

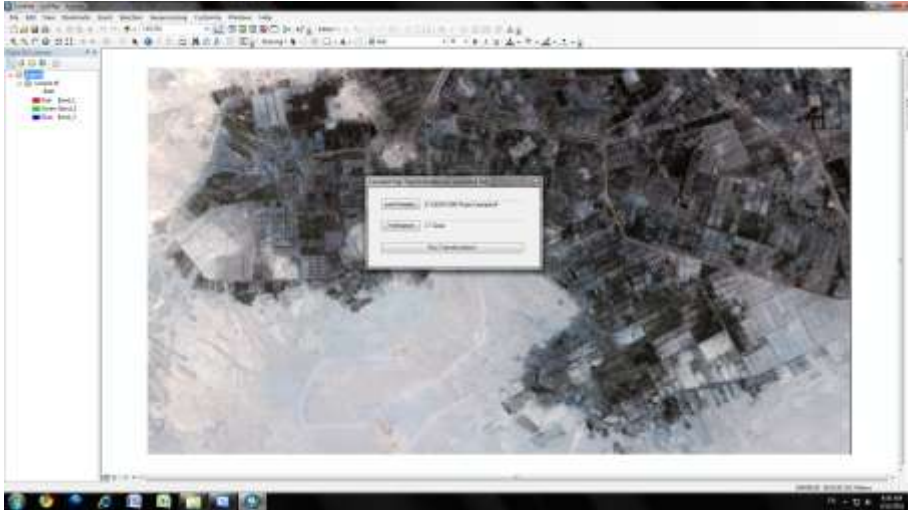
Tasseled Cap Transformation for 7-Band Landsat 4 Thematic Mapper Imagery

Christopher Brown

Tasseled Cap Transformation

- Why make this transformation?
 - Separates 7-band Landsat Thematic Mapper Imagery into Brightness, Wetness, Greenness, and residual rasters
- How will it be used?
 - Summarizes image variability for vegetation analysis
- Why is this add-in necessary?
 - Functionality does not currently exist in ArcMap
 - All 36 band coefficients are stored in code
 - Saves time in manual data processing

Dockable Window GUI



Get Imagery Location using GxDialog

```
'initialize and open mini browser
  with pGxDialog
    .AllowMultiselect = False
    .ButtonCaption = "Select"
    .Title = "Load 7-Band Thematic Mapper Imagery for Tasseled Cap
      Transformation"
    'using raster dataset filter
    .ObjectFilter = pFilter
    'open dialog passing handle to Application from AddIn
    bobjectSelected = .DoModalOpen
      (My.ArcMap.Application.hwnd, pGxObjects)
  End with

'if no file is selected, exit
  If bobjectSelected = Nothing Then Exit Sub
```

Check Imagery and Add to Map

```
'get raster dataset
  pGxDataset = pGxObjects.Next
  pGxRasterDataset = pGxDataset.Dataset
  'if imagery doesn't have 7 bands, exit
  pGxRasterBandCollection = pGxRasterDataset

'check that there are 7 bands
If pGxRasterBandCollection.Count <> 7 Then
  MsgBox.Show("Input imagery does not have 7 bands!",
    "Error", MessageBoxButtons.OK, MessageBoxIcon.None,
    MessageBoxDefaultButton.Button1)
  If DialogResult.OK Then Exit Sub
End If

'if no dataset has been selected, exit
If pGxRasterDataset Is Nothing Then Exit Sub

'add dataset to map
AddGeoDatasetRasterToMap(pGxRasterDataset, pMap)
```

Suburban Alexandria, Egypt



Use Coefficients as Function Arguments

```
'execute six tasseled cap transformations with coefficients
    My.ArcMap.Application.StatusBar.ProgressBar.Step()
pBrightness = TasseledCapTransformation(pGxRasterDataset,
    0.3037, 0.2793, 0.4743, 0.5585, 0.5082, 0.1863)
    My.ArcMap.Application.StatusBar.ProgressBar.Step()
pGreenness = TasseledCapTransformation(pGxRasterDataset, -
    0.2848, -0.2435, -0.5436, 0.7243, 0.084, -0.18)
    My.ArcMap.Application.StatusBar.ProgressBar.Step()
pwetness = TasseledCapTransformation(pGxRasterDataset,
    0.1509, 0.1973, 0.3279, 0.3406, -0.7112, -0.4572)
    My.ArcMap.Application.StatusBar.ProgressBar.Step()
pHaze = TasseledCapTransformation(pGxRasterDataset, -
    0.8242, 0.0849, 0.4392, -0.058, 0.2012, -0.2768)
    My.ArcMap.Application.StatusBar.ProgressBar.Step()
pFifth = TasseledCapTransformation(pGxRasterDataset, -
    0.328, 0.0549, 0.1075, 0.1855, -0.4357, 0.8085)
    My.ArcMap.Application.StatusBar.ProgressBar.Step()
pSixth = TasseledCapTransformation(pGxRasterDataset,
    0.1084, -0.9022, 0.412, 0.0573, -0.0251, 0.0238)
    My.ArcMap.Application.StatusBar.ProgressBar.Step()
```

Progress Bar

```
'set up the progress bar
    Dim pStatus As IStatusBar =
        My.ArcMap.Application.StatusBar
    Dim pStepProg As IStepProgressor =
        pStatus.ProgressBar
    With pStepProg
        .Position = 0
        .MaxRange = 100
        .Message = "Update progress:"
        .StepValue = 2
        .Show()
    End With

'progress bar increment
    My.ArcMap.Application.StatusBar.ProgressBar.Step()
```

Set Map Algebra Environments

```
'set cell size, mask, spatial reference from input
    pRasterAnalysisEnvironment.SetCellSize
        (esriRasterEnvSettingEnum.esriRasterEnvMinOf)

    pRasterAnalysisEnvironment.Mask = CType(pGxRasterDataset, IGeoDataset2)

    pRasterAnalysisEnvironment.OutSpatialReference =
        CType(pGxRasterDataset, IGeoDataset2).SpatialReference

'set map algebra extent

    pEnvelope.XMin = CType(pGxRasterDataset, IGeoDataset2).Extent.XMin
        pEnvelope.YMin = CType(pGxRasterDataset,
    IGeoDataset2).Extent.YMin
        pEnvelope.XMax = CType(pGxRasterDataset,
    IGeoDataset2).Extent.XMax
        pEnvelope.YMax = CType(pGxRasterDataset,
    IGeoDataset2).Extent.YMax
        pExtentProvider = CType(pEnvelope, System.Object)

    pRasterAnalysisEnvironment.SetExtent(esriRasterEnvSettingEnum.esriRaster
    EnvValue, pExtentProvider)

    pRasterAnalysisEnvironment.SetAsNewDefaultEnvironment()
```

Get Raster Bands and Bind for Map Algebra

```
'get raster bands from raster dataset
    pRasterBands = CType(pGxRasterDataset,
        IRasterBandCollection)
    pBand1 = pRasterBands.Item(0)
    pBand2 = pRasterBands.Item(1)
    pBand3 = pRasterBands.Item(2)
    pBand4 = pRasterBands.Item(3)
    pBand5 = pRasterBands.Item(4)
    pBand7 = pRasterBands.Item(6)

'bind raster bands
    pMapAlgebraOp.BindRaster(pBand1, "Band1")
    pMapAlgebraOp.BindRaster(pBand2, "Band2")
    pMapAlgebraOp.BindRaster(pBand3, "Band3")
    pMapAlgebraOp.BindRaster(pBand4, "Band4")
    pMapAlgebraOp.BindRaster(pBand5, "Band5")
    pMapAlgebraOp.BindRaster(pBand7, "Band7")
```

Multiply Each Band by Coefficient

```
'execute map algebra expressions to multiply each band by correct coefficient

Dim str As String = "[Band1] * " & coeff1
pBand1Out = pMapAlgebraOp.Execute(str)
My.ArcMap.Application.StatusBar.ProgressBar.Step()

str = "[Band2] * " & coeff2
pBand2Out = pMapAlgebraOp.Execute(str)
My.ArcMap.Application.StatusBar.ProgressBar.Step()

str = "[Band3] * " & coeff3
pBand3Out = pMapAlgebraOp.Execute(str)
My.ArcMap.Application.StatusBar.ProgressBar.Step()

str = "[Band4] * " & coeff4
pBand4Out = pMapAlgebraOp.Execute(str)
My.ArcMap.Application.StatusBar.ProgressBar.Step()

str = "[Band5] * " & coeff5
pBand5Out = pMapAlgebraOp.Execute(str)
My.ArcMap.Application.StatusBar.ProgressBar.Step()

str = "[Band7] * " & coeff7
pBand7Out = pMapAlgebraOp.Execute(str)
My.ArcMap.Application.StatusBar.ProgressBar.Step()
```

Sum All Output Bands and Return

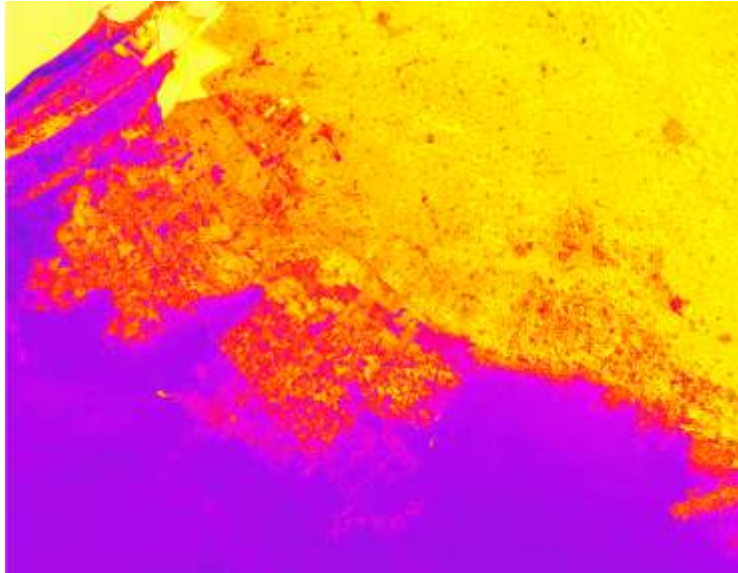
```
'rebind raster bands with new values
pMapAlgebraOp.BindRaster(pBand1Out, "Band1Out")
pMapAlgebraOp.BindRaster(pBand2Out, "Band2Out")
pMapAlgebraOp.BindRaster(pBand3Out, "Band3Out")
pMapAlgebraOp.BindRaster(pBand4Out, "Band4Out")
pMapAlgebraOp.BindRaster(pBand5Out, "Band5Out")
pMapAlgebraOp.BindRaster(pBand7Out, "Band7Out")

'add each band together to create output geodataset
str = "[Band1Out] + [Band2Out] + [Band3Out] +
[Band4Out] + [Band5Out] + [Band7Out]"
outputGeoDataset = pMapAlgebraOp.Execute(str)

'convert output geodataset to raster
'can cause bug if random location in C:/temp already exists
pRasterConversion = New RasterConvertHelper
outputRaster = pRasterConversion.ToRaster1
(outputGeoDataset, "GRID", pRasterAnalysisEnvironment)

Return outputRaster
My.ArcMap.Application.StatusBar.ProgressBar.Step()
```

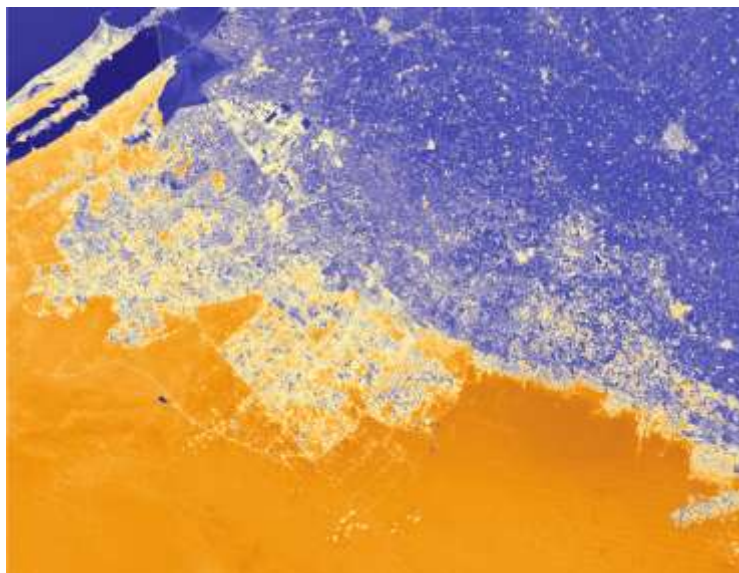
Output Raster 1: Brightness



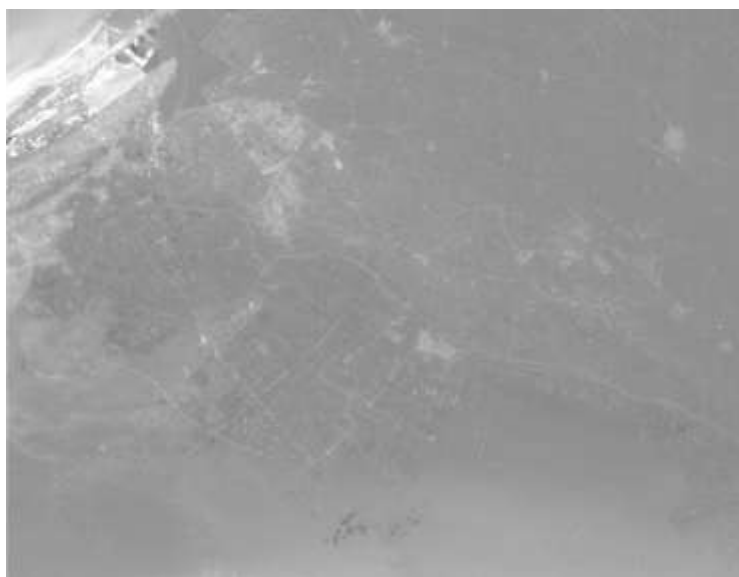
Output Raster 2: Greenness



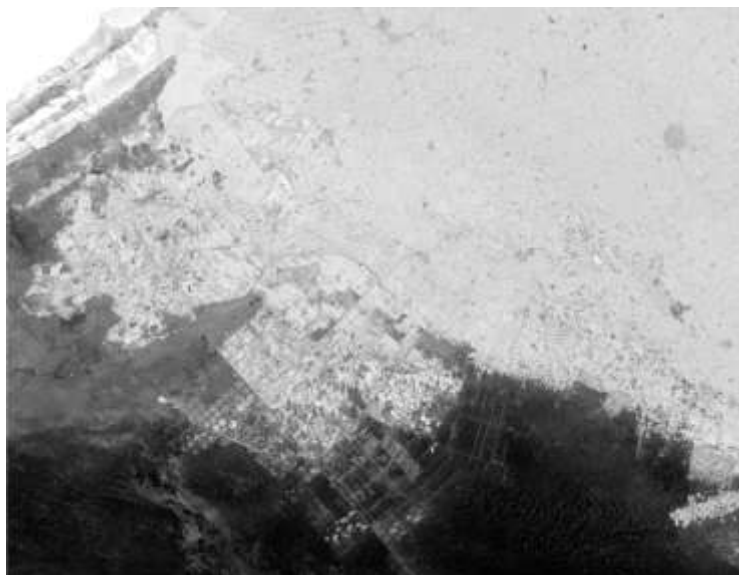
Output Raster 3: Wetness



Output Raster 4: Haze



Output Raster 5: Residual Variation



Output Raster 6: Residual Variation



Add Final Tasseled Cap Rasters to Map

```
'create tasseled cap layers from rasters and add them to map
  SaveRasterAddToMap("Brightness", pBrightness, pMap)
  SaveRasterAddToMap("Greenness", pGreenness, pMap)
  SaveRasterAddToMap("Wetness", pWetness, pMap)
  SaveRasterAddToMap("Haze", pHaze, pMap)
  SaveRasterAddToMap("Fifth", pFifth, pMap)
  SaveRasterAddToMap("Sixth", pSixth, pMap)
```

Get Save Location using GxDialog

```
'use GxDialog to choose where to save output/loop to see if file exists
Do
  With pGxDialog
    .AllowMultiSelect = False
    .ButtonCaption = "Save"
    .Title = "Save " & name & " Raster"
    .Name = name & ".tif"
    'using raster dataset filter
    .ObjectFilter = pFilter
    'open dialog passing handle to Application from AddIn
    bObjectSelected = .DoModalSave(My.ArcMap.Application.hwnd)
  End With
Loop Until File.Exists(pGxDialog.FinalLocation.FullName) = False
```

Sum All Output Bands and Return

```
'save raster and add to map
  pRasterWorkspace = pWorkspaceFactory.OpenFromFile
    (pGxDialog.FinalLocation.FullName, 0)
  pRasterBandCollection = pRaster
  pRasterBandCollection.SaveAs
    (pGxDialog.Name, pRasterWorkspace, "TIFF")

'create layer and add to map
  pRLayer = New RasterLayer
  pRLayer.CreateFromRaster(pRaster)
  pRLayer.Name = name
  My.ArcMap.Document.AddLayer(pRLayer)
```