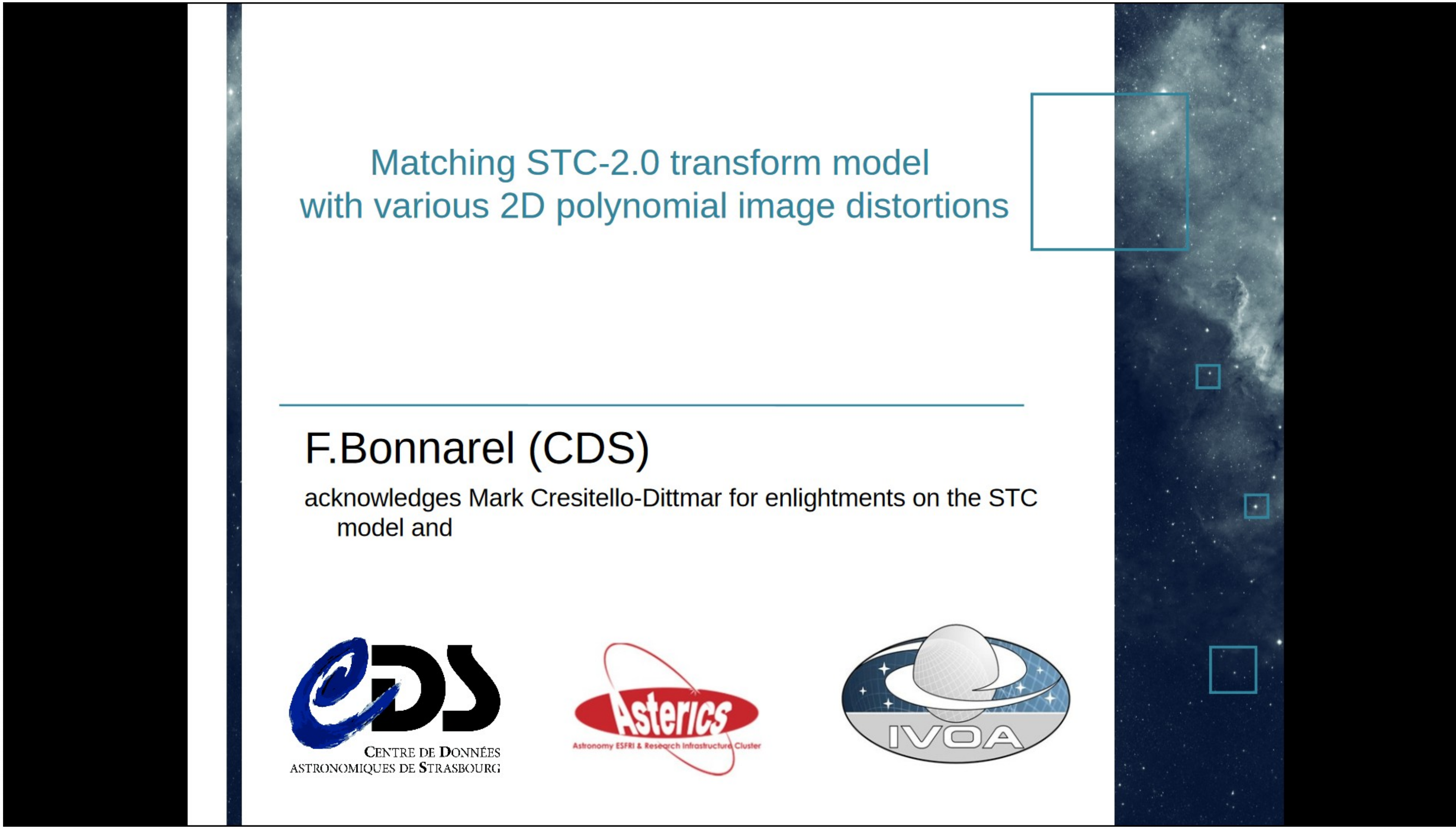
Matching Transform model with various 2D polynomial image distortions Paris upgrade

F.Bonnarel (CDS)

Centre de Données Astronomiques de Strasbourg

Follow-up of my College Park talk



Transform datamodel 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

- « Transform » allows to represent these coordinate transforms

- Hey !!! Isn't that done via WCS keywords already ?
  - Yes for linear, Not satisfactory for distortions !!!!

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'

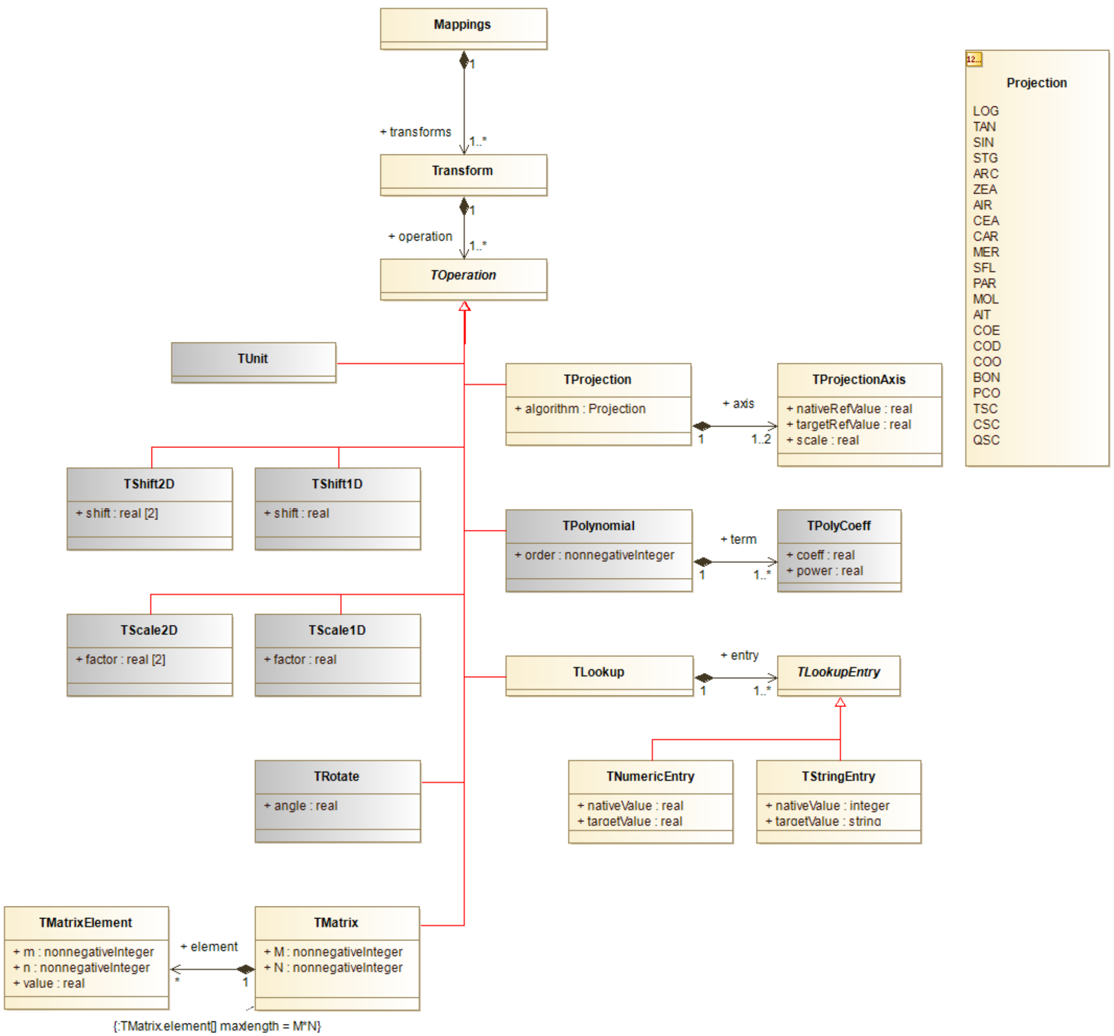
- **« TPV » projection code and SCAMP**
  - **usage of PVn_m parameters.**
  - **Polynomial Transformation AFTER bilinear transformation**
  - **Possible « radial » distortion (skipped below)**
  - **X' = cd1_1*Dx+cd1_2*Dy   Y’= cd2_1*Dx +cd2_2*Dy**
  - **X = PV1_0+PV1_1*X'+PV1_2*Y'+PV1_4*X'2 +PV1_5*X'*Y'+PV1_6*Y'2+...**
  - **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 Trasnform datamodel provide an homogenous description for all these « flavors » ?**

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
  - ….

College Park 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 !

New Trans model

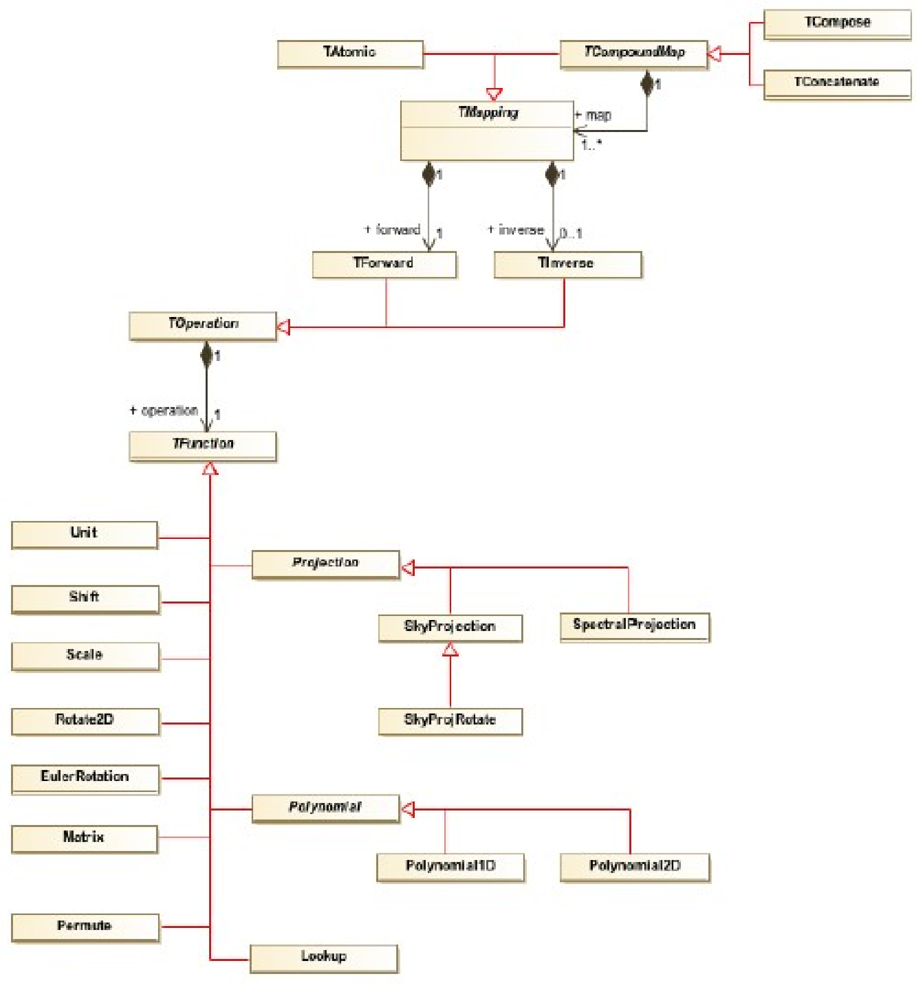


Figure 2: Overview of WCS Transform model elements

What’s new

- EulerRotation
- Polynomial2D
- Spectral transformation
- Composition (sequence)
- Concatenation (parallel)
- Invert flag and Tforward/Tinvert operations

Simple WCS use case
STC transform representation (1)

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

1 Tconcatenate
    Tinvert = « true »
    1 Tforward.Shift.offset = -crpix1
    2 Tforward.Shift.offset = -crpix2
2 Tforward.Matrix
    Tinvert = « true »
    Tforward.Matrix. M=2
    Tforward.Matrix.N=2
    Tforward.Matrix.element
        Matrix.element.m=1
        Matrix.element.n=1
        Matrix.element.value=CD1_1
    Tforward.Matrix.element
        Matrix.element.m=2
        Matrix.element.n=1
        …….

Simple WCS use case
STC transform representation (2)

3 SkyProjRotate

Tinvert = true

Skyprojrotate.algorithm =TAN, SIN, etc.

Skyprojrotate.referenceValue[0] = crval1

Skyprojrotate.referenceValue[1] = crval2

SIP-like WCS
STC transform representation (1)

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

1 Tconcatenate

1 Tforward.Shift.offset = -crpix1

2 Tforward.Shift.offset = -crpix2

SIP-like WCS
STC transform representation (2)

2 TConcatenate

Tforward.Polynomial2D

Polynomial2D.Order = n
Polynomial2D.term.coeff = A_2_0
Polynomial2D.term.power[0]=2
Polynomial2D.term.power[1]=0

Polynomial2D.term.coeff = A_1_1
Polynomial2D.term.power[0]=1
Polynomial2D.term.power[1]=1

……….

Tforward.Polynomial2D

- **Polynomial2D.Order = n (may be different than n)**

Polynomial2D.term.coeff = B_2_0
Polynomial2D.term.power[0]=2
Polynomial2D.term.power[1]=0

Polynomial2D.term.coeff = B_1_1
Polynomial2D.term.power[0]=1
Polynomial2D.term.power[1]=1

……….

SIP-like WCS
STC transform representation (3)

3) Tforward.Matrix

Tforward.Matrix. M=2

Tforward.Matrix.N=2

Tforward.Matrix.element

Matrix.element.m=1

Matrix.element.n=1

Matrix.element.value=CD1_1

Tdorward.Matrix.element

Matrix.element.m=2

Matrix.element.n=1

…….

4 SkyProjRotate

Skyprojrotate.algorithm =TAN, SIN, etc.

Skyprojrotate.referenceValue[0] = crval1

Skyprojrotate.referenceValue[1] = crval

PV-like WCS
STC transform representation (1)

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

1 Tconcatenate
- **Tforward.shift.offset = -crpix1**
- **Tforward.shift.offset = -crpix2**

3 Tforward.Matrix
- **Matrix.M=2**
- **Matrix.N=2**
- **Matrix.element**
  - **m=1**
  - **n=1**
  - **value=CD1_1**
- **Matrix.element**
  - **m=2**
  - **n=1**
  - **…….**

PV-like WCS
STC transform representation (2)

```
3 TConcatenate
        Tforward.Polynomial2D
            Order = n
            Polynomial2D.term
                coeff = PV_1_1
                power[0]=0
                power[1]=0
            Polynomial2D.term
                coeff=PV_1_2
                power[0]=1
                power[1]=0
              ……….
        Tforward.Poynomial2D
            Order = n
            Polynomial2D.term
                coeff = PV_2_0
                power[0]=0
                power[1]=0
            Polynomial2D.term
                coeff=PV_2_1
                power[0]=0
                power[1]=1
              …….
```

PV-like WCS STC transform representation (3)

4 **SkyProjRotate**
Skyprojrotate.algorithm =TAN, SIN, etc.
Skyprojrotate.referenceValue[0] = crval1
Skyprojrotate.referenceValue[1] = crval2

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

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

1 Tconcatenate$

Tforward.scale

Scale.factor = xpixelsiz

Tforward.scale

Scale.factor = ypixelsiz

2 Tconcatenate

– **Tforward.shift**

-Shift.offset = -pp03

– **Tforward.shift**

– **- Shift.offset = -pp06**

3 Tconcatenate

Tforward.scale

Scale.factor = -1/1000

Tforward.scale

Scale.factor = 1/1000

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

```
4 TConcatenate
        Tforward.Polynomial2D
            Order = 3
            Polynomial2D.term
                coeff = AMDX1
                power[0]=1
                power[1]=0
            Polynomial2D.term
                coeff=AMDX2
                power[0]=0
                power[1]=1
             ……….
        Tforward.Poynomial2D
            Order = 3
            Polynomial2D.term
                coeff = AMDY1
                power[0]=0
                power[1]=1
            Polynomial2D.term
                coeff=AMDY2
                power[0]=1
                power[1]=0
             ……
```

DSS-like FITS header solution-> Transform representation (3)

5 SkyProjRotate

Skyprojrotate.algorithm =TAN, SIN, etc.

Skyprojrotate.referenceValue[0] = crval1

Skyprojrotate.referenceValue[1] = crval2

Possible serialisations

1) json : from Top mapping element to leaves

--→ hierarchy of embedded json objects

Composition : sequence

Concatenation : array

Object names = « vo-dml ids »

2) VODML-lite mapping (L.Michel proposal) attempt for annotating tables conatining more than 1 transformation

Conclusion

--

1 **Most of the problems for polynomial dsitortions solved.**

2 Why do we have Toperation/Tfunction separation (One Operation contains only one function and nothing else)

3 ° JSON (Or YAML) « Official » serialisation to be considered to go with individual images (FITEX extension?)

4) VODML-lite mapping feasible
