◇ソフトウェア開発◇

動的構造解析装置DIP3000の画像データ処理プログラムIPDA

吉田 孝*

Data processing program IPDA for image data of DIP3000

Takashi Yoshida*

1．はじめに

X線を使用して物を見ると言うことは想像以上に大変な作業である。とはいえ，ここ10数年の間にはるかに短時間に解析が進むようになった。その最大の理由はコンピュータの進歩であろう。コンピュータの高速化と記憶容量の大容量化は目覚しいものがある。さらに，コンピュータの小型化と低価格化によって，「自宅でちょっと解析」ということが可能になった。

X線を観測する手段には，シンチレーションカウンターと写真フィルムが一般的である。シンチレーションカウンターとは，放射線（荷電粒子，X線，ガンマ線）が当たると蛍光を発する物質を利用してこの光を光電効果によって電気信号に変えるものである。他方，写真フィルムは古くから使われてきたが最近になって，X線を観測する別の方法として，イメージングプレートが開発された。

イメージングプレート（以下，IPと略す）はシンクロトロン放射光等の強度の強いX線を有効に利用するために開発された。IPは，BaFBr:Euという輝尽発光体からできている。X線が照射されると，荷電子帯や不純物Euの準位から電子が励起され伝導帯に持ち上げられる。伝導体の電子は格子欠陥にトラップされて準安定な状態に落ち着く。このトラップされた電子の量がX線照射量に比例する。これがIPへの記録である。この準安定状態に落ちている電子にHe-Neレーザー（波長633nm）等の光を照射すると，伝導帯に励起され，価電子帯やEuの準位に落ちる。この過程で，Photo Stimulated Luminescence (PSL) という光が放射される。この光の強度を測定することでX線の照射量が分かる。これがIPからの再生である。IPの最大の特徴は，100ミクロンの分解能と6桁にもおよぶダイナミックレンジである。

センター設置のマックサイエンス製のワイセンベルグ型IPカメラDIP3000においては，1枚のIPに記録するために，4400×2600ピクセルで，16bit/pixelの記録を行っている。1回の記録のために約23MBのデータの記憶領域が必要である。結晶を回転させて全回折強度を測定したり，あるいは温度等の条件を変える測定をすると，このサイズのファイルが10～30個は作成されることになる。

DIP3000にはマックサイエンス製XPRESS，MAX-DENZOといったソフトウエアが用意されている。XPRESSはX線，ゴニオメータ，IPの移動や消去等のシステムの制御のほかに測定したIP写真の階調表示，3次元ネット表示，数値データの表示やプロファイルの表示が可能である。MAX-DENZOは方位決定，反射の指数付け，積分強度の摘出，複数のファイルのスケーリング，マージング，データの表示が可能である。これらは，全てWINDOW環境によるマウス処理システムである。WINDOWのシステムは初心者には分かりやすいが，数10枚にわたる画像ファイルに対し同じ場所を正確にカットしたり，プロファイルの作成は大変な作業である。ドットデータや角度データを逆格子に変換するといった作業は，標準のシス

*山口短期大学（防府市台道）
Yamaguchi Junior College
E-mail：yoshida@yamaguchi-jc.ac.jp

テムではできない。また，データ処理のために専用のコンピュータが必要になると効率が悪い。そこで，市販のコンピュータによってデータの処理ができ，さらに数10枚のデータファイルを連続して処理できるソフトウェアを作成することにした。

IPの情報を処理するためには，どのようなソフトウェアが必要であろうか。まず，測定強度を読み取った結果で出力される数値データを画像データに変換する必要がある。次に，ピーク形状をプロファイルとして数値データに変換する。また，ピークの位置と強度を数値データとして出力する必要がある。実験では一般的に，温度，外部電場，外部磁場等の条件を変えて測定することが多い。そこで，ピーク位置，強度，ピーク形状（含むFWHM）におけるそれらの依存性を1つのファイルにする必要があろう。今回，これら一連のプログラムを新規に作成した。ここでは，それらプログラムの基本的な使用方法から，実際にチオ尿素の解析について述べる。

2．プログラミング環境

今回プログラムを作成するにあたって，まずコンセプトを明確にした。

(1) マルチプラットホームに対応する。
(2) スクリプトによる連続自動処理が可能なシステムとする。すなわち，会話型やウインドウタイプの処理にしない。
(3) 個々を可能な限り小さくまとめメンテナンスを容易にするとともに，信頼性を確保する。

私は20年ほどプログラムに携わってきた。せっかく作成したプログラムも環境が変わると使用できなくなる。プログラムの書き換えという膨大な無駄な時間が必要になる。そこで1 つの解答を得た。それは，固有の機種及びOS に依存するコードは書かないと言うことである。それは，WindowsAPIも同様である。

同様のことがコンパイラにもいえる。言語は何を使用すべきか。コンパイラは何を使用すべきか。将来変更すべき個所が発生した場合，容易に変更できるのか。変更箇所は独立なユニットにできるかがキーである。

プログラムの形態には，会話処理型とバッチ処理型（非会話型）処理がある。旧来の大型コンピュータにおけるプログラミングのようなコマンドライン形式では，バッチ処理型のプログラムが作成されてきた。コンピュータの高速化と低価格化によって，従来では，想像もつかなかったソフトウェアがワークステーションやパーソナルコンピュータ上で動作している。それに伴い，多くのソフトウェアが会話処理型に書き換えられて来た。特に，近年のWindowsは会話型処理しか考慮されていない。会話型処理はシステムが実行中に条件を変更できるのが最大の特徴である。逆に，実行条件をログとして記録しない限り，どのような条件で処理を実行したのか分からなくなる。また，システムの実行中は席を離れることができない。人がいなければ動かないシステムといえる。バッチ処理の優れた点は全ての条件を与えた上で処理を行うことである。これにより，処理条件の記録と処理の再現が可能になる。さらに，システムのバックグラウンドでの実行や，自動処理が可能である。以上のことから，スクリプト形式のバッチ処理が可能なシステムを採用した。

今回のプログラムには，実は前身となるプログラムがあった。これはNEC PC-98シリーズのDOSシステムで動作するプログラムである。完成までに半年以上を費やした。これは，画面上でラウエ写真等を表示し，会話形式でピークのプロファイルを作成するものである。このプログラムの機能を強化し，DIP220，DIP3000の画像を処理することができた。しかしながら，機能の強化は，プログラムサイズの肥大化であった。この約1年後にIBM PC/ATのコンピュータが市場に出回る。しかも，Windowsシステムである。当初，このプログラムの書き換えを試みたが，時間の無駄であることに気付いた。また，PC-UNIXのシステム構成の素晴らしさにも驚いた。そこで，メンテナンスを容易にするためにもOSからの切り離しと，個々を可能な限り小さくまとめることによって，信頼性を最優先するプログラム作りに切り替えた。それがこのプログラムである。

今回，DIP3000用にデータ処理システムを新規にソースプログラムから書き起こした。名称は，Imaging Plates Data Analysis system (IPDA)

とした。

3. 各コマンドの意味と基礎知識

処理プログラムは，C言語のプログラムをコンパイルした基本プログラムとそのスクリプトファイル（バッチファイル）で構成される。C言語のソースファイルとそのプログラムの概要を表1に示す。コマンドの説明は，以下の表記方法で示した。

command［入力1 ］［入力2 ］…

［］で囲まれた入力変数は，必須である。全プログラムに共通した変数名として，xi，yiは始点の座標（dot）である。また，xn，ynは画像のサイズ（dot）であり，終点座標ではない。拡張子を指定していないfileはIPのファイル名であり，拡張子は（.ipf）である。ファイル名の前の＋は，アペンディング（追加記録）を示す。よって新規にファイルを作成する場合，このファイルが無いことを確認する必要がある。

3.1 tipcut.c

IPファイルを必要な部分を切り取り新規にIPファイルとして作成する。

tipcut [oxn] [oyn] [file1] [xi] [yi] [xn] [yn] [file2]

oxn：既存IPファイルのX軸方向のサイズ

oyn：既存IPファイルのY軸方向のサイズ

file1：既存IPファイル

xi,yi：切り取りを開始する座標（x,y）

xn,yn：切り取るドットサイズ（終点座標ではない）

file2：新規IP ファイル

〈実行例〉

tipcut 4400 2600 foo.ipf 0 0 2200 1300 new.ipf

3.2 tipcon.c

IPファイルを粗視化（平均化）して新規にIPファイルとして作成する。粗視化数coによりco×co ドットを平均して1 ドットにする。データサイズは，1/（co×co）になる。

tipcon [xn] [yn] [file1] [co] [file2]

co: 粗視化数

xn,yn: 入力ファイルのサイズ

file1: 既存のIPファイル名

file2: 新規のIP ファイル名

〈実行例〉

tipcon 4400 2600 foo.ipf 2 new.ipf

2×2ドットを平均して1ドットにする。データサイズは，1/4になる。

3.3 tip2bmp.c

IPファイルをグレースケール256階調のビットマップ（Windows）に変換し，新規ファイルを作成する。

表1. C言語のソースプログラム

Source	概　要
tipcut.c	IP ファイルを必要な部分を切り取り新規にIP ファイルとして作成
tipcon.c	IP ファイルを粗視化して新規にIP ファイルとして作成
tip2bmp.c	IP ファイルをグレースケール256 階調のビットマップ(Windows)に変換し新規に作成
tip2bmpc.c	IP ファイルをカラーフルスケールのビットマップ(Windows)に変換し新規に作成
tipwcut.c	IP ファイルからワイセンベルク写真の2θ-ω積分線にそってデータを切り取り新規にIP データを作成
tip2asc.c	IP ファイルのY軸方向を積算し，X軸(dot)に対する強度をアスキーデータにする
tipascx.c	IP ファイルのX軸方向を積算し，Y軸(dot)に対する強度をアスキーデータにする
wnpre.c	dot データから，逆格子に変換するための準備データを計算する
tdot2wn.c	IP ファイルのX軸(dot)から，逆格子に変換する
twnpeek.c	データから，ピークを求める
twnfwhm.c	データとピークから，FWHMを求める

tip2bmp [xn] [yn] [file.ipf] [min] [max] [file.bmp]
min: この値以下は，白となる
max: この値以上は，黒となる
file.bmp: 新規ビットマップファイル（Windows版）

〈実行例〉
tip2bmp 4400 2600 foo.ipf 0 200 new.bmp

3.4 tip2bmpc.c

IPファイルをカラーフルスケール32bitのビットマップ（Windows）に変換し新規にファイルを作成する。
tip2bmpc [xn] [yn] [file.ipf] [min] [max] [file.bmp]
min : この値以下は，黒となる
max : この値以上は，赤となる

3.5 tipwcut.c

IPファイルからワイセンベルク写真の2θ-ω積分線にそってデータを切り取り新規にIPファイルを作成する。
tipwcut [oxn] [oyn] [file1] [xi] [yi] [xn] [yn] [fs] [fn] [ls] [file2]
fs: スタート角
fn: エンド角
ls:IP shift length[/0.1mm]

3.6 tip2asc.c

IPファイルのY軸方向を積算し，X軸（dot）に対する強度をアスキーデータにする。出力ファイルの拡張子をここでは（.dot）とした。
tip2asc [yi] [xn] [yn] [file1.ipf] [file2.dot]

3.7 tipascx.c

IPファイルをX 軸方向を積算し，Y軸（dot）に対する強度をアスキーデータにする。
tipascx [xn] [yn] [file1.ipf] [file2.dot]

3.8 wnpre.c

dotデータから，逆格子に変換するための準備データを計算する。
wnpre [start_angle] [end_angle] [casset move] [H] [X] [Y] [K1] [X1] [Y1] [GAMMA]
start_angle : ωのスタート角度（deg）
end angle : ωのエンド角度（deg）
casset move : カセット移動距離（mm）
H : (H 0 0)
X,Y : 座標（1347,1044）
K1 : (H K1 0)
X1,Y1 : 座標（1337,556）
GAMMA（deg）: 結晶の格子定数のγ

〈実行例〉
wnpre 70 90 120 6 1347 1044 1 1337 556 90

3.9 tdot2wn.c

IPファイルのX軸のドットデータを，逆格子に変換する。出力されるファイルはアスキーデータであるが，ここでは拡張子（.wn）とした。
tdot2wn [start_angle] [end_angle] [casste move] [H] [X] [Y] [b*] [omega*] [file1.dot] [file2.wn]
start_angle : ωのスタート角度（deg）
end_angle : ωのエンド角度（deg）
casste move : カセット移動距離（mm）
H : (H 0 0)
X,Y : 座標（1347,1044）

3.10 twnpeek.c

データから，ピークを求める。このプログラムの出力ファイルは，アペンディングする。
twnpeek [Temp] [file1.wn] [+file2.pk]
Temp : 温度
file1.wn : 逆格子変換されたプロファイルのデータ
file2.pk : 出力するピークファイル

3.11 twnfwhm.c

データとピークから，FWHMを求める。このプログラムの出力ファイルは，アペンディングする。
twnfwhm [Temp] [file1.wn] [file2.pk]

[+file3.fw]
Temp : 温度
file1.wn 逆格子変換されたプロファイルのデータ
file2.pk ピークファイル
file3.fw 出力するFWHM ファイル

4. スクリプトの作り方

オリジナルデータ (foo1.ipf, foo2.ipf, foo3.ipf) から始点座標 (900,0), サイズ (2600,2600) を切り取る。さらに, 2×2サイズの粗視化を行いビットマップデータにするスクリプトファイルを作成する。このファイル中で%1と%2は入力変数で, IPファイルと出力するビットマップファイルである。

[filename=mybmp.bat]
```
tipcut 4400 2600 %1 900 0 2600 2600 tmp1.ipf
tipcon 2600 2600 tmp1.ipf 2 tmp2.ipf
tip2bmp 1300 1300 tmp2.ipf 0 200 %2
del tmp1.ipf
del tmp2.ipf
```

[mybmp の実行]
```
mybmp foo.ipf foo.bmp
```

さらに, 全データを自動で実行するスクリプトファイルは, 以下のようになる。

[filename=autobmp.bat]
```
call mybmp foo1.ipf foo1.bmp
call mybmp foo2.ipf foo2.bmp
call mybmp foo3.ipf foo3.bmp
```

4.1 ipf2bmp.bat

IPファイルの座標XI,YIからサイズXN,YNを切り取りBMPマップファイルに変換する。

ipf2bmp.bat [file.ipf] [XI] [YI] [XN] [YN] [file.bmp]

5. 実例

実際にチオ尿素のサンプルを使用して解析方法を述べる。ここでの目的は, 超格子反射の指数(040)での強度の温度依存性を調べることにする。そこで, この指数でのピークプロファイルを作成し, 温度と強度, 温度とサテライトの逆格子のシフト量, 温度とピークのFWHM (半値幅) を算出する。まず, DIP-3000 によるラウエ写真とワイセンベルグ写真を測定した。

5.1 準備

準備として次の測定を行った。データファイルがすでにあるものとする。

(1) 室温におけるラウエ写真
IP Data File: d293.ipf

(2) 低温のワイセンベルグ写真
測定条件は, a軸ビーム入射, c軸回転の写真であり, 対象指数(020), (040), (060), (061)を測定するためにωのスタート角70度, エンド角90度, 及びカセット移動距離120mmとした。IPファイル名の数字は, 測定温度を示す。ここでは, 120Kから200Kまで測定している。
IP data File: d120.ipf, d130.ipf, d140.ipf, d150.ipf, d160.ipf, d170.ipf, d180.ipf, d190.ipf, d200.ipf

5.2 写真の出力

まずはじめに測定した写真をビットマップファイルに変換する。これにより, 画面に表示させたり, 印刷が可能になる。IPのデータは4400×2600ドット (約23MB) あるので, 必要な部分にカットしたり粗視化を行う。

＜ラウエ写真の場合＞
写真の中央を正方形に切り取ることにする(図1)。

```
tipcut 4400 2600 d293.ipf 900 0 2600 2600 tmp1.ipf
tipcon 2600 2600 tmp1.ipf 2 tmp2.ipf
tip2bmp 1300 1300 tmp2.ipf 0 200 d293.bmp
```

＜ワイセンベルグ写真の全体写真の場合＞
```
tipcon 4400 2600 d160.ipf 5 t2.ipf
tip2bmp 880 520 t2.ipf 0 200 d160.bmp
```

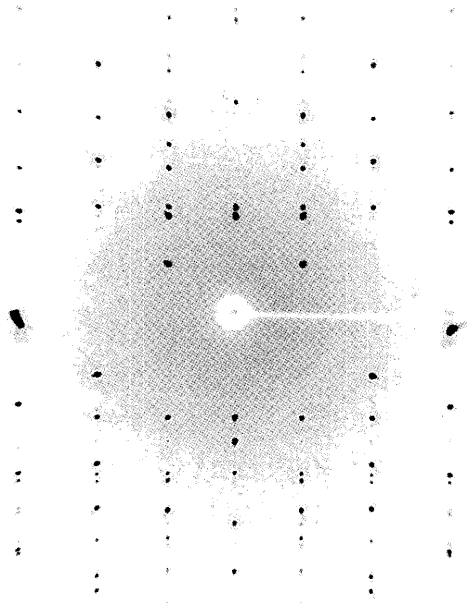

図1. ラウエ写真

＜ワイセンベルグ写真の部分写真の場合＞
指数（040）を中心に（020),（060),（061）をカバーする範囲1200ドット四方を切り取り，ビットマップファイルにする（図2)。

ipf2bmp d160.ipf 1100 0 1200 1200 d160.bmp

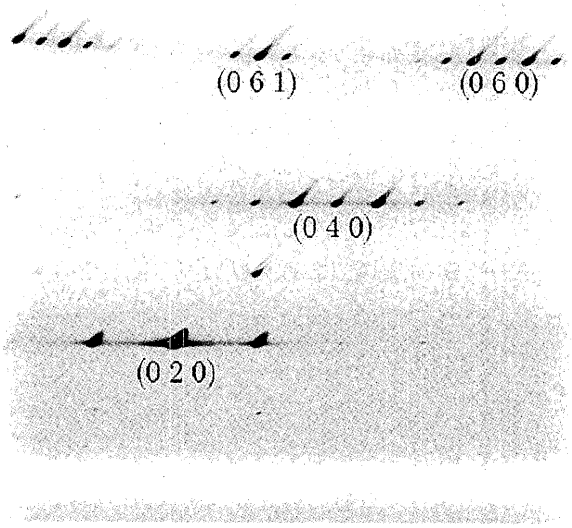

図2. ワイセンベルク写真

5.3 ピークポジションの取得とc*の計算

IPファイル上の観測したい指数のピークの座標をドットデータで確認する。ドットデータを逆格子に変換するために（060）と（061）を使用した。この２つの指数からc*を計算し，表２を作成する。
例）IP data: d120.ipf
ωのスタート角70度，エンド角90度，カセット移動距離120mm,（060）の（x,y）=（1347,1044),（061）の（x,y）=（1337,556），GAMMA=90degの時

wnpre 70 90 120 6 1347 1044 1 1337 556 90

実行結果
beam center, camera radius = 2219.0 1226.0 1500.0
X -ray wave length = 0.710730
Start omega = 70.000000 Coupling= 6.000000
a = 7.439837 omega = 73.033333
2theta = 33.689919 omega = 81.166664
—b*=0.118001 —

表2. ピークポジションとc*

file	(040)	(060)	(061)	c*
d120	1640,709	1347,1044	1337,556	0.118001
d130	1641,706	1348,1042	1339,559	0.118074
d140		（以下　省略）		

5.4 各指数周辺における強度プロファイル

測定されたIPファイルは，xy座標（ドット）とその強度であるので，このファイルから縦軸が強度，横軸がドットを示す強度プロファイルを求める。ipファイルのｘ座標方向５ドット分の平均強度を縦軸にし，ｙ 座標方向が横軸となる。

＜(020) 付近のピークプロファイル＞
tipcut 4400 2600 d120.ipf 1927 0 5 1000 t1.ipf
tip2asc 5 1000 t1.ipf d120.dot

ここで，(020) のピーク位置が(x,y) = (1930,381) である。よって，前後５ ドット分の積算を考慮して，(xi,yi,xn,yn) = (1927,0,5,1000) とした。

5.5 強度プロファイルの逆格子変換

次に，横軸がドットのファイル（.dot）を逆格子（.wn）に変換する(図3)。ここではd120.ipfのファイルの指数(020)に注目している。

tdot2wn 70 90 120 2 1930 381 0.1180 90 d120.dot d120.wn

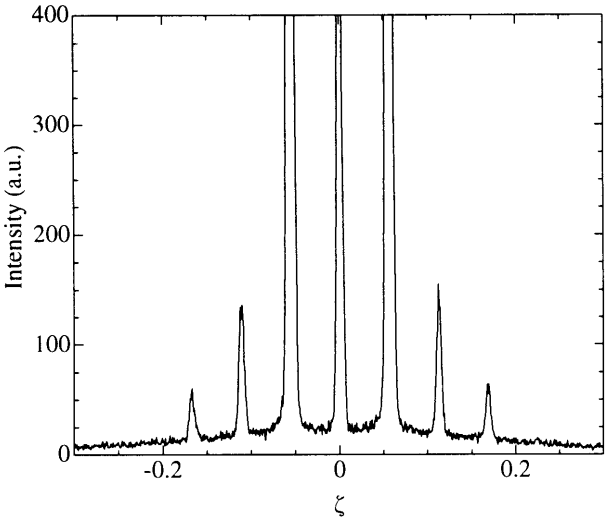

図3. ピークプロファイル

5.6 ピークサーチ

逆格子変換の強度プロファイルからピークポジションを探す。この時点では，バックグラウンドはすべて0になっている。

twnpeek 170 d170.wn d170.pk

そこで，ピークプロファイルから各ピークでのバックグラウンドを読みとり，エディタを使用して記入する必要がある。

5.7 FWHM

twnfwhm 170 d170.wn d170.pk d170.fw

このプログラムも出力ファイルは，アペンディングされる。入力ミスをした場合などは，d170.fwを削除して再実行する。

5.8 ピークサーチとFWHM

アペンディングモードで使用

twnpeek 170 d170.wn data.pk
twnpeek 180 d180.wn data.pk
twnpeek 190 d190.wn data.pk

back ground 補正

twnfwhm 170 d170.wn data.pk data.fw
twnfwhm 180 d180.wn data.pk data.fw
twnfwhm 190 d190.wn data.pk data.fw

6. おわりに

一般に，プログラムの作成には，多くの時間を要する。しかしプログラムを作成すれば，解析に要する時間を格段に減らすことができる。自分で処理プログラムを作成するということは，既存のソフトウェアでは出来ない処理か，または不満があるからであろう。自作であるか，商用ソフトウェアであるかは，問題ではない。他人の作ったプログラムが使い辛いのは，そのソフトウェアが優れたソフトウェアでは無いからである。

優れたプログラムとは，

(1) 処理が信頼できる。当然のことであるが，この当たり前の事ができないソフトウェアが意外に多い。
(2) システムのメンテナンス性と拡張性が優れている。肥大するソフトウェアは，信頼性を低下させる。
(3) 優れたユーザインターフェースを持つ。

今回のプログラムによって，当初の目的を満足し，実際にチオ尿素のデータ処理を短時間に行うことができた。しかし，これが最良の方法であるとは言い切れない。マルチプラットホームを確保しつつ優れたユーザーインターフェースを持たせるためにはどのような方法とるべきか，現在思案中であり今後の課題である。プログラムIPDA は，http://www.yamaguchi-jc.ac.jp/lab/yoshida/から入手できる。

謝辞

本プログラムの開発にあたり，増山博行教授（山口大学理学部），DIP3000の装置責任者の濱田亮氏（山口大学理学部，故人）にさまざまな助言を頂きました。心より感謝致します。

参考文献

1) (株) マック・サイエンス: DIP3000 システム取

2) T. Yoshida & H. Mashiyama: Metastable Commensurate Structure in Thiourea. J. Korean Phys. Soc. 32, pp.S920-S923 (1998.2).