

# Automatic alignment of paintings and photographs depicting a 3D scene

Bryan C. Russell

University of Washington

In collaboration with: Josef Sivic, Jean Ponce, H  l  ne Dessales

Work done while at INRIA, WILLOW project-team,   cole Normale Sup  rieure

# Goal

## Inputs



## Photographs



## Painting

# Goal

Inputs

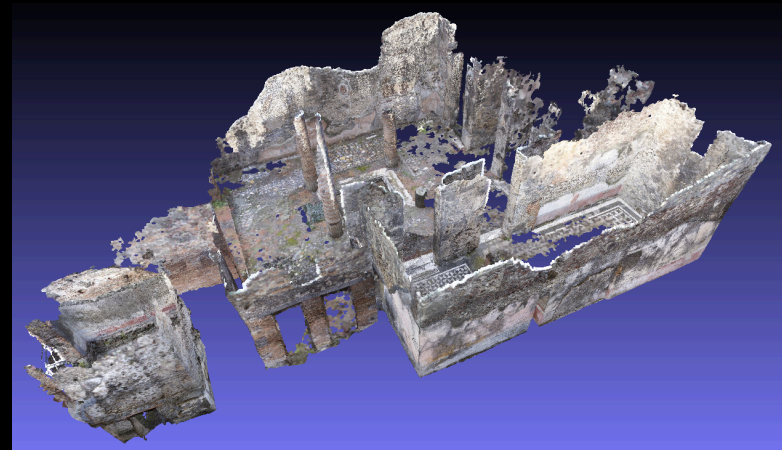


Photographs



Painting

Outputs



3D model

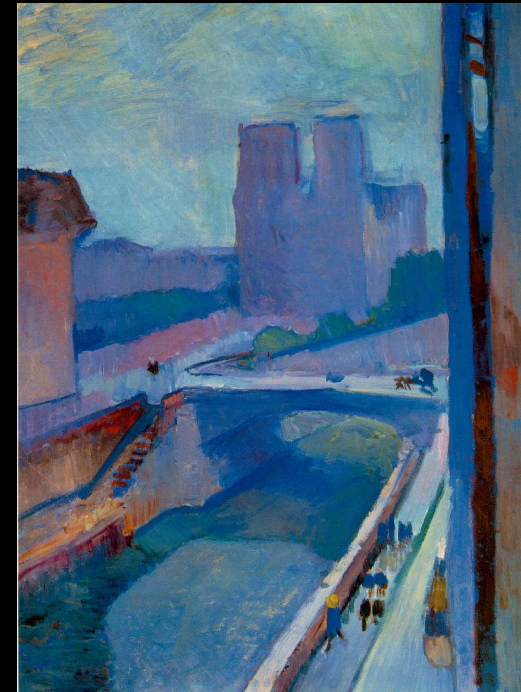
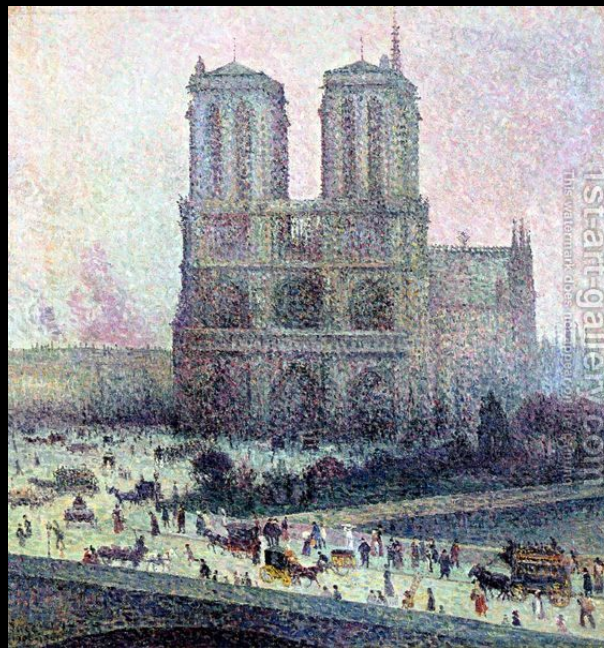


Viewpoint of painting



# Why do this?

There are many non-photographic depictions of our world



Ultimate goal: to reason about these depictions



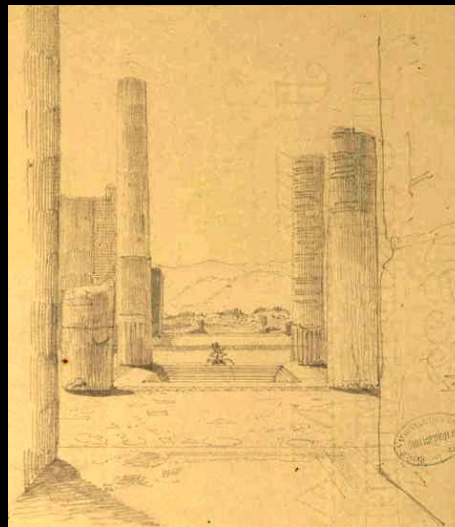
# Why do this?

For this work, we will focus on paintings where the artist made an effort to accurately render the 3D scene

Paintings from Pompeii, Casa di Championnet:



Scholander



Blouet (1825)



Gell (1814-1817)

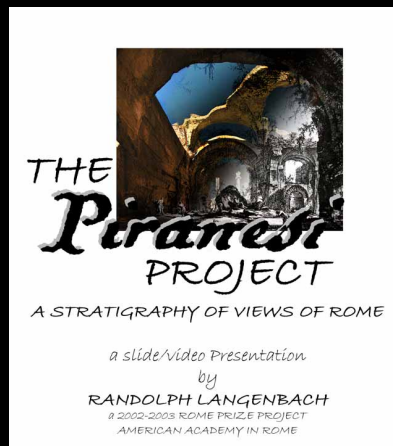
- Useful for applications where drawings and paintings are the primary record (e.g. archaeology)
- We believe that a camera lucida was used to aid the artists
- 1817 - First photograph taken [Wikipedia]

# 9 MINUTE PREVIEW OF *The Piranesi Project*

by

© Randolph Langenbach, 2003

The normal running time for the full show is approximately 50 minutes.



PRESS

**“ENTER”**

TO BEGIN SHOW

The PREVIEW will then run automatically.  
Running time of this preview is approximately 9  
minutes

















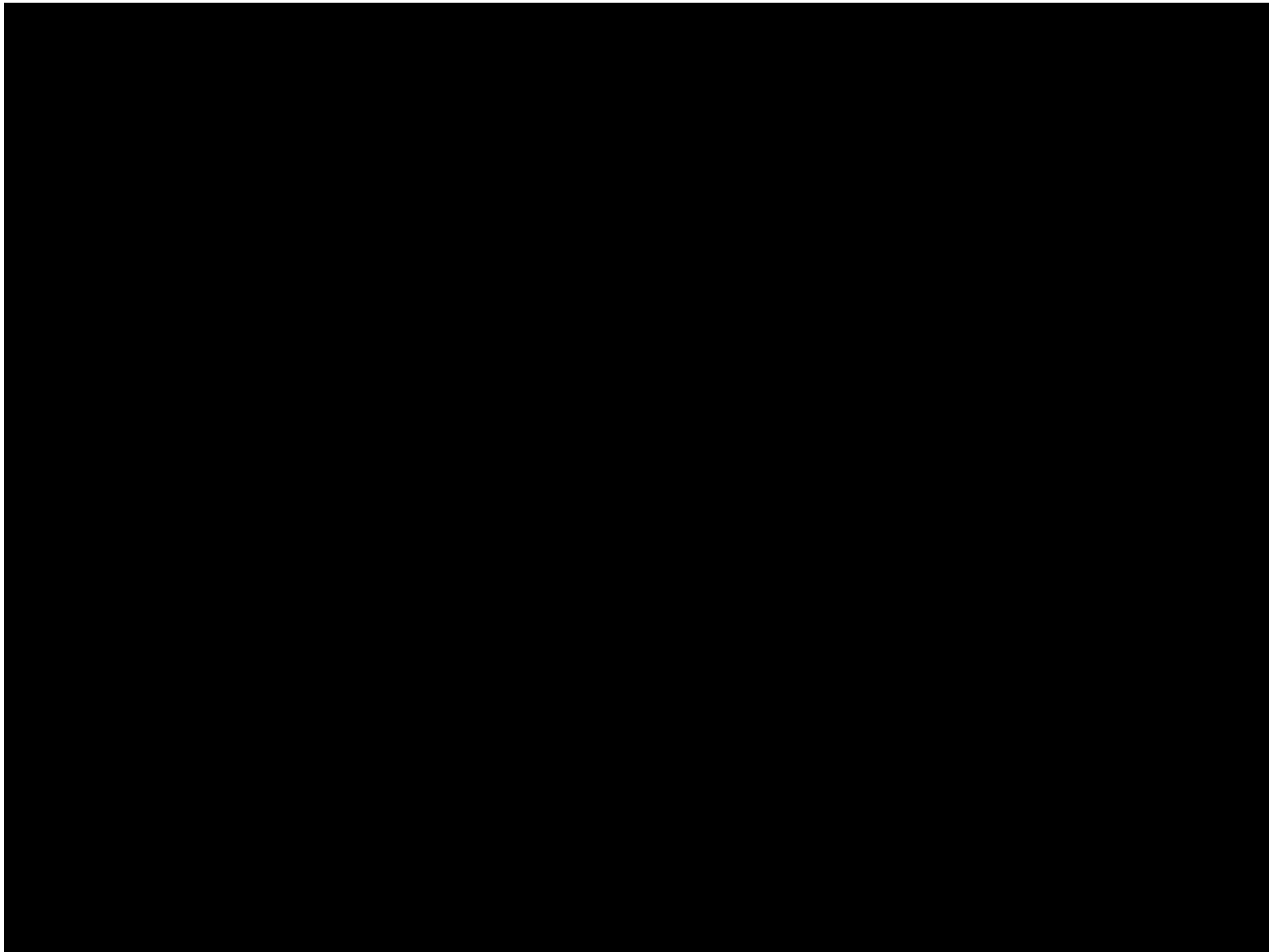










































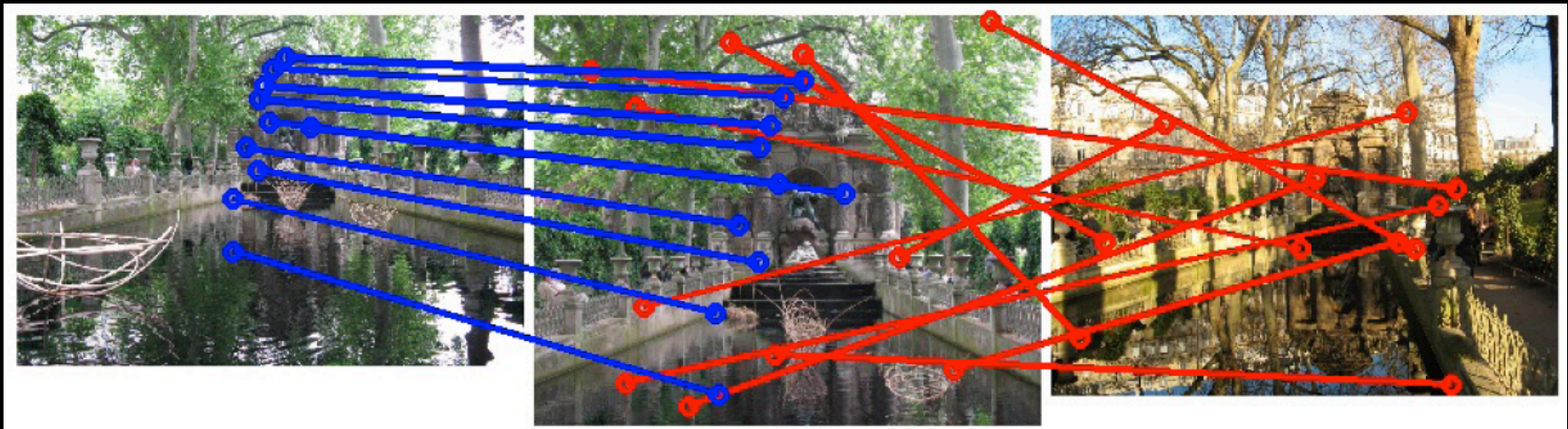






# Why is this hard?

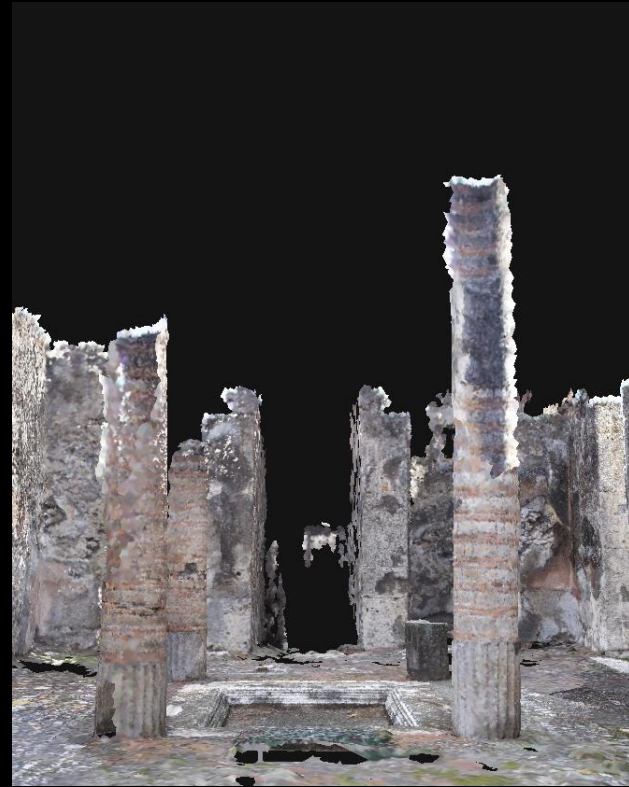
Local feature matching using SIFT:



A. Shrivastava, T. Malisiewicz, A. Gupta, A. Efros  
Data-driven Visual Similarity for Cross-domain Image Matching  
To appear SIGGRAPH Asia 2011

# Difficulty in finding correspondences

Color, geometry, illumination, shading, shadows and texture may be rendered by the artist in a realistic, but “non physical” manner

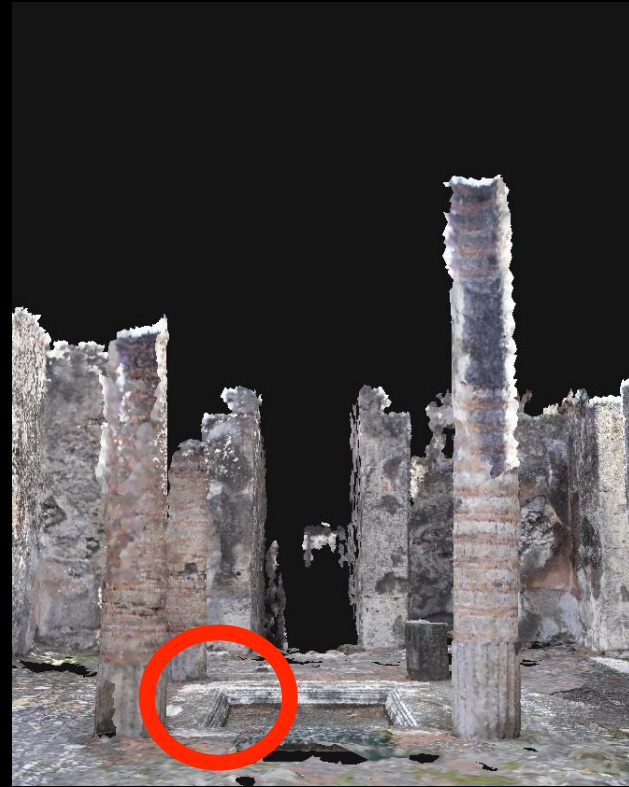
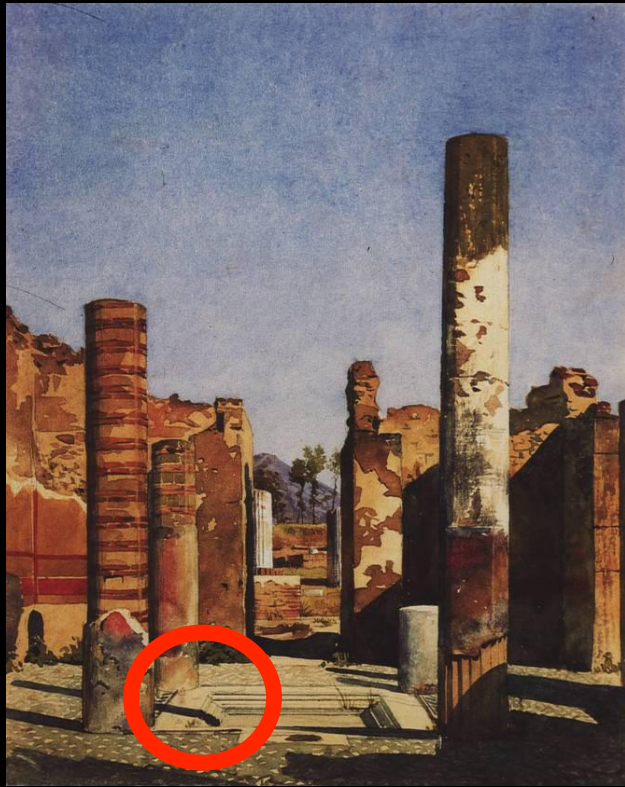


- 121 putative matches total across 563 photographs using SIFT matching
- 0 correct putative matches



# Difficulty in finding correspondences

Color, geometry, illumination, shading, shadows and texture may be rendered by the artist in a realistic, but “non physical” manner



- 121 putative matches total across 563 photographs using SIFT matching
- 0 correct putative matches

# Difficulty in finding correspondences

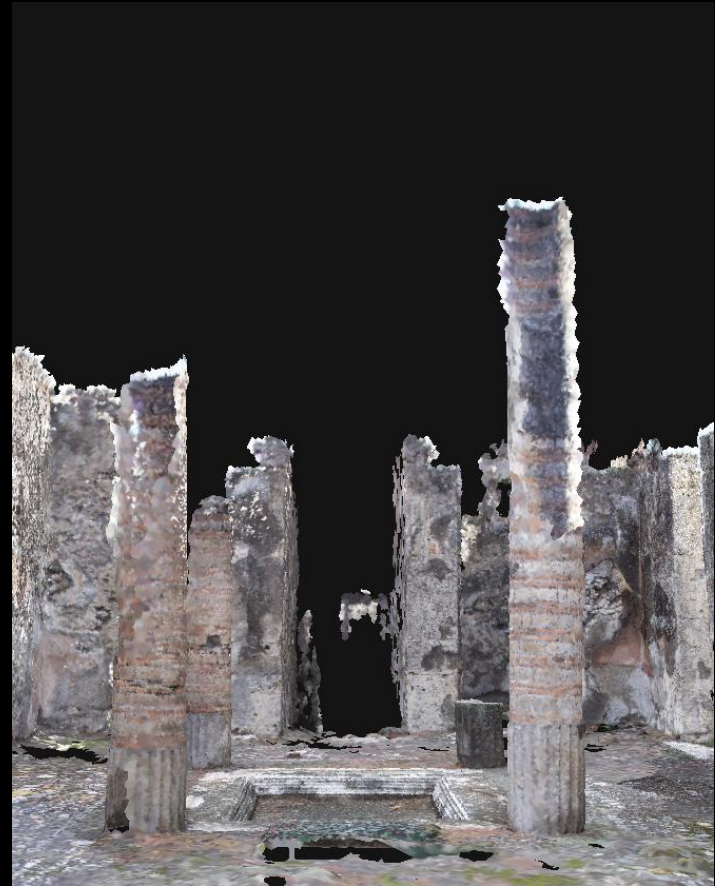
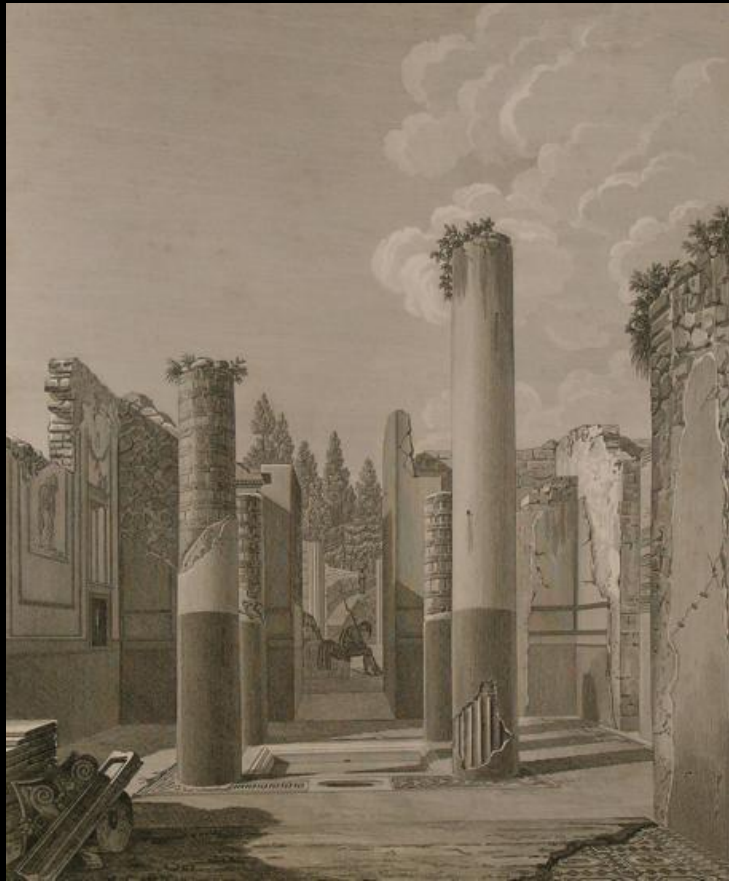
Color, geometry, illumination, shading, shadows and texture may be rendered by the artist in a realistic, but “non physical” manner



- 121 putative matches total across 563 photographs using SIFT matching
- 0 correct putative matches



# Why is this hard?



Structural changes over time

# Why is this hard?



Structural changes over time



# Why is this hard?



Murals missing



Back column collapse



# Problem statement

Inputs

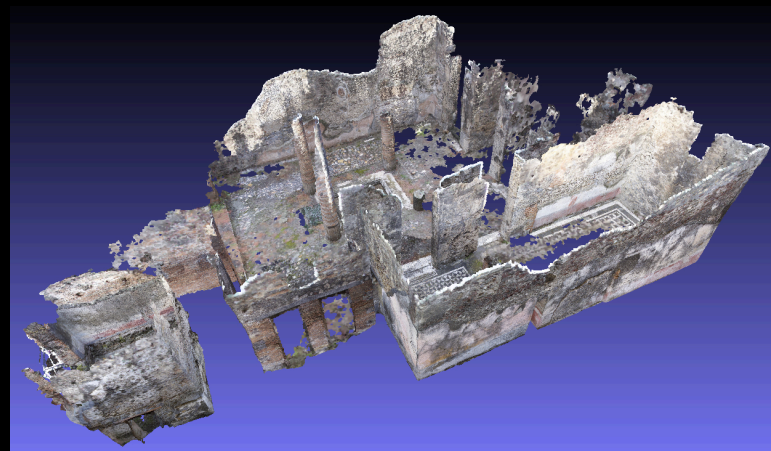


Photographs



Painting

Outputs



3D model: 3D vertices + triangle indices



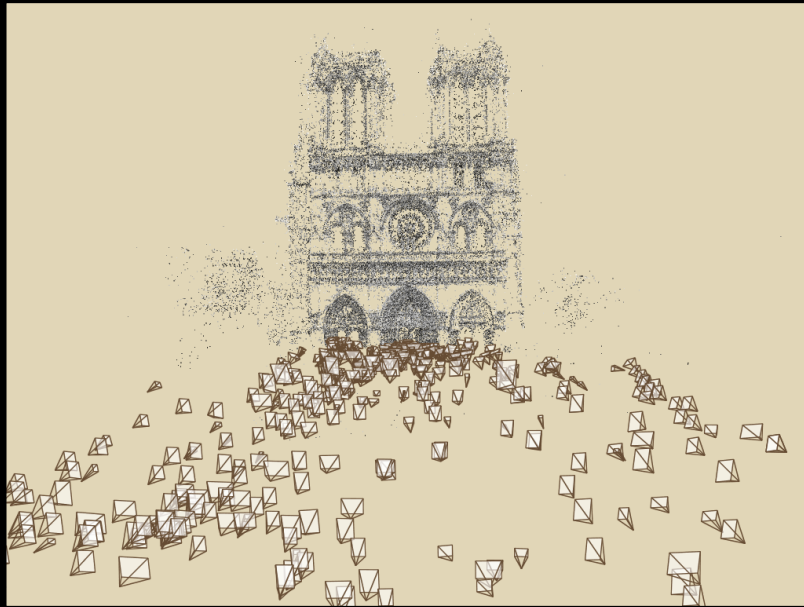
Camera parameters  $\Theta$

Camera center, rotation, principal point, focal length



# Multiview geometry

## Structure from motion (SfM)



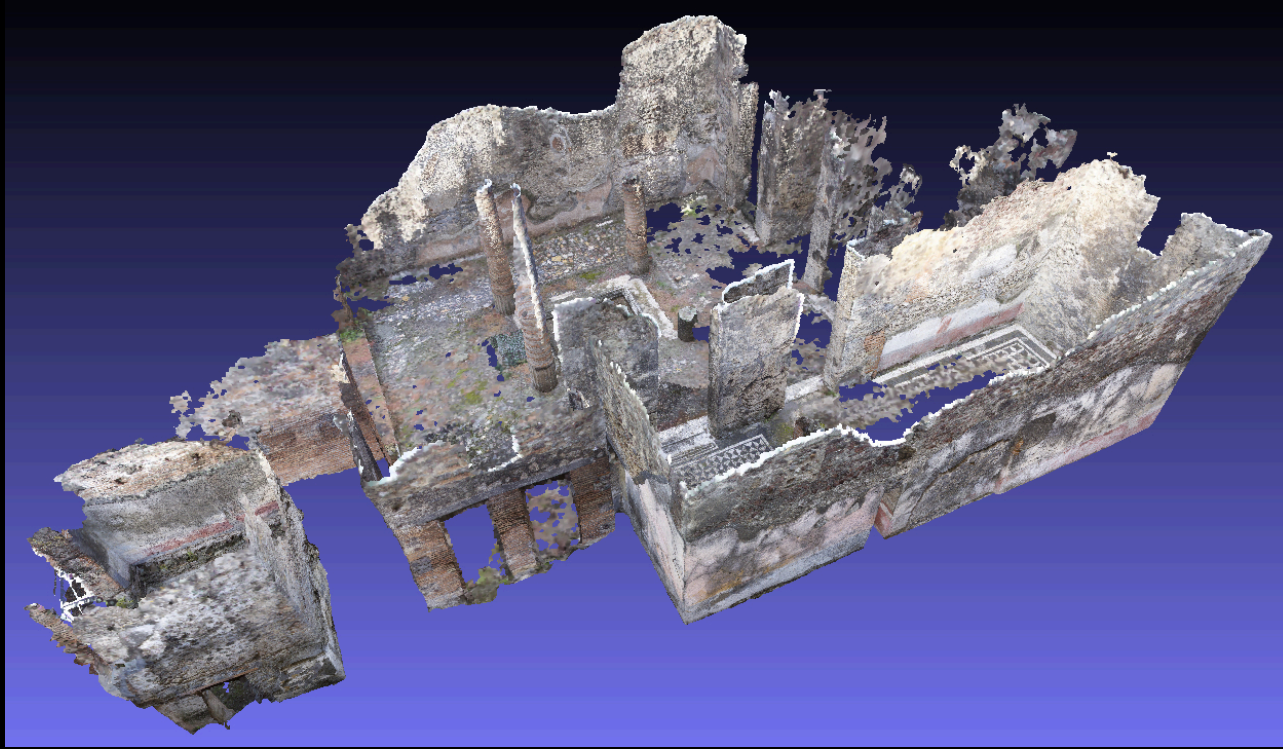
- N. Snavely, S. M. Seitz, R. Szeliski, 2007
- M. Vergauwen, L. Van Gool, 2006
- M. Brown, D. Lowe, 2005
- F. Schaffalitzky, A. Zisserman, 2002

## Dense multiview stereo



- Y. Furukawa, J. Ponce, 2009
- P. Labatut, J.-P. Pons, R. Keriven, 2009
- M. Goesele, N. Snavely, B. Curless, H. Hoppe, S. M. Seitz., 2007

# Pompeii 3D model details



- Used publicly available software
  - Bundler, PMVS, Poisson surface reconstruction
- 563 photographs (4752 x 3164 resolution)
- Bundler recovers 559 cameras
- PMVS recovers 111M points (discard points due to speed)
- Final mesh: 10M vertices, 20M triangles



# Technical contribution: two-stage alignment procedure

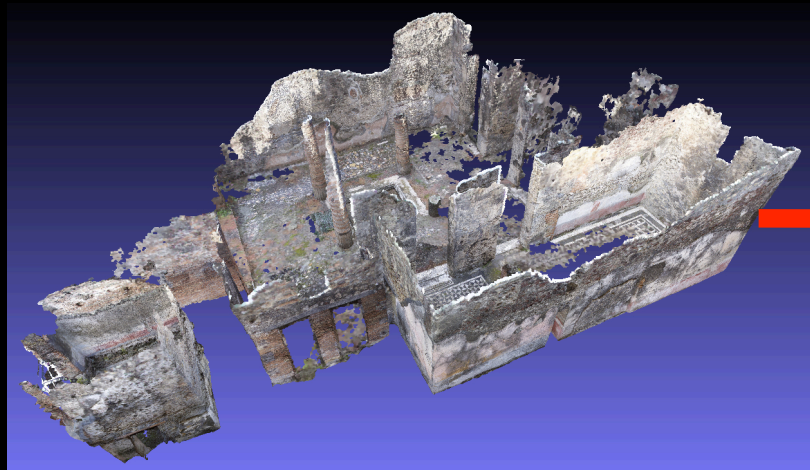
- Coarse alignment by view-sensitive retrieval
- Fine alignment by matching view-dependent contours

# Technical contribution: two-stage alignment procedure

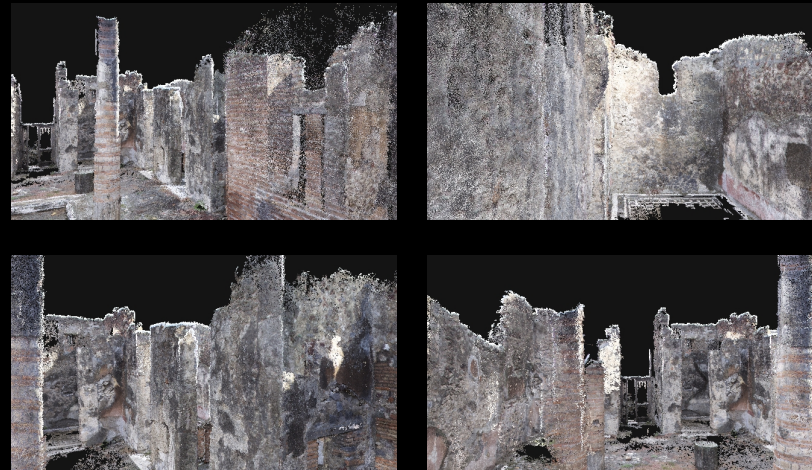
- Coarse alignment by view-sensitive retrieval
- Fine alignment by matching view-dependent contours



# Goal: retrieve similar viewpoint



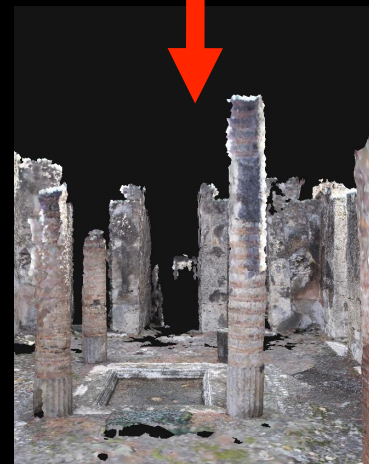
3D model



Sampled viewpoints



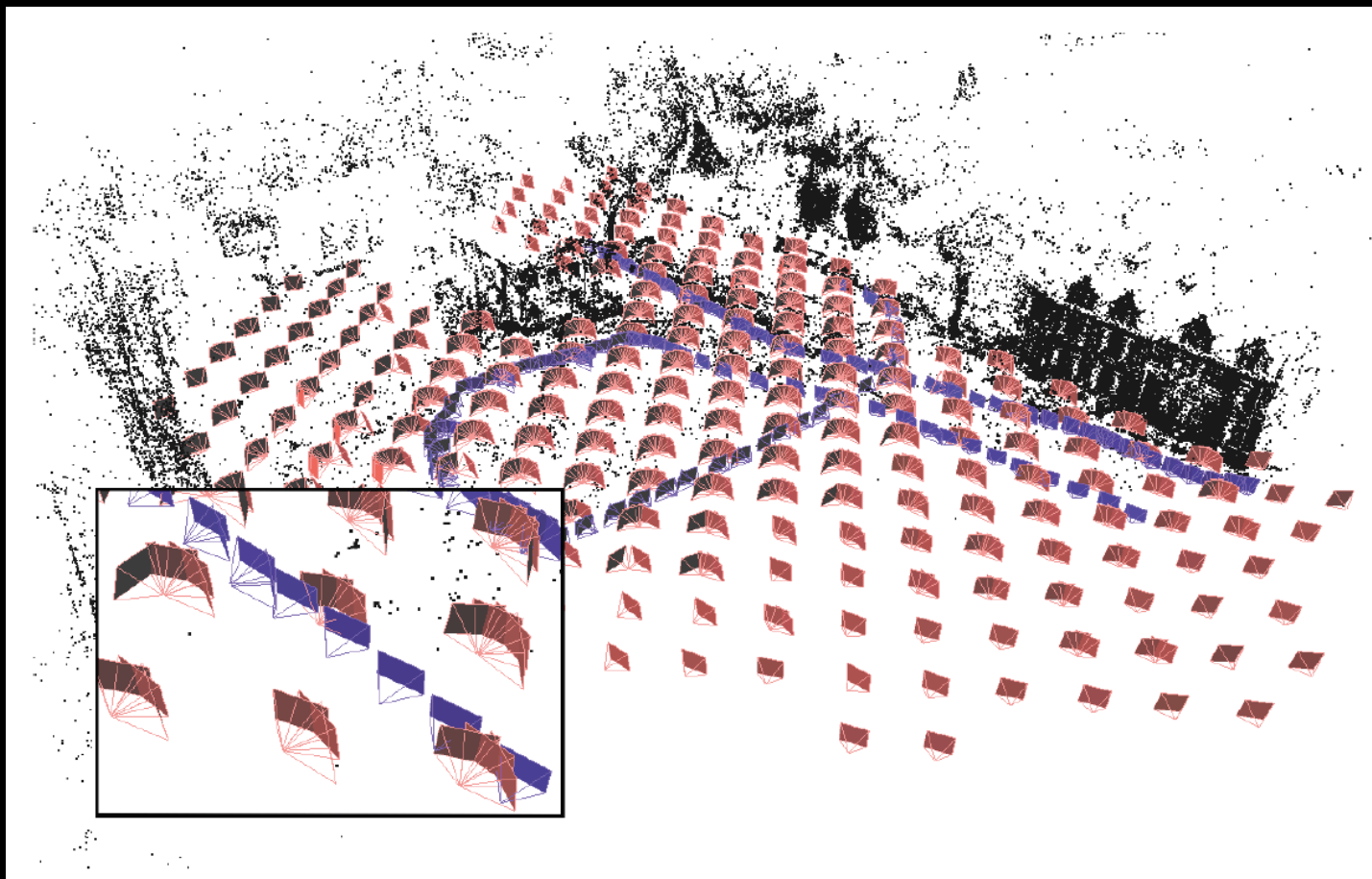
Painting



Retrieved viewpoint

# Sampling virtual viewpoints

Irschara et al. CVPR, 2009



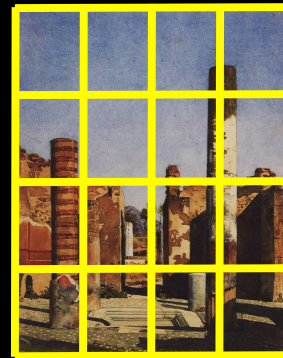
Virtual viewpoints used to group visual words  
for fast vocabulary tree-based indexing



# Viewpoint retrieval using gist feature matching



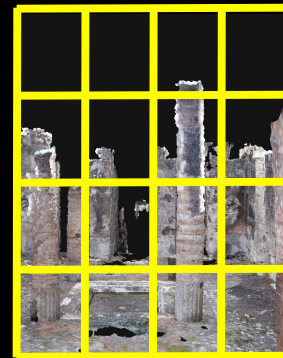
Painting



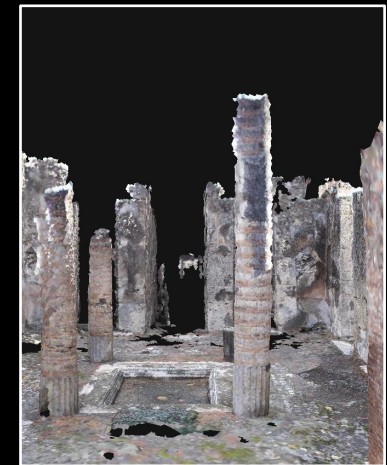
Gist



8K sampled viewpoints



Gist



Retrieved viewpoint

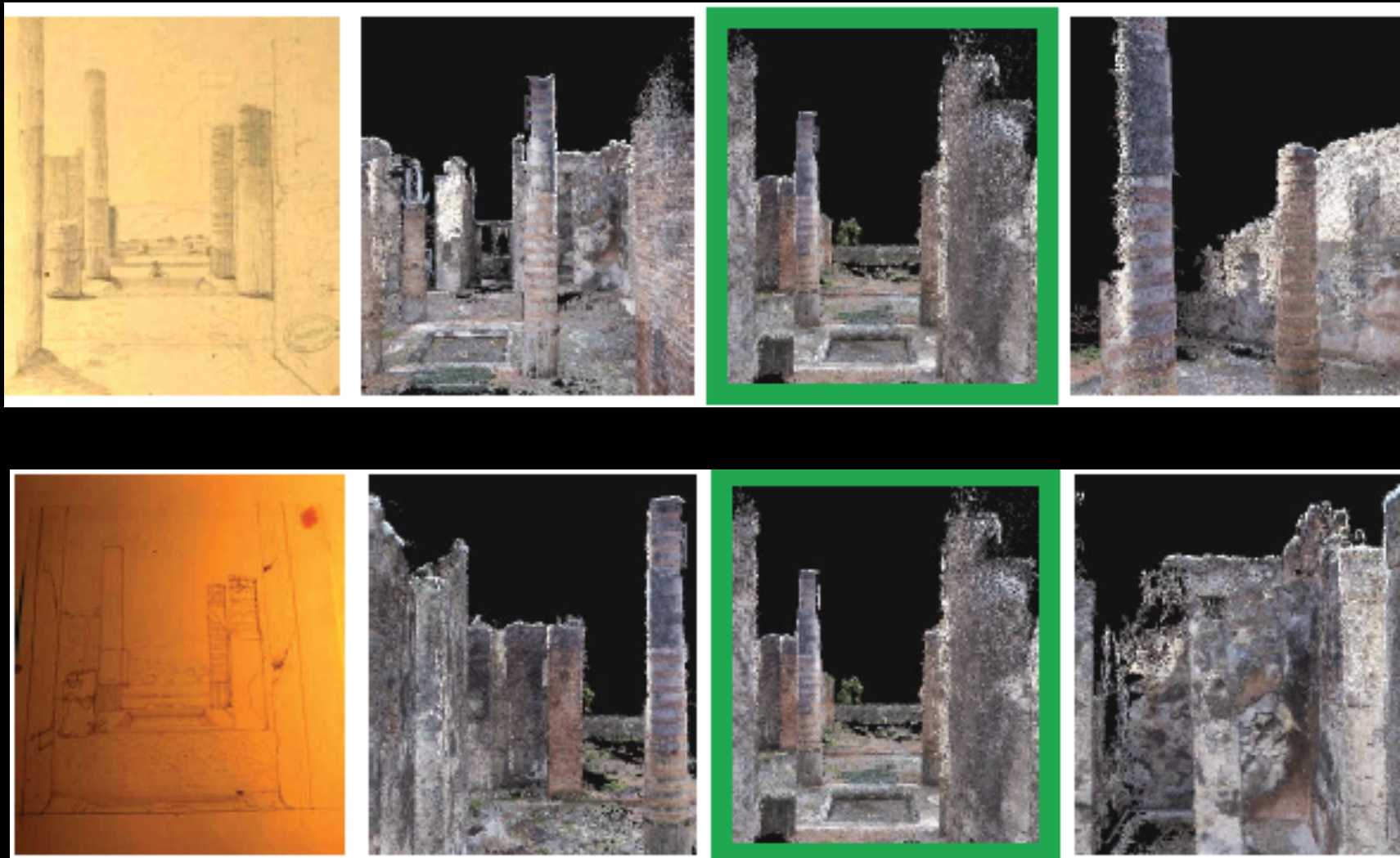
- Assume camera held level with ground plane at fixed height
- Sample camera locations in dense 2D grid; 8 orientations at each point

# Viewpoint retrieval results using gist feature matching





# Viewpoint retrieval results using gist feature matching



# Technical contribution: two-stage alignment procedure

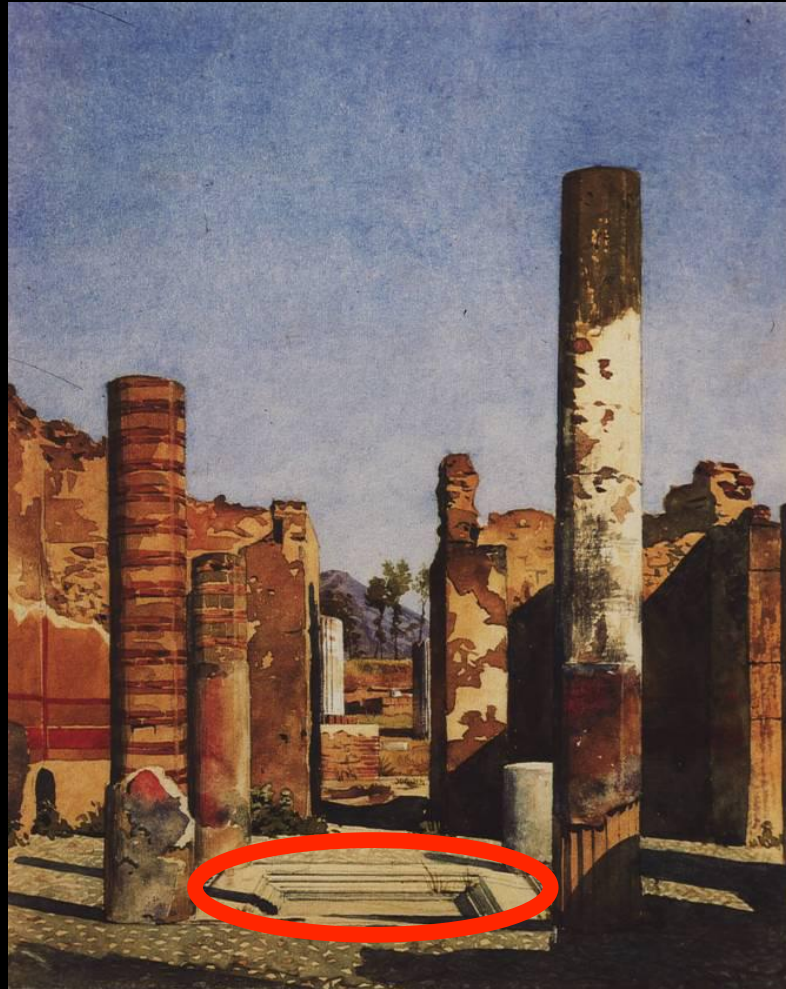
- Coarse alignment by view-sensitive retrieval
- Fine alignment by matching view-dependent contours



# Features for matching painting to 3D model



# Features for matching painting to 3D model





# Features for matching painting to 3D model



# Features for matching painting to 3D model





# Features for matching painting to 3D model



Goal: extract contours corresponding to major structures

# Extracting contours from painting



Painting



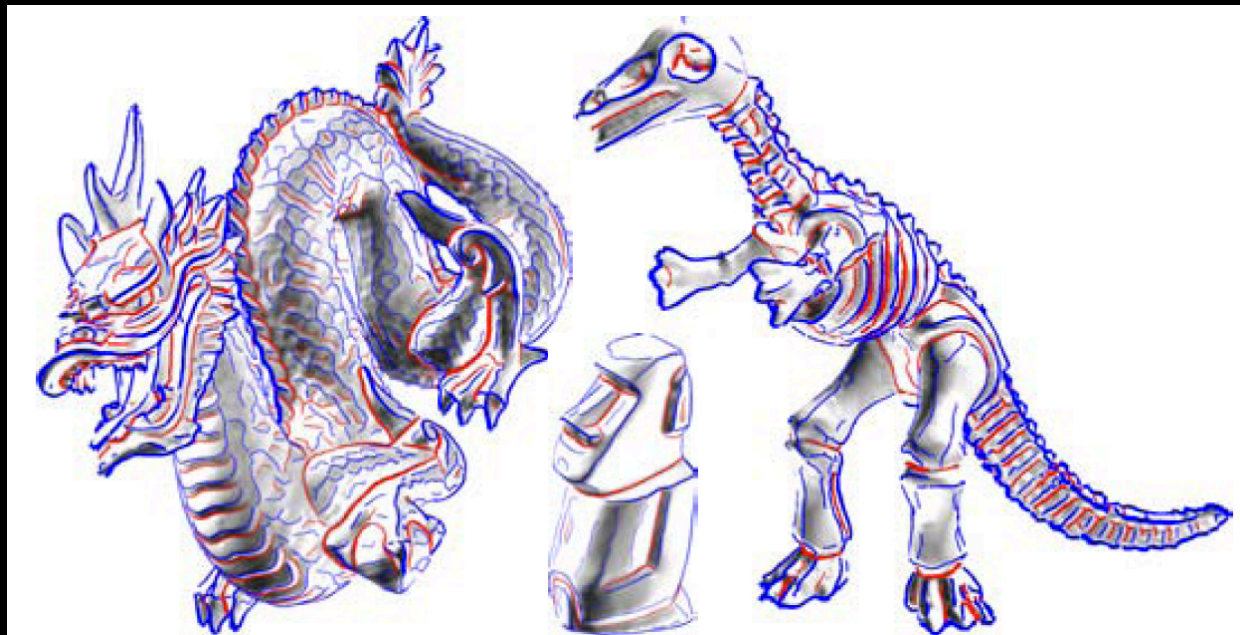
Global probability of boundary (gPB)  
[Maire et al. 2008]

At each pixel: thresholded  
response + orientation



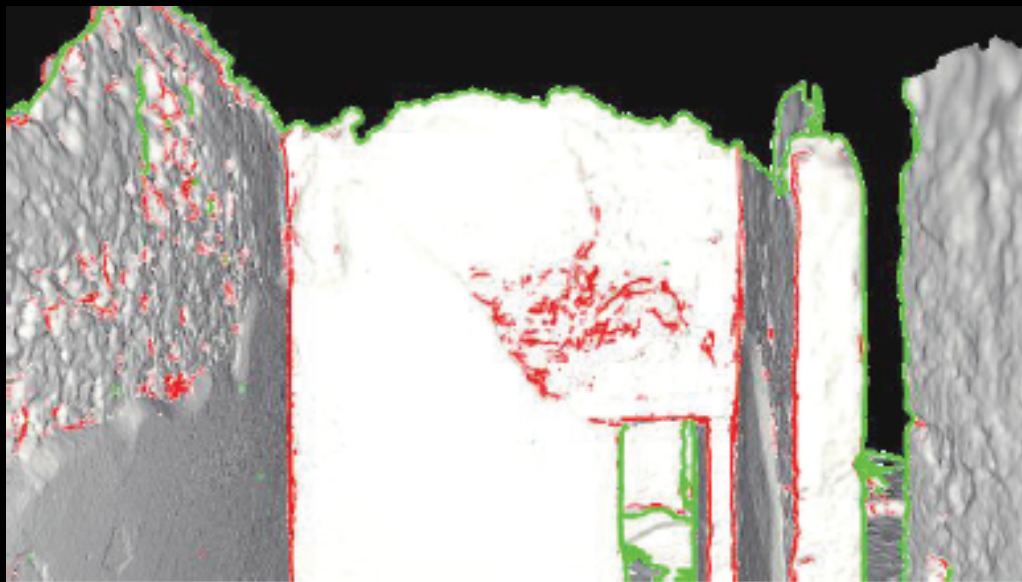
# Extracting contours from 3D model

- Viewpoint dependent: occlusions
- Viewpoint independent: folds and creases
  - Use ridges and valleys [Ohtake et al. 2004]



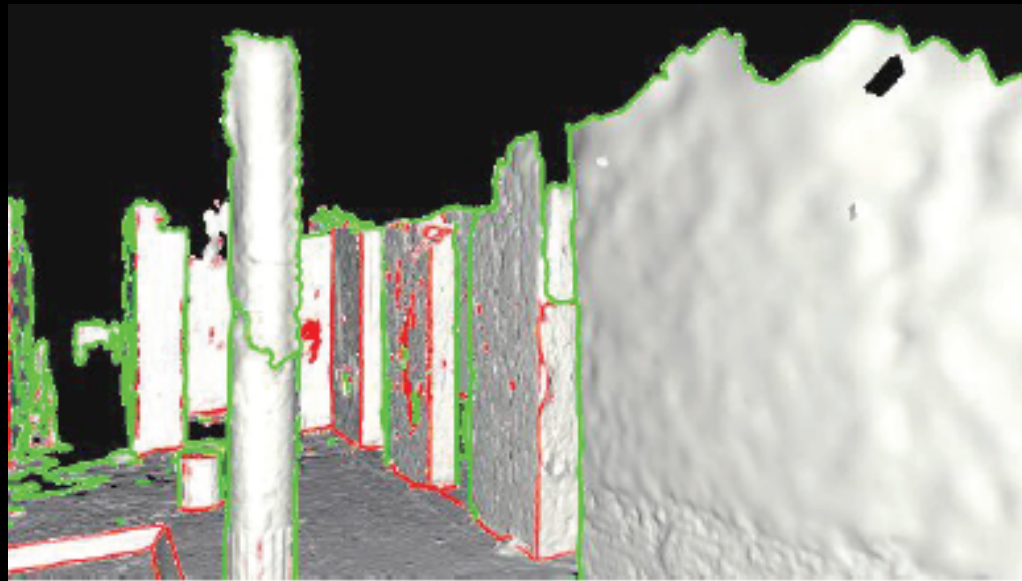
Used publicly available rtsc code: <http://www.cs.princeton.edu/gfx/proj/sugcon>

# Extracting contours from 3D model

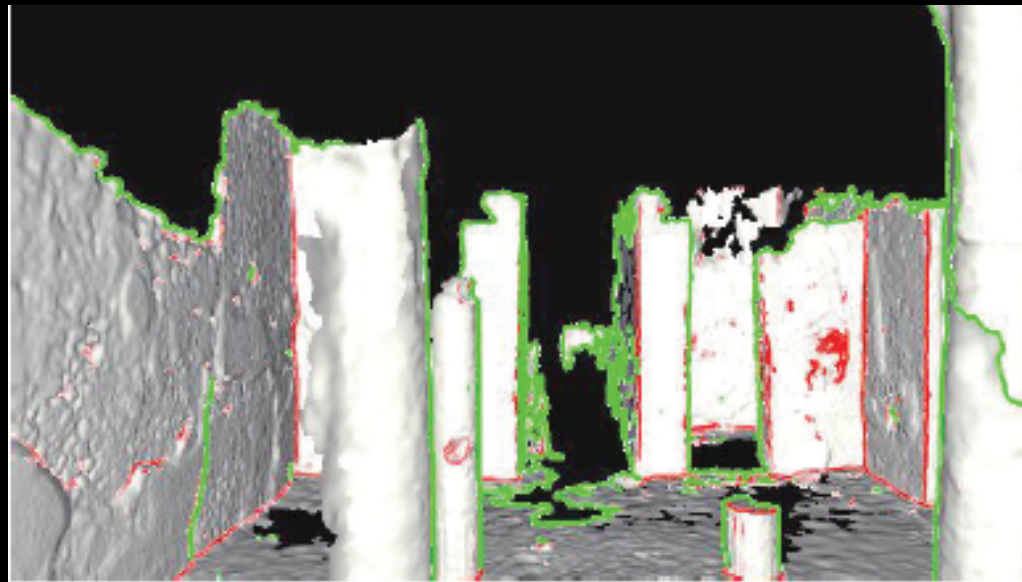
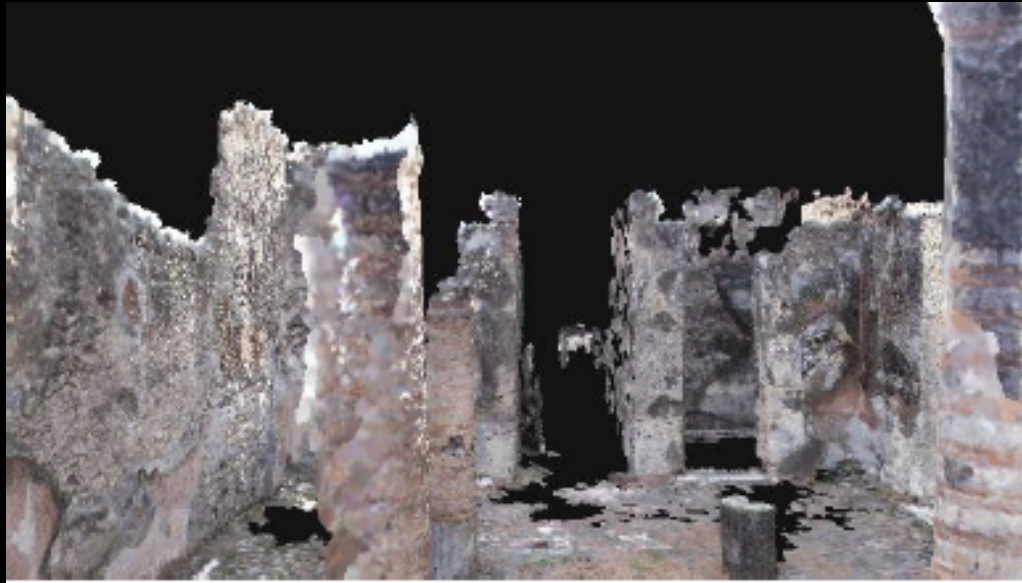




# Extracting contours from 3D model



# Extracting contours from 3D model

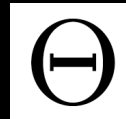
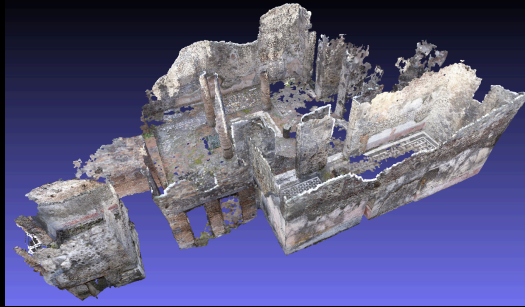




# Painting and 3D model contours



Painting contours



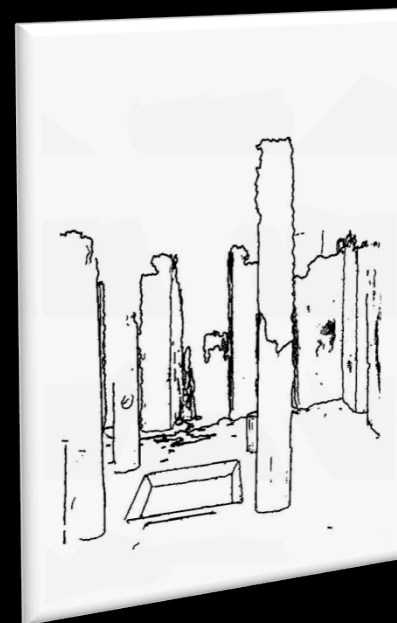
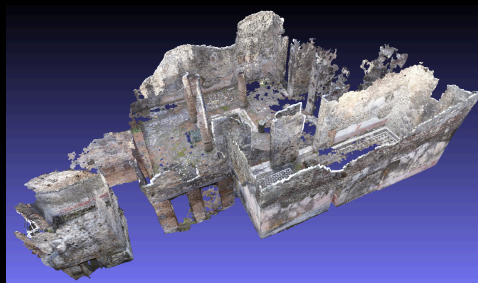
3D model contours

# Dense alignment cost

$$\min_{\Theta} \sum_{\mathbf{x}_i \in S_{\mathcal{P}}} \min_{\mathbf{x}_j \in S_{\mathcal{M}}(\Theta)} |\mathbf{x}_i - \mathbf{x}_j|^2$$



Painting contours

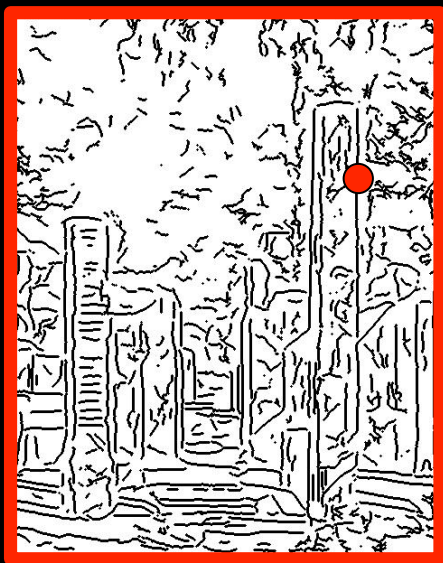


3D model contours

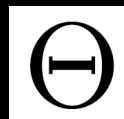
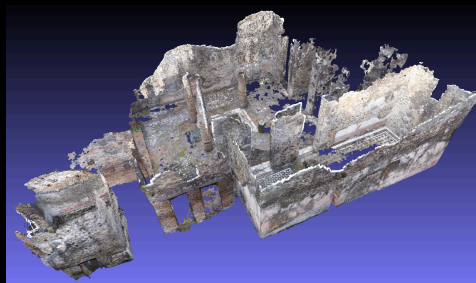


# Dense alignment cost

$$\min_{\Theta} \sum_{\mathbf{x}_i \in S_{\mathcal{P}}} \min_{\mathbf{x}_j \in S_{\mathcal{M}}(\Theta)} |\mathbf{x}_i - \mathbf{x}_j|^2$$



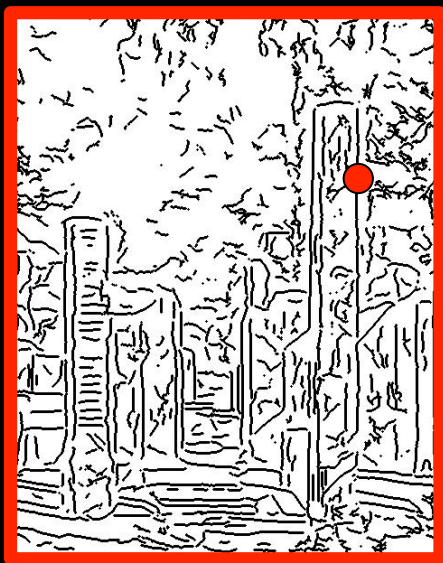
Painting contours



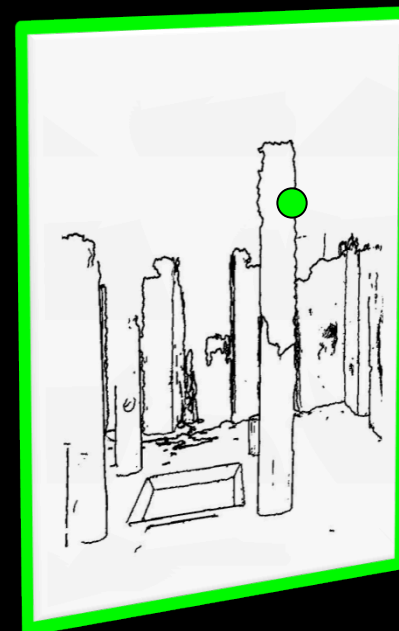
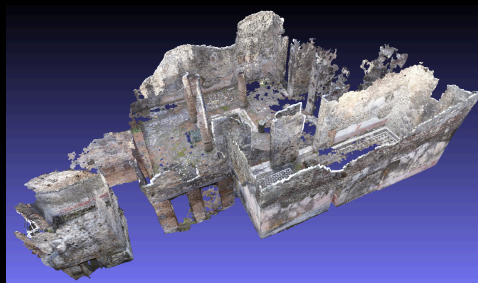
3D model contours

# Dense alignment cost

$$\min_{\Theta} \sum_{\mathbf{x}_i \in S_{\mathcal{P}}} \min_{\mathbf{x}_j \in S_{\mathcal{M}}(\Theta)} |\mathbf{x}_i - \mathbf{x}_j|^2$$



Painting contours



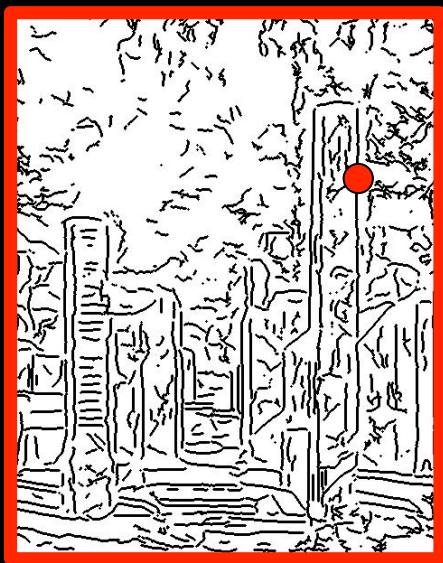
3D model contours



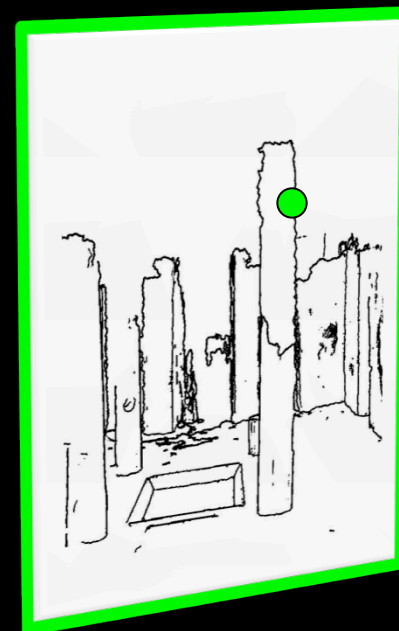
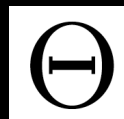
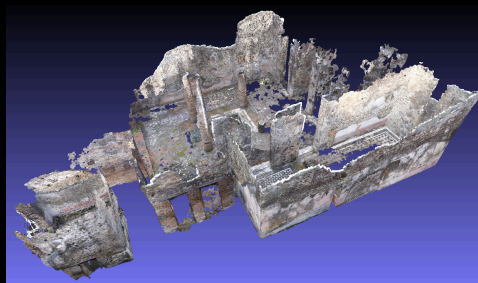
# Dense alignment cost

$$\min_{\Theta} \sum_{\mathbf{x}_i \in S_{\mathcal{P}}} \min_{\mathbf{x}_j \in S_{\mathcal{M}}(\Theta)} \min(|\mathbf{x}_i - \mathbf{x}_j|^2, \gamma)$$

Truncated error – robustness to outliers



Painting contours

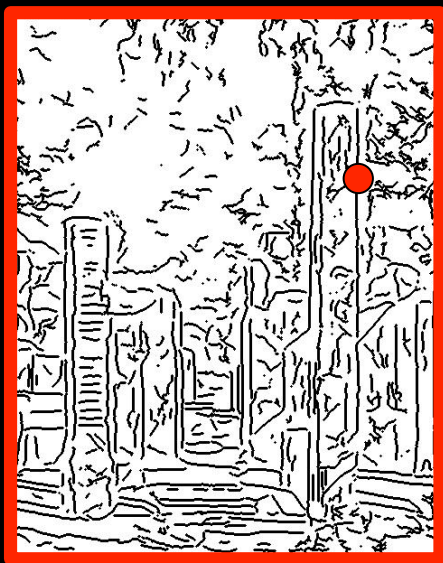


3D model contours

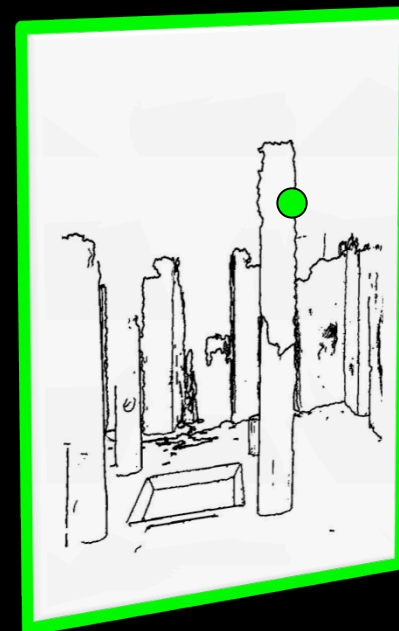
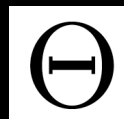
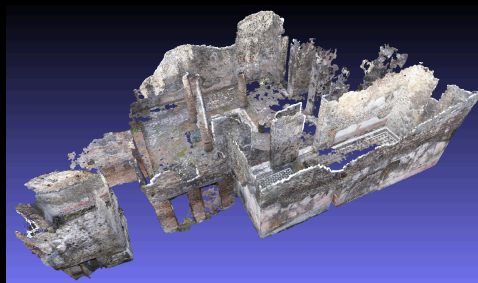
# Dense alignment cost

$$\min_{\Theta} \sum_{\phi} \sum_{\mathbf{x}_i \in S_{\mathcal{P}}(\phi)} \min_{\mathbf{x}_j \in S_{\mathcal{M}}(\phi, \Theta)} \min(|\mathbf{x}_i - \mathbf{x}_j|^2, \gamma)$$

2D edge orientation



Painting contours



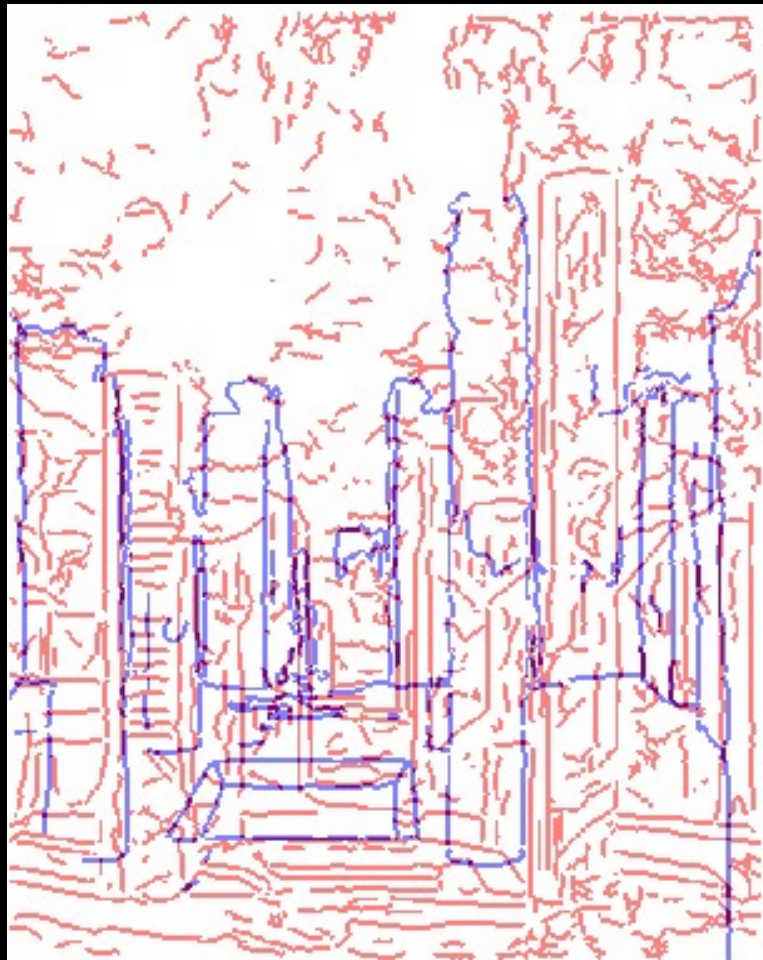
3D model contours



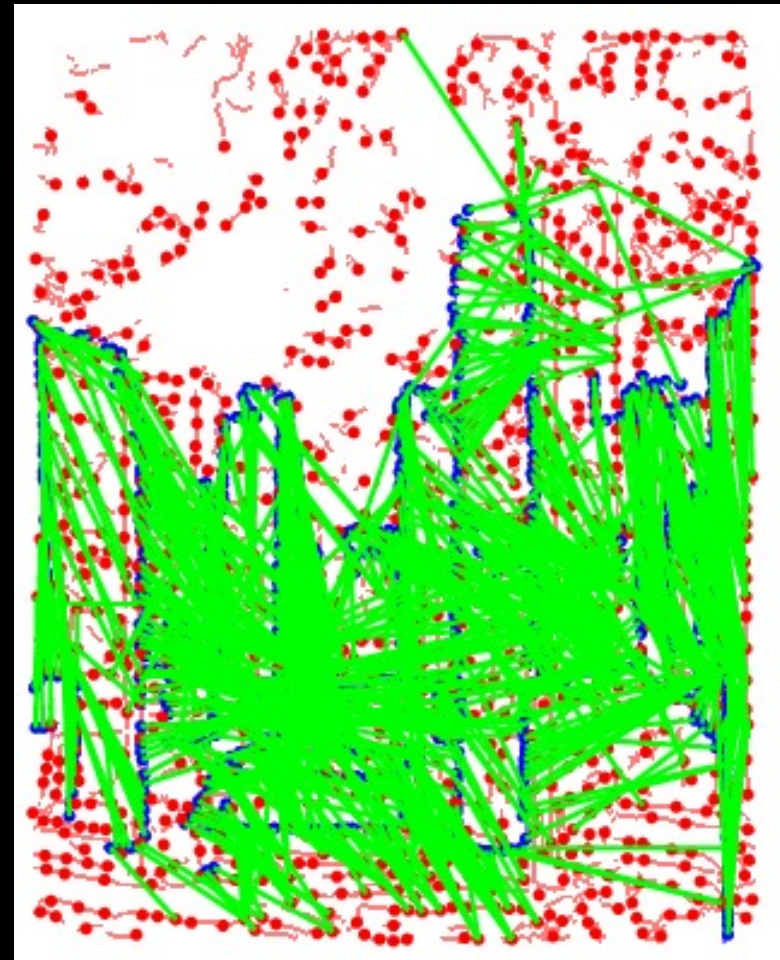
# Minimizing dense alignment cost via ICP-like fine alignment

- Repeat until convergence:
  - Find putative correspondences between edge point sets
  - Recover inlier correspondences and estimate camera parameters
  - Update viewpoint

# Putative correspondences



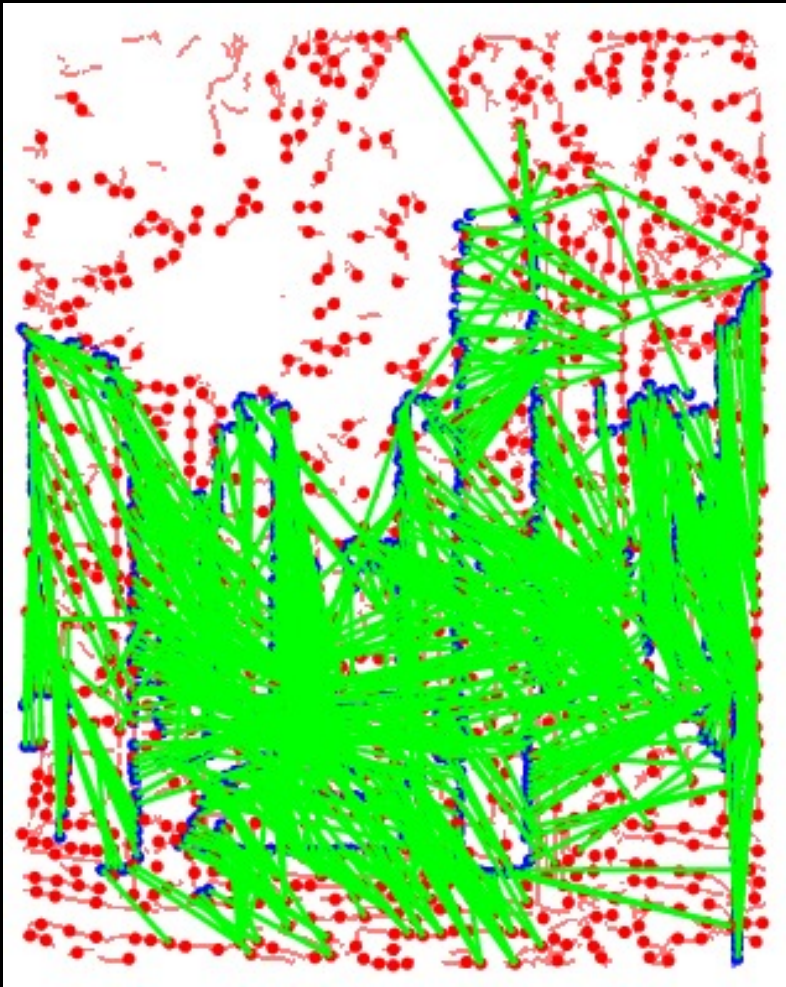
Painting contours – red  
3D model contours – blue



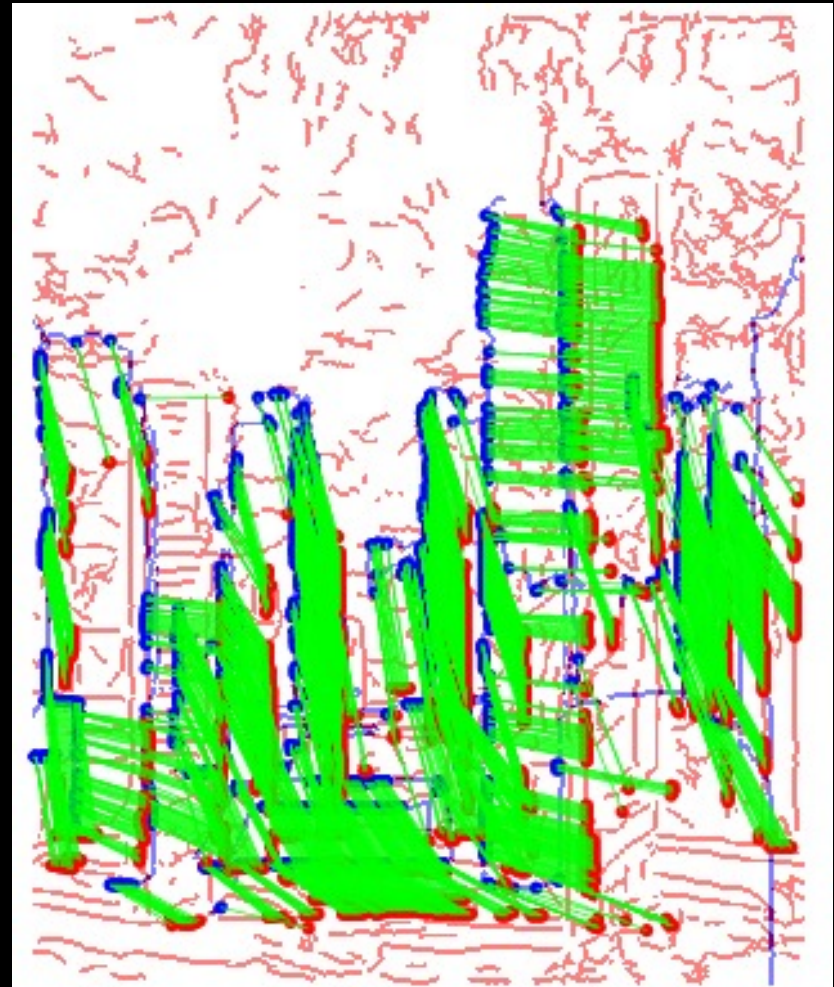
Shape context sample points  
and putative correspondences



# Inlier correspondences



Putative correspondences



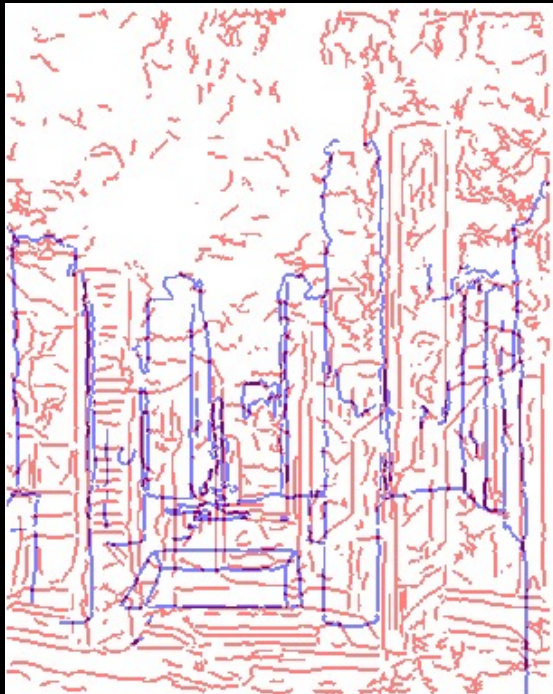
Dense inliers

# Recovering inlier correspondences and estimating transformation

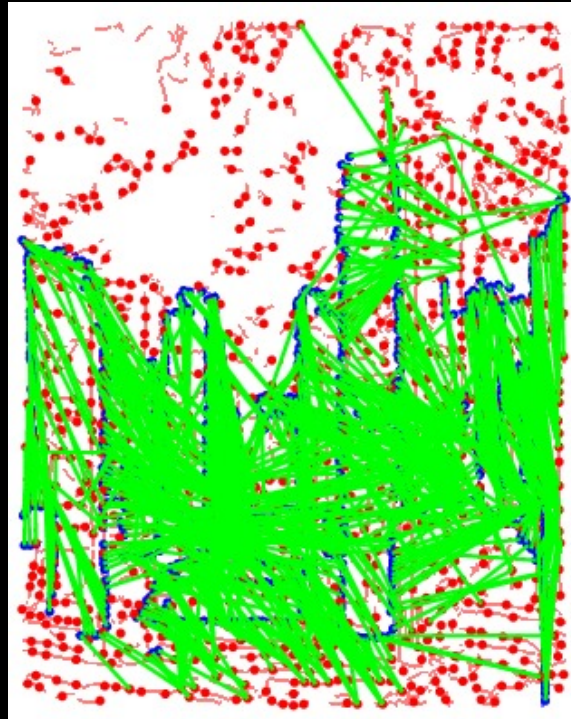
- PnP problem: recover camera center and rotation using RANSAC (assume fixed intrinsic parameters)
  - Sample 3 putative correspondences at each RANSAC iteration
  - Up to 4 solutions possible
- Validate transformation using dense cost function
- Use dense set of inliers to estimate all camera parameters
- Reduce search space for correspondences with each iteration



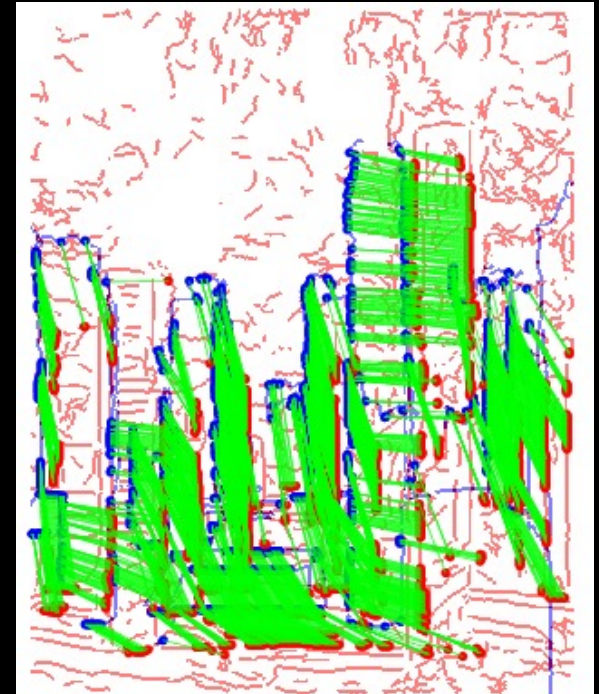
# Iteration 1



Painting and 3D model  
contours

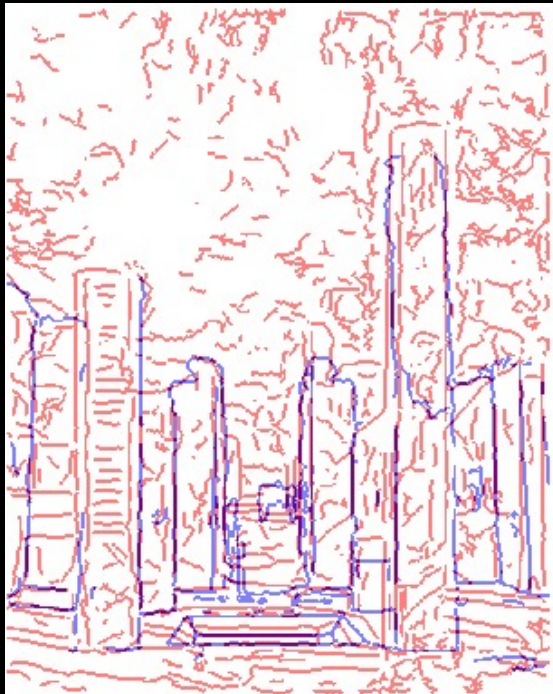


Shape context sample  
points and putative  
correspondences



Dense inliers

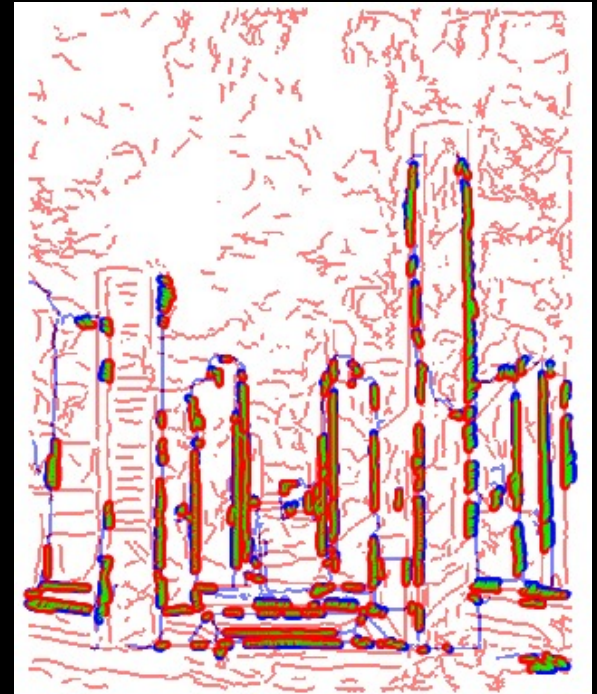
# Iteration 2



Painting and 3D model  
contours



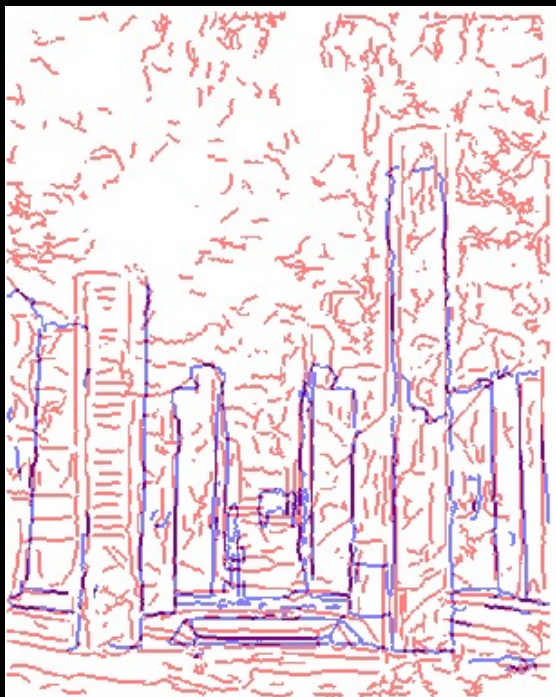
Shape context sample  
points and putative  
correspondences



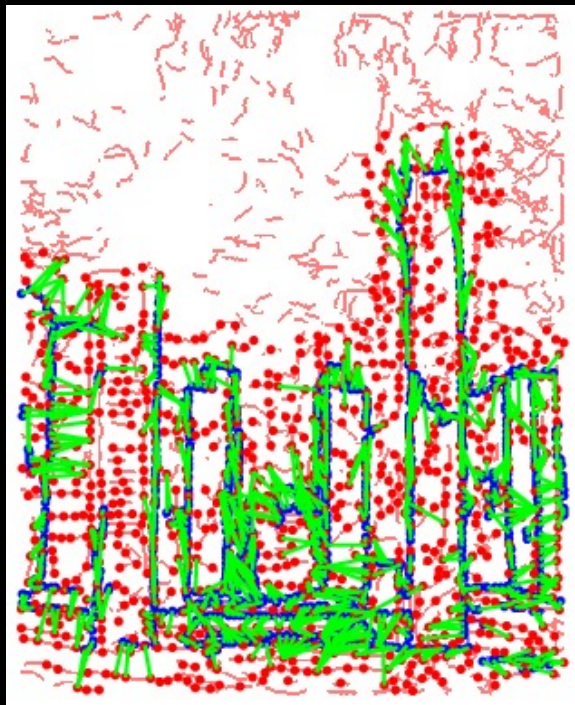
Dense inliers



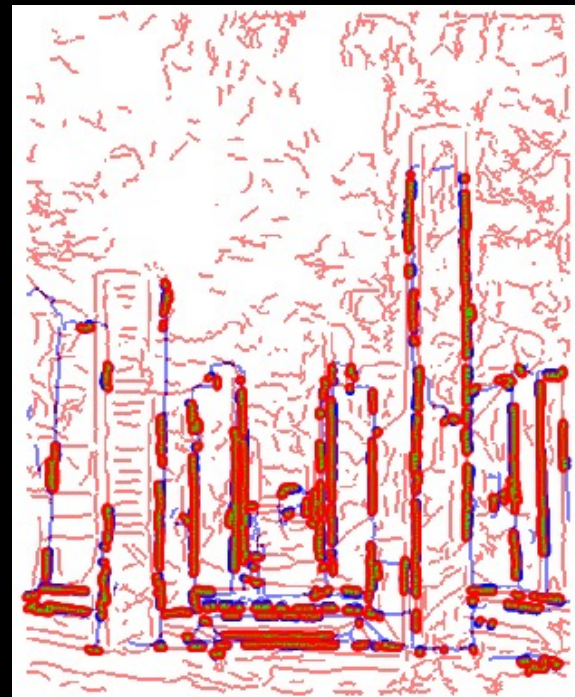
# Iteration 3



Painting and 3D model  
contours

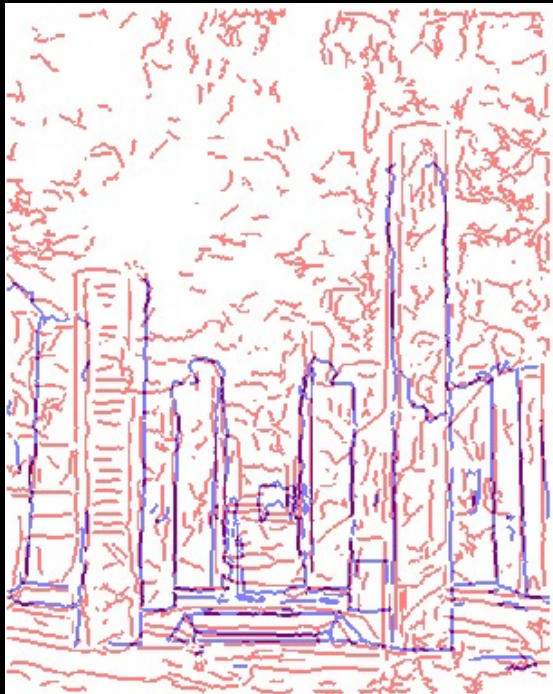


Shape context sample  
points and putative  
correspondences

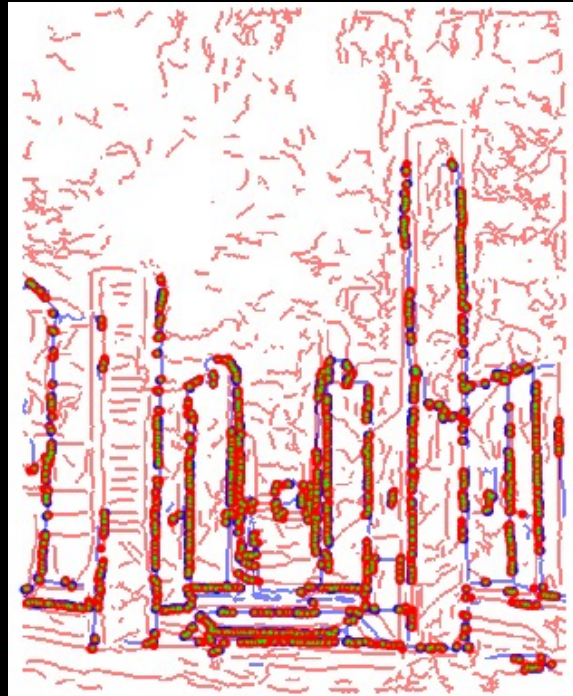


Dense inliers

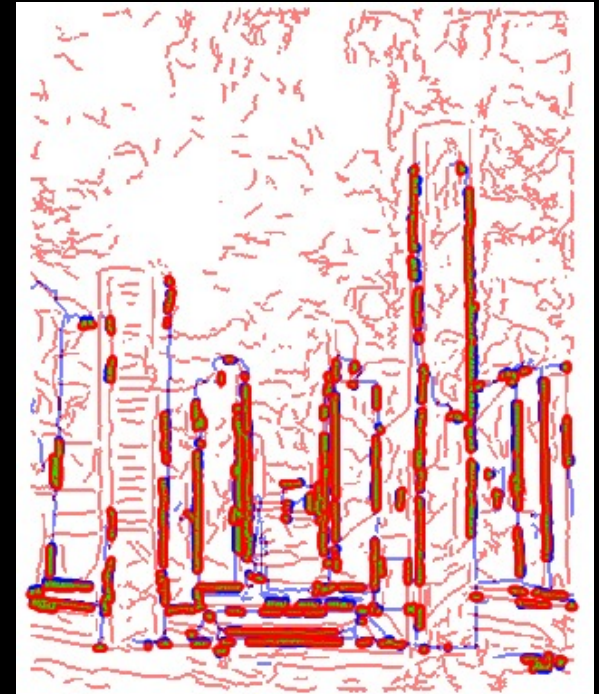
# Iteration 7



Painting and 3D model  
contours



Shape context sample  
points and putative  
correspondences



Dense inliers



# Alignment results



Painting

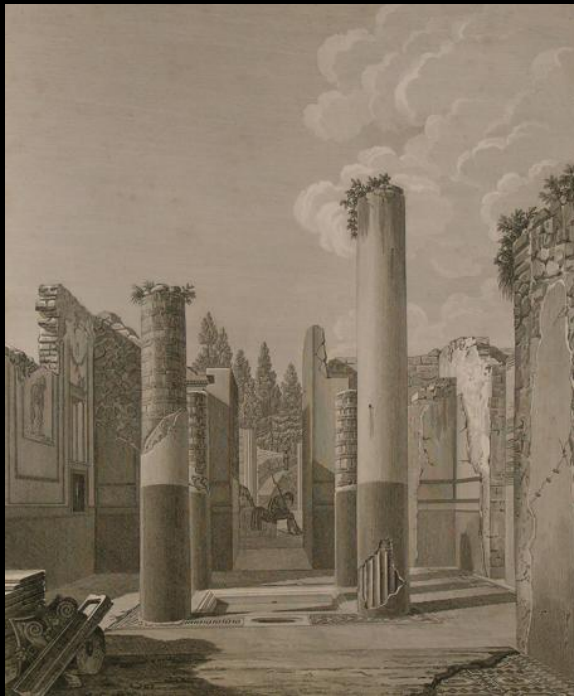


3D model contours  
overlaid

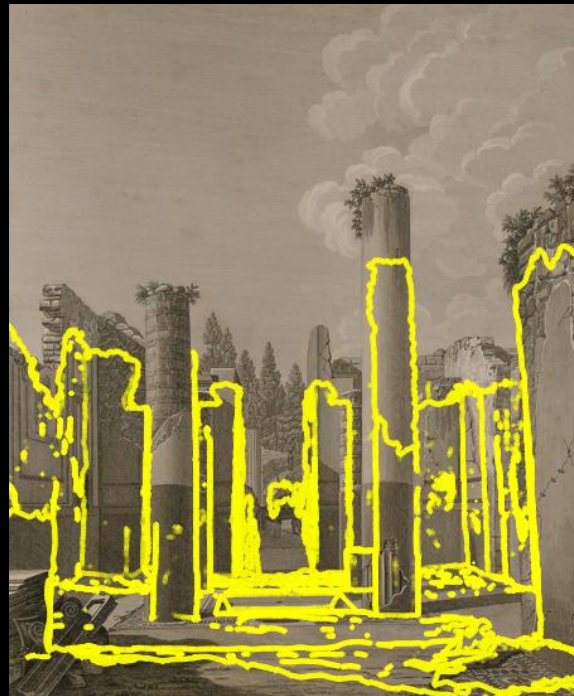


Rendering from  
3D model

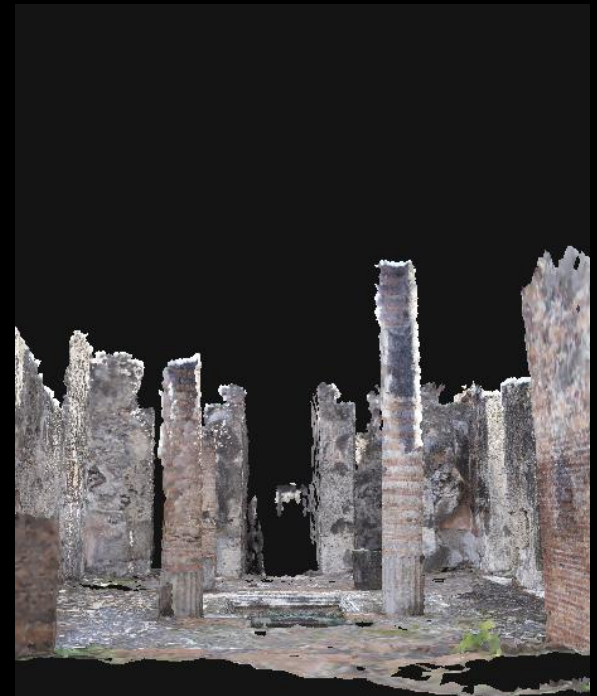
# Alignment results



Painting



3D model contours  
overlaid



Rendering from  
3D model



Painting



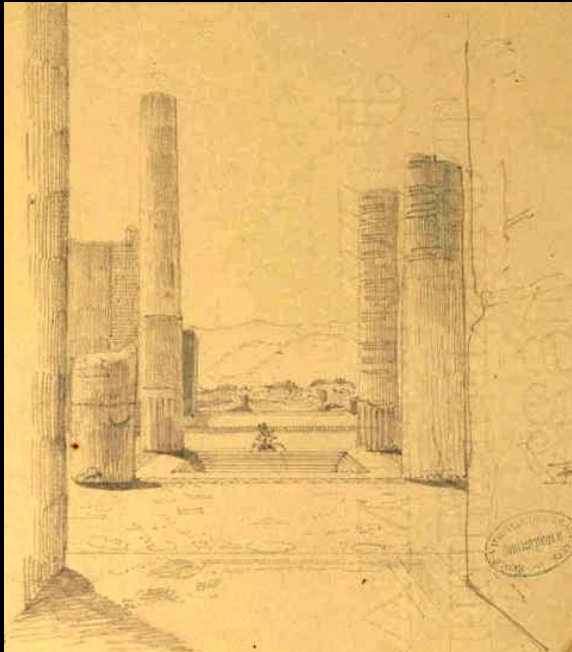
3D model contours  
overlaid



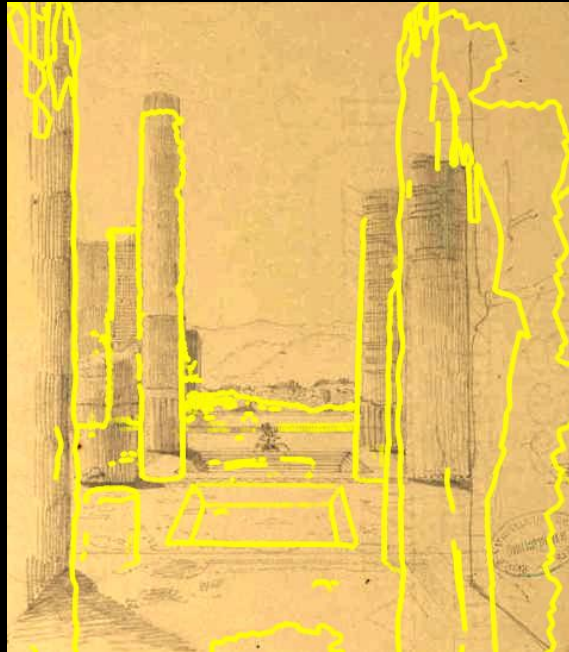
Rendering from  
3D model



# Failure cases



Painting



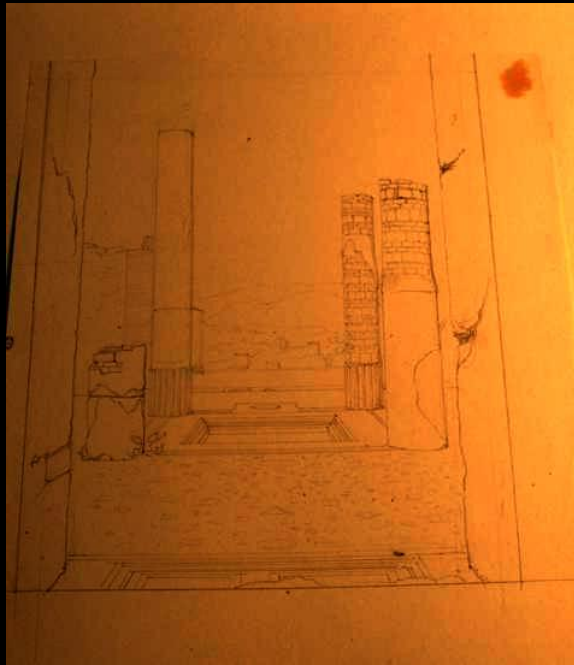
3D model contours  
overlaid



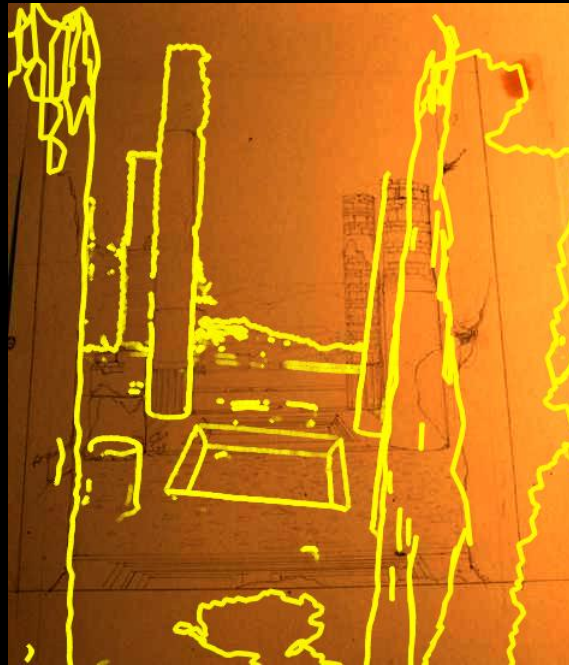
Rendering from  
3D model



# Failure cases



Painting



3D model contours  
overlaid



Rendering from  
3D model

# Quantitative evaluation

Reprojection error, in pixels  
(percentage of diagonal length)

Res	476x600
GT	3.60 (0.47)
Alg	5.65 (0.74)



Res - Resolution

GT – Ground truth error

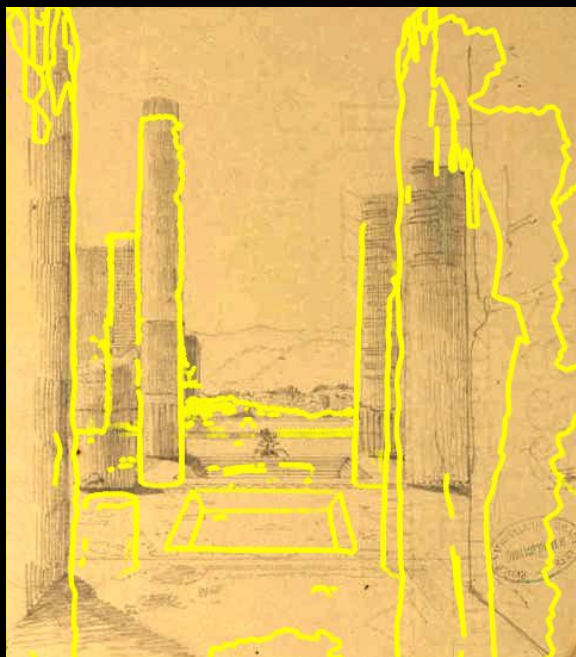
Alg – Algorithm error



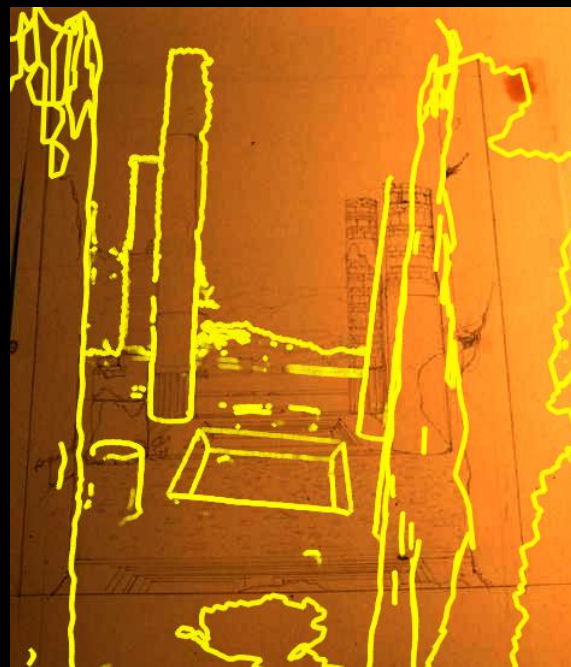
# Quantitative evaluation

Reprojection error, in pixels  
(percentage of diagonal length)

Res	476x600	480x547	475x550
GT	3.60 (0.47)	2.52 (0.35)	4.05 (0.56)
Alg	5.65 (0.74)	36.95 (5.08)	46.03 (6.33)



Res - Resolution



GT – Ground truth error

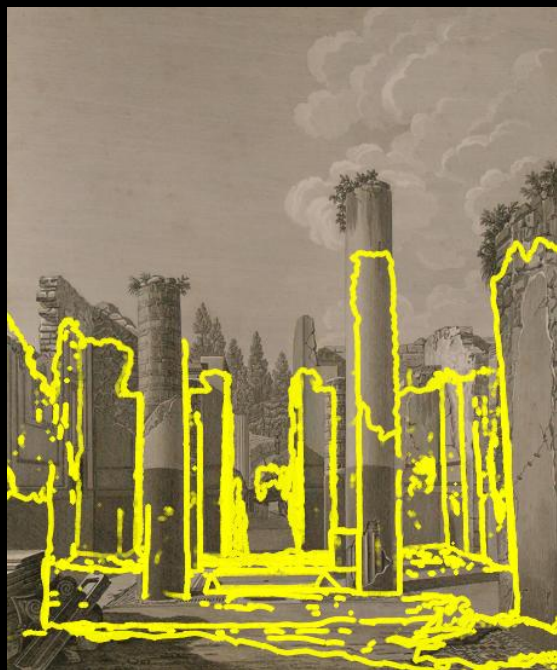
Alg – Algorithm error

# Quantitative evaluation

Reprojection error, in pixels  
(percentage of diagonal length)

Res	476x600	480x547	475x550	456x550	474x578	459x550
GT	3.60 (0.47)	2.52 (0.35)	4.05 (0.56)	7.88 (1.10)	8.16 (1.09)	9.79 (1.37)
Alg	5.65 (0.74)	36.95 (5.08)	46.03 (6.33)	12.06 (1.69)	17.80 (2.38)	18.25 (2.55)

Ground truth has higher error; possible drawing error



Res - Resolution

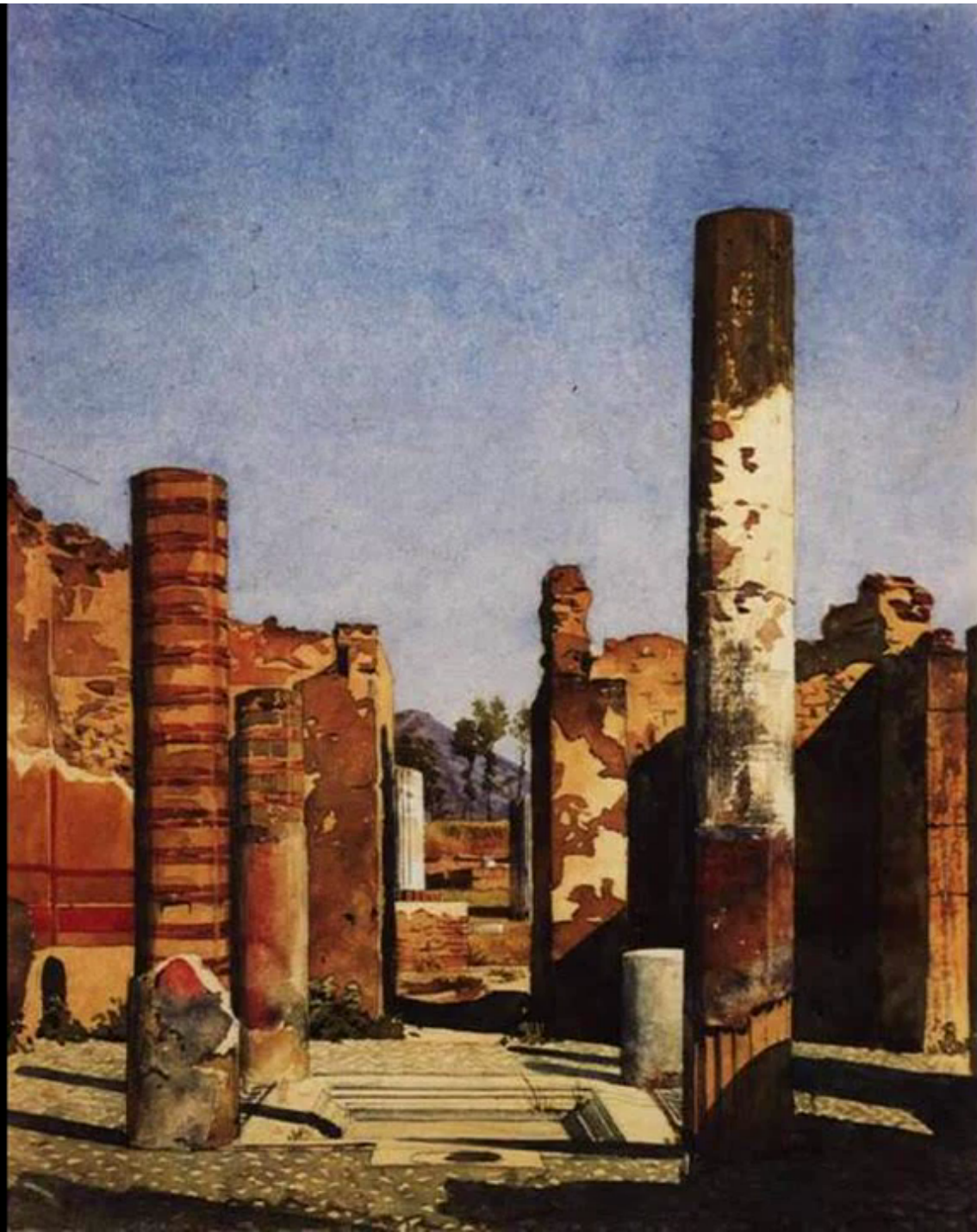


GT – Ground truth error



Alg – Algorithm error





# Conclusions

- Shown successful alignment of historical paintings of an archaeological site to a **noisy 3D model** constructed from modern photographs
- System handles drastic changes in appearance, which is difficult for current systems relying on local feature matching
- 3D model allows alignment from unseen viewpoints
- Scaling up to additional sites may require successful large-scale painting retrieval



# Project page

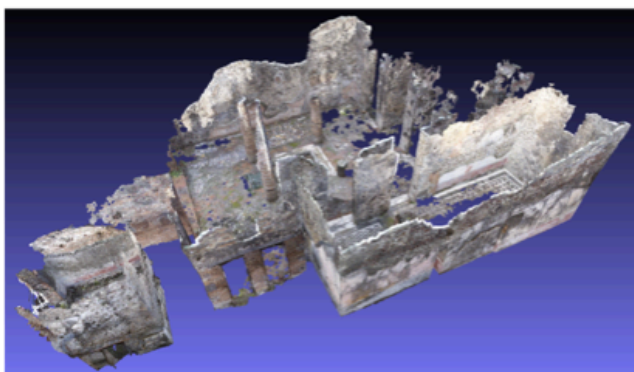
<http://www.di.ens.fr/willow/research/paintingalignment/>

www.di.ens.fr/willow/research/paintingalignment/

Google

## Automatic Alignment of Paintings and Photographs Depicting a 3D Scene

Bryan C. Russell, Josef Sivic, Jean Ponce, H  l  ne Dessales



3D model automatically obtained from  
photographs



Input painting



Automatic alignment of  
painting to 3D model

### Overview

This paper addresses the problem of automatically aligning historical architectural paintings with 3D models obtained using multi-view stereo technology from modern photographs. This is a challenging task because of the variations in appearance, geometry, color and texture due to environmental changes over time, the nonphotorealistic nature of architectural paintings, and differences in the viewpoints used by the painters and photographers. Our alignment procedure consists of two novel aspects: (i) we combine the gist descriptor with the view-synthesis/retrieval of Irshara et al. to obtain a coarse alignment of the painting to the 3D model, and (ii) we have developed an ICP-like viewpoint refinement procedure, where 3D surface orientation