```
----------------------------------------
Date:    Fri, 08 Jun 2012 14:32:17 +0900
Subject: Re: 32bit_float_in_tiff
----------------------------------------
```

うえすぎさま、

なかのです。TIFF 画像のファイルを作る「sif」の浮動小数点数（float）版を書いてみました。かなり手抜きです。パッケージ「hvd」用に書いた「ファイル上の画素値データのブロック化や圧縮なし」の sif のコードを書き換えました。

整数ではなく浮動小数点数をファイルに書きこむことを除けば以前の sif との違いはごくわずかです。値「IEEE 形式浮動小数点数」を保持する「sample format tag」をファイルに書き込むようにしただけです。

添付した書庫ファイル float.taz（gzip'ed TAR 形式ファイル；AIST で今年度から採用になった Google mail では ZIP 形式ファイルを添付できないため）に float 版の sif 関連の以下のファイル群を入れておきました。

ファイル Makefile
　説明省略
sif_float.h
　float 版の sif のヘッダファイル。関数 StoreImageFile_Float() と、それが使う以下の変数の型のデフォルトの値を定義している。

　FOM
　　float on memory。メモリ上で使用する浮動小数点数の変数の型でデフォルトは double（８バイト長の倍精度浮動小数点数）
　FOF
　　float on file。ファイルの上に格納する浮動小数点数の型でデフォルトは float（４バイト長の単精度浮動小数点数型）

sif_float.c
　関数 StoreImageFile_Float() のコード。その詳細は後で説明します。
error.c
　警告の表示とプログラムのアボート用関数 Error() のコード。
cell.h、rif.h と rif.c
　説明省略
t_i2f.c と t_i2f.exe
　普通の TIFF 画像（16 ビット以下の整数の画素値から構成される白黒もしくはグレースケール画像）を浮動小数点数の画素値の TIFF 画像に変換するプログラムのソースコードと Windows 用の実行ファイル。
　このプログラムの起動法は以下の通り。

```
t_i2f  TIFF_int  base  step  TIFF_float
```

　ただし、

　TIFF_int と TIFF_float
　　整数画素値と浮動小数点数画素値の TIFF のファイル名
　base と step
　　画素値の変換の際に用いる係数値。整数と浮動小数点数の画素値をそれぞれ i と f として、それらを以下の式で変換する。

$$f = base + step \times i$$

関数 StoreImageFile_Float() の呼び出し方は以下の通りです。

```
void  StoreImageFile_Float(
    char  *path,    /* TIFF のファイル名 */
    int  Nx,        /* 画像の横画素数 */
    int  Ny,        /* 画像の縦画素数 */
    FOM  **cell,    /* 浮動小数点数画素値へのポインタのポインタ */
    char *desc      /* コメント（不要なら NULL を指定すれば OK） */
);
```

型 FOM の画素値データの２次元配列（画素値データへのポインタのポインタ）である cell の初期化は以下のようなコードによって行えば良いです。

```
FOM     **cell;
...
if ((cell=(FOM **)malloc(sizeof(FOM *)*Ny))==NULL) exit(1)
for (y=0; y<Ny; y++) {
    if ((cell[y]=(FOM *)malloc(sizeof(FOM)*Nx))==NULL) exit(1);
    ...
    for (x=0; x<Nx; x++) {
        cell[y][x]=座標値 (x,y) の画素の値
    }
}
```

とりあえず以上です。t_i2f を走らせて、sif_float で作った TIFF ファイルが正しいかどうかを確かめて下さい。とり急ぎ、

```
----------------------------------------
Date:    Mon, 11 Jun 2012 12:23:04 +0900
Subject: new_ver._of_sif_float
----------------------------------------
```

うえすぎさま、

なかのです。先日紹介した float 版の sif（旧名：sif_float）用のファイル群を以下のように変えました。今後はここに書いたものを使おうと思っています。

(0)
文字列 "sif_float" は長すぎるので float 版の sif の総称を "sif_f" とすることにしました。これに応じて、ソースやインクルードファイルを改名しました。ただし、浮動小数点数の画素値をファイルに書く関数 StoreImageFile_Float() の名前は以前のままです。

(1)
新しいインクルードファイル "sif_f.h" の内容は以前に紹介した "sif_float.h" のものと概ね同じで、メモリ上に置く画素値の変数の型名「FOM」のデフォルト値（double）と、関数 StoreImageFile_Float() の呼び出し方を定義しています。

しかし、ファイル上に格納する際の画素値の型 FOF のデフォルト値（float）の定義は sif_f.h から削除しました。この定義は関数 StoreImageFile_Float() の複数のソースファイル（後述）のそれぞれに書き込みました。

(2)
以前に紹介したソースファイル "sif_float.c" を "sif_f_fast.c" に改名しました。その内容も少し変えましたが、関数 StoreImageFile_Float() の呼び出し方は以前のままなので問題にならないでしょう。

(3)
通常使っている "sif.c" に対応した、きちんとしたコード "sif_f_full.c" を書きました。この上の StoreImageFile_Float() の呼び出し方は "sif_f_fast.c" のものと同一ですが、以下が "sif_f_fast.c" のコードとは異なります。

[a] 画素値データを適当な長さの「strip」に分割し、かつ、それらの配置に関するインデックスを付けてファイル上に格納。

[b] 格納した画素値データの値域を示す TIFF の tags（SMinSampleValue と SMaxSampleValue）を追加。

[c] コンパイル時に "gcc -DLZW ..." のようにしてマクロ「LZW」を宣言

すれば LZW 圧縮した画素値データをファイルに書き込む。

(4)
sif_f のテストプログラムとして書いた整数の画素値を浮動小数点数の画素値に変換する t_i2f のソースコード "t_i2f.c" も少しだけ書き換えました。しかし、これをコンパイルした実行ファイルの起動法は以前とまったく同じです。

```
t_i2f  TIFF_int  base  step  TIFF_float
```

(5)
"Makefile" を以前のものから大幅に書き換えました。sif_f_fast.c もしくは sif_f_full.c を組み込んだ以下の３個の t_i2f の実行ファイルをコンパイルするようにしてあります。

t_i2f_fast (t_i2f_fast.exe)
: sif_f_fast.c を組み込んだ t_i2f の実行ファイル

t_i2f_full (t_i2f_full.exe)
t_i2f_lzw (t_i2f_lzw.exe)
: これらはいずれも sif_f_full.c を組み込んだもの。コンパイルの際にマクロ LZW を宣言しているので、t_i2f_lzw は画素値を LZW 圧縮する。

以上のファイルを添付した書庫ファイル float.taz に入れておきましたので、どうぞご利用下さい。とり急ぎ、

```
----------------------------------------
Date:    Mon, 11 Jun 2012 19:07:15 +0900
Subject: rif_f
----------------------------------------
```

うえすぎさま、

なかのです。rif の float 版（rif_f）を書いてみました。それに関連したファイル（＋以前に紹介したファイル）をこの E-mail に添付した gzip 圧縮した TAR 形式書庫ファイル float.taz に入れておきました。

ファイル rif_f.h
: rif_f のヘッダファイル。画素値データを読み込むメモリ上の変数の型 FOM のデフォルトの値（double）と、関数 ReadImageFile_Float() の呼び出し方を定義している。

rif_f.c
: rif_f のソースファイル。従来の rif.c を書き換えました。

error.c、cell.h、rif.h、rif.c、
sif_f.h、sif_f_fast.c、sif_f_full.c、t_i2f.c と t_i2f_*.exe（３個）
: 以前に紹介済み

sif.h と sif.c
: 従来の sif のヘッダファイルとソースファイル。

t_f2i.c と t_f2i.exe
: rif_f のテストプログラム t_f2i のソースファイルと Windows 用実行ファイル。以下のようにして起動すると以前に紹介した t_i2f_*.exe による処理の逆の処理（浮動小数点数の画素値の TIFF 画像ファイルを整数画素値のものに変換する処理）を実行します。

```
t_f2i  TIFF_float  {base  step}  BPS  TIFF_int
```

ただし、

TIFF_float と TIFF_int
: TIFF のファイル名で同じものを指定しても良い（これは t_i2f の場合も同様）。TIFF_float として整数画素値の画像ファイルを指定してもかまわない（これは t_f2i のみ）。

base と step
: t_i2f の場合と同様な整数画素値と浮動小数点数画素値の変換式の係数値。これらの指定を省略すると、TIFF_float の上の浮動小数点数画素値の最小値と最大値で正規化した整数画素値になる。

BPS
: TIFF_int の整数画素値のビット数。

Makefile
: 変換した結果の整数画素値を LZW 圧縮してファイルに書き込むような t_f2i をコンパイルする設定にしてあります。

sif_f とは異なり rif_f には１個の版しかありません。rif_f は以下の形式の TIFF 画像ファイル上の画素値のデータを型 FOM の２次元配列に読み込むことができます：

画素値データの圧縮形式
: 圧縮なしと Machintosh PackBit 圧縮もしくは LZW 圧縮に対応

画像のタイプ（photometric interpretation）
: 白黒、グレースケールとカラーマップ（CM）画像のファイルを処理可能。ただし、白黒とグレースケール画像上の画素値と表示輝度の対応関係を指す "MinIsBlack" もしくは "MinIsWhite" 属性や CM 画像のファイルの上に格納されている表示色の CM データは無視する。

画素値の容量
: 整数の画素値の場合は 32 ビットまで。浮動小数点数の画素値の場合は 32 ビット（FOF = float）と 64 ビット（FOF = double）に対応。

上に書いたように、sif_f は整数と浮動小数点数の画素値のデータの両方を取り扱うことができます。関数 ReadImageFile_Float() の仕様は以下の通りです：

```
void  ReadImageFile_Float(
    char  *path,    /* TIFF のファイル名 */
    int  *Nx,       /* 画像の横画素数を入れる変数を指すポインタ */
    int  *Ny,       /* 画像の縦画素数を入れる変数を指すポインタ */
    FOM  ***cell,   /* 画素値データを入れる配列を指すポインタ */
    char  **desc    /* コメントを入れる変数を指すポインタ */
);
```

この関数の使い方については t_f2i のソースコードを参考にして下さい。なお、これは従来の rif でも同様ですが、ReadImageFile*() の引数の「ポインタ」として NULL を指定することが可能です。

例えば ***cell に NULL を指定した場合、画素値のデータを読まずにファイル上の画像の横画素数 Nx などを調べることができます。

とりあえず以上です。

```
----------------------------------------
Date:    Wed, 13 Jun 2012 13:21:43 +0900
Subject: new_ver._of_rif_f
----------------------------------------
```

うえすぎさま、

なかのです。たびたびですみません。TIFF 画像ファイル上の画素値のデータが通常の整数値なのか浮動小数点数の値なのかを判断できるように、rif_f の関数 ReadImageFile_Float() の仕様を少し変えて整数値を返すようにしました。

```
int  ReadImageFile_Float(
    char  *path,    /* TIFF のファイル名 */
    int  *Nx,       /* 画像の横画素数を入れる変数を指すポインタ */
    int  *Ny,       /* 画像の縦画素数を入れる変数を指すポインタ */
```

```
    FOM  ***cell,   /* 画素値データを入れる配列を指すポインタ */
    char  **desc    /* コメントを入れる変数を指すポインタ */
);
```

返される整数値（int）は以下の通りです。

TIFF 画像ファイル上の画素値が浮動小数点数：０
整数画素値：１〜32（その画素値の容量のビット数）

この返り値の情報が不要なら関数 ReadImageFile_Float() に void のキャストを付けて呼び出して下さい（付けないとコンパイラが警告するのでウルサイ）。

```
(void)ReadImageFile_Float(path,&Nx,&Ny,&cell,&desc);
```

ファイル上に格納されている画素値の「種類」に応じて画像の読み込み後に値の補正を行う必要があるなら、以下のようにすれば良いでしょう。

```
if  ((BPS=ReadImageFile_Float(path,&Nx,&Ny,&cell,NULL))==0) {
    画素値が浮動小数点数の場合の処理
}
else {
    BPS ビットの整数画素値に対する処理
}
```

なお、メモリ上で浮動小数点数の画素値を格納する２次元配列の型 FOM はデフォルトで「double」なので、52 ビットまでの整数画素値を桁落ちなしで保持することができます。FOM を「float」に定義し直すとそれは 24 ビットまでになりますが、普通の画像なら高々 16 ビット画素値なので問題にならないと思います。

以上のような新版の rif_f のヘッダ（"rif_f.h"）とソースファイル（"rif_f.c"）をこの E-mail に添付した gzip 圧縮した TAR 形式書庫ファイル float.taz に入れておきました。また、新版の rif_f に応じた修正を加えたプログラム t_f2i のソース（"t_f2i.c"）と Windows 用の実行ファイル（"t_f2i.exe"）もそこに一緒に入れてあります。どうぞご利用下さい。

とりあえず以上です。

```
----------------------------------------
Date:    Wed, 20 Jun 2012 12:13:21 +0900
Subject: archive_files_for_sif_f+rif_f
----------------------------------------
```

うえすぎさま、

なかのです。先日紹介した sif_f_* や rif_f のソースコードなどを入れた２種類の書庫ファイル（内容は同じです）を www-bl20 にアップしておきました：

```
http://www-bl20.spring8.or.jp/~sp8ct/tmp/float.taz
http://www-bl20.spring8.or.jp/~sp8ct/tmp/float.zip
```

先日紹介した時から sif_f_fast.c のコードだけを少し変えましたが、関数 StoreImageFile_Float() を呼び出す側に影響はありません。とり急ぎ、

```
P.S.
    整数画素値の TIFF を取り扱う従来からの sif や rif に関しても上記の書庫ファイルに入っているソースコードが最新版です。念のため。
```

```
--------
2016/1/6
--------
```

Intel C++ compiler（icc）に最適化オプション "-fast" を指定してコンパイルすると "rif_f.c" の実行コードが誤動作するので、そうならないような C 言語コードに修正した。また、"sif_f_fast.c" と "sif_f_full.c" の C 言語コードも少しだけ書き換えたが、こちらには問題があったわけではない。これらの改変したソースファイルとそれらを用いてコンパイルし直した 32 bit Windows 用の実行ファイルなどを以前と同じ２個の書庫ファイルのそれぞれに再格納した。

```
http://www-bl20.spring8.or.jp/~sp8ct/tmp/float.taz
http://www-bl20.spring8.or.jp/~sp8ct/tmp/float.zip
```

```
----------------------------------------
Date:    Wed, 27 Jul 2016 16:18:24 +0900
Subject: float_TIFF
----------------------------------------
```

うえすぎさま、

なかのです。以前に紹介した浮動小数点数画素値を格納した TIFF 画像ファイルの読み書き用の関数 ReadImageFile_Float() と StoreImageFile_Float() の C 言語コード（rif_f* と sif_f*；sif_f* には２種類アリ）に関する話です。

(0)
これらに関する E-mails などをまとめた PDF ファイルをアップロードしておきました：http://www-bl20.spring8.or.jp/~sp8ct/tmp/float.pdf

(1)
2012/6/11 の E-mail では「コード sif_f_full.c が出力するフルスペックの TIFF 画像ファイルだけにそこに格納されている浮動小数点数画素値の最小値と最大値を示す TIFF の tags（SMinSampleValue と SMaxSampleValue）を記録」と書きました。しかしながら、ImageJ のような浮動小数点数画素値の TITF を処理可能なプログラムではこれらの tags が必須のようなので、簡易版の TIFF 画像ファイルにもそれらを書き込むようコード sif_f_fast.c を書き換えました。

(2)
関数 StoreImageFile_Float() は画像をスキャンして得た浮動小数点数画素値の最小値と最大値を型 FOM の外部変数 SIF_F_min と SIF_F_max に格納します。ファイル sif_f.h をインクルードすれば、これらの値を引用可能です。

(3)
StoreImageFile_Float() の最後の引数である文字列へのポインタ *desc としてファイル sif_f.h で宣言されている外部変数 SIF_F_desc を指定すると、この関数は上記の SIF_F_min と SIF_F_max の値を変換した文字列を SIF_F_desc が指す場所に格納した後にコメント（TIFF の Image Description tag）として TIFF 画像ファイルに書き込みます。

(4)
上記のように SIF_F_desc を指定した場合、関数 StoreImageFile_Float() は sif_f.h で宣言されている外部変数 SIF_F_form に格納されているフォーマットの文字列に従って SIF_F_min と SIF_F_max の値の文字列への変換を行います。ただし、後述するようにデフォルトのフォーマットの文字列が初期値として格納されているので、SIF_F_form に自分で文字列を設定しなくても問題ありません。

(5)
StoreImageFile_Float() の呼び出しの前に SIF_F_min と SIF_F_max の値それぞれに対する "%lf" のような倍精度浮動小数点数用の変換フォーマット文字を含む文字列を SIF_F_form に設定しておけば、それに応じて変換した文字列を TIFF 画像ファイルにコメントとして書き込むことができます。

(6)
コード sif_f*.c では SIF_F_form に格納可能な文字数をマクロ定数 SIF_F_LEN で、また、それに格納するフォーマットの文字列の初期値を SIF_F_FORM で設定します。これらの定数の値はコンパイルの時に指定可能で、それを行わなかった場合のデフォルトの値はファイル sif_f.h に宣言されています。

```
#define  SIF_F_LEN  2048
#define  SIF_F_FORM  "%lf\t%lf"
```

(7)
上記の (2) 〜 (6) の改変に伴う既存の関数 StoreImageFile_Float() を用いたプログラムの再コンパイルは不要です。ただし、これらにより以前に紹介した SIXM 用のプログラムのコードを簡略化できるので、今後それを行うつもりです。

とりあえず以上です。

2017/3/29

整数画素値の TIFF 画像（integer-TIFF）と浮動小数点数画素値のもの（float-TIFF）の変換用の２個のプログラム

t_i2f（t_i2f_fast を改名）：integer-TIFF を float-TIFF に変換
t_f2i：float-TIFF を integer-TIFF に変換

と同様に、バイナリもしくは ASCII-TEXT 形式で画素値のデータが書き込まれている画像と float-TIFF の変換処理を行う以下の４個のプログラムを作成した。

bin2t_f image.bin TIFF_format
ファイル image.bin に格納されているバイナリ画素値の３次元画像をスライスごとに float-TIFF に変換し、それらを TIFF_format で指定された０から始まるスライス番号を含むパス名のファイルに書き込む。

t_f2bin image.tif image.bin
t_f2bin image/ nameFile image.bin
image.tif として指定された float-TIFF のスライス画像、もしくは、ディレクトリ image/ の下の nameFile で指定された複数個の float-TIFF からなる３次元画像をバイナリ画素値の画像に変換し、ファイル image.bin に書き込む。ただし、image.bin として "-" を指定するとそのデータを標準出力に書き出す。

txt2t_f image.txt TIFF_format
image.txt に格納されている ASCII-TEXT 形式画素値の３次元画像をスライスごとに float-TIFF に変換し、それらを TIFF_format で指定された０から始まるスライス番号を含むパス名のファイルに書き込む。

t_f2txt image.tif PV_format image.txt
t_f2txt image/ nameFile PV_format image.txt
image.tif として指定された float-TIFF のスライス画像、もしくは、ディレクトリ image/ の下の nameFile で指定された複数個の float-TIFF からなる３次元画像上の浮動小数点数の画素値を PV_format で指定した ASCII-TEXT 形式の文字列に変換し、ファイル image.txt に書き込む。ただし、image.txt として "-" を指定するとそのデータを標準出力に書き出す。

注
TIFF_format と PV_format は C 言語の標準ライブラリ関数 sprintf() もしくは fprintf() にそのまま渡す整数値（TIFF_format）と浮動小数点数（PV_format）の文字列への変換形式を指定するためのフォーマットの文字列である。それぞれに以下のような文字列を指定すれば良い。
TIFF_format
１個のスライス画像だけを変換する場合
TIFF のファイル名そのものを指定すれば良い。
複数個のスライス画像の場合
それぞれのスライス画像をあらかじめ作成しておいたディレクトリ（image/）の下のスライス番号（整数値０〜）を名前の一部とするファイルに格納するなら
image/%04d.tif
のような文字列を指定すれば良い。
PV_format
大抵の場合は "%f" もしくは "%lf" を指定すれば良い。

ただし、これらのプログラムが処理するバイナリおよび ASCII-TEXT 形式の画素値の画像ファイル（image.bin と image.txt）の仕様は以下の通りである。

１行目（image.bin と image.txt で共通）
以下の３個の整数値の ASCII-TEXT の文字列が空白もしくはタブコード区切りで並んでいる。
[1] スライス画像の横画素数（Nx）
[2] スライス画像の縦画素数（Ny）
[3] スライス画像の枚数（Nz）
ただし、Nz の記載は省略可能で、その場合は Nz = 1 と見なす。なお、プログラム bin2t_f と txt2t_f はどちらも、これらの値に続く１行目の残りの文字を読み飛ばす。
2行目（最初の改行コードの次）以降のデータ
image.bin
little-endian の並びの IEEE754 形式で表現した倍精度浮動小数点数の画素値のバイナリデータがスキャンライン順に並んでいる。
image.txt
浮動小数点数の画素値を表すフリーフォーマットの文字列が空白、タブコードもしくは改行区切りでスキャンライン順に並んでいる。ただし、連続した複数の区切り文字は１個の区切り文字と見なす。なお、プログラム t_f2txt は以下の区切り文字を出力する。
横一列の画素値の最後のものの後：改行コード "\n"
それ以外の画素の後の区切り文字：タブコード "\t"
２番目以降のスライス画像の直前：改行コード "\n"