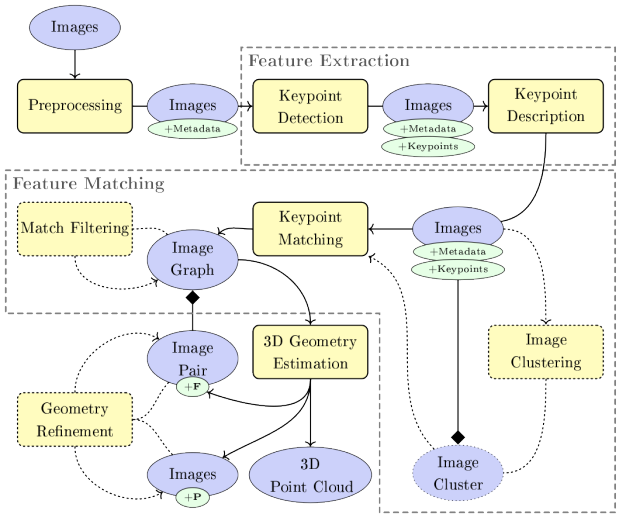
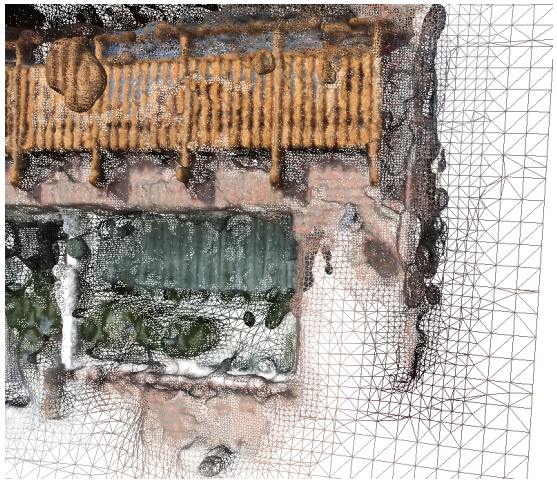
Master's thesis

A Modular Framework for Image-based 3D Reconstruction



Carsten Brandt

May 18, 2017



3D Reconstruction

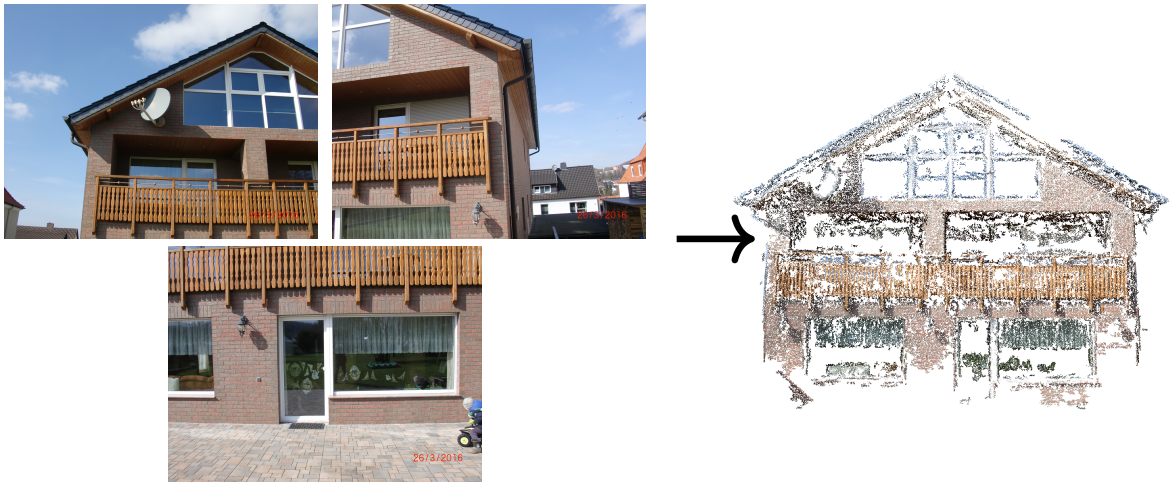
3D Reconstruction

Structure from Motion (SfM)

3D Reconstruction

Structure from Motion (SfM)

3D Reconstruction



Structure from Motion (SfM)

3D Reconstruction

Structure from Motion (SfM)

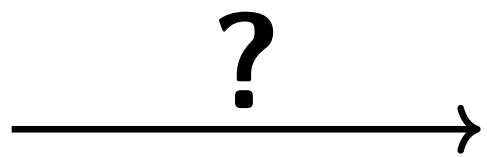
Table of Contents

Table of Contents

Problem Definition

Problem Definition

Problem Definition



Problem Definition

?

Keypoint Detection

Image Matching

Problem Definition

?

Keypoint Detection

Image Matching

Geometry Estimation

Problem Definition

?

Keypoint Detection

Image Matching

Geometry Estimation

Triangulation

Problem Definition

?

Keypoint Detection

Image Matching

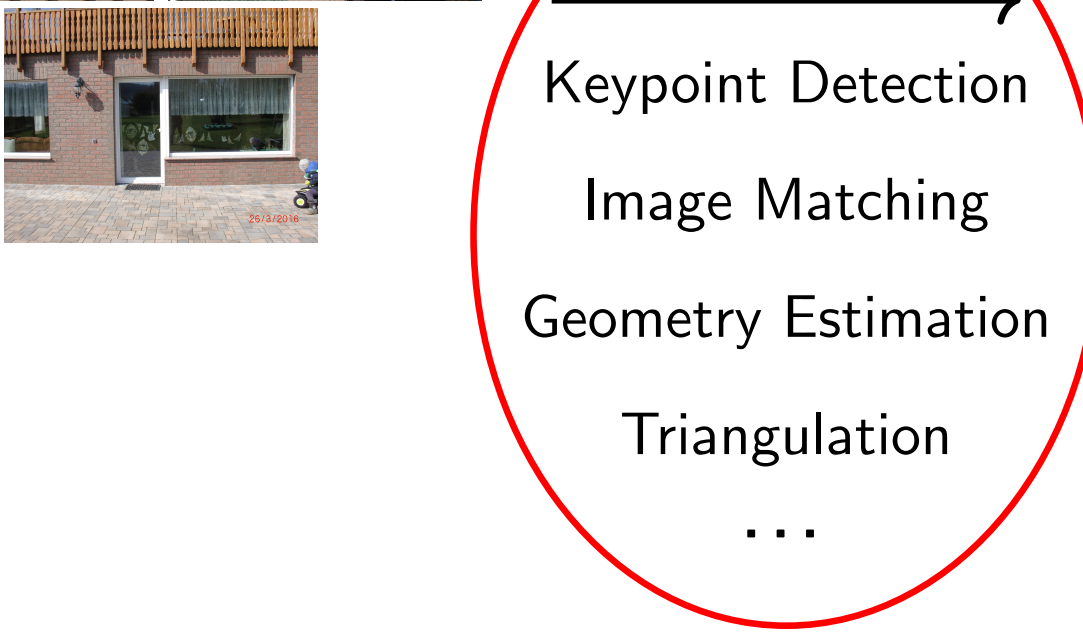
Geometry Estimation

Triangulation

...

Problem Definition

Processing Chain



Existing implementations

Existing implementations:

Existing implementations

Existing implementations:

- single program for everything

Existing implementations

Existing implementations:

- single program for everything
- not flexible for different use case

Existing implementations

Existing implementations:

- single program for everything
- not flexible for different use case
- hard to adjust/improve

Existing implementations

Existing implementations:

- single program for everything
- not flexible for different use case
- hard to adjust/improve

Existing implementations

Existing implementations:

- single program for everything
- not flexible for different use case
- hard to adjust/improve

Framework goals:

Existing implementations

Existing implementations:

- single program for everything
- not flexible for different use case
- hard to adjust/improve

Framework goals:

- flexible processing chain creation

Existing implementations

Existing implementations:

- single program for everything
- not flexible for different use case
- hard to adjust/improve

Framework goals:

- flexible processing chain creation
- applicable to large reconstruction problems

Existing implementations

Existing implementations:

- single program for everything
- not flexible for different use case
- hard to adjust/improve

Framework goals:

- flexible processing chain creation
- applicable to large reconstruction problems
- visualisation options

Existing implementations

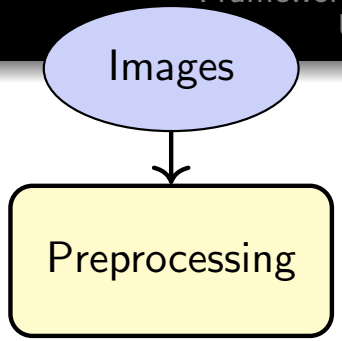
Existing implementations:

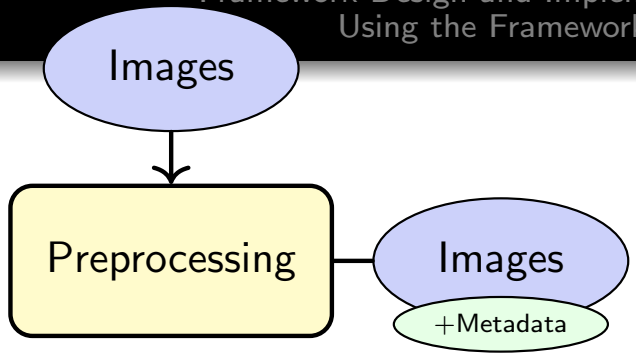
- single program for everything
- not flexible for different use case
- hard to adjust/improve

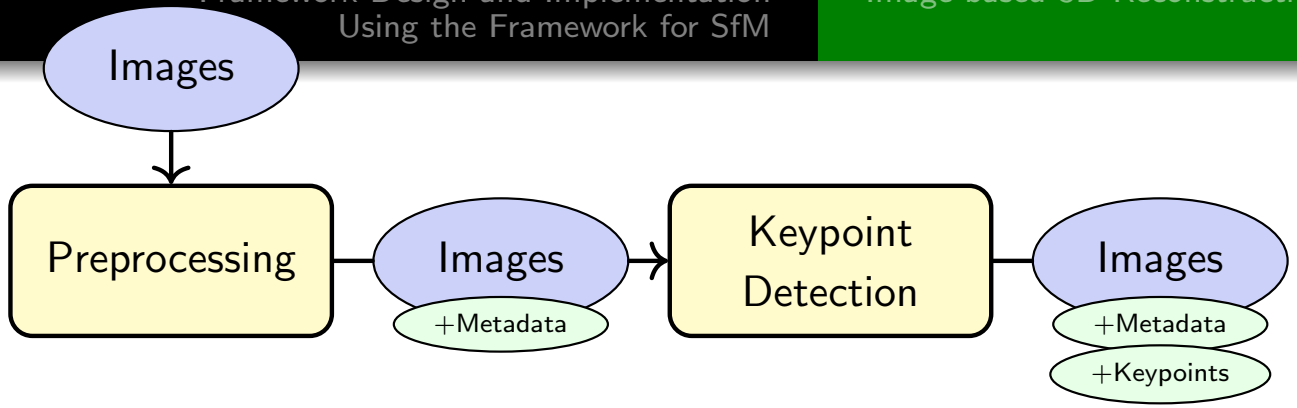
Framework goals:

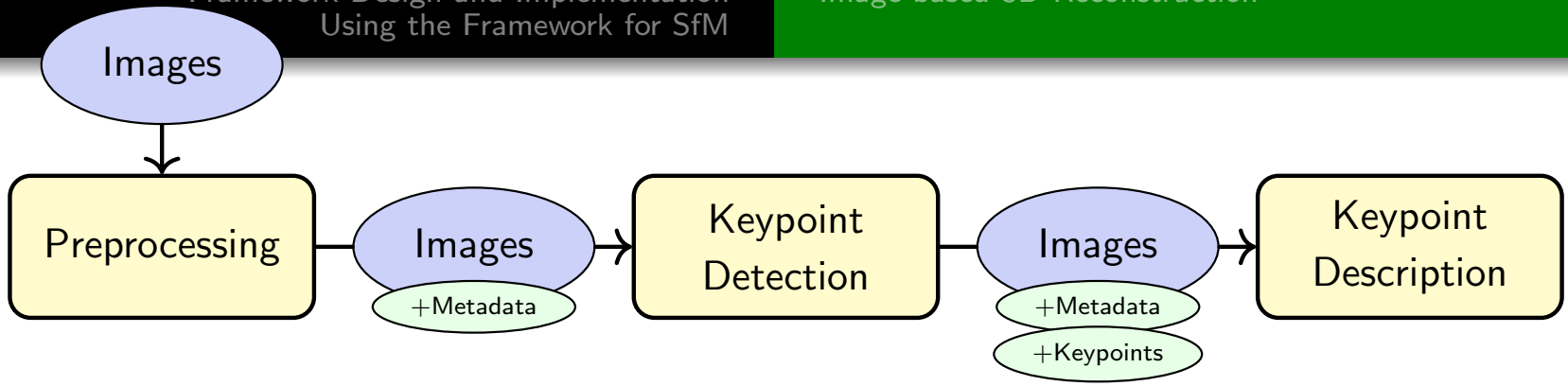
- flexible processing chain creation
- applicable to large reconstruction problems
- visualisation options

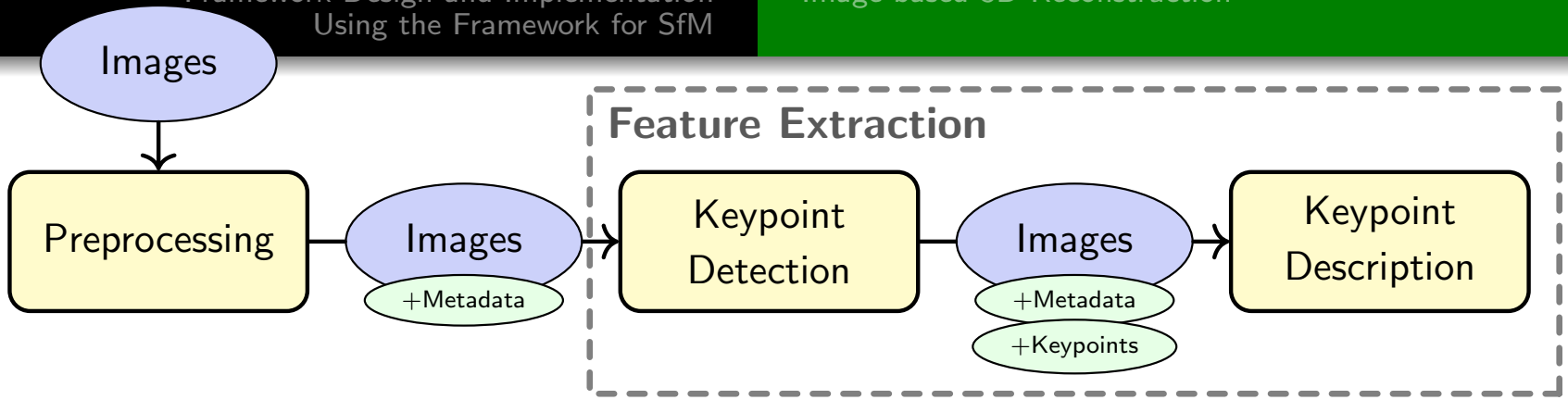
Table of Contents

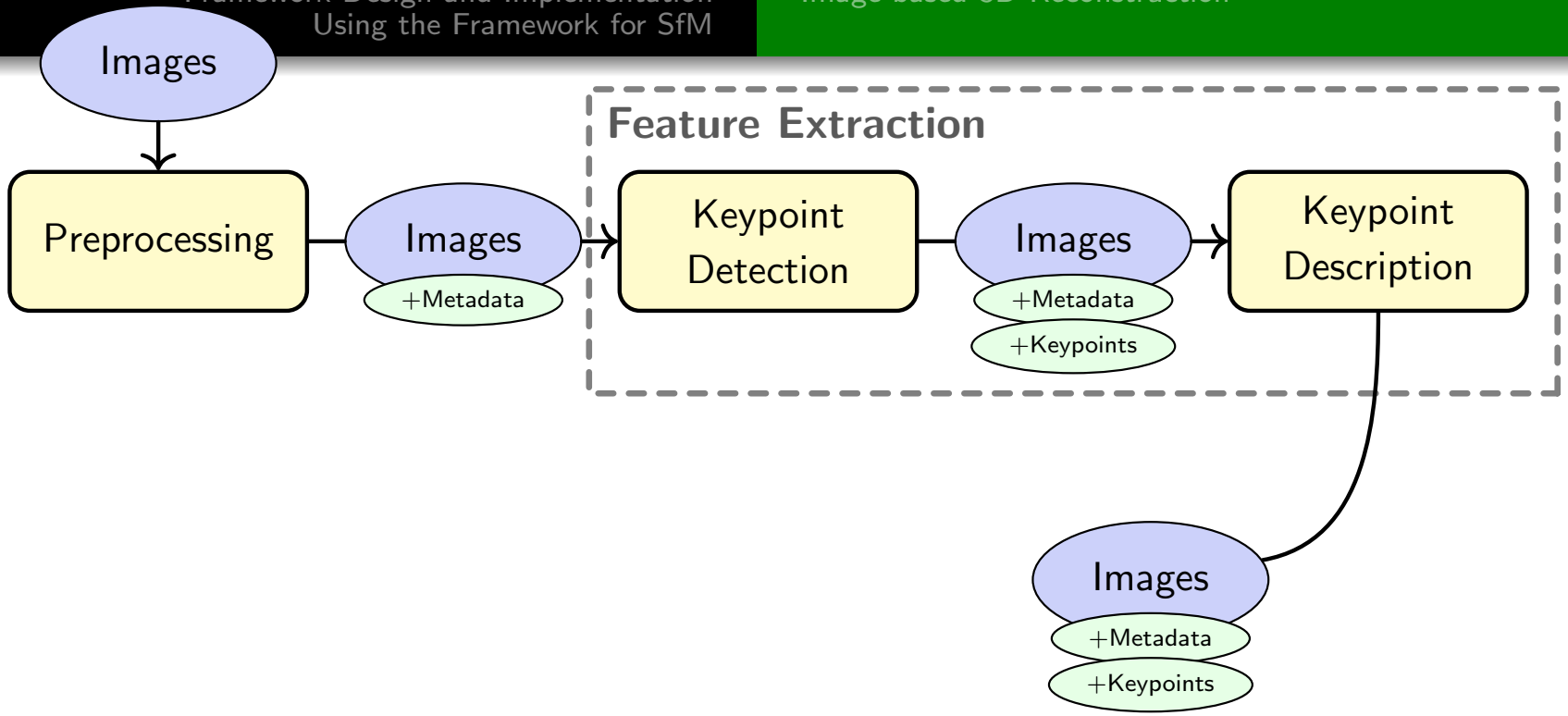

Images
Preprocessing

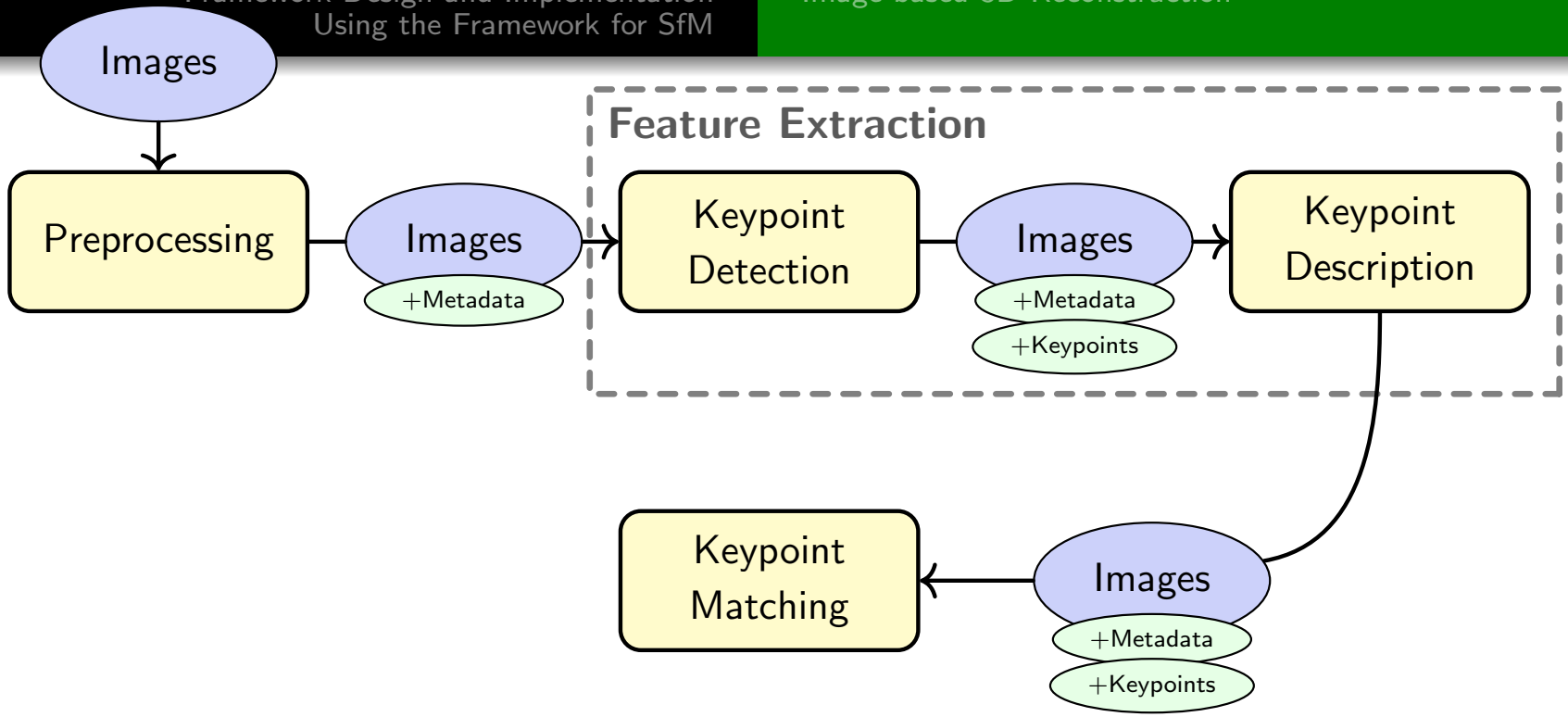

Images
Preprocessing
Images
+Metadata

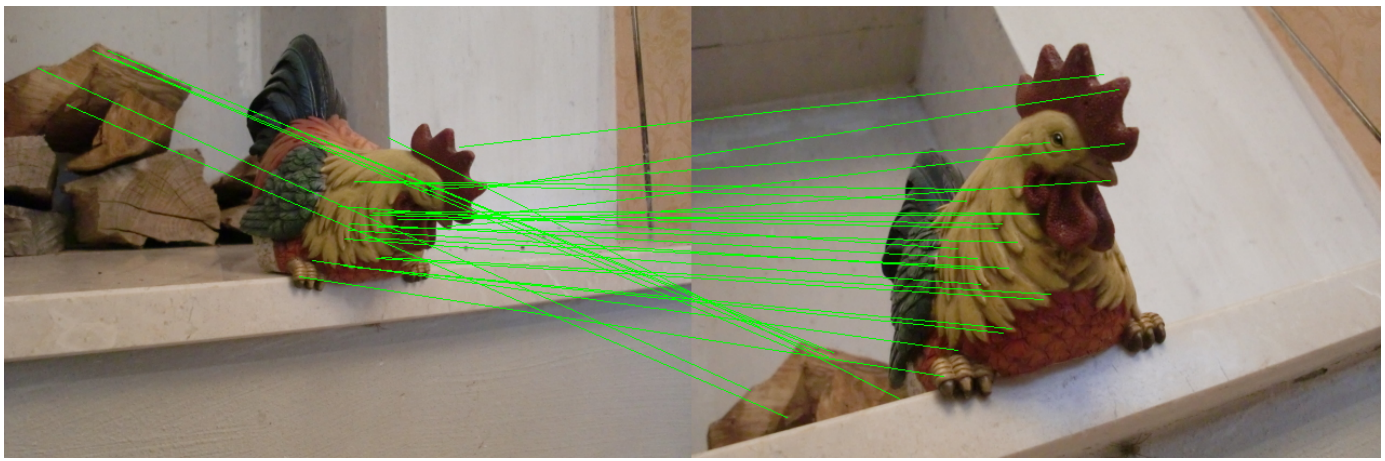

Images
Preprocessing
Images
+Metadata
Keypoint Detection
Images
+Metadata
+Keypoints

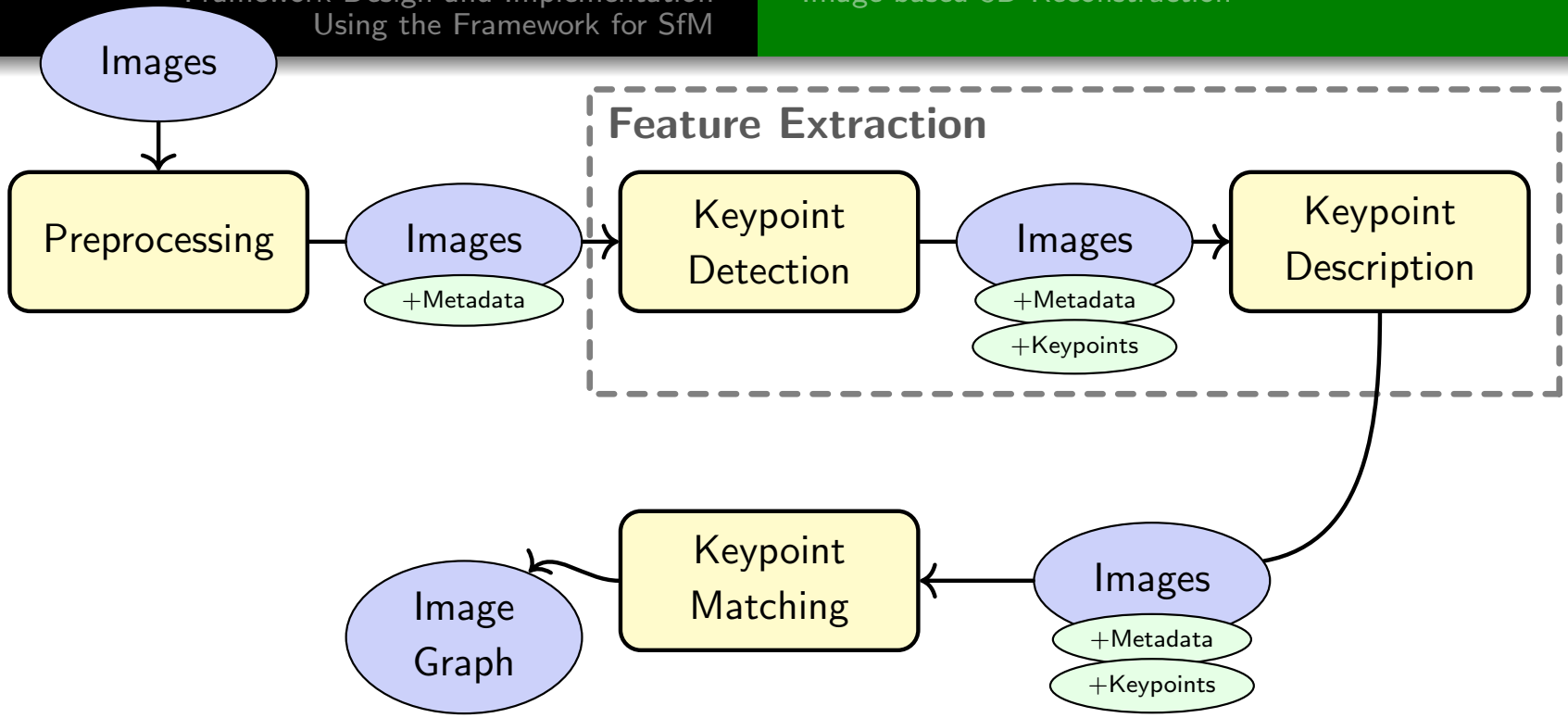

Images
Preprocessing
Images
+Metadata
Keypoint Detection
Images
+Metadata
+Keypoints
Keypoint Description

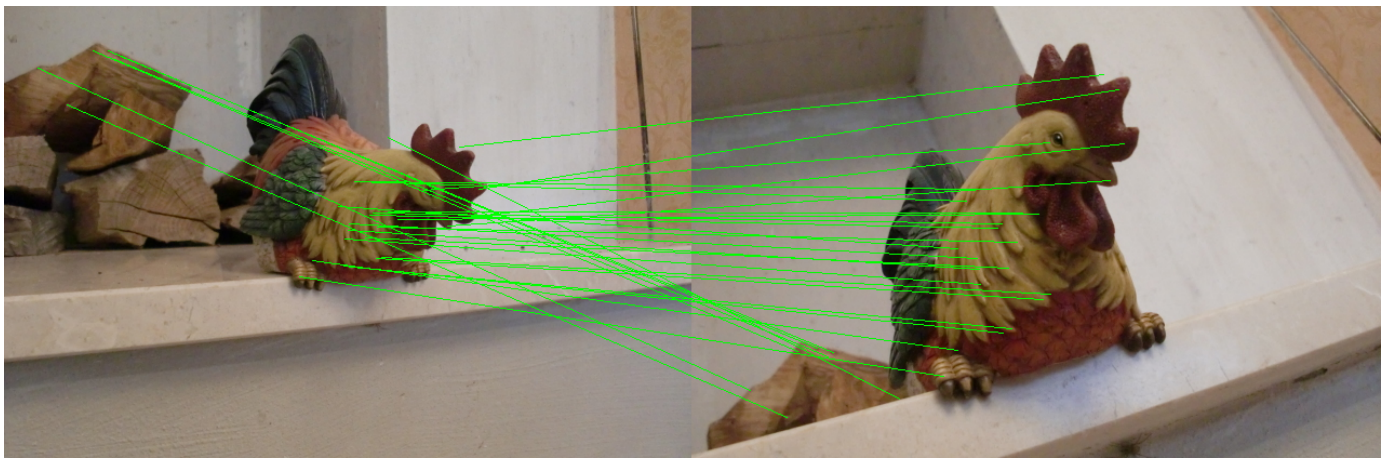

Images
Preprocessing
Images
+Metadata
Feature Extraction
Keypoint Detection
Images
+Metadata
+Keypoints
Keypoint Description

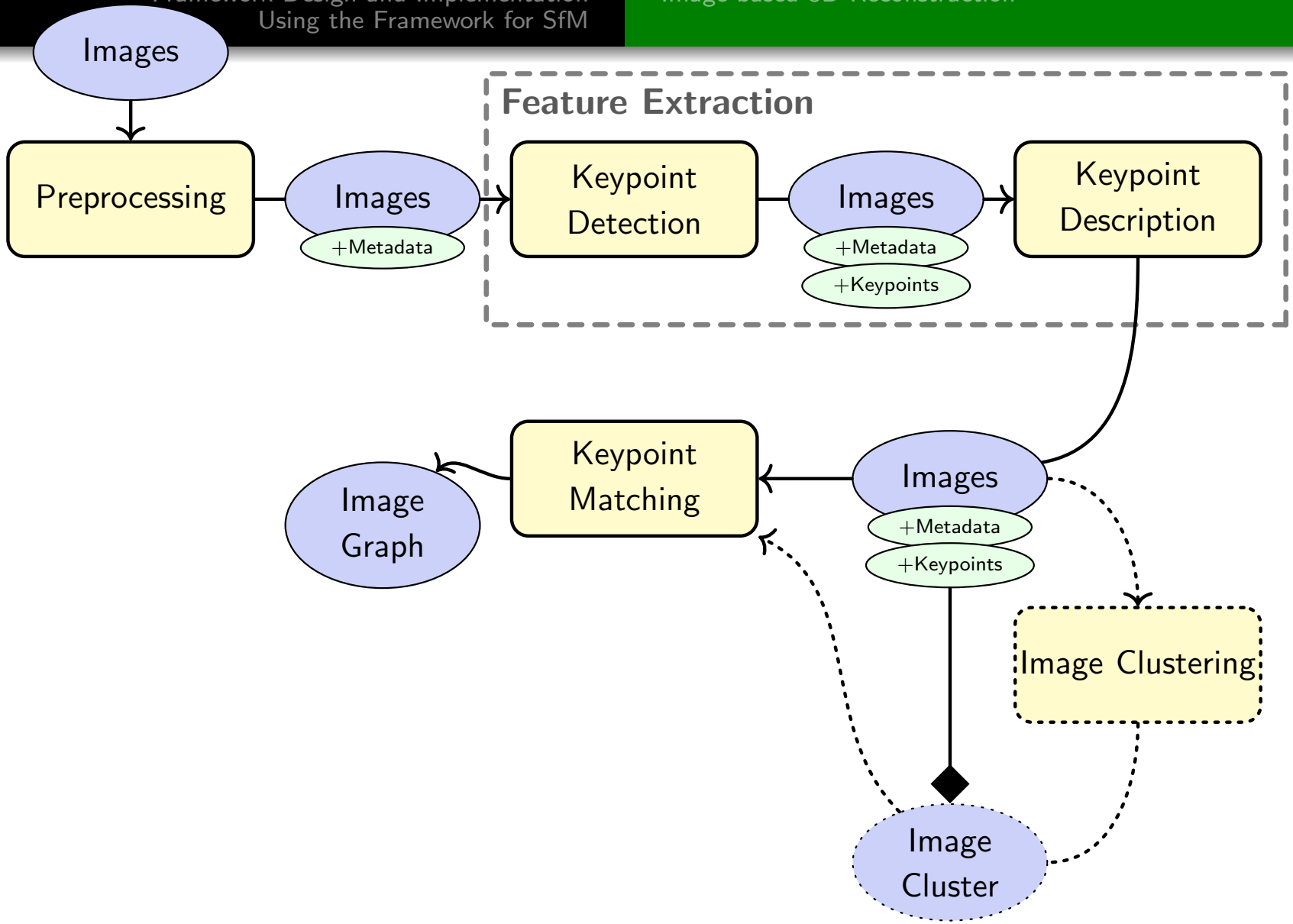

Images
Preprocessing
Images
+Metadata
Feature Extraction
Keypoint Detection
Images
+Metadata
+Keypoints
Keypoint Description
Images
+Metadata
+Keypoints

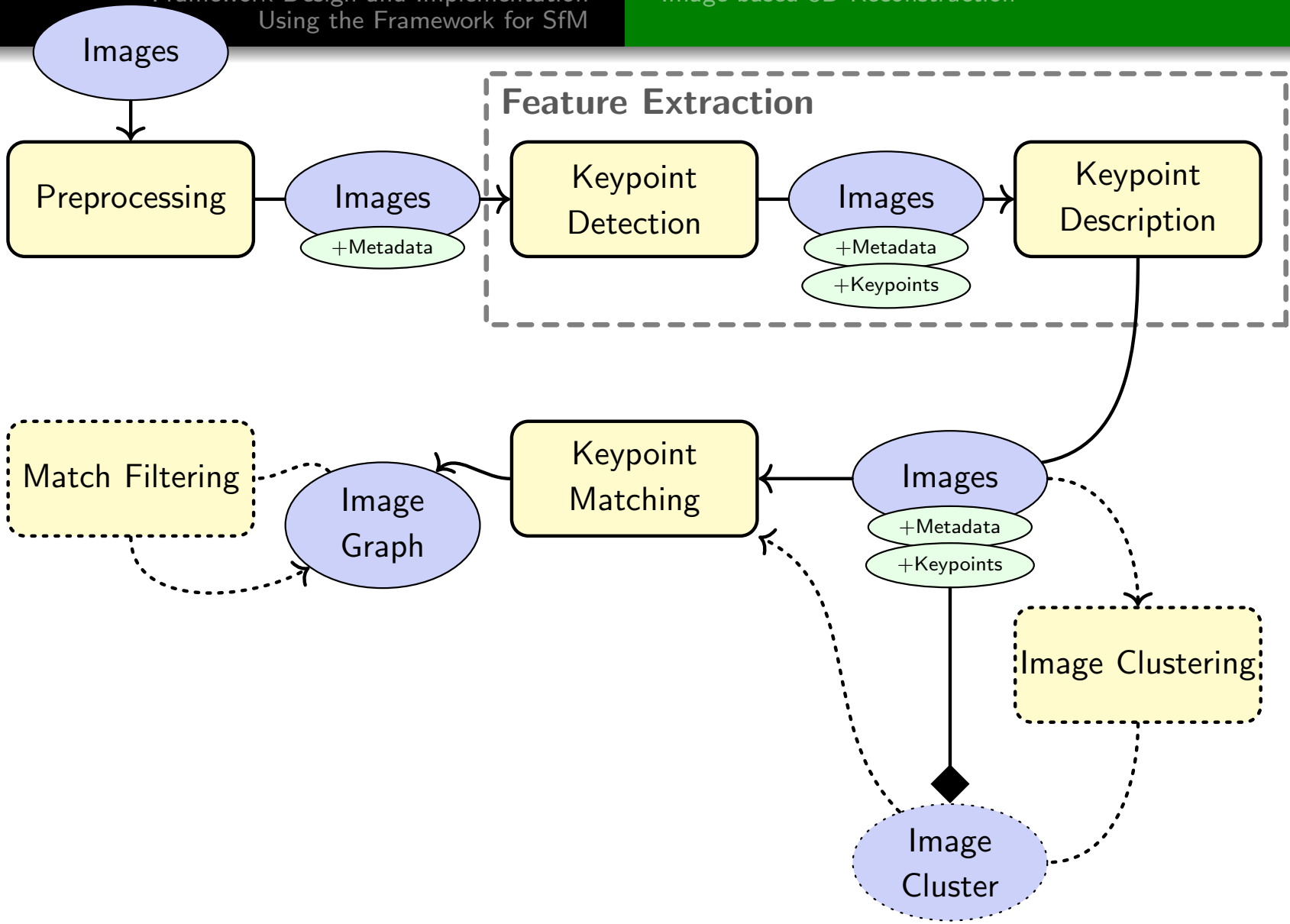

Images
Feature Extraction
Preprocessing
Images
+Metadata
Keypoint Detection
Images
+Metadata
+Keypoints
Keypoint Description
Keypoint Matching
Images
+Metadata
+Keypoints

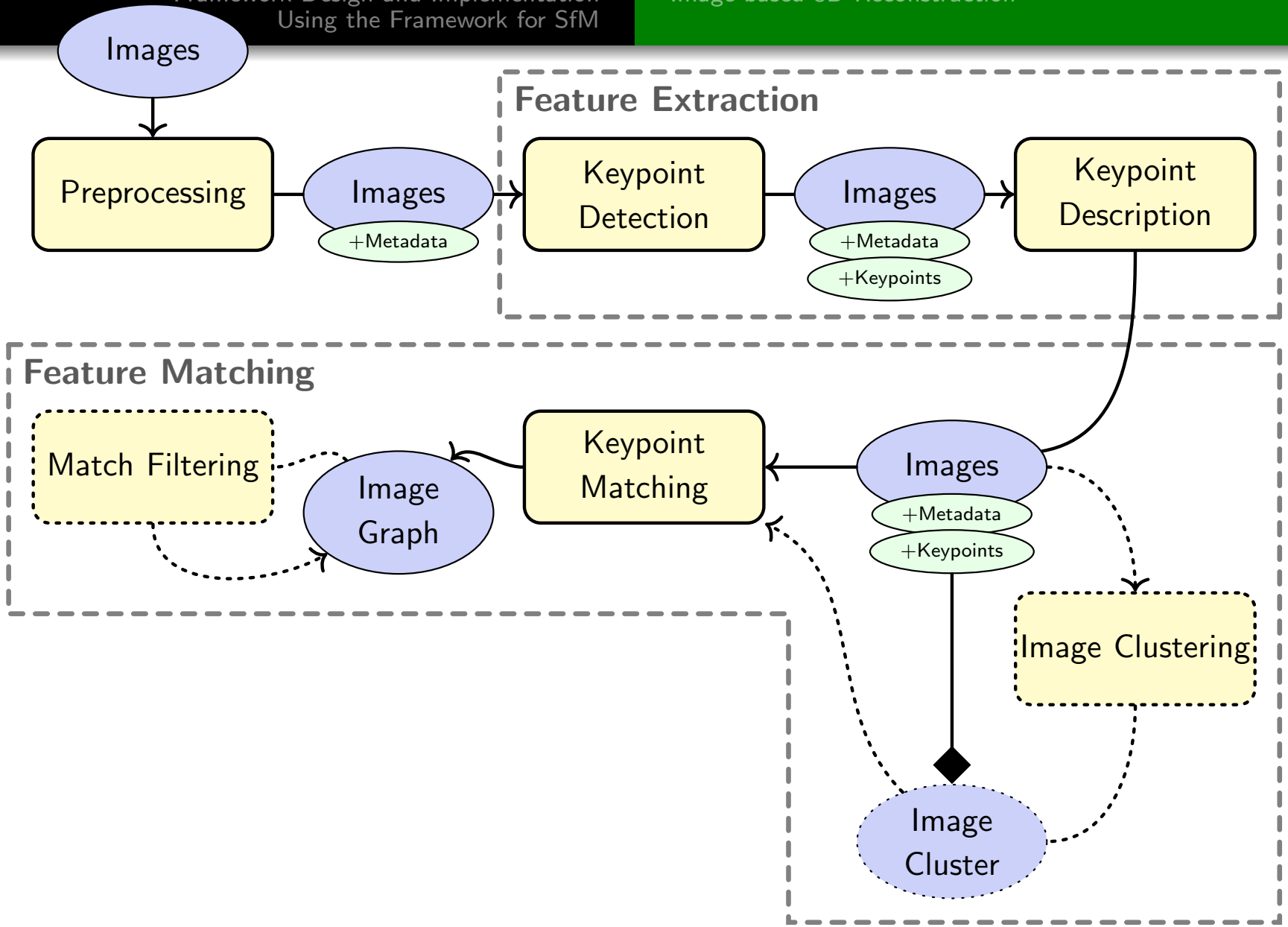


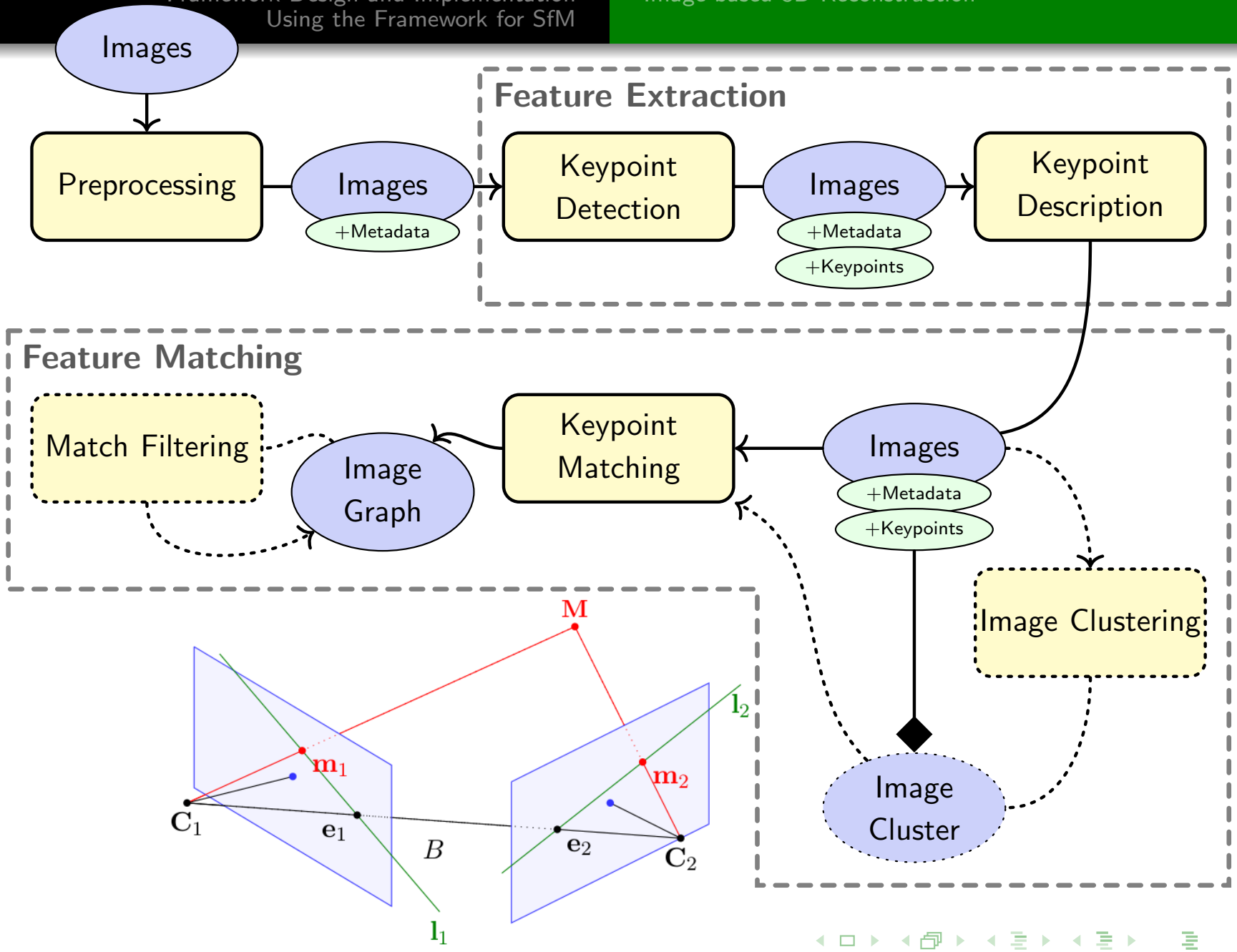

Images
Feature Extraction
Preprocessing
Images
+Metadata
Keypoint Detection
Images
+Metadata
+Keypoints
Keypoint Description
Image Graph
Keypoint Matching
Images
+Metadata
+Keypoints

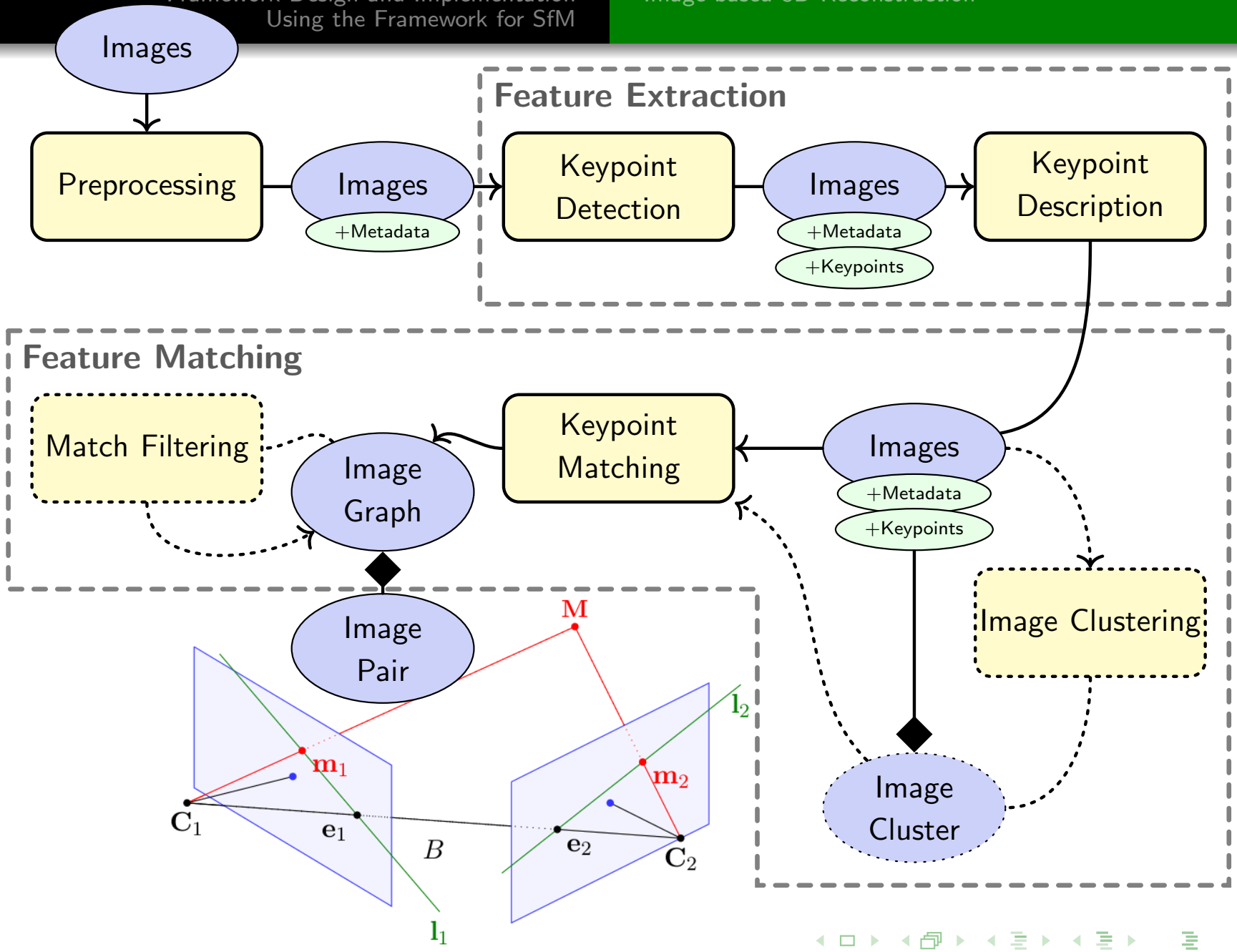


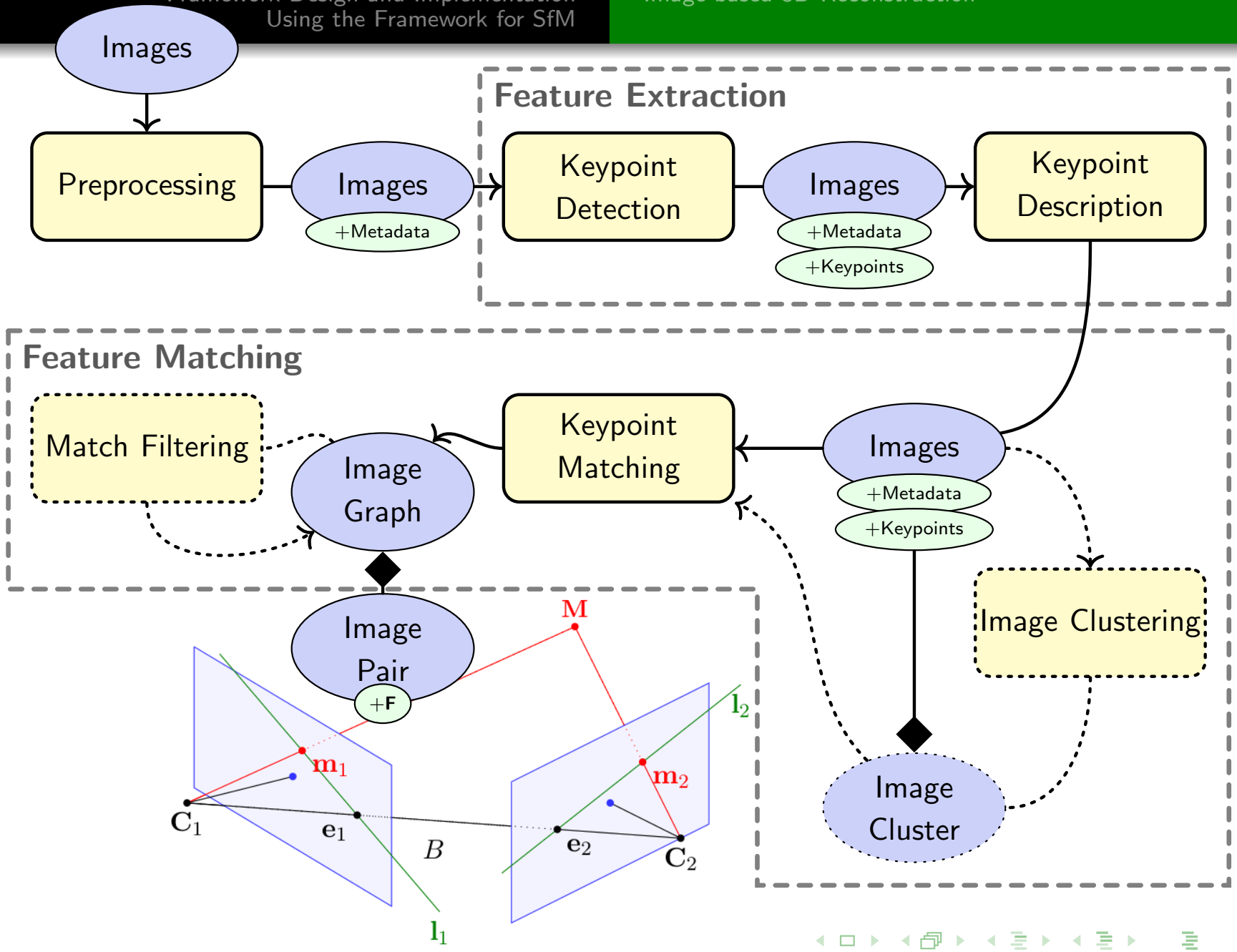

Images
Feature Extraction
Preprocessing
Images
+Metadata
Keypoint Detection
Images
+Metadata
+Keypoints
Keypoint Description
Image Graph
Keypoint Matching
Images
+Metadata
+Keypoints
Image Clustering
Image Cluster

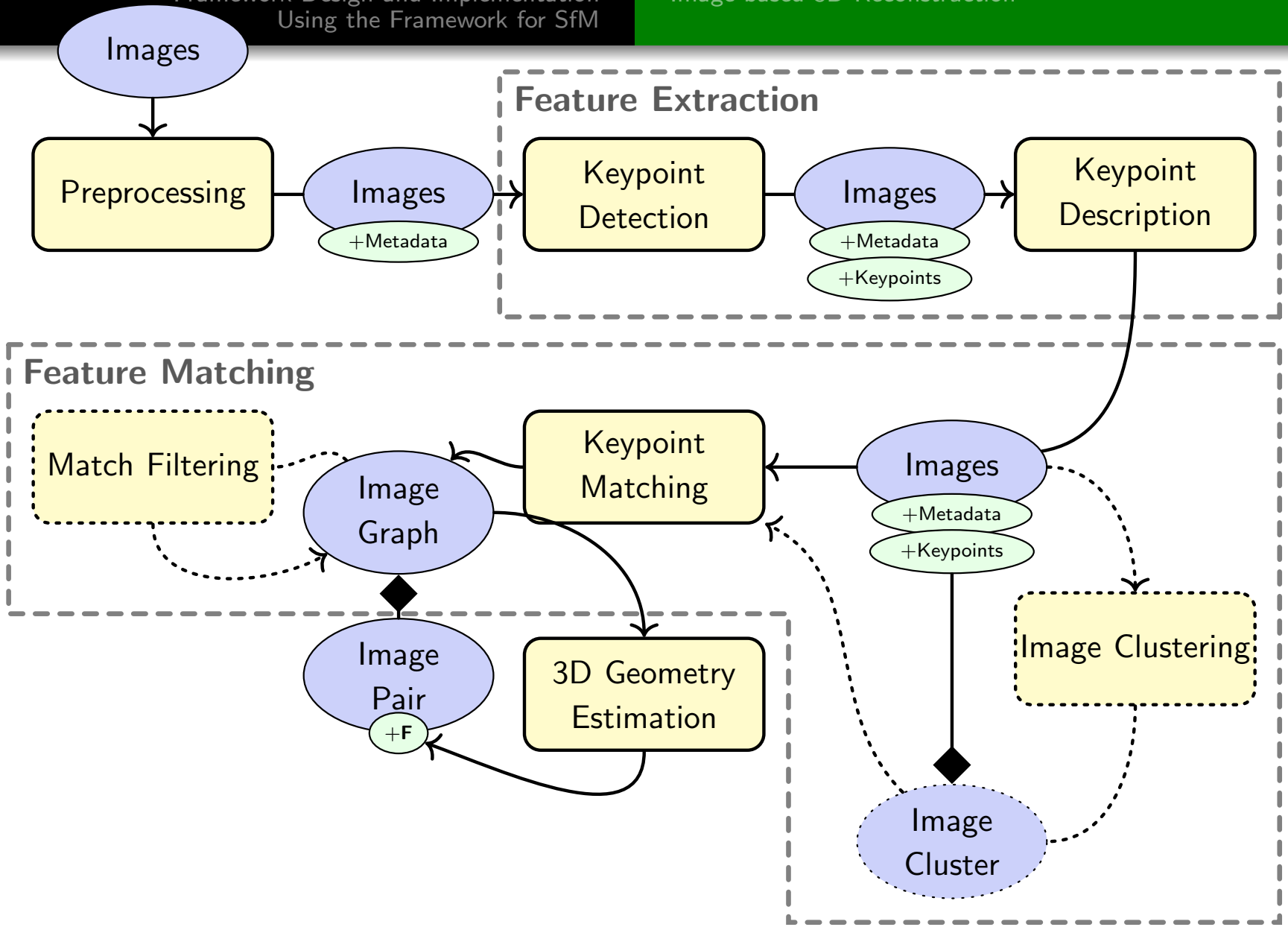

Images
Feature Extraction
Preprocessing
Images
+Metadata
Keypoint Detection
Images
+Metadata
+Keypoints
Keypoint Description
Match Filtering
Image Graph
Keypoint Matching
Images
+Metadata
+Keypoints
Image Clustering
Image Cluster

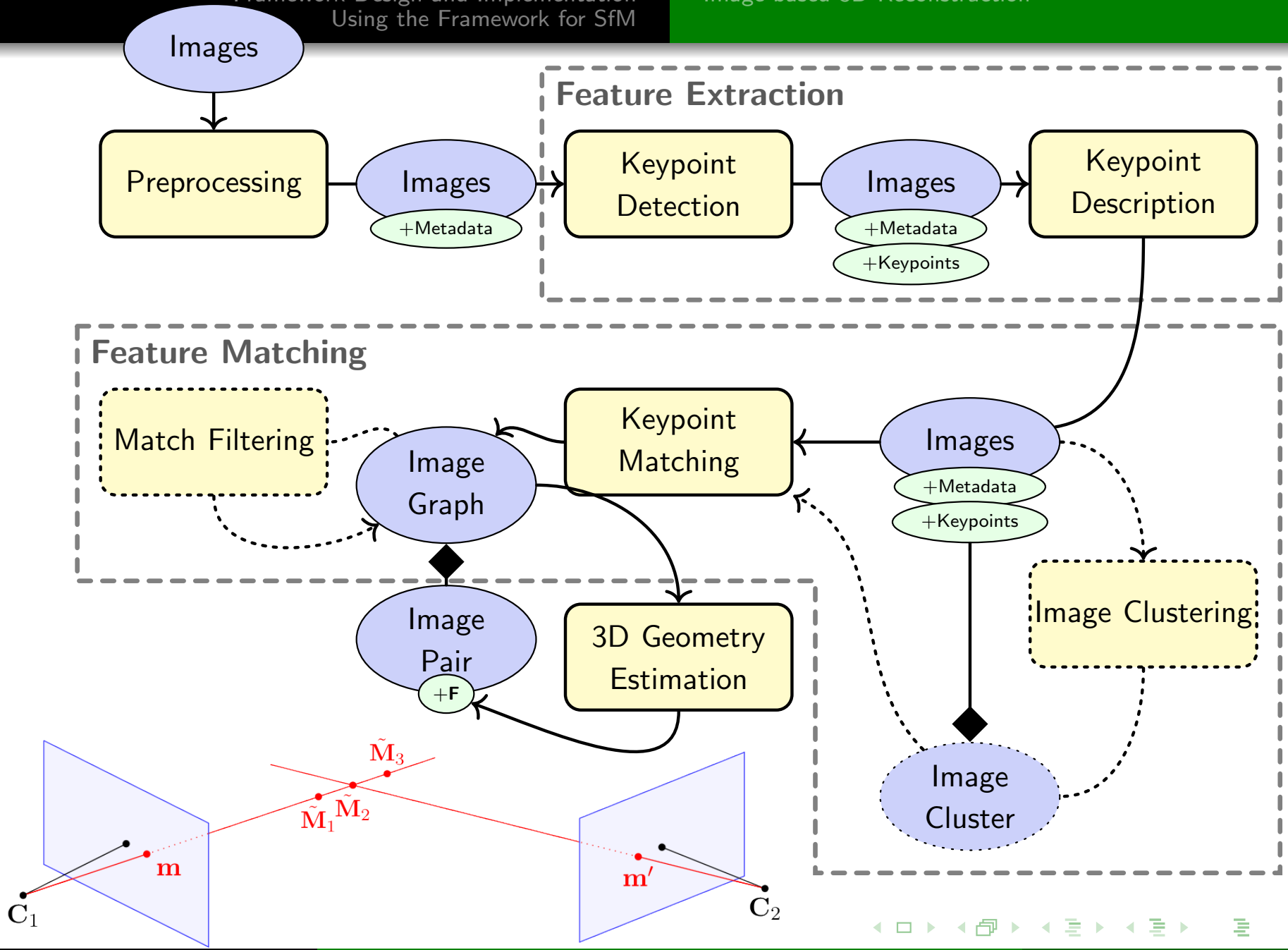

Images
Feature Extraction
Preprocessing
Images
+Metadata
Keypoint Detection
Images
+Metadata
+Keypoints
Keypoint Description
Feature Matching
Match Filtering
Image Graph
Keypoint Matching
Images
+Metadata
+Keypoints
Image Clustering
Image Cluster

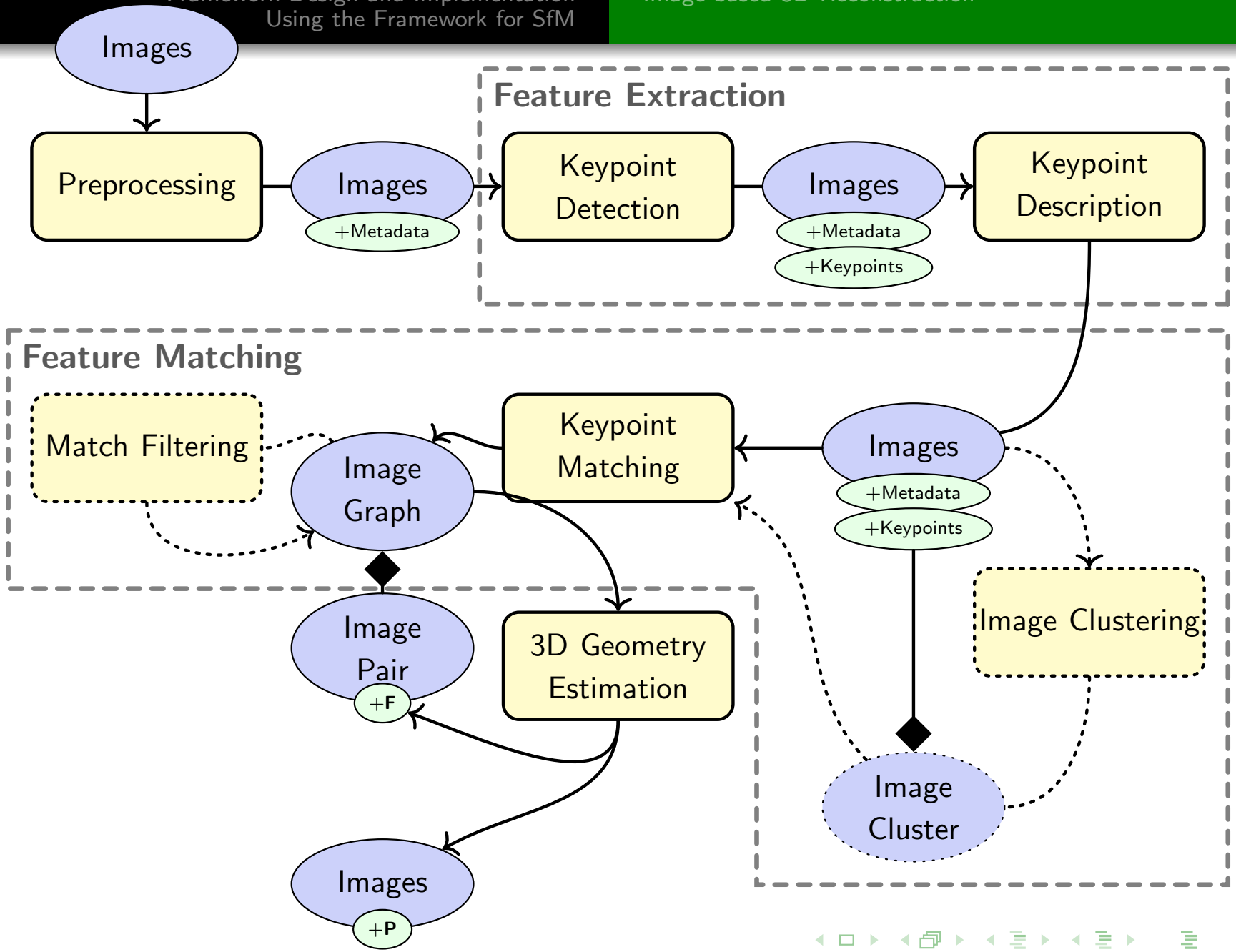

Images
Feature Extraction
Preprocessing
Images
+Metadata
Keypoint Detection
Images
+Metadata
+Keypoints
Keypoint Description
Feature Matching
Match Filtering
Image Graph
Keypoint Matching
Images
+Metadata
+Keypoints
Image Clustering
Image Cluster
M
m1
m2
l2
C1
e1
B
e2
C2
l1

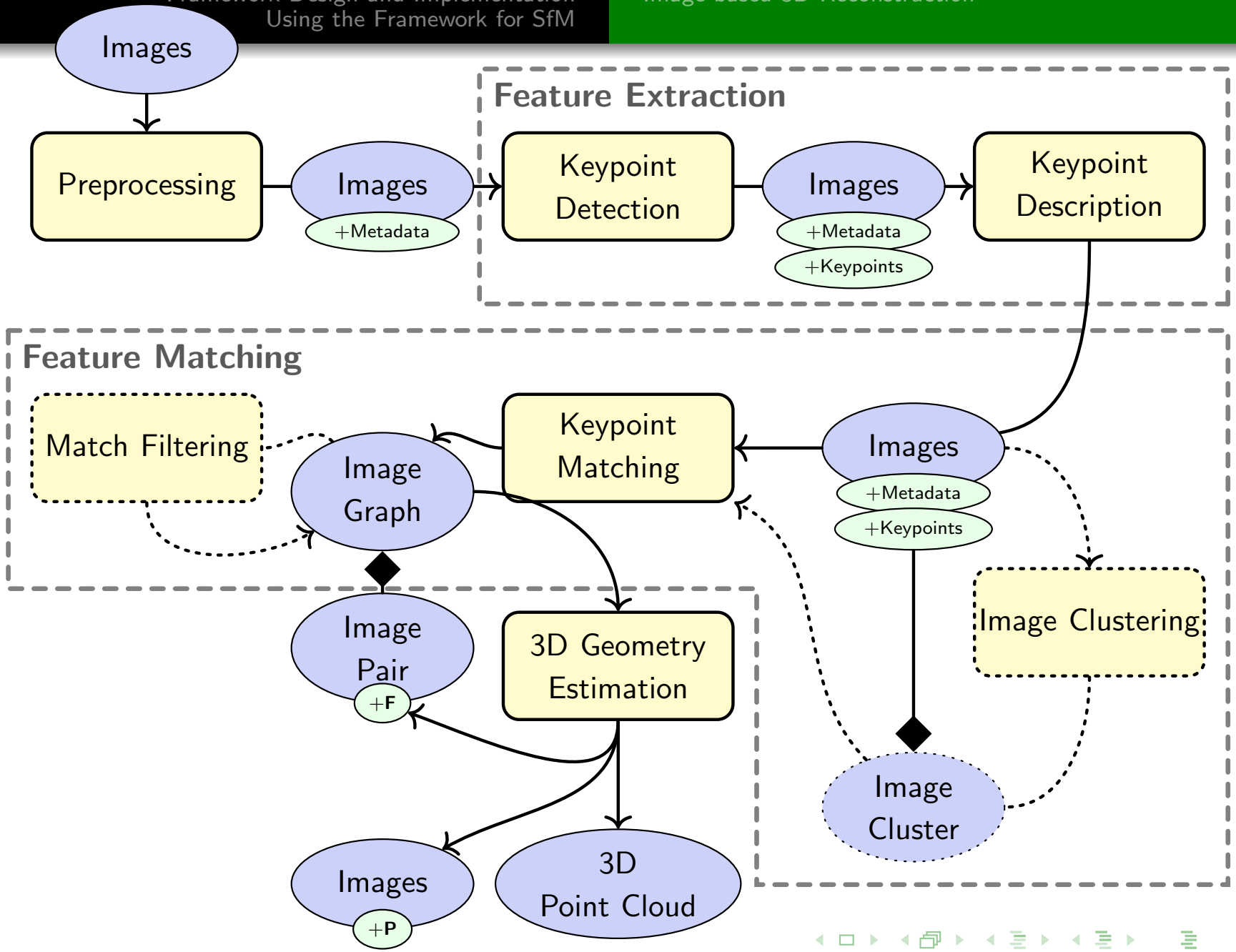

Images
Feature Extraction
Preprocessing
Images
+Metadata
Keypoint Detection
Images
+Metadata
+Keypoints
Keypoint Description
Feature Matching
Match Filtering
Image Graph
Keypoint Matching
Images
+Metadata
+Keypoints
Image Clustering
Image Pair
Image Cluster
M
$\mathbf{m}_1$
$\mathbf{m}_2$
$\mathbf{l}_2$
$\mathbf{C}_1$
$\mathbf{e}_1$
B
$\mathbf{e}_2$
$\mathbf{C}_2$
$\mathbf{l}_1$

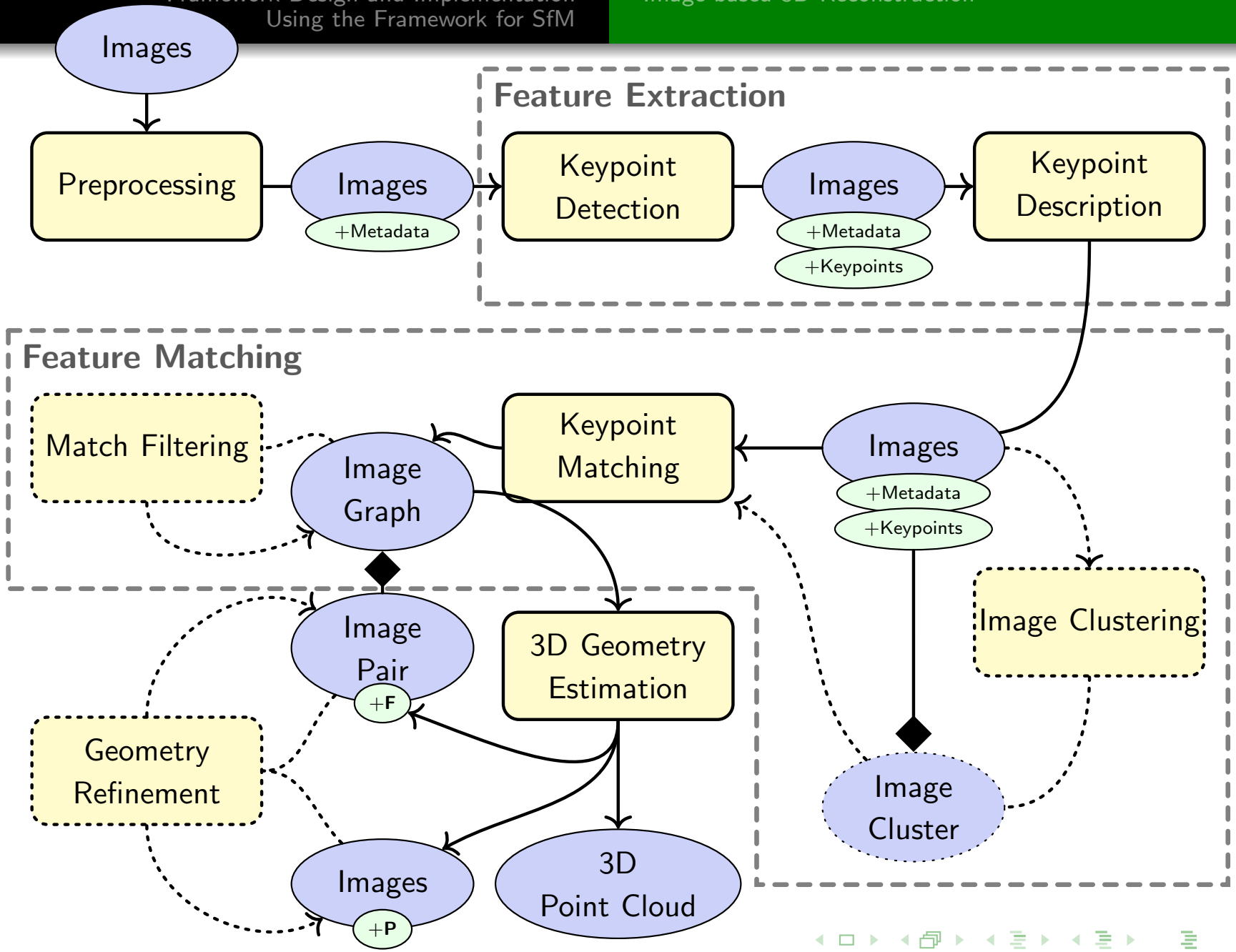

Images
Feature Extraction
Preprocessing
Images
+Metadata
Keypoint Detection
Images
+Metadata
+Keypoints
Keypoint Description
Feature Matching
Match Filtering
Image Graph
Keypoint Matching
Images
+Metadata
+Keypoints
Image Clustering
Image Pair
+F
Image Cluster
M
m1
m2
C1
e1
B
e2
C2
l1
l2


Images
Feature Extraction
Preprocessing
Images
+Metadata
Keypoint Detection
Images
+Metadata
+Keypoints
Keypoint Description
Feature Matching
Match Filtering
Image Graph
Keypoint Matching
Images
+Metadata
+Keypoints
Image Pair
+F
3D Geometry Estimation
Image Clustering
Image Cluster

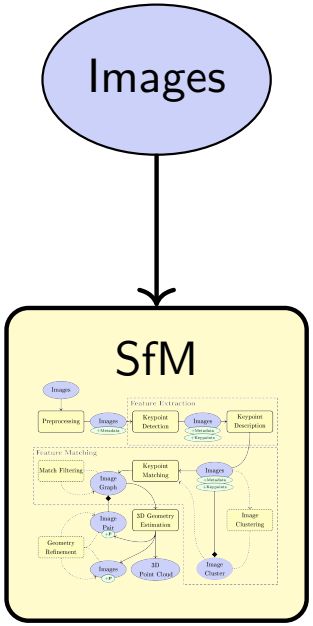

Images
Feature Extraction
Preprocessing
Images
+Metadata
Keypoint Detection
Images
+Metadata
+Keypoints
Keypoint Description
Feature Matching
Match Filtering
Image Graph
Keypoint Matching
Images
+Metadata
+Keypoints
Image Clustering
Image Pair
+F
3D Geometry Estimation
Image Cluster
$\tilde{M}_3$
$\tilde{M}_1$
$\tilde{M}_2$
m
m′
C_1
C_2


Images
Feature Extraction
Preprocessing
Images
+Metadata
Keypoint Detection
Images
+Metadata
+Keypoints
Keypoint Description
Feature Matching
Match Filtering
Image Graph
Keypoint Matching
Images
+Metadata
+Keypoints
Image Clustering
Image Pair
+F
3D Geometry Estimation
Image Cluster
Images
+P


Images
Feature Extraction
Preprocessing
Images
+Metadata
Keypoint Detection
Images
+Metadata
+Keypoints
Keypoint Description
Feature Matching
Match Filtering
Image Graph
Keypoint Matching
Images
+Metadata
+Keypoints
Image Clustering
Image Pair
+F
3D Geometry Estimation
Image Cluster
Images
+P
3D Point Cloud

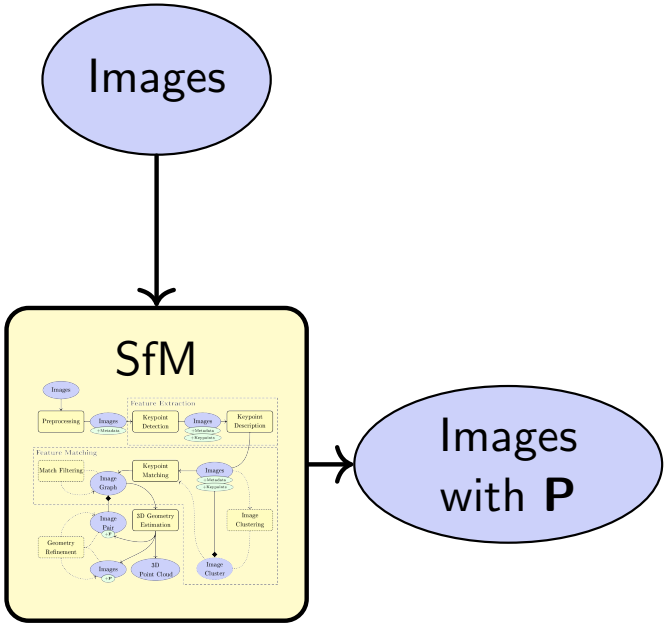

Images
Feature Extraction
Preprocessing
Images
+Metadata
Keypoint Detection
Images
+Metadata
+Keypoints
Keypoint Description
Feature Matching
Match Filtering
Image Graph
Keypoint Matching
Images
+Metadata
+Keypoints
Image Clustering
Image Pair
+F
3D Geometry Estimation
Geometry Refinement
Image Cluster
Images
+P
3D Point Cloud

Highlevel overview of image-based 3D reconstruction



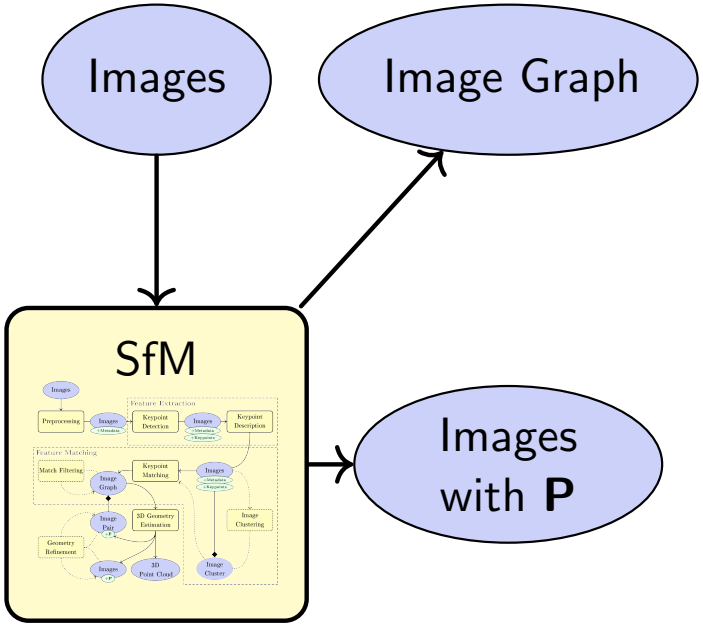
Mesh

Highlevel overview of image-based 3D reconstruction



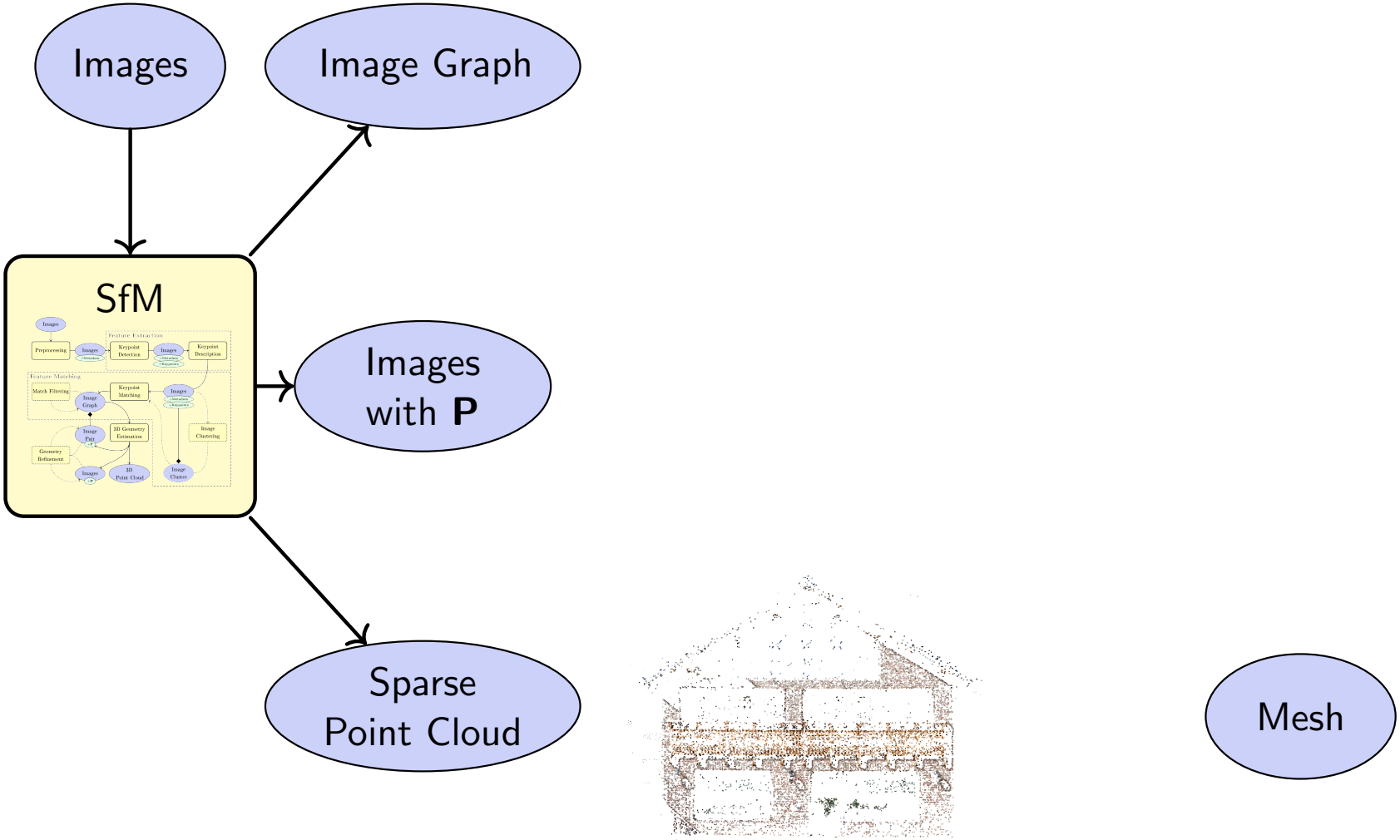
Mesh

Highlevel overview of image-based 3D reconstruction

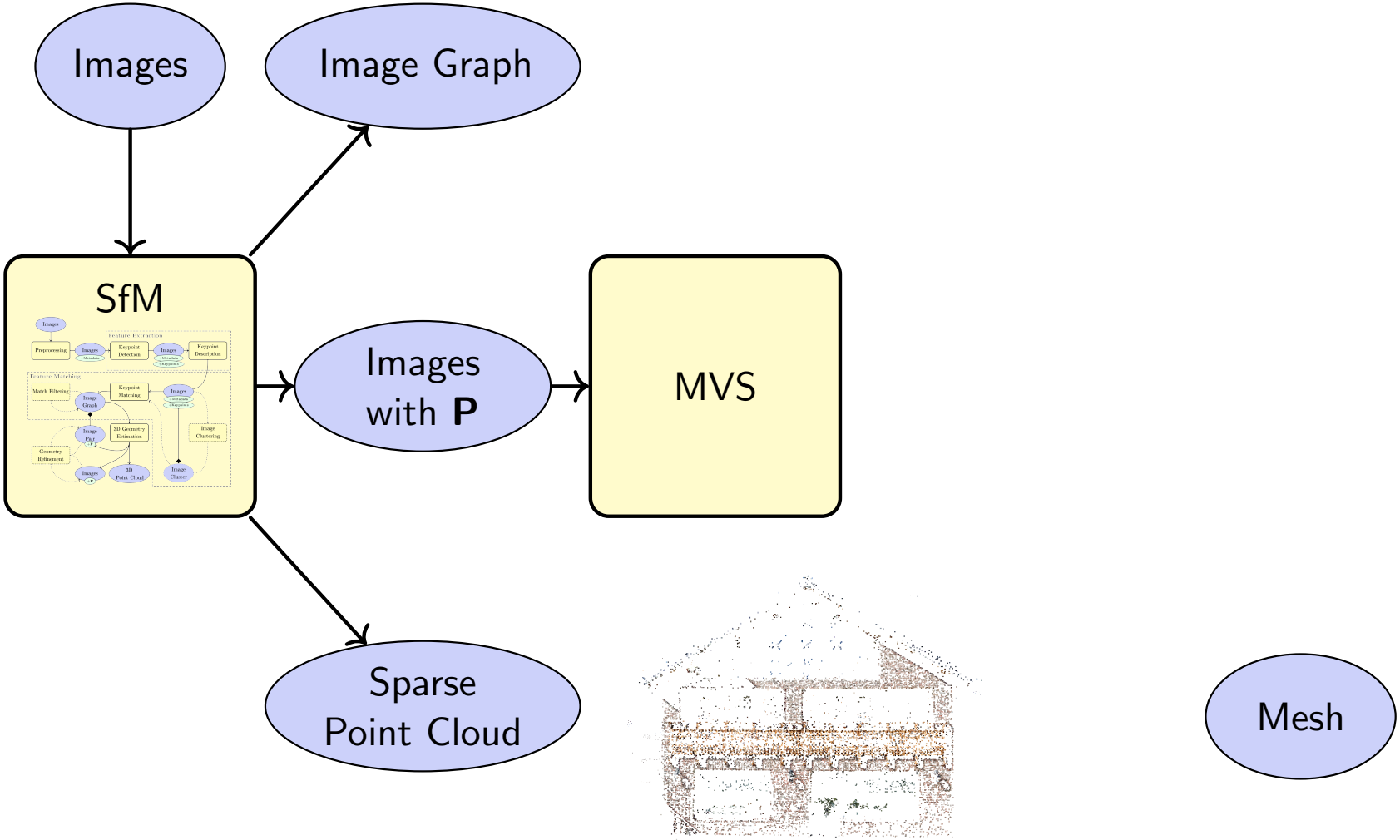


Mesh

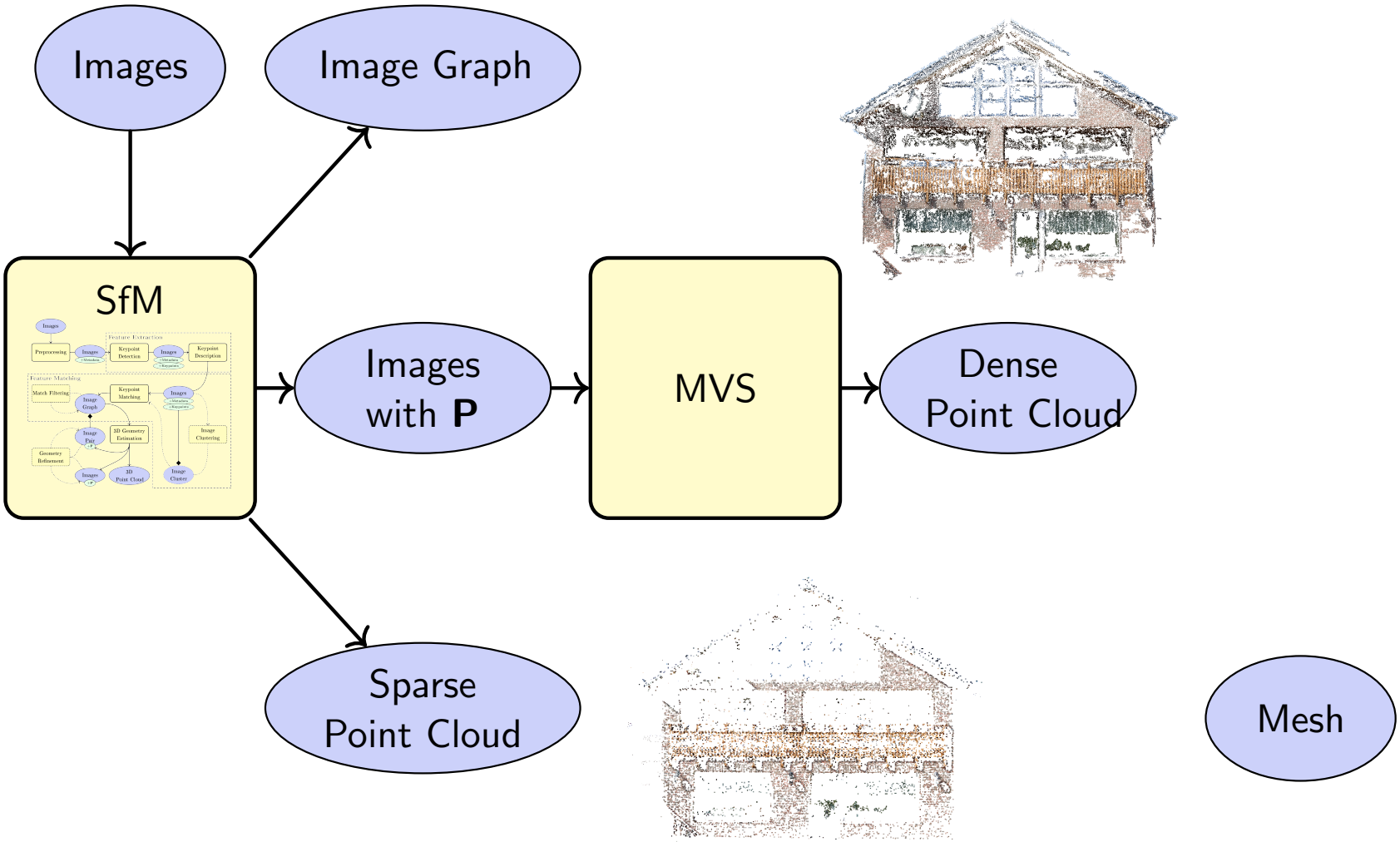
Highlevel overview of image-based 3D reconstruction



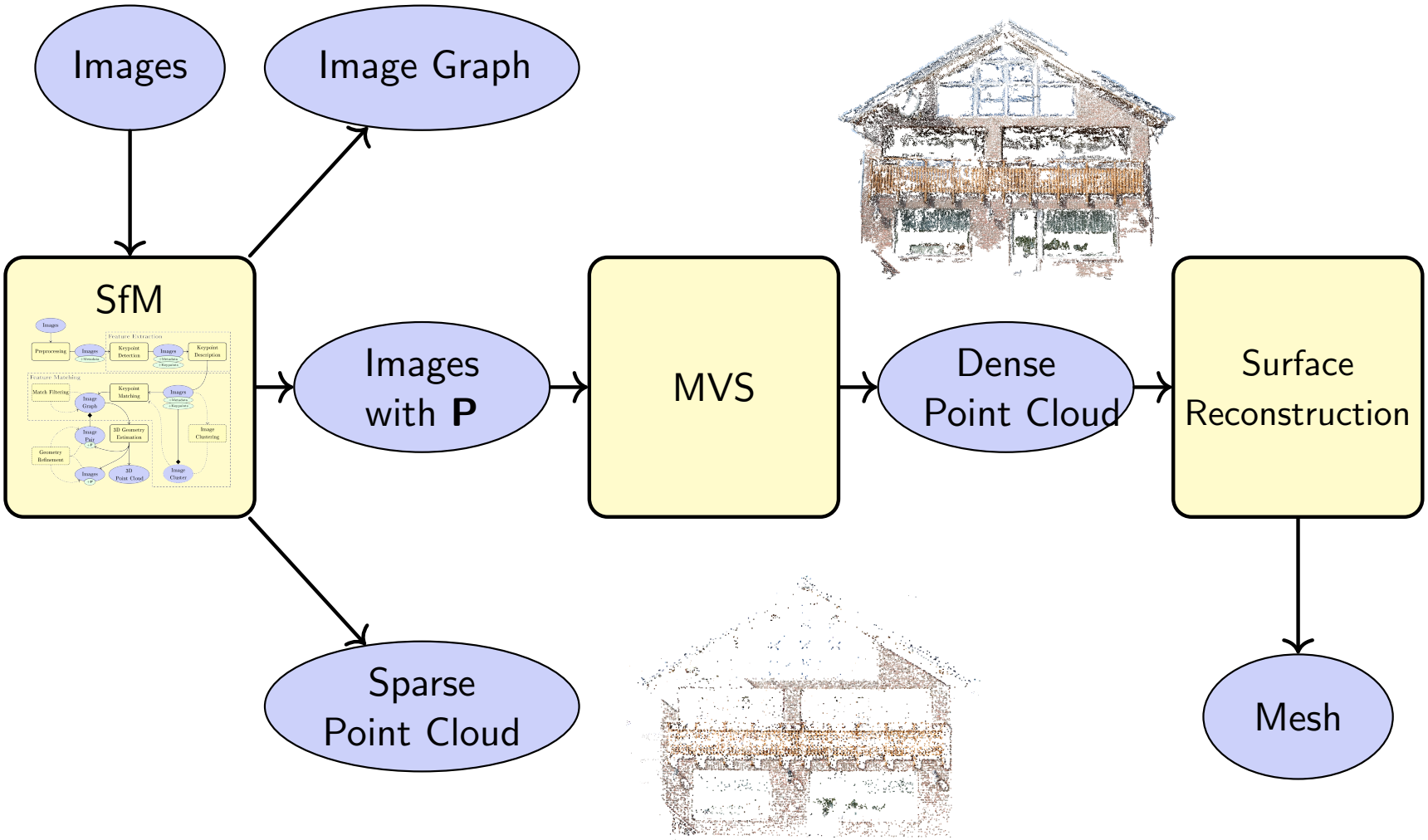
Highlevel overview of image-based 3D reconstruction



Highlevel overview of image-based 3D reconstruction



Highlevel overview of image-based 3D reconstruction



Highlevel overview of image-based 3D reconstruction

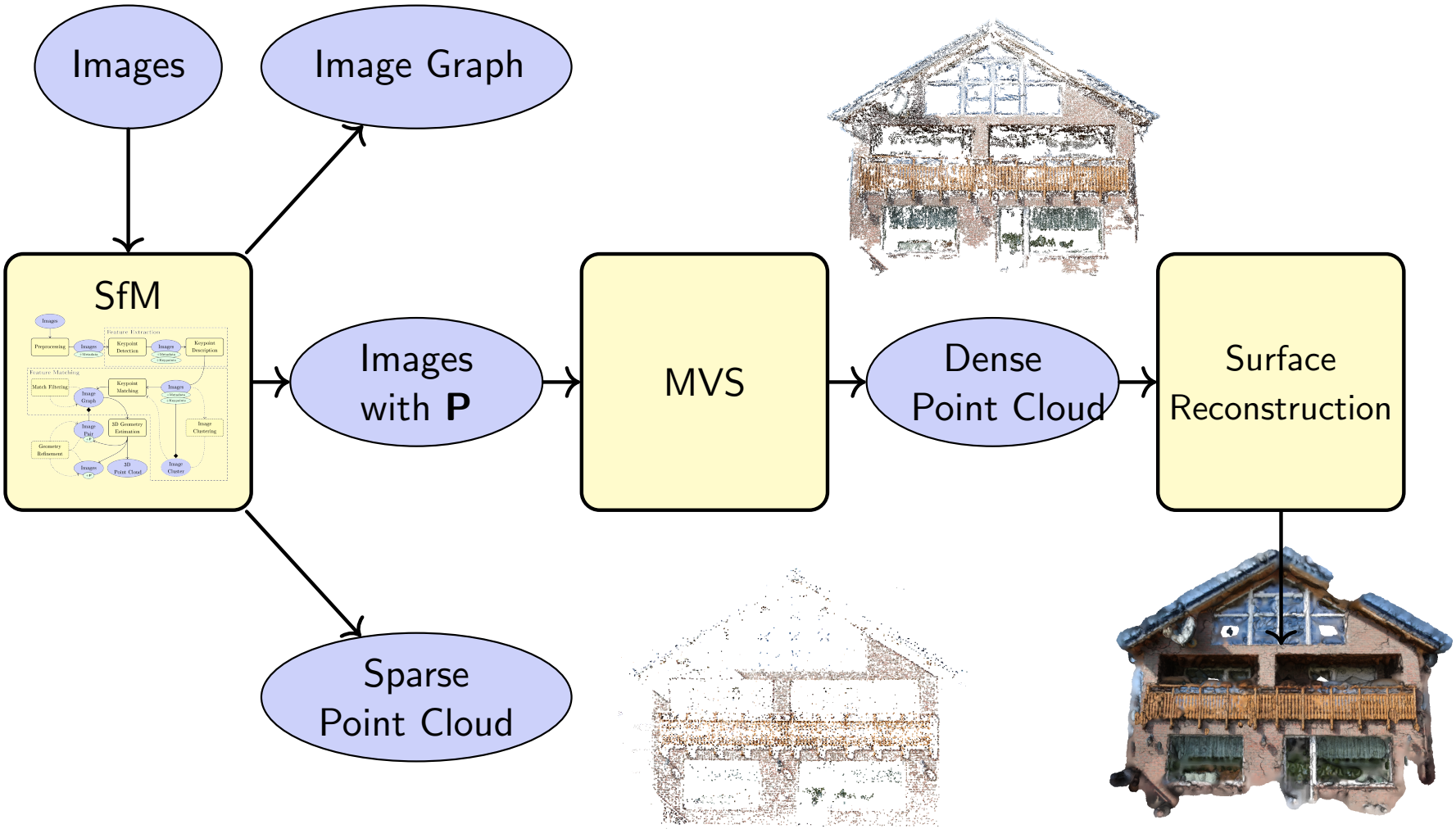


Table of Contents

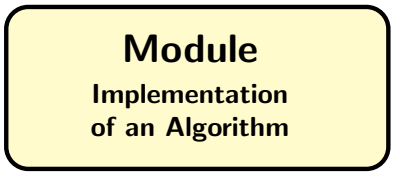
Module and Data

Module

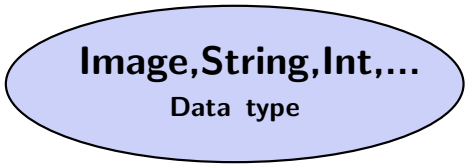
Module
Implementation of an Algorithm

Module and Data

Module

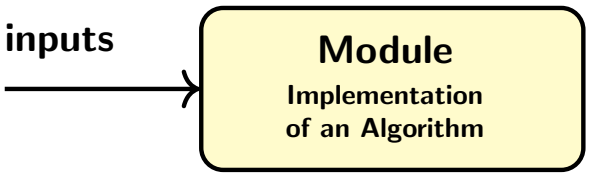


Data

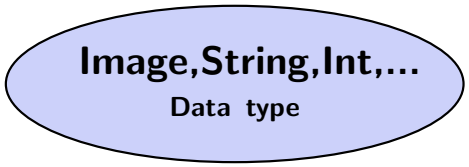


Module and Data

Module

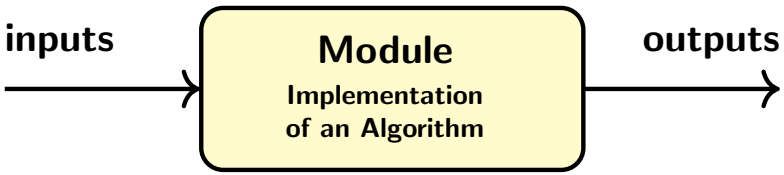


Data

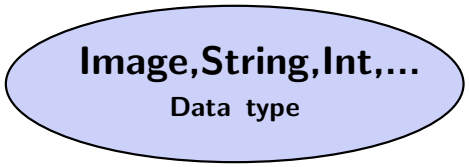


Module and Data

Module

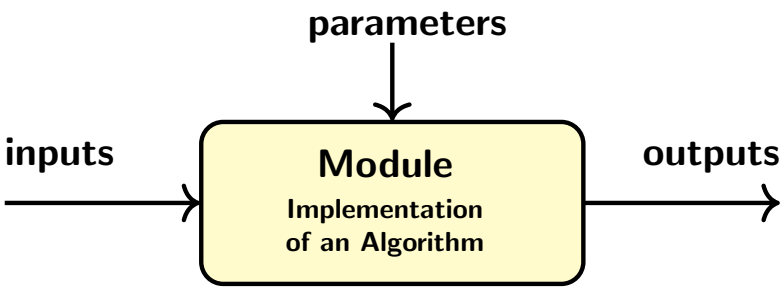


Data

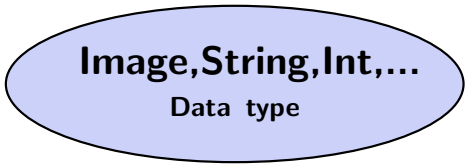


Module and Data

Module

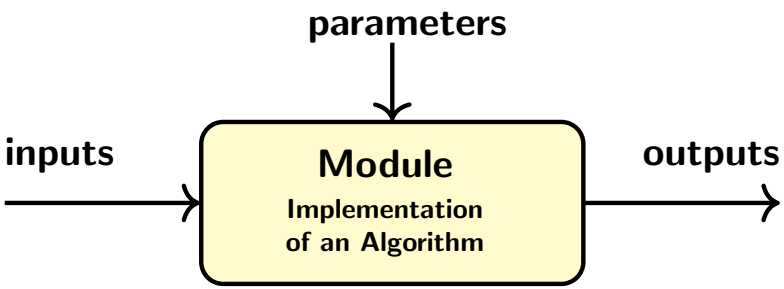


Data

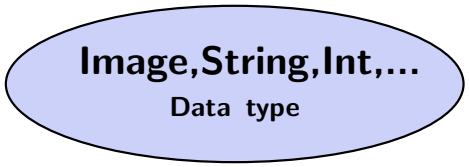


Module and Data

Module



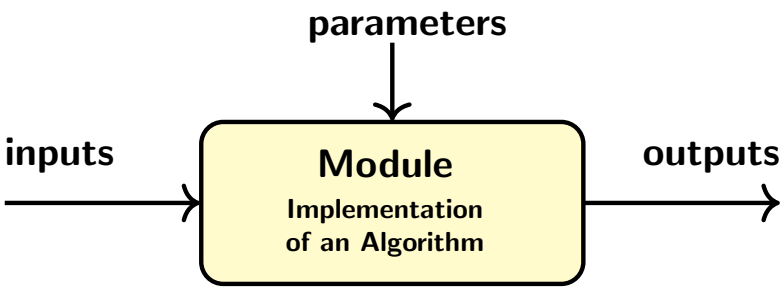
Data



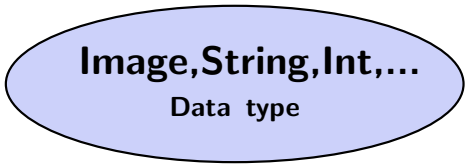
- clear interface $\rightarrow$ reuse of code in different contexts

Module and Data

Module



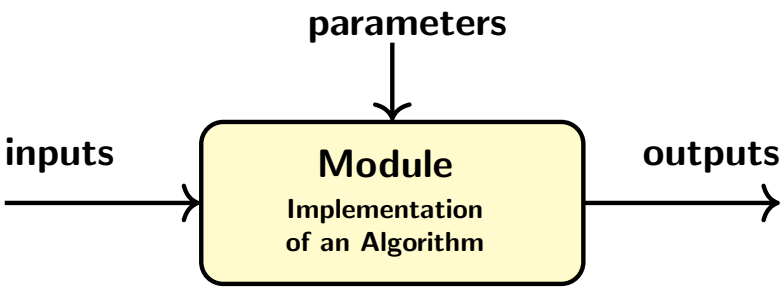
Data



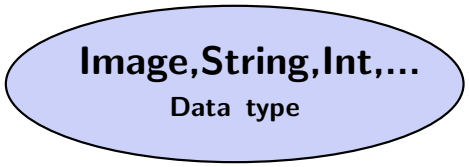
- clear interface $\rightarrow$ reuse of code in different contexts
- self-contained unit $\rightarrow$ parallelizable

Module and Data

Module



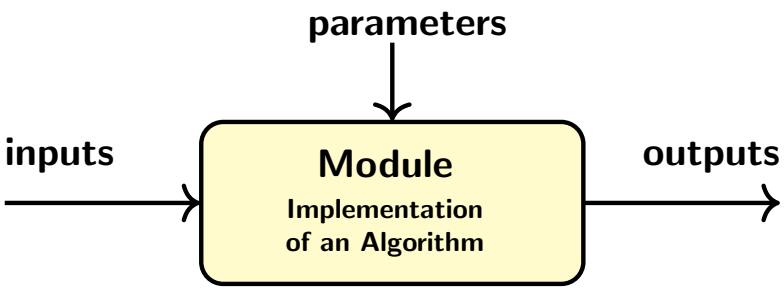
Data



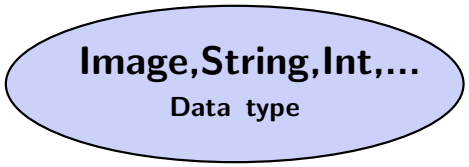
- clear interface $\rightarrow$ reuse of code in different contexts
- self-contained unit $\rightarrow$ parallelizable
- common data type implementations $\rightarrow$ data flow control

Module and Data

Module

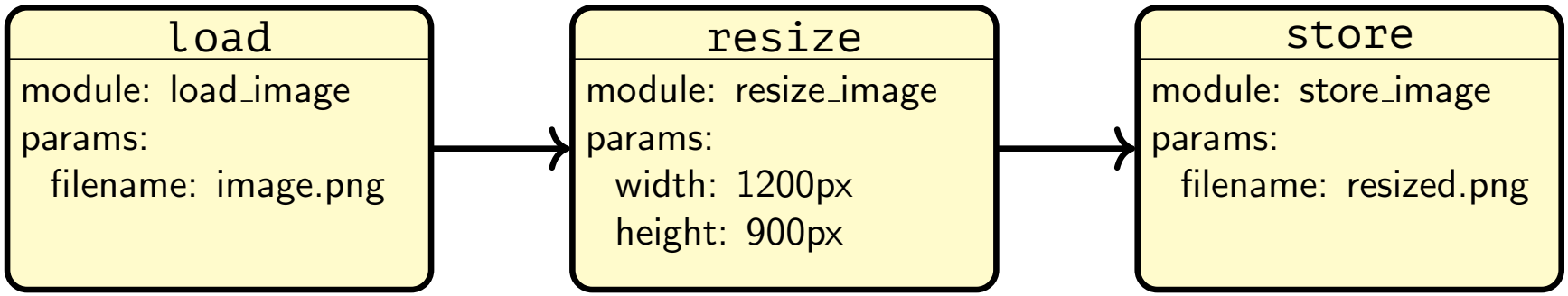


Data

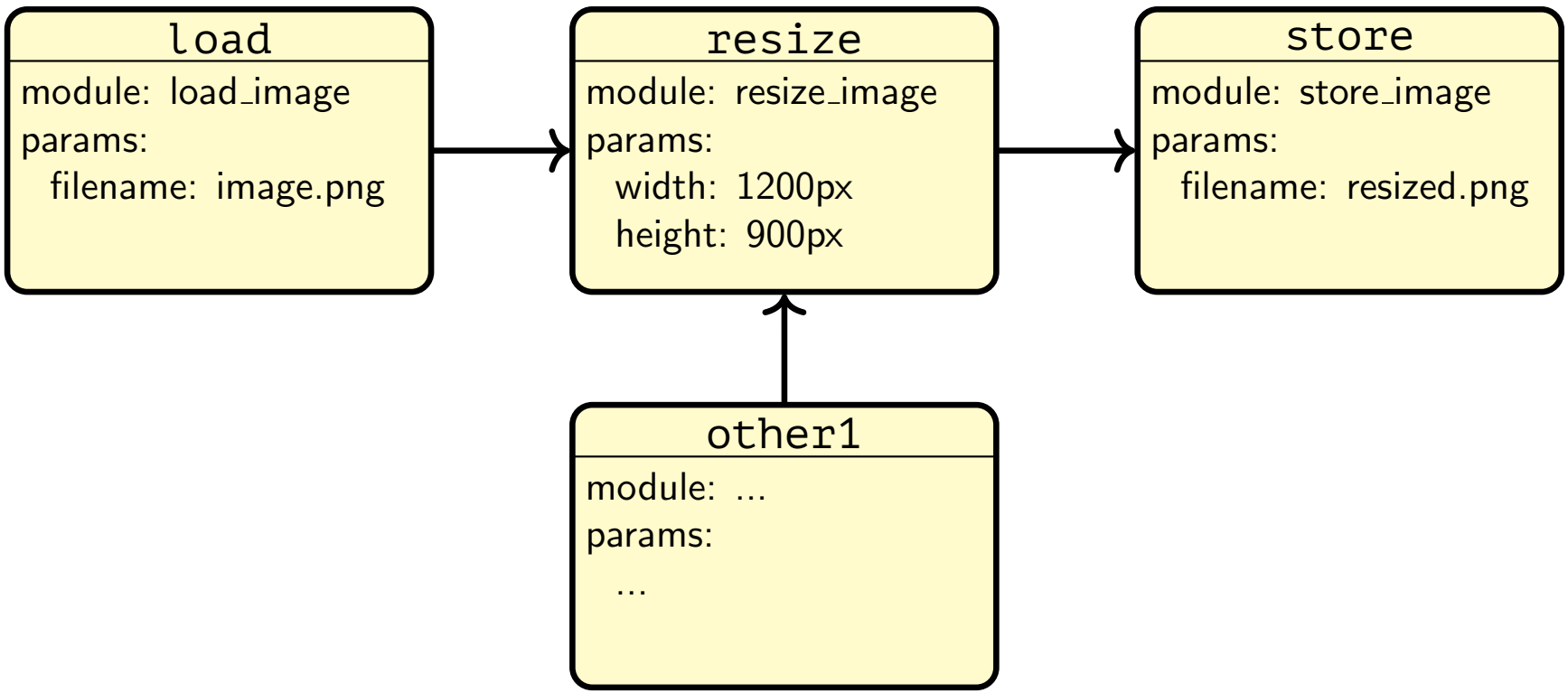


- clear interface $\rightarrow$ reuse of code in different contexts
- self-contained unit $\rightarrow$ parallelizable
- common data type implementations $\rightarrow$ data flow control
- processing chain is a dependency graph

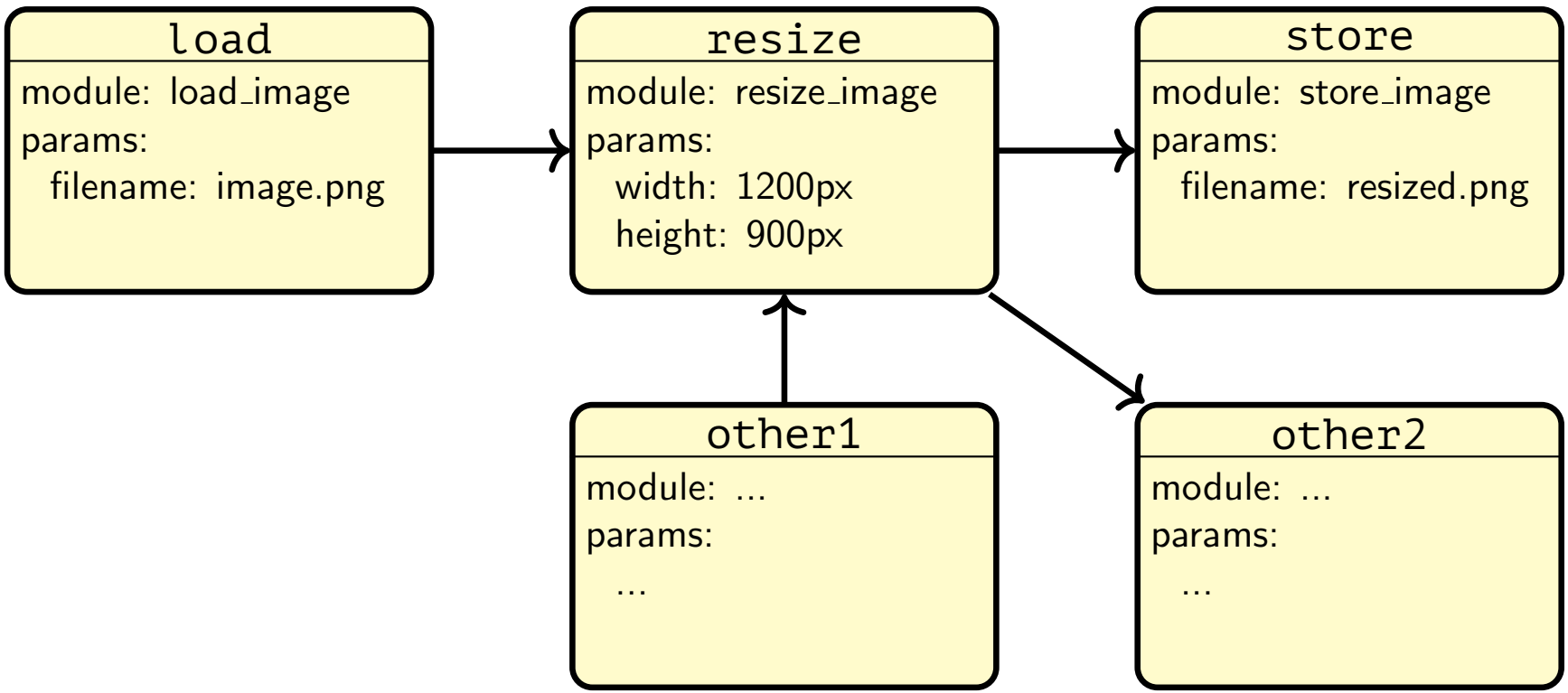
Processing Chain



Processing Chain



Processing Chain



Implementation of a Module

```
1 #include <uipf/data.hpp>
2 #include <uipf/data/opencv.hpp>
3
4 using namespace uipf;
5
6 #define UIPF_MODULE_ID "opencv.imgproc.resize"
7 #define UIPF_MODULE_NAME "Resize Image"
8 #define UIPF_MODULE_CATEGORY "opencv"
9 #define UIPF_MODULE_CLASS OpenCVResizeImage
  // ...
```

Implementation of a Module

```
1 #include <uipf/data.hpp>
2 #include <uipf/data/opencv.hpp>
3
4 using namespace uipf;
5
6 #define UIPF_MODULE_ID "opencv.imgproc.resize"
7 #define UIPF_MODULE_NAME "Resize Image"
8 #define UIPF_MODULE_CATEGORY "opencv"
9 #define UIPF_MODULE_CLASS OpenCVResizeImage
  // ...
```

Implementation of a Module

```
1 #include <uipf/data.hpp>
2 #include <uipf/data/opencv.hpp>
3
4 using namespace uipf;
5
6 #define UIPF_MODULE_ID "opencv.imgproc.resize"
7 #define UIPF_MODULE_NAME "Resize Image"
8 #define UIPF_MODULE_CATEGORY "opencv"
9 #define UIPF_MODULE_CLASS OpenCVResizeImage
  // ...
```

Implementation of a Module

```
11 #define UIPF_MODULE_INPUTS \
12   {"image", DataDescription(data::OpenCVMat::id(), \
13             "the input image.")}
14
15 #define UIPF_MODULE_OUTPUTS \
16   {"image", DataDescription(data::OpenCVMat::id(), \
17             "the resized image.")}
18
19 #define UIPF_MODULE_PARAMS \
20   {"width",  ParamDescription("new width.") }, \
21   {"height", ParamDescription("new height.") }
22
23 #include <uipf/Module.hpp>
24
25 void OpenCVResizeImage::run() {
26   // Module Implementation goes here.
27 }
```

Implementation of a Module

```
11 #define UIPF_MODULE_INPUTS \
12   {"image", DataDescription(data::OpenCVMat::id(), \
13             "the input image.")}
14
15 #define UIPF_MODULE_OUTPUTS \
16   {"image", DataDescription(data::OpenCVMat::id(), \
17             "the resized image.")}
18
19 #define UIPF_MODULE_PARAMS \
20   {"width",  ParamDescription("new width.") }, \
21   {"height", ParamDescription("new height.") }
22
23 #include <uipf/Module.hpp>
24
25 void OpenCVResizeImage::run() {
26   // Module Implementation goes here.
27 }
```

Implementation of a Module

```
11 #define UIPF_MODULE_INPUTS \
12   {"image", DataDescription(data::OpenCVMat::id(), \
13             "the input image.")}
14
15 #define UIPF_MODULE_OUTPUTS \
16   {"image", DataDescription(data::OpenCVMat::id(), \
17             "the resized image.")}
18
19 #define UIPF_MODULE_PARAMS \
20   {"width",  ParamDescription("new width.") }, \
21   {"height", ParamDescription("new height.") }
22
23 #include <uipf/Module.hpp>
24
25 void OpenCVResizeImage::run() {
26   // Module Implementation goes here.
27 }
```

Implementation of a Module

```
11 #define UIPF_MODULE_INPUTS \
12   {"image", DataDescription(data::OpenCVMat::id(), \
13             "the input image.")}
14
15 #define UIPF_MODULE_OUTPUTS \
16   {"image", DataDescription(data::OpenCVMat::id(), \
17             "the resized image.")}
18
19 #define UIPF_MODULE_PARAMS \
20   {"width",  ParamDescription("new width.") }, \
21   {"height", ParamDescription("new height.") }
22
23 #include <uipf/Module.hpp>
24
25 void OpenCVResizeImage::run() {
26   // Module Implementation goes here.
27 }
```

Implementation of a Module

```
11 #define UIPF_MODULE_INPUTS \
12   {"image", DataDescription(data::OpenCVMat::id(), \
13             "the input image.")}
14
15 #define UIPF_MODULE_OUTPUTS \
16   {"image", DataDescription(data::OpenCVMat::id(), \
17             "the resized image.")}
18
19 #define UIPF_MODULE_PARAMS \
20   {"width",  ParamDescription("new width.") }, \
21   {"height", ParamDescription("new height.") }
22
23 #include <uipf/Module.hpp>
24
25 void OpenCVResizeImage::run() {
26   // Module Implementation goes here.
27 }
```

Implementation of a Module: Data Interface

```
1  void OpenCVResizeImage::run() {
2
3    OpenCVMat::ptr img = getInputData<OpenCVMat>("image");
4    int width = getParam<int>("width", -1);
5    int height = getParam<int>("height", -1);
6
7    // do something here
8
9    OpenCVMat::ptr newImage(new OpenCVMat(m));
10   newImage->filename = image->filename;
11   newImage->exif = image->exif;
12   setOutputData<OpenCVMat>("image", newImage);
13
14 }
```

Implementation of a Module: Data Interface

```
1  void OpenCVResizeImage::run() {
2
3    OpenCVMat::ptr img = getInputData<OpenCVMat>("image");
4    int width = getParam<int>("width", -1);
5    int height = getParam<int>("height", -1);
6
7    // do something here
8
9    OpenCVMat::ptr newImage(new OpenCVMat(m));
10   newImage->filename = image->filename;
11   newImage->exif = image->exif;
12   setOutputData<OpenCVMat>("image", newImage);
13
14 }
```

Implementation of a Module: Data Interface

```
1  void OpenCVResizeImage::run() {
2
3    OpenCVMat::ptr img = getInputData<OpenCVMat>("image");
4    int width = getParam<int>("width", -1);
5    int height = getParam<int>("height", -1);
6
7    // do something here
8
9    OpenCVMat::ptr newImage(new OpenCVMat(m));
10   newImage->filename = image->filename;
11   newImage->exif = image->exif;
12   setOutputData<OpenCVMat>("image", newImage);
13
14 }
```

Implementation of a Module: Data Interface

```
1  void OpenCVResizeImage::run() {
2
3    OpenCVMat::ptr img = getInputData<OpenCVMat>("image");
4    int width = getParam<int>("width", -1);
5    int height = getParam<int>("height", -1);
6
7    // do something here
8
9    OpenCVMat::ptr newImage(new OpenCVMat(m));
10   newImage->filename = image->filename;
11   newImage->exif = image->exif;
12   setOutputData<OpenCVMat>("image", newImage);
13
14 }
```

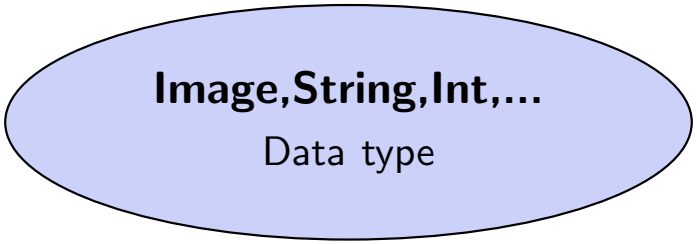
Implementation of a Module: Data Interface

```
1  void OpenCVResizeImage::run() {
2
3    OpenCVMat::ptr img = getInputData<OpenCVMat>("image");
4    int width = getParam<int>("width", -1);
5    int height = getParam<int>("height", -1);
6
7    // do something here
8
9    OpenCVMat::ptr newImage(new OpenCVMat(m));
10   newImage->filename = image->filename;
11   newImage->exif = image->exif;
12   setOutputData<OpenCVMat>("image", newImage);
13
14 }
```

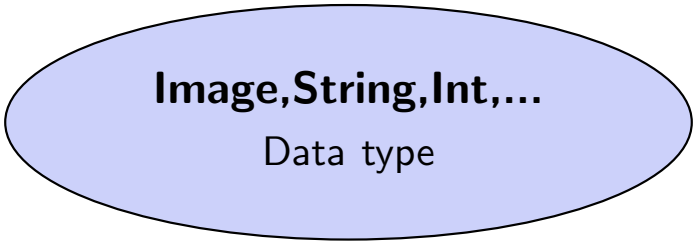
Implementation of a Module: Data Interface

```
1  void OpenCVResizeImage::run() {
2
3    OpenCVMat::ptr img = getInputData<OpenCVMat>("image");
4    int width = getParam<int>("width", -1);
5    int height = getParam<int>("height", -1);
6
7    // do something here
8
9    OpenCVMat::ptr newImage(new OpenCVMat(m));
10   newImage->filename = image->filename;
11   newImage->exif = image->exif;
12   setOutputData<OpenCVMat>("image", newImage);
13
14 }
```

Data structure implementation

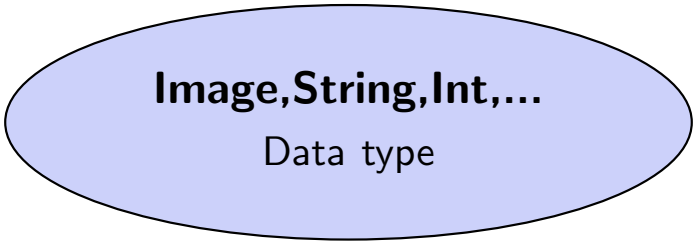


Data structure implementation



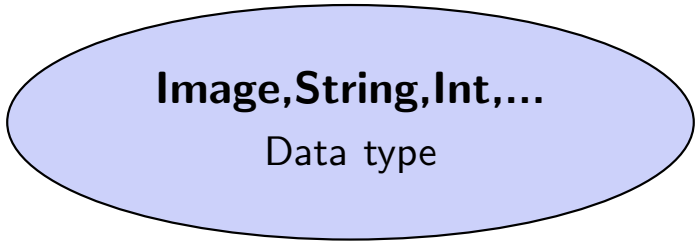
- Data structure is a container

Data structure implementation



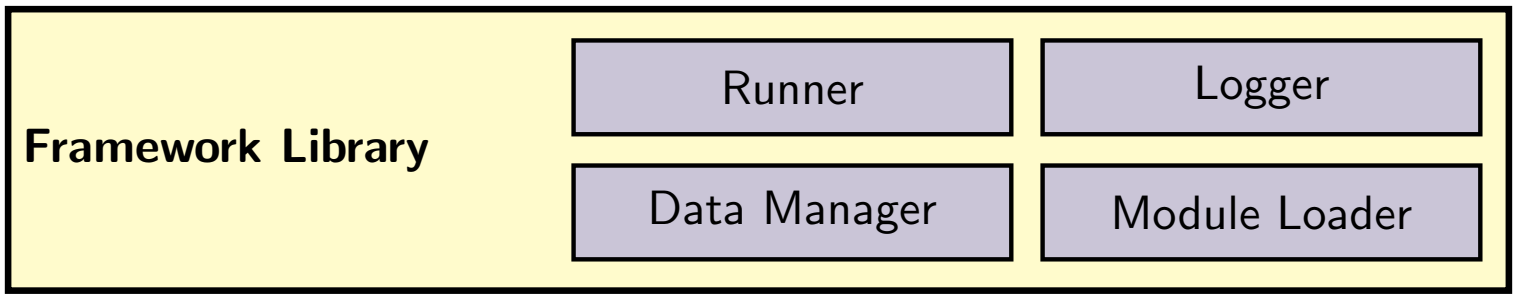
- Data structure is a container
- Interface for visualisation and serialisation

Data structure implementation

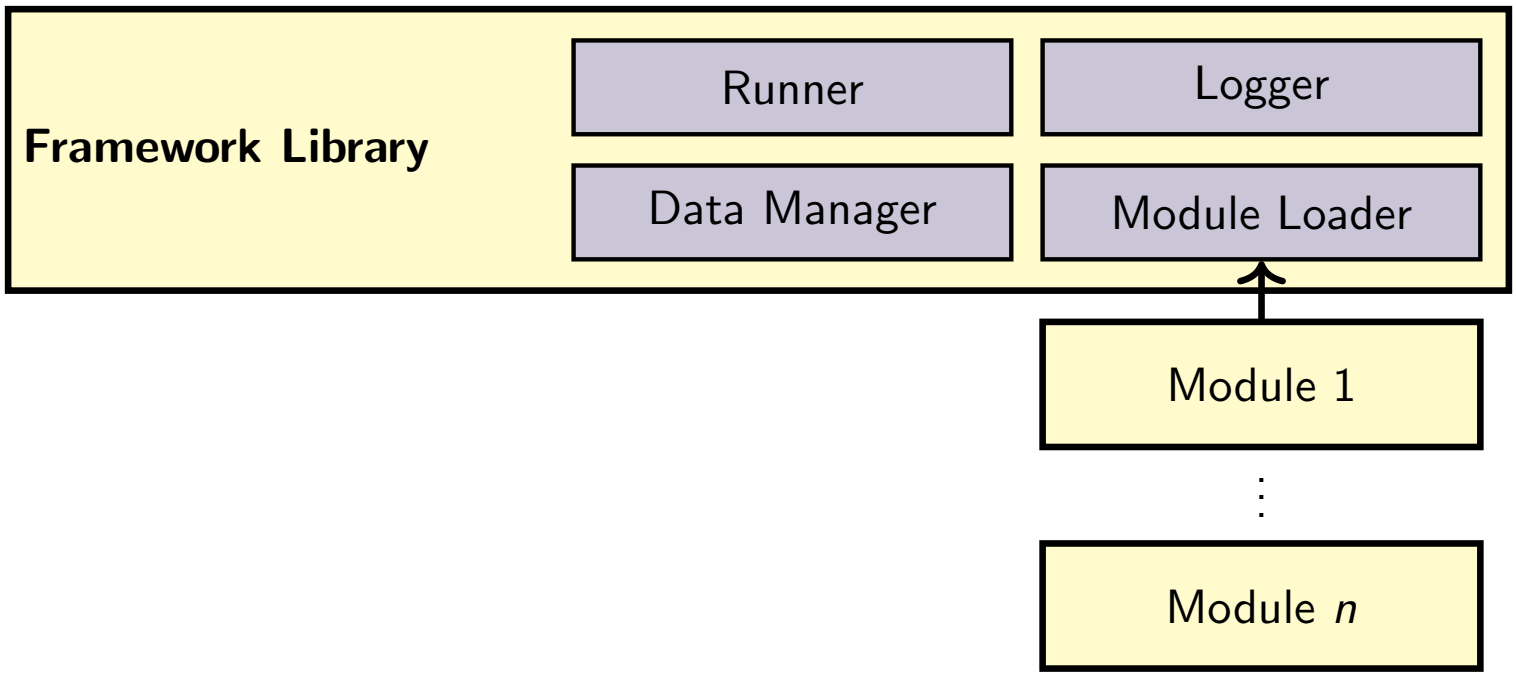


- Data structure is a container
- Interface for visualisation and serialisation
- Custom datatypes possible

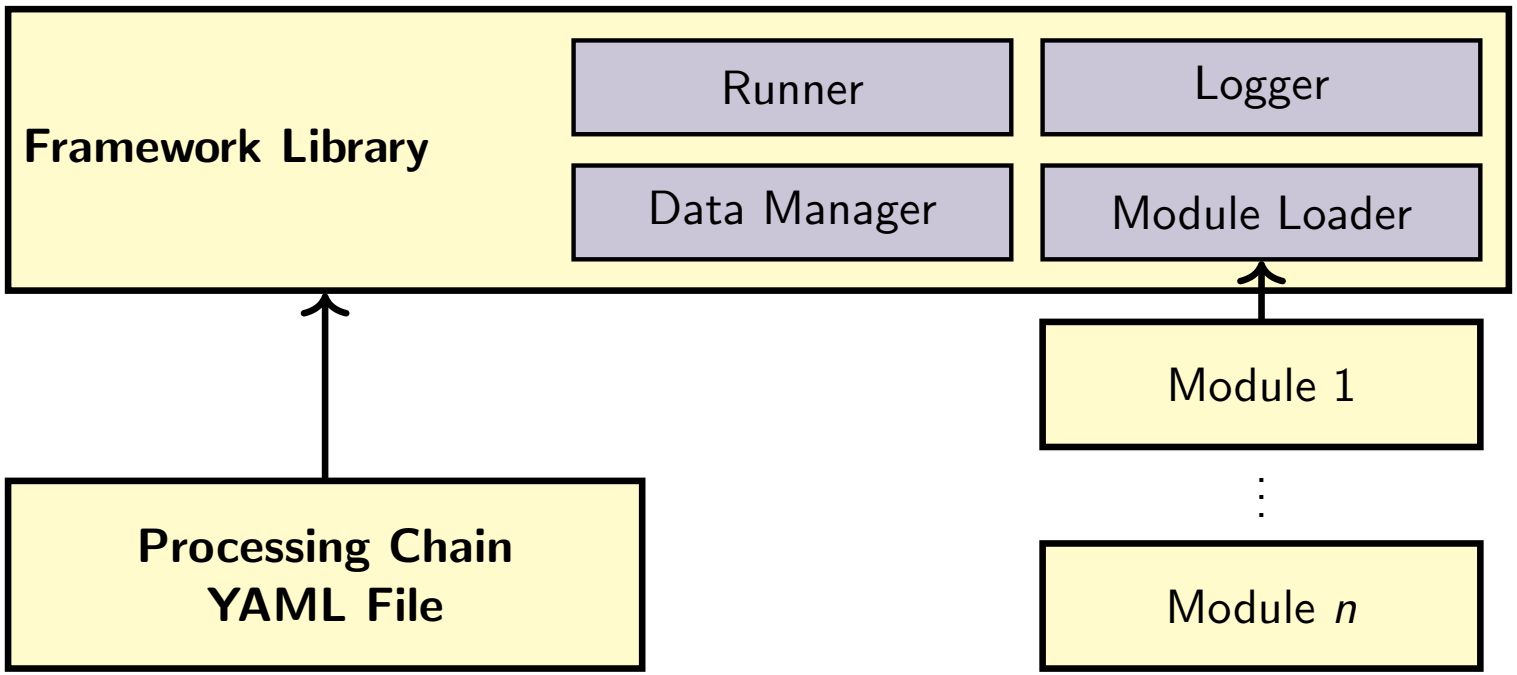
Framework overview



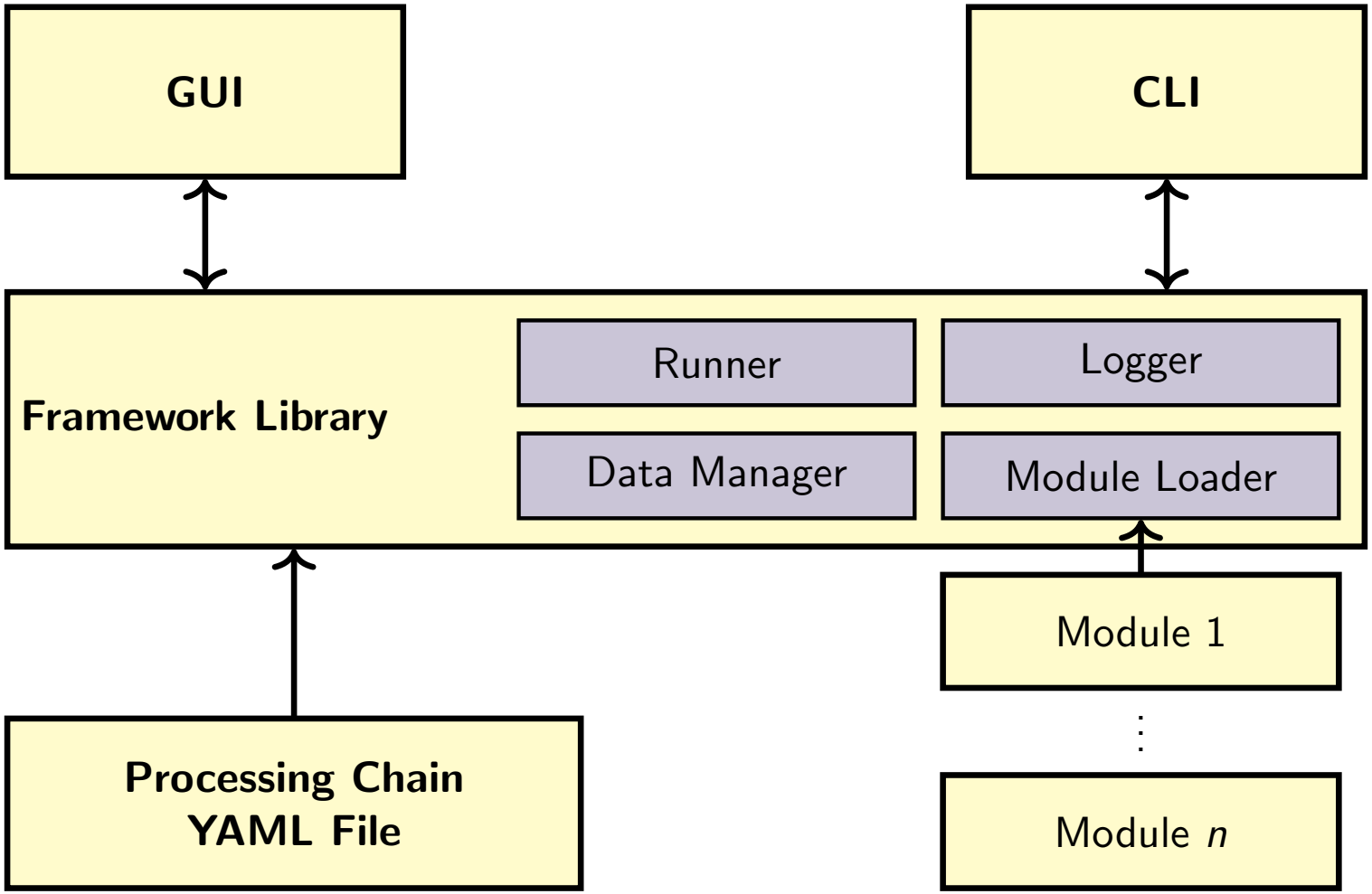
Framework overview



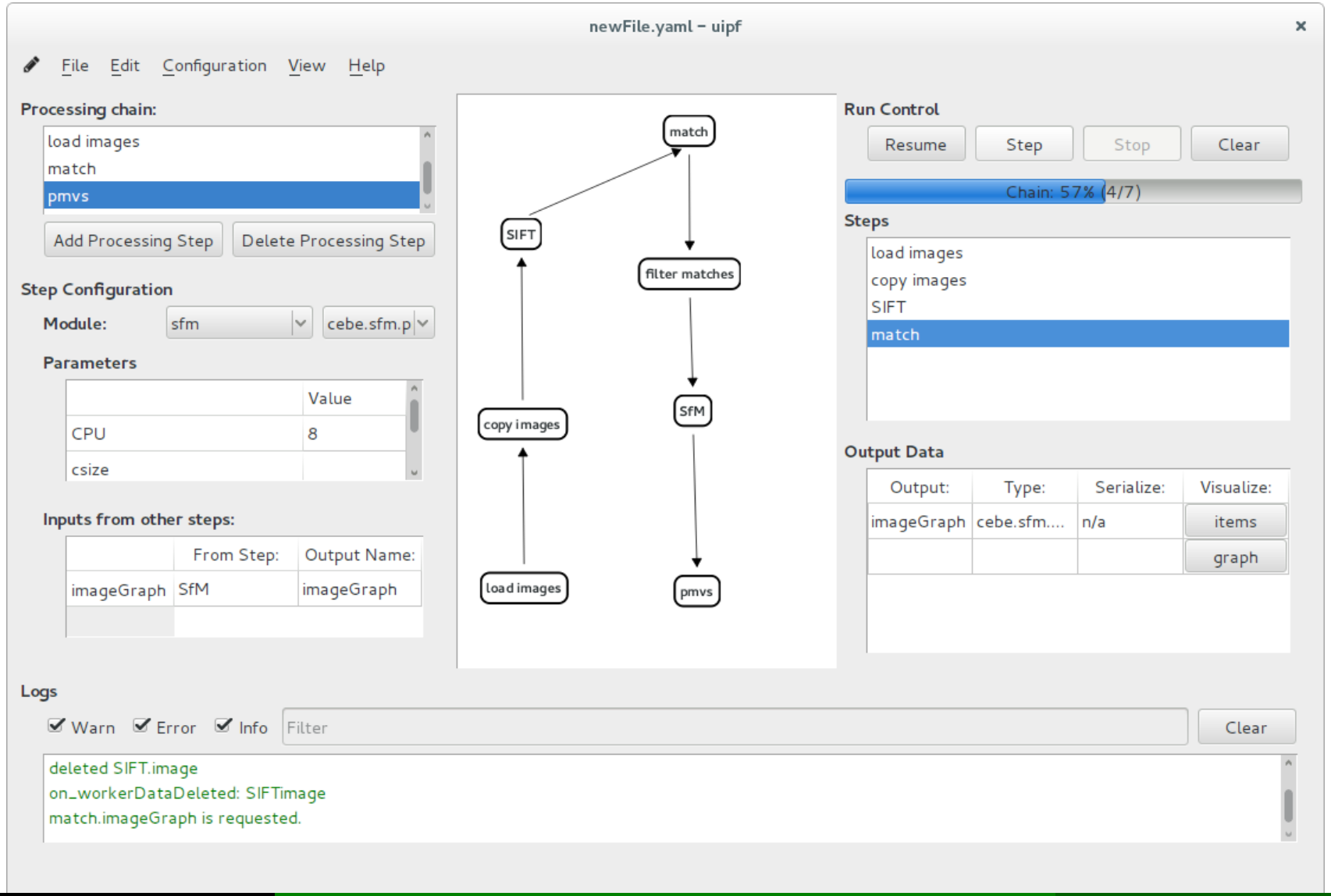
Framework overview



Framework overview



GU Interface



CL Interface

```
# run a processing chain
uipf -c processing-chain.yaml
```

CL Interface

```
# run a processing chain
uipf -c processing-chain.yaml

# run a single module
uipf uipfsfm.keypoint -i image.jpg -o points.txt
```

CL Interface

```
# run a processing chain
uipf -c processing-chain.yaml

# run a single module
uipf uipfsfm.keypoint -i image.jpg -o points.txt

# list and document modules
uipf -l
uipf --info uipfsfm.keypoint.sift
```

Table of Contents

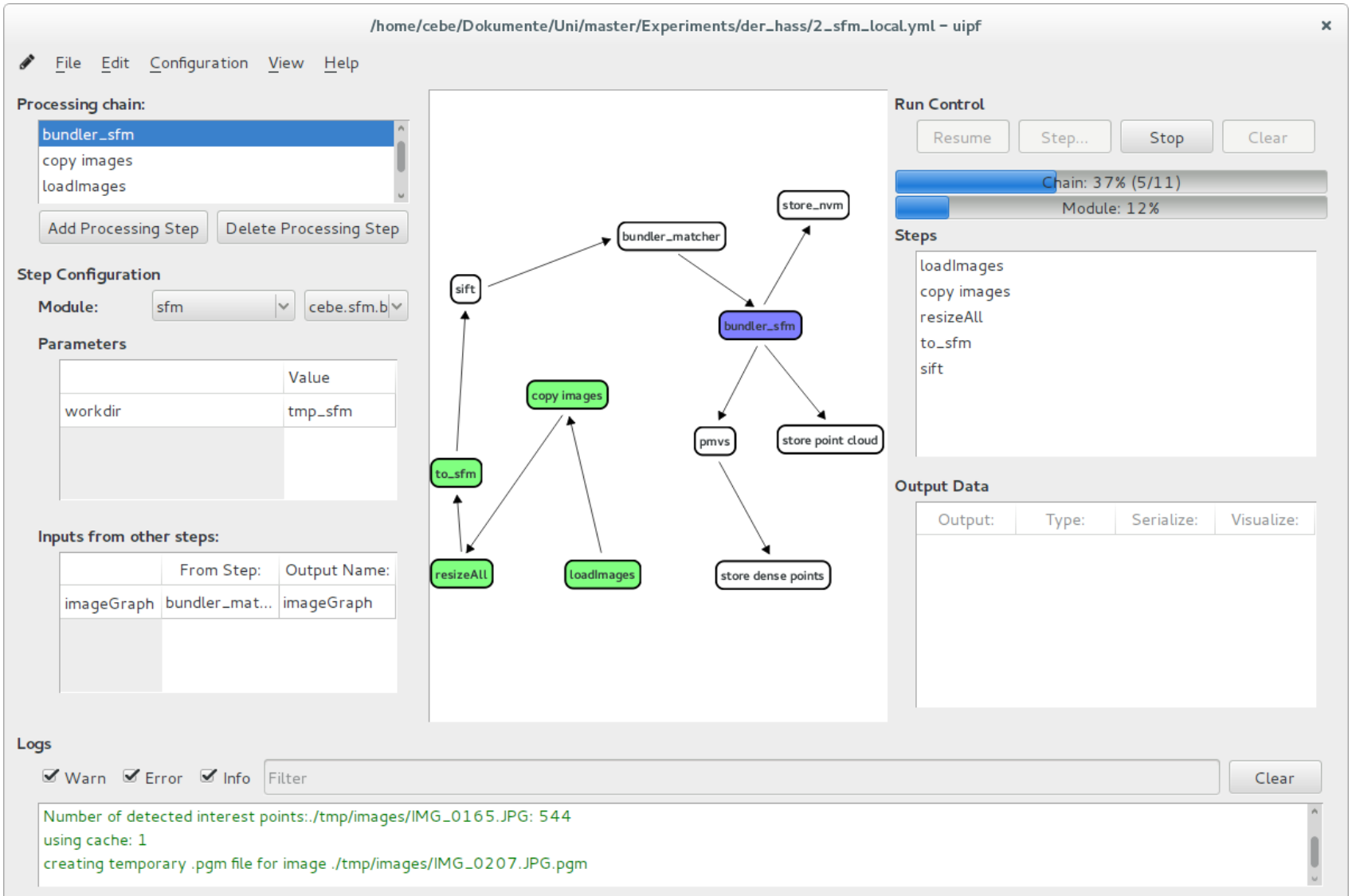
Dataset

79 images
$2592 \times 1728px$

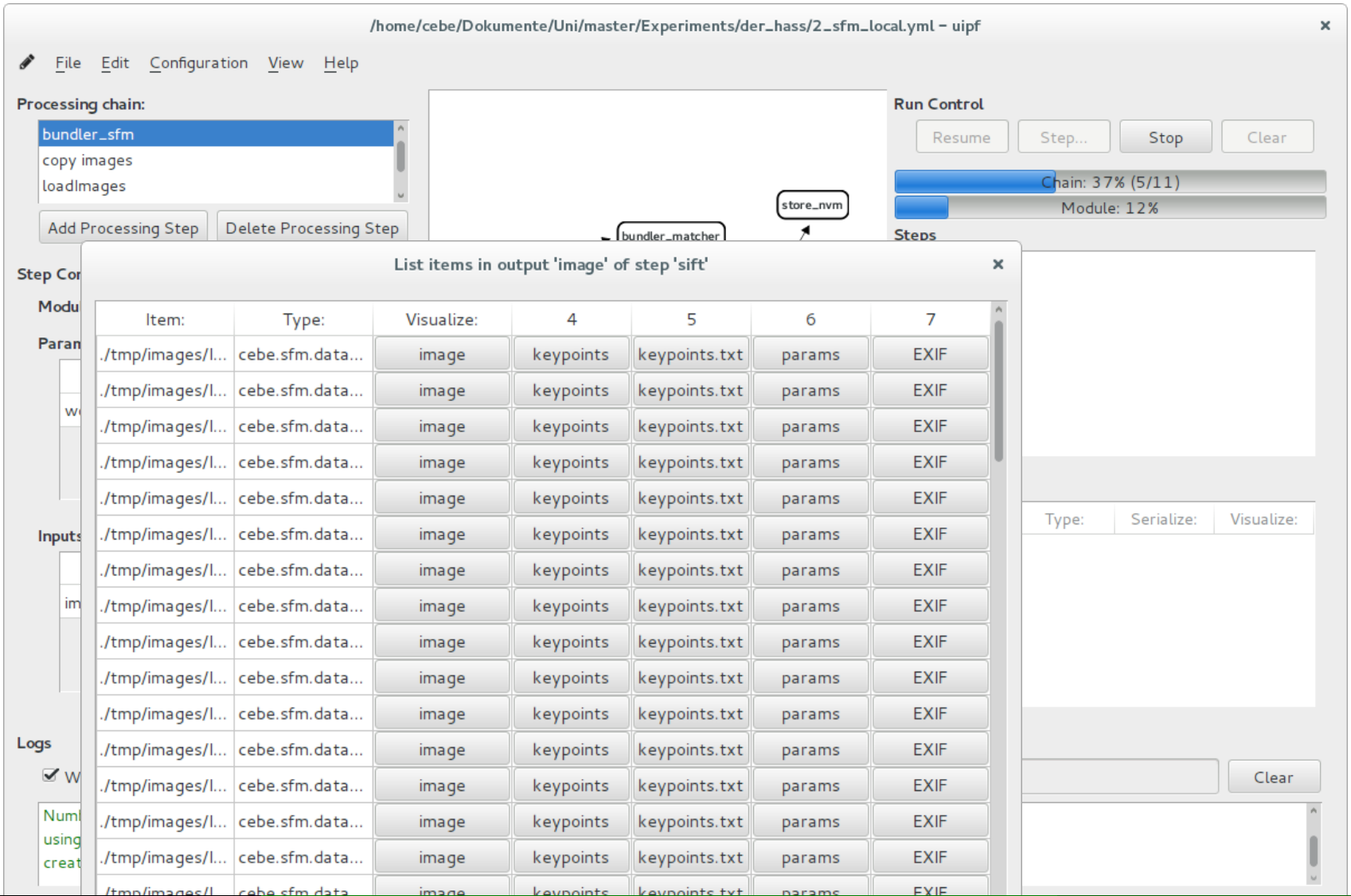
"Der Hass" dataset by [Fuhrmann 2014]

Simon Fuhrmann, Fabian Langguth and Michael Goesele. MVE - A Multi-View Reconstruction Environment. In GCH, pages 1118, 2014.

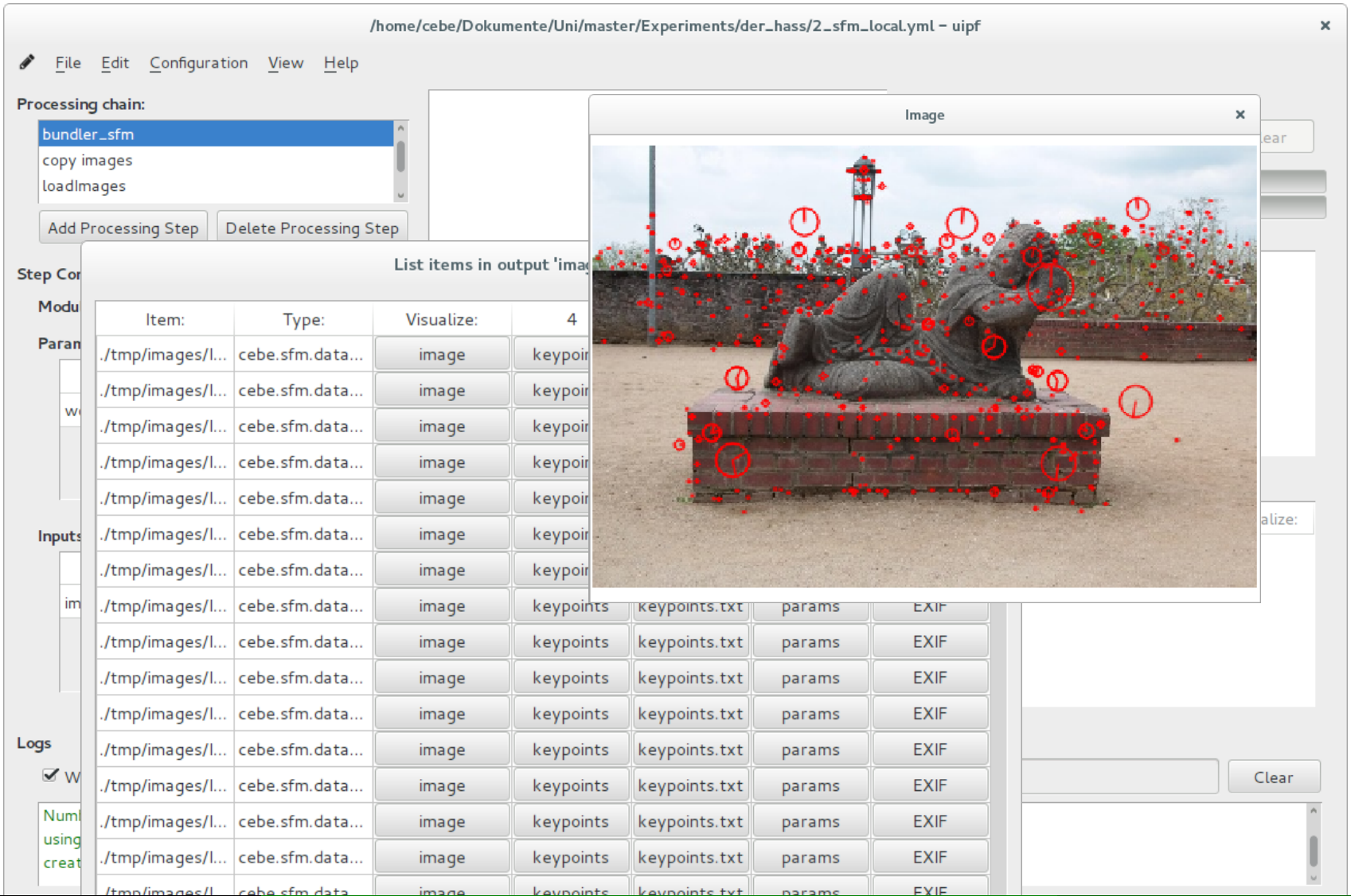
Processing Chain



Processing Chain



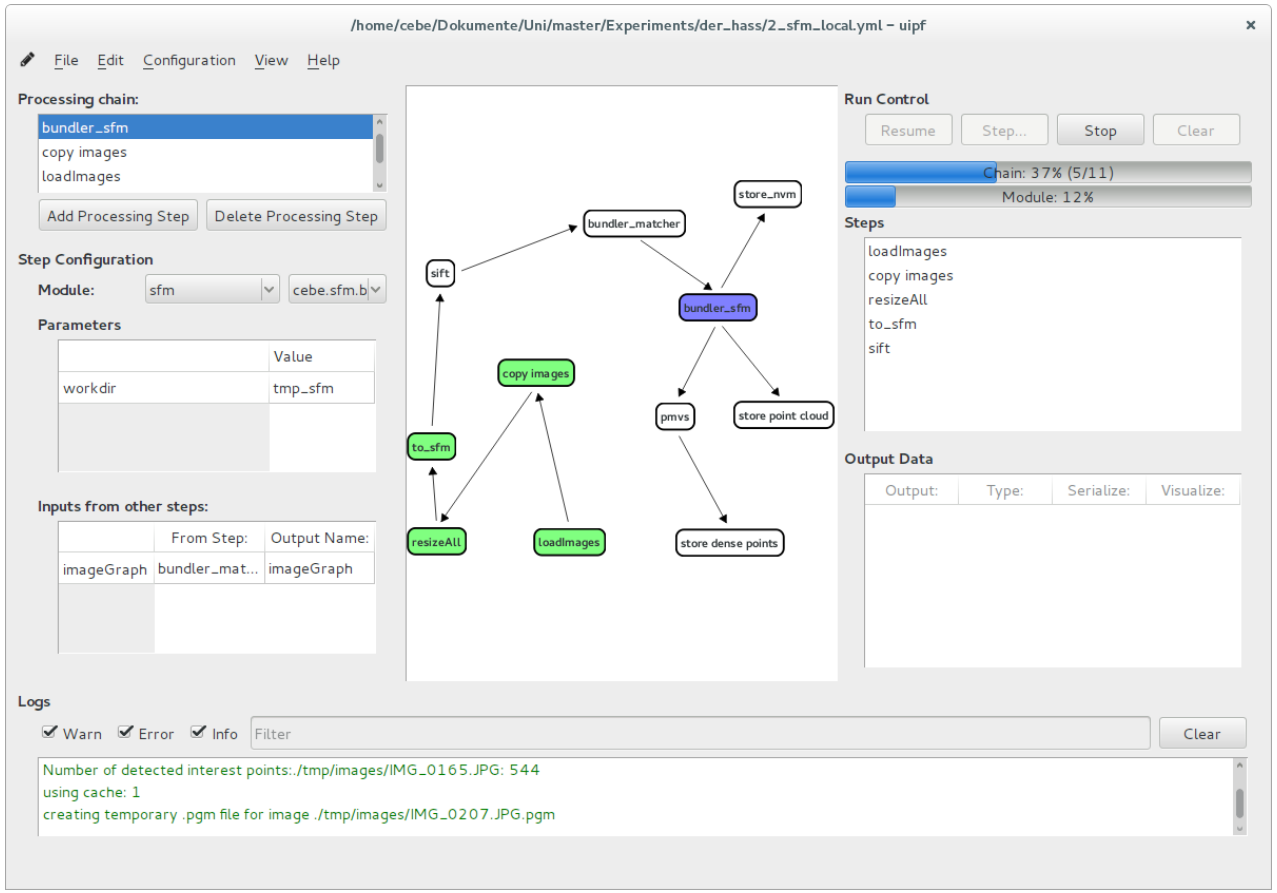
Processing Chain



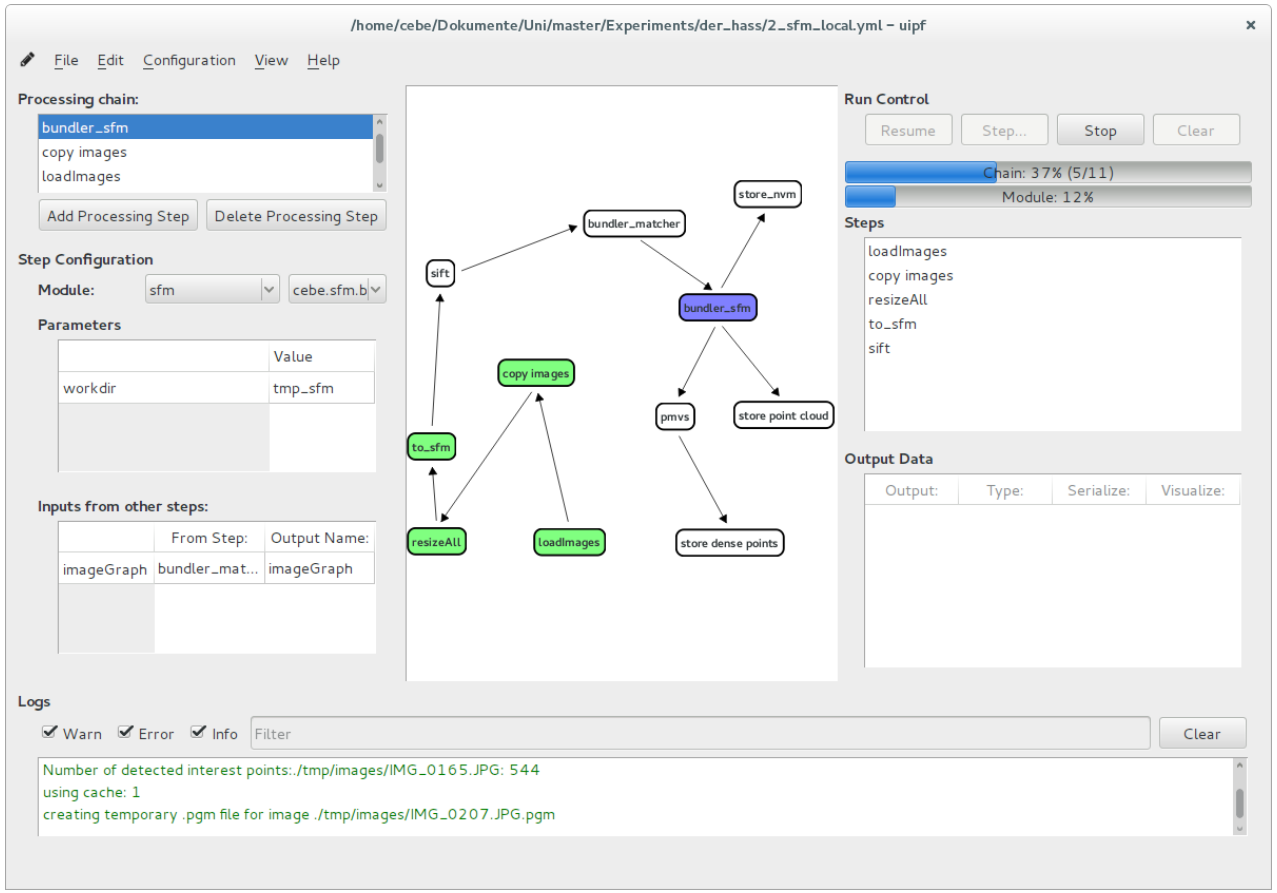
Sparse Point Cloud

Mesh

Conclusion


/home/cebe/Dokumente/Uni/master/Experiments/der_hass/2_sfm_local.yml – uipf
File
Edit
Configuration
View
Help
Processing chain:
bundler_sfm
copy images
loadImages
Add Processing Step
Delete Processing Step
Step Configuration
Module:
sfm
cebe.sfm.b
Parameters
Value
workdir
tmp_sfm
Inputs from other steps:
From Step:
Output Name:
imageGraph
bundler_mat...
imageGraph
sift
bundler_matcher
store_nvm
bundler_sfm
copy images
pmvs
store point cloud
to_sfm
resizeAll
loadImages
store dense points
Run Control
Resume
Step...
Stop
Clear
Chain: 37% (5/11)
Module: 12%
Steps
loadImages
copy images
resizeAll
to_sfm
sift
Output Data
Output:
Type:
Serialize:
Visualize:
Logs
Warn
Error
Info
Filter
Clear
Number of detected interest points:./tmp/images/IMG_0165.JPG: 544
using cache: 1
creating temporary .pgm file for image ./tmp/images/IMG_0207.JPG.pgm

Conclusion



https://github.com/uipf/uipf