Australian Government | ansto

Optimised Multi-Dimensional Image Scanning With RASCAN

Nader Afshar, Martin De Jonge, David Paterson, Daryl Howard, Andrew Starritt

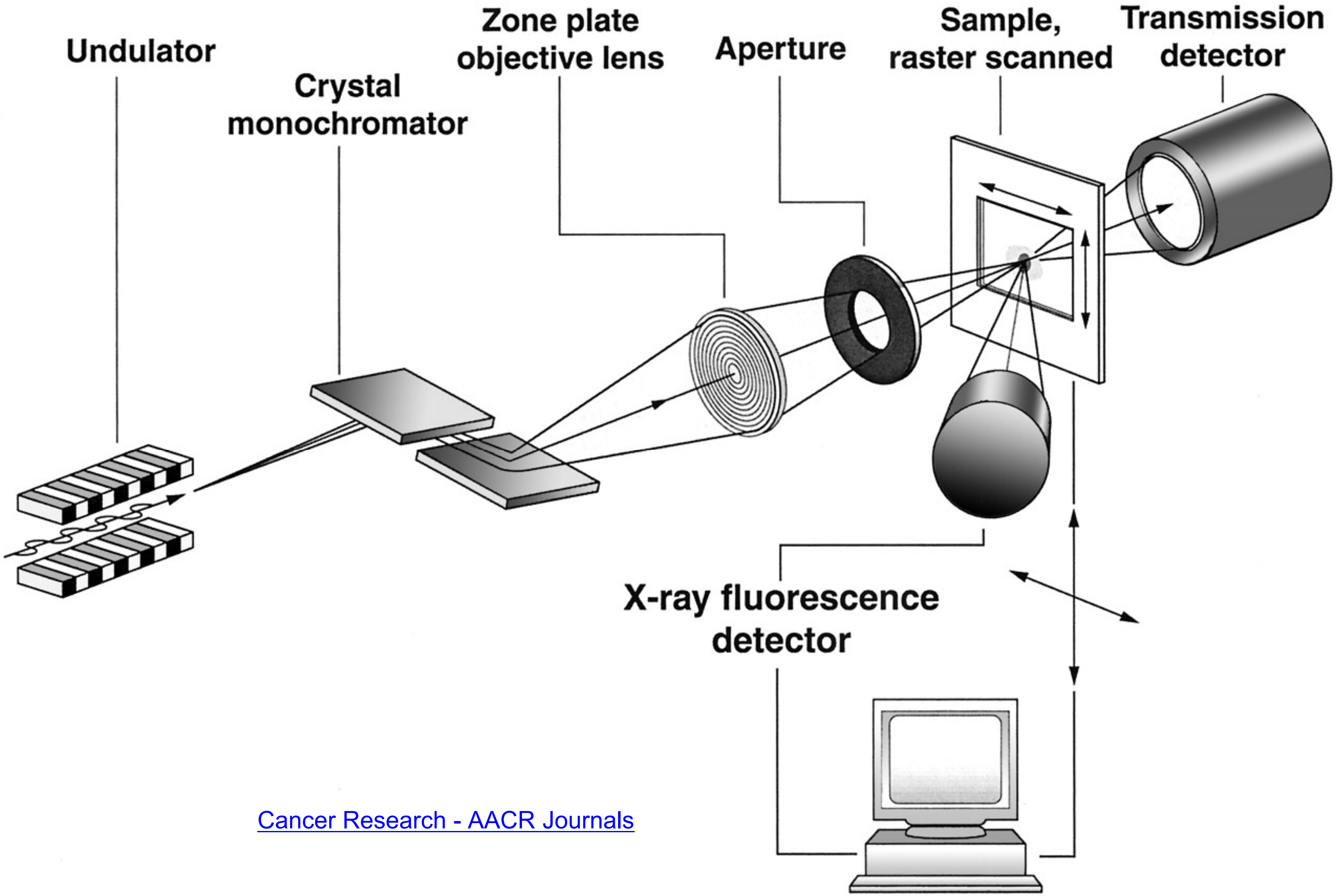
VICTORIA State Government

Australian Synchrotron

Synchrotron X-ray Fluorescence Microscopy

Australian Government

ansto

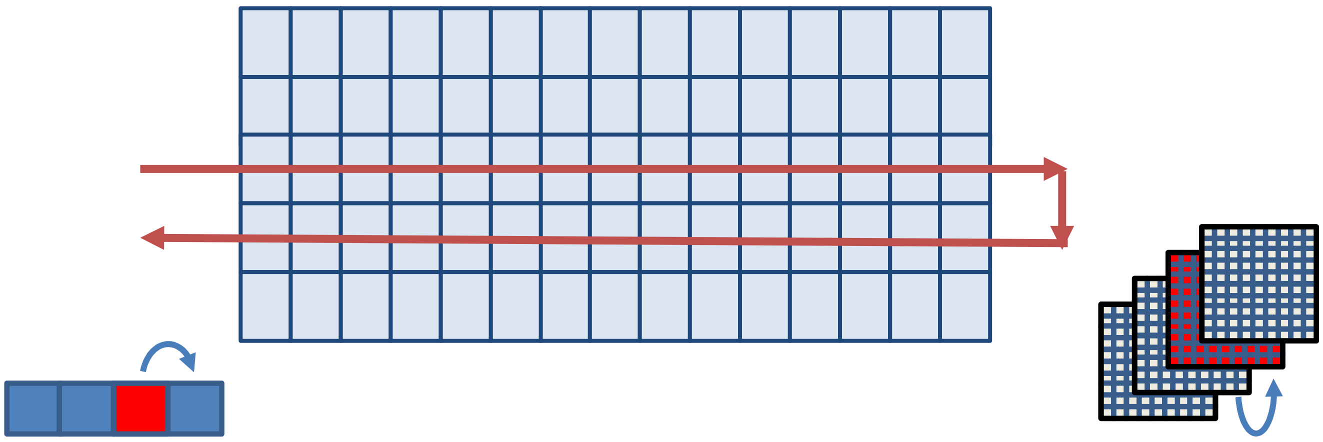


[Cancer Research - AACR Journals]

synchrotron.org.au

Australian Synchrotron

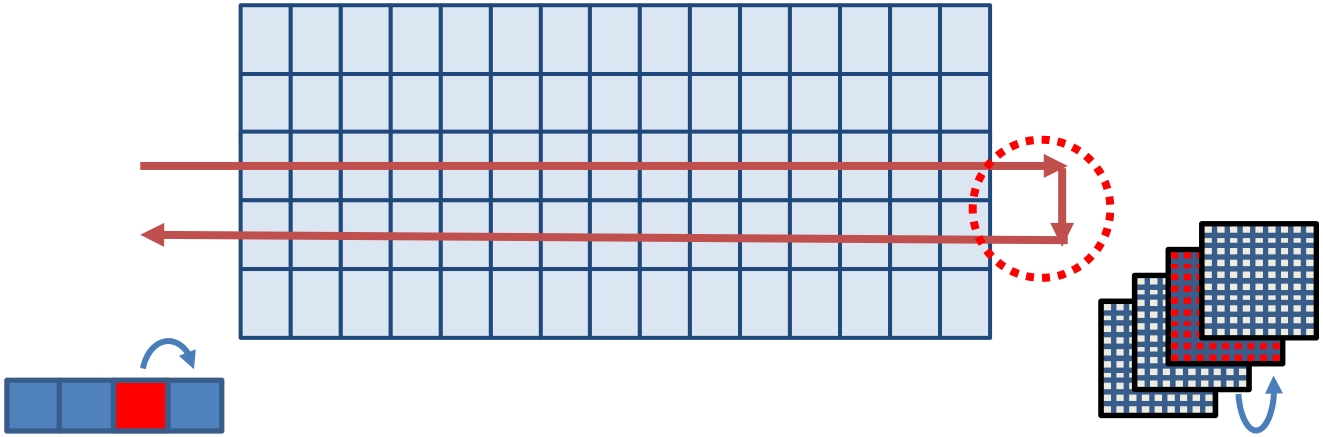
Scanning Microscopy: Overheads



	Pixel by Pixel	Line by Line (1000 pixels)	3rd dimension (eg: angle, energy)
Typical dwell	**1 ms**	**1000 ms**	**1000 s** (1000 Lines)
Motion Overhead	**150 ms** typical	**355 ms** @ XFM	**2 s + 355 s**

- Detectors have become faster by 3 orders of magnitude
- Science is now limited by motion overheads

Scanning Microscopy: Overheads



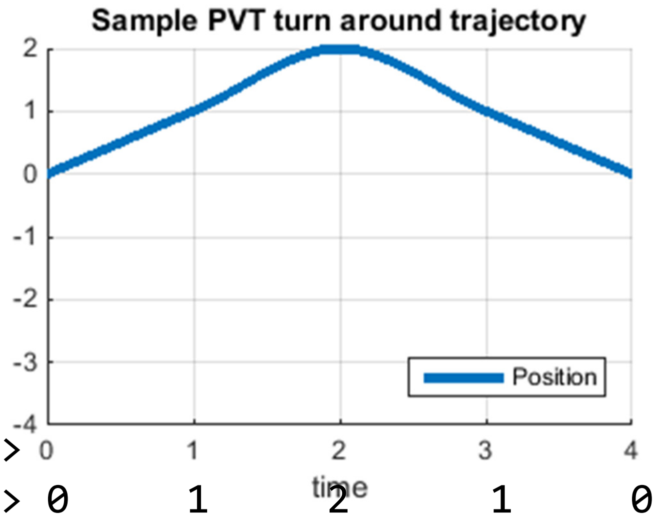
	Pixel by Pixel	Line by Line (1000 pixels)	3^{rd} dimension (eg: angle, energy)
Typical dwell	1 ms	**1000 ms**	**1000 s** (1000 Lines)
Motion Overhead	**150 ms** typical	**355 ms** @ XFM	**2 s + 355 s**

Overheads > 35%

- Detectors have become faster by 3 orders of magnitude
- Science is now limited by motion overheads

nD Fly-scan formulation

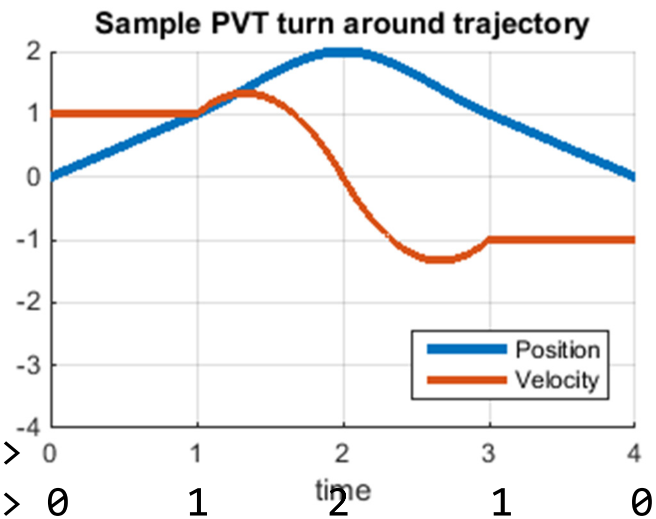
- Common approach: Use arbitrary motion trajectories *implemented* as an array of Position-Velocity-Time (PVT) elements **{ Point[1..n],Velocity[1..n],Time[1..n] }**



```
T-> 0      1      2      3      4
P-> 0      1      2      1      0
V-> 1      1      0     -1     -1
```

nD Fly-scan formulation

- Common approach: Use arbitrary motion trajectories *implemented* as an array of Position-Velocity-Time (PVT) elements **{ Point[1..n],Velocity[1..n],Time[1..n] }**



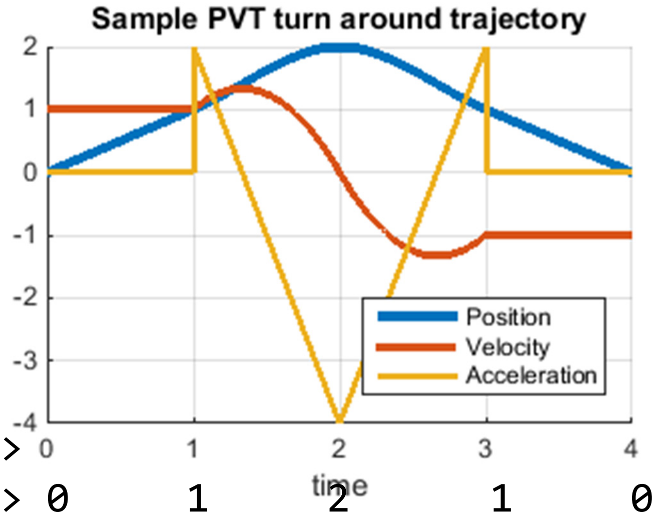
```
T-> 0        1        2        3        4
P-> 0        1        2        1        0
V-> 1        1        0       -1       -1
```

nD Fly-scan formulation

Australian Government

ansto

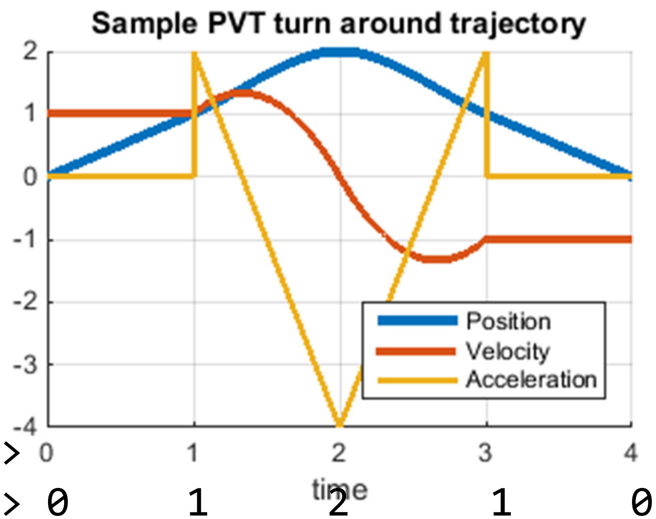
- Common approach: Use arbitrary motion trajectories *implemented* as an array of Position-Velocity-Time (PVT) elements **{ Point[1..n],Velocity[1..n],Time[1..n] }**



T->	0	1	2	3	4
P->	0	1	2	1	0
V->	1	1	0	-1	-1

nD Fly-scan formulation

- Common approach: Use arbitrary motion trajectories *implemented* as an array of Position-Velocity-Time (PVT) elements **{ Point[1..n],Velocity[1..n],Time[1..n] }**

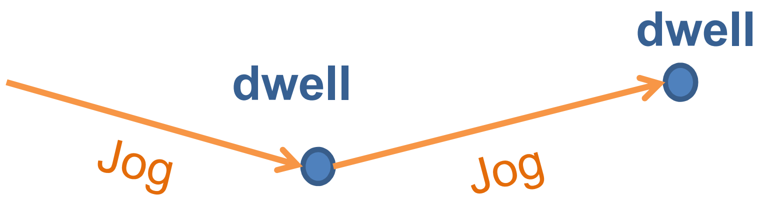


T->	0	1	2	3	4
P->	0	1	2	1	0
V->	1	1	0	-1	-1

➢ **Setting PVT points is NOT trivial. User application needs to deal with kinematics.**

nD Fly-scan formulation

- Step-scan motion is a sequence of "Dwell at a Point" elements
{ Point[1..n] }, Dwell[1..n]=DwellTime

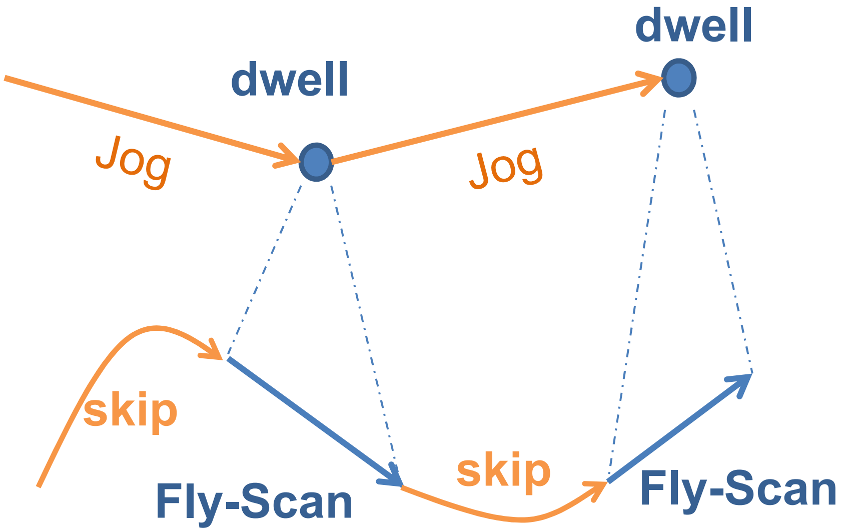


Science request **Overhead motion**

nD Fly-scan formulation

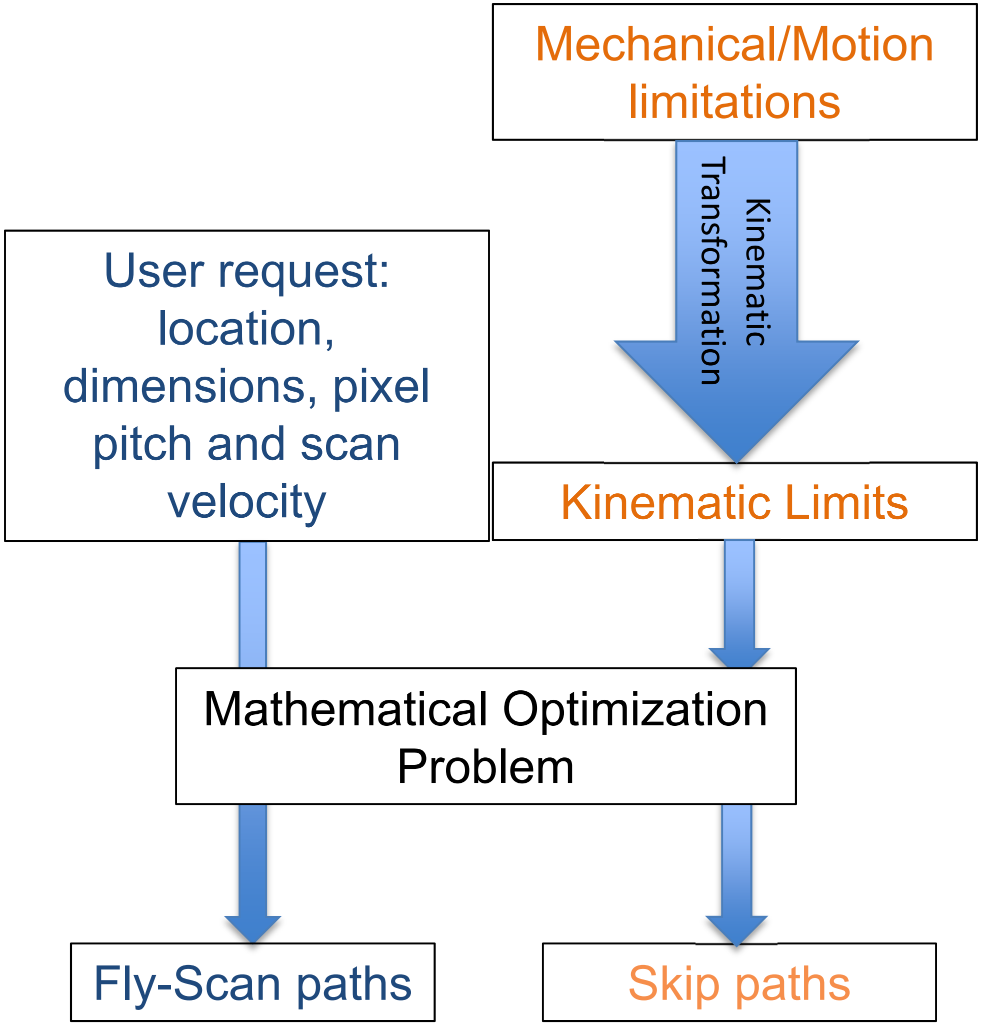
- Step-scan motion is a sequence of "Dwell at a Point" elements
  { Point[1..n] }, Dwell[1..n]=DwellTime

- Fly-scan motion can be *formulated* as a sequence of "Scan along a Vector" elements
  { Vector[1..n] }, Vel[1..n]=ScanVel



Science request **Overhead motion**

2D Raster Imaging



Mechanical/Motion limitations

Kinematic Transformation

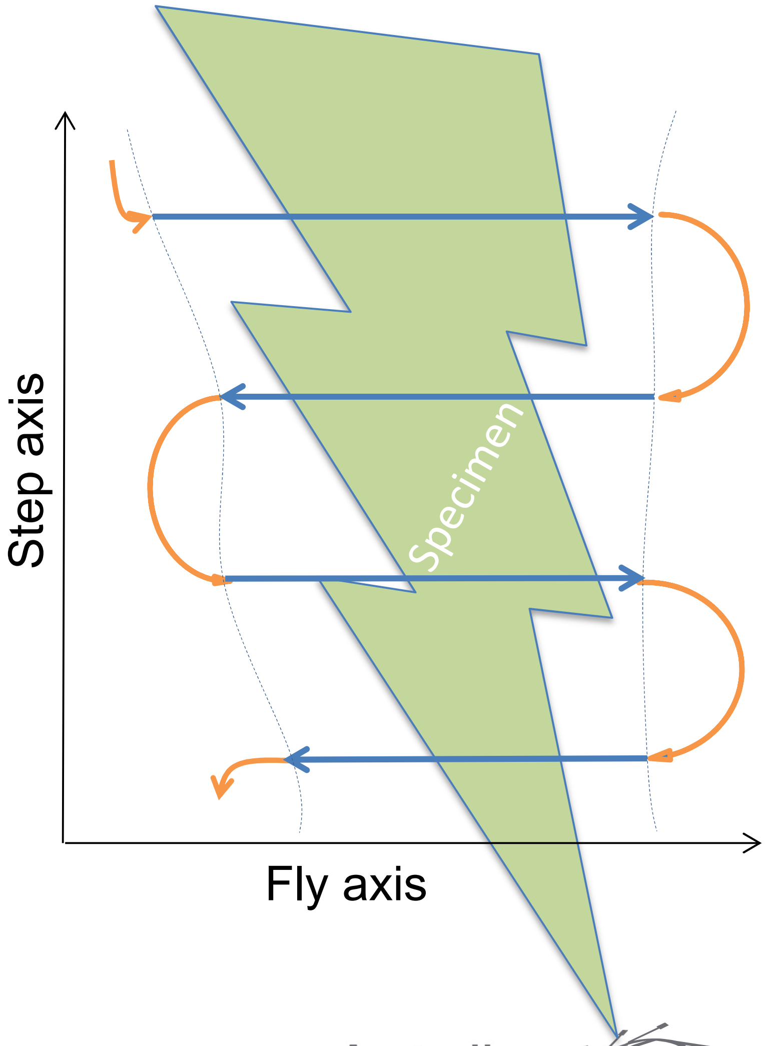
Kinematic Limits

User request: location, dimensions, pixel pitch and scan velocity

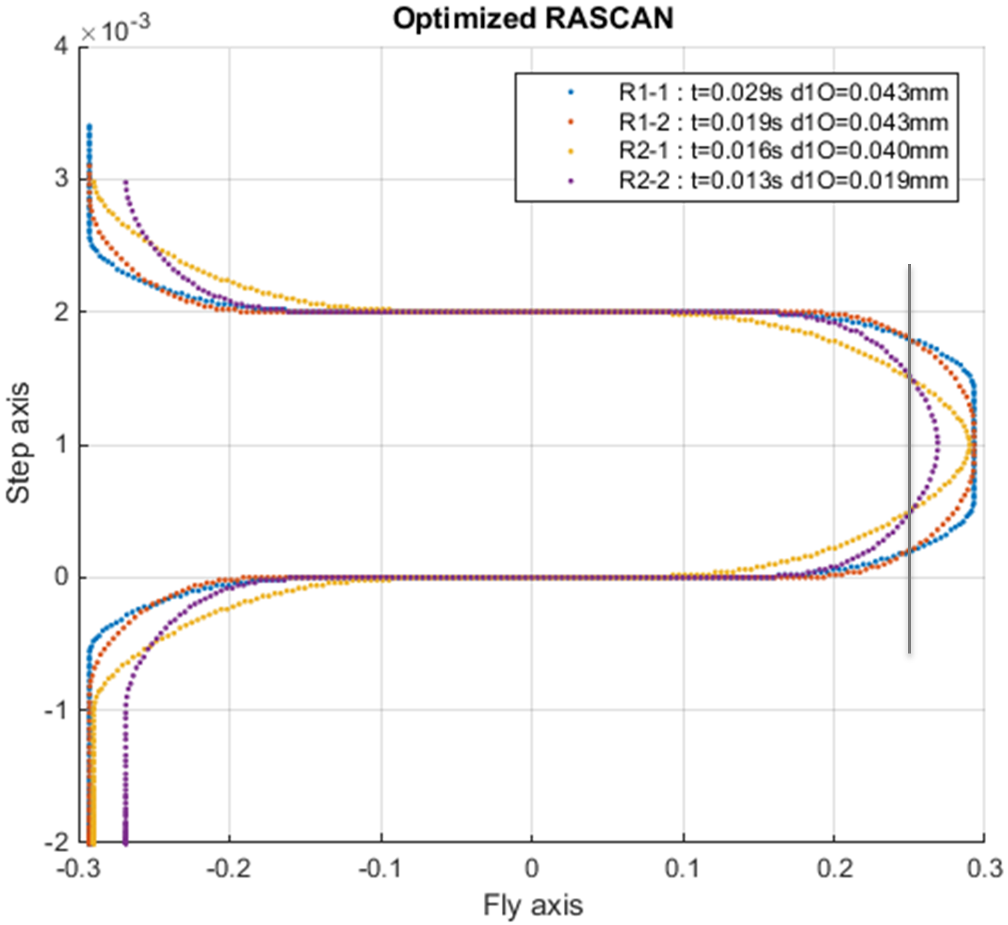
Mathematical Optimization Problem

Fly-Scan paths

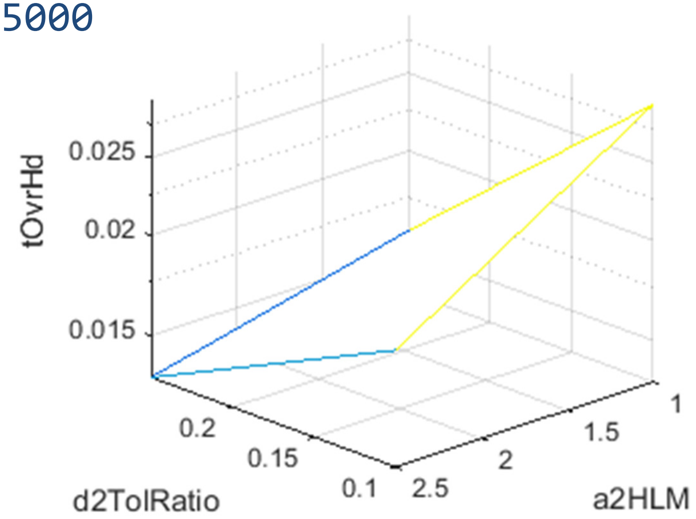
Skip paths



Optimized solutions



```
d2Step=0.0020
v1Scan=4
v1TolRatio=0.8000
d2TolRatio=[0.1,0.25];

a1HLM=500; pE1Res=0.0001;
pE2Res=0.00002;
a2HLM=[1,2.5];

d1Span=0.5000
d2Start=0
nLines=2
```

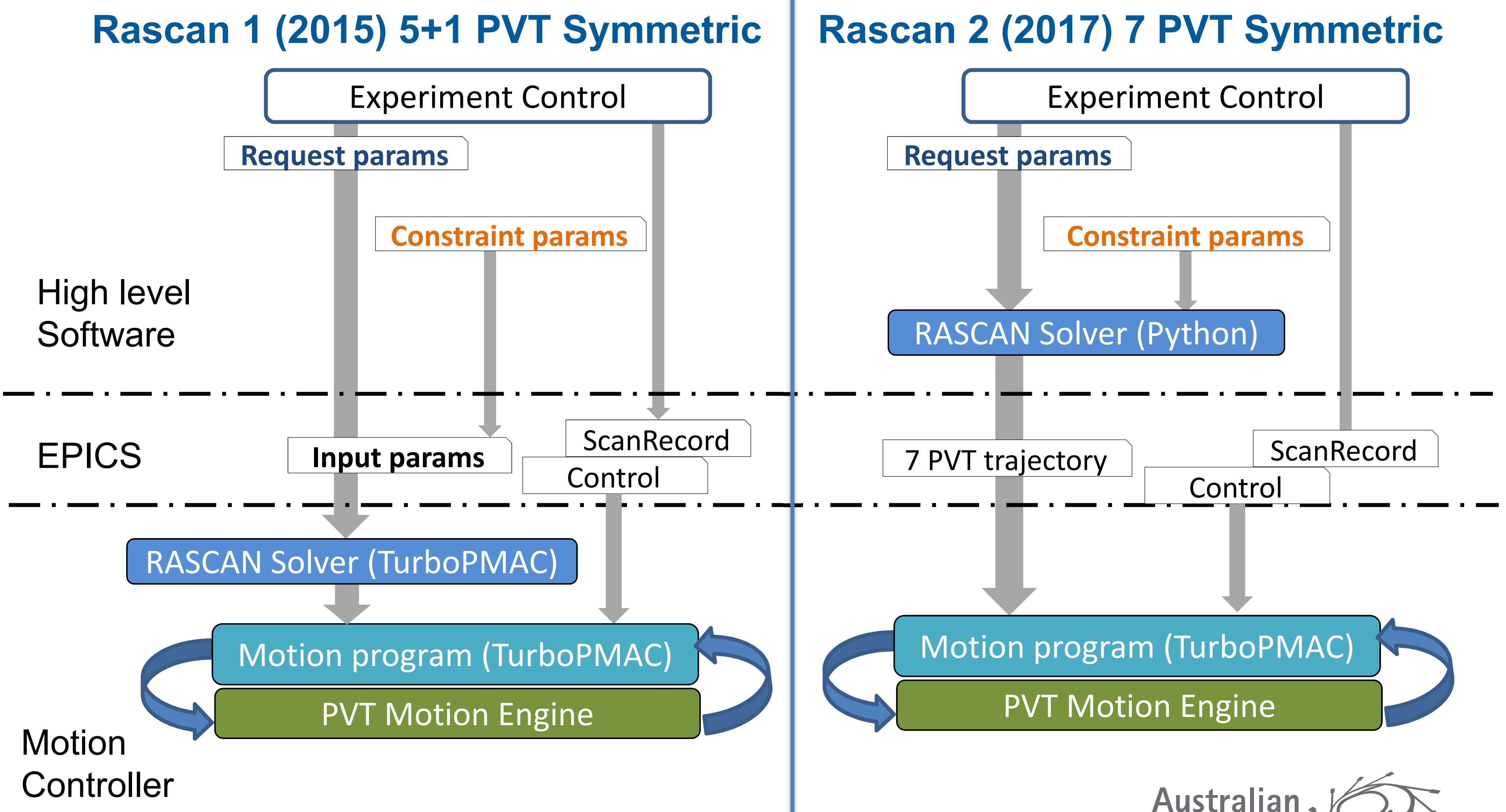


Naming: Fly-Axis is 1, Step-Axis is 2
Units are mm and sec

RASCAN Implementation

Australian Government

ansto

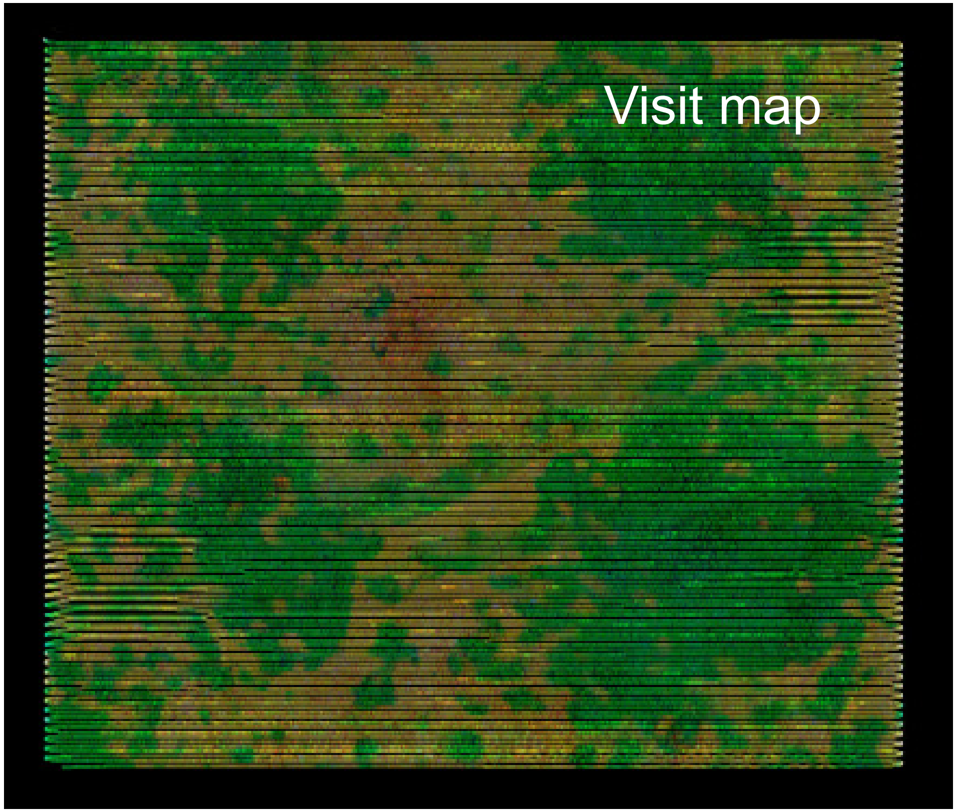


RASCAN Motion Program

```
while (lineN < lineEnd )
    lineDir = (lineN is even? 1 else -1)

    INC PVT (tOut)      X (lineDir * d1Out): (lineDir * v1Out)    Y (d2Out):(v2In)
    INC PVT (tIn)   X (lineDir * d1In): (lineDir * v1In)   Y (d2In):(0)

    INC PVT ( tMid[ lineN ])  X (lineDir * tMid[ lineN ] * v1Scan):(lineDir * v1Scan ) Y (0):(0)

    INC PVT (tIn)   X (lineDir * d1In): (lineDir * v1Out)        Y (d2In):(v2In)
    INC PVT (tOut)      X (lineDir * d1Out): (0)                 Y (d2Out):(v2Out)

    lineN = lineN+1
    ...
endwhile
```

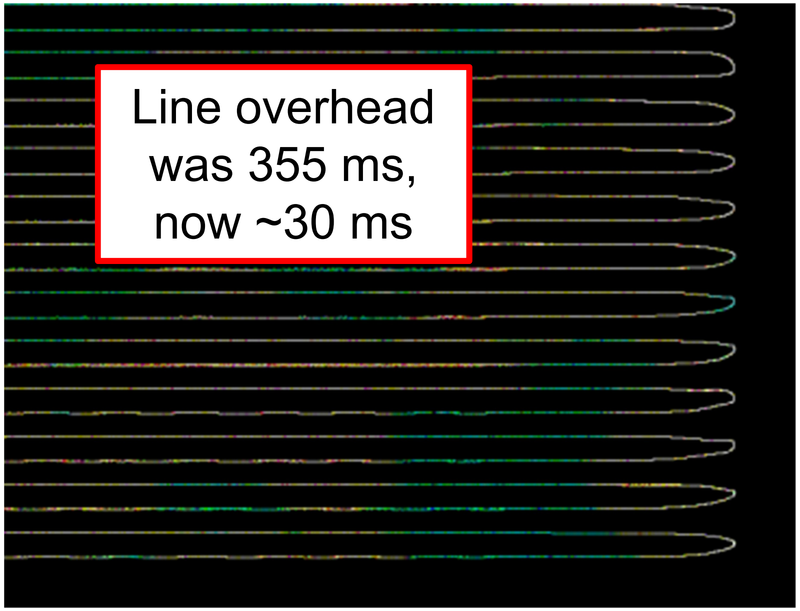
RASCAN 1.0 results at XFM

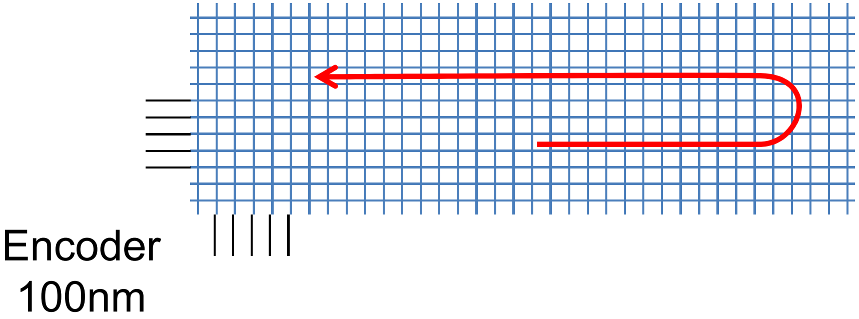
Australian Government



0.5 mm wide, 4 mm/s, 2x2um, 250 lines

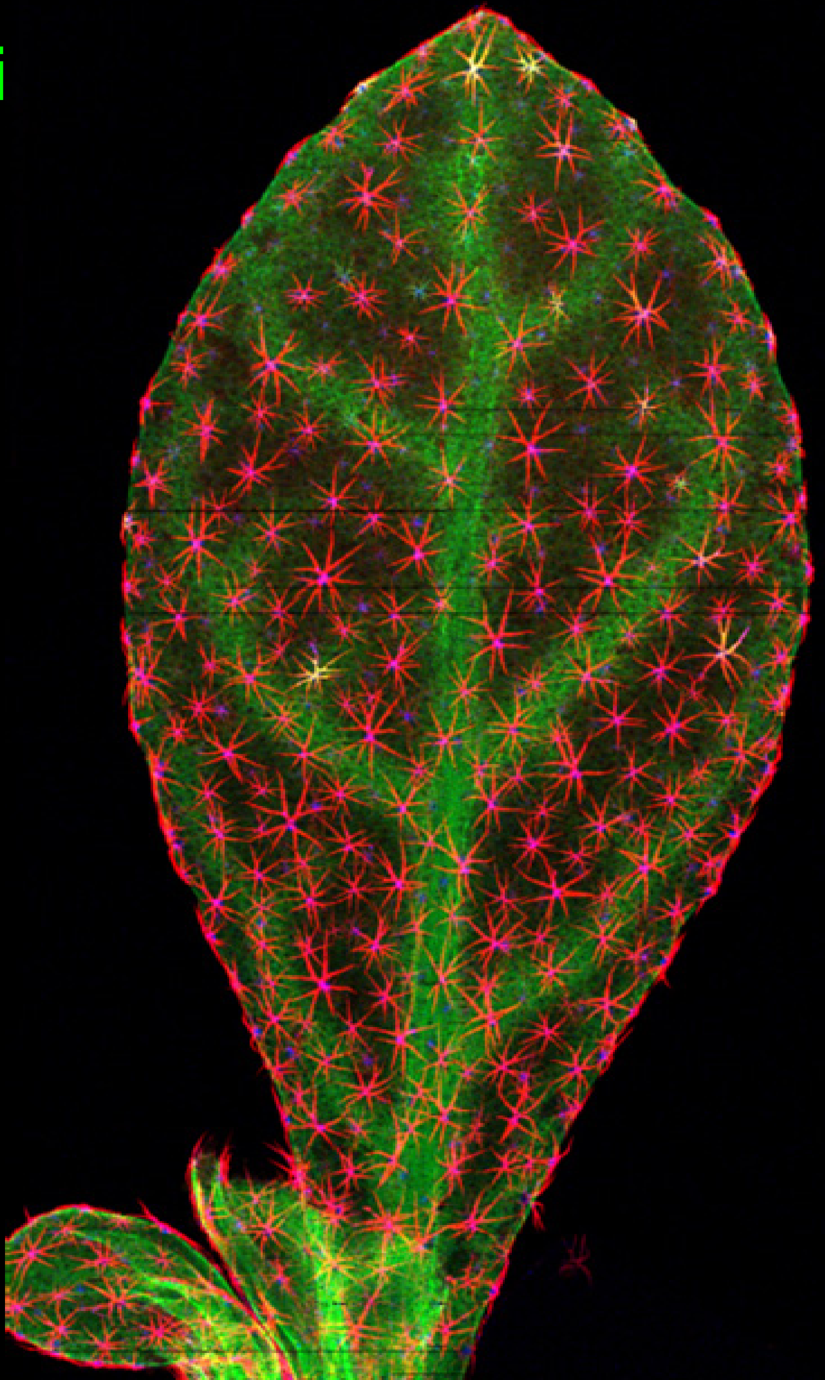
	Line by Line	Rascan
Line O/H	355 ms / line	~30 ms / line
Daily O/H	1-4 hours	10-30 min


Line overhead was 355 ms, now ~30 ms



Controller: Delta-Tau GeoBrickLV
Implementation: Nader Afshar
Stage design: Walsh, Afshar, LeGuen

Ca Ni
Mn



Alyssum leaf

van der Ent, Harris (2016)

2300 * 3500 pix = 8 Mpix,
dwell = 200 µs, dx = dy = 2 µm

Exposure time = 25 min

Duration = 31 min

Estimated duration without RASCAN = 46 min

Summary

- Separation of “**required**” and “**overhead**” components
- Effective mathematical formulation for Fly-scan optimisation problem
- Trajectory is optimized for tracking precision as well as speed
- Motion problem solved below user application level
- Implementation is robust and scalable
- Run-time control over Width, Pause/Resume, Shutter, etc.

- Asymmetric skip trajectories
- Adaptive estimation/correction of kinematic features limits

Thank you for your attention

nader.afshar@synchrotron.org.au

For information on job opportunities at the Australian Synchrotron:
paul.martin@synchrotron.org.au