

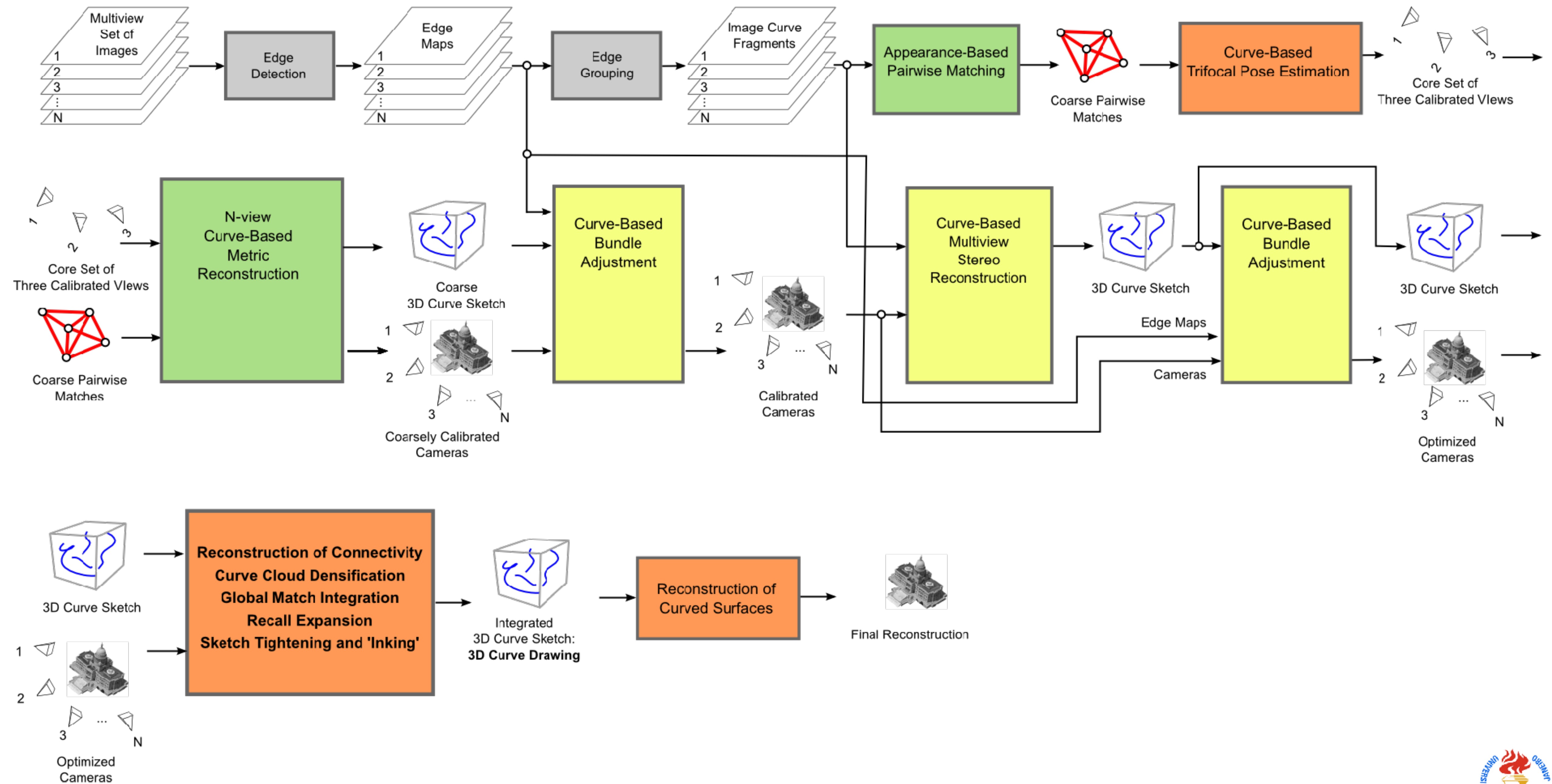
Latest version is on Google Docs:

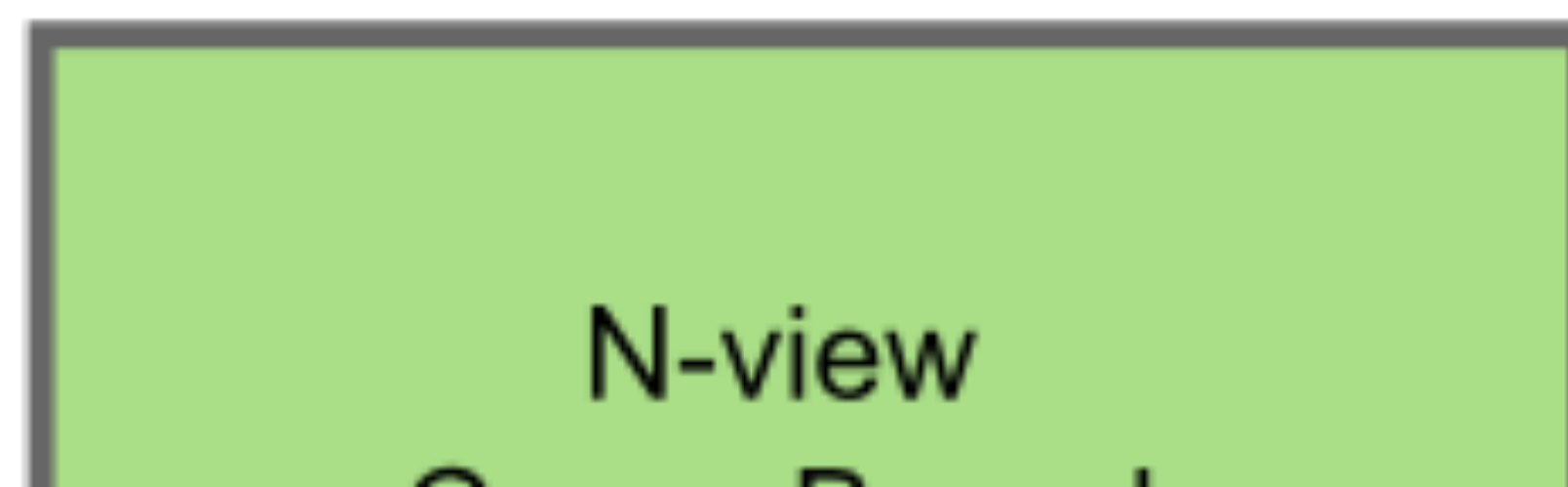
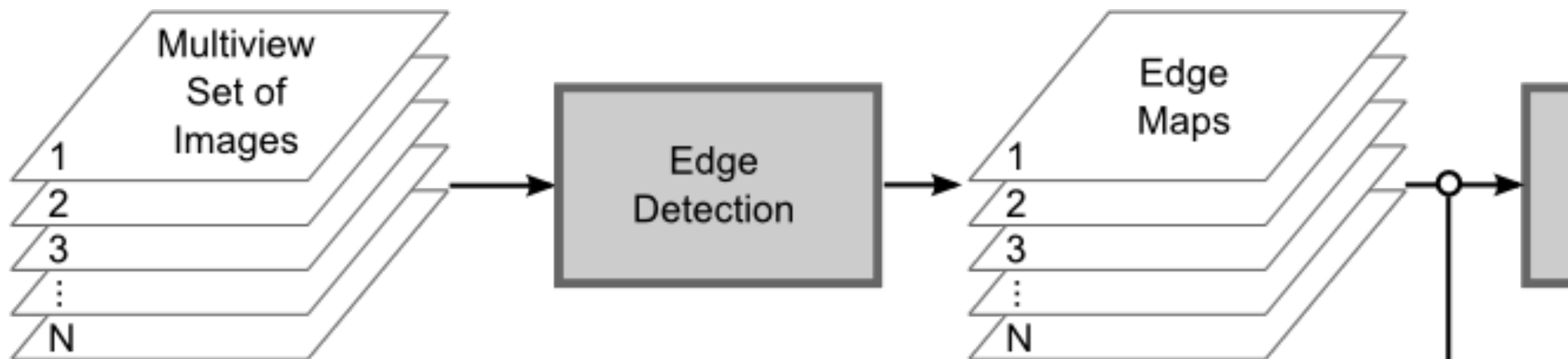
https://docs.google.com/presentation/d/1v6S7F-9oL92CpOHQIcLdq8eL13A_PrQiXaCyId27LVs

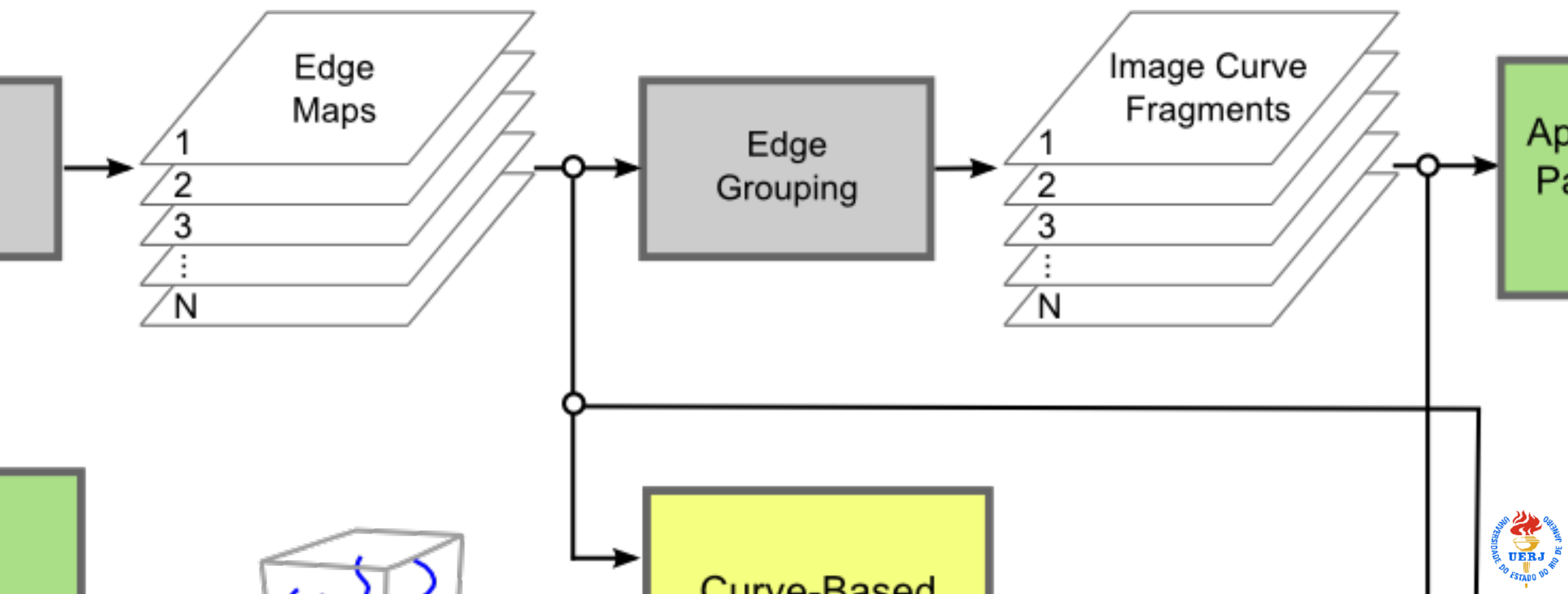
curve-based

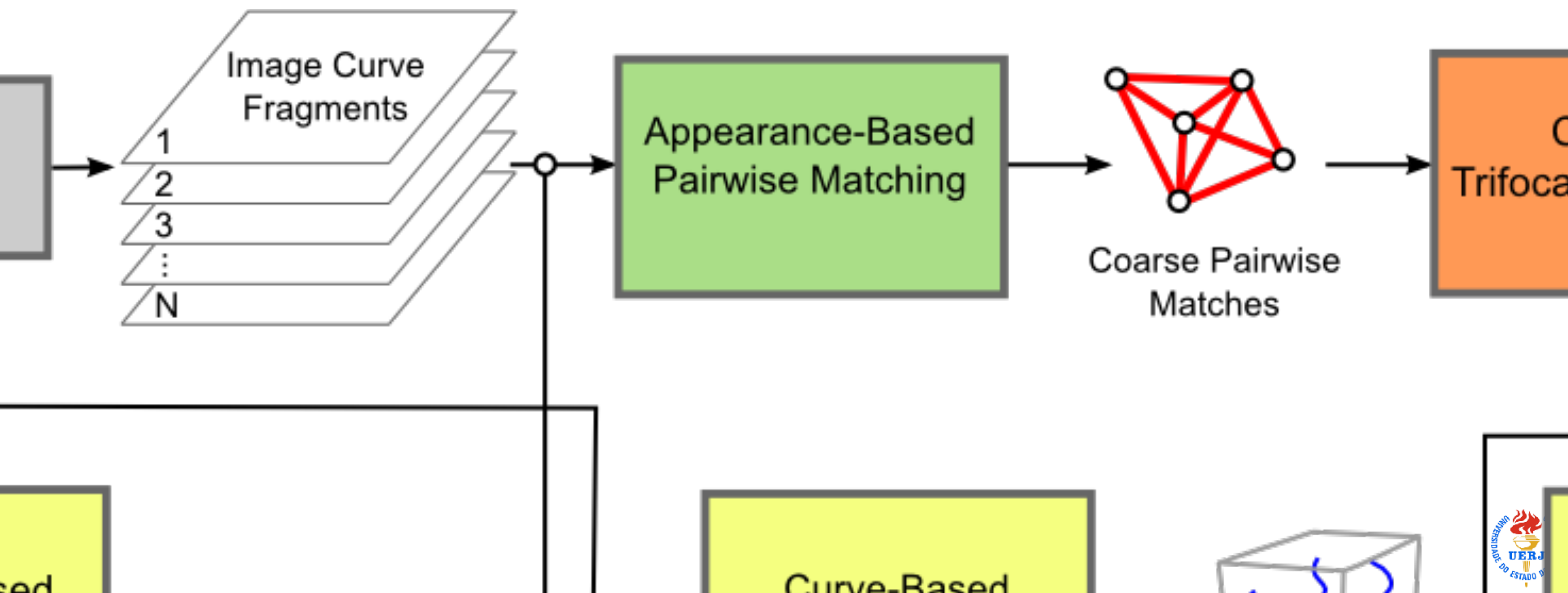
Total 3D vision system

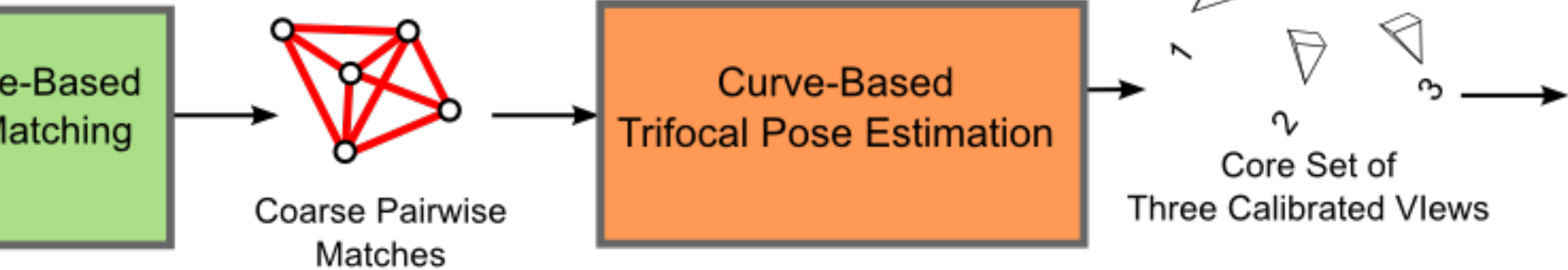
Code effort



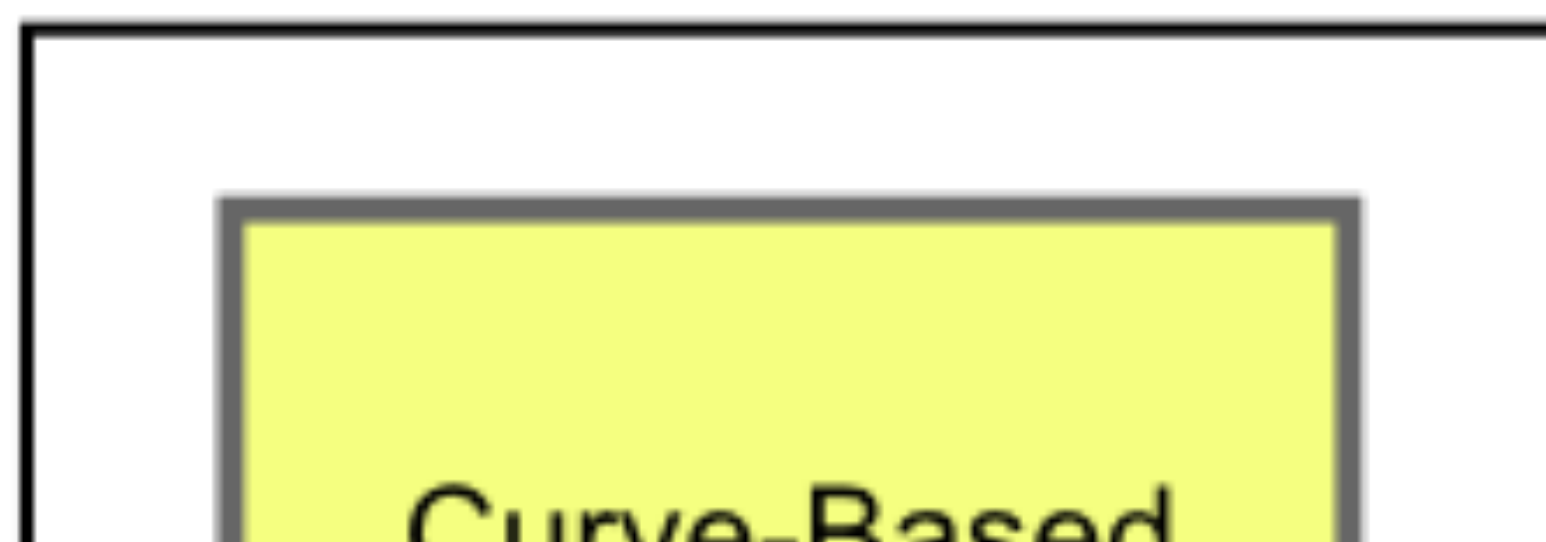


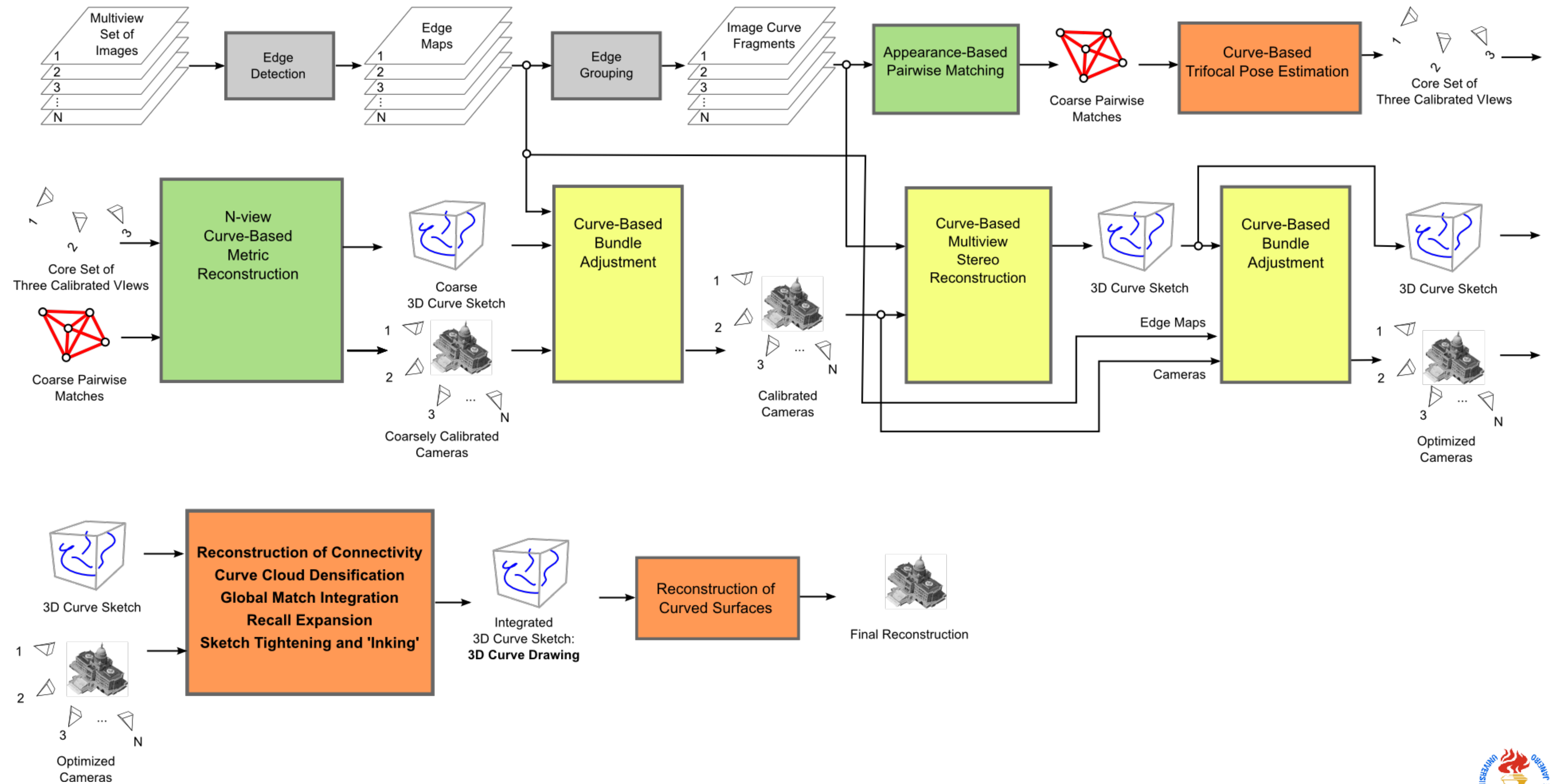


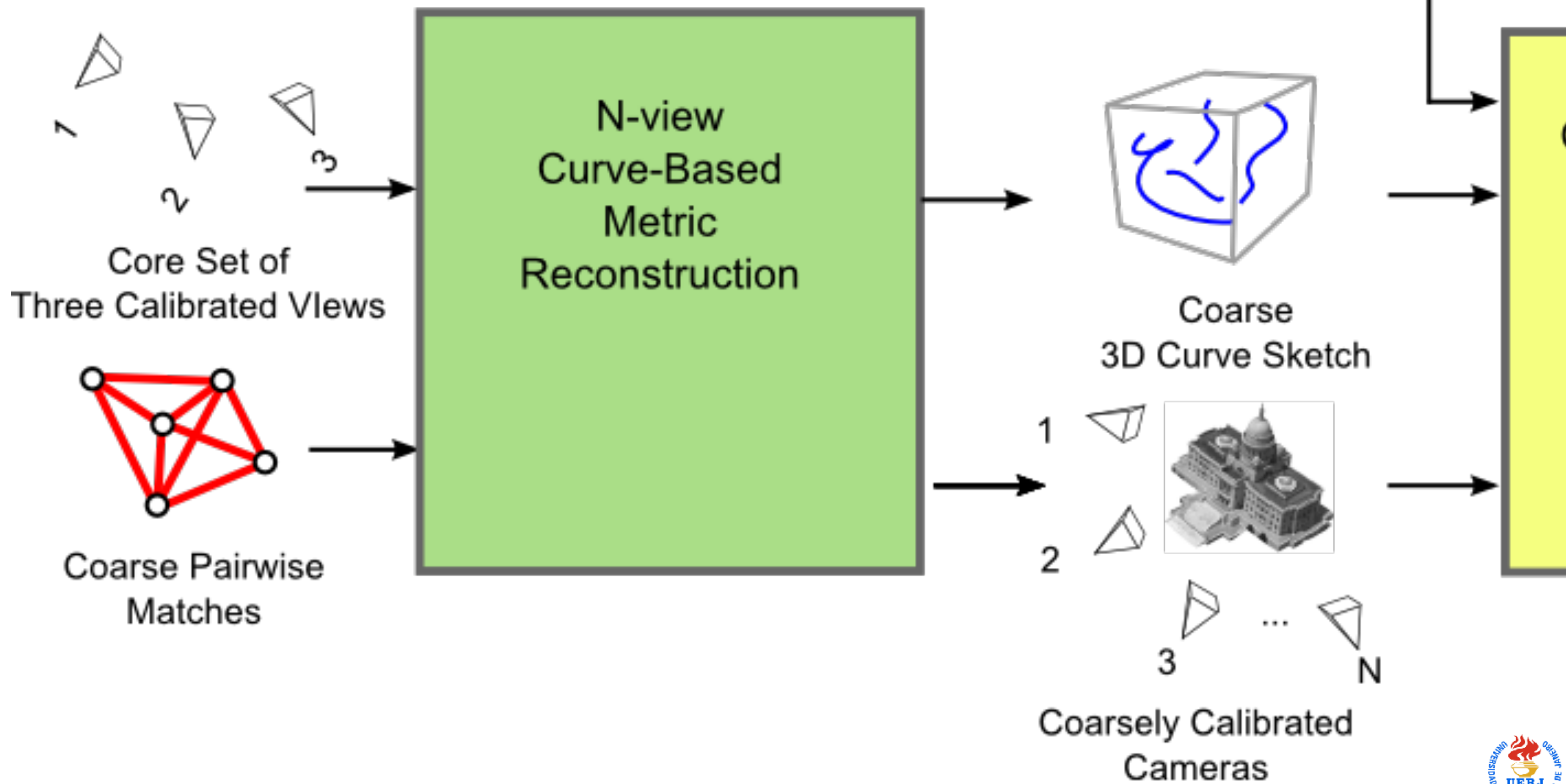


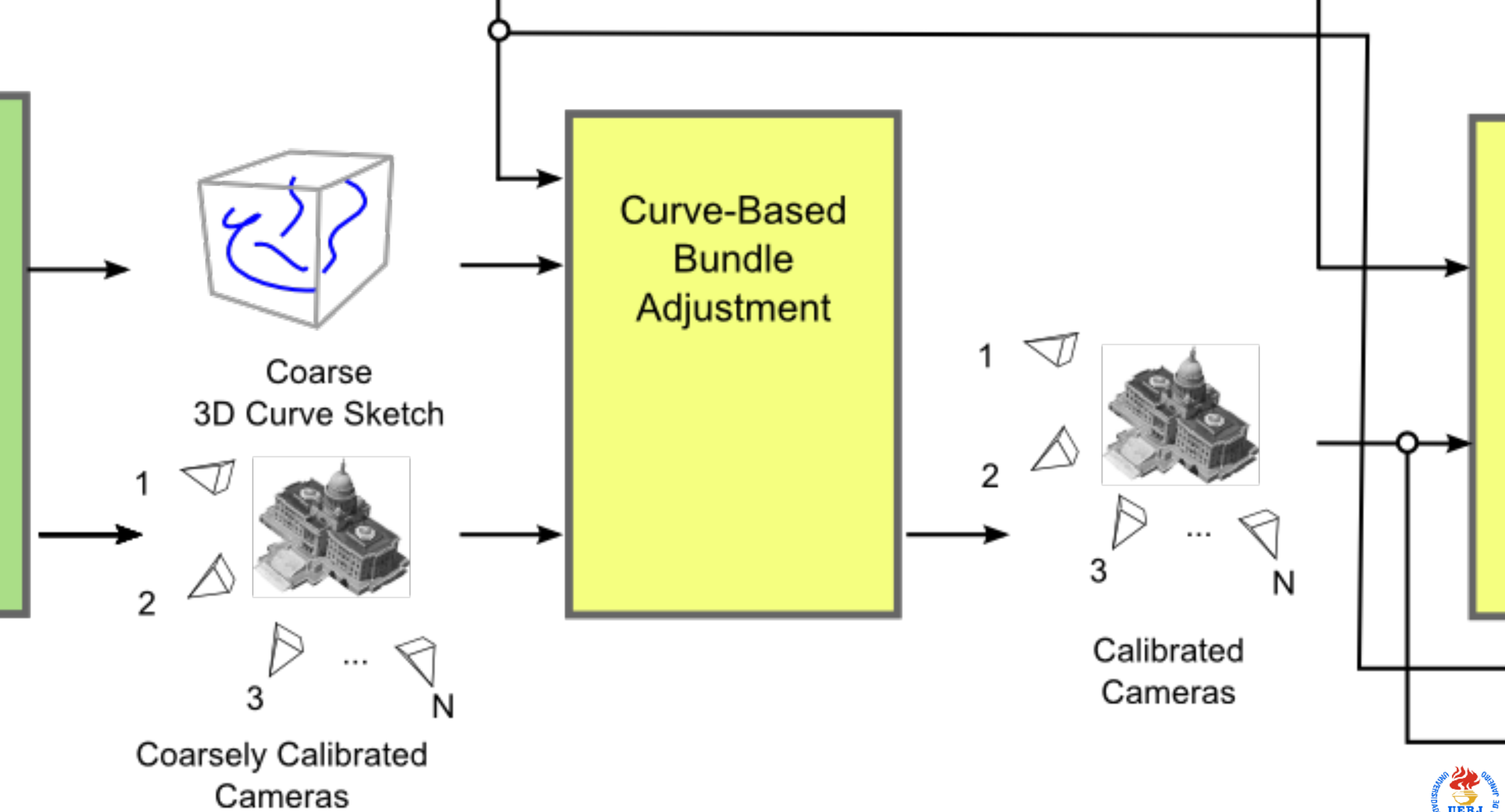


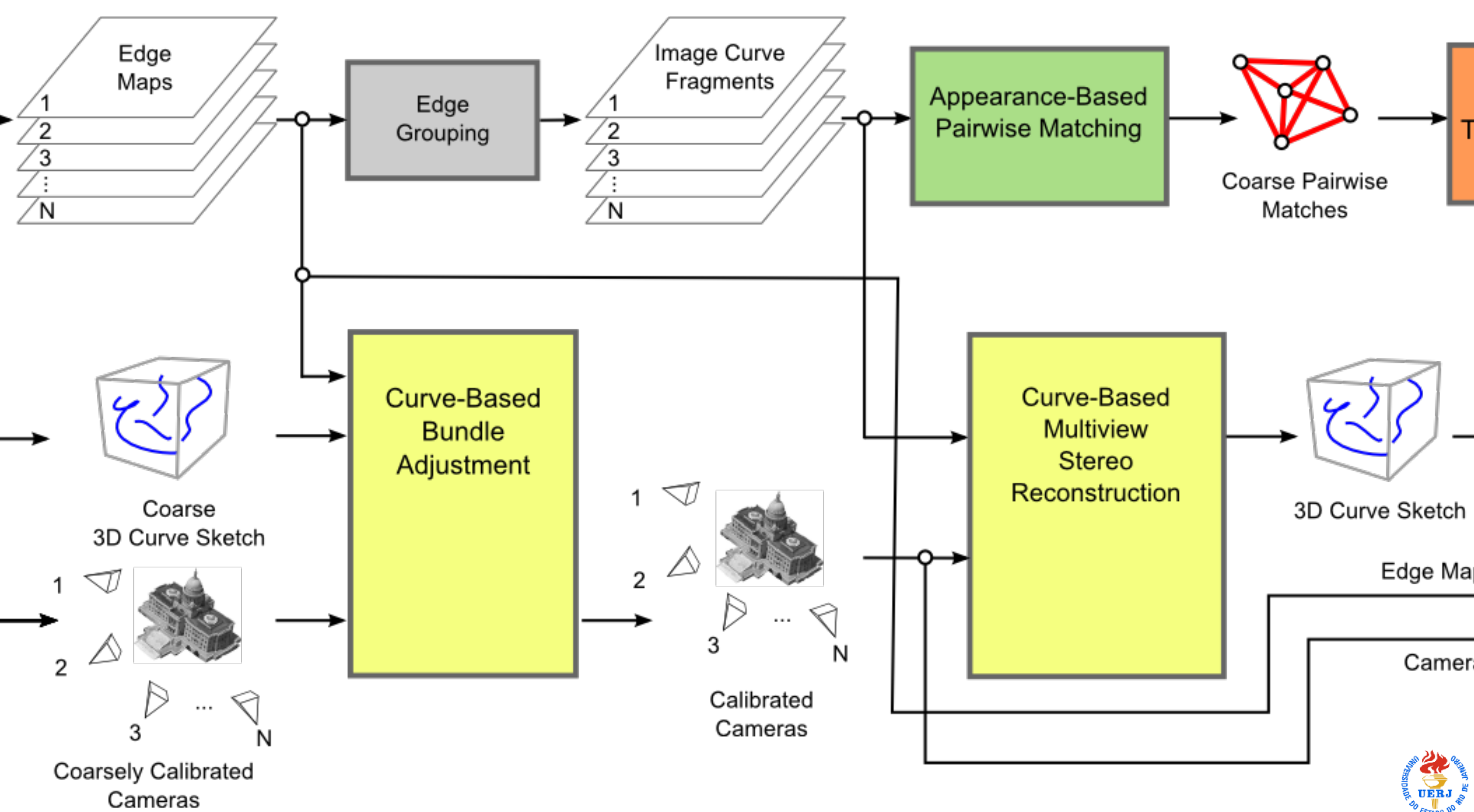
Curve-Based

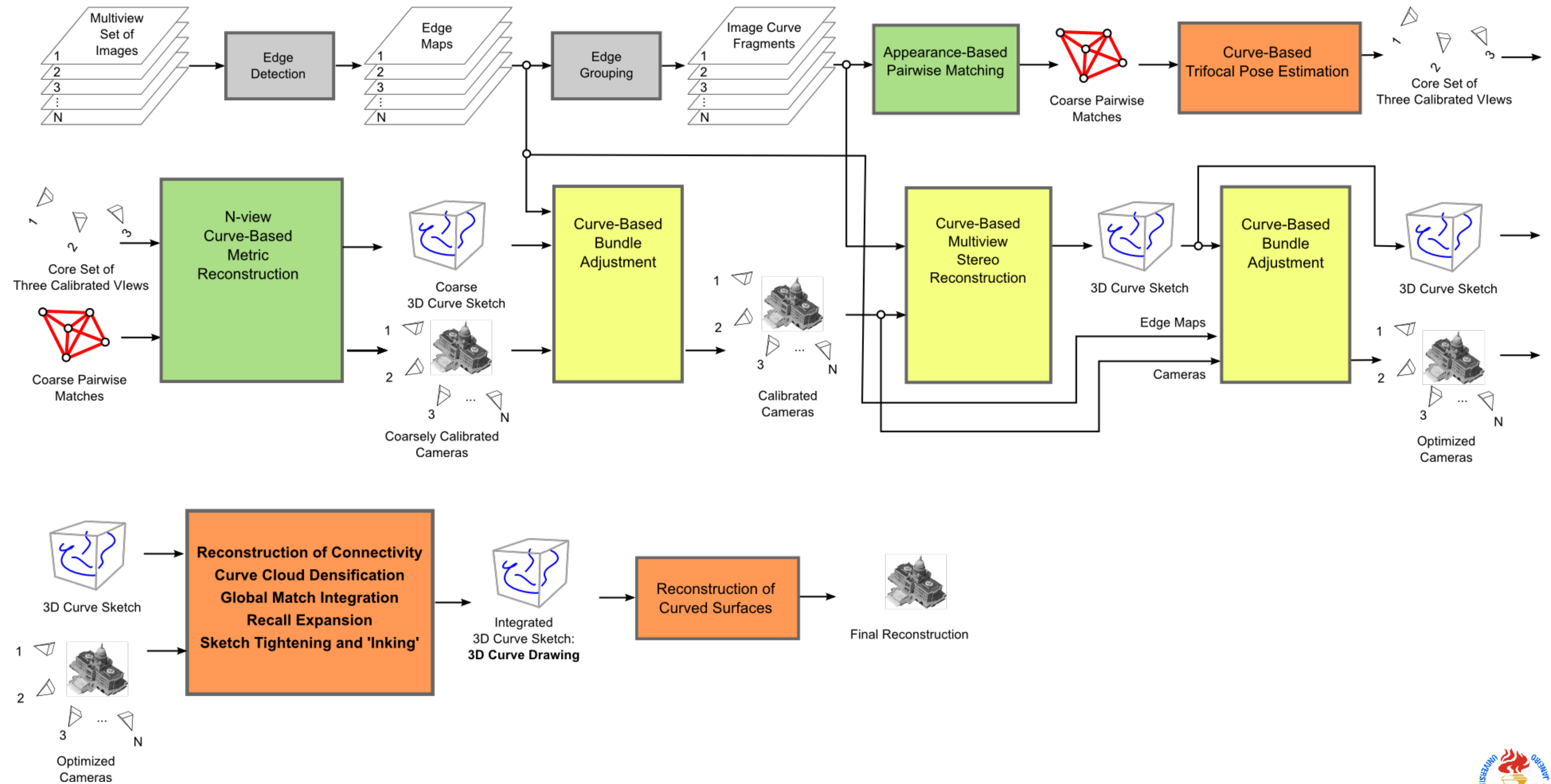




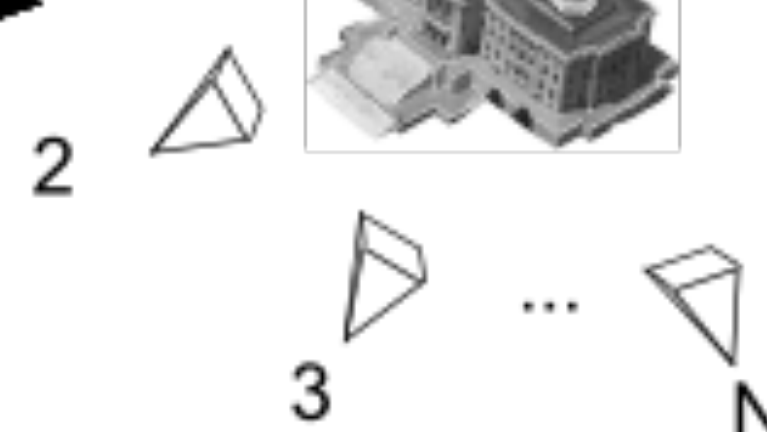




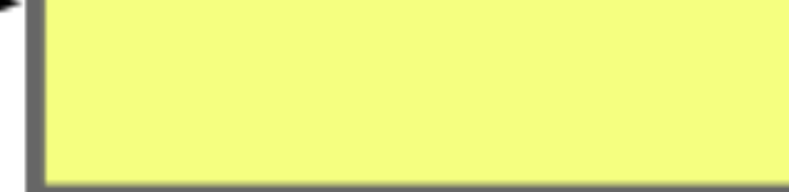




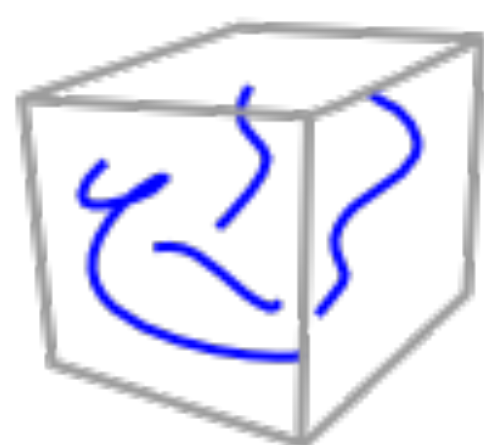
Coarse Pairwise
Matches



Coarsely Calibrated
Cameras



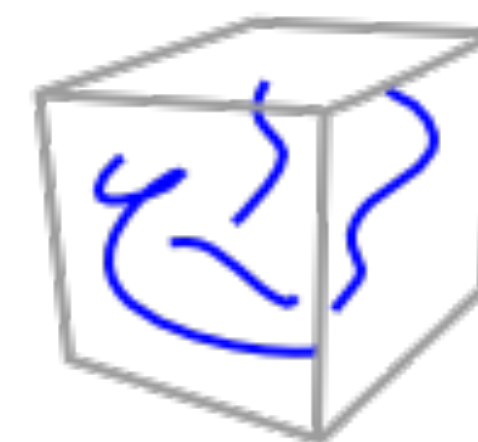
3
Calibra
Camer



3D Curve Sketch



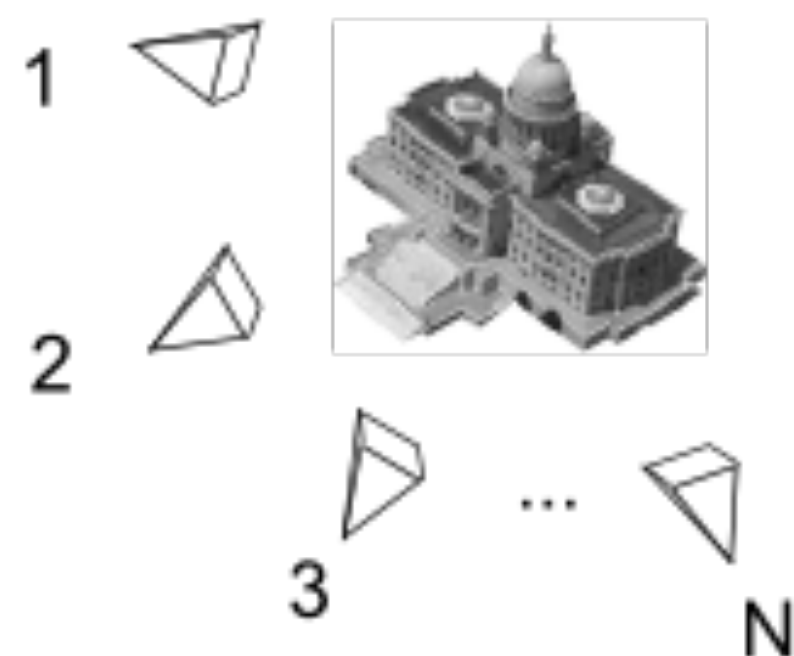
Reconstruction of Connectivity
Curve Cloud Densification
Global Match Integration
Recall Expansion
Sketch Tightening and 'Inking'



Integrated
3D Curve Sketch:
3D Curve Drawing



**Reconstruction of
Curved Surfaces**



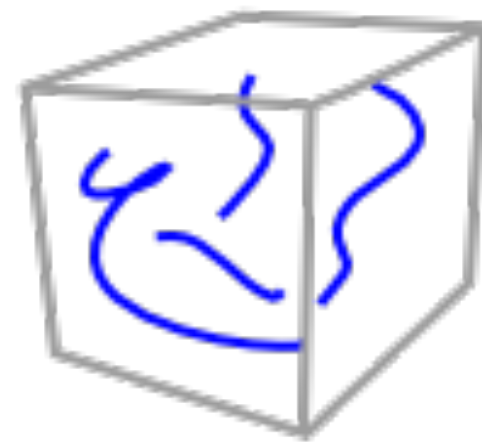
Optimized
Cameras



Coarsely Calibrated
Cameras

Calibrated
Cameras

Construction of Connectivity
Global Cloud Densification
Global Match Integration
Recall Expansion
Tightening and 'Inking'



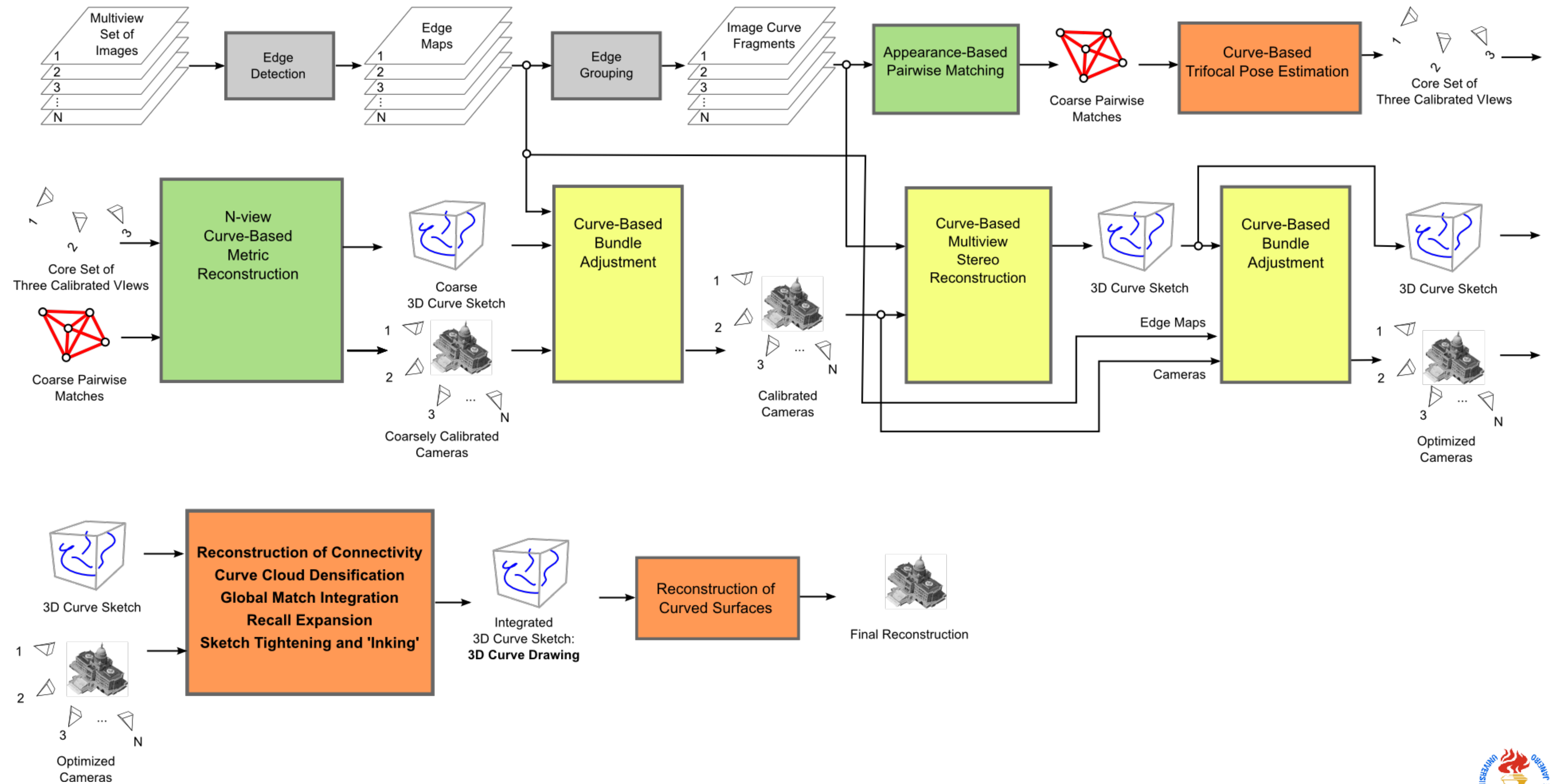
Integrated
3D Curve Sketch:
3D Curve Drawing



Reconstruction of
Curved Surfaces



Final Reconstruction



Goals

1. Resume our code where we left off
2. Grow our 3D system to a complete system
3. Shareable code
 1. Mechanisms to build an end-user system when desired
 2. The end-user system should be lean and "simple to compile"
 3. Experimental code should be kept separate
4. Maintainable code
 1. Modularity: isolate problems in small modules
 2. No human bottlenecks: hierarchy of developers, consistent dedication

Short-term

1. Resume our code where we left off
2. Get lofting up and running
3. Finish and integrate OpenMVG SIFT Orientation pipeline
4. Pure edge-orientation pipeline without linked curves
5. 3D curve sketch: SfM pipeline with curves
6. 3D lofting: SfM pipeline with surfaces
7. Improve all these

Methods

To build and maintain a complete vision system

- **Bare minimum:**
 - Doable, short-term weekly tasks for each person
 - Weekly meetings:
 - progress report: 1 slide per person
 - Should not take over basic research

Methods

To build and maintain a big system

- Define and track medium-term goals/milestones
- Need to break down to small-enough tasks

Task 0.0

Brush-up on professional programming

1. UNIX (Linux and Mac)

LUPE tutorial <http://wiki.nosdigitais.teia.org.br/LUPE>

2. Git

Task 0.1

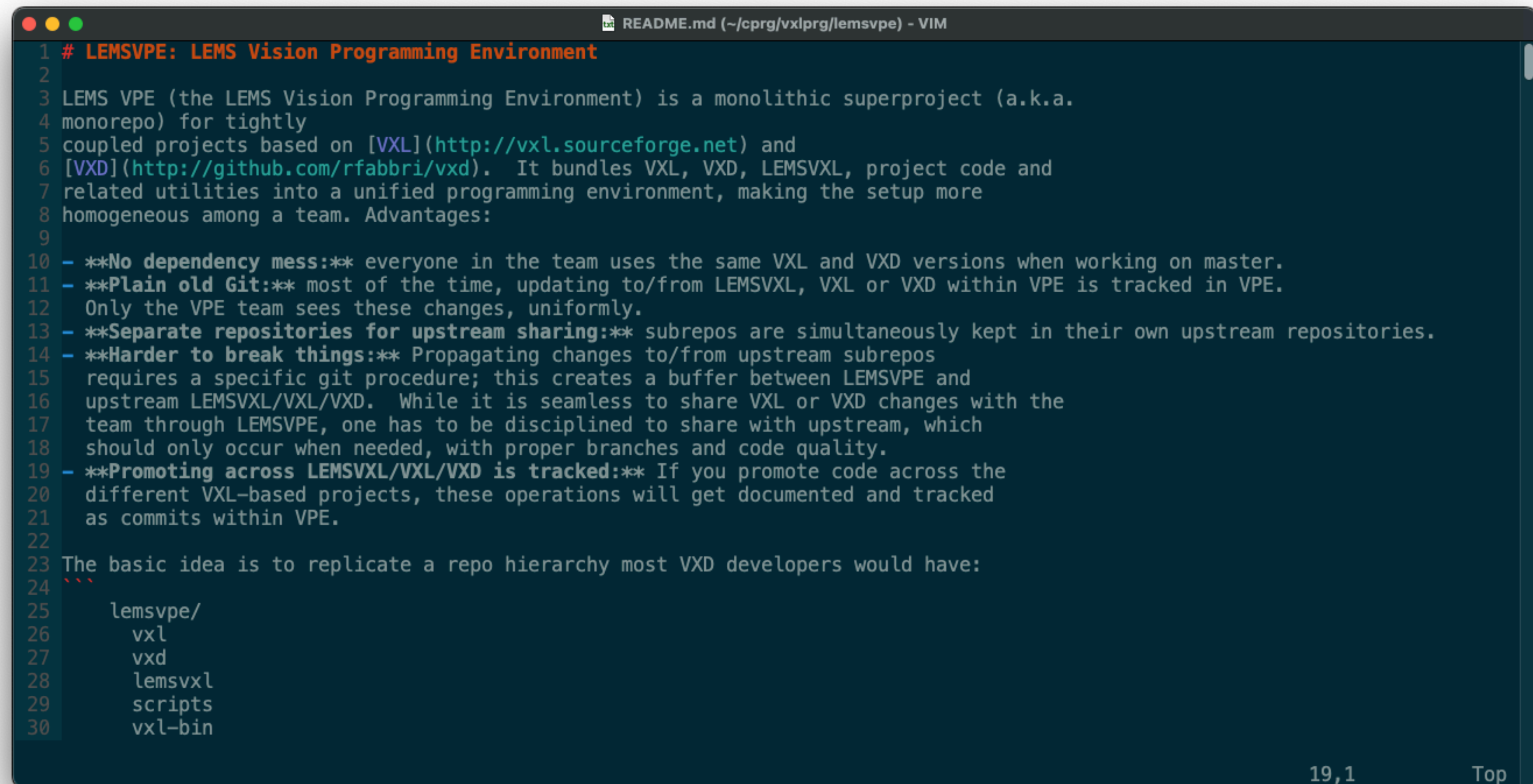
Compile & set up programming environment

1. Download LEMSVPE
 1. (recall where it is) cdlvpe if configured
 2. place it in cprg/vxlprg/lemsvpe
2. Git pull
3. **Configure Shortcuts + Utils for programming** < — important
 1. ~/bin in path
 2. utils
 1. cdlvpe
 2. mymake
 3. cd with cdpath working
 4. sw shell

Documentation

lemsvpe/doc

lemsvpe/README.md



```
1 # LEMSPE: LEMS Vision Programming Environment
2
3 LEMS VPE (the LEMS Vision Programming Environment) is a monolithic superproject (a.k.a.
4 monorepo) for tightly
5 coupled projects based on [VXL](http://vxl.sourceforge.net) and
6 [VXD](http://github.com/rfabbri/vxd). It bundles VXL, VXD, LEMSVXL, project code and
7 related utilities into a unified programming environment, making the setup more
8 homogeneous among a team. Advantages:
9
10 - **No dependency mess:** everyone in the team uses the same VXL and VXD versions when working on master.
11 - **Plain old Git:** most of the time, updating to/from LEMSVXL, VXL or VXD within VPE is tracked in VPE.
12   Only the VPE team sees these changes, uniformly.
13 - **Separate repositories for upstream sharing:** subrepos are simultaneously kept in their own upstream repositories.
14 - **Harder to break things:** Propagating changes to/from upstream subrepos
15   requires a specific git procedure; this creates a buffer between LEMSPE and
16   upstream LEMSVXL/VXL/VXD. While it is seamless to share VXL or VXD changes with the
17   team through LEMSPE, one has to be disciplined to share with upstream, which
18   should only occur when needed, with proper branches and code quality.
19 - **Promoting across LEMSVXL/VXL/VXD is tracked:** If you promote code across the
20   different VXL-based projects, these operations will get documented and tracked
21   as commits within VPE.
22
23 The basic idea is to replicate a repo hierarchy most VXD developers would have:
24 ```
25     lemsvpe/
26         vxl
27         vxd
28         lemsvxl
29         scripts
30         vxl-bin
```

19,1 Top




https://www.youtube.com/watch?v=Akel3_Nv2uI

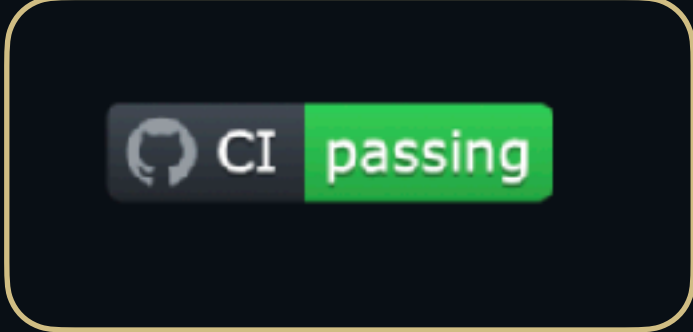
Automated builds

TO-DO

☰ README.md

OpenMVG (open Multiple View Geometry)



License	Documentation	Continuous Integration (Linux/MacOs/Windows)	Build	Code Quality	Chat
license MPL2	doc readthedocs Wiki		local/docker build tutorial	CodeQL passing codefactor A	chat on gitter

Bug tracker

TO-DO

Use github tools

State of the Code

Lofting

code state

Bloft: Blender lofting using Shell+python script

`lemsvpe/lemsvxl/contrib/rfabbri/mw/scripts`

Older code and videos in

`my home/rfabbri/3d-curve-drawing/lofting`

Where is Anil's matlab code for Lofting?

for 22jun23

For next week

minimal tasks

Ric:

- Tell Anil to search for lofting scripts
- Synthesize to everyone how to run curve sketch

**Gabriel (research: trifocal + OpenMVG + MINUS/
Homotopy)**

- Run curve sketch
- Finish UNIX tutorial
- low priority: fix the bug in compiling mw/rfabbri

Carlos (research: occluding countours)

- Lofting, shubao
 - try lofting, try blender, document in lofting.md

Zichang Gao (research:Lofting, in lab til jan'24)

- Same as Carlos
- Loft actual 3D curves from Chieng-Hang

Yilin Zheng (truly multiview curve drawing/grouping)

Chieng-Hang (research trifocal GPU)

- done: Search/send anil's hard drive outcome: no response
 - done: found python script some example data but not matlab
- Document/share document on running 3D drawing

Last week

where we left off

next meeting
always start here

- Overall goal: run, not just compile 3D sketch + lofting**
- Document**

Run 3D Curve Sketch

Milestones

1. lemsvpe/README.md compile steps
2. **lemsvpe/README.md** configure programming enviromnent steps
3. build vxd/bmcscd
4. build lemsvxl/***/contrib/rfabbri/mw/tests
5. build GUI and run sg
6. lemsvpe/doc/3d-curve-drawing.md
7. bmcscd/README.md
8. edge/contour commandline detection
9. mcs
10. visualize in Matlab and Meshlab

```
Initializing stereo driver w nviews = 50
Running stereo driver.
Reading camera: 3x4 matrix, fname = /home/chchien/datasets/amsterdam-house-full-test//00.jpg
bsold_load_cem attempt at loading Reading camera: 3x4 matrix, fname = /home/chchien/datasets/amsterdam-house-full-test//02.jpg
/home/chchien/datasets/amsterdam-house-full-test//00.cemvwith ext.cemv
bsold_load_cem attempt at loading /home/chchien/datasets/amsterdam-house-full-test//02.cemvwith ext.cemv
Reading camera: 3x4 matrix, fname = /home/chchien/datasets/amsterdam-house-full-test//01.jpg
Reading camera: 3x4 matrix, fname = /home/chchien/datasets/amsterdam-house-full-test//03.jpg
Reading camera: 3x4 matrix, fname = /home/chchien/datasets/amsterdam-house-full-test//04.jpg
Reading camera: 3x4 matrix, fname = /home/chchien/datasets/amsterdam-house-full-test//05.jpg
Loaded: /home/chchien/datasets/amsterdam-house-full-test//00.cemv.
Opened cemv file /home/chchien/datasets/amsterdam-house-full-test//00.cemv for reading
Curves: #curves = 133
Started tgt computation.
The data file used to speed-up Euler Spiral computation bgld_eulerspiral_lookup_table.bvl is corrupted or missing.
Loaded: /home/chchien/datasets/amsterdam-house-full-test//02.cemv.
Opened cemv file /home/chchien/datasets/amsterdam-house-full-test//02.cemv for reading
Finished tgt computation.
Curves: #curves = 284
Started tgt computation.
#edgels: 271539
Finished with edge map loading.
[ix,iy]: [0,0]
x,y: 0.0978559, 0.164815
mcs: /home/chchien/BrownU/chchien-lemsvpe/vxd/contrib/brld/bsegd/sdetd/algosdetd_convert_edgemap.cxx:100: bool sdetd_convert_edgemap_to_image(const sdet_edgemap&, vil_image_view<unsigned char>&, const vgl_box_2d<int>&): Assertion '!edgemap.edge_cells(pt_y,pt_x).empty()' failed.
Aborted (core dumped)
```

For next week

minimal tasks

Ric:

- slides to google doc
- Synthesize/Document to everyone how to run curve sketch, drawing

Gabriel (research: trifocal + OpenMVG + MINUS/Homotopy)

- Fix bug curve sketch
- Setup the programming environment better (run sg anywhere)
- Run curve drawing (matlab)

Carlos (research: occluding contours)

- Lofting, shubao
 - try older blenders (2.6, 2.8)
 - learn manual lofting BSurfaces with Blender 3.6

Zichang Gao (research:Lofting, in lab til jan'24)

- 1) Show manual lofting two curves drawn by hand in blender
- 2) Loft actual 3D curves from Yilin manually (BSurfaces)
- 3) Point cloud -> surface (not the same as Lofting tool)
- 4) Curves and points —> surface (future)

Yilin Zheng (truly multiview curve drawing/grouping)

- Highlevel overview of where research is

Chieng-Hang (research trifocal GPU)

- Search further with Ric
- Focus on finding Anil's part of lofting

Latest version is on Google Docs:
https://docs.google.com/presentation/d/1v6S7F-9oL92CpOHQlcLdq8eL13A_PrQiXaCyld27LVs