# Multi-view stereo matching: intro

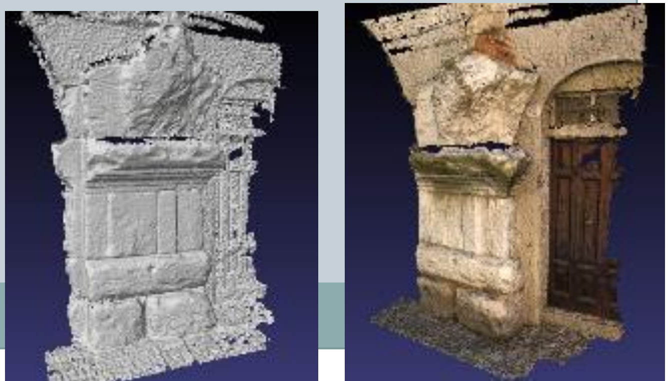
VISUAL COMPUTING LAB
ISTI-CNR PISA, ITALY

# 3D from Images

Recap:

we want to have a fully automatic dense photogrammetry pipeline, starting from uncalibrated images to create a 3D model

i.e. Having the PC doing automatically both processing steps: camera calibration&oriantation and dense stereo match



# Calibration and Orientation step

We know calibration and orientation can be obtained by having a set of photo-to-photo correspondences.

We do need a method to extract correspondences from photos, and possibly a method that scales well with the number of photos (remember we said the manual correspondence picking does not scale well)

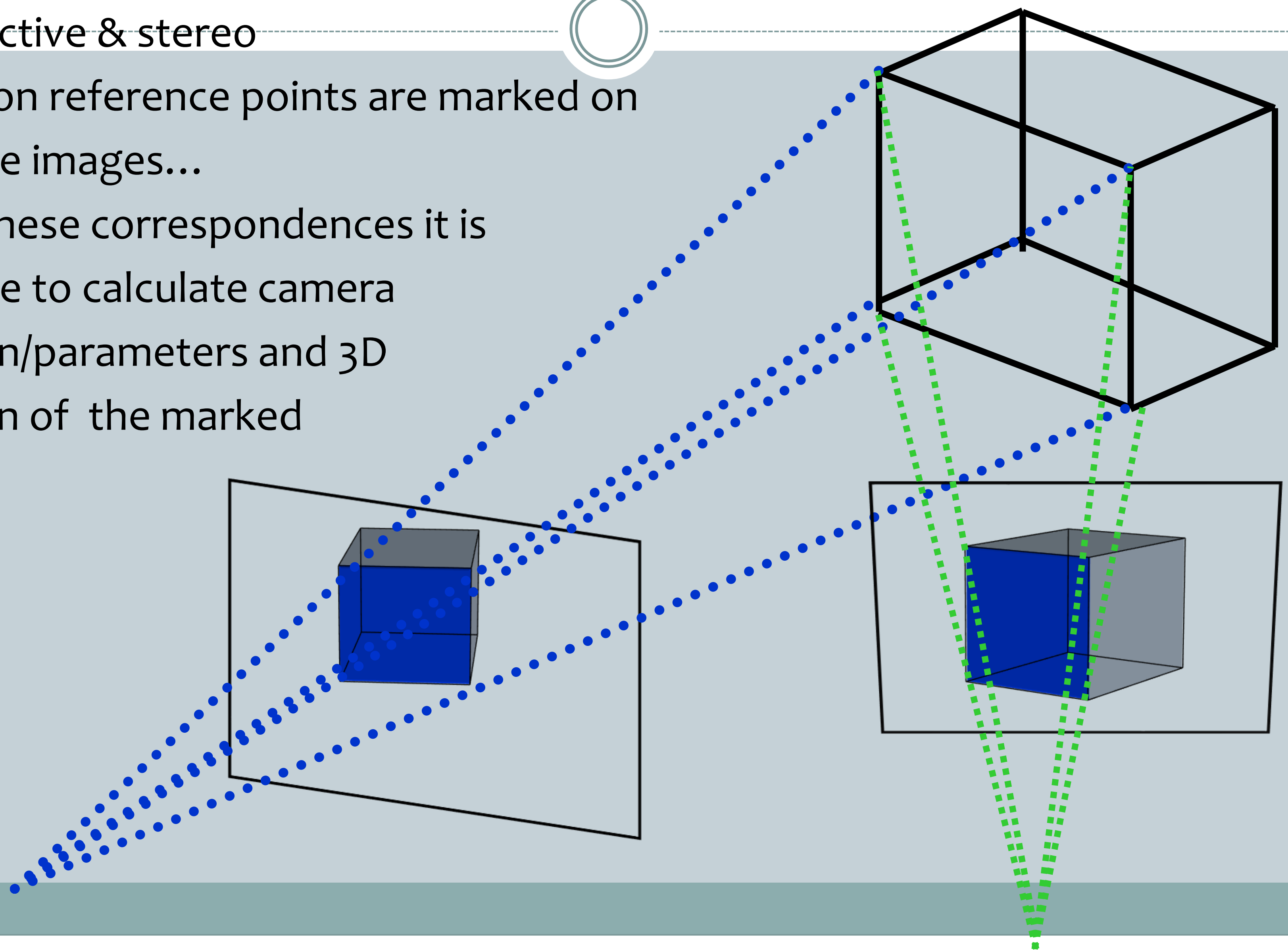
Plese note: if the camera intrinsics are known (pre-calibration) or if the photos are undistorted, this steps works much better

# Remind… photogrammetry

Perspective & stereo

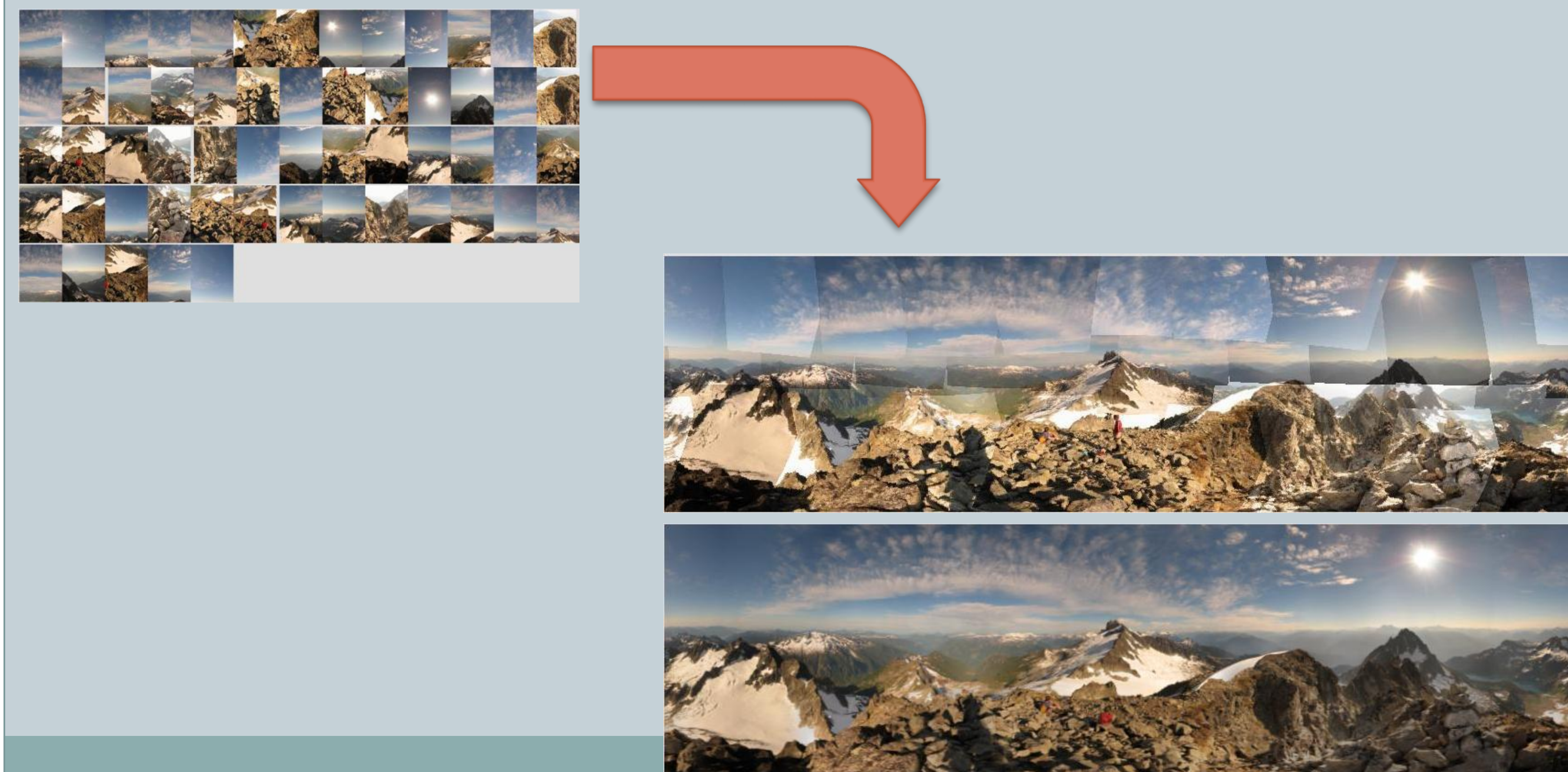
Common reference points are marked on multiple images…

From these correspondences it is possible to calculate camera position/parameters and 3D location of  the marked points



# Photo match & stitch

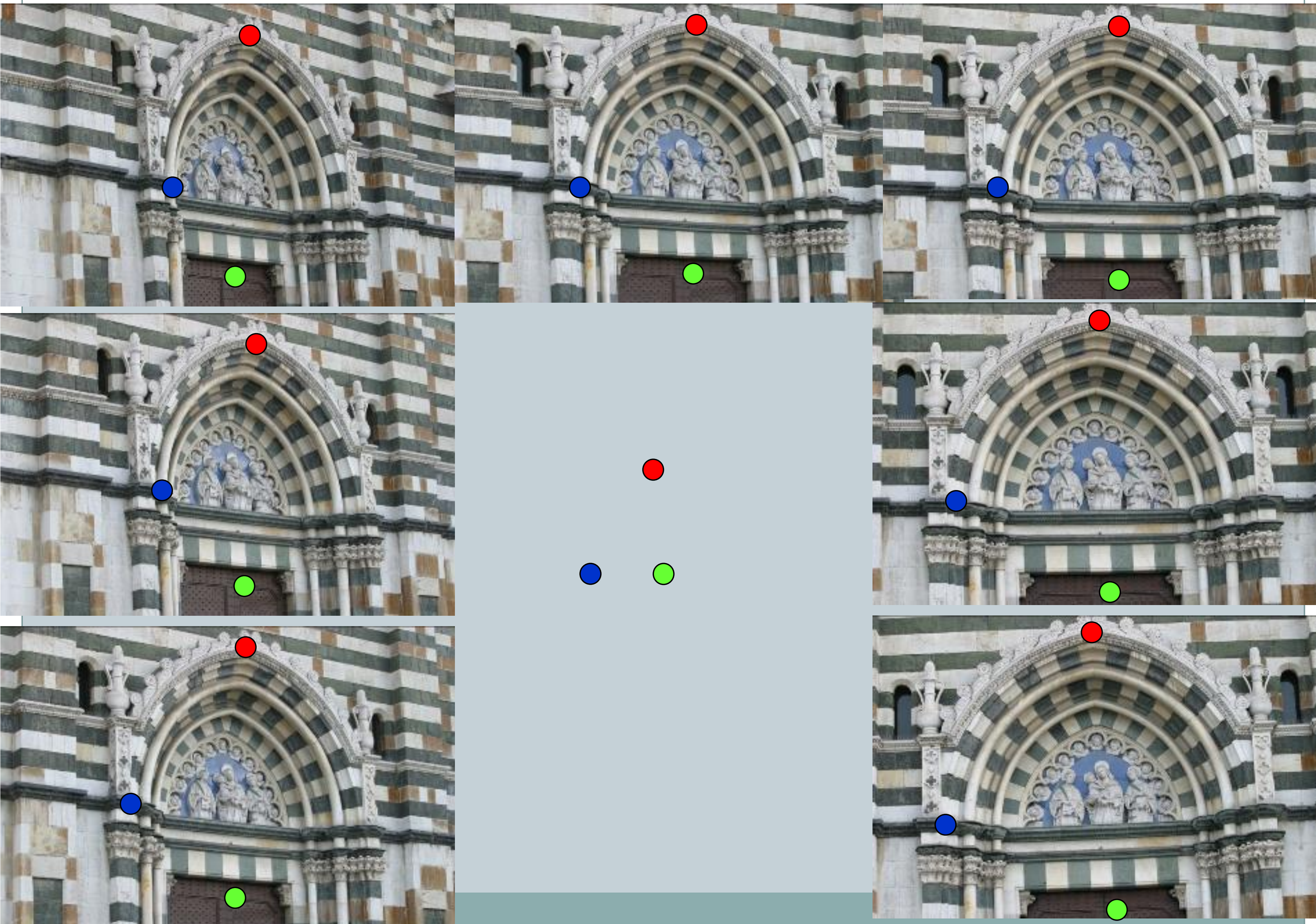
Maybe outside of 3D reconstruction, you have used similar methods

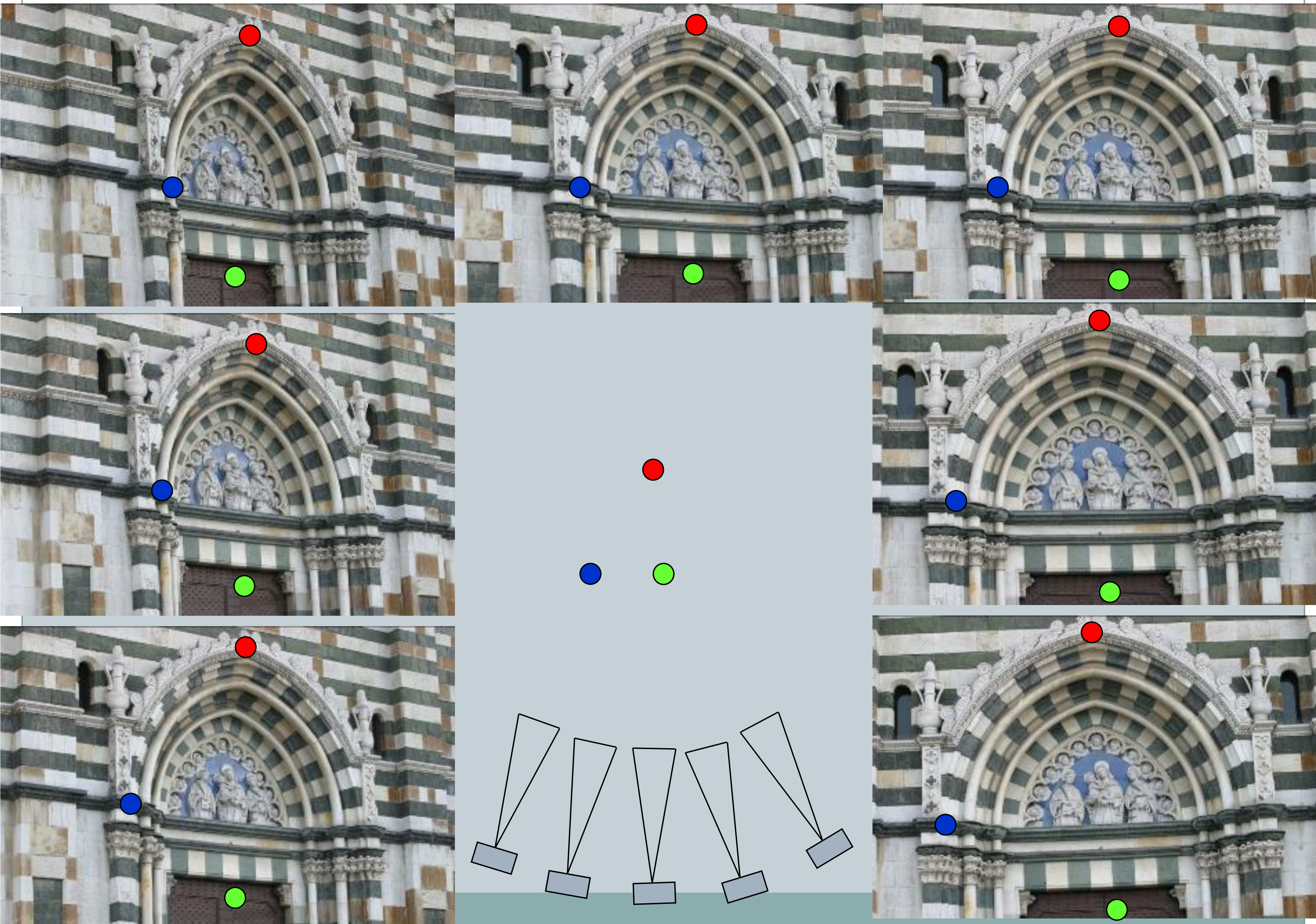


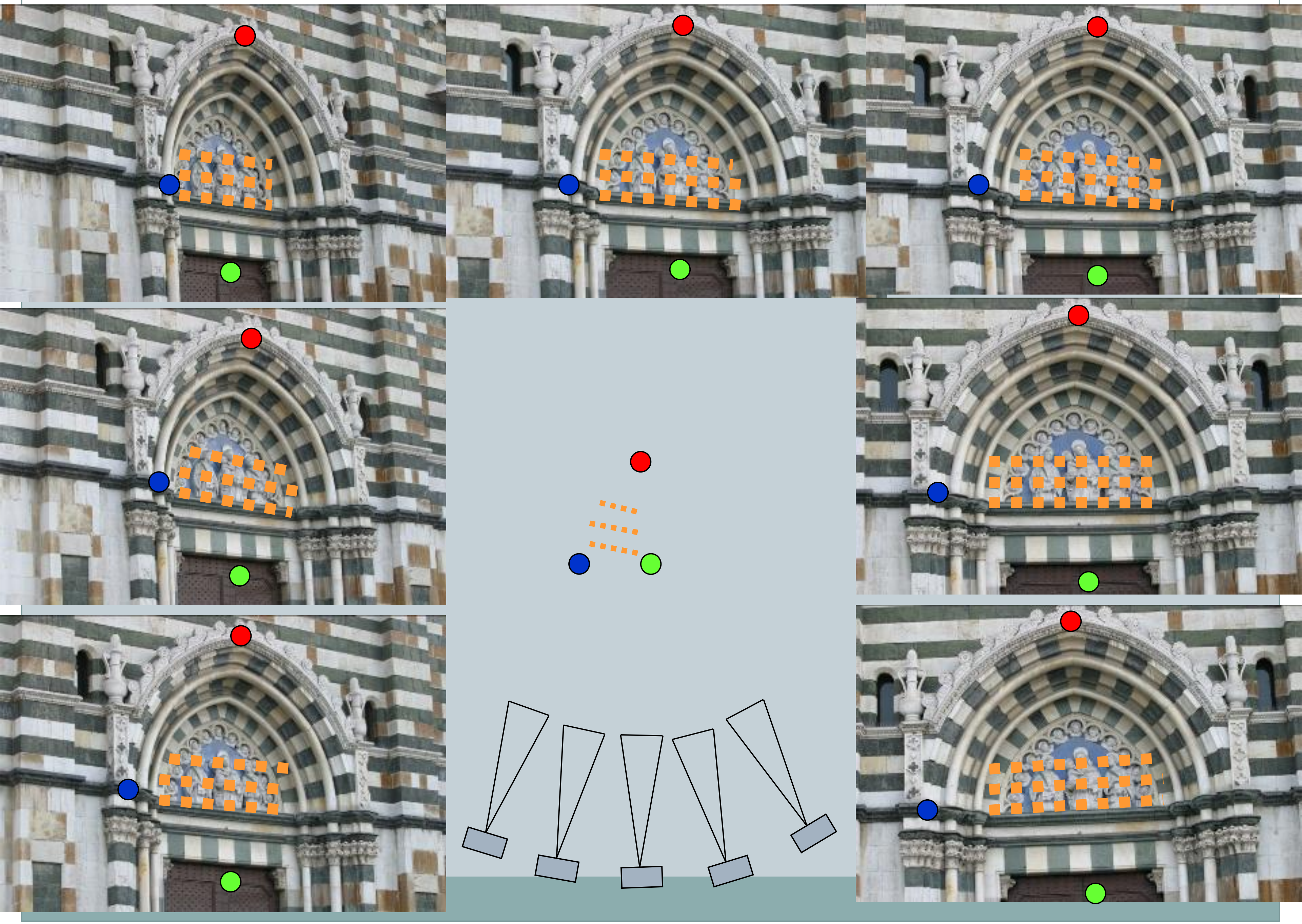
# Working principle

All the existing tools follow the same scheme:

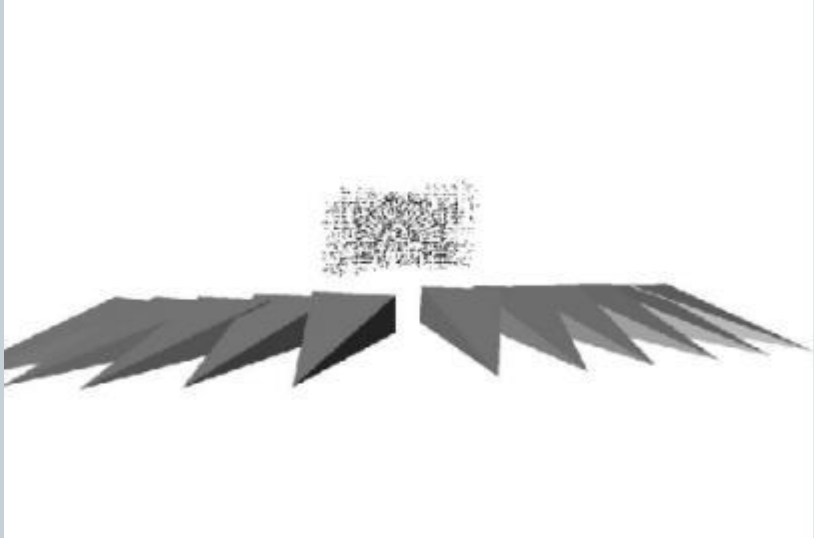
- Using euristincs and local analysis, find some *salient* points in the input images (FEATURE EXTRACTION).
- Match the salient points across images, determining overlap between images (MATCHING).
- From the matched points, determine position, focal lenght and distortion of the camera at the time of the shot (CALIBRATION).
- Using the computed cameras, perform a *dense* match trying to determine 3D coordinates for all pixels (DENSE MATCHING).

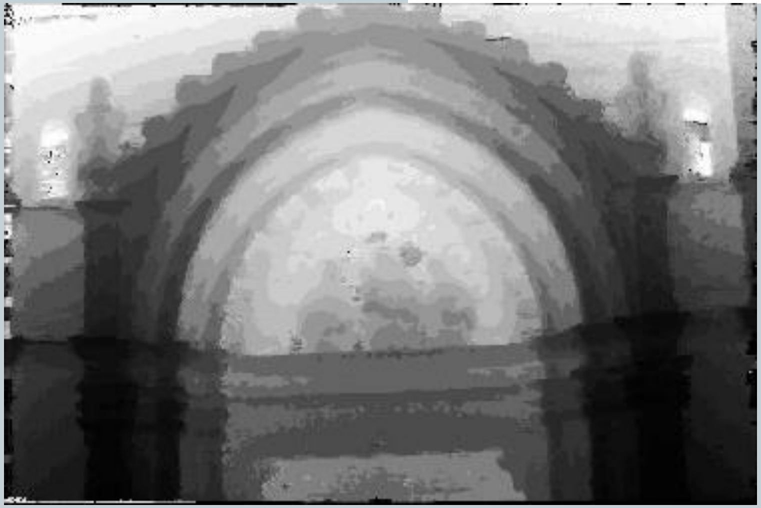


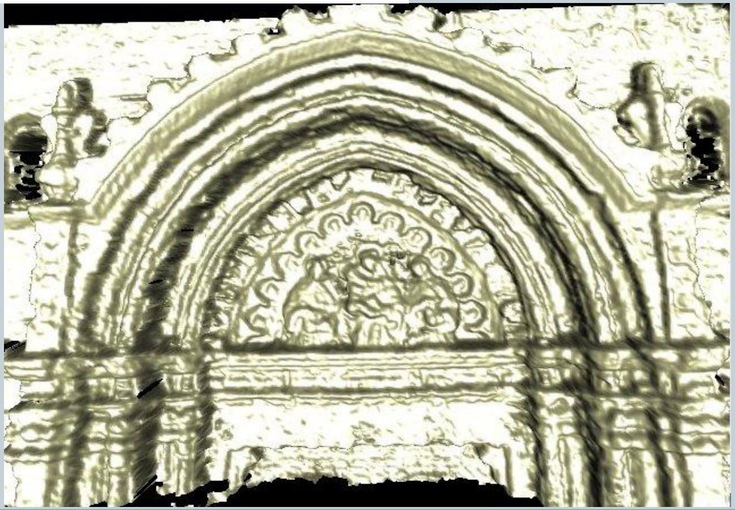




# Working principle







# SIFT - SURF

SIFT: Scale Invariant Feature Transform

SURF: Speeded Up Robust Feature

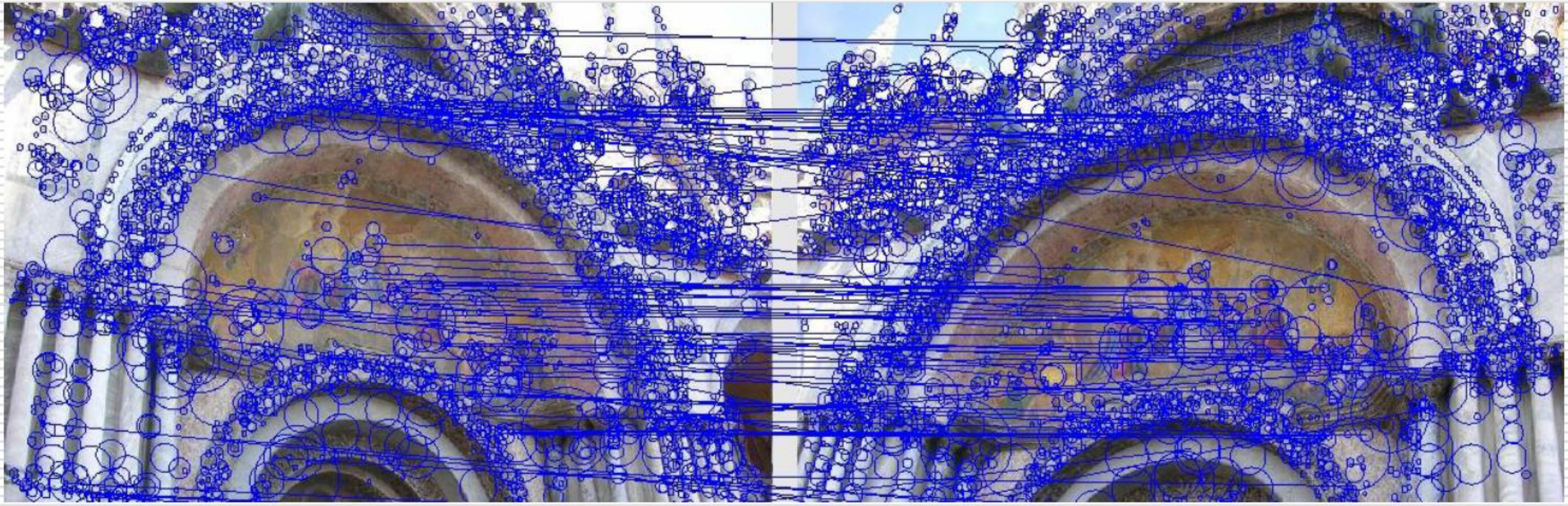
Local descriptors of an image "feature points", they are used to efficiently determine salient points and match them across images.

Many variants, and really diverse is the scheme for multi-image matching from one software to another.

# Bag of feature

A lot more correspondence points are used, with respect to manual photogrammetry…

Computer point match is less accurate than human, more points->error reduction. More points-> coherence check (ransac)



# All for one, one for all

Another component used in these tools is:

**Bundle Adjustement**

Cameras are determined independently, using the detected corresponences, and a global optimization step is often necessary to ensure a good fitting.

Many ready-to-use libraries for bundle adjustement exists...

# A problem of SCALE

All these tools have a problem in common: the returned geometry is at an unknown scale... every proportion is correct, it is only that the measure unit is unknown. This is because nothing is known about the scene and the camera (you may have been taken a photo of a car or of a car model).

How to solve this? You need a measurement taken on the real object and the corresponding measure from the computed 3D model to calculate the scale factor!

Most tools have a way to calculate/specify this scaling factor at the time of model creation... in any case, it will always be possible to apply a scaling factor to the whole result :).

# A problem of SCALE

This issue is common also to pure Photogrammetry tools!

Photogrammetry software has inbuilt tools to apply scale, with multiple measurements and residual error calculation.

If you are using markers of known size/pattern size, or some metric details of the scene are known (like the offset of the camera in the MENCI tool), the scale is calculated automatically.

# Why not before?

Well, some of the algorithmic basis was already known, bute there were some missing pieces:

- Hardware resources and parallelization
- A robust, scale-independent feature extractor (then SIFT came...)
- Dense matching at its best

# A plethora of tools

TRy 3D from images is easy, you need a camera and one of the many software tools...

A lot of free tools, often a "toolchain" of existing tools.

Some semi-free or very cheap software.

Many commercial implementations, sometimes bunbled to custom-made devices.

# Online - Offline

Computing 3D reconstruction from photos is a cumbersome task, computationally. A reconstruction may take hours, or even more than one day...

For this reason some tools are implemented as web-services.

The data is sent to a remote server, ad you receive the results

- ☺ Good performances, remote code is regularly updated
- ☹ You need network access, you have to send away your data, limited control on parameters

# Online - Offline

Some other solutions are essentially “local”.

- ☺ Full control on parameters and on “ad-hoc” strategies
- ☹ Resources and time needed.

# Regard3D

Free tool that implements a couple of the pipelines which have been made available to the community.

Multiplatform and quite well developed.

# VisualSFM

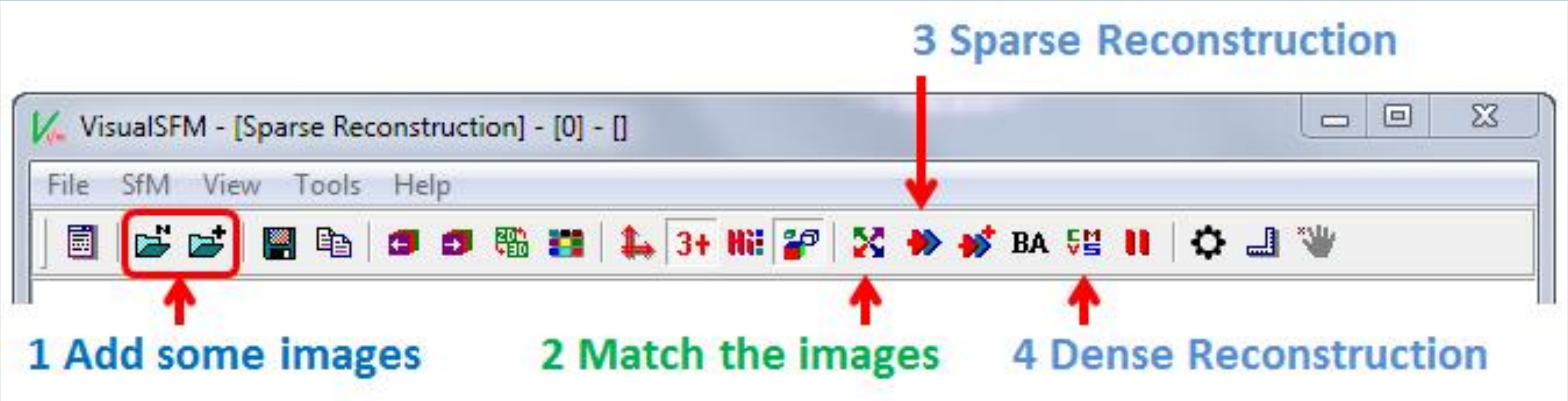
Free tool (not opesource, but some components are opensource).

Grown a lot in usability and performances...

Completely local. Easy to install (under windows) and use.

Good result at no cost...

**http://ccwu.me/vsfm/**



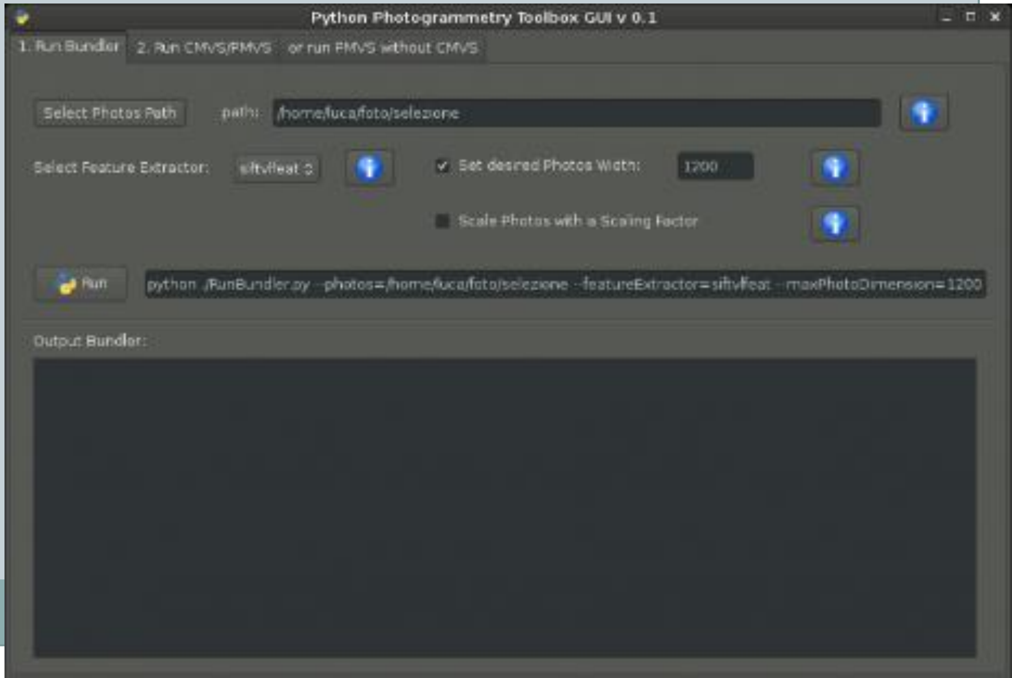
# Python Photogrammetry Toolbox

Developed by Arc-Team, open source and free, for Debian and Win (32 and 64bit)

- http://www.arc-team.com/

Good: completely local, interface, control on parameters, video tutorial

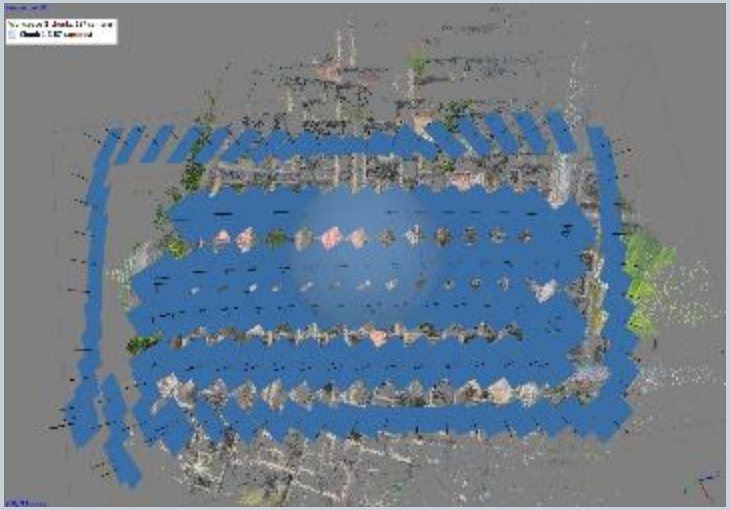
Bad: completely local, a bit tricky to install…

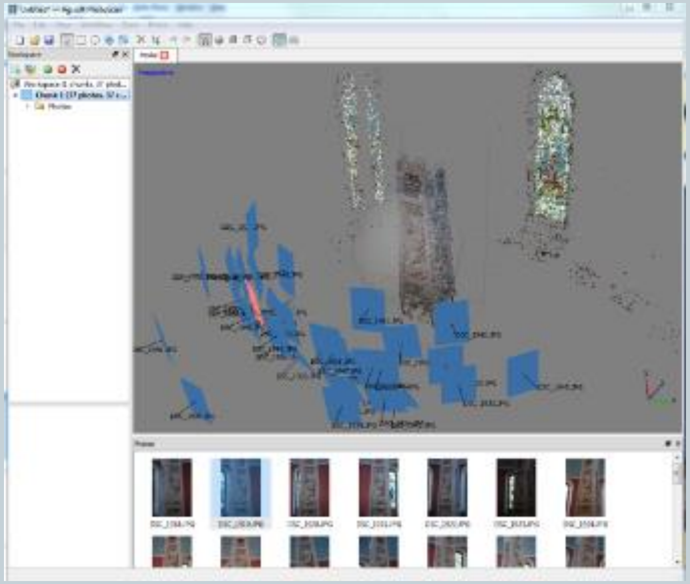


# PhotoScan

Commercial, low cost tool: 59 € for educational license, 179€ standard license. (win, mac & linux)

Fast, work on **local** machine, directly produce textured model. Very robust and reliable... We have used it with good results on many diverse datasets.





They also have an integrated tool for camera calibration

# PhotoScan

Photoscan is the DE FACTO standard tool in CH...

It’s cheap, easy to use, and reliable.

It works incredibly well with DRONES

PRO version has a georeferencing tool, can use markers for automatic scaling, and has a lot of exporting features specific for survey, CAD and GIS tools.

# Autodesk 123Dcatch

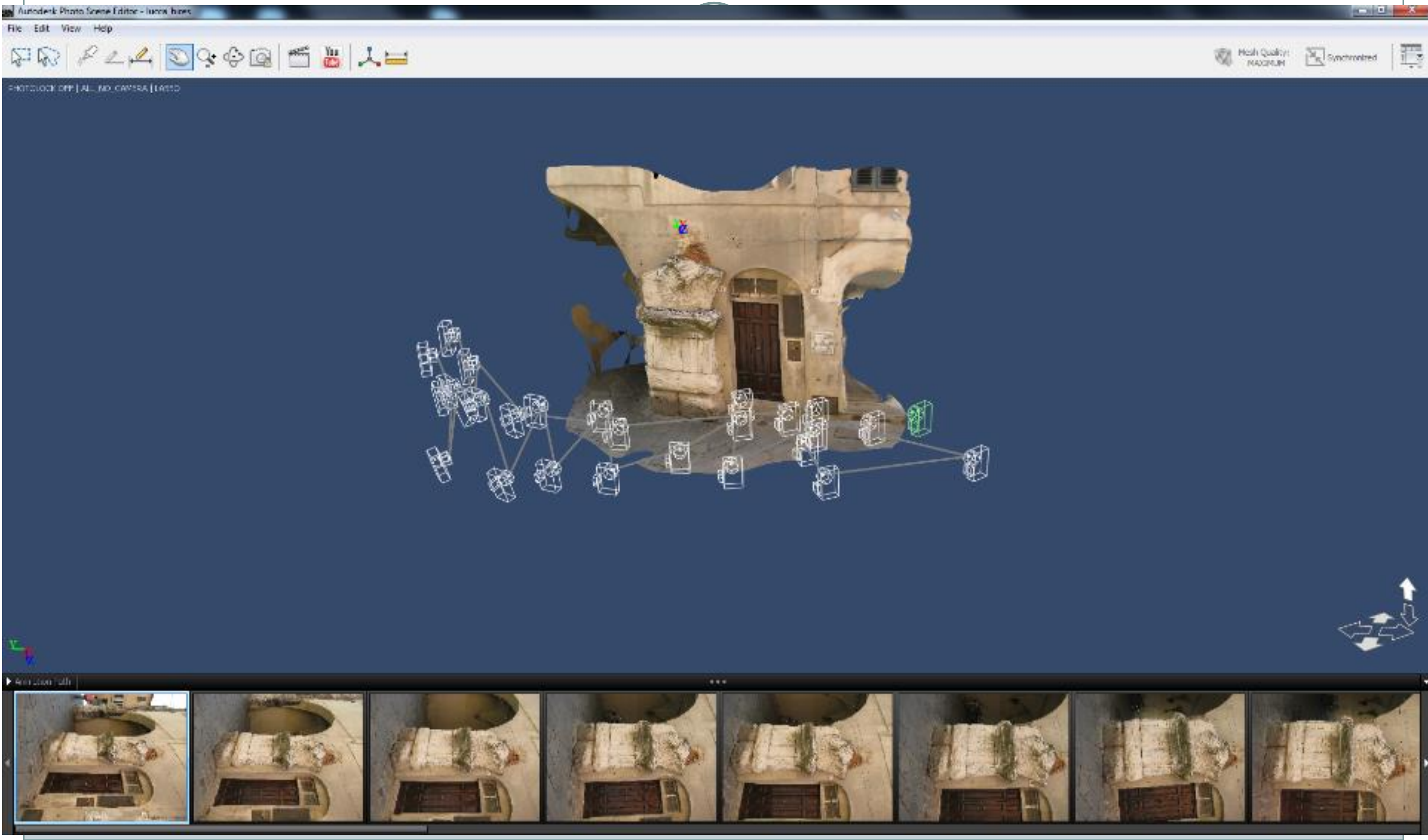
- Very well engineered tool...
- Works on a **remote server**
- Produces a *complete*, *textured* model

http://www.123dapp.com/catch

It is free (*for now*), and works very very well. It is fast, works on difficult datasets and the results looks good. However, not really high resolution, and there is less control over the process. It is a good tool to start with...

The PRO version is **Autodesk RECAP**, with lot more control over the process

# Autodesk 123Dcatch



# Autodesk MEMENTO/RECAP

Just released in beta now.

Complete tool for the mesh acquisition from photos, cleaning, processing, fixing.

Implements the complete processing pipeline for 3D from photos, plus a lot of useful tools for mesh manipulation (although using a very simple approach).

Tailored to output PRINTABLE 3D models, and to create online visualization of 3D models

https://memento.autodesk.com/about

# Photos

And now, let us talk about the photos...

- Do not worry if your first set does not comes out, retry, trying to understand what went wrong.

We will give basic rules, try to follow them at the begin, and the more you got experienced, you will see some may be regarded only as «suggestions»

# Equipment

What kind of camera should I use?

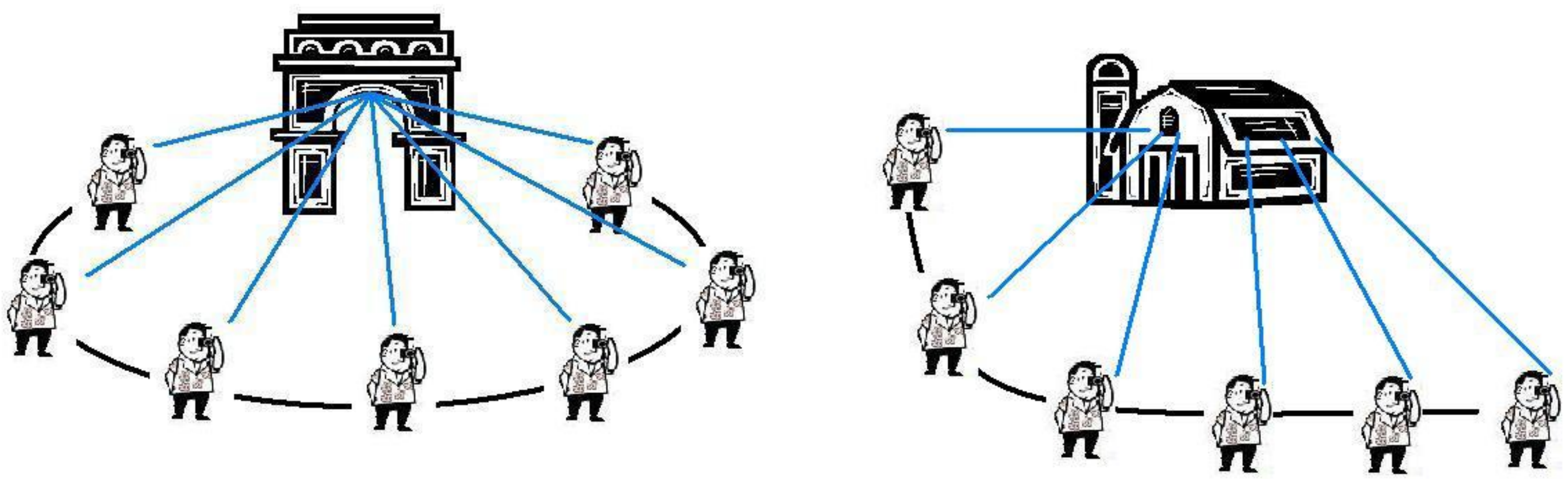
- More pixels = more 3D points = longer upload and processing time
- Using 20-30 Mpixel photos will probably crash the tools, 5-10 are ok, and the result will be better than expected
- Good lens → less distortion → better result
- Good lens → more light → better result

A good compact camera may be enough. DSLR have better lenses.

Mirrorless *may* distort too much (avoid pancake lenses).

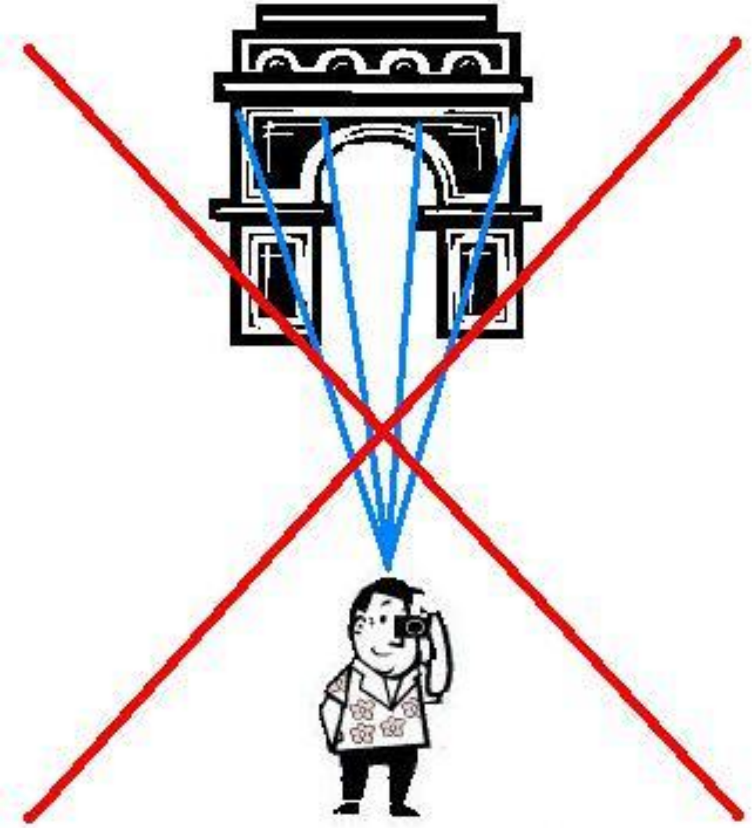
# Good sequence

- ***Walk with the camera in an arc around the scene, while keeping the scene in frame at all times, shoot every few steps***
- ***Keep the zoom FIXED (not always true)***



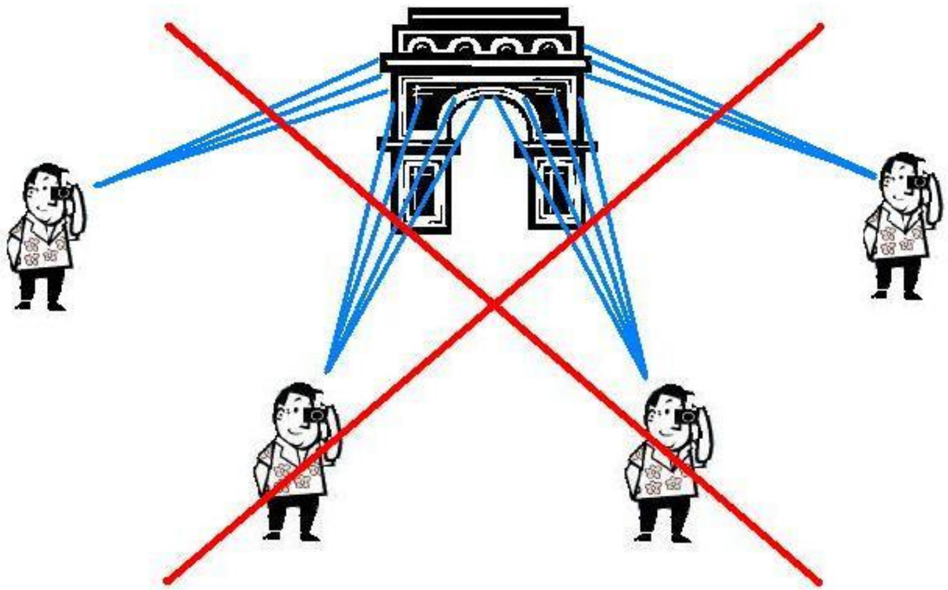
# Bad sequences

- ***Do not pan from the same location***, as if you were recording a panorama. It is not possible to determine enough 3D information from such a sequence.



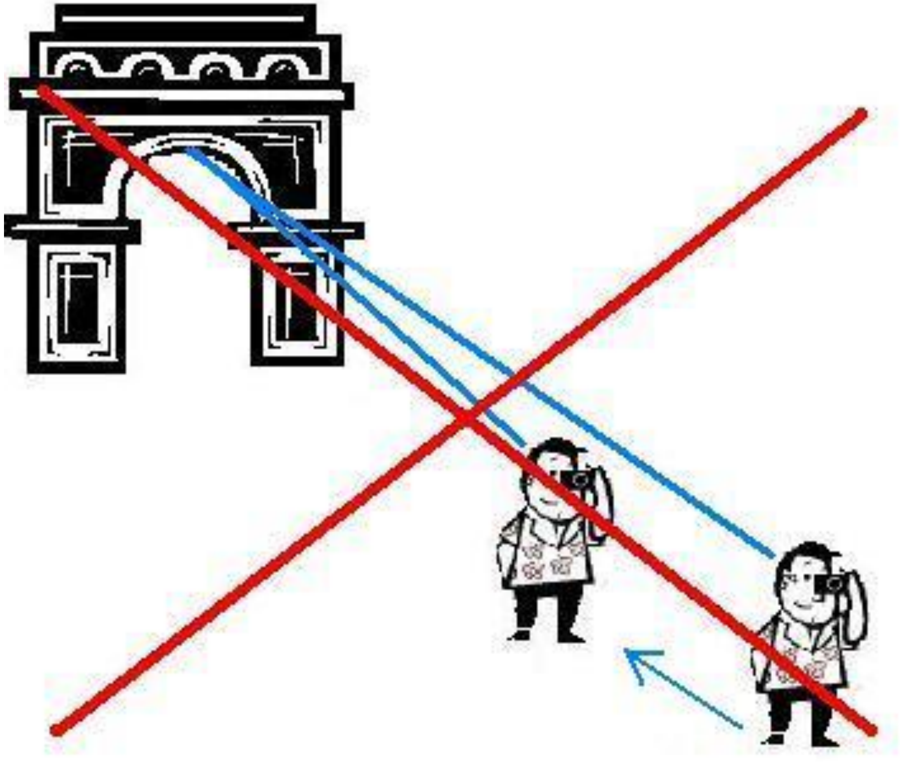
# Bad sequences (2)

- Don’t shoot multiple panorama-like sub-sequences from different viewpoints



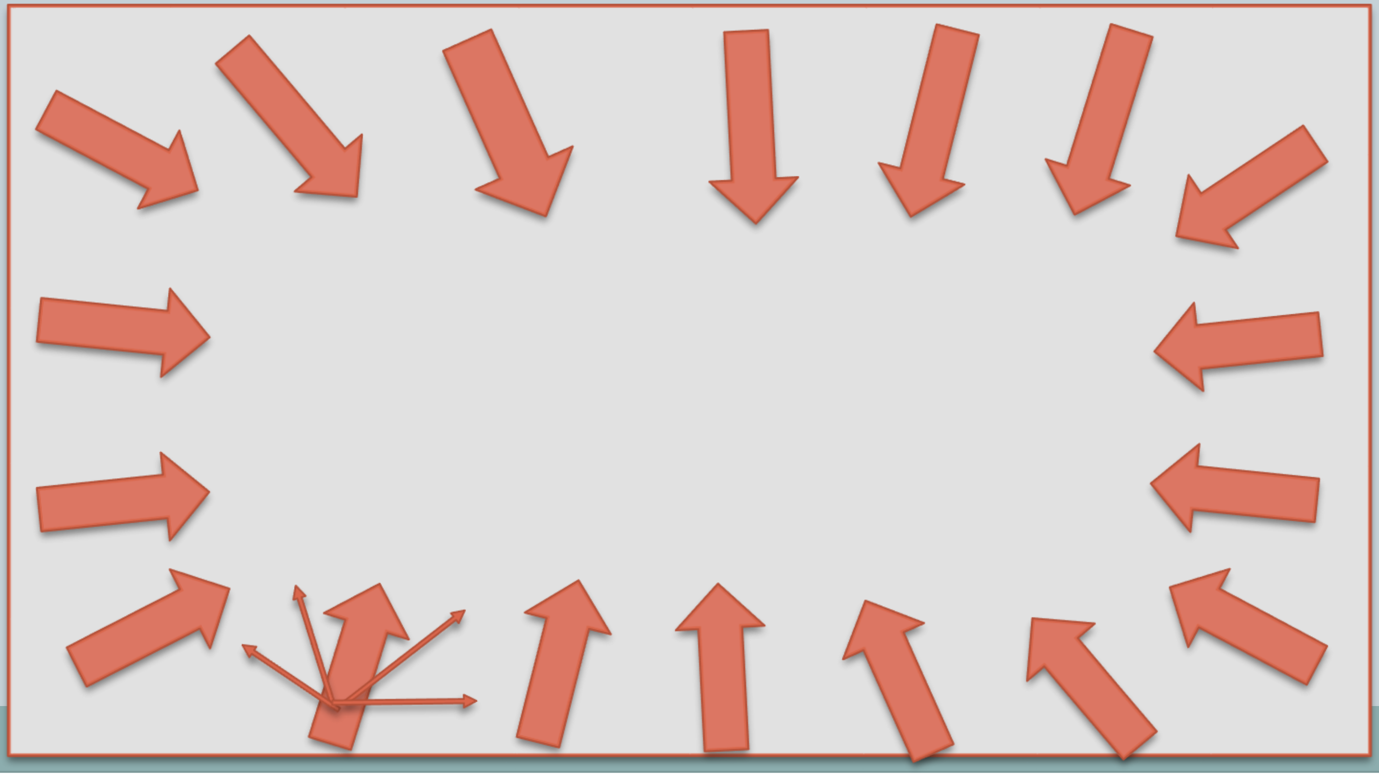
# Bad sequences (3)

- Do not walk in an EXACTLY straight line towards or inside the scene you want to reconstruct



# Good sequence

- ***If inside, walk the perimeter, looking at the opposite side***
- ***In this case you may take more photos for each point, but NOT like a panorama (small or no overlap)***

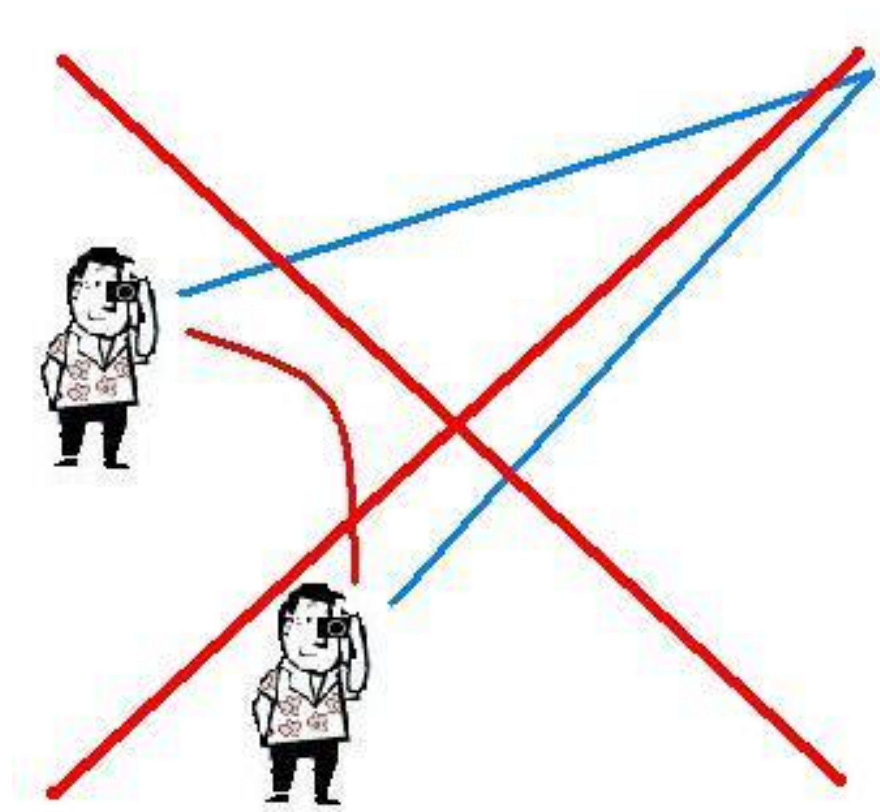


# Good sequence

- ***Shoot from different heights... This helps a lot***
- ***You can mix photos taken at different distances: i.e. shoot the whole object going around, then get closer and cover the object again framing smaller areas, then get closer again and frame details***
- ***Background is important!!!!***

# Bad sequences (4)

- It is better to shoot a lot of pictures than few ones.
- The viewing angle between images should not be too large, i.e. adjacent images should not be too far apart
- Consider 15-20 degree as a good step...



# Bad sequences (5)

NO TURNTABLE

NO PLANAR SCENE

# Practical Problems

All information is retrieved from the images, so take care when you shoot them!

The texture (color, intensity) of the scene/object is critical!

- Enough texture must be available on the object
- Appearance of object must stay the same!

# Not Enough Texture

# No Constant Appearance

# No Constant Apperance

# No Constant Apperance

# No Static Scene

Dynamic Scene cannot be reconstructed

# Don’t use blurry images

- Blurry images (due to movements or out-of-focus) must avoided
- This causes problems during the reconstruction process and/or degrades the final result

# Self-Occlusions

- Self-occlusions have to be treated with care (be sure that your photos cover all the self-occluded parts).

# Lighting Conditions

**Overcast sky is perfect due to uniform illumination.**

**In general changing conditions should be avoided...**
**NO FLASH (if possible)**

**Moving Shadows should be avoided...**