

# Web タイルセットを使用した画像の自動登録

TNTmipsのジオリファレンス処理には自動登録(Auto-Register)の機能 があり、1つの画像に対して同じ領域の参照画像を使って数百から数千の コントロールポイントを自動的に生成できます。(テクニカルガイド「ジ オリファレンス:参照画像への自動登録(Georeference: Auto-Register to Reference Image)」を参照)。これと同様の機能がTNT地理空間スクリ プト(SML)を使ったスクリプトを利用しても実現できます。マイクロ イメージ社はこの機能を実演する2種類のサンプルスクリプトを提 供しています。1つは参照画像としてマイクロイメージ社のサイト にあるWeb タイルセットを使うもの(次ページに抜粋を掲載)、も う1つは参照画像としてローカルのTNTプロジェクトファイル中 のカラー合成ラスタを使うものです。

自動登録の設定と操作は IMAGE\_PIPELINE\_AUTOREGISTER クラスにカプセル化されています。このクラスのメンバを用い てスクリプト作成者は「初期精度」や「最大残差」などの自 動登録に必要なパラメタの全てを設定できます。サンプルスク リプト中のクラスの使用方法を示す概略図を下記に示していま す。マルチバンドの入力画像から参照画像と照合するバンド1 つが選択され(図の右側)、画像用パイプラインソースを作る ために使われます。参照用画像に対してもパイプラインソース が作られます。Web タイルセットを参照画像として使用する場 合、インターネット上のWeb タイルセットのURL 文字列(ロー カルのWeb タイルセットの場合はファイルパス)を使用して

タイルセット用パイプラインソースが作られます。照合に使用するカラー成分を選別す るのに選択フィルタが使われます。ローカルのコンポジット画像を参照用に使う場合は、 照合に使用するカラーはパイプラインソース用のコンストラクタ内で直接指定できます ので、選択フィルタは必要ありません。その結果生じた入力画像と参照画像用のパイプ ラインステージはパラメタとして AUTOREGISTER クラスの Run()メソッドへ送られま す。それは処理を実行し、メモリ内にコントロールポイントジオリファレンスを生成し ます。その後 Save()メソッドを使用してこのジオリファレンスサブオブジェクトを入 力画像の各バンドに書き出します。



SML での自動登録処理はイメージパイプライン構造体を使用します が、完全なパイプライン操作ではありません。新しい"ターゲット" 画像は作られず、リファレンス情報だけが作られます。



EPP 5 4 ス 20m 5 SPOT4 画像(カラー赤外線 バンドを使用)。サンプル SML の自動 登録処理でコントロールポイントを生成。 参照画像はマイクロイメージ社サイトにある 2010 年サウスダコタ州 GM\_BM Web タイルセットです (全米農業イメージプログラム (NAIP)の ナチュラルカラー正射画像の郡データ)。

### 自動登録の設定:

入力画像と参照画像中の緑のスペクトル成分 を照合 初期精度評価=20 セル 生成するポイントの間隔=75 セル 最大ポイント残差=1.5 セル 相関領域サイズ=128 セル 最良適合モデル=2 次多項式

#### 自動登録処理で 537 個のコントロールポイン トを生成

#### 二乗平均平方根 (RMS) 残差:

X = 0.36 セル、Y = 0.43 セル、XY = 0.56 セル (モデル = 平面投影)

された 4 バンドのスポッ **平** ト衛星画像の自動登録処 X 理用に設計されています。

これらのサンプルスク

リプトは TNT プロジェク

トファイルにインポート

**平均絶対残差:** X = 0.28 セル、Y = 0.32 セル

しかし、これらのスクリプトは GeoTIFF や GeoJP2 のようなファイ ル形式の単一バンドやマルチバンド、カラーコンポジット画像を処 理できるよう簡単に修正できます。また多数の入力画像を使ってバッ チ処理を行うこともできます。

また、これらのスクリプトの自動登録設定は、面積やセルサイズ、 形状が異なる入力画像に対しても適応することができます。ローカ ルの画像を参照する例では参照画像から読み込んだ座標参照系 (CRS) が自動登録処理と出力のジオリファレンスに用いられます。しかし、 入力画像の CRS(指定がない場合のデフォルト)や他の CRS を使う ことも可能です。自動登録処理をより一般的な画像処理スクリプト に統合し、出力画像の CRS ヘリサンプリングしたり、画像強調やそ の他の大量処理も行うことができます。

(次ページに続く)

www.microimages.com/downloads/scripts.htm にはダウンロード可能な多くのサンプルスクリプトがあり、スクリプトやクエリーで TNT 製品のスク リプト言語の特徴をどのように利用したらよいか解説しています。

## AutoRegisterSPOT4bandWebTilesetRef.sml (自動登録の参照画像としてインターネット上の Web タイルセットを使用しています)

$ \begin{bmatrix} p + 1 \\ p \neq p$	numeric err;	自動登録クラスのパラメータを設定。座標参照系は入力画像から自動的に設 定されます
edwn():       コンソールを2017         annowed comparison of the second s	proc ReportError(numeric linenum, numeric err) { printf("FAILED -line: %d, error: %d\n", linenum - 1, err); PopupError(err);	print("Setting up auto-register"); statusC.Message = "Setting up auto-register"; class IMAGE_PIPELINE_AUTOREGISTER autoReg;
EMEST 5 EMEN (C) FUTURE (EMES)     EAST AND (C) CONTRAINANCY,     EAST AND (C) CONTRAIN	} clear(); コンソールをクリア	autoReg.SetAutoGenerateGCPs(1); コントロールポイントの自動生成の設定
class STRING promptS: class RVC_OBJETEM inpOb]temList[]: momeric numBads;       momeric numBads;       momeric numBads;       momeric numBads;         DigGetObjects(promptS, "Raster", inpOb]temList[]: momeric numBads;       momeric numBads;       momeric numBads;       momeric numBads;         DigGetObjects(promptS, "Raster", inpOb]temList, "ExistingOnly"; numeric i; for i = 1 to inpOb]temList GetNumItemS()       DigItem //y>2x872/201;       momeric numBads;       momeric numBads;         full_file_Bad Sid = %selm", i, inpOb]temList[] GetObjectPath() ; }       momeric numBads;       momeric numBads;       momeric numBads;         full_file_Bad Sid = %selm", i, inpOb]temList[] GetObjectPath() ; }       momeric numBads;       momeric numBads;       momeric numBads;         full_self       momeric numBads;       momeric numBads;       momeric numBads;       momeric numBads;         full_self       momeric numBads;       momeric numBads;       momeric numBads;       momeric numBads;         full_self       momeric numBads;       momeric numBads;       momeric numBads;       momeric numBads;         full_self       momeric numBads;       momeric numBads;       momeric numBads;       momeric numBads;         full_self       momeric numBads;       momeric numBads;       momeric numBads;       momeric numBads;         full_self       momeric numBads;       momeric numBads;       momeric numBads; </td <td>自動登録する画像バンドを選択</td> <td>autoReg.SetModel("PlaneProjective"); 初期モデルの設定</td>	自動登録する画像バンドを選択	autoReg.SetModel("PlaneProjective"); 初期モデルの設定
numeric nümBands;	class STRING prompt\$; prompt\$ = "Choose 4 SPOT MS bands:"; class RVC OBJITEM inpObjItemList[];	autoReg.SetUseExistingGCPs(0, 0); 既存のコントロールポイントは使用しない(処理終了時に削除)
DigGetObjects(promptS, "Raster", inpObjItemList, "ExistingOaly"); numeric i; for i = 10 impObjItemList.GetNumItem() (bijtem />>2 #72+038# = 100 ±// autoReg.SetGCPSpacing(100); #72+038# = 100 ±// autoReg.SetGCPSpacing(100); #72+038# = 100 ±// autoReg.SetGCPSpacing(100); #72+2 ± 038# = 100 ±// autoReg.SetGCPSpacing(100); #72+2 ± 04 autoReg.SetGCPSpacing(	numeric numBands; 用の RVC_OBJITEM のハッシュ	autoReg.SetInitialAccuracy(20); 初期精度=20セル
	DlgGetObjects(prompt\$, "Raster", inpObjItemList, "ExistingOnly");	autoReg.SetGCPSpacing(100); ポイントの間隔 = 100 セル
$ \begin{array}{llllllllllllllllllllllllllllllllllll$	numeric i; for i = 1 to inpOhiltemList GetNumItems() Ohiltem $1/2\sqrt{275}$	autoReg.SetMaxGCPResidual(2); 最大残差 = 2 セル
print(* band xd = xbsir, *, impledjeterinski]; GKODJECPTan(/),       autoReg SetMaxModel(*PolynomialOrder2*);       Qetade t= ybeQetade         gemasic@5 Web 97/http://www.microimages.com/geodata/kappa/SD2010 GM state/       print(*PolynomialOrder2*);       Qetade t= ybeQetade         jotage 1       autoReg SetMaxModel(*PolynomialOrder2*);       Qetade t= ybeQetade         jotage 2       jotage 2       jotage 2         jotage 2       jotage 2       jotage 2       jotage 2         jotage 2       jotage 2       jotage 2       jotage 2         jotage 2       jotage 2       jotage 2       jotage 2         jotage 2       jotage 2       jotage 2       <	For the important state of turn tension $(1 - 1)^{-1}$ is the turn tension $(1 - 1)^{$	autoReg.SetCorrPatchSize(128); 相関領域サイズ = 128 セル
■要認識に使う Web タイルセットの URL を得る         Class STRING urß, defaulfS;         defaulf = "http://www.microimages.com/geodata/kappa/SD2010 GM state/         SD2010 GM state/         SD2010 GM state tsd";         prompt = "Enter or past URL 0 web tileset reference.";         url = PopupString(prompts, defaults, "Web Tileset URL");         Lac > 2.x5 x7T0202 ± 0.000 × 5.4 x6 ± 0.0000	$\begin{cases} \text{prime} \text{ band } \text{od} = \text{ossu},  is inposite intersection of the intersectio$	autoReg.SetMaxModel("PolynomialOrder2"); 最良適合モデルを設定
ekas STRING urfs, default5: default5 = "http://www.microimages.com/geodata/kappa/SD2010 GM state/ group5 = "inter or past URL of web tileset reference:"; urfs = PoopuSiningforomps, default5, "Web Tileset URL";	参照画像に使う Web タイルセットの URL を得る	自動登録を実行
$ \begin{split} \  \mathbf{rt} \mathbf{r}^{T} \mathbf{r}^{T}$	class STRING url\$, default\$; default\$ = "http://www.microimages.com/geodata/kappa/SD2010 GM state/ SD2010 GM state.tsd"; promnt\$ = "Enter or paste URL of web tileset reference:":	<pre>print("Parameters set, starting auto-register"); statusC.Message = "Running auto-register"; autoReg.Run(inpSource, refSource);</pre>
$\overline{\lambda\tau} - 9.3 \% 4 7 \pi 0^2 \xi 4 0 \exists \gamma \tau h k f k f k f k f k f k f k f k f k f k$	url\$ = PopupString(prompt\$, default\$, "Web Tileset URL");	自動登録クラスより 2 乗平均平方根 (RMS)XY 残差と結果モデルを得る
自動登録に使用する入力画像の単一パンド用のパイプラインソースを作成。 ObjtemList の一次元インデックスによって3=Band、3=greenを指定class IMAGE_PIPELINE_SOURCE_RVC inpSource(inpObjtemLis[3]); err = inpSource initialize();if (rer <0) then ReportError(_context.CurrentLineNum, err); else print("reference source initialized");if (rersultRMS <= 1.5) { print("residual less than cut-off, saving result"); else print("input source initialized");if (resultRMS <= 1.5) { { print("residual less than cut-off, saving result"); else print("input source initialized");if (resultRMS <= 1.5) { { print("residual less than cut-off, saving result"); else print("reference source initialized");class IMAGE_PIPELINE_SOURCE_TILESET refTileset(url\$); err = refTileset.Initialized");class IMAGE_PIPELINE_SOURCE_TILESET refTileset(url\$); err = refTileset.Initialized");if (resultRMS <= 1.5) { { { minit("residual less than cut-off, saving result"); else print("reference source initialized");if (resultRMS <= 1.5) { { { { { { { { { { { { { { { { { { { { { { { { { { { { { { { { { { { { { { { { { { { { { { { { 	ステータスダイアログとそのコンテキストを作成         class STATUSDIALOG statusD; statusD.Create(); class STATUSCONTEXT statusC; statusC = statusD.CreateContext();       プロセス状態を表示するため のステータスダイアログ         ステータスダイアログ用コンテキ	numeric resultRMS;         class STRING resultModel;         resultRMS = autoReg.GetResultRMSResidual();         resultModel = autoReg.GetResultModel();         printf("Result residual = %.2f\n", resultRMS);         printf("Result model = %s\n", resultModel);
elass IMAGE_PIPELINE_SOURCE_RVC inpSource(inpObjItemList[3]); err = inpSource.Initialize(); if (err < 0) then ReportError(_context.CurrentLineNum, err); else print("input source initialized");	自動登録に使用する入力画像の単一バンド用のパイプラインソースを作成。 ObjltemList の一次元インデックスによって3=Band、3=greenを指定	コントロールポイント全体の2乗平均平方根をチェックし、 処理が終われば全ての入力のジオリファレンスを保存する
参照 Web タイルセット画像 (ナチュラルカ ラー合成)用のバイブラインソースを作成for i = 1 to inpObjItemList.GetNumItems() { autoReg.SaveGeoreference(inpObjItemList[i]); printf("saved georeference (inpObjItemList[i]); printf("saved georeference for input %d\n", i); } }class IMAGE_PIPELINE_SOURCE_TILESET refTileset(url\$); err = refTileset.Initialize(); if (err < 0) then ReportError(_context.CurrentLineNum, err); else print("reference source initialized");for i = 1 to inpObjItemList.GetNumItems() { autoReg.SaveGeoreference (inpObjItemList[i]); printf("saved georeference for input %d\n", i); } }参照タイルセットの縁のパンドを選別するフィルタ array numeric srcSamples[1]; srcSamples[1] = 2;縁のパンドへのインデックスclass IMAGE_PIPELINE_FILTER_SELECT refTilesetGreen(refTileset, srcSamples, 1); err = refTilesetGreen.Initialize(); if (err < 0) then ReportError(_context.CurrentLineNum, err); else print("select filter for reference tileset initialized");statusC.Message = "Script Ran to Completion"; statusD.Destroy(); print("Done");class IMAGE_PIPELINE_FILTER_SELECT refTilesetGreen(refTileset, srcSamples, 1); err = refTilesetGreen.Initialize(); if (err < 0) then ReportError(_context.CurrentLineNum, err); else print("select filter for reference tileset initialized");statusC.Message = "Script Ran to Completion"; 	<pre>class IMAGE_PIPELINE_SOURCE_RVC inpSource(inpObjItemList[3]); err = inpSource.Initialize(); if (err &lt; 0) then ReportError(_context.CurrentLineNum, err); else print("input source initialized");</pre>	<pre>if (resultRMS &lt;= 1.5) {     print("residual less than cut-off, saving result");</pre>
参照Web タイルセット画像 (アナムラルガ ラー合成)用のバイブラインソースを作成class IMAGE_PIPELINE_SOURCE_TILESET refTileset(url\$); err = refTileset.Initialize(); if (err < 0) then ReportError[_context.CurrentLineNum, err); else print("select filter for reference tileset initialized");参照タイルセットの緑のパンドを選別するフィルタ array numeric srcSamples[1]; srcSamples[1] = 2;緑のパンドへのインデックスclass IMAGE_PIPELINE_FILTER_SELECT refTilesetGreen(refTileset, srcSamples, 1); else print("select filter for reference tileset initialized");AutoRegisterSPOT4band.sml ではプロジェクトファイル内のローカルのカラー 合成画像を参照画像として使用しています。		for i = 1 to inpObjItemList.GetNumItems()
class IMAGE_PIPELINE_SOURCE_TILESET refTileset(url\$); err = refTileset.Initialize(); if (err < 0) then ReportError(_context.CurrentLineNum, err); else print("reference source initialized");	参照WEDダイルセット画像(ナデュラルガ ラー合成)用のパイプラインソースを作成	autoReg.SaveGeoreference(inpObjItemList[i]); printf("saved georeference for input %d'\n", i);
if (err < 0) then ReportError(_context.CurrentLineNum, err); else print("reference source initialized");	<pre>class IMAGE_PIPELINE_SOURCE_TILESET refTileset(url\$); err = refTileset.Initialize();</pre>	}
参照タイルセットの緑のパンドを選別するフィルタ       statusC.Message = "Script Ran to Completion";         array numeric srcSamples[1];       緑のパンドへのインデックス         srcSamples[1] = 2;       緑のパンドへのインデックス         class IMAGE_PIPELINE_FILTER_SELECT refTilesetGreen(refTileset, srcSamples, 1);       print("Done");         err = refTilesetGreen.Initialize();       if (err < 0) then ReportError(_context.CurrentLineNum, err);	<pre>if (err &lt; 0) then ReportError(_context.CurrentLineNum, err); else print("reference source initialized");</pre>	最後のステータスダイアログメッセージを設定し、ダイアログを閉じる
array numeric srcSamples[1]; srcSamples[1] = 2; 緑のバンドへのインデックス print("Done"); class IMAGE_PIPELINE_FILTER_SELECT refTilesetGreen(refTileset, srcSamples, 1); err = refTilesetGreen.Initialize(); if (err < 0) then ReportError(_context.CurrentLineNum, err); else print("select filter for reference tileset initialized"); AutoRegisterSPOT4band.smlではプロジェクトファイル内のローカルのカラー 合成画像を参照画像として使用しています。	参照タイルセットの緑のバンドを選別するフィルタ	<pre>statusC.Message = "Script Ran to Completion"; statusD.Destrov():</pre>
class IMAGE_PIPELINE_FILTER_SELECT refTilesetGreen(refTileset, srcSamples, 1); err = refTilesetGreen.Initialize(); if (err < 0) then ReportError(_context.CurrentLineNum, err); else print("select filter for reference tileset initialized"); dxtoRegisterSPOT4band.smlではプロジェクトファイル内のローカルのカラー 合成画像を参照画像として使用しています。	array numeric srcSamples[1]; srcSamples[1] = 2; 緑のバンドへのインデックス	print("Done");
if (err < 0) then ReportError(_context.CurrentLineNum, err); else print("select filter for reference tileset initialized"); AutoRegisterSPOT4band.sml ではプロジェクトファイル内のローカルのカラー 合成画像を参照画像として使用しています。	<pre>class IMAGE_PIPELINE_FILTER_SELECT refTilesetGreen(refTileset,</pre>	
	if (err < 0) then ReportError(_context.CurrentLineNum, err); else print("select filter for reference tileset initialized");	AutoRegisterSPOT4band.sml ではプロジェクトファイル内のローカルのカラー 合成画像を参照画像として使用しています。