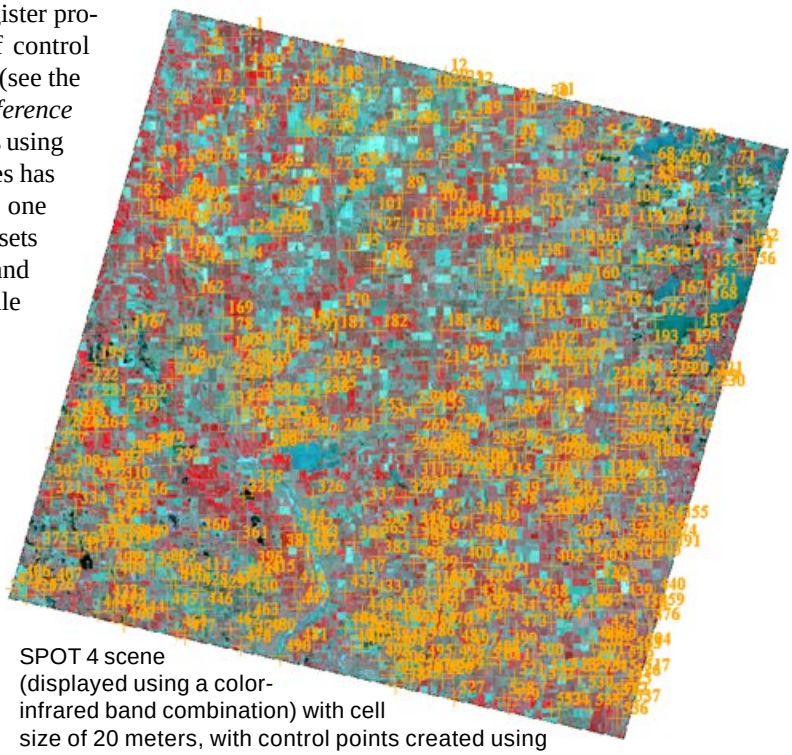
Auto-Register Images using Web Tileset

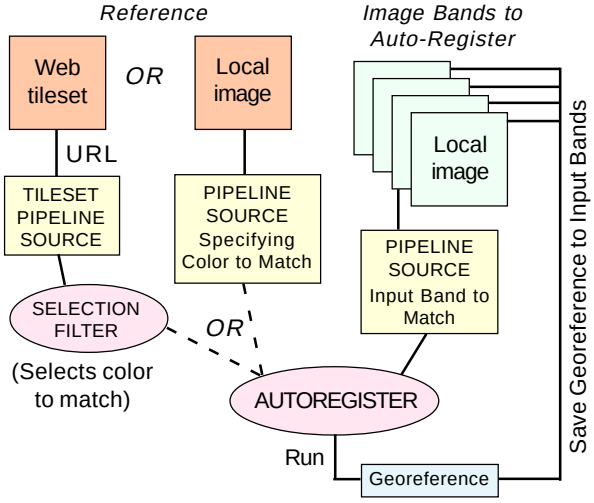
The Georeference process in TNTmips includes an Auto-Register procedure to automatically generate hundreds to thousands of control points for an image using a reference image of the same area (see the Technical Guide entitled *Georeference: Auto-Register to Reference Image*). This same capability is also available for use in scripts using the TNT Geospatial Scripting Language (SML). MicroImages has provided two sample scripts to demonstrate this capability: one (excerpted on the reverse) that uses one of the standard web tilesets hosted at the MicroImages website as the reference image, and one that uses a local color-composite raster in a TNT Project File as the reference.



SPOT 4 scene (displayed using a color-infrared band combination) with cell size of 20 meters, with control points created using the Auto-Register procedure in the sample SML script. The reference image is the South Dakota 2010 GM_BM web tileset hosted at microimages.com (derived from NAIP natural-color county orthoimages).

Auto-Register settings and operations are encapsulated in the IMAGE_PIPELINE_AUTOREGISTER class. This class has members that allow the script writer to set all necessary auto-register parameters, such as Initial Accuracy and Maximum Residual. A schematic diagram showing how the class is used in the sample scripts is shown below. One band is selected from the multiband input image (right side of diagram) to match to the reference and is used to create an image pipeline source. A pipeline source is likewise created for the reference. If using a web tileset as reference, a tileset pipeline source is created using the URL string for the web tileset on the internet (or the filepath for a local web tileset). A selection filter is used to select the color component to use for matching. If a local composite image is used as reference, the color to use for matching can be specified directly in the constructor for the pipeline source, so no selection filter is required. The resulting pipeline stages for the input image and reference are passed as parameters to the Run() method of the AUTOREGISTER class, which carries out the operation and creates the resulting control-point georeference in memory. A Save() method on the class is then used to write this georeference subobject to each of the bands of the input image.

Schematic of the Sample Auto-Register Scripts



Although the auto-register operation in SML utilizes image pipeline structures, it is not a complete pipeline operation, as no new "target" image is created, only georeference information.

Auto-Register Settings:

Match green spectral component in input and reference
Initial Accuracy Estimate = 20 cells
Generated Point Spacing = 75 cells
Maximum Point Residual = 1.5 cells
Correlation Patch Size = 128 cells
Maximum Adaptive Model = Order 2 Polynomial

Auto-Register produced 537 control points

RMS Residuals:

X = 0.36 cells, Y = 0.43 cells, XY = 0.56 cells
(Model = Plane Projective)

Mean Absolute Residuals:

X = 0.28 cells, Y = 0.32 cells

These sample scripts are designed for standalone auto-registration of a 4-band SPOT satellite image that has been imported to a TNT Project File. However, these scripts can be adapted easily to process any single-band or multiband image or color composite in a variety of file formats (such as GeoTIFF or GeoJP2) or to do batch processing of a number of input images.

The auto-register settings in these scripts likewise can be tailored to the dimensions, cell-size, and geometry of different input images. In the local reference example the coordinate reference system (CRS) read from the reference image is used for the auto-registration procedure and the output georeference, but you can use the CRS of the input image (the default if not specified) or any other valid CRS instead. The auto-register procedure can also be integrated into a more general image-processing script that carries out resampling to the output CRS, image enhancement, and other production procedures.

(over)

Many sample scripts have been prepared to illustrate how you might use the features of the TNT products' scripting language for scripts and queries. These scripts can be downloaded from www.microimages.com/downloads/scripts.htm.

AutoRegisterSPOT4bandWebTilesetRef.sml
(uses a web tileset on the internet as the reference image for auto-register)

```
numeric err;

    error checking procedure

proc ReportError(numeric linenum, numeric err) {
   printf("FAILED -line: %d, error: %d\n", linenum - 1, err);
   PopupError(err);
   }

clear();        clear the console

    choose image bands to auto-register

class STRING prompt$;
prompt$ = "Choose 4 SPOT MS bands:";
class RVC_OBJITEM inpObjItemList[];        hash of RVC_OBJITEMs for
numeric numBands;                          selected raster objects

DlgGetObjects(prompt$, "Raster", inpObjItemList, "ExistingOnly");

numeric i;
for i = 1 to inpObjItemList.GetNumItems()        check ObjItem hash
   {
   printf("Band %d = %s\n", i, inpObjItemList[i].GetObjectPath() );
   }

get URL for the web tileset to use for the reference

class STRING url$, default$;
default$ = "http://www.microimages.com/geodata/kappa/SD2010 GM state/
SD2010 GM state.tsd";
prompt$ = "Enter or paste URL of web tileset reference:";
url$ = PopupString(prompt$, default$, "Web Tileset URL");

   create status dialog and its context

class STATUSDIALOG statusD;    status dialog to show process status
statusD.Create();
class STATUSCONTEXT statusC;        context for the status dialog
statusC = statusD.CreateContext();

   create pipeline source for the single band of the input to use
   for Auto-Register; specify by 1-based index to the ObjItemList
   where 3 = Band 3 = green:

class IMAGE_PIPELINE_SOURCE_RVC inpSource(inpObjItemList[3]);
err = inpSource.Initialize();
if (err < 0) then ReportError(_context.CurrentLineNum, err);
else print("input source initialized");

    create pipeline source for the reference
    web tileset (natural color-composite)

class IMAGE_PIPELINE_SOURCE_TILESET refTileset(url$);
err = refTileset.Initialize();
if (err < 0) then ReportError(_context.CurrentLineNum, err);
else print("reference source initialized");

  filter to select the green component of the reference tileset

array numeric srcSamples[1];
srcSamples[1] = 2;        index for green component

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");

   set auto-register class parameters; coordinate reference
   system will be set automatically from the input image

print("Setting up auto-register...");
statusC.Message = "Setting up auto-register...";
class IMAGE_PIPELINE_AUTOREGISTER autoReg;

autoReg.SetAutoGenerateGCPs(1);    set to auto-generate control points

autoReg.SetModel("PlaneProjective");    set initial model

autoReg.SetUseExistingGCPs(0, 0);    don't use existing control points
                                     (will be deleted when done)
autoReg.SetInitialAccuracy(20);      Initial Accuracy = 20 cells

autoReg.SetGCPSpacing(100);          Point Spacing = 100 cells

autoReg.SetMaxGCPResidual(2);        Maximum Residual = 2 cells

autoReg.SetCorrPatchSize(128);       Correlation Patch Size =128 cells

autoReg.SetMaxModel("PolynomialOrder2");    set Maximum Model

    run auto-register

print("Parameters set, starting auto-register");
statusC.Message = "Running auto-register...";
autoReg.Run(inpSource, refSource);

get RMS XY residual and result model from the auto-register class

numeric resultRMS;
class STRING resultModel;
resultRMS = autoReg.GetResultRMSResidual();
resultModel = autoReg.GetResultModel();
printf("Result residual = %.2f\n", resultRMS);
printf("Result model = %s\n", resultModel);

   check the overall control point RMS and if it passes,
   save georeference to all of the inputs

if (resultRMS <= 1.5)
   {
   print("residual less than cut-off, saving result");

   for i = 1 to inpObjItemList.GetNumItems()
      {
      autoReg.SaveGeoreference(inpObjItemList[i]);
      printf("saved georeference for input %d\n", i);
      }
   }

  set final status dialog message and destroy the dialog

statusC.Message = "Script Ran to Completion";
statusD.Destroy();

print("Done");
```

A similar sample script using a local color-composite image in a Project File as the reference is entitled:

AutoRegisterSPOT4band.sml