Matching STC-2.0 transform model with various 2D polynomial image distortions

F.Bonnarel (CDS)

acknowledges Mark Cresitello-Dittmar for enlightments on the STC model and

STC Transform scope

- Data Cubes gather measurement along different axes
  - Sometimes all independant (event lists),
  - Sometimes some are dependant (ND images)
  - « Coordinates » on independant axes.
- Generally pixels are « device » coordinates
- Calibration process allows to map onto World Coordinates

  → Process results in a coordinate transform
- « STC transform » allows to represent these coordinate transforms

- Hey !!! Isn't that done via WCS keywords already ?

FITS WCS bilinear case

- Pixel coordinates (x,y) to World coordinates (lon, lat)
- Keywords
    - RADESYS, spatial frame
    - CTYPE1,CTYPE2, projection
    - CRPIX1,CRPIX2, pixel coordinates of origin
    - CRVAL1,CRVAL2, world coordinates of origin
    - CD1_1, CD1_2, CD2_1, CD2_2, for bilinear transformation

FITS WCS bilinear case

Dx = x-crpix1

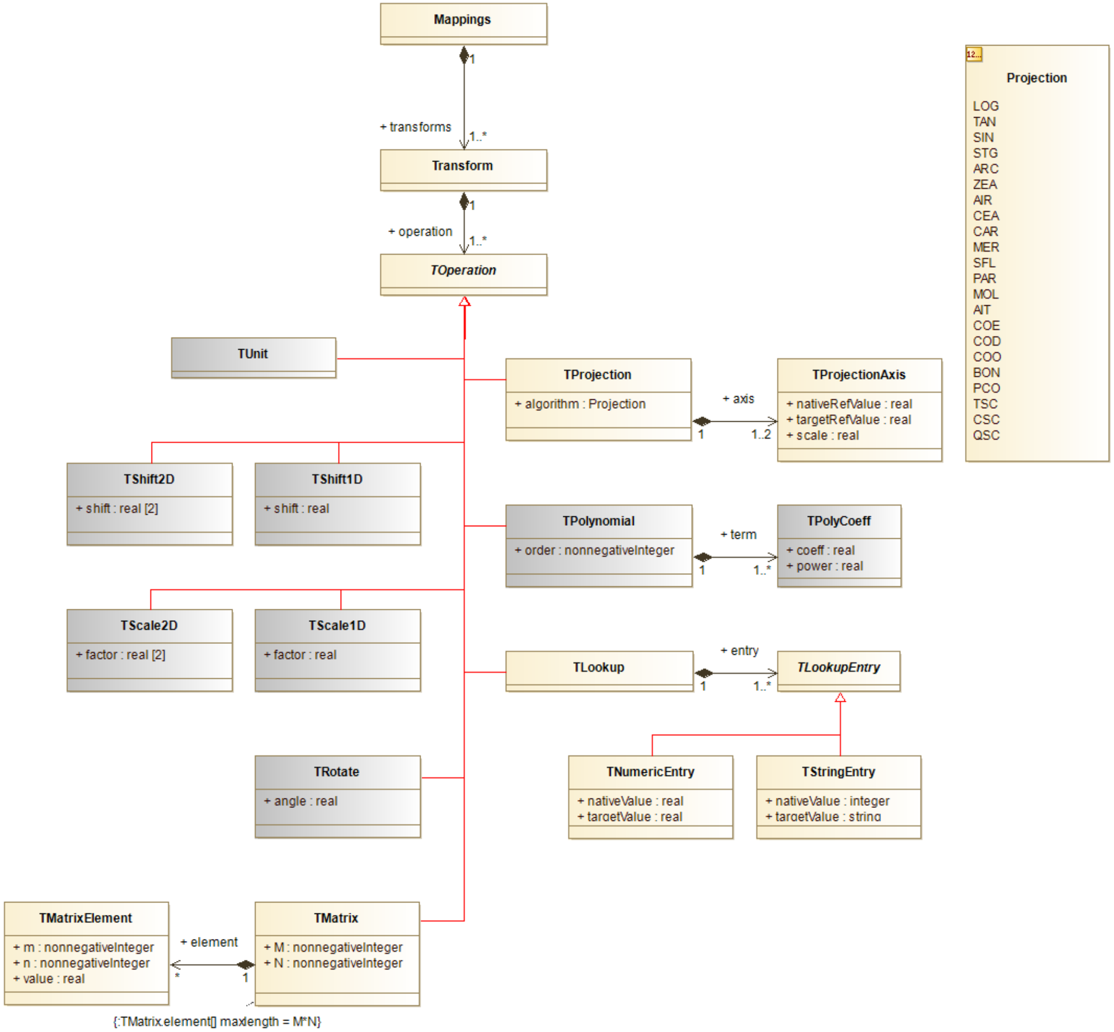
Dy = y-crpix2

X = cd1_1*Dx +cd1_2*Dy

Y =cd2_1*Dx+cd2_2*Dy

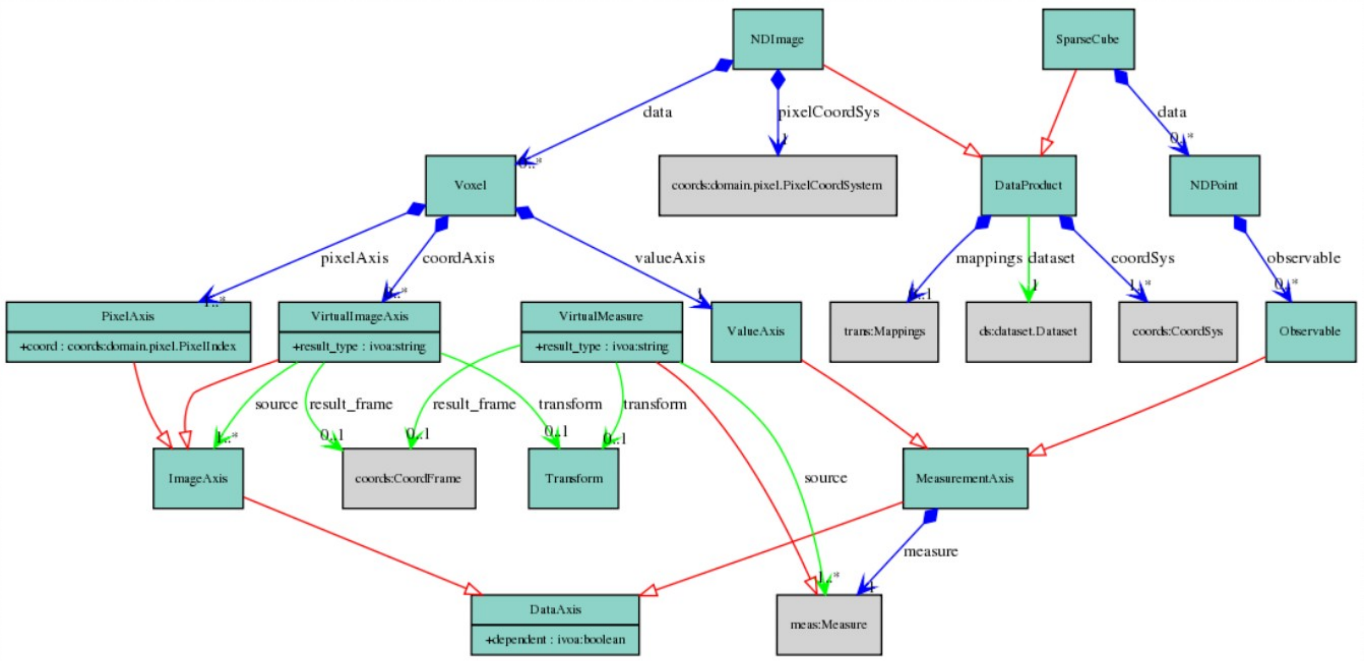
(lon,lat) = deProj(proj,X,Y)

STC2 transform model



- Transforms made of successive ordered operations =
  - Translations = Tshift2D, Tshift1D
  - Linear transformation = Tmatrix
  - Polynomial transformation =Tpolynomial
  - Projection = Tproj (Projection)
  - Scaling = TScale2D,TScale1D
  - Rotating = TRotate
  - ….

Cube data model, relationship to STC transform



cube data model

In Ndimages :

- Each Voxel has
  - (native) PixelAxis,
  - Target CoordFrame
  - VirtualImageAxis
  - and ValueAxis

- VirtualImageAxis states for World Coordinates (or intermediary Coordinates)

Simple WCS use case
STC transform representation (1)

PixelAxis x,y and VirtualAxis to frame « ICRS » and transform made of 3 operations

1 Tshift2D

- shift[0] = -crpix1
- shift[1] = -crpix2

2 Tmatrix

- M=2
- N=2
- TmatrixElement
  - m=1
  - n=1
  - value=CD1_1
- TmatrixElement
  - m=2
  - n=1
  - …….

Simple WCS use case
STC transform representation (2)

3 Tprojection .invert

algorithm =TAN, SIN, etc.

nativeRefVal = crval1

tragetRefVal = crval2

-----> (lon , lat) = (ra, dec)

Coordinate transforms : Polynomial distortions

- Pixels are generally measurement records in the focal plane of the telescope
- The linear scheme may be unsufficient to tackle pixel to intermediate coordinates transformation
- For large Fields of view the focal plane may become (non plane) focal surface
- Introduction of distortions –-> 2D Polynomial operations on each pixel coordinate

Different methods to code distortions in WCS :

failure to standardization (Brian Schmidt, ADASS XXV 2015)

- SIP coefficients (A_n_m,B_n_m, n+m ≤ polynom order): polynomial transformation BEFORE applying bilinear transformation

X' = A_0_0*Dx^{0} * Dy^{0} + A_1_0*Dx^{1} * Dy^{0} +A_1_1*Dx^{1} * Dy^{1} +..

Y' = B_0_0*Dx^{0} * Dy^{0} + B_1_0*Dx^{1} * Dy^{0} +B_1_1*Dx^{1} * Dy^{1} +..

X = cd1_1*X'+cd1_2*Y'   Y= cd2_1*X' +cd2_2*Y'

- « TPV » projection code and SCAMP
  - usage of PVn_m parameters.
  - Polynomial Transformation AFTER bilinear transformation
  - Possible « radial » distortion (skipped below)
  - X' = cd1_1*Dx+cd1_2*Dy   Y’= cd2_1*Dx +cd2_2*Dy
  - X = PV1_0+PV1_1*X'+PV1_2*Y'+PV1_4*X'2 +PV1_5*X'*Y'+PV1_6*Y'2+...
  - Y = PV2_0+PV2_1*Y'+PV2_2*X'+PV1_4*Y'2 +PV1_5*X'*Y'+PV1_6*X'2 +...

Different methods to code distortions in WCS :

failure to standardization (Brian Schmidt, ADASS XXV 2015)

DSS : no usage of WCS parameters, no explicit biLINEAR transform

- FITS KeyWORDS : PPO3,PPO6,XPIXELSIZ,YPIXELSIZ, AMDXn,AMDYn...
- X' (mm) = (pp03-xpixelsiz*x)/1000
- Y'(mm) = (ypixelsiz*y-pp06)/1000
- X = amdx1*X'+amdx2*Y'+amdx3+amdx4*X'^{2}+amdx5*x*y+…
- Y = amdy1*Y'+amdy2*X'+amdy3+amdy4*Y'^{2}+amdy5*x*y+…
- (rad,dec) =deProj(TAN ,X,Y)

Can STC provide an homogenous description for all these « flavors » ?

SIP-like WCS
STC transform representation (1)

PixelAxis x,y and VirtualAxis to frame « ICRS » and transform made of 4 operations

1 Tshift2D

shift[0] = -crpix1

shift[1] = -crpix2

SIP-like WCS
STC transform representation (2)

2 TPolynomialOp *(Proposed extension because current draft only has 1D polynomials)*

```
        Tpolynomial2D[0]
                Order = n
                TpolyCoeff2D
                        coeff = A_2_0
                        power[0]=2
                        power[1]=0
                TpoyCoeff2D
                        coeff=A_1_1
                        power[0]=1
                        power[1]=1
                          ……….
         Tpoynomial2D[1]
                Order = p (may be different than n)
                TpolyCoeff2D
                        coeff = B_2_0
                        power[0]=2
                        power[1]=0
                TpoyCoeff2D
                        coeff=B_1_1
                        power[0]=1
                        power[1]=1
                          ……
```

SIP-like WCS
STC transform representation (3)

3 **Tmatrix**

- M=2
- N=2
- TmatrixElement
  - m=1
  - n=1
  - value=CD1_1
- TmatrixElement
  - m=2
  - n=1
  - …….

4 Tprojection .invert

- algorithm =TAN, SIN, etc.
- nativeRefVal = crval1
- tragetRefVal = crval2

-----> (lon , lat) = (ra, dec)

PV-like WCS
STC transform representation (1)

PixelAxis x,y and VirtualAxis to frame « ICRS » and transform made of 4 operations

```
1  Tshift2D
                    shift[0] = -crpix1
                    shift[1] = -crpix2

3  Tmatrix
                    M=2
                    N=2
                    TmatrixElement
                        m=1
                        n=1
                        value=CD1_1
                    TmatrixElement
                        m=2
                        n=1
                        …….
```

PV-like WCS
STC transform representation (2)

3 TPolynomialOp *(Proposed extension because current draft only has 1D poiynomials)*

```
Tpolynomial2D[0]
    Order = n
    TpolyCoeff2D
        coeff = PV_1_1
        power[0]=0
        power[1]=0
    TpoyCoeff2D
        coeff=PV_1_2
        power[0]=1
        power[1]=0
      ……….
Tpoynomial2D[1]
    Order = n
    TpolyCoeff2D
        coeff = PV_2_0
        power[0]=0
        power[1]=0
    TpoyCoeff2D
        coeff=PV_2_1
        power[0]=0
        power[1]=1
      …….
```

PV-like WCS
STC transform representation (3)

4 Tprojection .invert

algorithm =TAN, SIN, etc.

nativeRefVal = crval1

tragetRefVal = crval2

-----> (lon , lat) = (ra, dec)

DSS-like FITS header solution-> STC transform representation (1)

PixelAxis x,y and ViirtualAxis to frame « ICRS » and tranform made of 5 operations

1 Tscale2D

factor[0] = xpixelsiz

factor[1] = ypixelsiz

2 Tshift2D

shift[0]=-pp03

shift[1]=-pp06

3 Tscale2D

factor[0] = -1/1000

factor[1] = 1/1000

DSS-like FITS header solution-> STC transform representation (2)

4 TPolynomialOp *(Proposed extension because current draft only has 1D poiynomials)*

```
Tpolynomial2D[0]
    Order = 3
    TpolyCoeff2D
        coeff = AMDX1
        power[0]=1
        power[1]=0
    TpoyCoeff2D
        coeff=AMDX2
        power[0]=0
        power[1]=1
      ……….
Tpoynomial2D[1]
    Order = 3
    TpolyCoeff2D
        coeff = AMDY1
        power[0]=0
        power[1]=1
    TpoyCoeff2D
        coeff=AMDY2
        power[0]=1
        power[1]=0
      ……
```

DSS-like FITS header solution-> STC transform representation (3)

5 Tprojection .invert

algorithm =TAN

nativeRefVal = crval1

tragetRefVal = crval2

-----> (lon , lat) = (ra, dec)

Conclusion

- Radial distortion to be considered (3D ->2D transform???)
- Extension of Polynomial transform to 2D → 2D (or 3D → 2D) needed
- Apart from that, STC transform provides a unified representation for building transformations by combination of simple operations in any order : Yes !