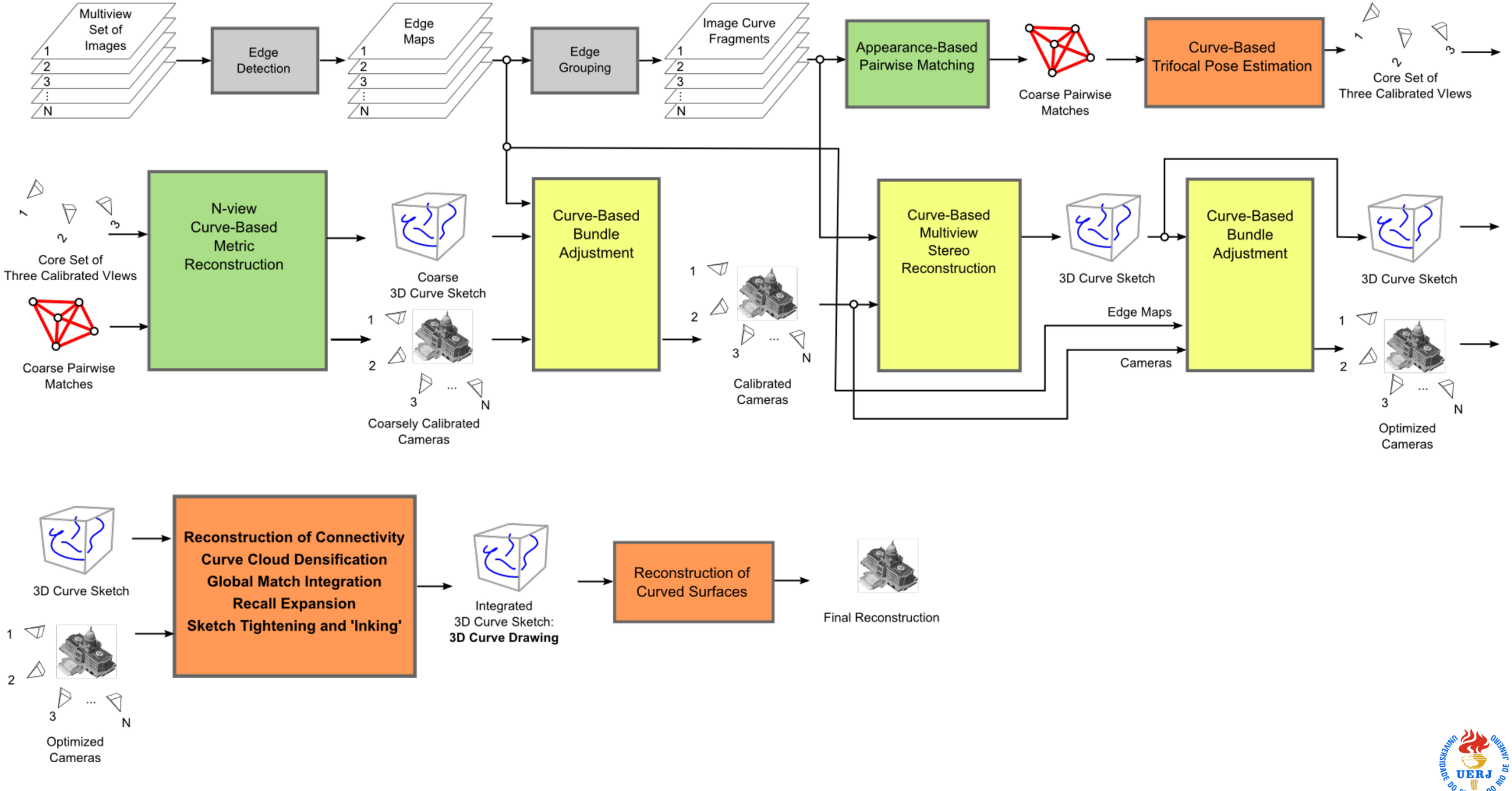
curve-based

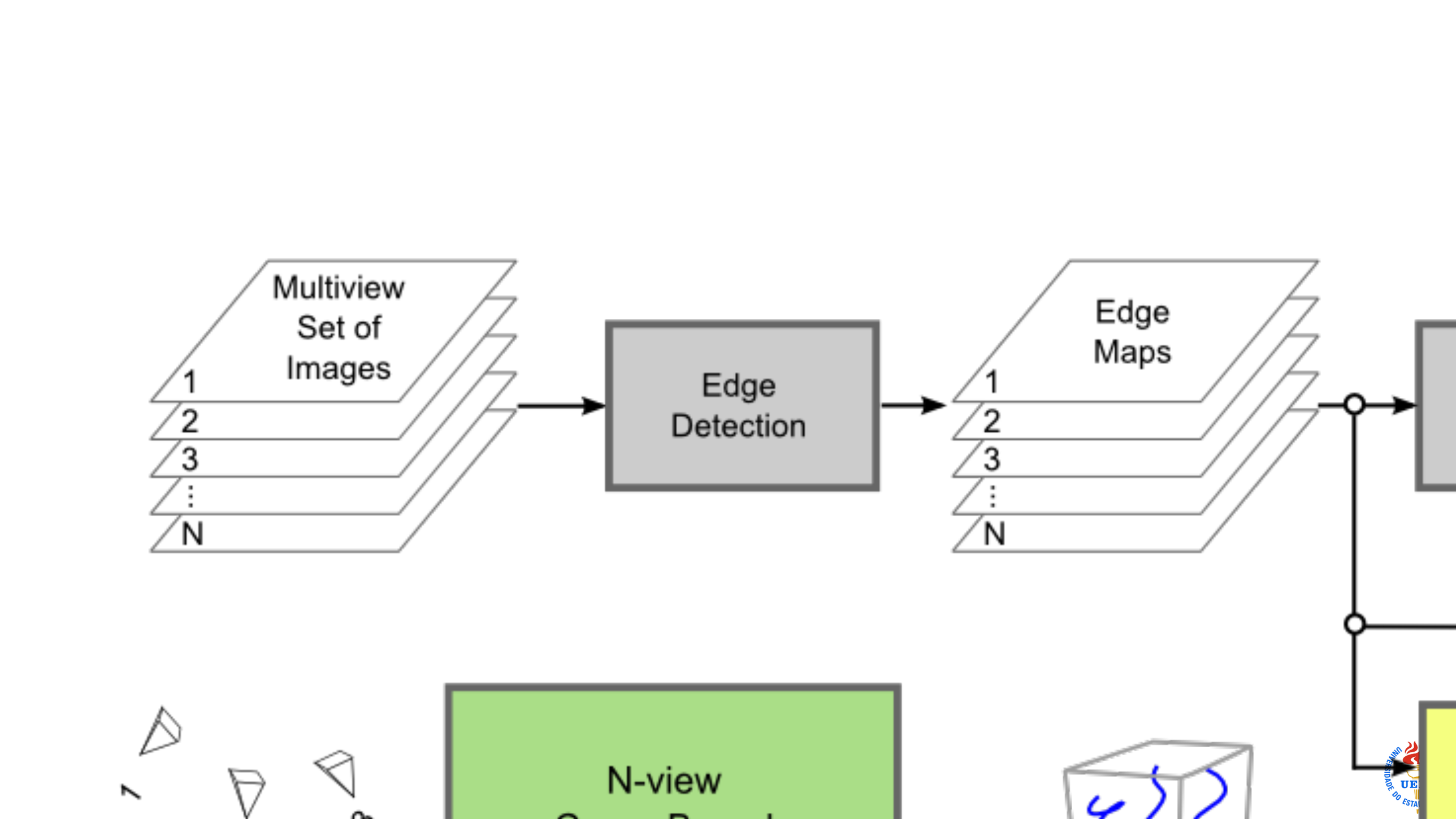
Total 3D vision system

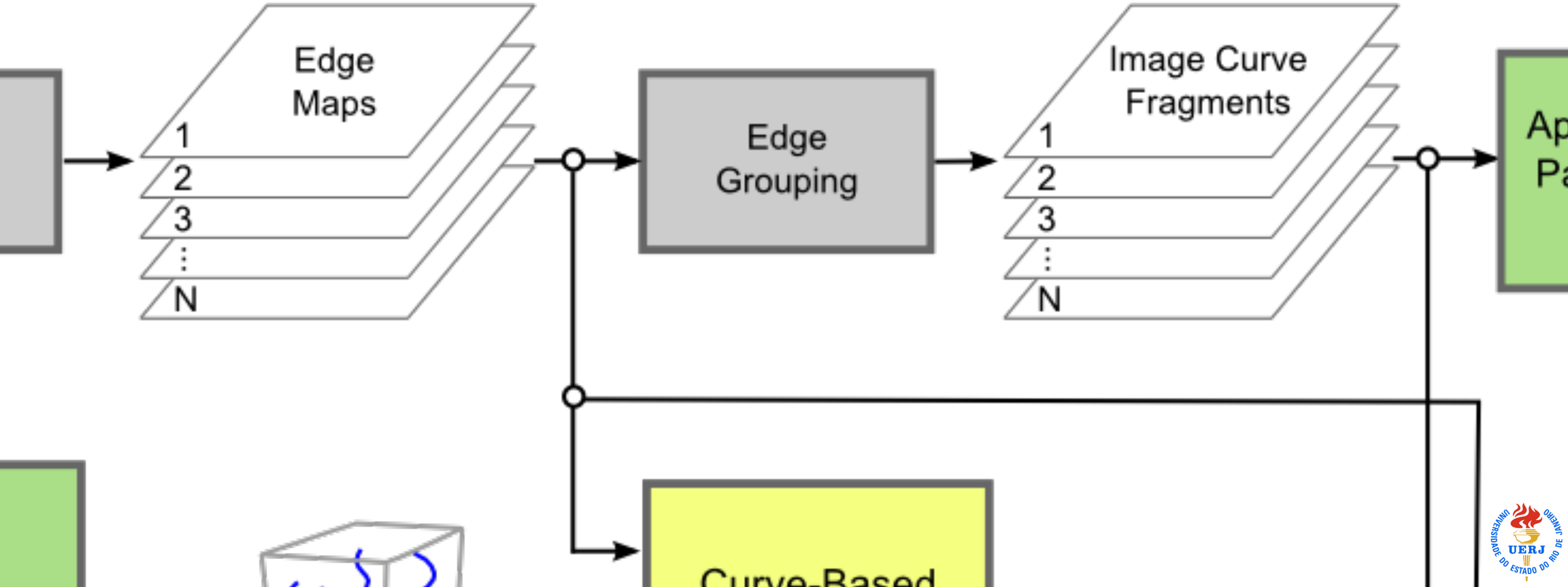
Code effort

prof. Ricardo Fabbri, Rio de Janeiro State University

15jul23


Multiview Set of Images
1
2
3
N
Edge Detection
Edge Maps
Edge Grouping
Image Curve Fragments
Appearance-Based Pairwise Matching
Coarse Pairwise Matches
Curve-Based Trifocal Pose Estimation
Core Set of Three Calibrated VIews
Core Set of Three Calibrated VIews
Coarse Pairwise Matches
N-view Curve-Based Metric Reconstruction
Coarse 3D Curve Sketch
Coarsely Calibrated Cameras
Curve-Based Bundle Adjustment
Calibrated Cameras
Curve-Based Multiview Stereo Reconstruction
3D Curve Sketch
Edge Maps
Cameras
Curve-Based Bundle Adjustment
3D Curve Sketch
Optimized Cameras
3D Curve Sketch
Optimized Cameras
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
Final Reconstruction


Multiview
Set of
Images
1
2
3
N
Edge
Detection
Edge
Maps
1
2
3
N
N-view


Edge
Maps
1
2
3
N
Edge
Grouping
Image Curve
Fragments
1
2
3
N

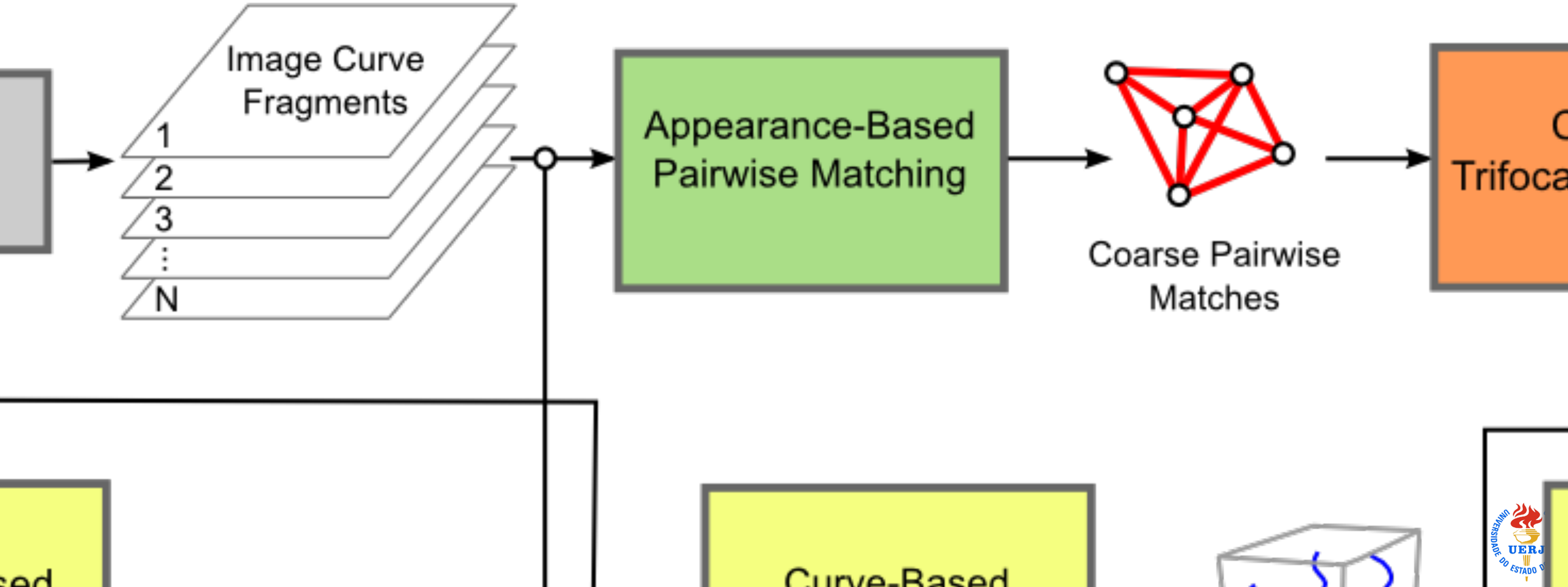


Image Curve Fragments
1
2
3
N
Appearance-Based Pairwise Matching
Coarse Pairwise Matches
Curve-Based

e-Based
Matching

Coarse Pairwise
Matches

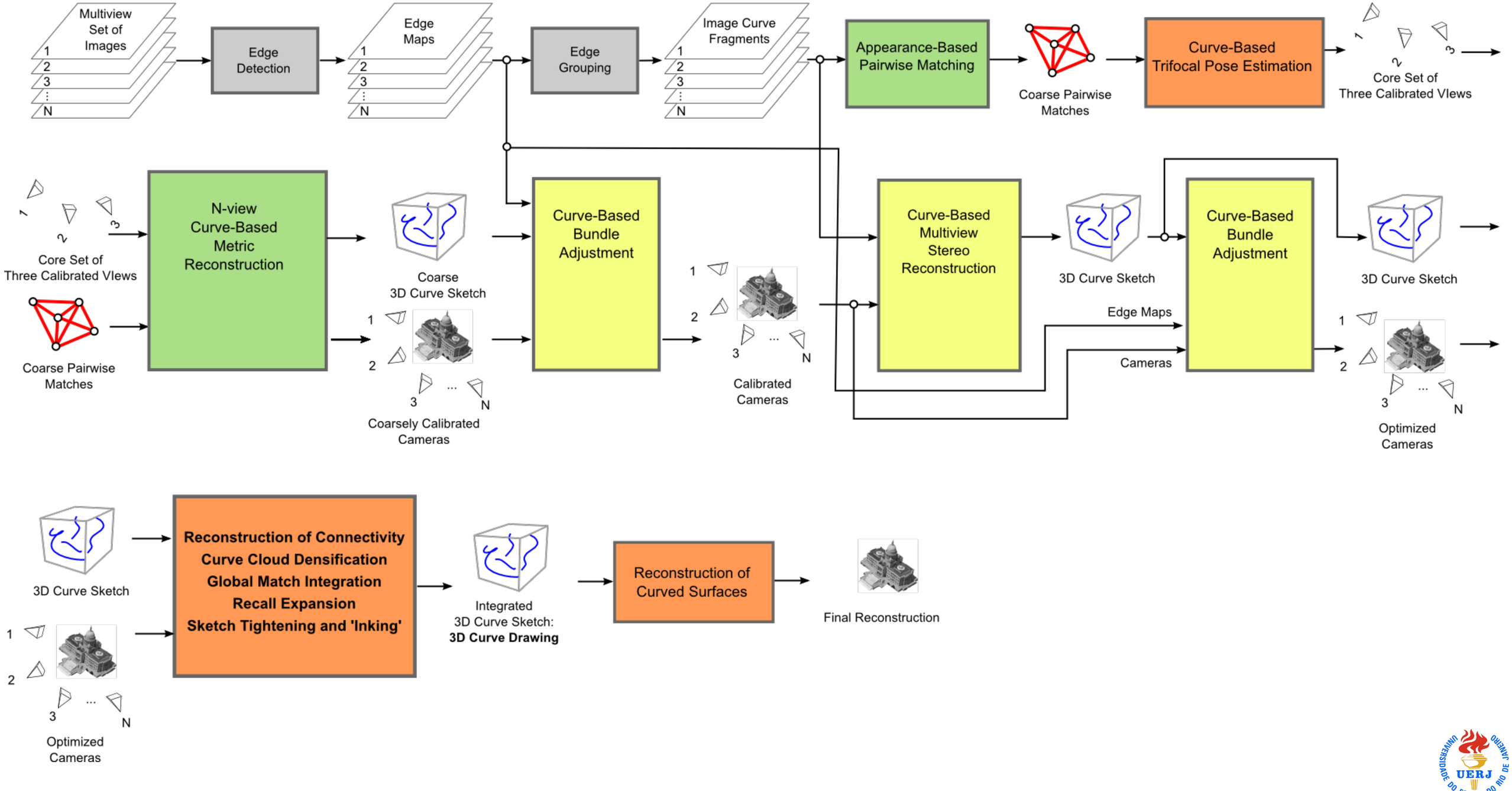
Curve-Based
Trifocal Pose Estimation

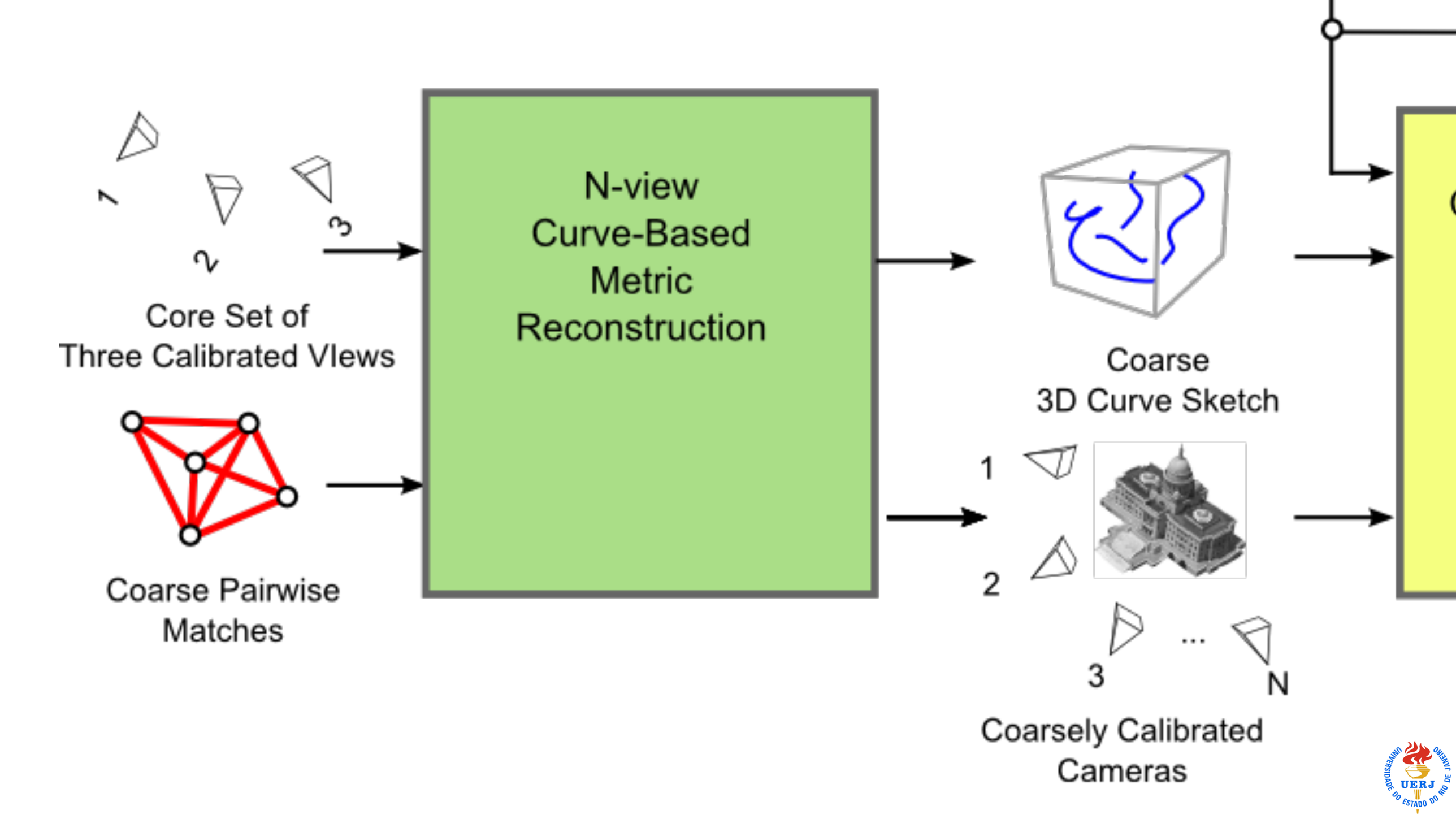
1
2
3

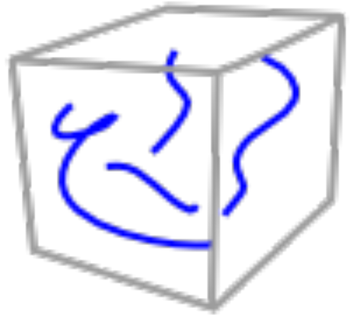
Core Set of
Three Calibrated Views

rve-Based

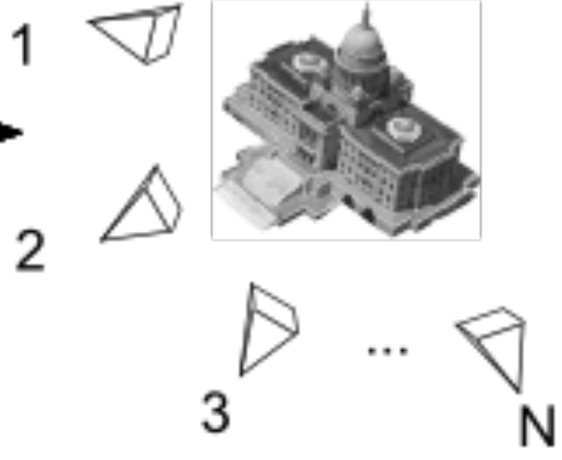
Curve-Based


Multiview Set of Images
1
2
3
N
Edge Detection
Edge Maps
Edge Grouping
Image Curve Fragments
Appearance-Based Pairwise Matching
Coarse Pairwise Matches
Curve-Based Trifocal Pose Estimation
Core Set of Three Calibrated VIews
Core Set of Three Calibrated VIews
Coarse Pairwise Matches
N-view Curve-Based Metric Reconstruction
Coarse 3D Curve Sketch
Coarsely Calibrated Cameras
Curve-Based Bundle Adjustment
Calibrated Cameras
Curve-Based Multiview Stereo Reconstruction
3D Curve Sketch
Edge Maps
Cameras
Curve-Based Bundle Adjustment
3D Curve Sketch
Optimized Cameras
3D Curve Sketch
Optimized Cameras
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
Final Reconstruction


1
2
3
Core Set of
Three Calibrated Vlews
Coarse Pairwise
Matches
N-view
Curve-Based
Metric
Reconstruction
Coarse
3D Curve Sketch
1
2
3
...
N
Coarsely Calibrated
Cameras

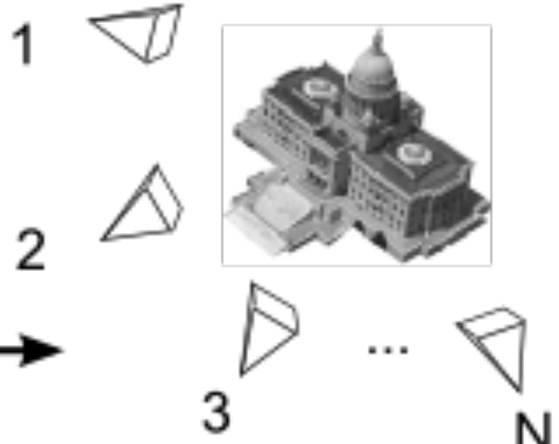


Coarse
3D Curve Sketch

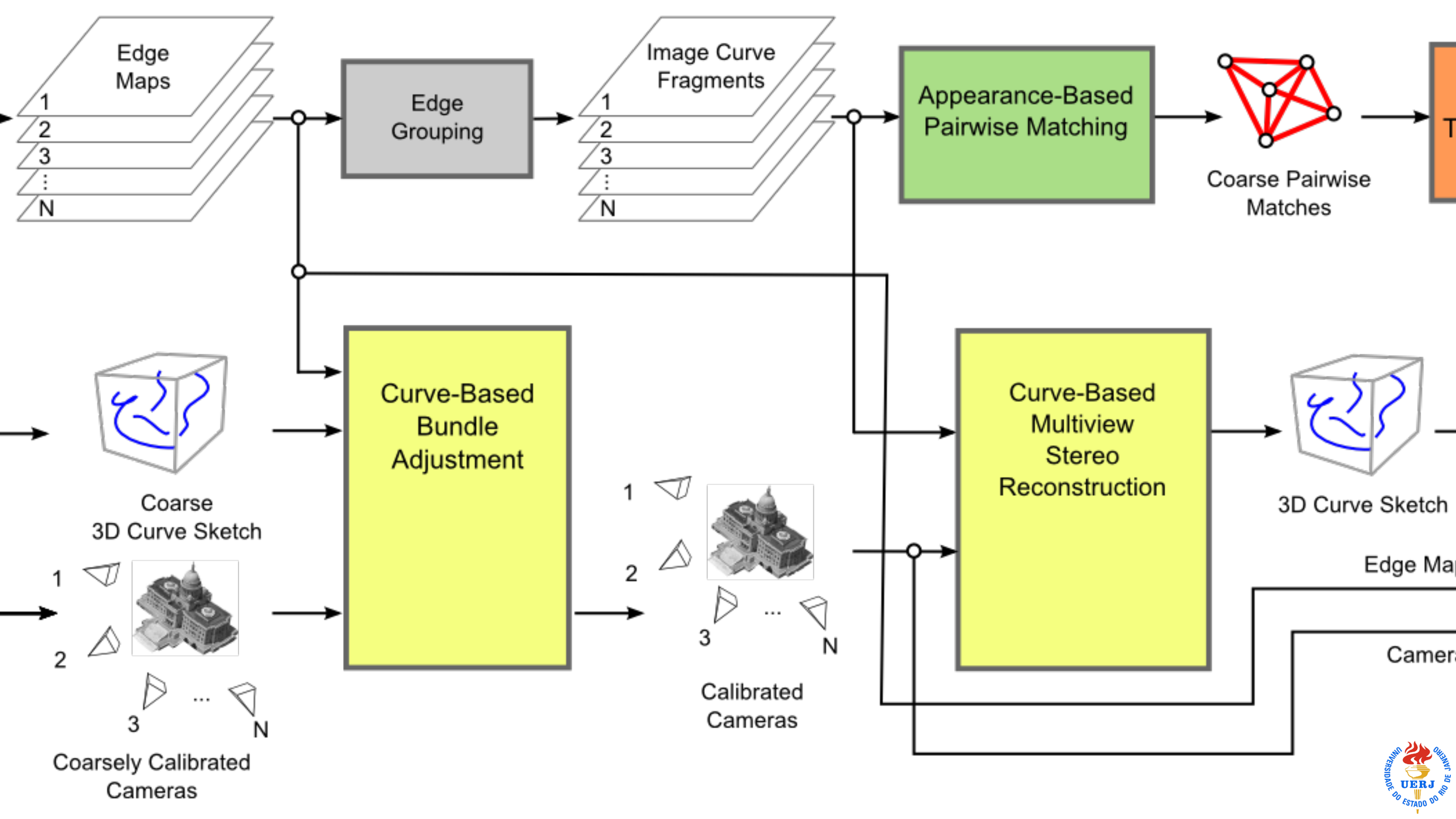


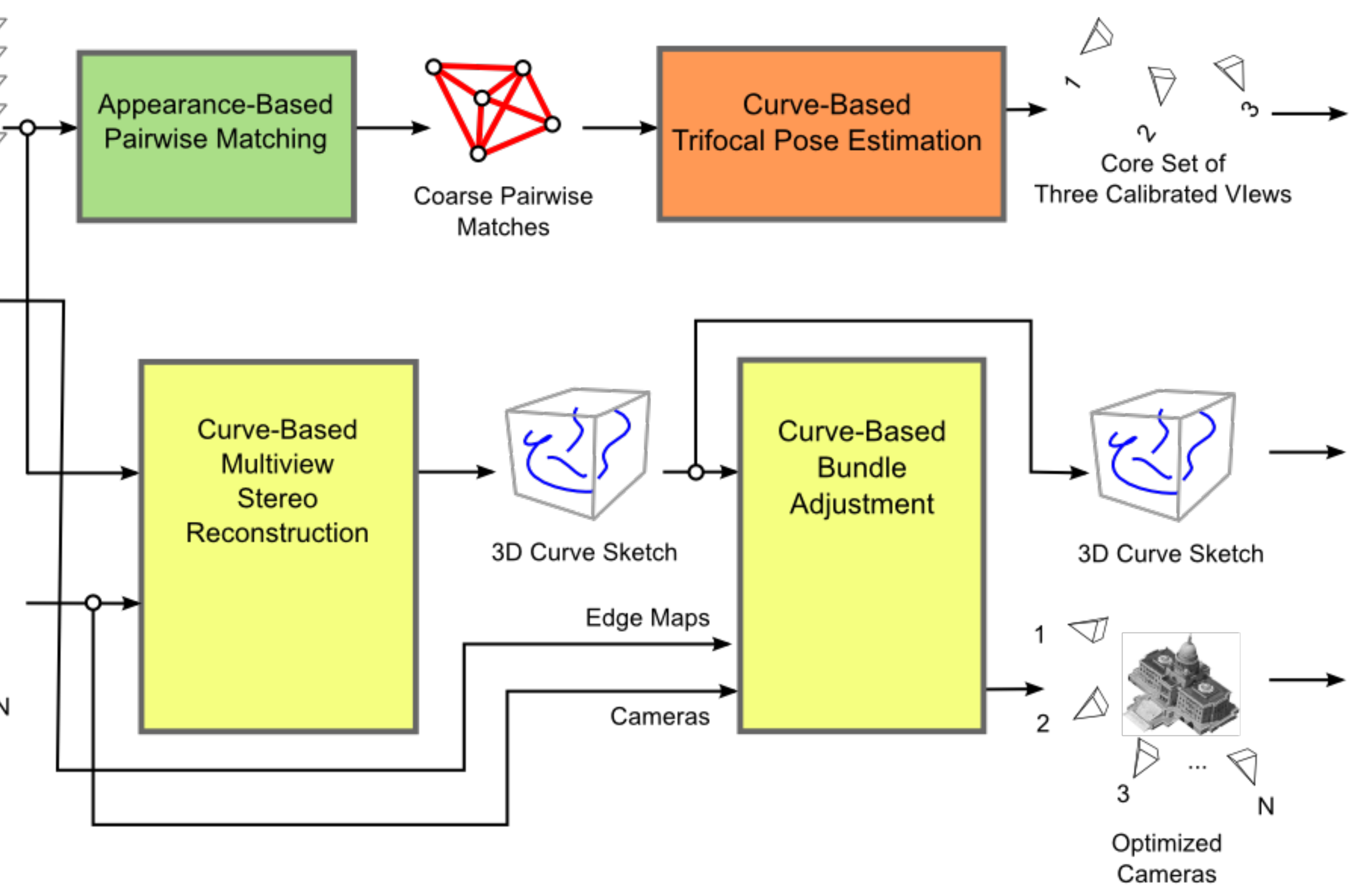
Coarsely Calibrated
Cameras

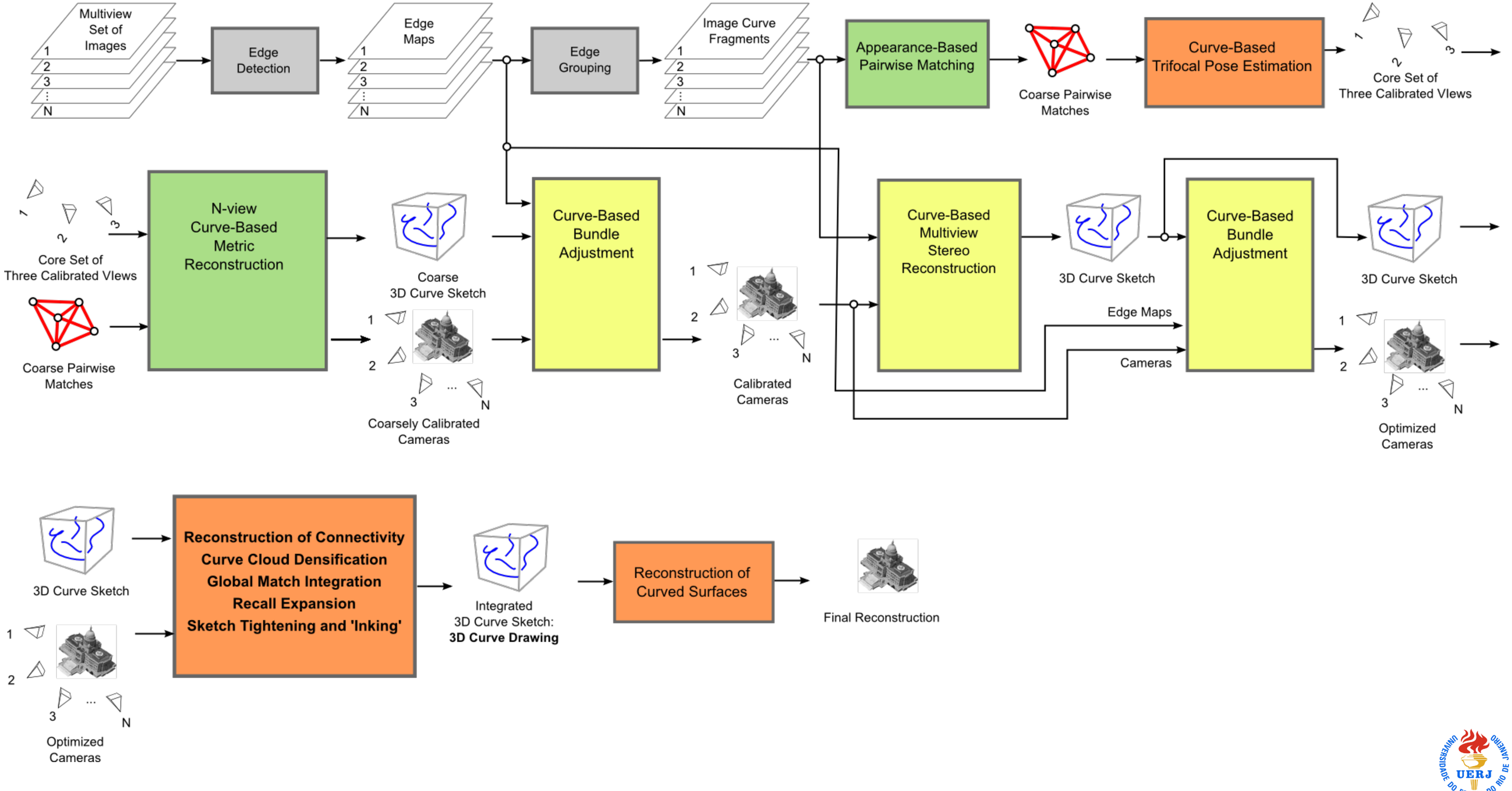
Curve-Based
Bundle
Adjustment



Calibrated
Cameras


Edge Maps
1
2
3
N
Edge Grouping
Image Curve Fragments
1
2
3
N
Appearance-Based Pairwise Matching
Coarse Pairwise Matches
Coarse 3D Curve Sketch
Curve-Based Bundle Adjustment
Curve-Based Multiview Stereo Reconstruction
3D Curve Sketch
1
2
3
...
N
Coarsely Calibrated Cameras
Calibrated Cameras
Edge Ma
Camer


Appearance-Based
Pairwise Matching
Coarse Pairwise
Matches
Curve-Based
Trifocal Pose Estimation
1
2
3
Core Set of
Three Calibrated Views
Curve-Based
Multiview
Stereo
Reconstruction
3D Curve Sketch
Curve-Based
Bundle
Adjustment
3D Curve Sketch
Edge Maps
Cameras
1
2
3
...
N
Optimized
Cameras


Multiview Set of Images
1
2
3
N
Edge Detection
Edge Maps
Edge Grouping
Image Curve Fragments
Appearance-Based Pairwise Matching
Coarse Pairwise Matches
Curve-Based Trifocal Pose Estimation
Core Set of Three Calibrated VIews
Core Set of Three Calibrated VIews
Coarse Pairwise Matches
N-view Curve-Based Metric Reconstruction
Coarse 3D Curve Sketch
Coarsely Calibrated Cameras
Curve-Based Bundle Adjustment
Calibrated Cameras
Curve-Based Multiview Stereo Reconstruction
3D Curve Sketch
Edge Maps
Cameras
Curve-Based Bundle Adjustment
3D Curve Sketch
Optimized Cameras
3D Curve Sketch
Optimized Cameras
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
Final Reconstruction

Coarse Pairwise Matches

2

3

...

N

Coarsely Calibrated Cameras

3

Calibra
Camer

3D Curve Sketch

1

2

3

...

N

Optimized Cameras

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

2

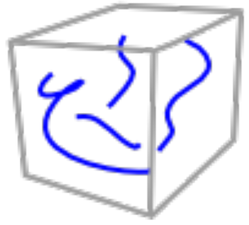
3 ... N

Coarsely Calibrated Cameras

N

Calibrated Cameras

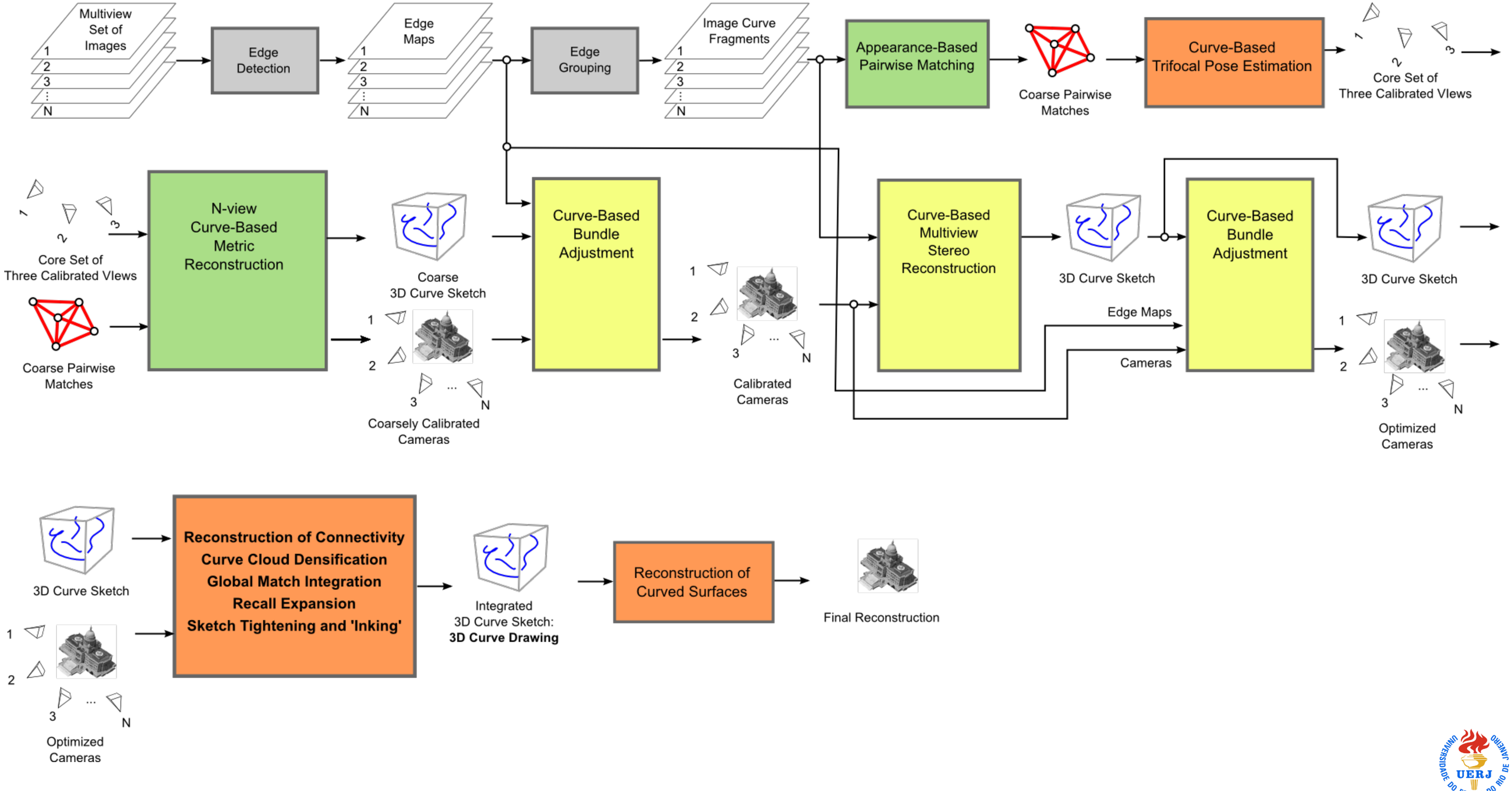
struction of Connectivity
ve Cloud Densification
bal Match Integration
Recall Expansion
n Tightening and 'Inking'



Integrated
3D Curve Sketch:
3D Curve Drawing

Reconstruction of Curved Surfaces

Final Reconstruction


Multiview Set of Images
1
2
3
N
Edge Detection
Edge Maps
Edge Grouping
Image Curve Fragments
Appearance-Based Pairwise Matching
Coarse Pairwise Matches
Curve-Based Trifocal Pose Estimation
Core Set of Three Calibrated VIews
Core Set of Three Calibrated VIews
Coarse Pairwise Matches
N-view Curve-Based Metric Reconstruction
Coarse 3D Curve Sketch
Coarsely Calibrated Cameras
Curve-Based Bundle Adjustment
Calibrated Cameras
Curve-Based Multiview Stereo Reconstruction
3D Curve Sketch
Edge Maps
Cameras
Curve-Based Bundle Adjustment
3D Curve Sketch
Optimized Cameras
3D Curve Sketch
Optimized Cameras
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
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

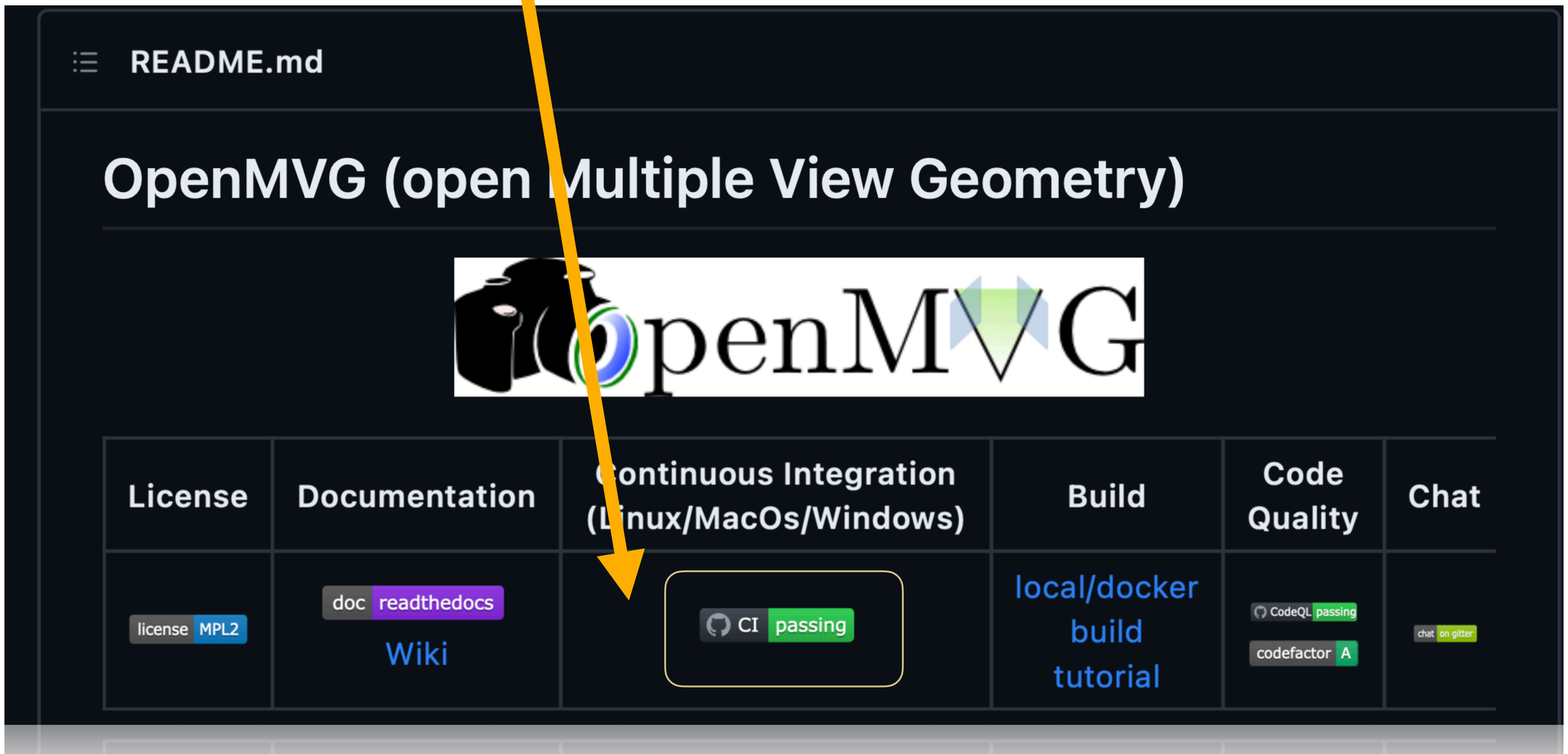
README.md (~/cprg/vxlprg/lemsvpe) - VIM

```
# LEMSVPE: LEMS Vision Programming Environment

LEMS VPE (the LEMS Vision Programming Environment) is a monolithic superproject (a.k.a.
monorepo) for tightly
coupled projects based on [VXL](http://vxl.sourceforge.net) and
[VXD](http://github.com/rfabbri/vxd).  It bundles VXL, VXD, LEMSVXL, project code and
related utilities into a unified programming environment, making the setup more
homogeneous among a team. Advantages:

- **No dependency mess:** everyone in the team uses the same VXL and VXD versions when working on master.
- **Plain old Git:** most of the time, updating to/from LEMSVXL, VXL or VXD within VPE is tracked in VPE.
  Only the VPE team sees these changes, uniformly.
- **Separate repositories for upstream sharing:** subrepos are simultaneously kept in their own upstream repositories.
- **Harder to break things:** Propagating changes to/from upstream subrepos
  requires a specific git procedure; this creates a buffer between LEMSVPE and
  upstream LEMSVXL/VXL/VXD.  While it is seamless to share VXL or VXD changes with the
  team through LEMSVPE, one has to be disciplined to share with upstream, which
  should only occur when needed, with proper branches and code quality.
- **Promoting across LEMSVXL/VXL/VXD is tracked:** If you promote code across the
  different VXL-based projects, these operations will get documented and tracked
  as commits within VPE.

The basic idea is to replicate a repo hierarchy most VXD developers would have:
```
    lemsvpe/
      vxl
      vxd
      lemsvxl
      scripts
      vxl-bin
```
                                                                    19,1          Top
```

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

Automated builds

TO-DO



Bug tracker
TO-DO

Use github tools

State of the Code

15jun23

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?

15jun23+7

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)

Yilin Zheng (multiview curve drawing/grouping)

Chieng-Hang (research trifocal GPU)

- Search/send anil's hard drive
- Document/share document on running 3D drawing

Last week

where we left off

- Overall goal: run, not just compile
- Document

next meeting

always start here