

はじめに

- ・ 教育用計算機室のアカウントを取得すること。
- ・ 新しく変更されたパスワードを忘れないようにすること。
- ・ 実験に関する情報のやりとりは電子メールで行うことが多いので、電子メールを使えるように設定しておくこと。また、メールはよくチェックすること。
- ・ unix に関連する情報は、大島教授のホームページ “ 教育活動のページ ” に掲載されているので、参考にさせていただきたい。また、“ 学内向け連絡・教材のページ ” には授業に関する注意事項が掲載されているので、なるべく目を通しておくこと。

<http://carrot.isl.tosho-u.ac.jp>

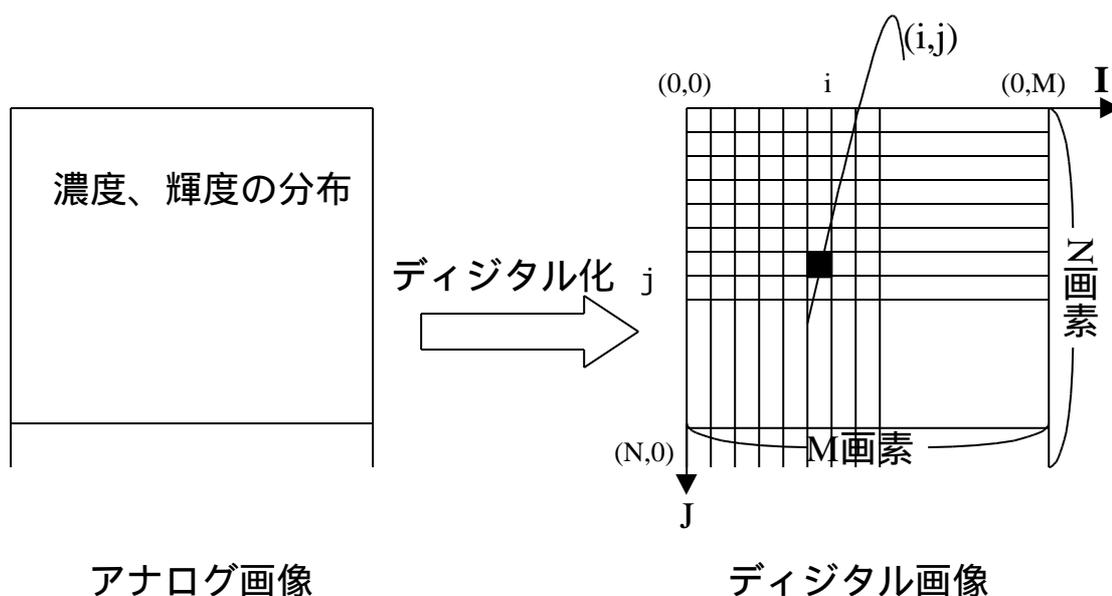
1 演習問題

1.1 デジタル画像の基礎

1.1.1 デジタル画像

デジタル画像とは、画面を格子状に分割し、その各格子（画素）における濃淡の値（画素値）を整数値で表現したものである。

分割の方法として、画像処理においては以下に示すような正方形格子が考えやすく、最もよく利用される。



1.1.2 2次元配列

今回は画像を 2 次元配列として取り扱う。2 次元配列は、平面上位置を縦と横の各座標値で表現するものである。ただし、デジタル画像の原点 $(0,0)$ の位置は

左上隅にあり、横軸は右方向、縦軸は下方向に正の値をとるものとする。デジタル画像を 2 次元配列として考えた場合、任意の画素 (i,j) の画素値はいくつである、という。画素値とは注目している画素における濃淡値を示す。

プログラム上では、2 次元配列は、gazou[j][i]のように表現される。(配列名は任意) i,j の値が座標を表し、gazou[j][i]が画素値を表す。

通常、j を縦座標、i を横座標として扱い、ループによって各画素にアクセスする場合は、2 重のループを用い、内側のループで横座標、外側のループで縦座標をまわす。これをラスタ走査という。

また、今回画素値は濃淡画像を用いるので、黒 灰色 白の連続した変化である。これを 8 ビット、すなわち 0~255 の 256 階調に変換して出力している。

1.2 演習課題「雑音除去」

1.2.1 画像における雑音

雑音という言葉は、元々何か目的があり、それを邪魔するものの総称として使われている。テレビのアンテナの調子が悪いと画像が乱れて見えにくくなる。このような状態を画像の劣化という。画像の劣化には次の 2 種類のものがある。

- ・画像が歪んだり、ぼやけたりする劣化
- ・画像上に邪魔な斑点、あるいは線などが現れる

後者の邪魔物を画像における雑音 (ノイズ) という。このような雑音には次のような性質がある

- ・画像上ランダムに雑音が見れる
- ・雑音の大きさが不規則である

このような雑音は画像に最も多く現れ、時々刻々位置も大きさも変化する。ビデオカメラで暗いところを撮影すると画像がざらついた感じになるが、これはランダム雑音のある画像の一例と考えられる。

画像で何らかの処理を行う際、雑音は計測結果の誤差や、雑音の大きさによってはプログラムの誤作動を引き起こす場合もある。このことから、画像から雑音を除去する作業は重要なものである。

1.2.2 平滑化による雑音除去

平滑化とはどのような処理だろうか。物体と物体、あるいは物体と背景との境目が輪郭となる。よって「画像の濃度に急な変化があるところ」が輪郭に見える、ということが出来る。境界線で濃度の変化が大きければ大きいほど、輪郭線がはっきり現れる（このような輪郭線をエッジとよぶ）。平滑化とは、このような急激な濃度変化をなだらかな変化に変え、画像をぼかす処理である。

雑音が見れている部分は、周囲の画素値に比べて急激な変化が見れているために目障りなのである。つまり、画像をぼかしてしまえば、細かい雑音は見えなくなるだろう、という発想が平滑化である。

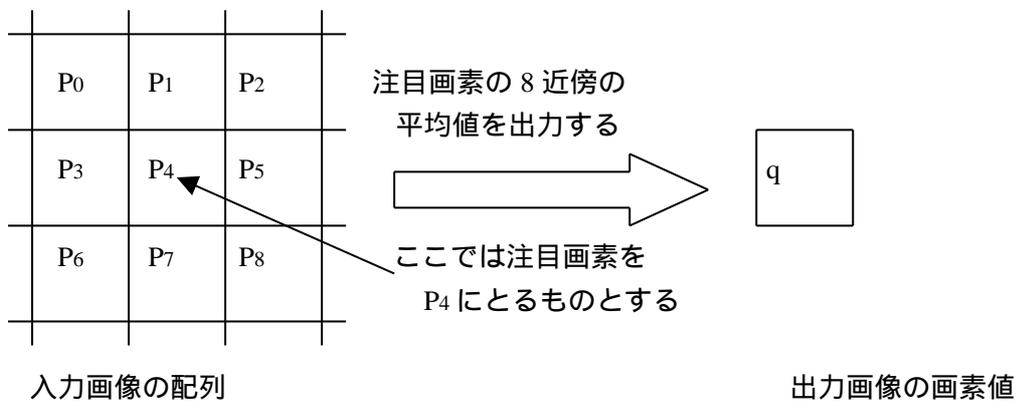
ただし、単純にぼかすだけでは雑音のみならず、目的の画像までぼけてしまう。平滑化において肝心な点は、いかにして雑音だけを除去するか、である。

1.2.3 平滑化

平滑化の処理で、よく用いられるものは次の 2 つである。

加算平均法

注目している画素と、その画素の周囲 8 近傍、計 9 つの画素の画素値の平均値を、出力画像の画素値とする。

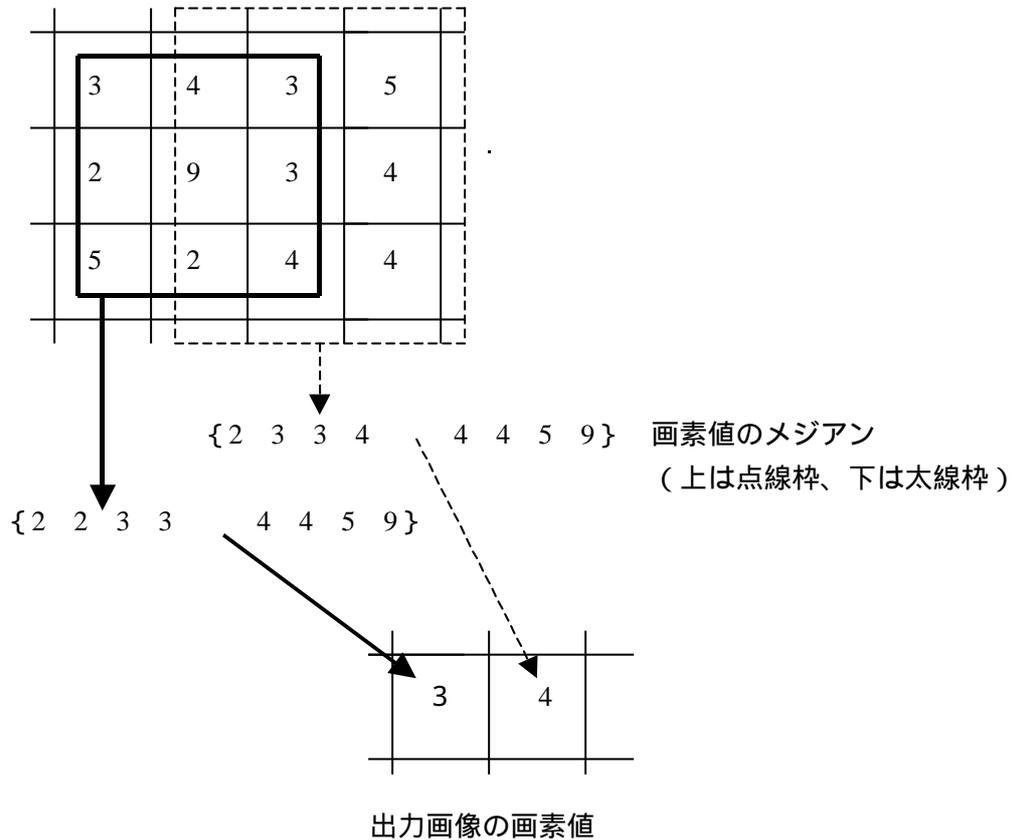


出力の画素値

$$b = \frac{a_1 + a_2 + a_3 + a_4 + a_5 + a_6 + a_7 + a_8 + a_9}{9}$$

メジアンフィルタ

注目画素近傍の画素値のメジアン（中央値）を出力画像の画素値とする。例として、以下のような入力画像の画素配列が与えられているとする。



1.2.4 サンプルプログラム

サンプルプログラム「sample.c」は、720 * 486 サイズで、雑音が含まれた状態の画像を取り込んで表示するものである。

今回の実験においては、プログラム中の「平滑化のアルゴリズム」を作り替えることで、雑音除去を行った画像を出力するようになっている。

簡単に変数と配列の説明をする。

JMAX,IMAX	入力画像の縦サイズ、横サイズ
input_data[JMAX][IMAX]	入力画像のデータ（画素値 0～255）
output_data[JMAX][IMAX]	出力画像のデータ（画素値 0～255）
i,j,k,l,m,n	int 型の変数、for ループなどに使用

もし変数が他に必要なら、初期設定の部分に必要な型で宣言すること。

サンプルプログラムは

```
make
```

でコンパイルを行うと「sample」という実行ファイル名ができるので

```
sample 画像ファイル名
```

と打ち込むことによって実行する。

サンプルプログラム内（アルゴリズム）で行われている処理は

- 1 . for(j=0 ; j<JMAX ; j++) {
for ループを用いて縦方向（J 方向）の 0 から JMAX までの走査を行う。

- 2 . for(i=0 ; i<IMAX ; i++) {
for ループを用いて横方向（I 方向）の 0 から IMAX までの走査を行う。
1,2 行は画像処理を行う上で最も基本となる「画像の走査」を行うもので、実際の走査の動きは原点から始まり、右端に到達後、1 列下の左端に戻り、再び右方向に進むといった流れを繰り返す。

- 3 . output_data [j] [i] = input_data [j] [i]
input_data （入力画像の配列）の濃淡情報を、output_data（出力画像の配列）にそのまま与える。

- 4 . } 2 行目のループの最後

- 5 . } 1 行目のループの最後

1.2.5 演習課題

- ・サンプルプログラムの「平滑化のアルゴリズム」の部分を 1.2.3 で説明した 2 通りの方法を用いて平滑化を行い、表示するように作り替える。必ず結果が出るように完成させること。
- ・雑音除去に関してはこれだけではなく他にも様々な手法があるので、それらについて図書館等で調べ、どのように雑音除去について役立てられているかを考察にまとめるのもよい。
- ・余裕があれば、それを実際にプログラムする。

ただし、サンプルプログラムを直接編集するのではなく、平滑化の種類ごとわかるようにファイルをコピーして編集すること。

画像の出力

現在のところ教育用計算機室で画像をきれいに出力する方法が見つかっていないので、以下のような方法で出力画像ファイルを研究室に送ること。

- 1 . 電子メールに出力画像ファイルを添付して、mhira@ipc.tosho-u.ac.jp宛に送信する
- 2 . HD、MO などのメディアに出力画像ファイルを保存して研究室にもってくる

結果、考察に必要な画像は、ファイルを受け取った時点から翌日、あるいは2日後にはこちらで出力しておくので、研究室まで取りにくること。(期限は特に設けないが、レポートの提出期限が実験終了後2週間であることを考慮し、早めに届けること。)

作成したソースプログラムは、以下の方法で印刷可能なので、必要な場合は各自で出力すること。

`lpr -P プリンタ名 ファイル名`

(例：`lpr -P1goukan 1 sample.c`)

プリンタ名：1goukan 1 (113 教室内)

1goukan 2 (113 教室内)

shin2goukan 1 (新2号館4階)

shin2goukan 2 (新2号館4階) などとする。

注意：xwd ファイルは決して出力しないこと！(何らかの対策をとらない限り、プリンタが半永久的に動き続ける。)

2 レポートについて

1.4 作成方法

レポートは以下のような構成にする

- 1 . 表紙 (自分で適当に作成、または配布資料の表紙を使用してもよい。)
- 2 . 実験の背景、目的
平滑化がどのようなことに役立つかなど。
- 3 . 平滑化のインプリメント (プログラム化)
実際にどのような仕組みでプログラムが動いているかを説明する。実際にプログラムの、平滑化のアルゴリズムの部分を用いし (印刷したもの、または手書きで写したもので可) 1 行 1 行どんな作業を行っているかを説明すること。また、新たに変数を宣言した場合は、その変数が何を表しているかも説明すること。
- 4 . 実験結果
それぞれのプログラムを実行して得られた結果を xwd を用いて取り込んで変換した PS ファイルを印刷し、レポートに添付すること。
- 5 . 考察
 - ・ 結果について考察しなさい。
 - ・ 雑音の除去にはこの他にどのような方法が考えられるか。
 - ・ その他疑問に思った点を考察しなさい。

< レポートに関する注意事項 >

プログラムが動き、正しい結果が得られて初めて合格の基準に達する。そのためレポートではしっかり動いているということをアピールするように書くこと。

途中経緯や、失敗例等は貴重な成果となるので積極的にレポートに取り入れること。

レポートは、図や表を積極的に活用し、読む人間に理解してもらうということを念頭に、わかりやすく書くこと。

考察は、感想文や体験文ではない。自分で問題を設定し、考察する能力を身につけること。

単にプログラムリストや結果を綴じただけのものや、項目に欠けがあるものはレポートとして認めがたいので受け取らない。

他人のレポートや、プログラムの丸写しが確認された場合、いずれの側も減点とする。

表紙だけでの提出は認めない。プログラムの完成が間に合わなくても最低限書けるところがあるはずなので、それらを書いた上で提出時に相談すること。時間割の都合上、課題を消化するためには実験時間だけでは不足するのが普通である。空き時間などを利用して課題には十分時間をかけること。

1.5 提出場所・期限

提出場所：(新 2 号館隣) 大学院棟 3 階「情報システム実験室」前に箱を設置するので入れておくこと。

提出期限：実験終了後 2 週間以内

1.6 再レポートについて

レポートに不備があった場合、提出したその場、あるいは掲示で再レポートとする。可否の結果は提出後 1 週間以内に情報システム実験室前に張り出される。再提出の期限は初回レポート受け取り後 1 週間以内とする。

1.7 質問等について

大島教官、本多技官、TA の学生 (平澤、李) まで直接問い合わせるか、下記までメールすること。

(新 2 号館隣) 大学院棟 3 階 情報システム研究室

教授	大島 正毅	oshima@ipc.tosho-u.ac.jp
技官	本多 健二	khonda@ipc.tosho-u.ac.jp
TA	平澤 雅人	mhira@ipc.tosho-u.ac.jp
	李 高	gli@ipc.tosho-u.ac.jp

参考書籍

本実験に於いては以下の書籍を参考にするとより理解が深まる。ただし、レポートに明らかにこれら書籍からそのままコピーしたと思われるものが発見された場合、再レポートとなるので注意すること。

- プログラミング言語C B.W.カーニハン/D.M.リッチー著 共立出版
- C言語によるはじめてのアルゴリズム入門 河西朝雄著 技術評論社
- UNIXプログラミング環境 B.W.カーニハン/R.パイク著 アスキー出版局
- X-window Ver.11 プログラミング/木下凌一・林秀幸著 日刊工業新聞社
- MAKEの達人 C.トンド/A.ネイサンソン/E.ヤント著 トップラン
- 画像処理標準テキストブック 財団法人 画像情報教育振興協会

2 UNIX の使い方

2.1 基本的なコマンド

- ls 自分が現在いるディレクトリ（カレントディレクトリ）の内容（ファイルやディレクトリ）を表示させる。
- pwd カレントディレクトリ名を表示させる。
- cd 別のディレクトリへ移動させる。
使用方：cd ディレクトリ名
ただし、ディレクトリ名を入れず cd だけだと自分のホームディレクトリに戻る。
- mkdir ディレクトリの新規作成。
使用法：mkdir 新規ディレクトリ名
- cat, more（テキスト）ファイルの内容を表示させる。
more の場合は、一画面ずつ表示させる。
使用法：cat もしくは more ファイル名
- cp ファイルをコピーする。
使用法：cp 元ファイル名 新ファイル名（もしくはそのファイルを置きたいディレクトリの場所（パス））
- mv ファイルの移動及びファイル名の変更。
使用法：mv 元ファイル名 新ファイル名（もしくは移動先ディレクトリ名）
- rm ファイルの削除。
使用法：rm ファイル名
- rmdir 空ディレクトリの削除。
使用法：rmdir 消したいディレクトリ名
- ln リンクを張る。
使用法：ln -s リンク元のディレクトリ名 リンク先のディレクトリ名
- man マニュアルコマンド。コマンドのマニュアルを表示する。

使用法 : man コマンド名

2.2 エディタの使い方

テキストを作成したり、編集修正をしたり、ファイルに入出力するためのソフトウェアをエディタという。ここでは代表的なエディタである mule についての使い方を載せる。

Mule の起動

mule ファイル名

カーソルの移動

1 語上に移動 CTRL+p

1 語左に移動 CTRL+b

1 語右に移動 CTRL+f

1 語下に移動 CTRL+n

行頭に移動 CTRL+a

行末に移動 CTRL+e

文頭に移動 ESC+<

文末に移動 ESC+>

1 画面分前に移動 ESC v 1 画面後ろに移動 CTRL+v

指定した行に移動 CTRL+o 行数

指定した文字列に移動 CTRL+S ESC 文字列

文字列検索 CTRL+s 文字列

削除

1 文字削除 CTRL+d

1 行削除 CTRL+k

"CTRL+k"によって最後に削除されたテキストを取り込む CTRL+y

直前の状態に戻す(undo) CTRL+_

改行

ポイントの後ろを何行か改行 ESC 数字

ポイントの後ろを 1 行改行 CTRL+m

置き換え

文字列置換 ESC % 置換対象となる語 置換する語

置き換えて次へ進む SPACE

置き換えるが移動しない ,

置き換えずにスキップ DEL

残りをすべて置き換える !

直前に置き換えた位置に戻る ^

再帰編集に入る(ESC-CTRL-c で抜ける)	CTRL-r
終了	ESC
ウィンドウ	
ウィンドウを全画面に	CTRL+x 1
ウィンドウ分割	CTRL+x 2
バッファ	
バッファ切り替え	CTRL+x b <u>バッファ名</u>
バッファを閉じる	CTRL+x k <u>バッファ名</u>
ファイル操作	
ファイル挿入	CTRL+x CTRL+i <u>ファイル名</u>
別ファイル読み込み	CTRL+x CTRL+f <u>ファイル名</u>
ファイルのセーブ (重要)	CTRL+x CTRL+s
別ファイル書き出し	CTRL+x CTRL+w <u>ファイル名</u>
mule の終了 (重要)	
	CTRL+x CTRL+c

2.3 コンパイルと実行

作成した C のプログラム名を `***.c` というように最後に「.c」と付くようにする(***には自由に名前を付けてよい)。作成したプログラムを実際に行う(使えるように)させるためには「コンパイル」という作業を行う必要がある。コンパイルには、いくつかの方法がある。もっとも基本的な方法は、「`gcc ***.c -o ***`」と打つことである。

エラーがでてくれば、そのエラーメッセージを参考にして修正し

エラーが出てこなくなったら、「*** (必要ならファイル名も)」と打つことでプログラムを実行できる。

ここで、コンパイルをする際には、エラーが出なかったがうまくプログラムが動かない、望みの結果が得られないといった場合にはデバッガ(下参照)を用いることによって解消することができる。

今回は、画像表示に UNIX のウィンドウシステムである「X-Window」を使用するため、複数のライブラリーを読み込む必要がある。そのため、Makefile を用いたコンパイルを行う。Makefile の基本となるものは用意してあるので、その 1 行目に書かれているファイル名の情報をコンパイルしたいファイルのものに変更して、「make」と打ち込むことでコンパイルすることができる。

参考文献 MAKE の達人 C.トンド/A.ネイサンソン/E.ヤント著 トッパン

デバッガ

少し複雑なプログラムを作る場合、予期せざる誤りを取り除く操作(デバッグ)を完成するまで何度も繰り返すことになる。文法上の誤りは gcc (C コンパイラ) が指摘してくれるが、文法上は正しくても思い違いなどから求める結果が得られないことが

よくある。このようなとき、途中の経過（どの段階で変数の値がいくらになっているかなど）をプリントさせてみたりすると（また、もし可能ならプログラムの実行を途中で止めてみる）プログラムの動きを確認でき早くデバッグできる。そのためにプログラムに本来は必要ない printf 文（時として入力文も）などを挿入することも行われる。これも役立つが、もっと便利なやり方がある。それがデバッガである。

デバッガの使い方

コンパイルするとき gcc に代えて gcc -g とする（これをデバッガオプション付のコンパイルという）。デバッガオプション付のコンパイルを行うと多少実行速度が遅くなるが、できあがった実行プログラム（a.out）をそのまま通常通り実行することもできる。

デバッグモードの実行は次のように行う。

コマンドが実行できる状態（コマンド行）で dbx a.out と入力する。実行プログラムが読み込まれデバッグモードとなる。この状態で種々の dbx のコマンドが入力できるようになる。

dbx コマンドの主なものは：

help	dbx コマンドとしてどんなものがあるか教えてくれる。help に続けて dbx コマンド名を入れれば dbx コマンドの使い方を教えてくれる。
run	プログラムを開始する。どこか途中で止まっているときはご破算でやりなおしとなる。
list	ソースプログラムを番号付で表示する（この番号をよく使う。なお、番号付のリストを手元に置きたいときは cat -n コマンドを利用できる）。
print	指定した変数の値を表示する。
stop at	指定した番号に制御が移るとき実行直前で止まる。
stop in	指定した関数に制御が移るとき実行直前で止まる。
step	止まっているとき、1 ステップだけ次に進む。
next	step と似ているが、step は実質的な 1 ステップ次に進む（例えば関数呼び出しの直前で止まっているとき関数の中に入って止まる）が、next はソースプログラム上で次へ行く（関数の中へ入り込まないで、形式的にその行を実行する）。
cont	止めたプログラムを継続する（従って次に stop at や stop in に会うまでどんどん進む）。
dump	使っている変数の一覧を表示する。
quit	デバッガを終了する。

デバッガを使いこなすとプログラムのデバッグが早くできるし、プログラムの働きを深く理解できる。なお、システムに備えられている関数はデバッグモードでコンパイルされていないので、中に入って見ることができない。また gcc コンパイラにおい

では、main プログラムが意図するように表示されないことがあるが、その場合、main プログラムの名前を MAIN に変え、形式的に設けた main プログラムから呼び出すようにすればよい。(注意、デバッグモードでの実行はスピードを除きほぼ完全に通常の実行と同じと考えてよいが、ごくまれに結果がことなることがある(似たような事情は実行速度を上げるためのオプティマイゼーション付コンパイルでも存在する。))

2.4 電子メールの概要と使い方

電子メールとはコンピュータ上で電子の郵便のやりとりをすることで使い方は、「mail 宛先のユーザ名(アドレス)」で出すことができる。届いたメールを読む場合には、「mail」とだけ打つことにより読むことができる。

実験や UNIX 全般についての質問は電子メールでも受け付ける。積極的に利用することが好ましい。質問先のアドレスは 2.4 を参考にすること。

現在 cad システムにおいて漢字の入力方法は整備されていない。情報処理センターから漢字を含むメールのやり取りができる。

また、上記いずれのシステムともインターネット接続されている(今日の常識)ので、簡単にデータの転送ができる。

2.5 Xwindow システムの遠隔使用

今日の計算機はインターネット接続されているのが普通である。インターネット接続されている計算機同士は、さまざまな形で、データをやり取りできる。電子メールはインターネット環境以前から存在するが、今日ではインターネットを前提としている。

ある計算機から別の計算機を使うには telnet (rlogin) コマンドが、またファイルを送受するには ftp (rcp) コマンドが使われる。

Xwindow システムには、サーバとクライアントという概念がある。モニタの管面に図形等を表示したり、マウスやキーボードのデータを読み込む計算機をサーバという。管面に表示するためのプログラムの動いている計算機をクライアントという。サーバとクライアントは同一計算機であってもよいし、別の計算機であってもよい。

Xwindow システムを用いる遠隔のある計算機 A (サーバ・クライアントになっている) で動かせるとき、手元の計算機 B で表示したいものとしよう。手元の計算機 B の上で xhost A というコマンドを入れると、自分がサーバになって指定したクライアント計算機からのアクセスを認めることを意味する。次に telnet 等で計算機 A にログインし、

```
setenv DISPLAY 計算機 B の IP アドレスまたはネットワーク上の名前 : 0.0
```

をすると、以後ログアウトするまで Xwindow 関連の入出力は計算機 B に対してなされる。

上記の操作は、cad 室内の計算機同士で容易に行えるので、試してみるとよい。また、

ほかのところにある Xwindow の使える計算機（パソコンでもよい、インターネット接続されていることが必要）からも可能なので、機会があれば試してみるとよい。この手法を習熟すれば、かつては困難とされていた、遠隔地間での高度なやり取りも容易に行うことができる。

© 200 1 Joho System lab. Tokyo Univ. of M.M.

2001 年 4 月 13 日	2001 初版
2001 年 4 月 27 日	2001 第 2 版
2001 年 5 月 18 日	2001 第 3 版

編集者 平澤 雅人、李 高

監 修 大島 正毅、本多 健

二、長谷川 為春