフォーマットのオブジェクトファイルとして出力する。 ③画像データ、節点、三角パッチを表示する。	ESG-3030 UNIX-C グラフィックスライブラリ
	dspmoire *menuで選択	①モアレ縞チェーン符号ファイルを入力としモアレ縞画像をz座標の値毎に色分けして画面に表示する。	ESG-3030 UNIX-C グラフィックスライブラリ
	dspmoire_4 *menu_4で選択	①4倍密度で細線化したモアレ縞チェーン符号ファイルを入力としモアレ縞画像をz座標の値毎に色分けして画面に表示する。	ESG-3030 UNIX-C グラフィックスライブラリ

表5-1 実験プログラムの一覧表(2)

	名 称	概 略 機 能	マシン、言語
モ ア レ 縮 か ら の 形 状 合 成	textmap	① WAVEFRONTフォーマットのオブジェクトファイルを入力としてテクスチャ節点を生成する。 ② テクスチャ画像はカメラまたはドラムスキャナ等で入力した VICOMフォーマットの画像データを用いる。 ③ .rlaファイルから.texファイルを生成する。	ESG-3030 UNIX-C グラフィックスライブラリ WAVEFRONT ライブラリ
	vicom	① VICOMフォーマットの画像データ (R,G,B)を.rlaフォーマットの画像データに変換する	ESG-3030 UNIX-C グラフィックスライブラリ WAVEFRONT ライブラリ
	stereoview	① 左右2台のカメラを定義して、WAVEFRONTフォーマットの previewファイルを作成する。 ② 指定したフレームに対して imageを実行して、ステレオ画像対を画面に表示する。	ESG-3030 UNIX-C グラフィックスライブラリ WAVEFRONT ライブラリ
	video	① imageを実行して生成させたステレオ画像対を1インチVTRに録画する。 ② 左右2台のカメラに対する画像を片方ずつ、又は交互に録画する。	ESG-3030 UNIX-C グラフィックスライブラリ WAVEFRONT ライブラリ
	dspimage	① WAVEFRONTフォーマットの previewファイルに対して imageを実行して生成させたステレオ画像対、又は既に生成させたステレオ画像対を画面に表示する。	ESG-3030 UNIX-C グラフィックスライブラリ WAVEFRONT ライブラリ

注) moire2及びmoire2_4についてはグラフィックス処理を除いた形でSUNにても実行可能である。但しmenuに相当するユーティリティは用意していないのでプログラム起動時にパラメータをキーボードから入力すること(詳しくは操作マニュアルを参照のこと)。

6. 実験プログラムの主な仕様

(モアレ縞の生成)

6-1. ソフトウェア環境

(1) ディレクトリ構成

```
Host name      cs12
Login name     moire
passwd        csk  大内に聞いて下さい。
```

表6-1 ディレクトリ構成

ディレクトリ階層		ディレクトリ内容	
/usr /moire /fmoire	/src	/abs	絶対次数処理
		/atr	メイン制御・カメラ制御
		/basic	共通ルーチン
		/hosei	補正用変換マトリックス作成処理
		/init	パラメータファイル(.jis、.jyo) 読み込み& 作成
		/irs	チェインコード化ファイル(.wav)作成処理
		/niso	基準縞相対次数付与処理
		/piso	1/4 縞相対次数付与処理
		/stage	ステージ制御処理
		/trace	細線化処理
		/vcm	画像処理 (for VICOM)
	/vcmint	VICOM コマンドCALLルーチン(for C プログラム)	
	/common	コモンデータ定義	
	/wav	チェインコード化ファイル格納エリア	
/log	相対次数情報格納エリア		
/bin	実行ファイル格納エリア		
/para	パラメータファイル格納エリア(.jis、.jyo)		
/usr1/3d/moire		画像データファイル格納エリア	

(2) 実行モジュール作成方法

表 6-2 実行モジュール作成のためのコマンドファイル

システム名	(A)	(B)
モアレ縞生成システム	moire.com	dmoire.com
モアレ縞細線化システム	trace.com	dtrace.com
変換マトリックス同定システム	calib.com	dcalib.com

(A) NODEBUGモードのモジュールを作成する。
 moire> /usr/moire/fmoire/com/moire.com <CR>
 moire> /usr/moire/fmoire/com/trace.com <CR>
 moire> /usr/moire/fmoire/com/calib.com <CR>

(B) DEBUGモードのモジュールを作成する。
 moire> /usr/moire/fmoire/com/dmoire.com <CR>
 moire> /usr/moire/fmoire/com/dtrace.com <CR>
 moire> /usr/moire/fmoire/com/dcalib.com <CR>

<注意>

1. 実行ファイルの作成は、すべて /usr/moire/fmoire/f68881work で行うこと。尚、ソースファイルを修正した場合は、lnコマンド（シンボリックリンク）を用いて修正済みファイルとのリンクを取り直してから、コンパイルすること。

ex.

```
moire> cd /usr/moire/fmoire/f68881work <CR>          ディレクトリの変更
moire> ln -s /usr/moire/fmoire/src/adim/adm000.f . <CR>   ファイルのリンク
moire> ./com/moire.com <CR>                             実行ファイル作成
```

2. 入出力ファイルのディレクトリを変更したい場合

手順

- 1) /usr/moire/fmoire/common/cgdef.txtファイルを修正する。その時必ず、パス名の文字数とcharacter宣言している文字数とを等しくすること。
- 2) 実行モジュール作成手順に従い、モジュールを作成しなおす。尚、再コンパイルが必要なプログラムを以下に示す。

```
adim/absa10.f  basic/ifopen.f  basic/slit_o.f  hosei/creimg.f
hosei/hosfil.f hosei/setuv.f  init/int130.f  init/int140.f !
init/int200.f  iris/irs100.f  isou/niso200.f main/atr000.f !
main/atr002.f  trace/lab_main.f vcm/vcm400.f  vcm/vcm410.f
vcm/vcm420.f
```

(3). loginファイルの変更

モアレ縞生成システムに於いては、VICOMライブラリを利用するため、
.loginファイルにおいてパスの設定を以下のように行うこと。

```
set path=( /usr/vbin )
```

6-2. モアレ縞生成システム (moire)

(1) 処理の流れ

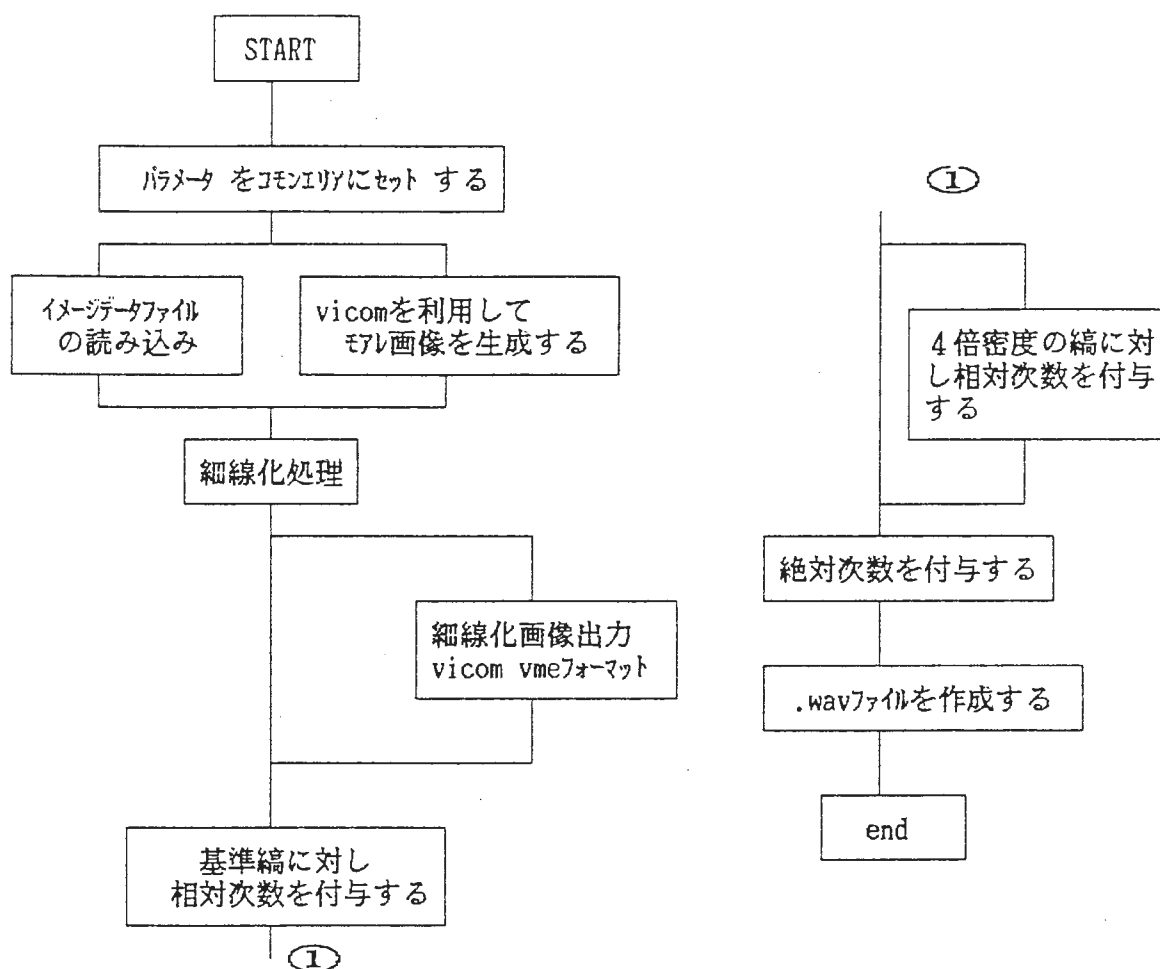


図6-1 モアレ縞生成システム処理フロー

(2) 起動例

<CASE-1 カメラ入力を行わない場合>

次のファイルの存在を確認してから実行する。

- ① . /usr/moire/fmoire/log/testdata.dat
- ② . /usr/moire/fmoire/log/<fn1>.chain
- ③ . /usr1/3d/moire/<fn2>.im

<fn1> : *.jis ファイル のチェーンコードファイル名
<fn2> : *.jis ファイル の変形スリット画像ファイル名

- ①のファイルが存在しない場合は、
「変換マトリックス同定システム」を利用して、変換
マトリックスファイル(testdata.dat)を作成しておく
こと。
- ②③のファイルが存在しない場合は、CASE-2のように
実行すること。

```
moire> pwd
/usr/moire/fmoire/obj
moire> moire.exe
```

```
#####
##                                     ##
##                                     ##
##      AUTOMATIC ACQUISITION HAS JUST STARTED !!!      ##
##                                     ##
##                                     ##
##      ATR Communication Systems Research Lab.          ##
##                                     ##
##                                     ##
#####
```

```
$common file name :
ohface12
```

```
$do you want to create .jis and .jyo file ? [yes] :
y
```

```
##### check input parameter #####
```

< .jyo file >

```
common file name          -----> ohface12
position of normal surface -----> 2210.5000000
position of rotation axis  -----> 2210.5000000
distance between l-s and z axis -----> 18.0000000
distance between cam and z axis -----> 200.0000000
focal length of light source -----> 90.0000000
focal length of camera    -----> 100.0000000
```

パラメータファイルの
ファイル名を入力
ここをyとする
とパラメータファイル
(.jis,.jyo)
作成に必要な
パラメータを聞いて
くるので、その
入力要求に従い
入力する。

```

angle between cam and z-axis  ----->  5.0000000
projection gratings pitch    ----->  0.0500000
sampling gratings pitch      ----->  0.0500000
pixel length in x direction  ----->  0.0068000
pixel length in y direction  ----->  0.0068000
magnitude                    ----->  0.1645000
number of sampling pixels    ----->  6
position of l-s lens center  ----->  94.6934814
position of cam lens center  ----->  88.6800003

```

\$ok ? (yes or no) :

y

check input parameter

```

< .jis file >
number of shots      ----->  1

no lshot
rotation angle      ----->  0.000
method to determine a-o----->  4

```

\$ok ? (yes or no) :

y

Do you want 4 times relative order ? ... ([y] /n)

y

Do you want to store thinned image ? ... (y/ [n])

n

Do you want to take image by camera ? ... ([y] /n)

n

No. lshot image is being processed !!!

Image data is being read now !!

Image data is being traced now !!

Relative order determination -<1>

Relative order determination -<2>

Absolute order determination

x = 391.0000000000000

y = 285.0000000000000

10.46871259948262

<CASE-2 カメラ入力を行う場合>

次のファイルの存在を確認してから実行する。

- ①. /usr/moire/fmoire/log/testdata.dat
本ファイルが存在しない場合は、
「カメラデータ校正マトリックス作成システム」
を利用して、校正マトリックスファイル(testdata.dat)
を作成すること。

```
moire> moire.exe
#####
##
##
##    AUTOMATIC ACQUISITION HAS JUST STARTED !!!    ##
##
##
##    ATR Communication Systems Research Lab.        ##
##
##
#####
```

\$common file name :

penta

\$do you want to create .jis and .jyo file ? [yes] :

n

check input parameter

< .jyo file >

common file name	----->	penta
position of normal surface	----->	2210.5000000
position of rotation axis	----->	2210.5000000
distance between l-s and z axis	----->	18.0000000
distance between cam and z axis	----->	200.0000000
focal length of light source	----->	90.0000000
focal length of camera	----->	100.0000000
angle between cam and z-axis	----->	5.0000000
projection gratings pitch	----->	0.0500000
sampling gratings pitch	----->	0.0500000
pixel length in x direction	----->	0.0068000
pixel length in y direction	----->	0.0068000
magnitude	----->	0.1645000
number of sampling pixels	----->	6
position of l-s lens center	----->	94.6934814
position of cam lens center	----->	88.6800003

\$ok ? (yes or no) :

y

check input parameter

< .jis file >

number of shots	----->	1
no lshot		
rotation angle	----->	0.000
method to determine a-o	----->	4

\$ok ? (yes or no) :

```

y:
Do you want 4 times relative order ? ... ( [y] /n)
y:
Do you want to store thinned image ? ... (y/ [n] )
n:
Do you want to take image by camera ? ... ( [y] /n)
y:※1
threshold level = 0.4
are you satisfied ? yes=y no=<cr>
y:
No, 1shot image is being processed !!!
Image data is being read now !!
Image data is being traced now !!
Relative order determination -<1>
Relative order determination -<2>
Absolute order determination
x = 517.0000000000000
y = 751.0000000000000
-1.60282769070556

```

物体の輪郭を表すマスク
 がくっきりと表示される
 ようにスレッシュホールド値
 を調節する。

※1 ここで”y”と入力するとカメラより画像を取り込みながらVICOM上で処理を
 することとなり、その後細線化処理の方へ進む。

6-3. モアレ縞細線化システム (trace)

(1) 処理の流れ

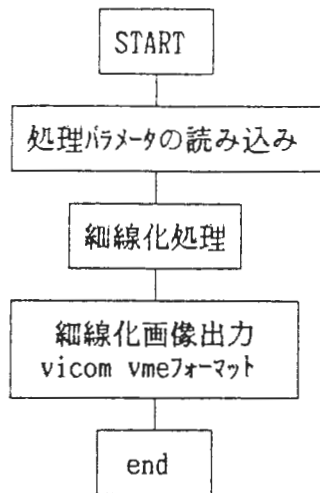


図6-2 モアレ縞細線化システム処理フロー

(2) 起動例

```
moire> cd /usr/moire/fmoire/obj <CR>
moire> trace.exe <CR>
Please input image data file name ...
mane1224_ori.i

Please input image data size ...
1024
Image data is being traced now !!
Image data is being written !!! (vicom vme format)
end...
```

注意) 画像ファイルの格納ディレクトリは、/usr/moire/fmoire/common/cgdef.txt
ファイルで定義してあるディレクトリで、かつファイルタイプは、".im" でな
なければならない。また画像データのサイズは、1024×1024しか対応していない
ので、必ず1024とすること。

(3) 起動結果例



写真6-1 人物の顔のモアレ縞画像 (入力画像)



写真6-2 人物の顔のモアレ縞画像 (細線化画像)

6-4. 光切断法のための変換マトリックス同定システム (calib)

(1) 処理の流れ

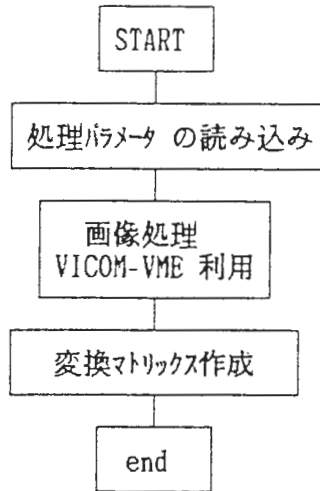


図6-3 光切断法のための変換マトリックス同定システム処理フロー

モアレ縞の絶対次数を光切断法によって求めるための変換マトリックスを同定するシステムである。アルゴリズムの詳細等については、SPIDER-II のマニュアルを参照すること。

なお、本システムはカメラでの撮影条件を変更する（格子面～基準面間距離・レンズの焦点距離等を変える）度に起動する必要がある。

(2) 起動方法

```
moire> cd /usr/moire/fmoire/obj <CR>
moire> calib.exe <CR>
$Parameter data input Start ... !
$If you want to cancel then please input "!" mark
$Please input rotate angle (degree)
$No1. shot angle ( from now position ) :
0.0

$No2. shot angle ( from No1. position ) :
5.0
$No3. shot angle ( from No2. position ) :
10.0
$Please input total spetial point      :
4
$Please input No1. shot (x,y,z) axis.
$Please input No. 1 axis.
$X =====>

10
$Y =====>
```

ディレクトリの変更。
実行モジュールの起動。

現在のステージ位置
の回転角度を入力す
る。尚、右回りを+、
左回りを-とする。

特徴点数を入力する。

特徴点の座標(x,y,z)
を入力要求に従い入力す
る。

```

:10:
$Z =====>
:10:
$Please input No. 2 axis.
$X =====>
:10:
$Y =====>
:10:
$Z =====>
:10:
$Please input No. 3 axis.
$X =====>
:10:
$Y =====>
:10:
$Z =====>
:10:
$Please input No. 4 axis.
$X =====>
:10:
$Y =====>
:10:
$Z =====>
:10:
threshold level = 0.4
are you satisfied ? yes=y no=<cr>
:y

```

特徴点のみがくっきりと表示されるようにスレシヨルド値を調節する。

2) 起動結果例



写真6-3 変換マトリックス同定のための入力画像

6-5. パラメータファイル例

<ohface12.jis>

1	---	撮影ショット数
4	---	第1ショットの絶対次数の求め方
0.000	---	第1ショットの回転角度 (TOTAL 角)
ohface12_ori_1	---	第1ショットのオリジナル画像ファイル名
ohface12_y-1_1	---	第1ショットの変形スリット画像ファイル名

< ohface12.jyo >

OHFACE12	-----	対象物
2210.5000000	-----	格子面 - 基準面間距離
2210.5000000	-----	格子面 - 回転軸間距離
18.0000000	-----	z軸 - プロジェクタの光軸間距離
200.0000000	-----	z軸 - カメラの光軸間距離
90.0000000	-----	プロジェクタのレンズの焦点距離
100.0000000	-----	カメラレンズの焦点距離
5.0000000	-----	カメラふれ角度
0.0500000	-----	基準格子ピッチ
0.0500000	-----	サンプリング格子ピッチ
0.0068000	-----	カメラ画素の横方向の大きさ
0.0068000	-----	カメラ画素の縦方向の大きさ
0.1645000	-----	カメラの撮影倍率
6	-----	サンプリングピクセル数
94.6934814	-----	レンズ中心 - 格子面間距離
88.6800000	-----	レンズ中心 - カメラ面間距離

7. 実験プログラムの主な仕様

(モアレ縞からの形状合成)

7-1. ソフトウェア環境 (ESG-3030の環境)

(1) ディレクトリ構成

```
Host name      ciris01
Login name     koezuka
passwd         csk  大内に聞いて下さい。
```

表7-1 ディレクトリ構成

ディレクトリ階層		ディレクトリ内容
/usr1 /people2 /moire	/src	moire1、textmap、stereoview、video、dspmoire、dspimage、vicom、menuのソースファイル及び実行モジュール作成のためのコマンドファイル
	/src2	moire2、moire2_4、dspmoire_4、menu_4のソースファイル及び実行モジュール作成のためのコマンドファイル
	/bin	実行モジュール
	/wrk	モアレ縞チェイン符号ファイル、WAVEFRONTフォーマットオブジェクトファイル、テクスチャデータ、WAVEFRONTフォーマット previewファイル、CG画像データ等のデータ

(2) 実行モジュール作成方法

表7-2 実行モジュール作成用コマンドファイル

プログラム名	コマンドファイル
moire1	clmoire
moire2	clmoire2
moire2_4	clmoire2_4
textmap	cltext
stereoview	clstereo
video	clvideo
dspmoire	cldspmoire
dspmoire_4	cldspmoire_4
dspimage	cldspimage
mksetup	clmksetup
vicom	clvicom
menu	clmenu
menu_4	clmenu_4

実行モジュールを作成するにはソースファイルが存在しているディレクトリにおいて各コマンドファイル名<CR>を入力する。

(3) 環境設定

本システムならびに WAVEFRONTソフトウェアを使用するのに必要な環境設定（デフォルトディレクトリ、パスの設定等）はログインファイル 'MOIRE' を実行する。

ログイン名moireでログインした場合は、mo<CR>及びwrk<CR>を入力するだけでよい。それ以外の場合は、/usr1/people2/moireの中にあるMOIREを実行する（source MOIRE<CR> を入力）。

7-2. ソフトウェア環境 (SUN3/260Cの環境)

(1) ディレクトリ構成

```
Host name      cs12
Login name     moire
passwd        csk  大内に聞いて下さい。
```

表7-3 ディレクトリ構成

ディレクトリ階層		ディレクトリ内容
/usr	/ccc	moire2、moire2_4のソースファイル 実行モジュール作成のためのコマンドファイル モアレ縞チェーン符号ファイル、WAVEFRONTフォーマットのオブジェクトファイル 実行モジュール
/koezuka	/com	
	/wav	
	/exe	

(2) 実行モジュール作成方法

表7-4 実行モジュール作成用コマンドファイル

プログラム名	コマンドファイル
moire2	clmoire2
moire2_4	clmoire2_4

実行モジュールを作成するにはソースファイルが存在しているディレクトリにおいて各コマンドファイル名<CR>を入力する。

(3) 環境設定

ログイン名 `moire` でログインした場合は、特になし。それ以外の場合は `/usr/moire` の中に `.login`、`.cshrc` 及び `.jcschrc` を実行する (`source .login<CR>`、`..`、`etc.`)

7-3. 処理の流れ

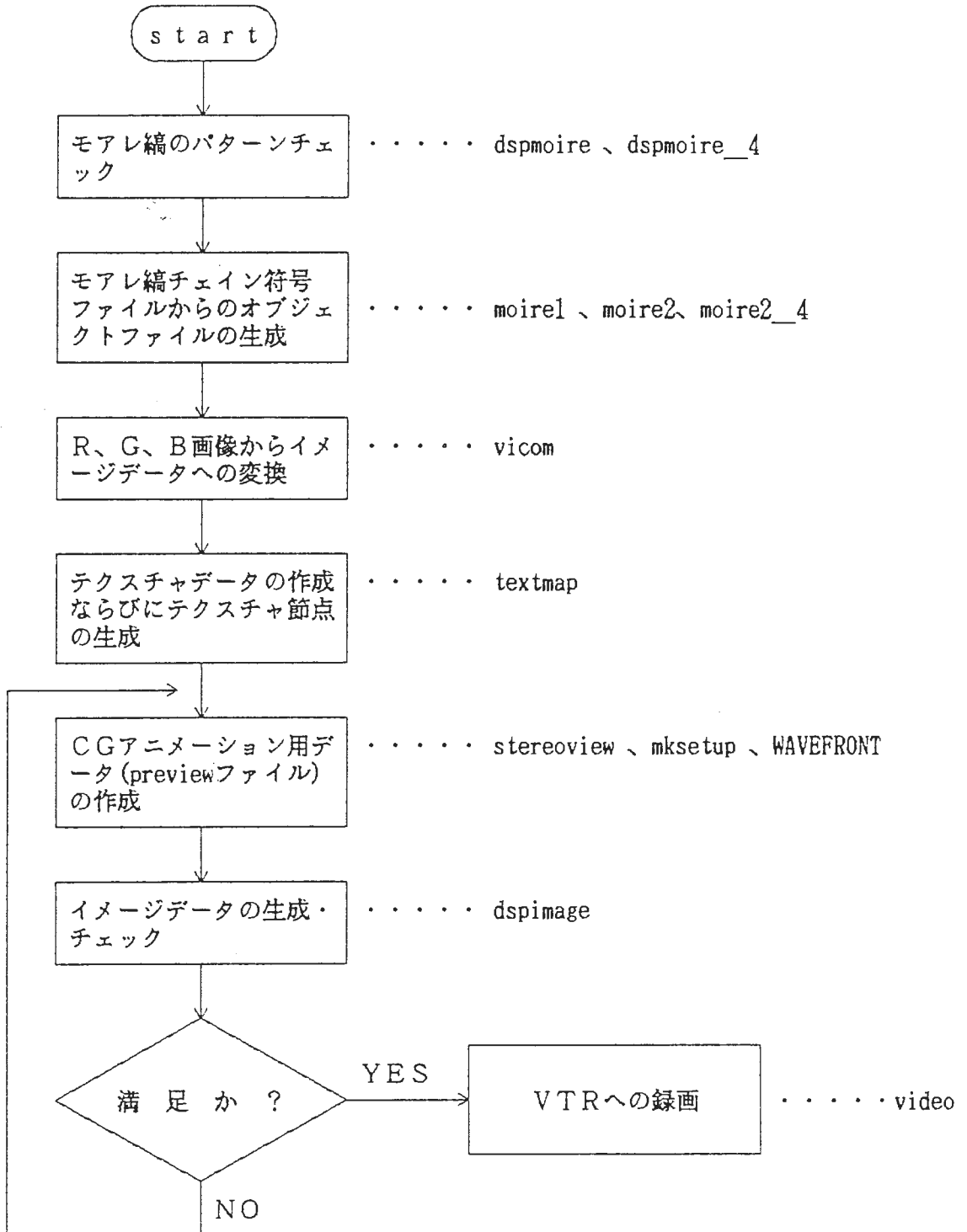


図7-1 3次元画像処理システムにおける処理の流れ

7-4. 機能仕様及びアルゴリズム

以下に各プログラムに関しての処理の流れと起動例についてのみ記す。機能仕様・アルゴリズムの詳細については東洋情報システム及び肥塚が作成した「3次元画像処理システム プログラム説明書Ⅰ」、「3次元画像処理システム プログラム説明書Ⅱ」、「3次元画像処理システム プログラムリスト」を参照すること。

使用方法の詳細については、「3次元画像処理システム 操作マニュアル」を参照のこと。

また WAVEFRONT SOFTWARE については、「WAVEFRONT USER'S MANUAL」を参照すること。

WAVEFRONT SOFTWARE は、VAX-8650 (a t r - s w) にもインストールしておりコマンド入力方式で全ての機能を使うことができる(対話形処理はできない)。テクスチャマッピングは約4倍の速度で処理するパフォーマンスがある。また4月以降 I R I S - 4 D 8 0 G T 及びその上での WAVEFRONT SOFTWARE を導入する予定であり、その時点では約10倍程度のパフォーマンス向上を見込める。

7-5. メニュー処理 (menu及びmenu_4)

(1) 処理の流れ

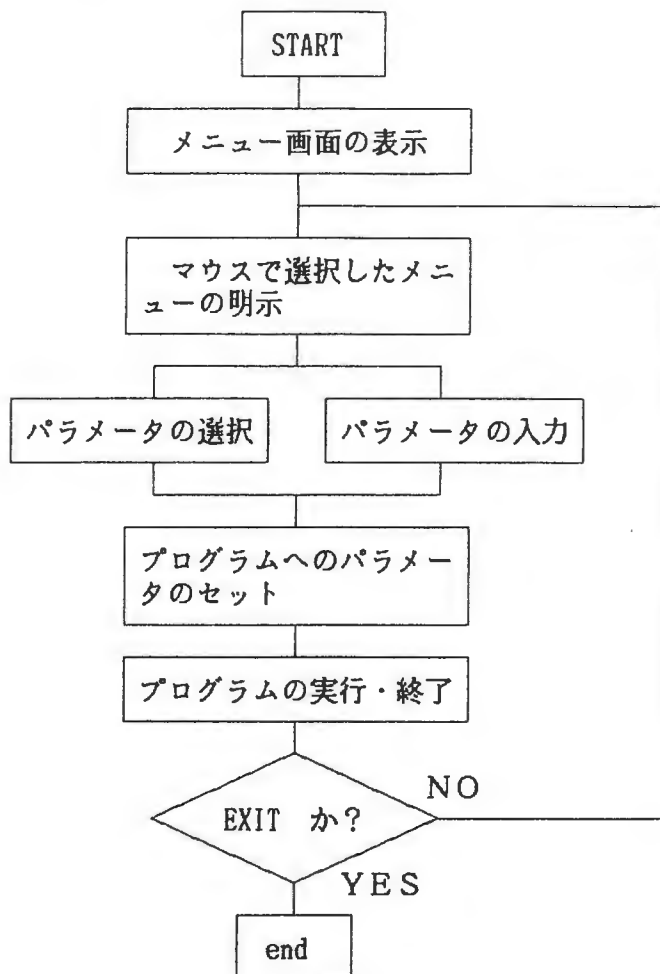


図7-2 メニュー処理の流れ

(2) 起動例

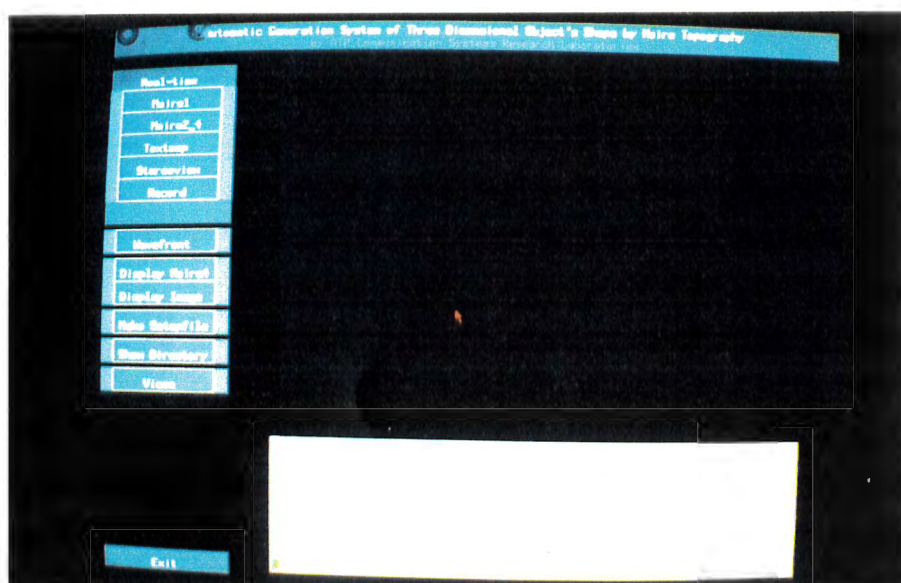


写真7-1 メニュー画面 (menu_4 起動後)

7-6. サーフェイスモデルの生成 (moire1)

(1) 処理の流れ

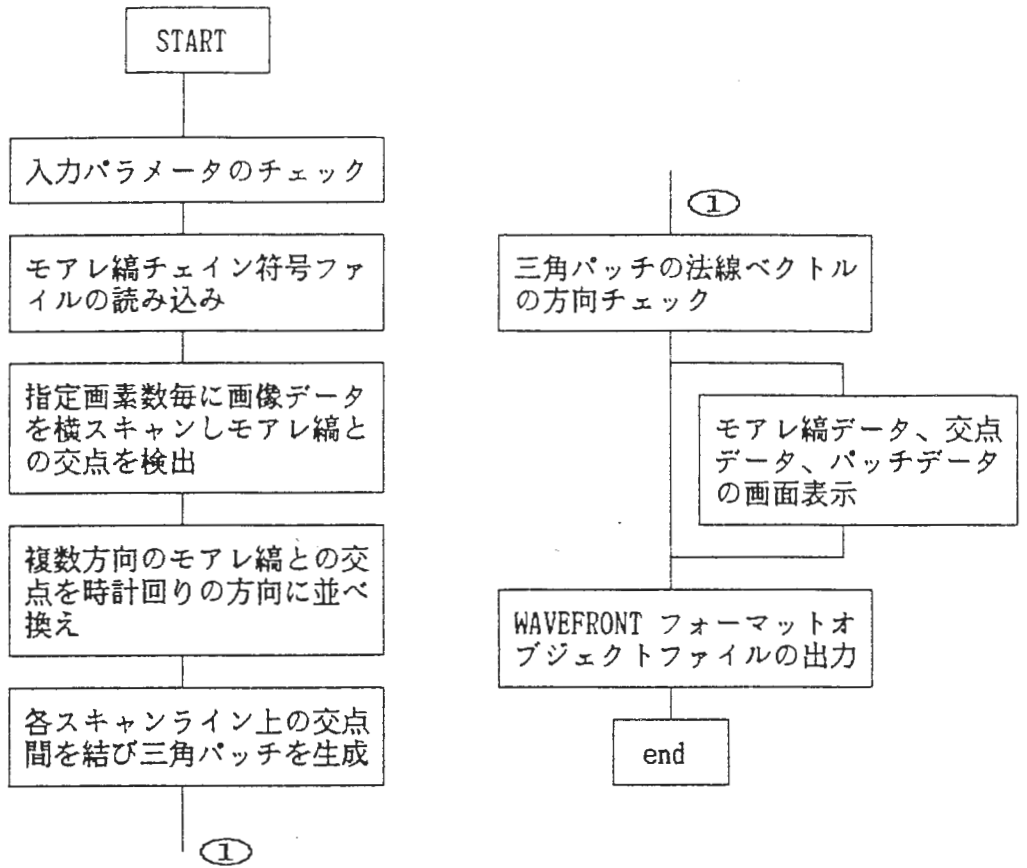


図7-3 moire1の処理の流れ

(2) 起動例 (キーボードよりの入力)

省略。

7-7. サーフェイスモデルの生成 (moire2及びmoire2_4)

(1) 処理の流れ

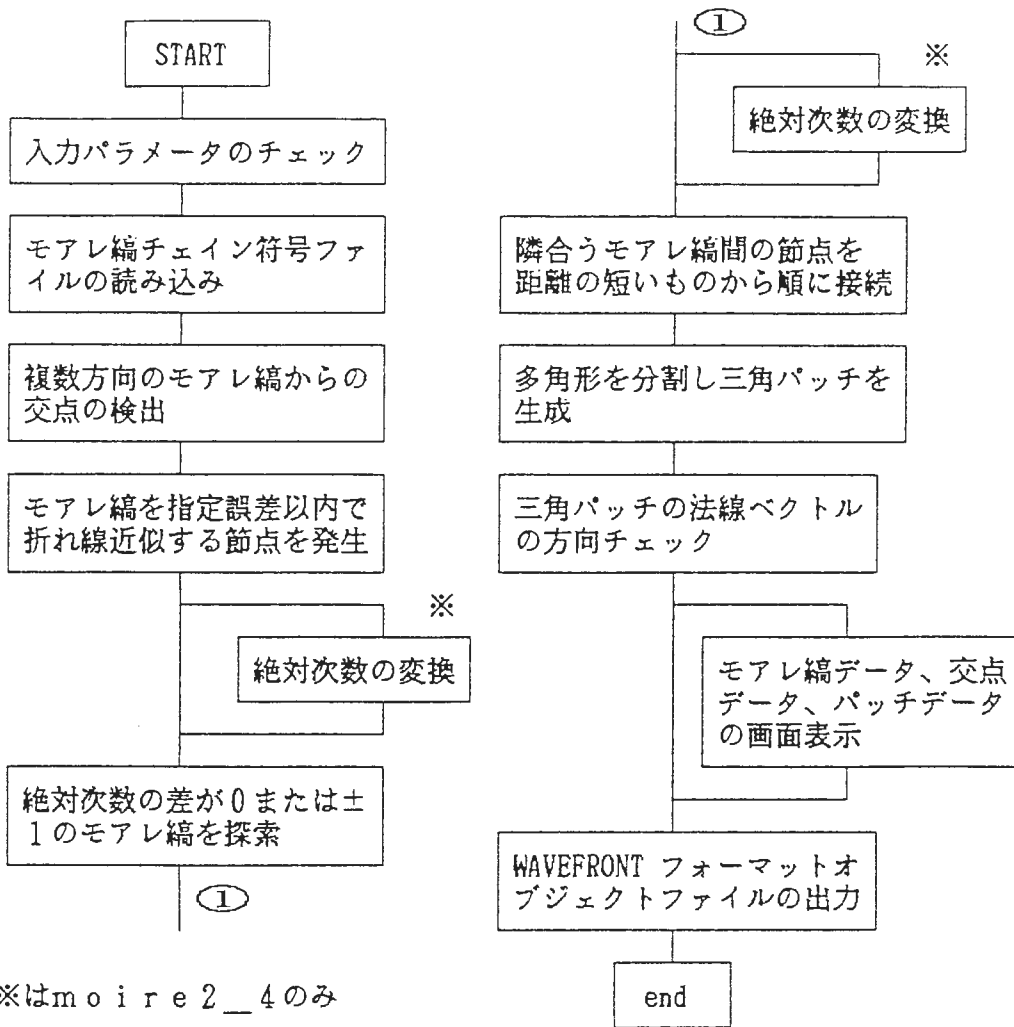


図7-4 moire2及びmoire2_4の処理の流れ

(2) 起動例 (キーボード入力、moire2_4の場合)

```
moire> moire2_4 ohface12.wav ohface12.obj -rl
```

```

*-----*
|           Moire2_4           |
| = make wareframe-object from moire-chain-data = |
|           Version 1.0       |
|           four times as dense as moire2         |
*-----*
  
```

- 1. Input file-name : ohface12.wav
- 2. Output file-name : ohface12.obj
- 3. Graphic flag : 0
- 4. Rotate flag : 1
- 5. Check flag : 0
- 6. Error value of (z) : 0.10
- 7. Max.length between verteces : 70.00

Read moire-chain-data...

<<< Input data >>>

```
1. File-name : ohface12
2. L : 2210.500000
3. d : 2210.500000
4. l1 & l2 & l : 18.000000 & 200.000000 & 218.000000
5. a & a' : 94.693481 & 88.680000
6. fg & fc : 90.000000 & 100.000000
7. beta : 5.000000
8. p0 & ps : 0.050000 & 0.050000
9. Rx & Ry : 0.006800 & 0.006800
10. ml : 6
11. m : 1
12. DATE : 09/01/89
13. OBJECT : OHFACE12
```

Check cross-data & delete some moire-chain...

(warning) : nz=8 --> Zn(237003.39) > Z(2210.50)

(warning) : nz=9 --> Zn(210679.53) > Z(2210.50)

(warning) : nz=8 --> Zn(237003.39) > Z(2210.50)

Make vertex-data...

```
<ipn=419><ipn=967><ipn=57><ipn=381><ipn=441><ipn=63><ipn=850><ipn=127><ipn=123><ipn=21
9><ipn=57><ipn=814><ipn=80><ipn=697><ipn=92><ipn=187><ipn=377><ipn=247><ipn=2067><ipn=
98><ipn=79><ipn=584><ipn=315><ipn=79><ipn=183><ipn=149><ipn=134><ipn=126><ipn=954><ipn
=394><ipn=152><ipn=56><ipn=656><ipn=177><ipn=115><ipn=717><ipn=117><ipn=286><ipn=91><i
pn=666><ipn=407><ipn=515><ipn=465><ipn=671><ipn=584><ipn=620><ipn=1147><ipn=60><ipn=56
><ipn=734><ipn=422><ipn=622><ipn=634><ipn=625><ipn=364><ipn=443><ipn=358><ipn=249><ipn
=195><ipn=77><ipn=56><ipn=229><ipn=147><ipn=802><ipn=64><ipn=689><ipn=422><ipn=130><ip
n=59><ipn=353><ipn=332><ipn=179><ipn=55><ipn=57><ipn=317> Make patch-data...
```

Output data to object-file...



写真7-2 moire2_4の起動例

7-8. テクスチャ節点の生成 (textmap)

(1) 処理の流れ

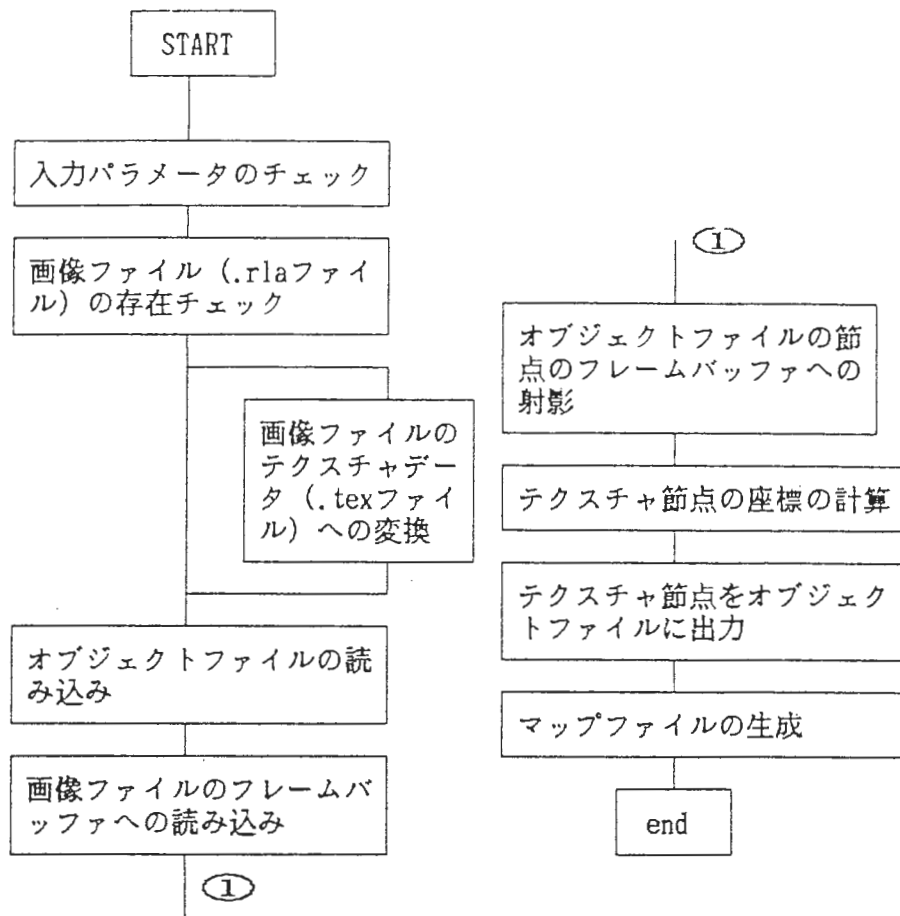


図7-5 textmapの処理の流れ

(2) 起動例 (キーボード入力)

```
moire> textmap:ohface12.obj:ohitest.obj:ohface_tex.rla:0:-on:ohface_tex.rla:180:-off:
```

```
*-----*
|           Texture Mapping           |
|                                     |
|           ---           Version 1.0           |
|                                     |
*-----*
```

```
read object file name : ohface12.obj
write object file name : ohitest.obj
```

```
image file ( 0 ) :      ohface_tex.rla  angle : 0.00
image file ( 1 ) :      ohface_tex.rla  angle : 180.00
```

```
+-----+
| Tex_bld_2D, version: 2.6  date: 1 Oct 1987 |
| copyright 1987, Wavefront Technologies, Inc |
+-----+
```

```
COLOR TEXTURE : ohface_tex.tex :
Image file    : ohface_tex.rla
```

Color samples : 3
Texture reso : 256

Texture file ohface_tex.tex created

fvxmin = -148.766403 fvxmax = 221.719161
fvymax = -293.956695 fvzmax = 274.645660
fvzmin = -9243.814450 fvzmax = -99.353012

object read inump 368 nvtx 358

```
***** no.0 *****  
read image file : ohface_tex.rla  
im_window :    0 511    0 511  
ac_window :   65 456 126 452
```

```
***** no.1 *****  
read image file : ohface_tex.rla  
im_window :    0 511    0 511  
ac_window :   65 456 126 452
```

write object
create map file

texture mapping terminated.

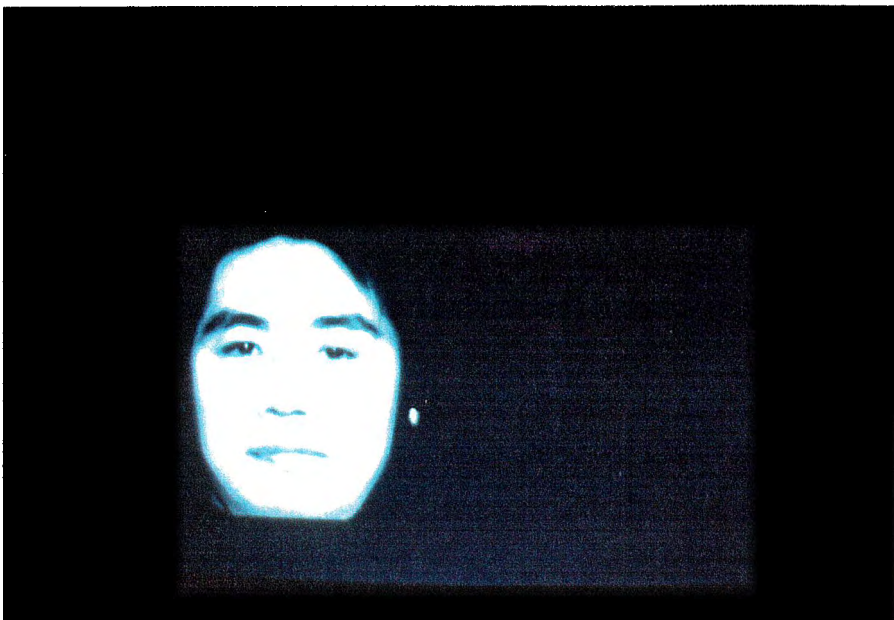
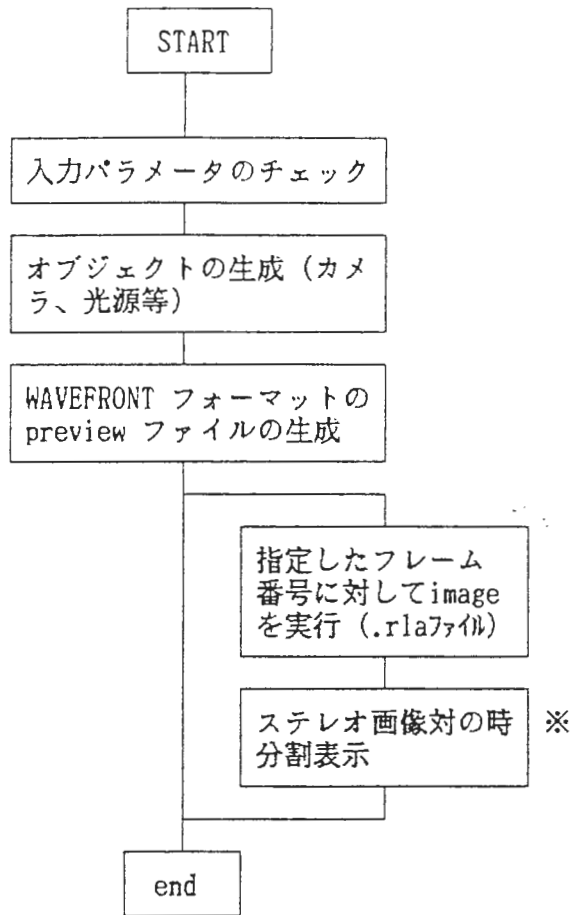


写真7-3 textmapの実行結果の例 (テクスチャ画像)

7-9. アニメーションデータの作成 (stereoview)

(1) 処理の流れ



※ステレオビューアの電源をONにしておくこと。センサーレベルを調節して同期させるとともに、ディレイを調節して左右の画像が正しく表示されるようにすること。

図7-6 stereoviewの処理の流れ

(2) 起動例 (キーボード入力)

```
moire> stereoview 0:ohctest.obj 0. 0. 0. L:white.lgt 500. 500. 500. P:ohctest.pv S:default.set f3:co500. cc60. ry120. --move1
```

```

*-----*
|               Stereoview               |
| = set cameras,lights etc... and make pv-file = |
|               Version 1.0               |
*-----*
  
```

<<< Input data - listing >>>

```

1. Object-data      : ohctest.obj( 0.00 0.00 0.00 )
2. PreView file-name : ohctest.pv
3. Setup file-name  : default.set
4. Light-data       : white.lgt( 500.00 500.00 500.00 )
5. Frame            : 3
6. Co               : 500.00
7. Cc               : 60.00
8. Xp               : 500.00
9. Animation-data   : tran( 0.00 0.00 0.00 )
                   : rota( 0.00 120.00 0.00 )
10. Light-move flag : 1
  
```

11. Image flag : 0(from 1 to 1 by 1)

Read setup-file...
Make commands for [Preview] ...
Make motion-data...
Run [Preview] ...
PreView 4.7 3030, Silicon Graphics
Copyright 1987 Wavefront Technologies

```
    ** Type '??' for pv commands **  
alias a alias  
alias bigcam cam -v 0 1 0 .75  
alias c cpic  
alias h history  
alias lgtcam cam -v .6 .8 .15 .35  
alias listl "cpic;csh ls *.lgt"  
alias listo "cpic;csh ls *.obj"  
alias listt "cpic;csh ls $WF_TEX_DIR"  
alias listw "cpic;csh ls *.wbin"  
alias normal "cam -r;setmon -60hz"  
alias .palette csh palette  
alias video "setmon -ntsc;o 1;cam -N ;recdev -vasd"  
alias workcam cam -v 0 1 .16 .7  
alias rokuga "pb -Qbfx"
```

```
pv>  
OBJECTS      : 5  
CHANNELS     : 31  
    ** Object 1, cam1, has 6 channels  
    ** Object 2, cam2, has 6 channels  
    ** Object 3, object, has 7 channels  
    ** Object 4, dummy, has 6 channels  
    ** Object 5, lgt, has 6 channels
```

File header written !
All frames will be loaded into core !

File completed !
File version = 4.70 (New Format)
ohtest.pv Opened 5 Objects 3 Frames - Window range 1 to 3

```
pv> pv> pv> pv> pv>  
pv> pv> Data modified from frame 1 to frame 3, step 1  
pv> pv> Data modified from frame 1 to frame 3, step 1  
pv> pv> Data modified from frame 1 to frame 3, step 1  
pv> pv> pv> pv> pv>  
pv> pv> Data modified from frame 1 to frame 3, step 1  
pv> pv> Data modified from frame 1 to frame 3, step 1  
pv> pv> Data modified from frame 1 to frame 3, step 1  
pv> pv> pv> pv> pv> Data modified from frame 1 to frame 3, step 1  
pv> pv> Data modified from frame 1 to frame 3, step 1  
pv> pv> Data modified from frame 1 to frame 3, step 1  
pv> pv> Data modified from frame 1 to frame 3, step 1  
pv> pv> Data modified from frame 1 to frame 3, step 1  
pv> pv> Data modified from frame 1 to frame 3, step 1  
pv> pv> closing ...
```

PreView file ohtest.pv is modified !
@@@ PreView running @@@ --> status=0

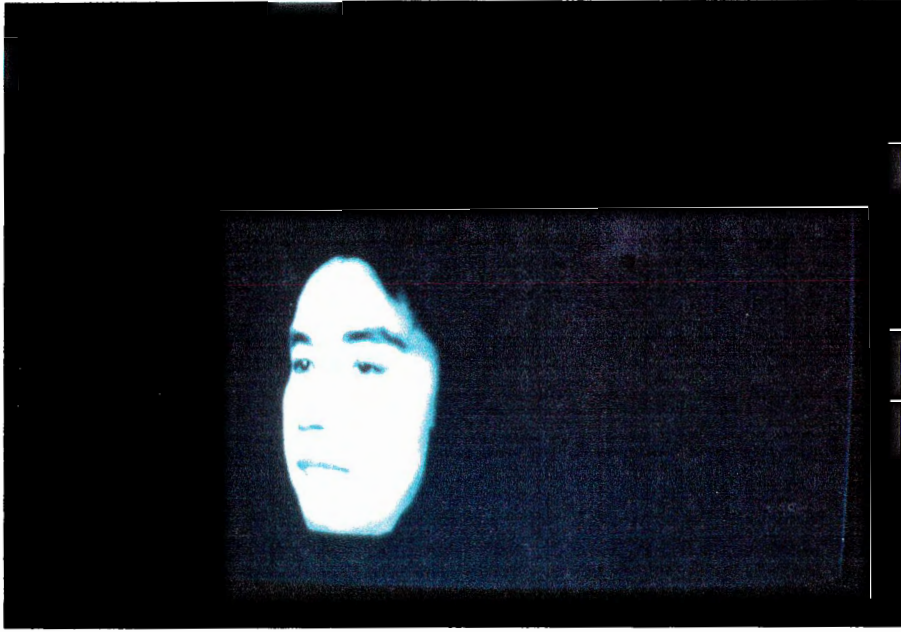


写真7-4 stereoviewの実行結果の例（右目画像を表示）

7-10. アニメーション画像の録画 (video)

(1) 処理の流れ

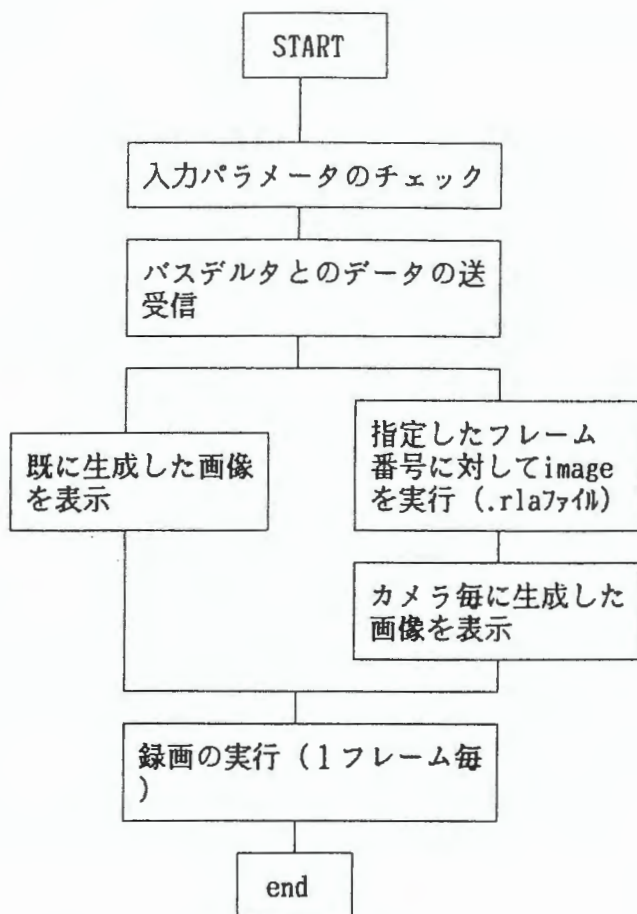


図7-7 videoの処理の流れ

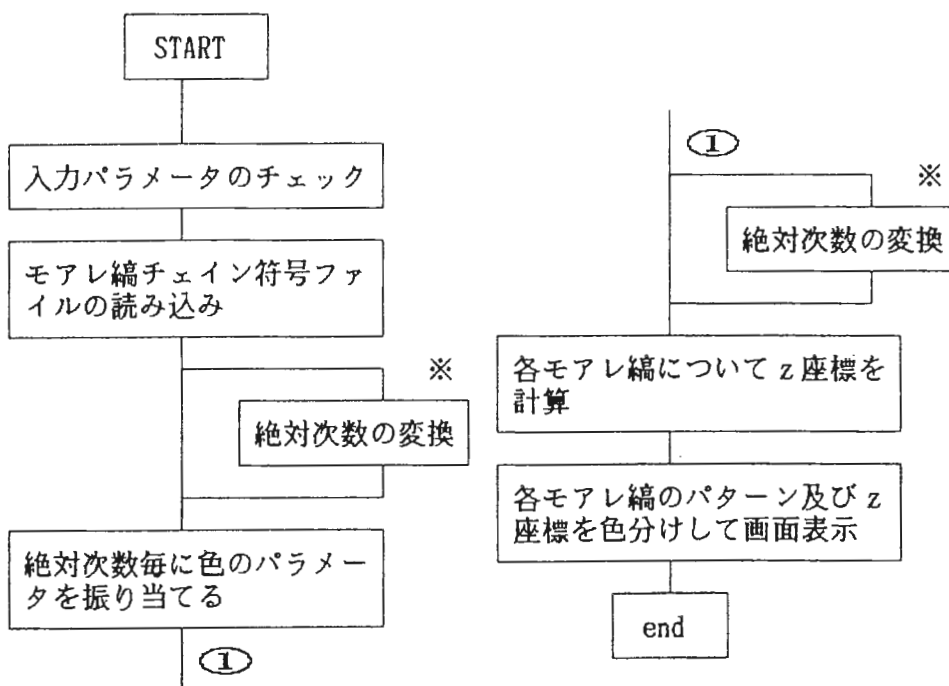
(2) 起動例 (キーボード入力)



写真7-5 videoの起動例 (BVH-2500のモニタ画面)

7-12. モアレ縞の画面表示 (dspmoire及びdspmoire_4)

(1) 処理の流れ



※はdspmoire_4のみ

図7-8 dspmoire及びdspmoire_4の処理の流れ

(2) 起動例 (キーボード入力、dspmoire_4の場合)

```
moire> dspmoire_4 ohfacel2.wav <CR>
```

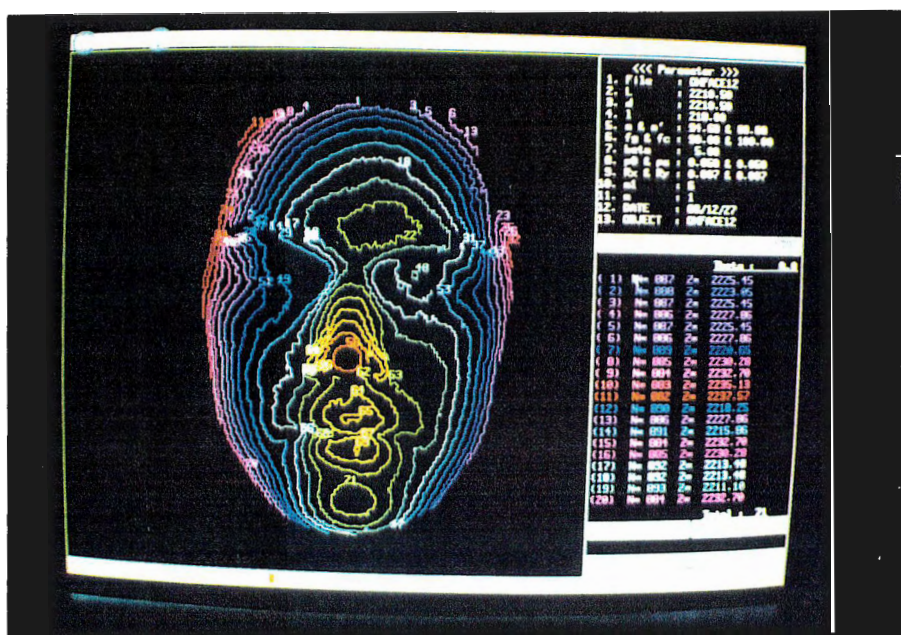
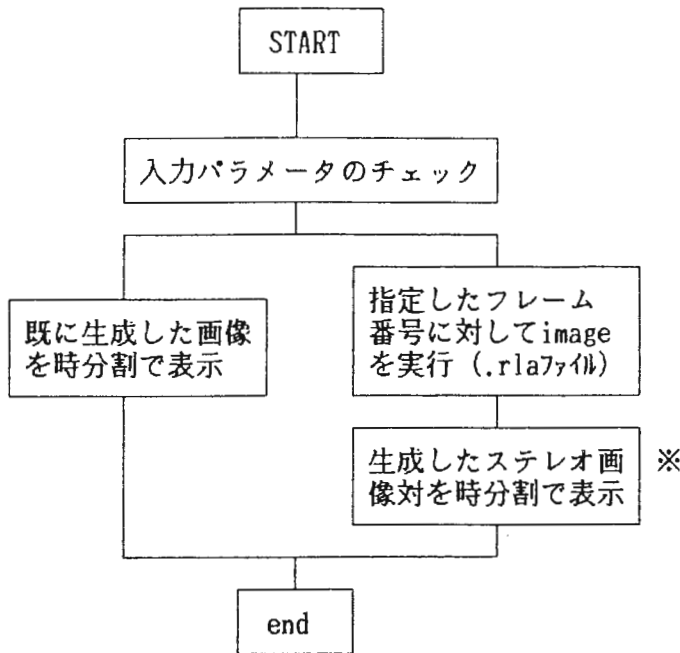


写真7-6 dspmoire_4の起動例

7-13. ステレオ画像対の表示 (dspimage)

(1) 処理の流れ



※ステレオビューアの電源をONにしておくこと。センサーレベルを調節して同期させるとともに、ディレイを調節して左右の画像が正しく表示されるようにすること。

図7-9 dspimageの処理の流れ

(2) 起動例 (キーボード入力)

```
moire> dspimage R:r-ohf.0002.rla l-ohf.0002.rla N
```



写真7-7 dspimageの起動例 (ステレオ画像対の表示)

7-14. セットアップファイルの作成 (mksetup)

(1) 処理の流れ

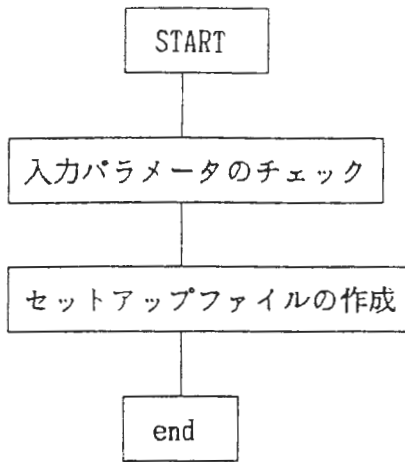


図7-10 mksetupの処理の流れ

(2) 起動例 (キーボード入力)

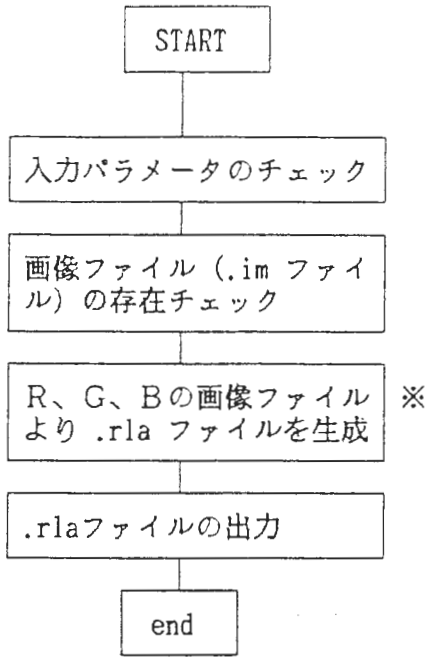
```
moire>mksetup test.set o2 13
```

```
*-----*
|               Mksetup               |
|   = make setup-file for [Stereoview] =   |
|               Version 1.0             |
*-----*
```

```
Object-parameter defined!!
Light-parameter defined!!
Write new setup-file...
```

7-14. 画像データの変換 (vicom)

(1) 処理の流れ



※モノクロ画像を変換する時はR、G、B全て同じ画像を用いれば良い。

図7-11 vicomの処理の流れ

(2) 起動例 (キーボード入力)

```
moire> ftp atr-cs
Connected to atr-cs.
220 atr-cs Wollongong FTP Server Process (Version 3.2) at Wed Mar 1 11:50-GMT+9:00
Name (atr-cs:koezuka): vicom
Password (atr-cs:vicom):
331 Enter PASS Command
230 User logged in, default directory DUAL: [VICOM]
ftp> bin
200 Command OK

ftp> cd DUAL:[VICOM.TEX]
200 Working directory changed to DUAL: [VICOM.TEX]

ftp> get OHFACE.TEX.IM ohface_r.im
200 PORT Command OK.
125 File transfer started correctly
226 File transfer completed ok
262144 bytes received in 2.6 seconds (99 Kbytes/s)

ftp> bye
221 Over and Out.
moire> cp ohface_r.im ohface_g.im
moire> cp ohface_r.im ohface_b.im
```



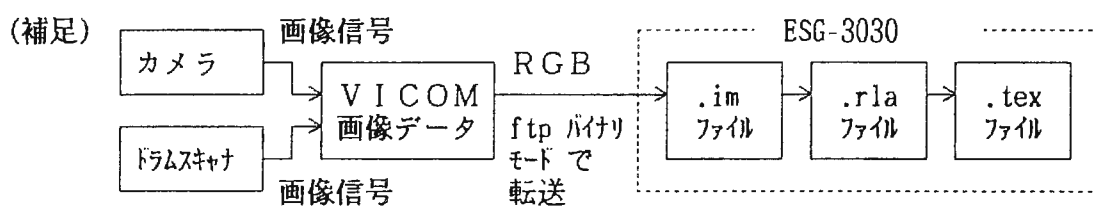
```
moire>vicom ohface.ria ohface_r.im ohface_g.im ohface_b.im
```

```
*-----*  
|           Data Conversion           |  
|           RGB file --> image file (.rla)           |  
|           Version 1.0               |  
*-----*
```

```
iumin : 126 iumax : 452 ivmin : 65 ivmax : 456
```



写真7-8 vicomの起動例



A P P E N D I X

A-1. モアレ縞生成システムにおけるVAX版とSUN版との相違点

モアレ縞生成システムは元々VAX/VMS上のVAX-FORTRANにおいて開発したものであり、VICOMのハードウェアの移行(VDP⇒VME)に伴いSUN上に移植した。表A-1に両者の機能的な相違点についてまとめた。なお、VAX版システムはVICOMのハードウェアがもはや接続されていないのでVICOMで画像処理を行う部分以外についてのみ使用可能である。

表A-1 モアレ縞生成システムにおけるVAX版とSUN版との相違点

相 違 点	V A X 版	S U N 版
メッセージ出力	日本語	英語
V I C O M画像処理	ハード....VICOM-VDP ソフト....VICOM コマンドコンバータ 詳しくはソースプログラムリストを参照すること。	ハード....VICOM-VME ソフト....VME ライブラリ 詳しくはソースプログラムリストを参照すること。
細線化結果画像	1画素2バイト出力 VICOM-VDPフォーマット	1画素1バイト出力 VICOM-VMEフォーマット

A-2. VAX-FORTRANからSUN-FORTRANへの移植上の注意点

SUN-FORTRANは 'with VMS Extensions' ということ、プログラムをほぼそのまま移植できる予定であった。しかしながら実際にはさまざまな問題点が生じかなりの手直しを余儀無くされた。ここでは、今後同様な問題が生じる場合を想定して移植に際して発生した問題点と対策について現在までに判明した分を以下にまとめておく。

表A-2 VAX-FORTRANからSUN-FORTRAN

への移植において発生した問題点と対策

項 目	手 法	
FORTRANからCの関数を呼び出す。	SUN	<p>Cで書かれたプログラムの関数名に__ (アンダーバー) をつける。</p> <p>ex. test.f abc.c</p> <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="border: 1px solid black; padding: 5px; text-align: center;"> . . call abc </div> <div style="border: 1px solid black; padding: 5px; text-align: center;"> abc_ { </div> </div>
	VAX	__を付ける必要はない。
FORTRANからCにデータを値渡しする。	SUN	<p>integer data はできる。 ex. %def(1)</p> <p>real data はできないのでアドレス渡しにする。</p>
	VAX	<p>real、integer 共にできる。</p> <p>ex. %def(1)、 %def(1.1)</p>
最大継続行数。	SUN	20行まで
	VAX	無制限
日本語の使用。	SUN	日本語のメッセージの出力ができない(未だサポートされていない?)。
	VAX	日本語のメッセージの出力ができる。
real data を添字に利用する。 real*4 a . . k = kk(a)	SUN	<p>できない。</p> <p>k = kk(int(a)) とする。</p>
	VAX	できる。
INCLUDE文	SUN	INCLUDE 'XXXXXX'
	VAX	INCLUDE '(XXXXXX)'
画像データの読み込み	SUN	UNIX 低水準読み込み (C)
	VAX	ダイレクトアクセスによるデータ読み込み (FORTRAN)
画像データの出力	SUN	UNIX 低水準出力 (C)
	VAX	ダイレクトアクセスによるデータの出力 (FORTRAN)

A-3. VICOM上でのユーティリティプログラム

モアレ縞生成に関してVICOM上に作成した各種ユーティリティプログラム（チェーンファイル）についての簡単な説明を掲げる。実行方法は、

- (1) `cs12` にログインする。
- (2) `cd /usr/moire/demo/prog<CR>` を入力する。
- (3) `cim` を起動する (`cim<CR>`) 。
- (4) `cal` プログラム名<CR>を入力する。

表A-3 VICOM上でのユーティリティプログラム

プログラム名	概 略 機 能
<code>hipint.vc</code>	カメラからの画像を入力する動作を<CR>を入力する毎に25回繰り返す。画像はイメージメモリの1に格納される。カメラレンズのピント合わせ等に用いる。
<code>vertchck.vc</code>	カメラからの画像を入力する動作を<CR>を入力する毎に25回繰り返すと共に、画面の中央部に垂直方向の赤いラインを重ねて表示する。カメラからの画像はイメージメモリ1に、赤いラインは2に、重ね合わせた画像は3に、それぞれ格納される。 カメラの姿勢の調節、基準格子の傾き調整等に用いる。
<code>tes.vc</code>	カメラからの画像を入力する動作を<CR>を入力する毎に2回繰り返す。画像はイメージメモリの1に格納される。顔画像の入力（変形格子像、テクスチャ）に用いる。
<code>moire6.vc</code>	モアレ縞生成のプログラム。イメージメモリ1の画像に対して、変形格子像を抽出しモアレ縞の生成を行う。光学系の調整・顔のモアレ縞生成に用いる。
<code>moiredem.vc</code>	モアレ縞生成のデモ用プログラム。ウサギの置物を対象物としたモアレ縞の生成及び細線化画像の表示を行う。