Consistent Multi-View Reconstruction from Epipolar Geometries with Outliers

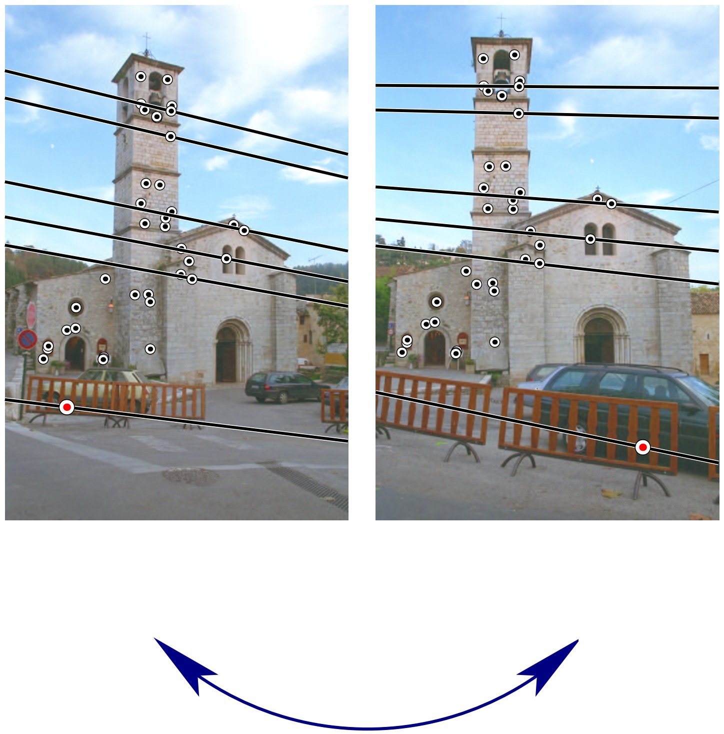
Daniel Martinec and Tomáš Pajdla

Center for Machine Perception
Department of Cybernetics, Faculty Electrical Engineering
Czech Technical University, Prague

What is it about. . .

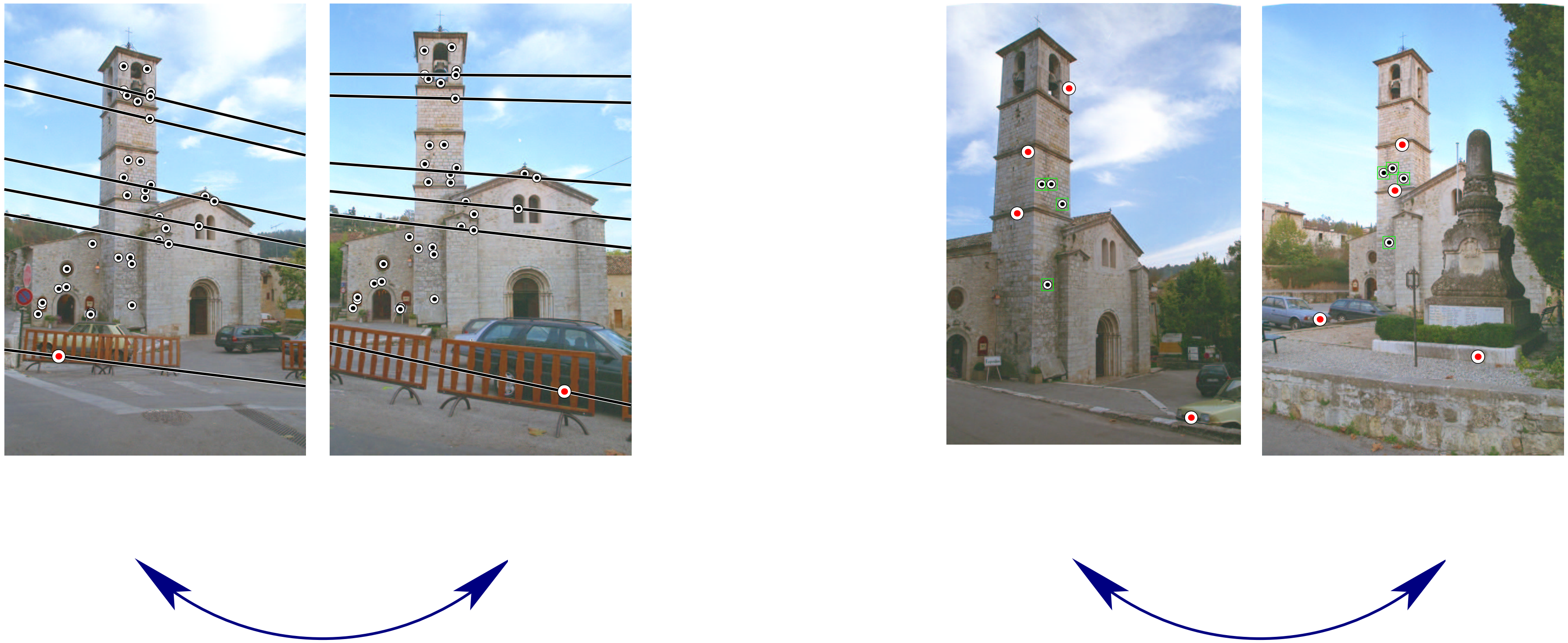
. . .

What is it about. . .



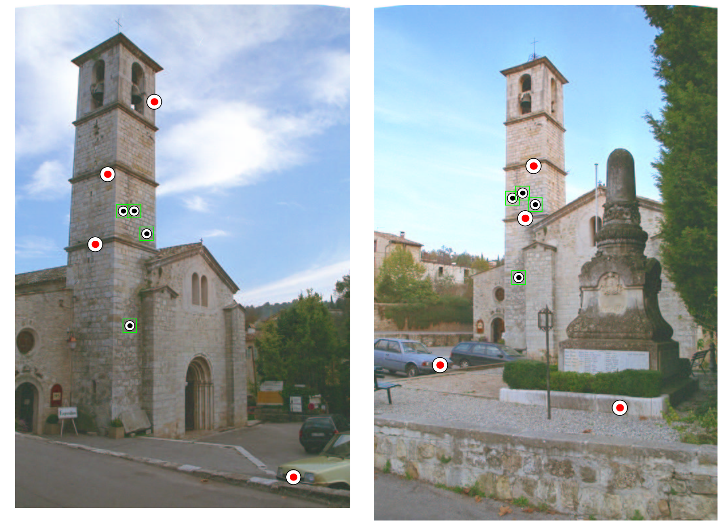
Epipolar geometries between image pairs avialable (Matas & al BMVC'02)

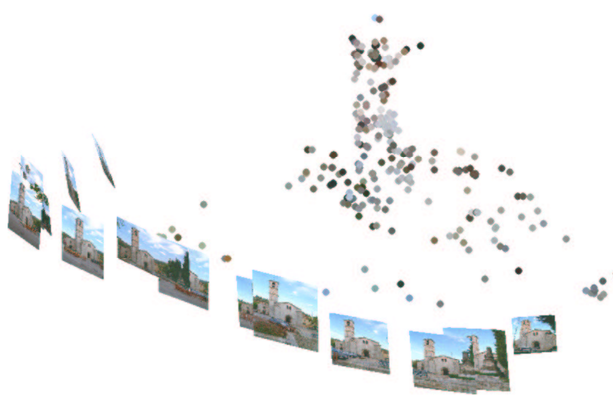
What is it about. . .



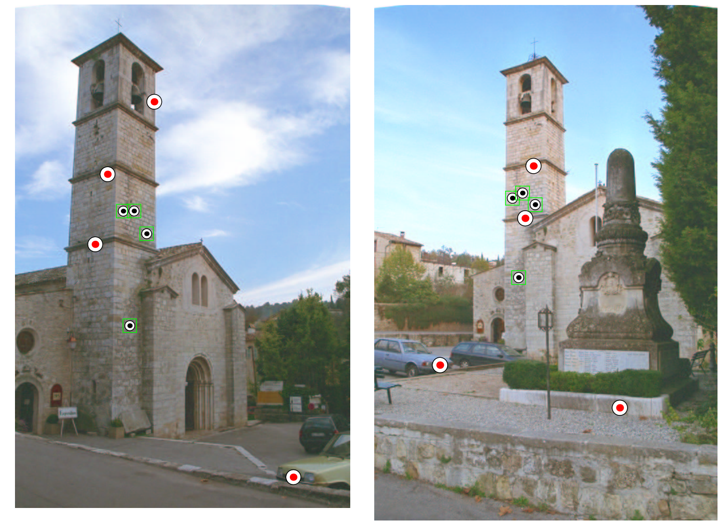
Epipolar geometries between image pairs avialable (Matas & al BMVC'2002)

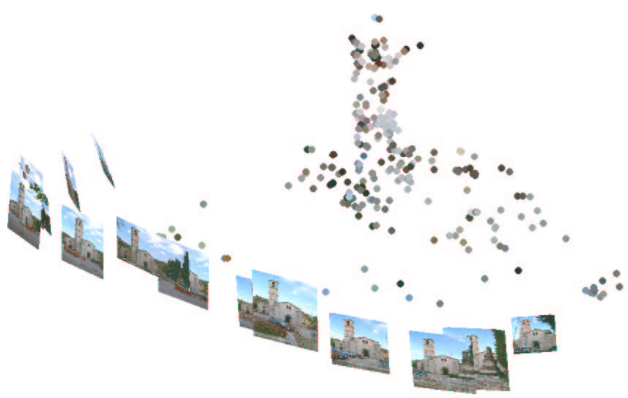
What is it about. . .





What is it about. . .



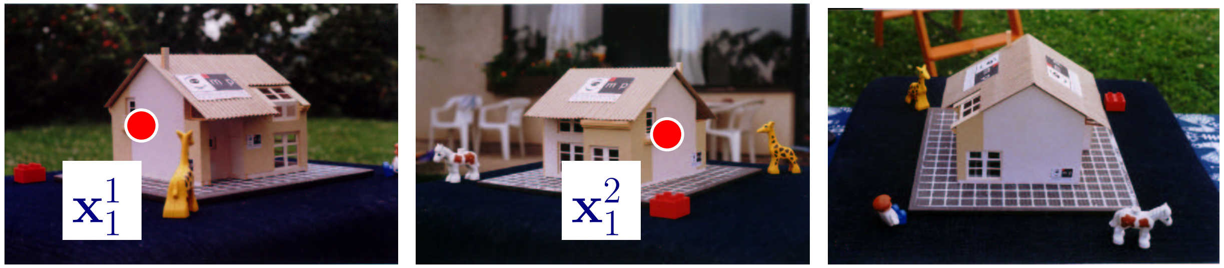


Goal: projective reconstruction consistent with all images

constructed from pairwise epipolar geometries

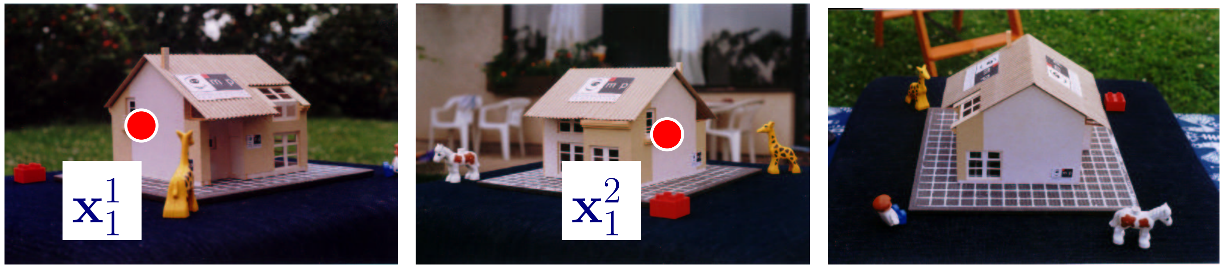
Problems

1. occlusions



Problems

1. occlusions



2. outliers

(may satisfy the epipolar geometry)

Contribution

- Technique for construction of
  consistent projective reconstruction from pairwise epipolar geometries
  - perspective cameras
  - occlusions
  - outliers

Contribution

- Technique for construction of
  consistent projective reconstruction from pairwise epipolar geometries
  - perspective cameras } Martinec & Pajdla ECCV'2002
  - occlusions } Martinec & Pajdla ECCV'2002
  - outliers — Martinec & Pajdla PCV'2002

Outline

INTRODUCTION

- Problem formulation, previous work

PART I: Perspective Cameras & Occlusions

- Projective depth estimation
- Filling the missing elements
- Experiments

PART II: Outlier Detection

- New idea & algorithm
- Experiments

Previous Work

[1] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization method. In IJCV(9)2, 1992.

Previous Work

[1] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization method. In IJCV(9)2, 1992.

[2] D. Jacobs. Linear fitting with missing data: Applications to structure from motion and to characterizing intensity images. In CVPR, 1997.

Previous Work

[1] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization method. In IJCV(9)2, 1992.

[2] D. Jacobs. Linear fitting with missing data: Applications to structure from motion and to characterizing intensity images. In CVPR, 1997.

[3] P. Sturm and B. Triggs. A factorization based algorithm for multi-image projective structure and motion. In ECCV, 1996.

Previous Work

[1] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization method. In IJCV(9)2, 1992.

[2] D. Jacobs. Linear fitting with missing data: Applications to structure from motion and to characterizing intensity images. In CVPR, 1997.

[3] P. Sturm and B. Triggs. A factorization based algorithm for multi-image projective structure and motion. In ECCV, 1996.

[4] D. Q. Huynh and A. Heyden. Outlier detection in video sequences under affine projection. In CVPR, 2001.

PART I: Perspective Cameras & Occlusions

Review of ECCV'2002

Problem Formulation & Solution

Problem Formulation & Solution

Perspective camera projection: $\lambda_p^i \underbrace{\mathbf{x}_p^i}_{3\times 1} = \underbrace{\mathrm{P}^i}_{3\times 4} \underbrace{\mathbf{X}_p}_{4\times 1}$

Problem Formulation & Solution

Perspective camera projection: $\lambda_p^i \underbrace{\mathbf{x}_p^i}_{3\times 1} = \underbrace{\mathtt{P}^i}_{3\times 4} \underbrace{\mathbf{X}_p}_{4\times 1}$

$$\underbrace{\begin{bmatrix} \lambda_1^1\mathbf{x}_1^1 & \lambda_2^1\mathbf{x}_2^1 & \dots & \lambda_n^1\mathbf{x}_n^1 \\ \times & \lambda_2^2\mathbf{x}_2^2 & & \times \\ \vdots & & \ddots & \vdots \\ \lambda_1^m\mathbf{x}_1^m & \times & \dots & \lambda_n^m\mathbf{x}_n^m \end{bmatrix}}_{\mathtt{R}} = \underbrace{\begin{bmatrix} \mathtt{P}^1 \\ \mathtt{P}^2 \\ \vdots \\ \mathtt{P}^m \end{bmatrix}}_{3m \times 4} \underbrace{[\mathbf{X}_1 \mathbf{X}_2 \dots \mathbf{X}_n]}_{4 \times n}$$

rescaled measurement matrix

motion

structure

Problem Formulation & Solution

Perspective camera projection: $\underbrace{\lambda_p^i \mathbf{x}_p^i}_{3\times 1} = \underbrace{\mathtt{P}^i}_{3\times 4} \underbrace{\mathbf{X}_p}_{4\times 1}$

$$\underbrace{\begin{bmatrix} \lambda_1^1 \mathbf{x}_1^1 & \lambda_2^1 \mathbf{x}_2^1 & \dots & \lambda_n^1 \mathbf{x}_n^1 \\ \times & \lambda_2^2 \mathbf{x}_2^2 & & \times \\ \vdots & & \ddots & \vdots \\ \lambda_1^m \mathbf{x}_1^m & \times & \dots & \lambda_n^m \mathbf{x}_n^m \end{bmatrix}}_{\mathtt{R}} = \underbrace{\begin{bmatrix} \mathtt{P}^1 \\ \mathtt{P}^2 \\ \vdots \\ \mathtt{P}^m \end{bmatrix}}_{3m \times 4} \underbrace{[\mathbf{X}_1 \mathbf{X}_2 \dots \mathbf{X}_n]}_{4 \times n}$$

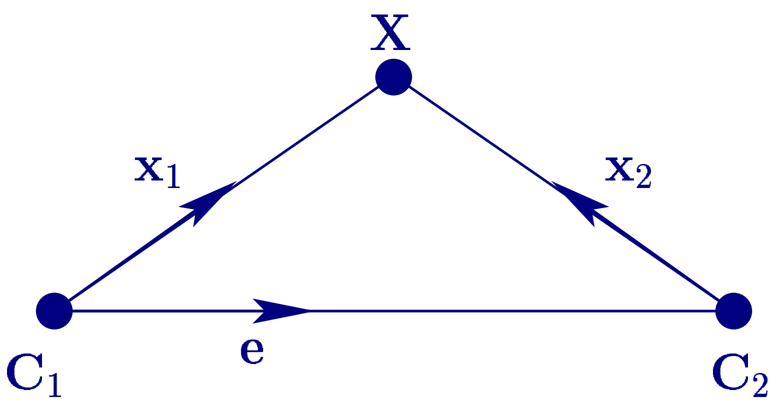
rescaled measurement matrix — motion — structure

1. Estimate (some) λ_j^i using, e.g., the epipolar geometry.
2. Fill $\times$.

   Repeat 1 and 2 until $\mathtt{R}$ is complete.
3. Factorize complete $\mathtt{R}$.

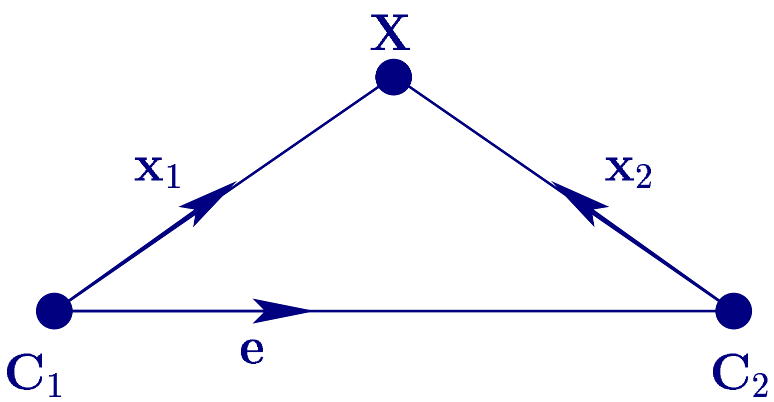
Estimation of λ_p^i (Sturm & Triggs)

uses the epipolar geometry



Estimation of λ_p^i (Sturm & Triggs)

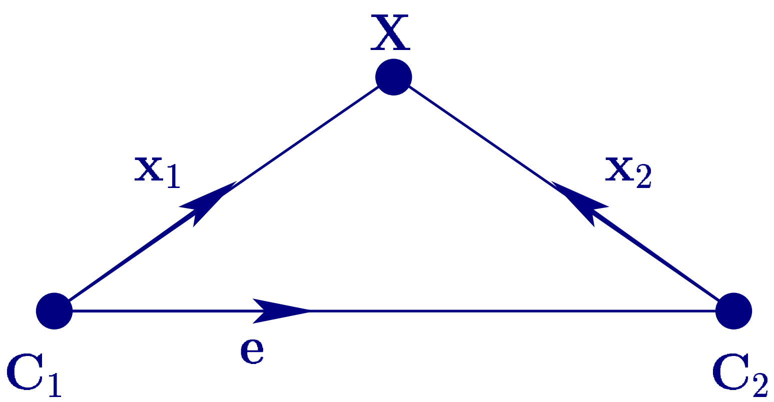
uses the epipolar geometry



$$\lambda_1 \mathbf{x}_1 = \tau \mathbf{e} + \lambda_2 \mathbf{x}_2$$

Estimation of λ_p^i (Sturm & Triggs)

uses the epipolar geometry

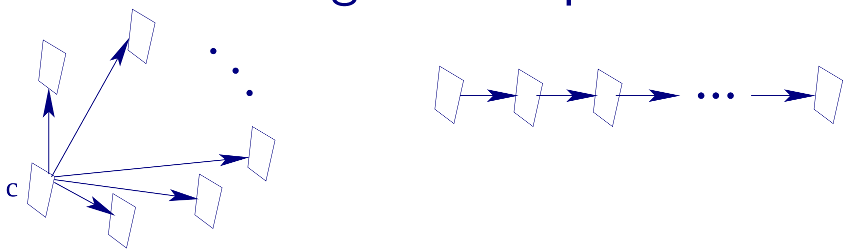


$$\lambda_1 \mathbf{x}_1 = \tau \mathbf{e} + \lambda_2 \mathbf{x}_2$$

$$\lambda_p^i = \frac{(\mathbf{e}^{ij} \wedge \mathbf{x}_p^i) \cdot (\mathtt{F}^{ij} \mathbf{x}_p^j)}{\| \mathbf{e}^{ij} \wedge \mathbf{x}_p^i \|^2} \lambda_p^j$$

- **Alternatives:**

  1. central image 2. sequence



Filling of $\times$

- Jacobs' algorithm: for affine cameras only ($\lambda_p^i = 1$)

Filling of $\times$

- Jacobs' algorithm: for affine cameras only ($\lambda_p^i = 1$)
  our algorithm: for perspective cameras (various λ_p^i)

Filling of $\times$

- Jacobs' algorithm: for affine cameras only ($\lambda_p^i = 1$)
  our algorithm: for perspective cameras (various λ_p^i)
- fill $\mathtt{R}$ so that rank $\mathtt{R} = 4$ (rank $\mathtt{R} = 4$ because $\mathtt{R} = \underbrace{\mathtt{P}}_{3m\times 4}\underbrace{\mathtt{X}}_{4\times n}$)

Filling of $\times$

- Jacobs' algorithm: for affine cameras only ($\lambda_p^i = 1$)
  our algorithm: for perspective cameras (various λ_p^i)
- fill $\mathtt{R}$ so that rank $\mathtt{R} = 4$ (rank $\mathtt{R} = 4$ because $\mathtt{R} = \underbrace{\mathtt{P}}_{3m\times 4}\underbrace{\mathtt{X}}_{4\times n}$)

a 4-tuple of "LI" columns

$$\mathtt{B}_k = \begin{bmatrix} ?\,\mathbf{x}_1^1 & \lambda_2^1\mathbf{x}_2^1 & \lambda_3^1\mathbf{x}_3^1 & \lambda_4^1\mathbf{x}_4^1 \\ \lambda_1^2\mathbf{x}_1^2 & \lambda_2^2\mathbf{x}_2^2 & \lambda_3^2\mathbf{x}_3^2 & \lambda_4^2\mathbf{x}_4^2 \\ \vdots & & & \\ \lambda_1^m\mathbf{x}_1^m & \lambda_2^m\mathbf{x}_2^m & \lambda_3^m\mathbf{x}_3^m & \times \end{bmatrix} = \begin{bmatrix} ?\,x_1^1 & \lambda_2^1x_2^1 & \lambda_3^1x_3^1 & \lambda_4^1x_4^1 \\ ?\,y_1^1 & \lambda_2^1y_2^1 & \lambda_3^1y_3^1 & \lambda_4^1y_4^1 \\ ?\,w_1^1 & \lambda_2^1w_2^1 & \lambda_3^1w_3^1 & \lambda_4^1w_4^1 \\ \lambda_1^2x_1^2 & \lambda_2^2x_2^2 & \lambda_3^2x_3^2 & \lambda_4^2x_4^2 \\ \lambda_1^2y_1^2 & \lambda_2^2y_2^2 & \lambda_3^2y_3^2 & \lambda_4^2y_4^2 \\ \lambda_1^2w_1^2 & \lambda_2^2w_2^2 & \lambda_3^2w_3^2 & \lambda_4^2w_4^2 \\ \vdots & & & \\ \lambda_1^mx_1^m & \lambda_2^mx_2^m & \lambda_3^mx_3^m & \times \\ \lambda_1^my_1^m & \lambda_2^my_2^m & \lambda_3^my_3^m & \times \\ \lambda_1^mw_1^m & \lambda_2^mw_2^m & \lambda_3^mw_3^m & \times \end{bmatrix}$$

linear hull of all possible fillings

$$\mathcal{B}_k = Span\left(\begin{bmatrix} \overbrace{\begin{matrix} 0 & x_1^1 \end{matrix}} & \lambda_2^1x_2^1 & \lambda_3^1x_3^1 & \overbrace{\begin{matrix} \lambda_4^1x_4^1 & 0 & 0 & 0 \end{matrix}} \\ 0 & y_1^1 & \lambda_2^1y_2^1 & \lambda_3^1y_3^1 & \lambda_4^1y_4^1 & 0 & 0 & 0 \\ 0 & w_1^1 & \lambda_2^1w_2^1 & \lambda_3^1w_3^1 & \lambda_4^1w_4^1 & 0 & 0 & 0 \\ \lambda_1^2x_1^2 & 0 & \lambda_2^2x_2^2 & \lambda_3^2x_3^2 & \lambda_4^2x_4^2 & 0 & 0 & 0 \\ \lambda_1^2y_1^2 & 0 & \lambda_2^2y_2^2 & \lambda_3^2y_3^2 & \lambda_4^2y_4^2 & 0 & 0 & 0 \\ \lambda_1^2w_1^2 & 0 & \lambda_2^2w_2^2 & \lambda_3^2w_3^2 & \lambda_4^2w_4^2 & 0 & 0 & 0 \\ \vdots & & & & & & & \\ \lambda_1^mx_1^m & 0 & \lambda_2^mx_2^m & \lambda_3^mx_3^m & 0 & \mathbf{1} & 0 & 0 \\ \lambda_1^my_1^m & 0 & \lambda_2^my_2^m & \lambda_3^my_3^m & 0 & 0 & \mathbf{1} & 0 \\ \lambda_1^mw_1^m & 0 & \lambda_2^mw_2^m & \lambda_3^mw_3^m & 0 & 0 & 0 & \mathbf{1} \end{bmatrix}\right)$$

Filling of $\times$

- Jacobs' algorithm: for affine cameras only ($\lambda_p^i = 1$)
  our algorithm: for perspective cameras (various λ_p^i)
- fill $\mathtt{R}$ so that rank $\mathtt{R} = 4$ (rank $\mathtt{R} = 4$ because $\mathtt{R} = \underbrace{\mathtt{P}}_{3m\times 4}\underbrace{\mathtt{X}}_{4\times n}$)

a 4-tuple of "LI" columns

$$\mathtt{B}_k = \begin{bmatrix} ?\ \mathbf{x}_1^1 & \lambda_2^1\mathbf{x}_2^1 & \lambda_3^1\mathbf{x}_3^1 & \lambda_4^1\mathbf{x}_4^1 \\ \lambda_1^2\mathbf{x}_1^2 & \lambda_2^2\mathbf{x}_2^2 & \lambda_3^2\mathbf{x}_3^2 & \lambda_4^2\mathbf{x}_4^2 \\ \vdots & & & \\ \lambda_1^m\mathbf{x}_1^m & \lambda_2^m\mathbf{x}_2^m & \lambda_3^m\mathbf{x}_3^m & \times \end{bmatrix} = \begin{bmatrix} ?\ x_1^1 & \lambda_2^1 x_2^1 & \lambda_3^1 x_3^1 & \lambda_4^1 x_4^1 \\ ?\ y_1^1 & \lambda_2^1 y_2^1 & \lambda_3^1 y_3^1 & \lambda_4^1 y_4^1 \\ ?\ w_1^1 & \lambda_2^1 w_2^1 & \lambda_3^1 w_3^1 & \lambda_4^1 w_4^1 \\ \lambda_1^2 x_1^2 & \lambda_2^2 x_2^2 & \lambda_3^2 x_3^2 & \lambda_4^2 x_4^2 \\ \lambda_1^2 y_1^2 & \lambda_2^2 y_2^2 & \lambda_3^2 y_3^2 & \lambda_4^2 y_4^2 \\ \lambda_1^2 w_1^2 & \lambda_2^2 w_2^2 & \lambda_3^2 w_3^2 & \lambda_4^2 w_4^2 \\ \vdots & & & \\ \lambda_1^m x_1^m & \lambda_2^m x_2^m & \lambda_3^m x_3^m & \times \\ \lambda_1^m y_1^m & \lambda_2^m y_2^m & \lambda_3^m y_3^m & \times \\ \lambda_1^m w_1^m & \lambda_2^m w_2^m & \lambda_3^m w_3^m & \times \end{bmatrix}$$

linear hull of all possible fillings

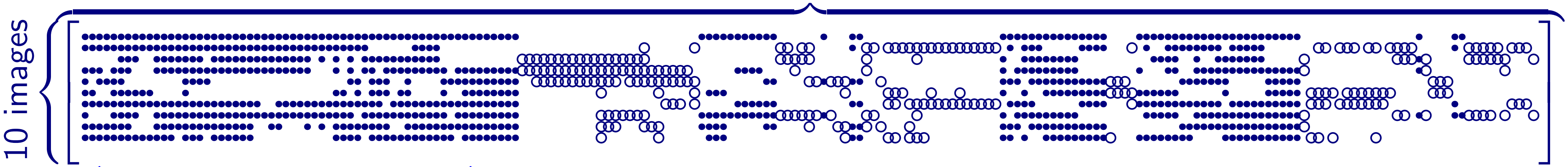
$$\mathcal{B}_k = Span\left(\begin{bmatrix} 0 & x_1^1 & \lambda_2^1 x_2^1 & \lambda_3^1 x_3^1 & \lambda_4^1 x_4^1 & 0 & 0 & 0 \\ 0 & y_1^1 & \lambda_2^1 y_2^1 & \lambda_3^1 y_3^1 & \lambda_4^1 y_4^1 & 0 & 0 & 0 \\ 0 & w_1^1 & \lambda_2^1 w_2^1 & \lambda_3^1 w_3^1 & \lambda_4^1 w_4^1 & 0 & 0 & 0 \\ \lambda_1^2 x_1^2 & 0 & \lambda_2^2 x_2^2 & \lambda_3^2 x_3^2 & \lambda_4^2 x_4^2 & 0 & 0 & 0 \\ \lambda_1^2 y_1^2 & 0 & \lambda_2^2 y_2^2 & \lambda_3^2 y_3^2 & \lambda_4^2 y_4^2 & 0 & 0 & 0 \\ \lambda_1^2 w_1^2 & 0 & \lambda_2^2 w_2^2 & \lambda_3^2 w_3^2 & \lambda_4^2 w_4^2 & 0 & 0 & 0 \\ \vdots & & & & & & & \\ \lambda_1^m x_1^m & 0 & \lambda_2^m x_2^m & \lambda_3^m x_3^m & 0 & \mathbf{1} & 0 & 0 \\ \lambda_1^m y_1^m & 0 & \lambda_2^m y_2^m & \lambda_3^m y_3^m & 0 & 0 & \mathbf{1} & 0 \\ \lambda_1^m w_1^m & 0 & \lambda_2^m w_2^m & \lambda_3^m w_3^m & 0 & 0 & 0 & \mathbf{1} \end{bmatrix}\right)$$

- $\mathcal{B}_k$ contains filled $\mathtt{B}_k$ (also filled $\mathtt{R}$) $\longrightarrow$ constraint on $\mathtt{R}$
- intersection of many $\mathcal{B}_k$ $\longrightarrow$ fill $\mathtt{R}$

Wide Base-Line Stereo

LM = lin. method, BA = bundle adjustment

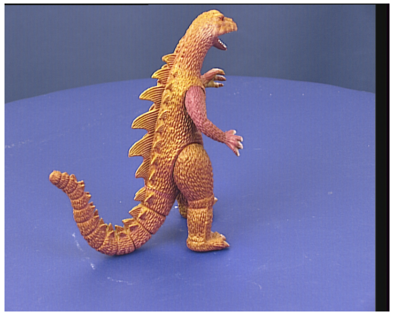
	Scene *House*	10 images [2952×2003]	
	Point detection	manual, 203 points in space	
	Depth estimation	central image No. 1	
	Amount of missing data		**47.83 %**
LM	Mean error per image point [pxl]		**3.91**
LM + BA			**1.44**

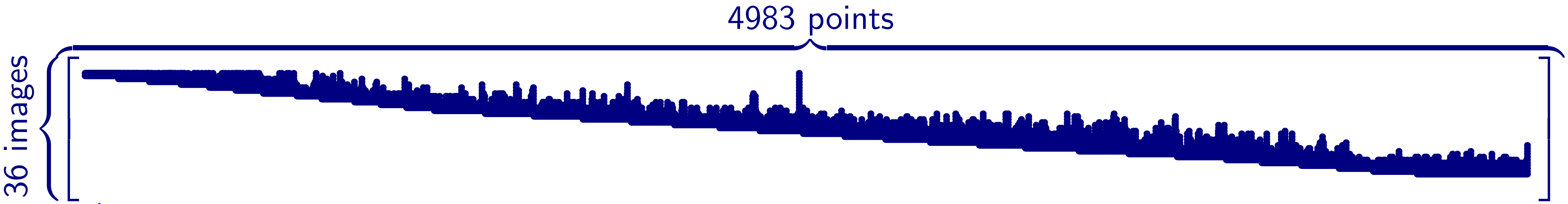


"●" $\lambda_p^i \longleftarrow \mathtt{F}$ (75.7 %), "○" $\lambda_p^i \longleftarrow \mathcal{B}$ (24.3 %), " " occlusion

Dense Sequence (Hannover)

	Scene *Dinosaur (Oxford)*	36 images [720×576]
	Point detection	Harris' operator, 4983 points in space
	Depth estimation	sequence
	Amount of missing data	**90.84 %**
LM	Mean error per image point [pxl]	**1.76**
LM + BA		**0.64**





"•" $\lambda_p^i \longleftarrow$ F (100.0 %), " " occlusion

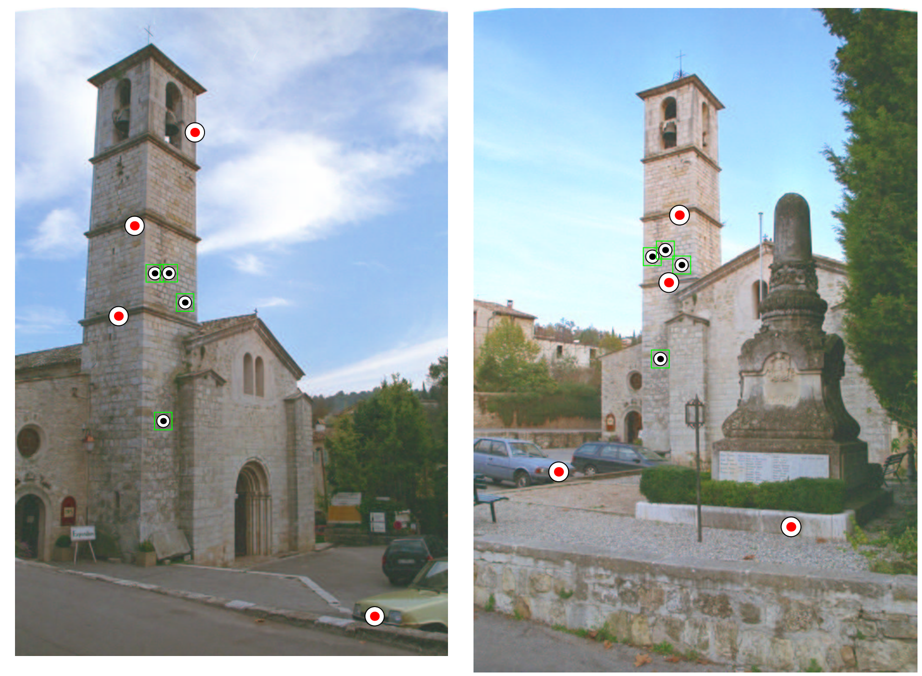
PART II: Outlier Detection

Review of PCV'2002

Correspondences from the Epipolar Geometry

some outliers satisfy the epipolar geometry

epipolar geometry from a dominant plane & outliers



Problem Formulation & Related Work

Perspective camera projection: $\lambda_p^i \underbrace{\mathbf{x}_p^i}_{3\times 1} = \underbrace{\mathtt{P}^i}_{3\times 4} \underbrace{\mathbf{X}_p}_{4\times 1}$

$\mathbf{x}_p^i$ . . . outliers

$$\underbrace{\begin{bmatrix} \lambda_1^1 \mathbf{x}_1^1 & \lambda_2^1 \mathbf{x}_2^1 & \dots & \lambda_n^1 \mathbf{x}_n^1 \\ \times & \lambda_2^2 \mathbf{x}_2^2 & & \times \\ \vdots & & \ddots & \vdots \\ \lambda_1^m \mathbf{x}_1^m & \times & \dots & \lambda_n^m \mathbf{x}_n^m \end{bmatrix}}_{\mathtt{R}} = \underbrace{\begin{bmatrix} \mathtt{P}^1 \\ \mathtt{P}^2 \\ \vdots \\ \mathtt{P}^m \end{bmatrix}}_{3m \times 4} \underbrace{[\mathbf{X}_1 \mathbf{X}_2 \dots \mathbf{X}_n]}_{4 \times n}$$

rescaled measurement matrix | motion | structure

Outlier Detection — Main Idea

Pairwise epipolar geometry $\longrightarrow$ not many outliers

Outlier Detection — Main Idea

Pairwise epipolar geometry $\longrightarrow$ not many outliers

Assumption

One consistent structure & random independent outliers

Outlier Detection — Main Idea

Pairwise epipolar geometry $\longrightarrow$ not many outliers

Assumption

One consistent structure & random independent outliers

Main Idea

Minimal configurations of points in triples of images are sufficient to validate inliers reliably.

Outlier Detection — Main Idea

Pairwise epipolar geometry $\longrightarrow$ not many outliers

Assumption

One consistent structure & random independent outliers

Main Idea

Minimal configurations of points in triples of images are sufficient to validate inliers reliably.

$\Longrightarrow$

1. Choose a triple of images and six common points in them in random $\longrightarrow \mathcal{T}$.
2. If $\mathcal{T}$ consistent with enough correspondences $\longrightarrow$ validate points not used for estimation $\mathcal{T}$ as inliers.

Repeat 1 and 2 until there is enough inliers.

Outlier Detection — Main Idea

Pairwise epipolar geometry $\longrightarrow$ not many outliers

Assumption

One consistent structure & random independent outliers

Main Idea

Minimal configurations of points in triples of images are sufficient to validate inliers reliably.

$\Longrightarrow$

1. Choose a triple of images and six common points in them in random $\longrightarrow \mathcal{T}$.
2. If $\mathcal{T}$ consistent with enough correspondences $\longrightarrow$ validate points not used for estimation $\mathcal{T}$ as inliers.

Repeat 1 and 2 until there is enough inliers.

$\longrightarrow$ estimate reconstruction using our method for occlusions [ECCV'2002] $(\mathbf{x}_p^i \rightarrow \times)$

Consistent Multi-View Reconstruction — Algorithm

1. Obtain **pairwise matches** by estimating epipolar geometries on (all) image pairs.

Consistent Multi-View Reconstruction — Algorithm

1. Obtain **pairwise matches** by estimating epipolar geometries on (all) image pairs.
2. Build the **measurement matrix** by joining the overlapping matches and ignoring the conflicting ones.

Consistent Multi-View Reconstruction — Algorithm

1. Obtain **pairwise matches** by estimating epipolar geometries on (all) image pairs.
2. Build the **measurement matrix** by joining the overlapping matches and ignoring the conflicting ones.
3. Find the **initial set of inliers** using validation by $\mathcal{T}$ (previous slide).

Consistent Multi-View Reconstruction — Algorithm

1. Obtain **pairwise matches** by estimating epipolar geometries on (all) image pairs.
2. Build the **measurement matrix** by joining the overlapping matches and ignoring the conflicting ones.
3. Find the **initial set of inliers** using validation by $\mathcal{T}$ (previous slide).
4. **Reconstruction** from inliers using [ECCV'2002].

Consistent Multi-View Reconstruction — Algorithm

1. Obtain **pairwise matches** by estimating epipolar geometries on (all) image pairs.
2. Build the **measurement matrix** by joining the overlapping matches and ignoring the conflicting ones.
3. Find the **initial set of inliers** using validation by $\mathcal{T}$ (previous slide).
4. **Reconstruction** from inliers using [ECCV'2002].
5. **Check** all data whether it is consistent with the reconstruction.

Consistent Multi-View Reconstruction — Algorithm

1. Obtain **pairwise matches** by estimating epipolar geometries on (all) image pairs.
2. Build the **measurement matrix** by joining the overlapping matches and ignoring the conflicting ones.
3. Find the **initial set of inliers** using validation by $\mathcal{T}$ (previous slide).
4. **Reconstruction** from inliers using [ECCV'2002].
5. **Check** all data whether it is consistent with the reconstruction.

   In each column, p, of $\mathtt{R}$:

   (a) Random triple of image points, $\mathtt{P} \longrightarrow \mathbf{X}_p$

   (b) If repr. error small $\longrightarrow$ inliers

   Repeat (a) and (b) until the column is sufficiently sampled

Consistent Multi-View Reconstruction — Algorithm

1. Obtain **pairwise matches** by estimating epipolar geometries on (all) image pairs.
2. Build the **measurement matrix** by joining the overlapping matches and ignoring the conflicting ones.
3. Find the **initial set of inliers** using validation by $\mathcal{T}$ (previous slide).
4. **Reconstruction** from inliers using [ECCV'2002].
5. **Check** all data whether it is consistent with the reconstruction.

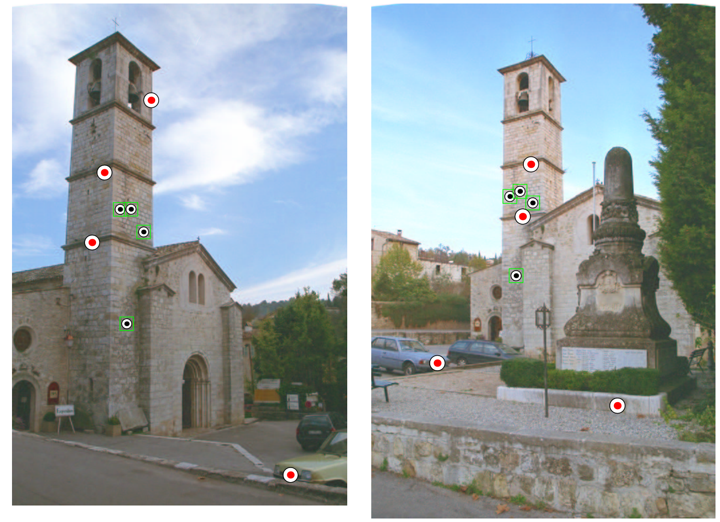
   In each column, p, of $\mathtt{R}$:

   (a) Random triple of image points, $\mathtt{P} \longrightarrow \mathbf{X}_p$

   (b) If repr. error small $\longrightarrow$ inliers

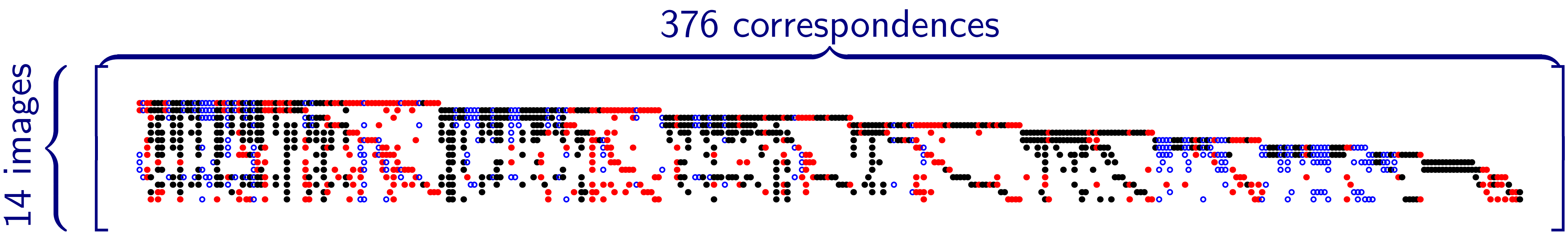
   Repeat (a) and (b) until the column is sufficiently sampled
6. **Iteration** of Steps 4 and 5 while any new inlier appeared

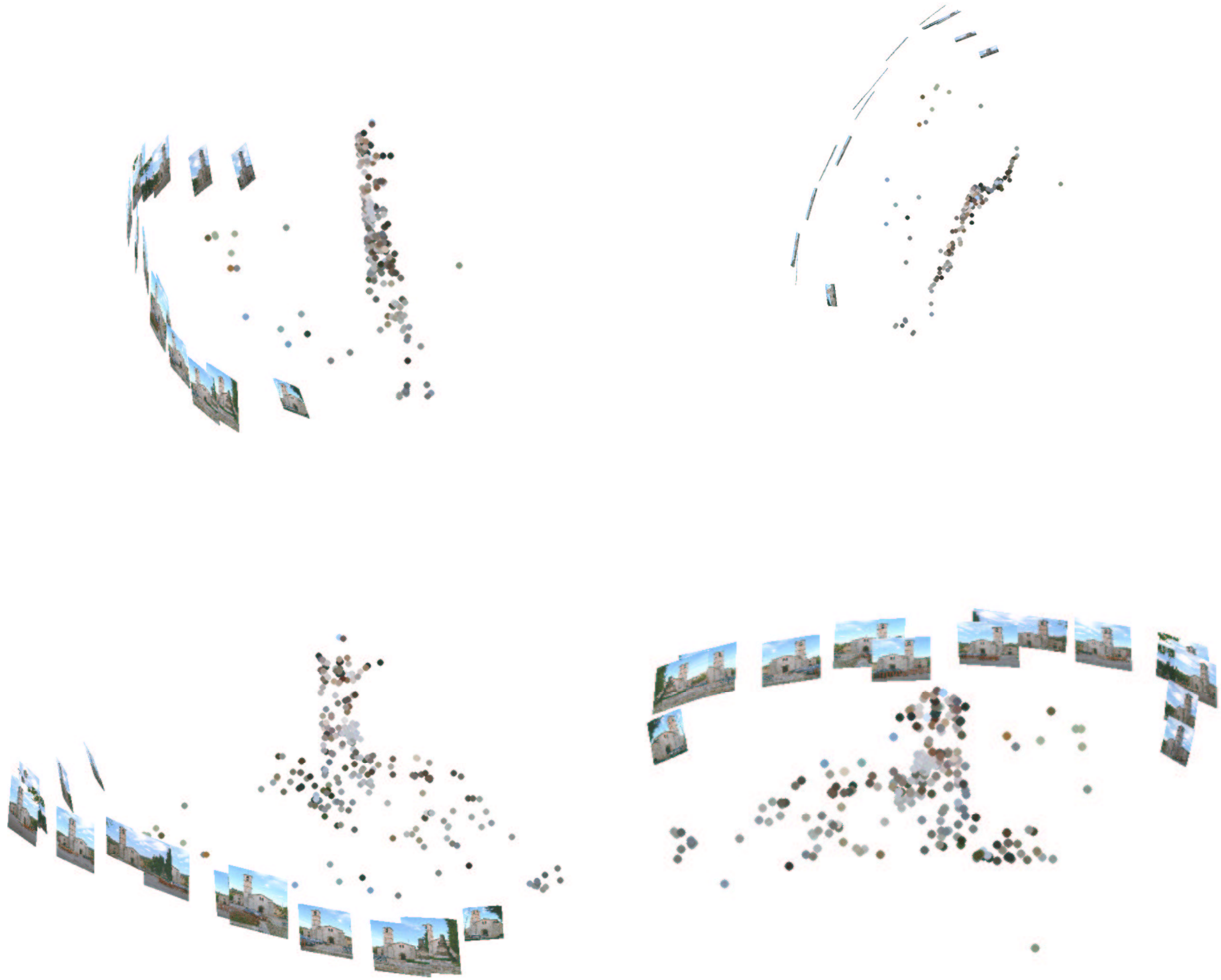
Experiments on WBS



Experiments on WBS

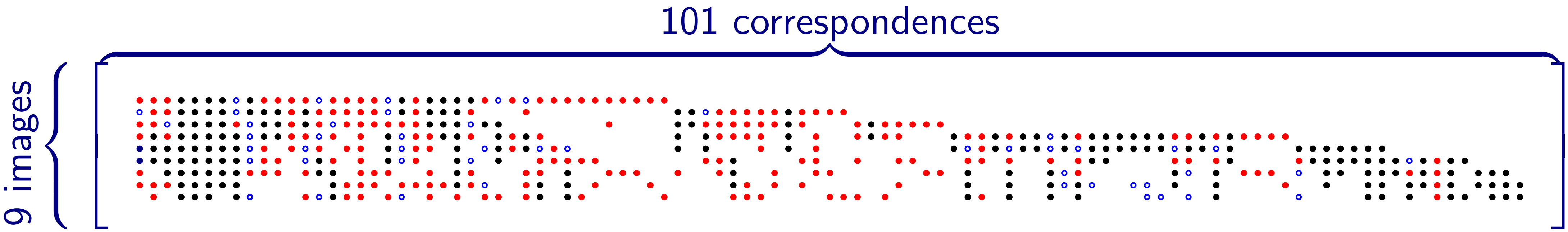
Scene *Valbonne all pairs (Oxford)*	14 images [768 × 512]
Outliers	**396** (28.14 % of 1407 image points)
Rec. / not-reconstructed cameras	14 / 0
Rec. / partially rec. / not-rec. corresp.	271 / 32 / 105 of 376
Mean / maximal reprojection error	**0.45** / 3.66 pxl (from inliers)





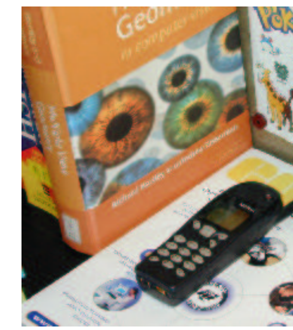
Some Other WBS Experitments. . .

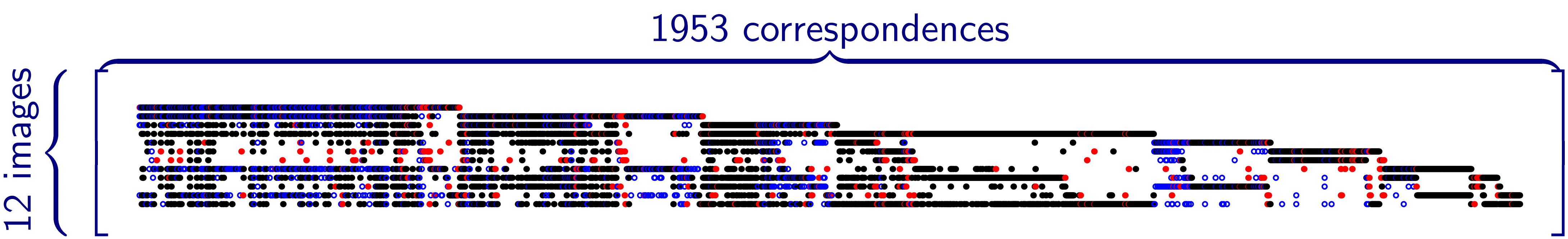
Scene *Movi-house (CMP)*	14 images [512×512]
Outliers	**207** (44.90 % of 461 image points)
Rec. / not-reconstructed cameras	9 / 0
Rec. / partially rec. / not-rec. corresp.	67 / 33 / 34 of 101
Mean / maximal reprojection error	**0.75** / 5.27 pxl (from inliers)



Some Other WBS Experitments. . .

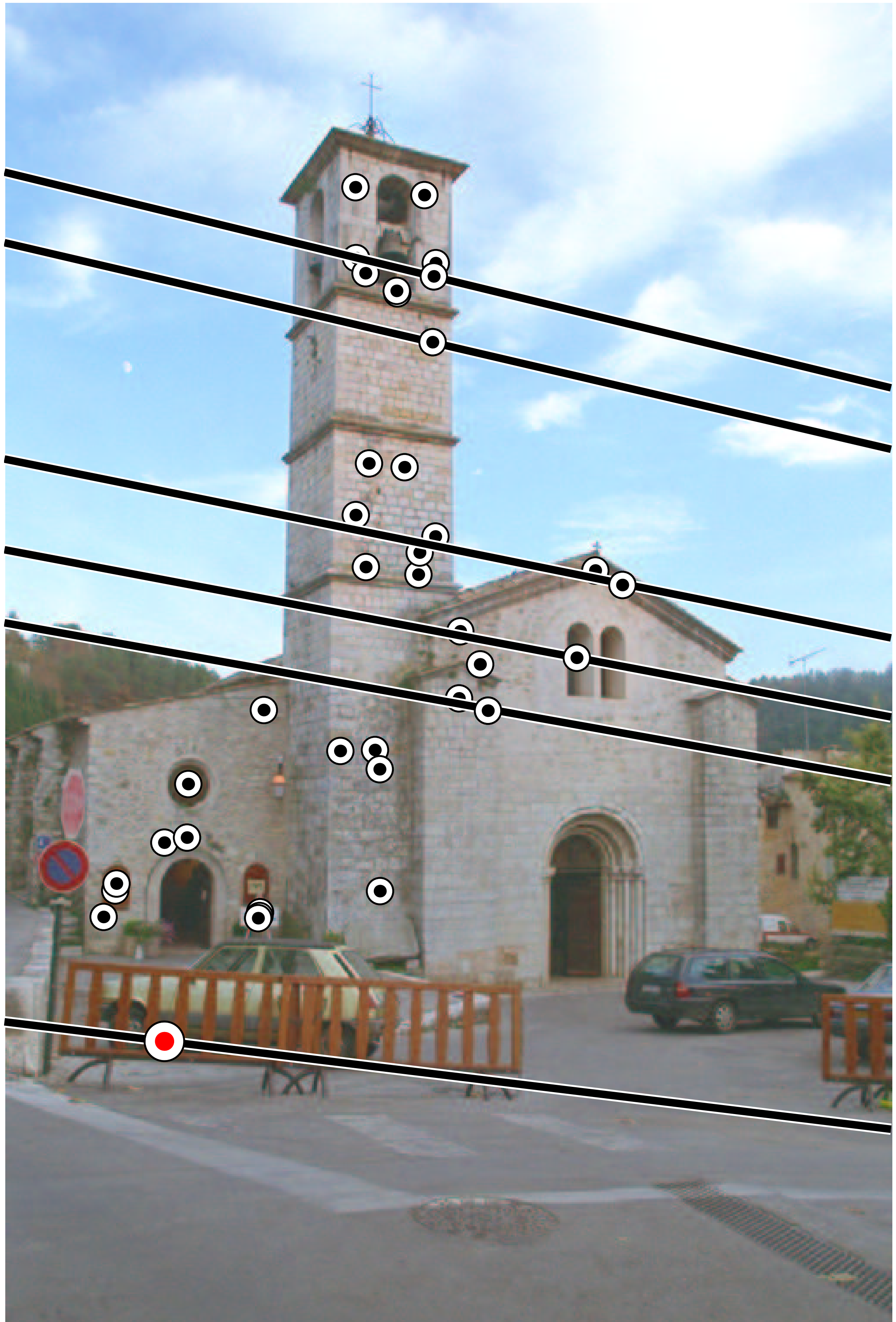
Scene *Shelf (CMP)*	12 images $[1200 \times 1600]$
Outliers	**414** (6.46 % of 6411 image points)
Rec. / not-reconstructed cameras	12 / 0
Rec. / partially rec. / not-rec. corresp.	1839 / 72 / 114 of 1953
Mean / maximal reprojection error	**0.51** / 4.90 pxl (from inliers)

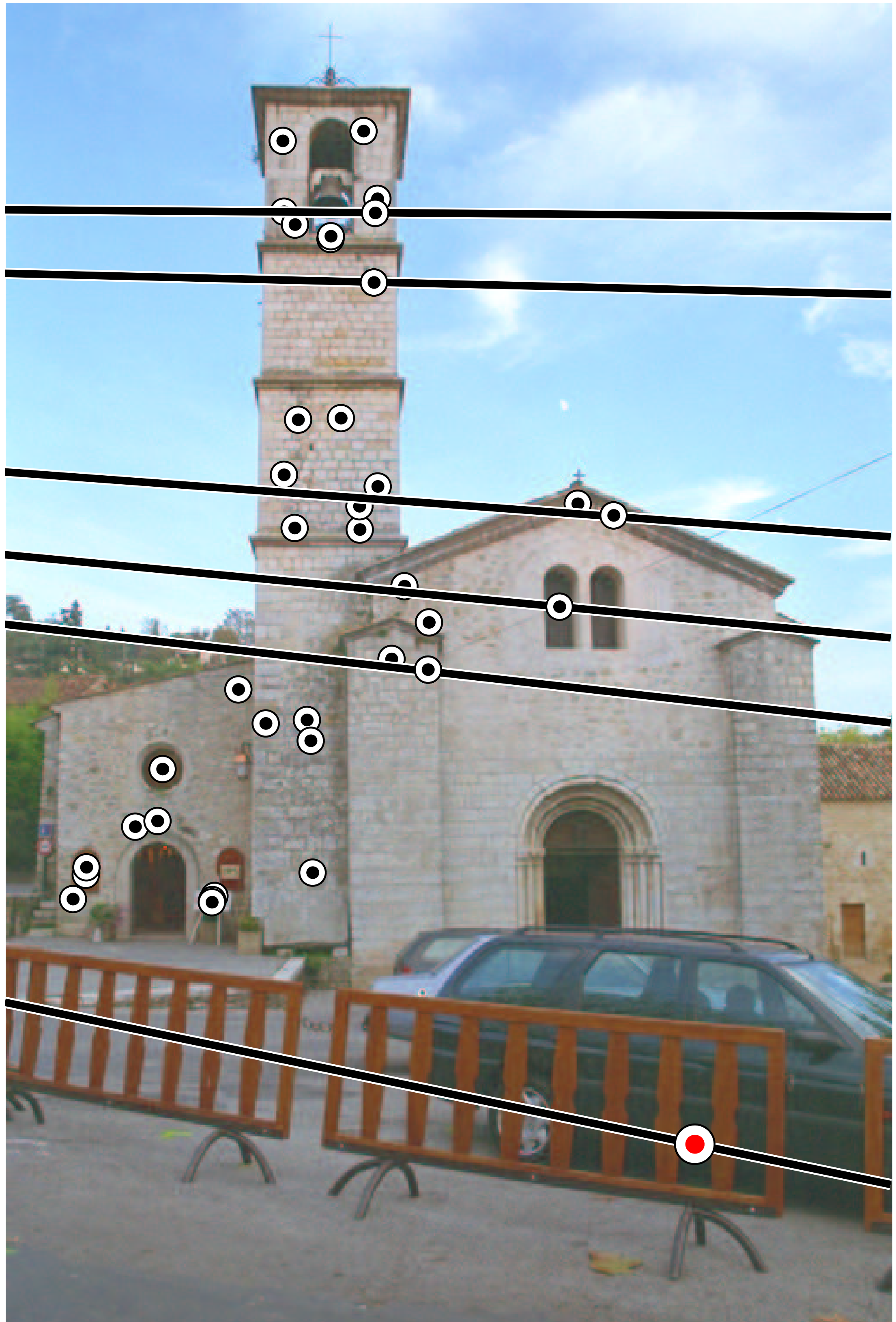


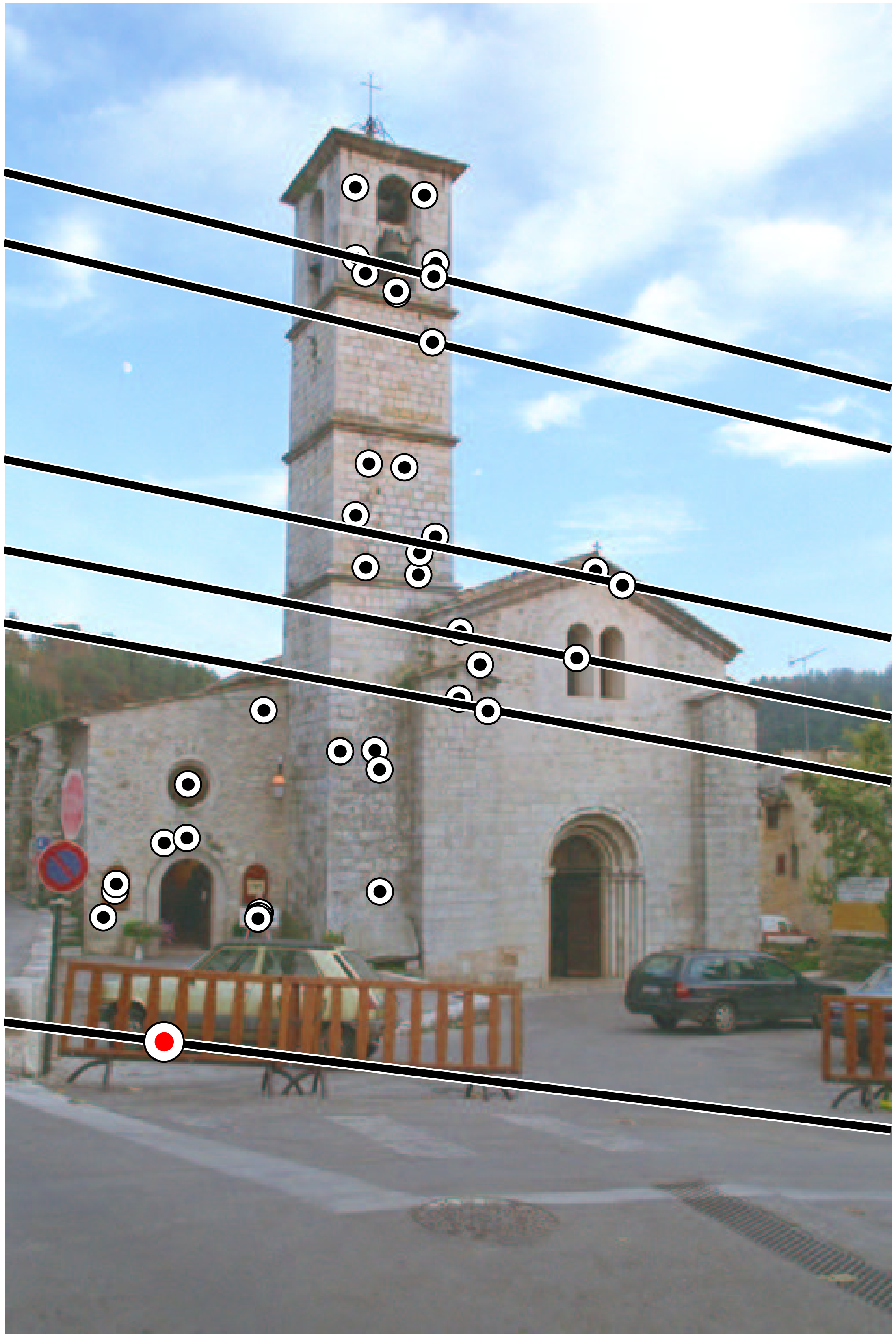


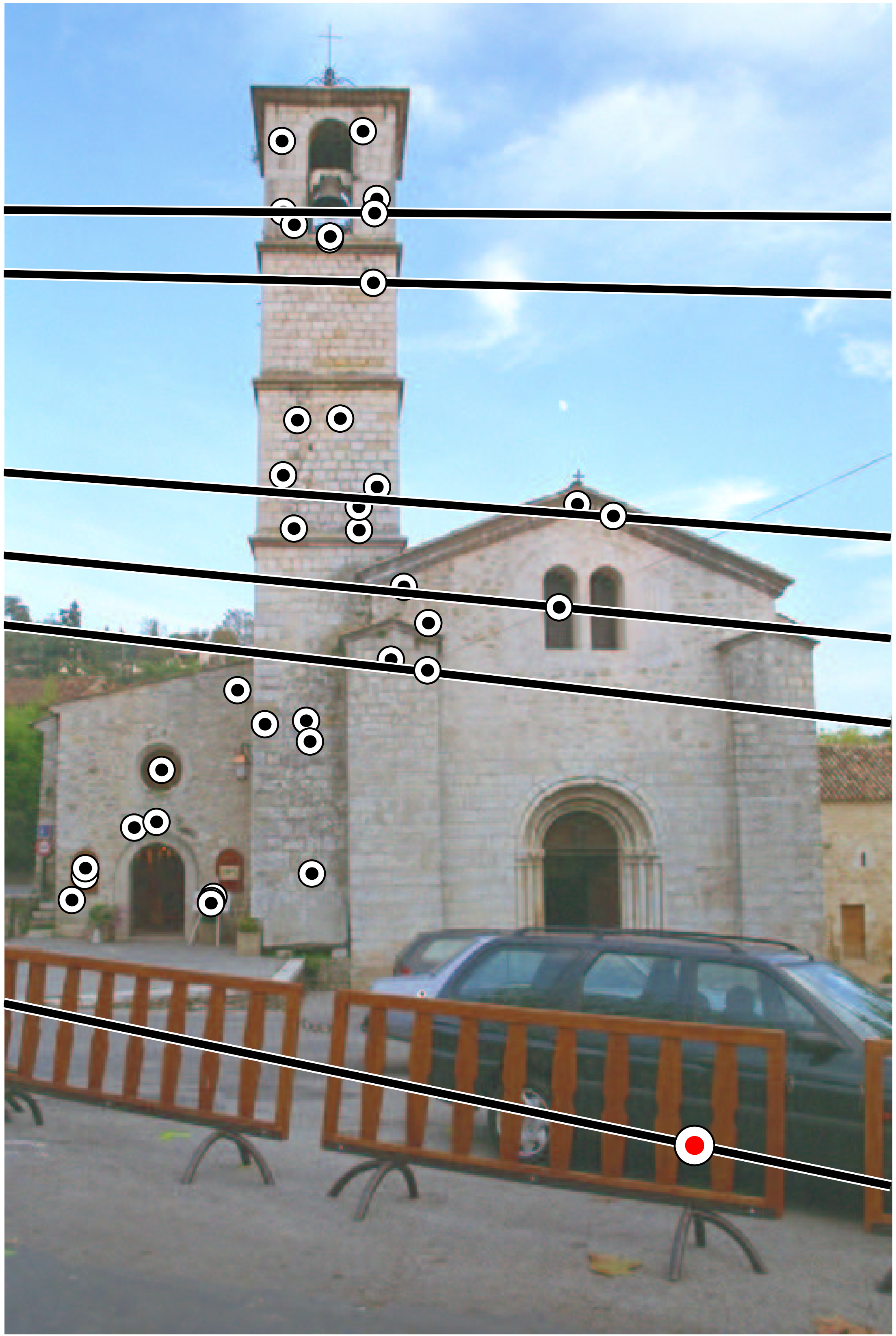
Conclusion

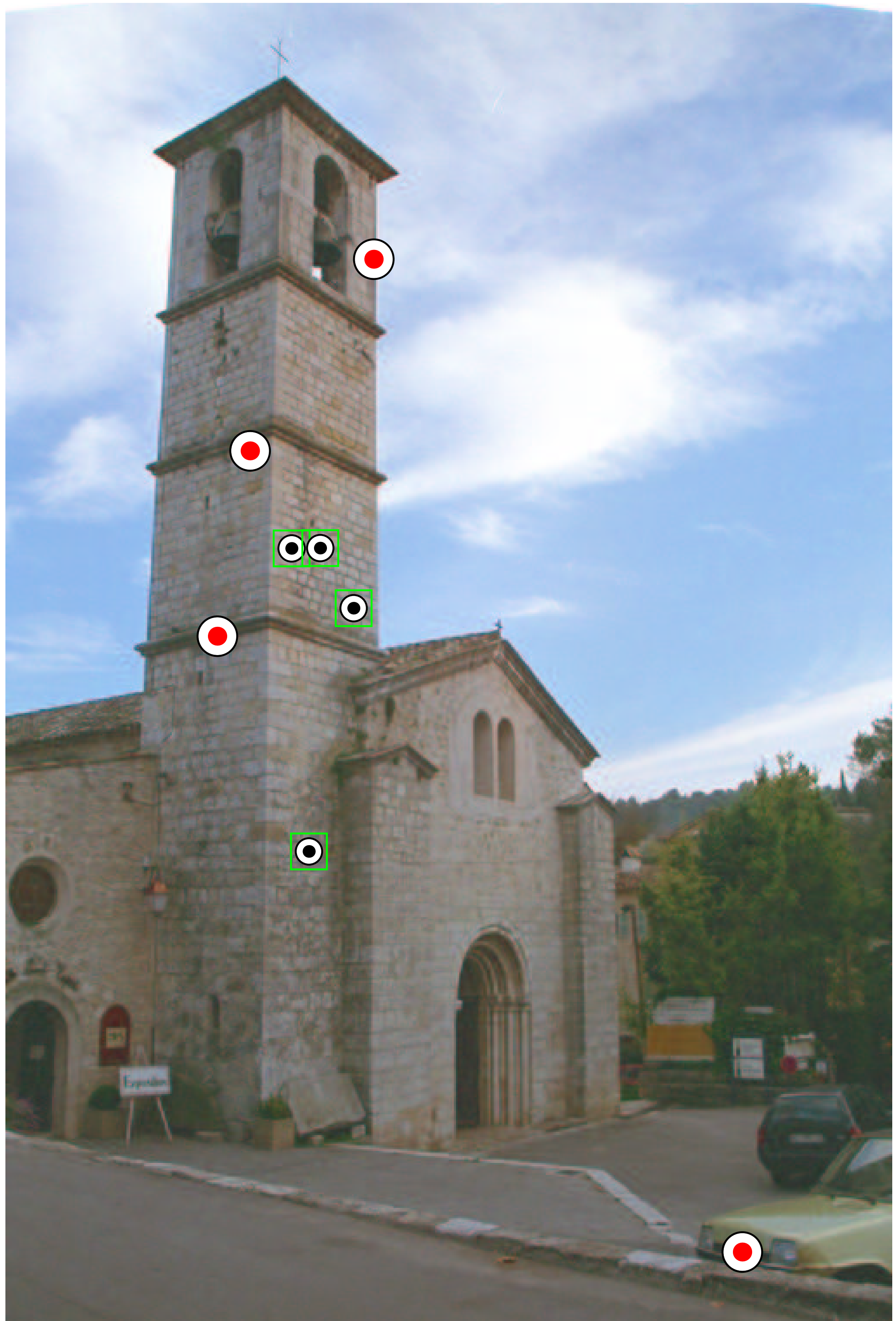
- A **new algorithm** for a general situation
  - **perspective** camera
  - **occlusions**
  - **outliers**
- Experiments
  - real scenes: applicable in wide base-line as well as on sequences

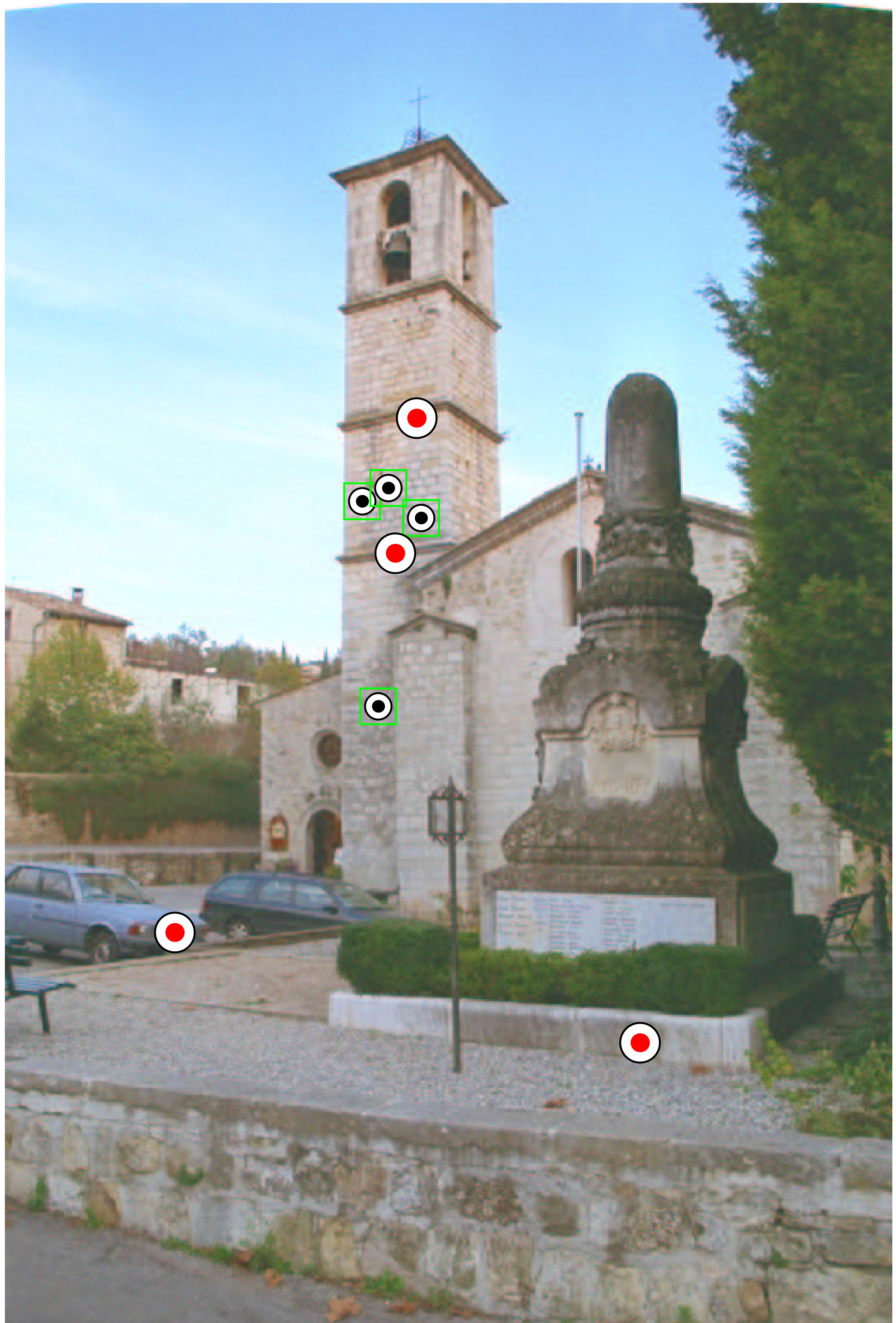


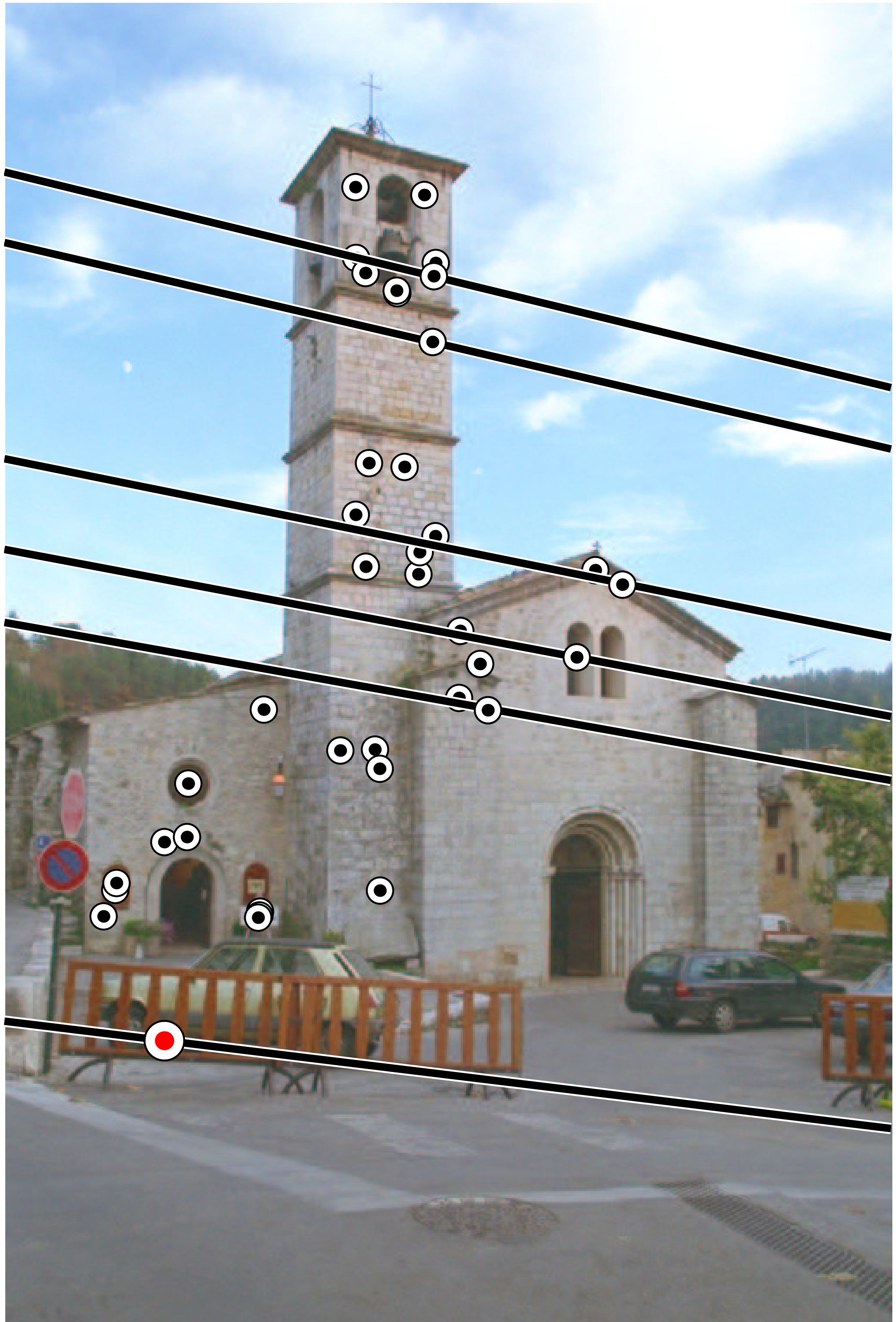


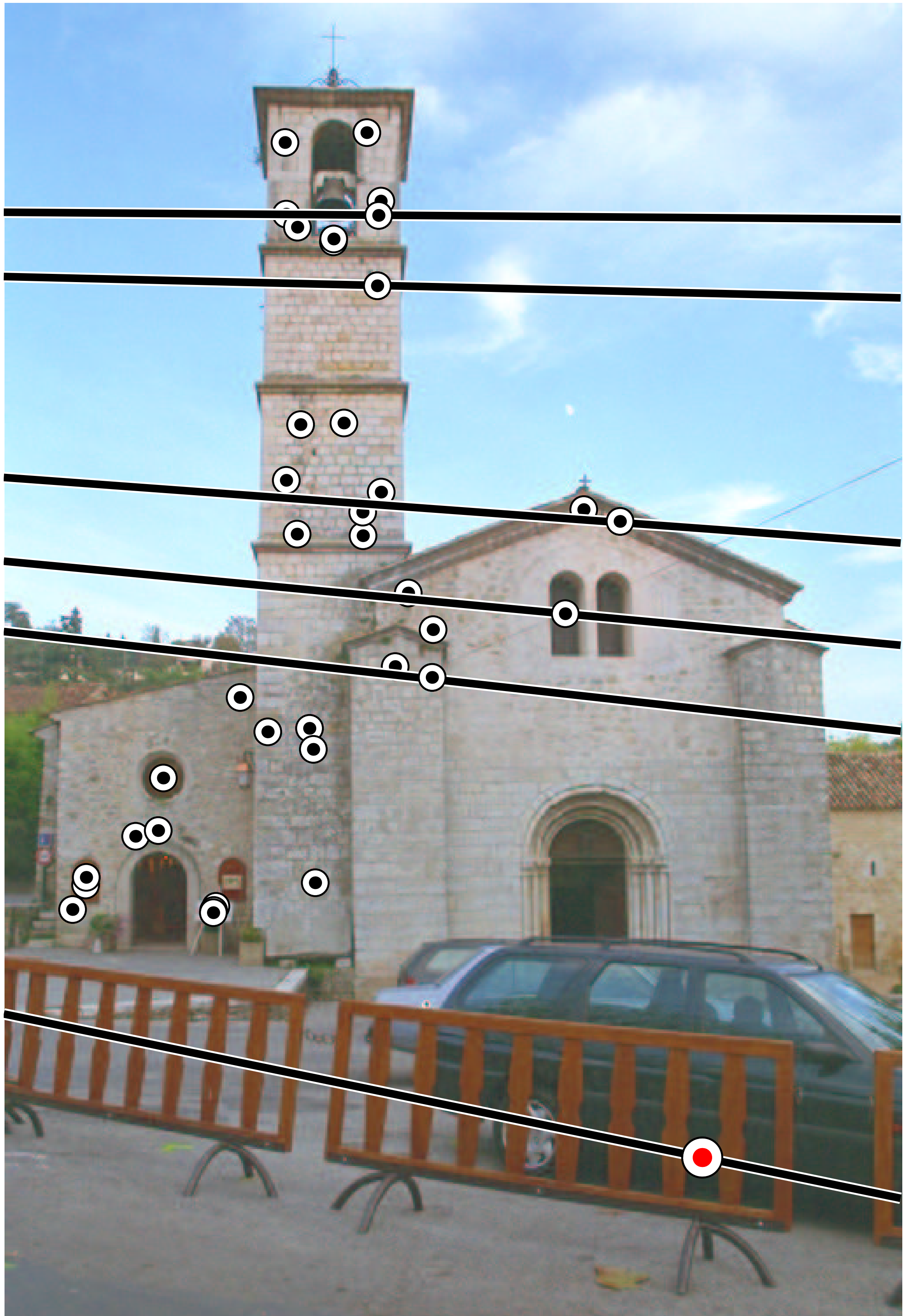


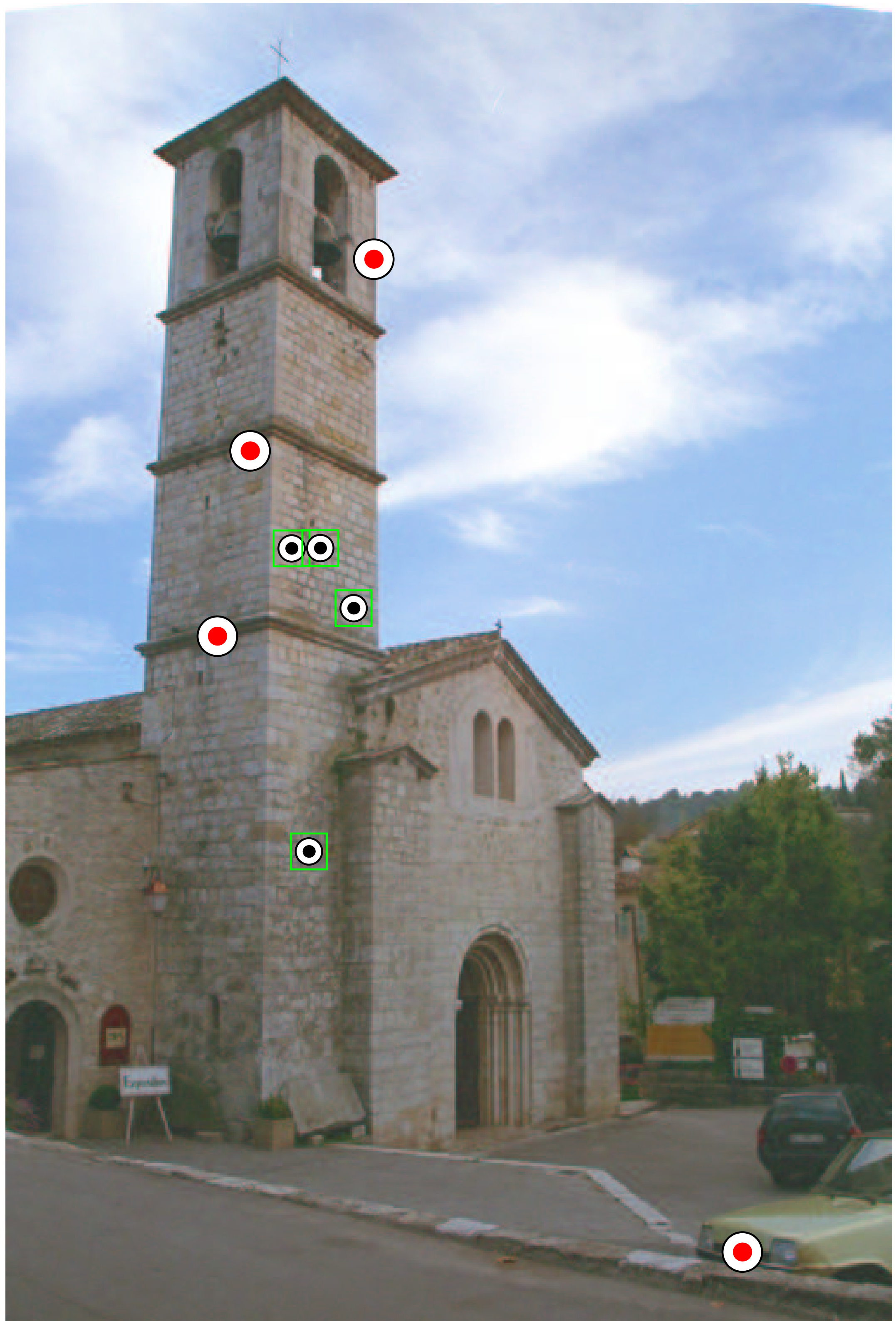


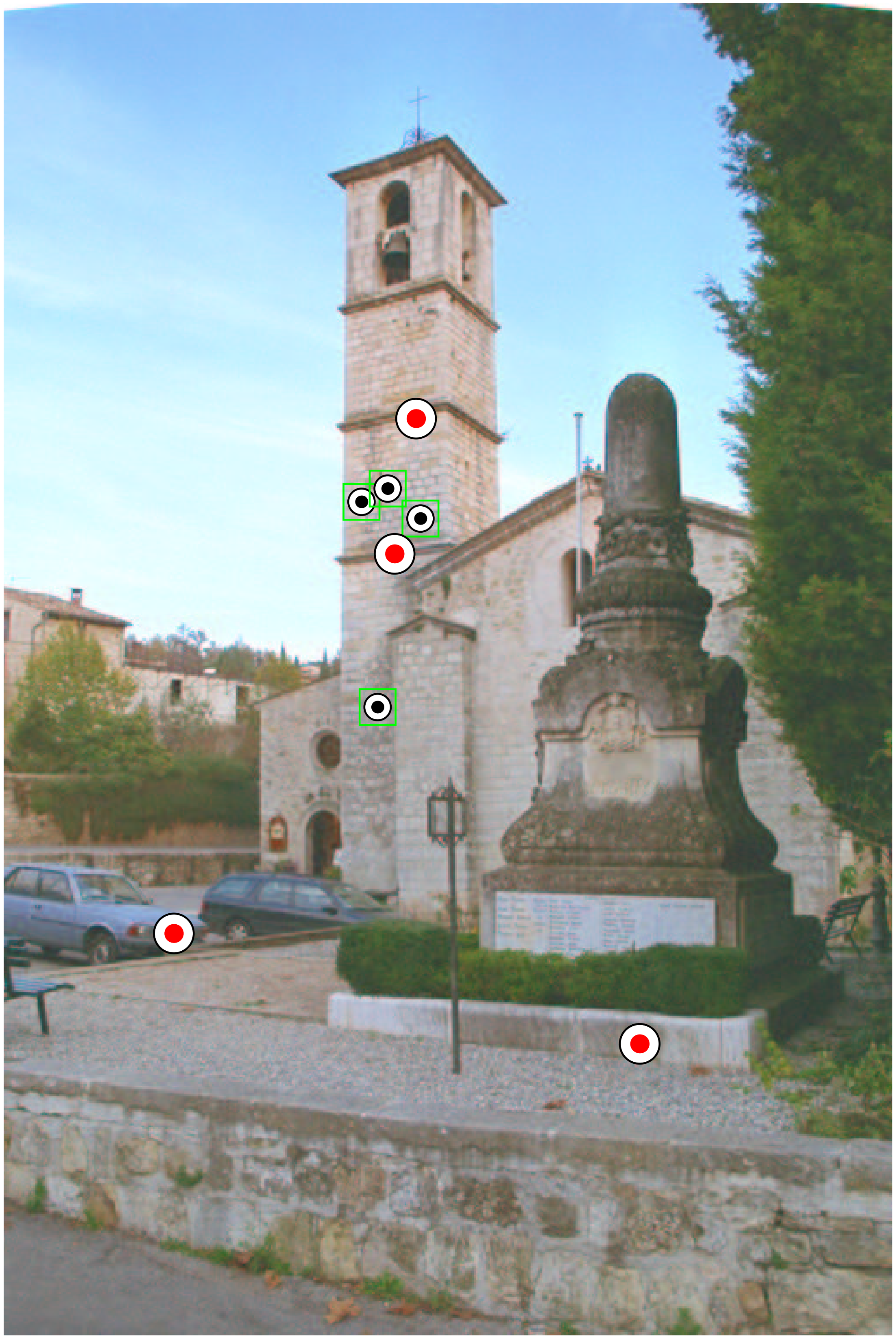


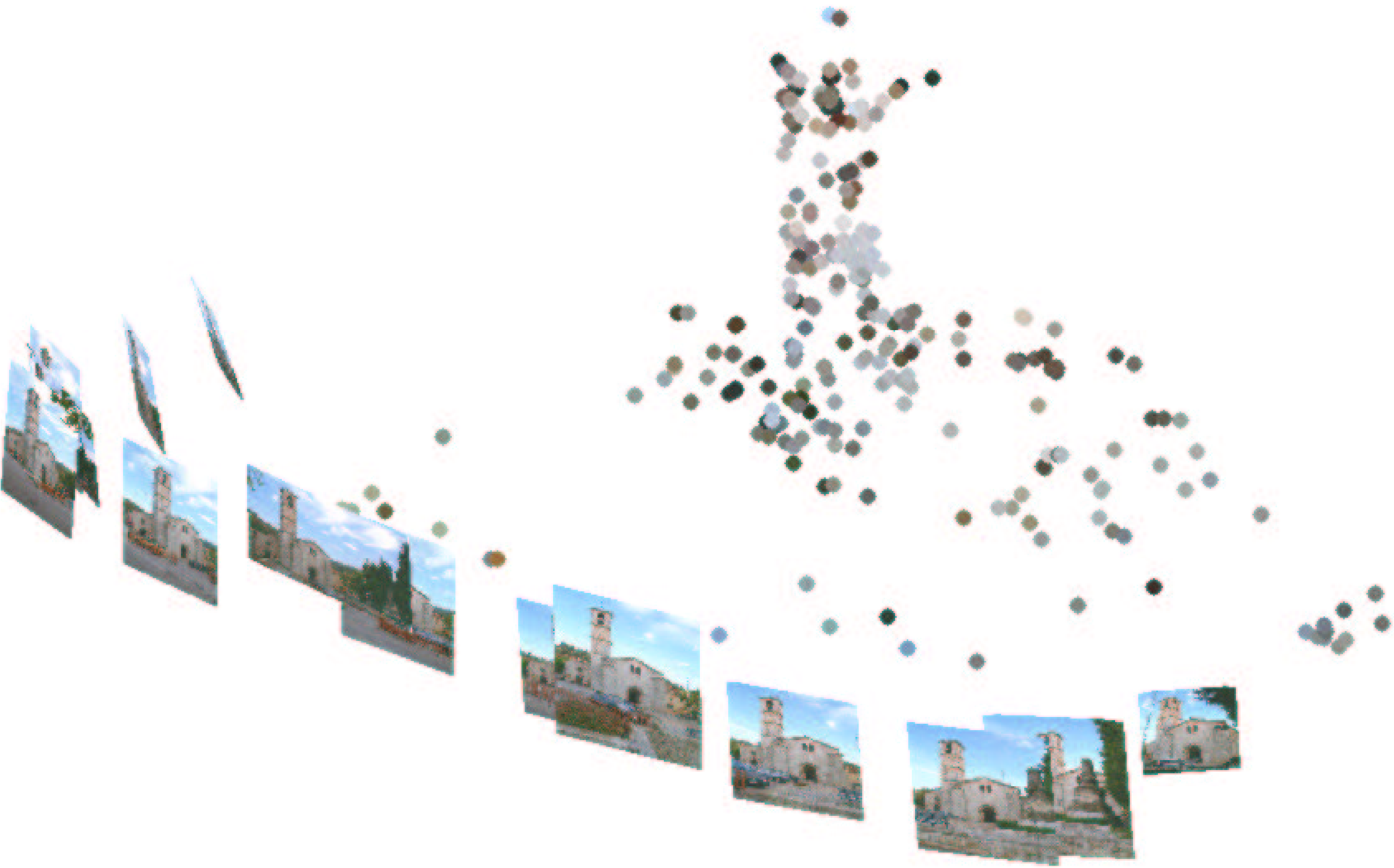


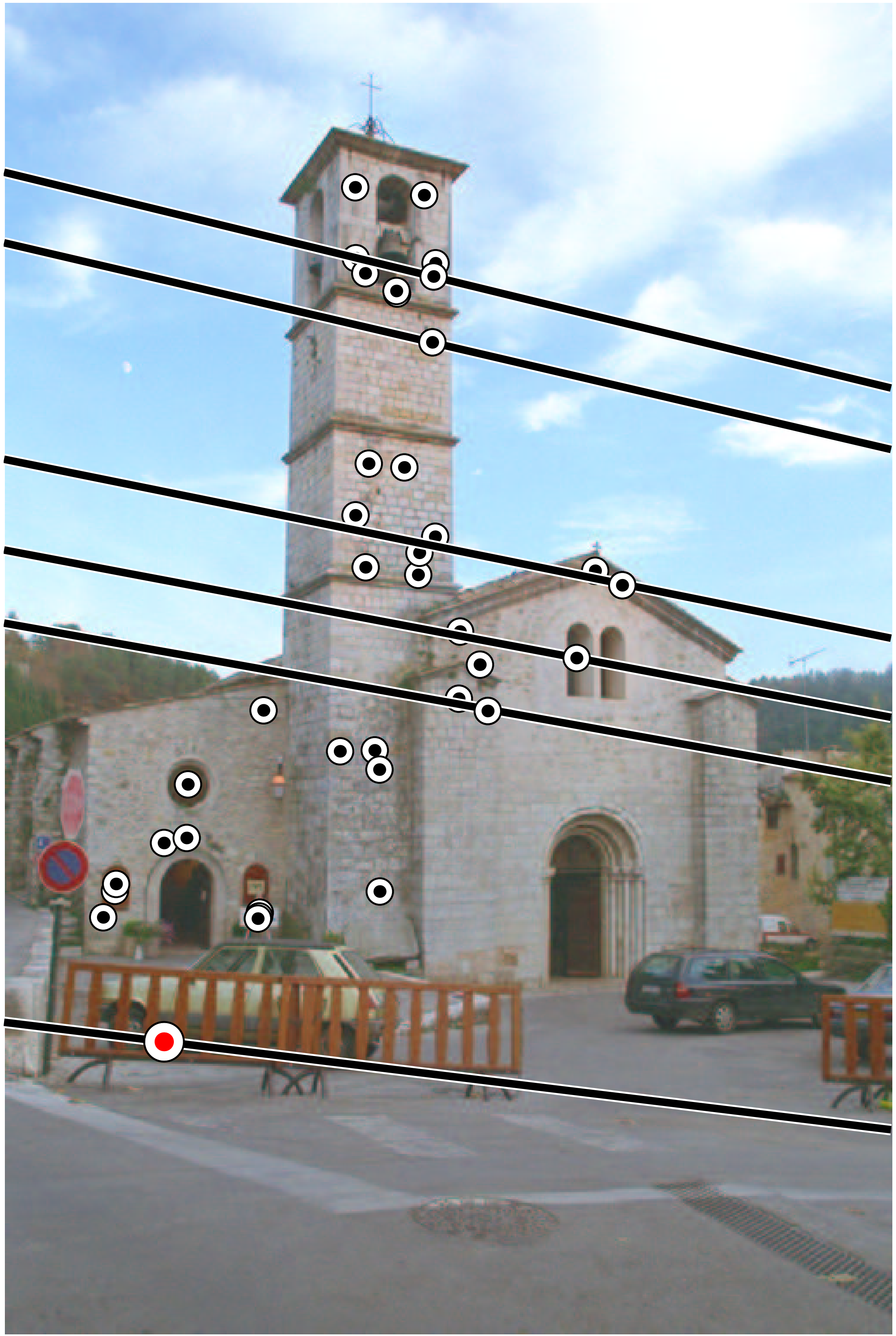


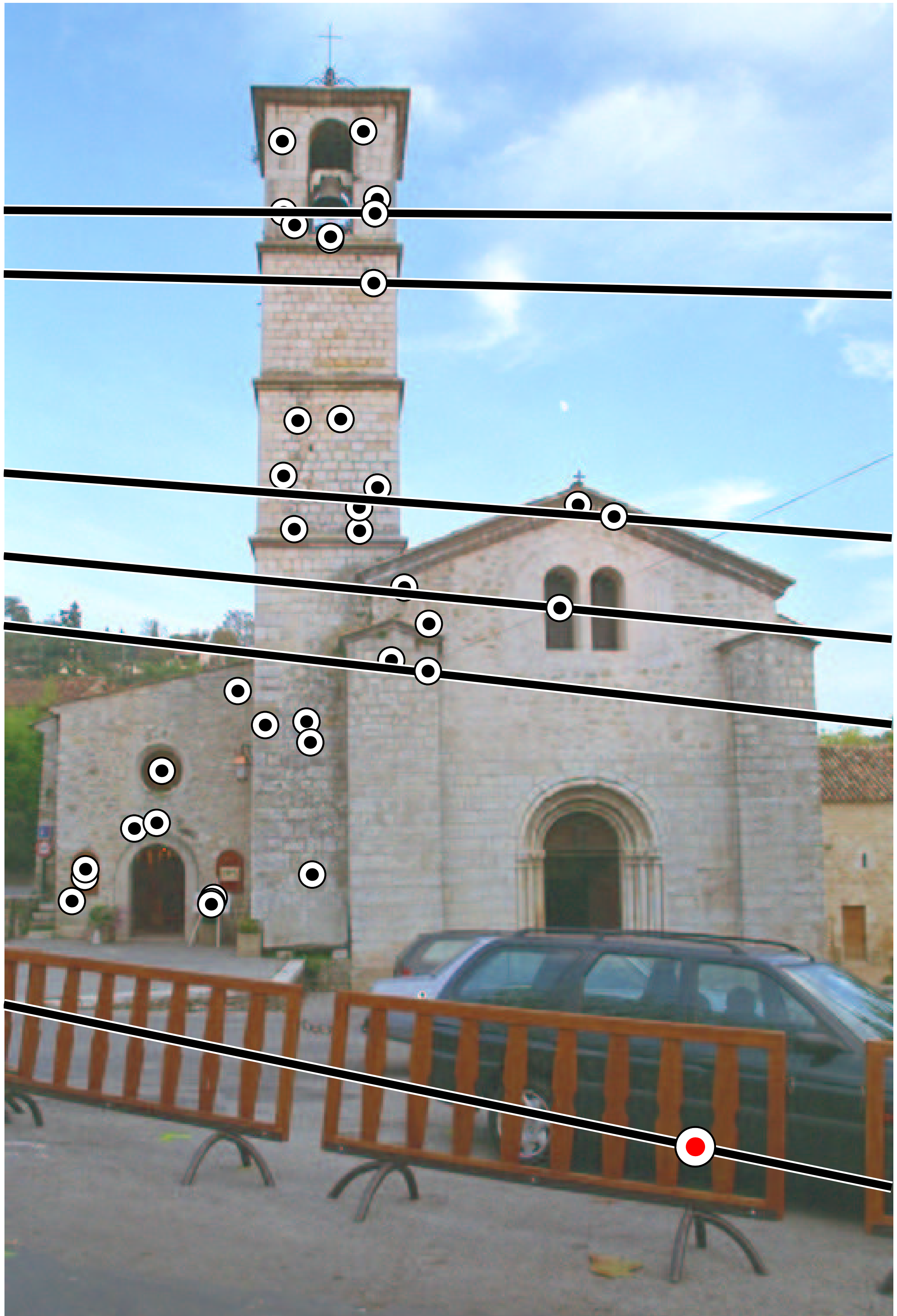


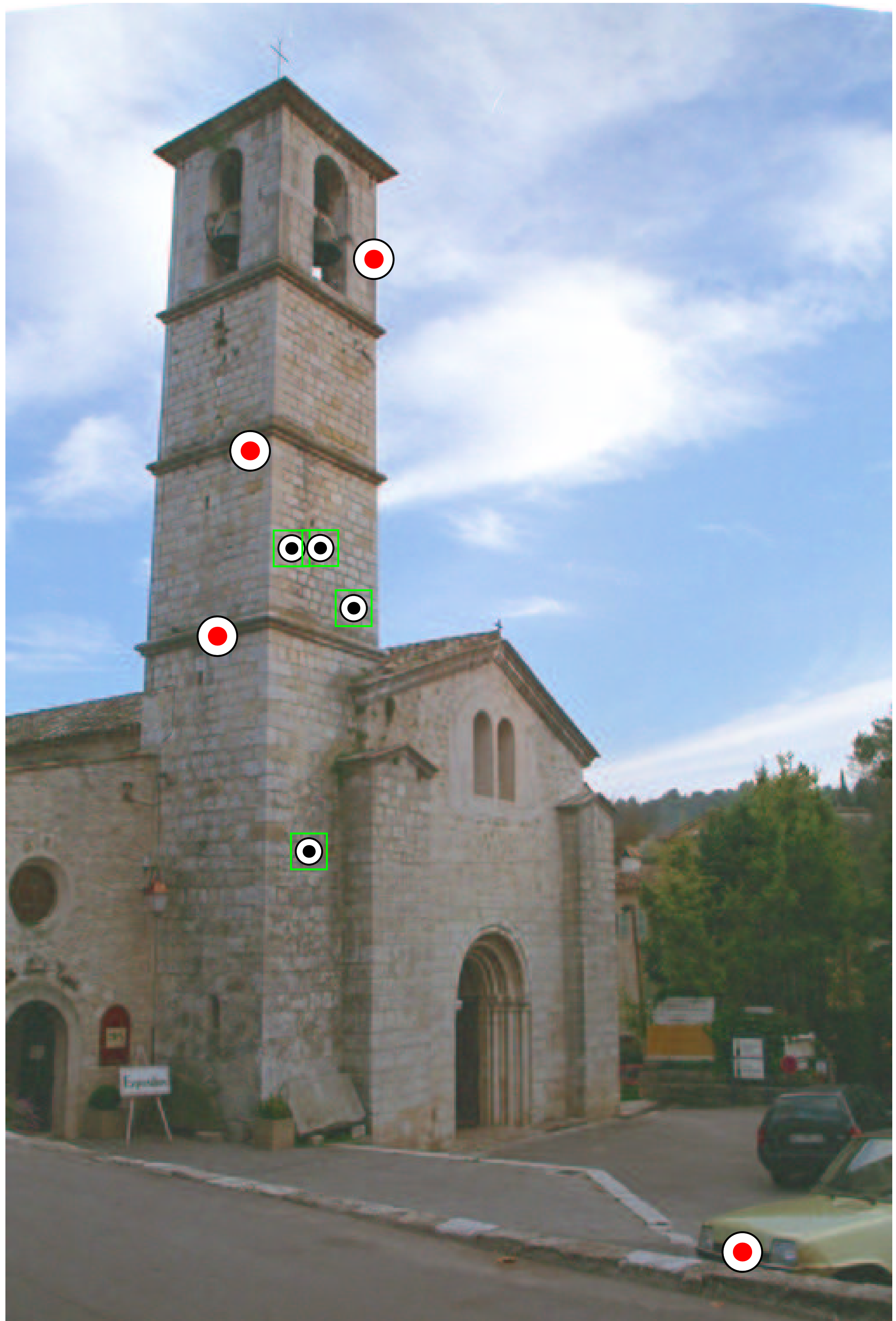


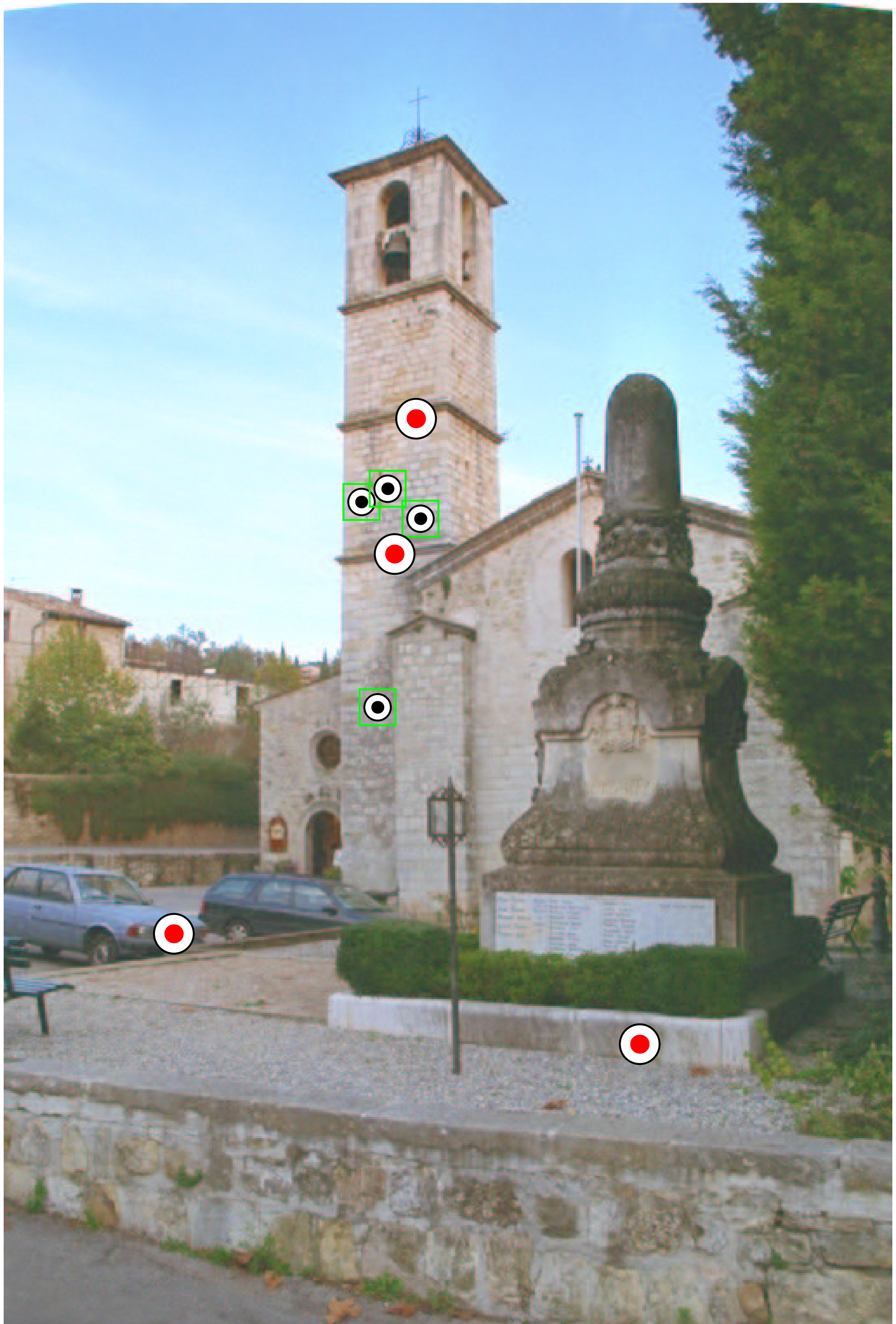


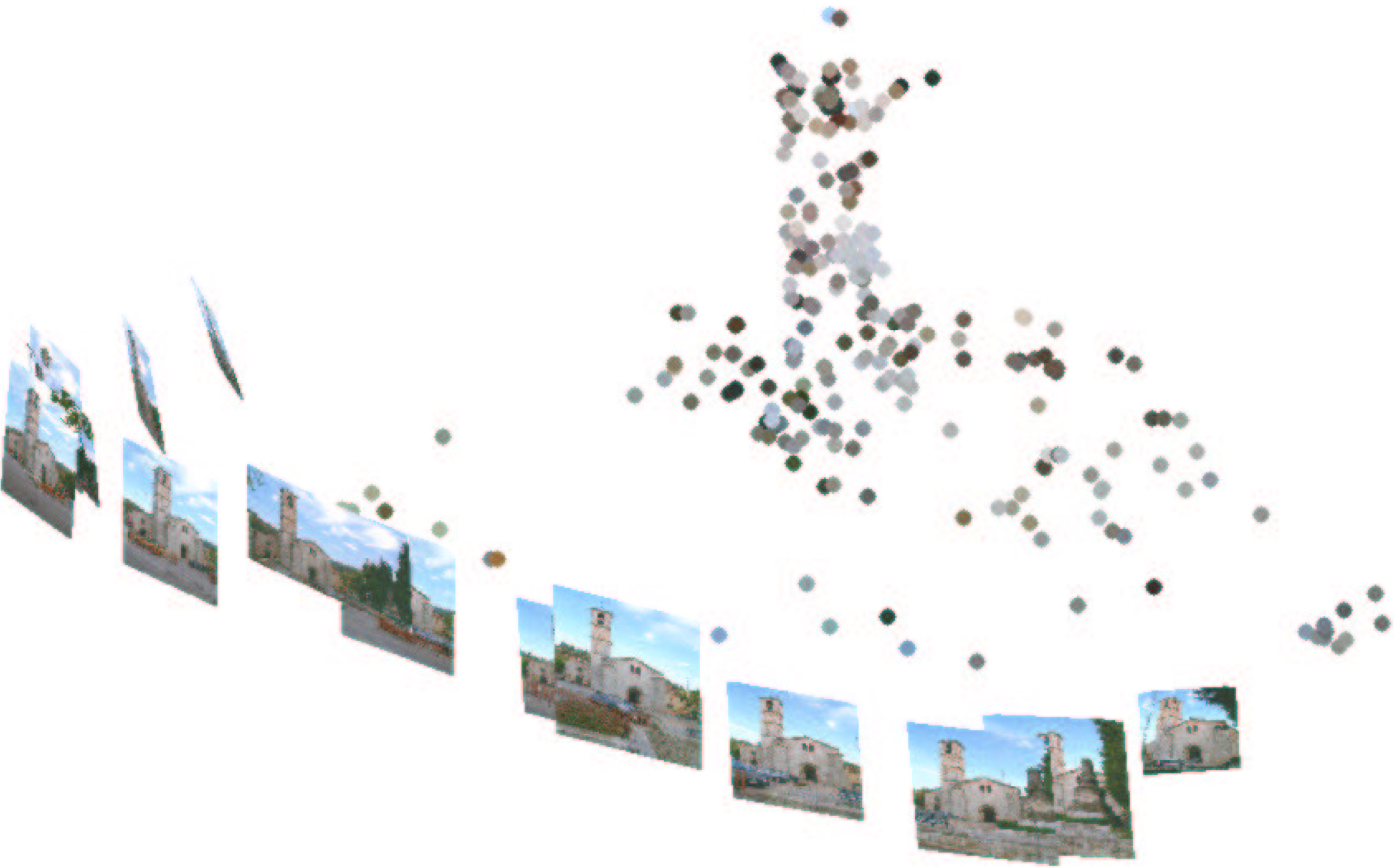


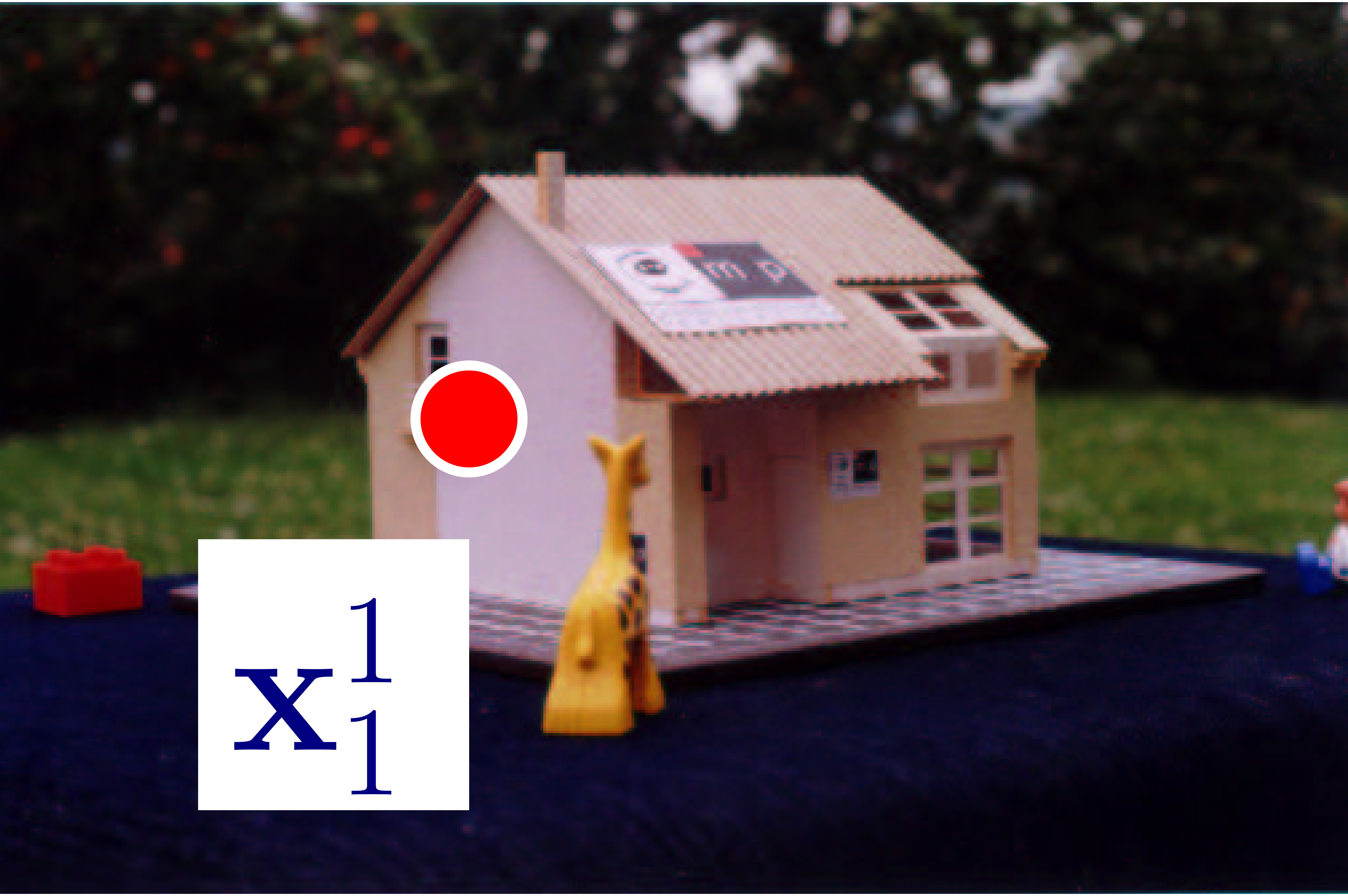


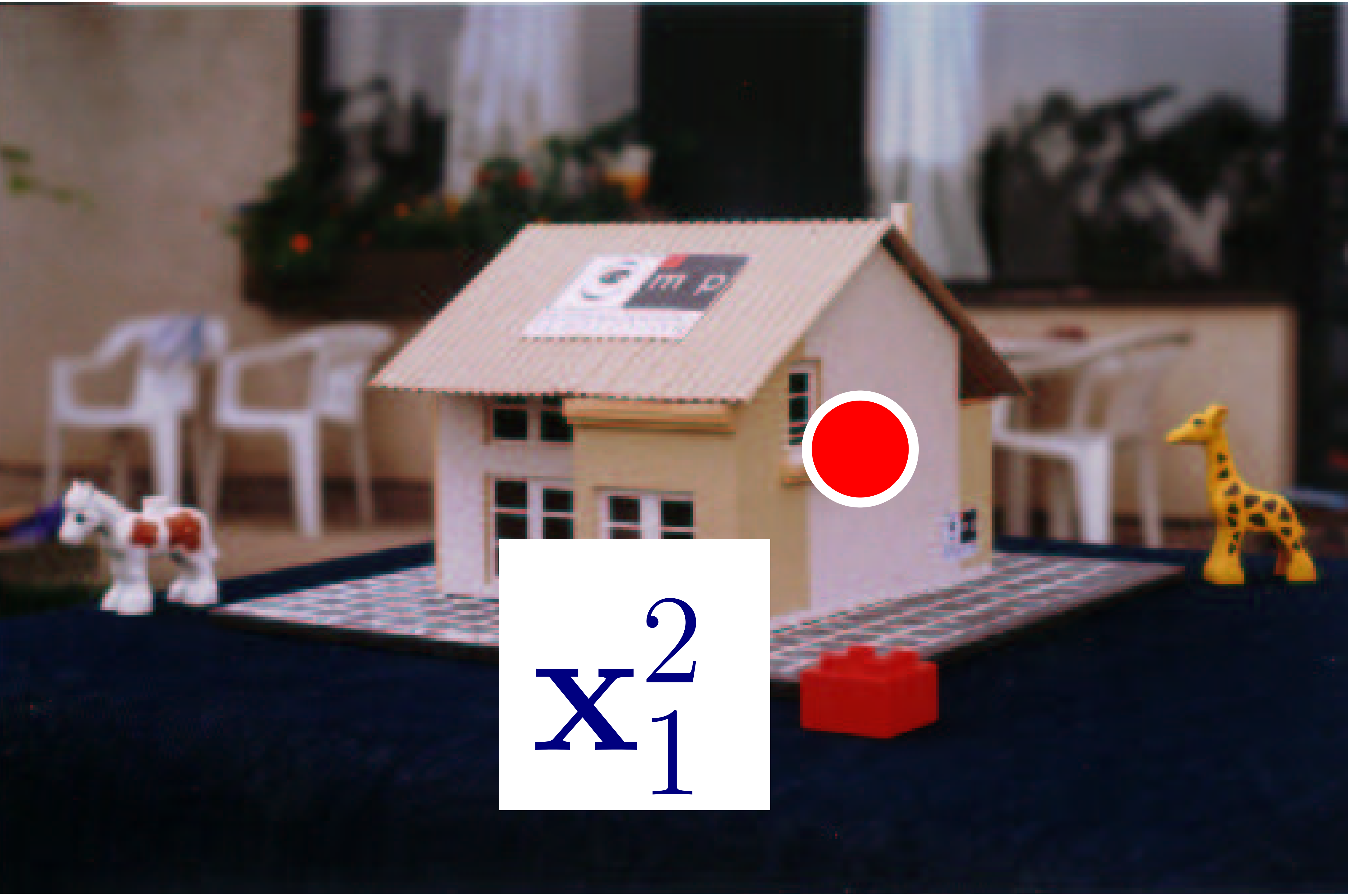


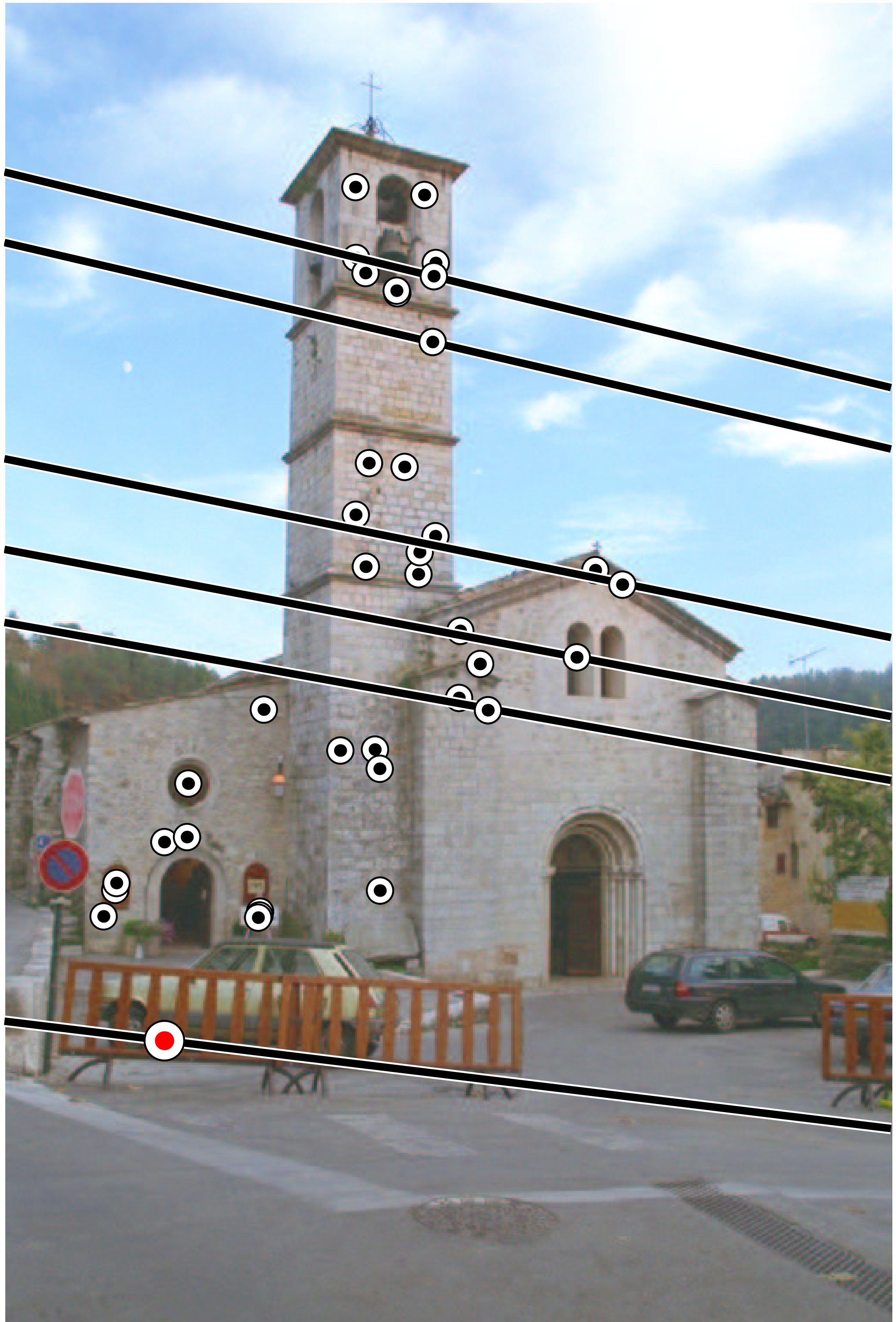


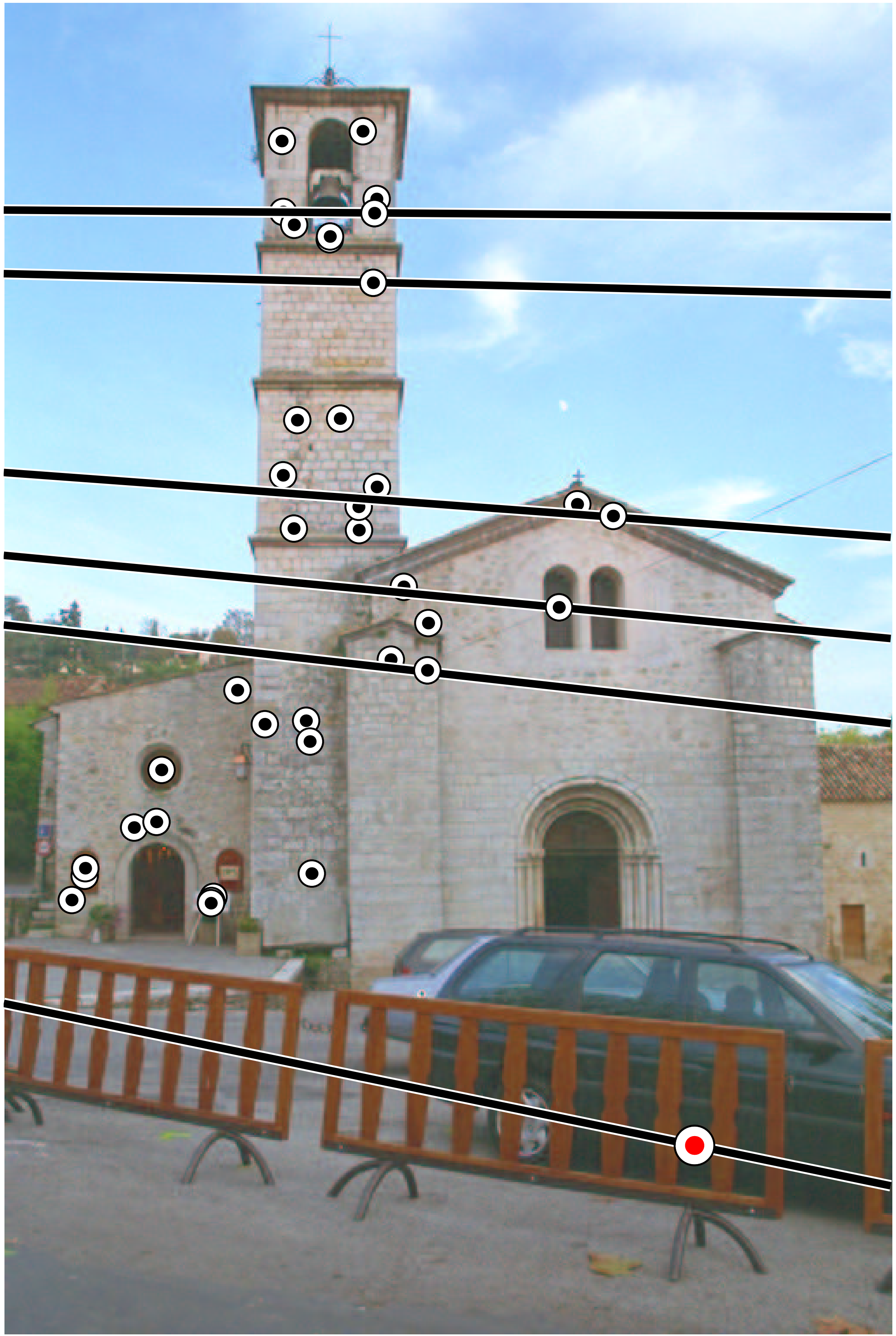


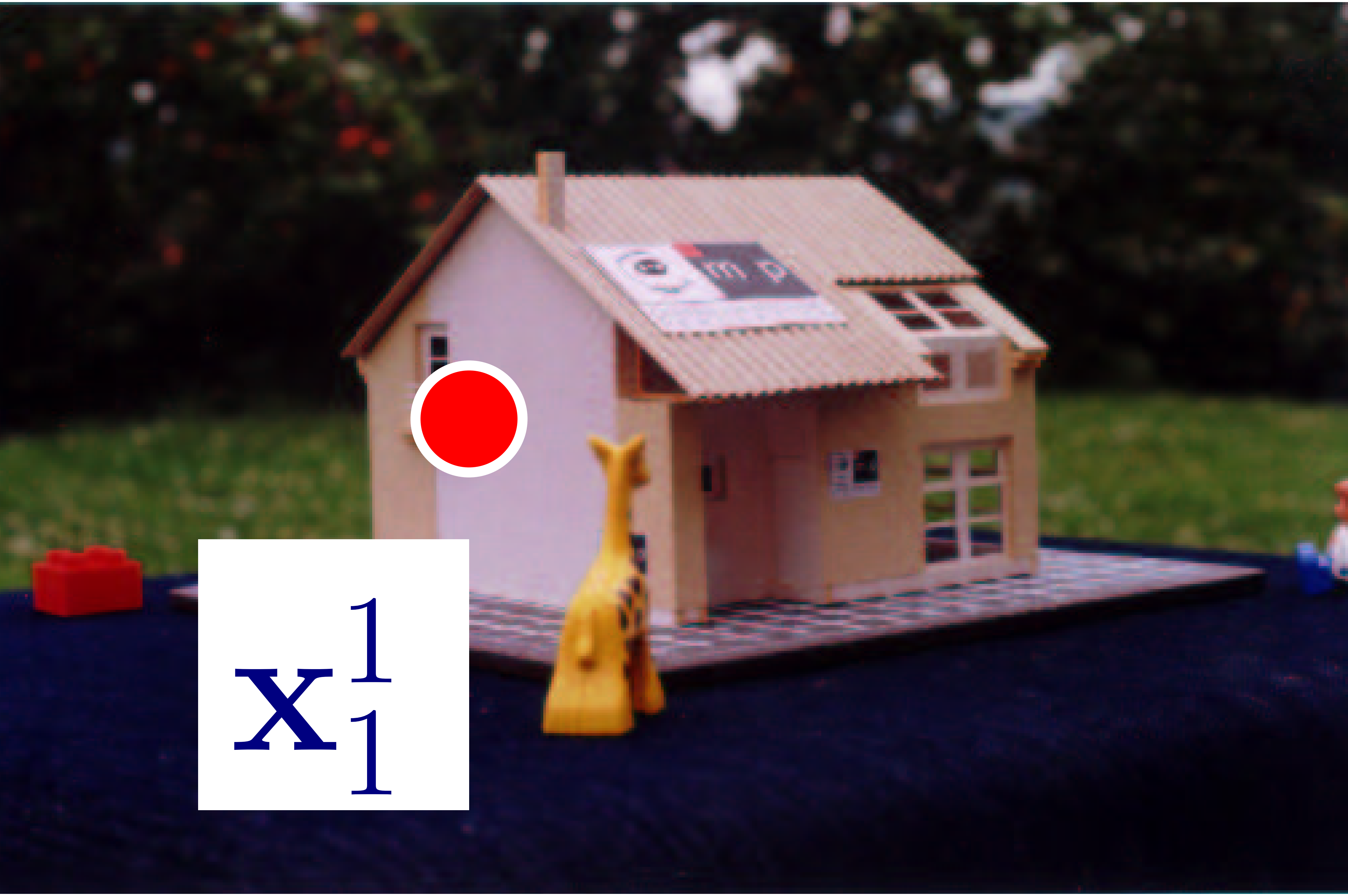


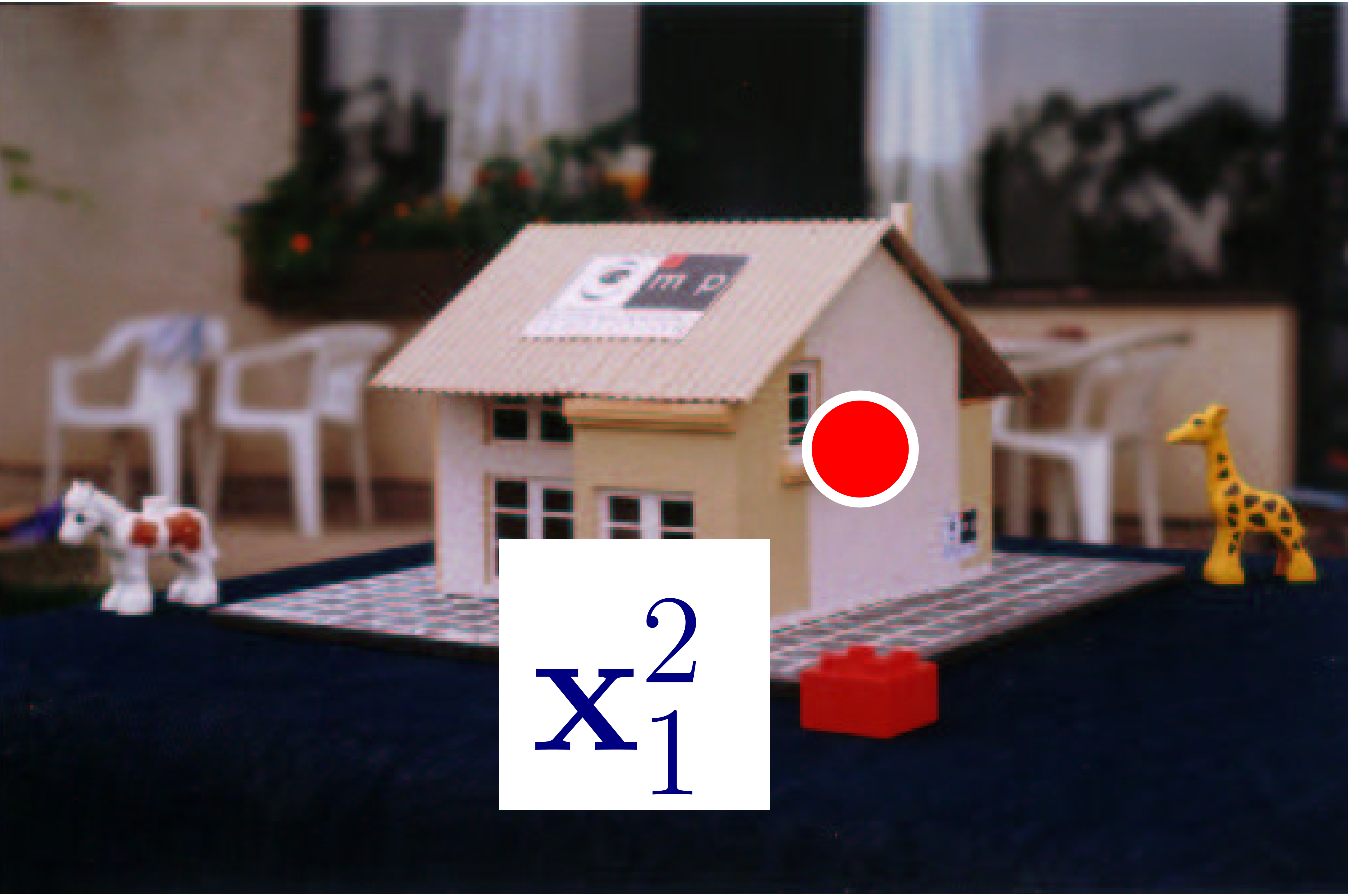

x_1^1

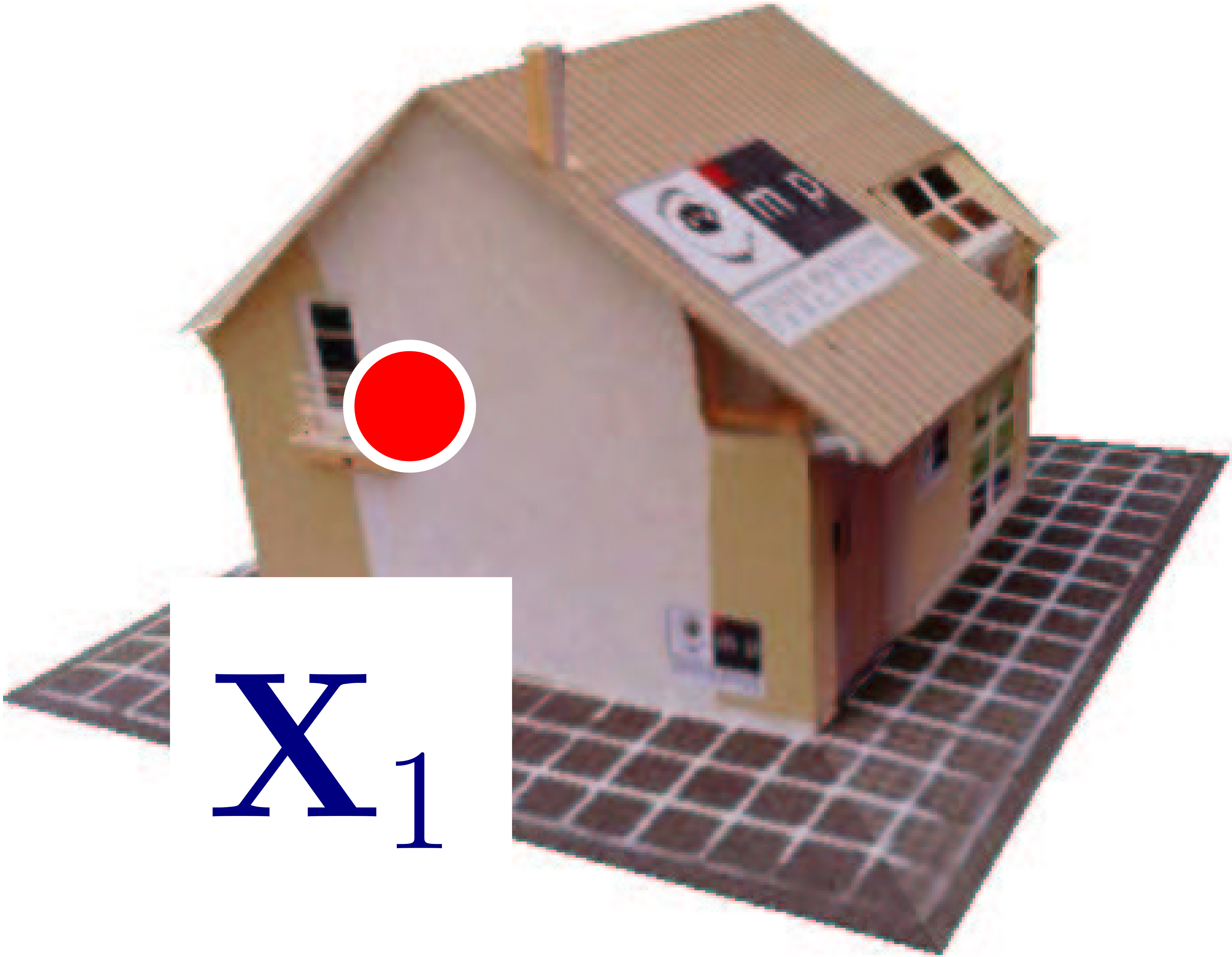

$\mathbf{x}_1^2$

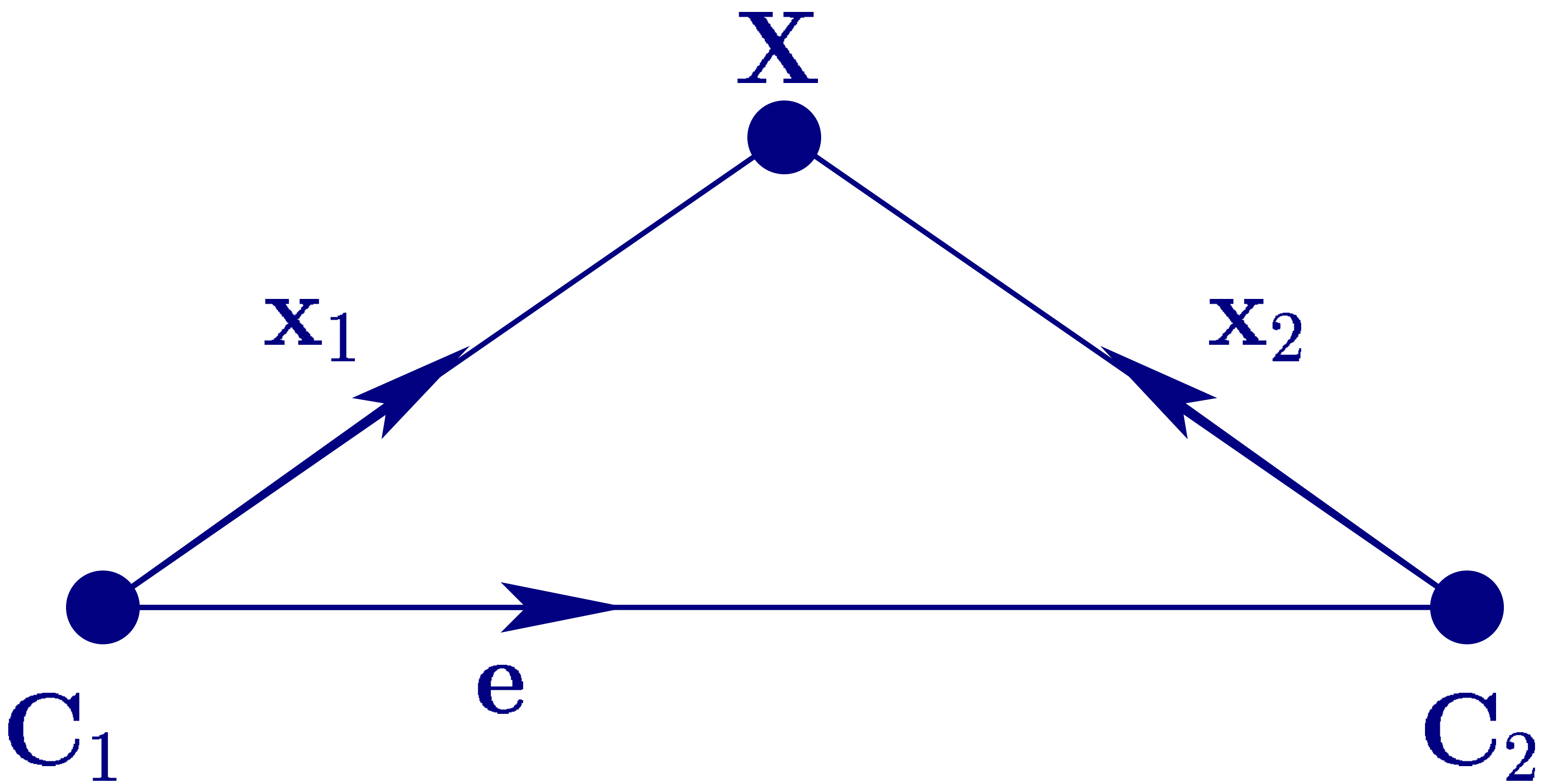


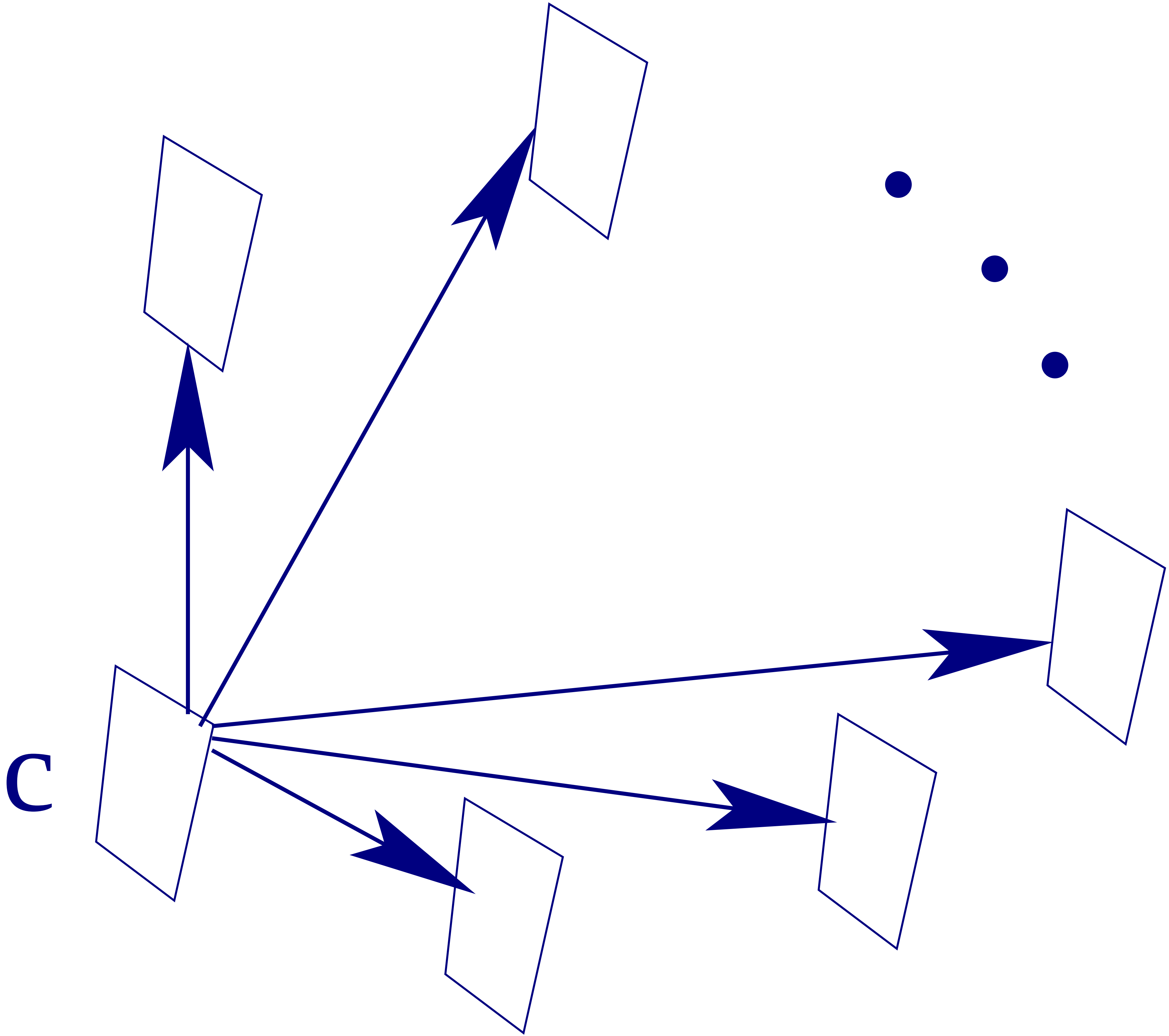


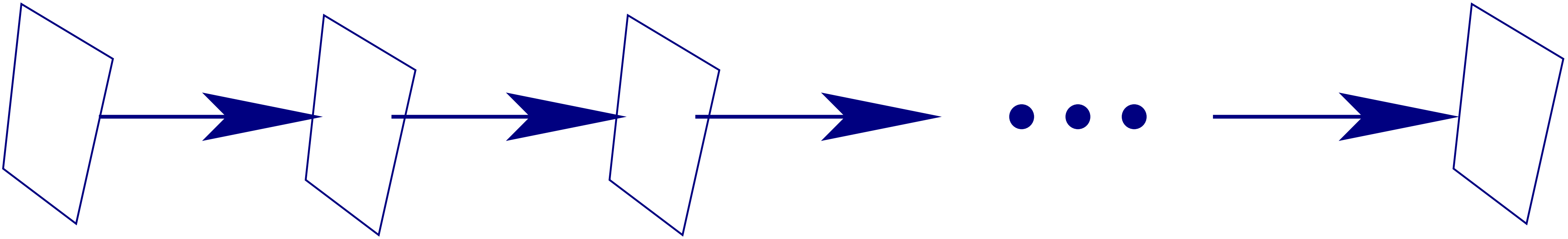

x_1^1


$\mathbf{x}_1^2$


X_1


X
$\mathbf{x}_1$
$\mathbf{x}_2$
$\mathbf{C}_1$
e
$\mathbf{C}_2$


C

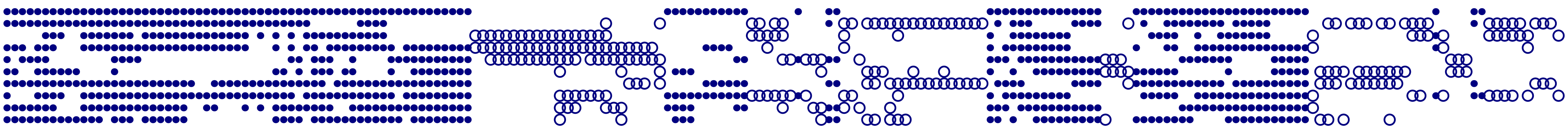


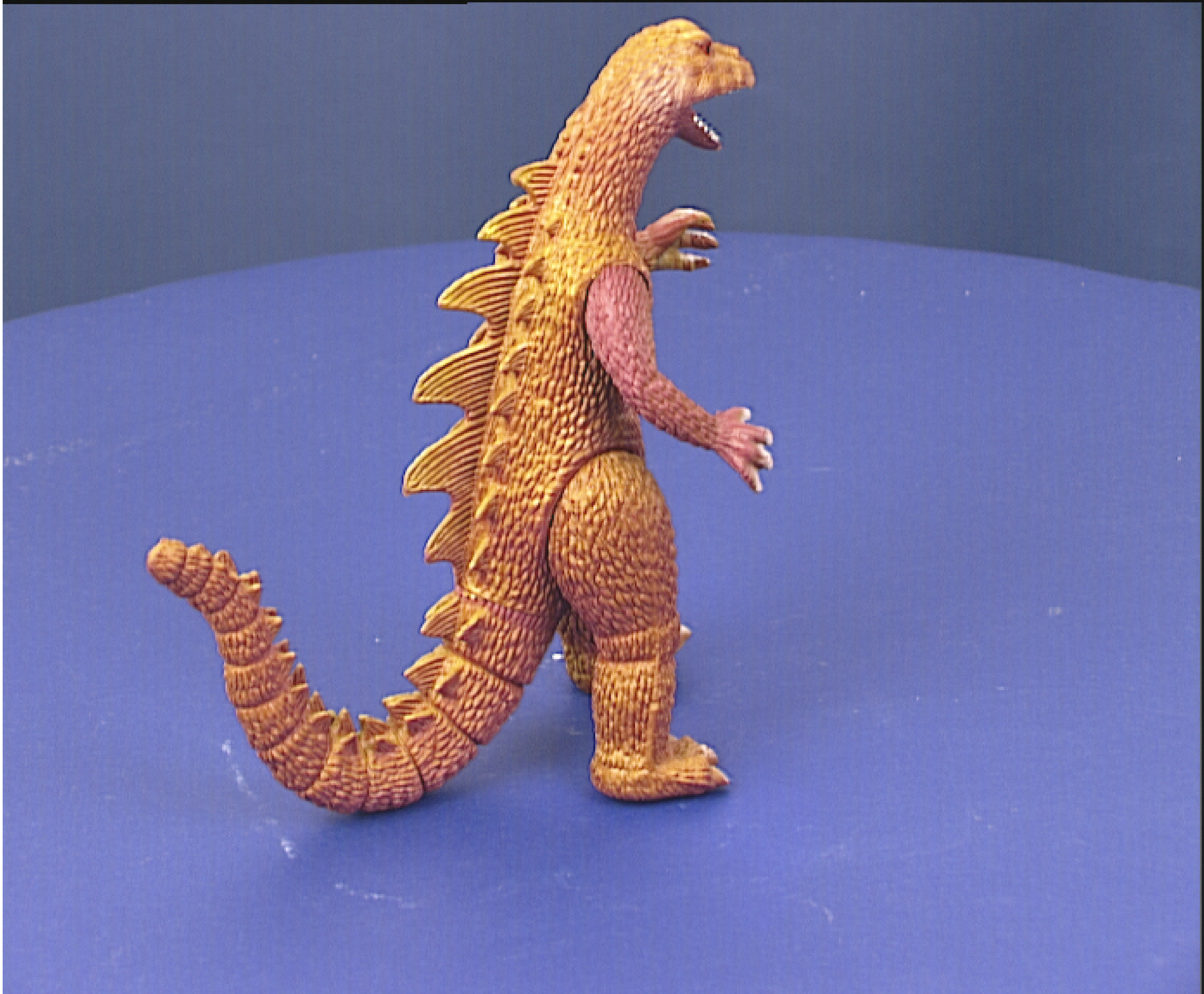
a 4-tuple of “LI” columns

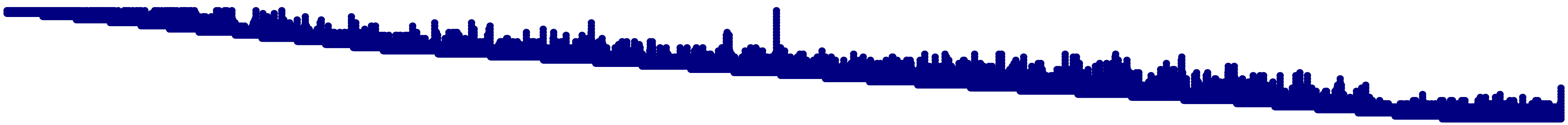
$$
\mathtt{B}_k = \begin{bmatrix} ?\ \mathbf{x}_1^1 & \lambda_2^1\mathbf{x}_2^1 & \lambda_3^1\mathbf{x}_3^1 & \lambda_4^1\mathbf{x}_4^1 \\ \lambda_1^2\mathbf{x}_1^2 & \lambda_2^2\mathbf{x}_2^2 & \lambda_3^2\mathbf{x}_3^2 & \lambda_4^2\mathbf{x}_4^2 \\ \vdots & & & \\ \lambda_1^m\mathbf{x}_1^m & \lambda_2^m\mathbf{x}_2^m & \lambda_3^m\mathbf{x}_3^m & \times \end{bmatrix} = \begin{bmatrix} ?\ x_1^1 & \lambda_2^1 x_2^1 & \lambda_3^1 x_3^1 & \lambda_4^1 x_4^1 \\ ?\ y_1^1 & \lambda_2^1 y_2^1 & \lambda_3^1 y_3^1 & \lambda_4^1 y_4^1 \\ ?\ w_1^1 & \lambda_2^1 w_2^1 & \lambda_3^1 w_3^1 & \lambda_4^1 w_4^1 \\ \lambda_1^2 x_1^2 & \lambda_2^2 x_2^2 & \lambda_3^2 x_3^2 & \lambda_4^2 x_4^2 \\ \lambda_1^2 y_1^2 & \lambda_2^2 y_2^2 & \lambda_3^2 y_3^2 & \lambda_4^2 y_4^2 \\ \lambda_1^2 w_1^2 & \lambda_2^2 w_2^2 & \lambda_3^2 w_3^2 & \lambda_4^2 w_4^2 \\ \vdots & & & \\ \lambda_1^m x_1^m & \lambda_2^m x_2^m & \lambda_3^m x_3^m & \times \\ \lambda_1^m y_1^m & \lambda_2^m y_2^m & \lambda_3^m y_3^m & \times \\ \lambda_1^m w_1^m & \lambda_2^m w_2^m & \lambda_3^m w_3^m & \times \end{bmatrix}
$$

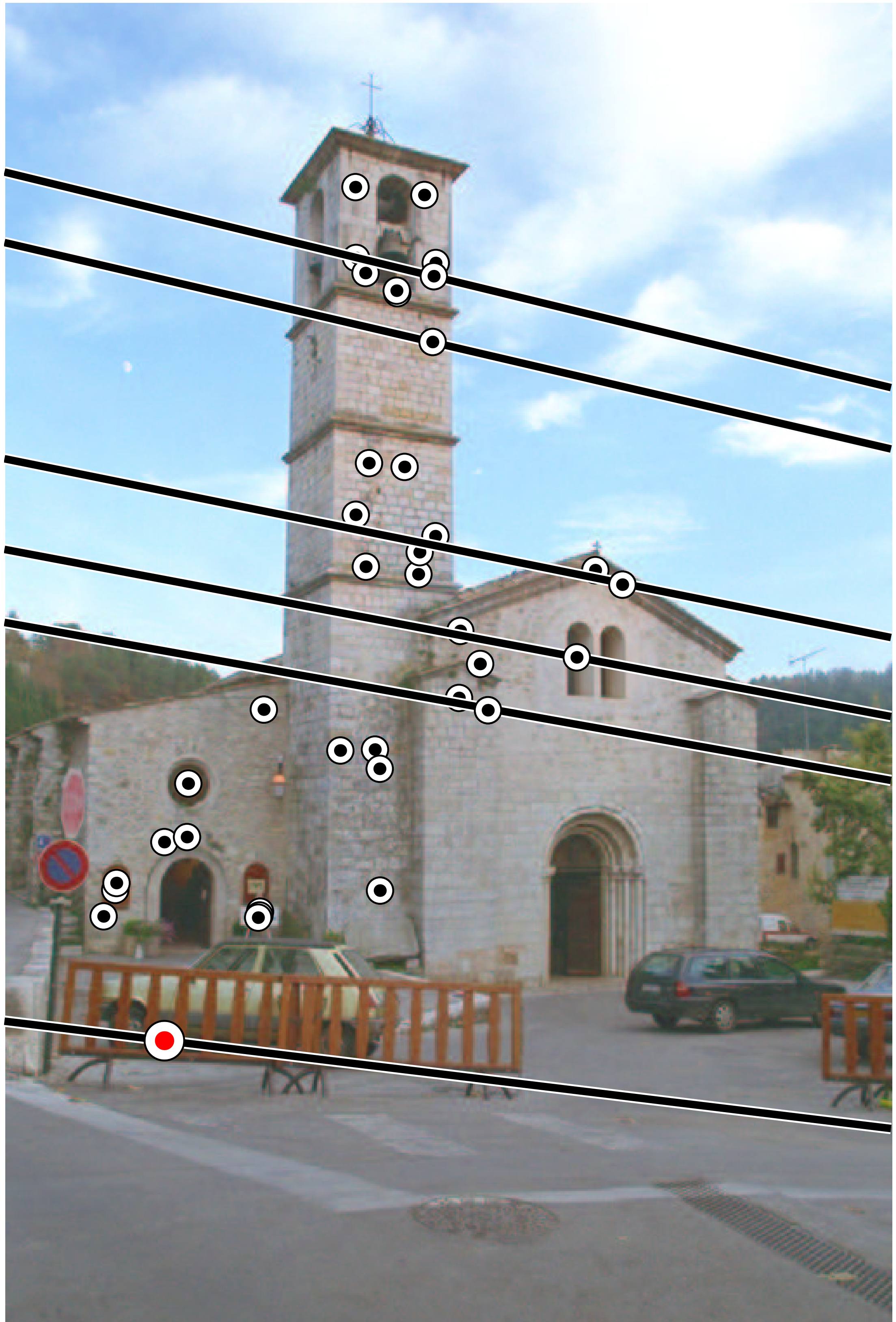
linear hull of all possible fillings

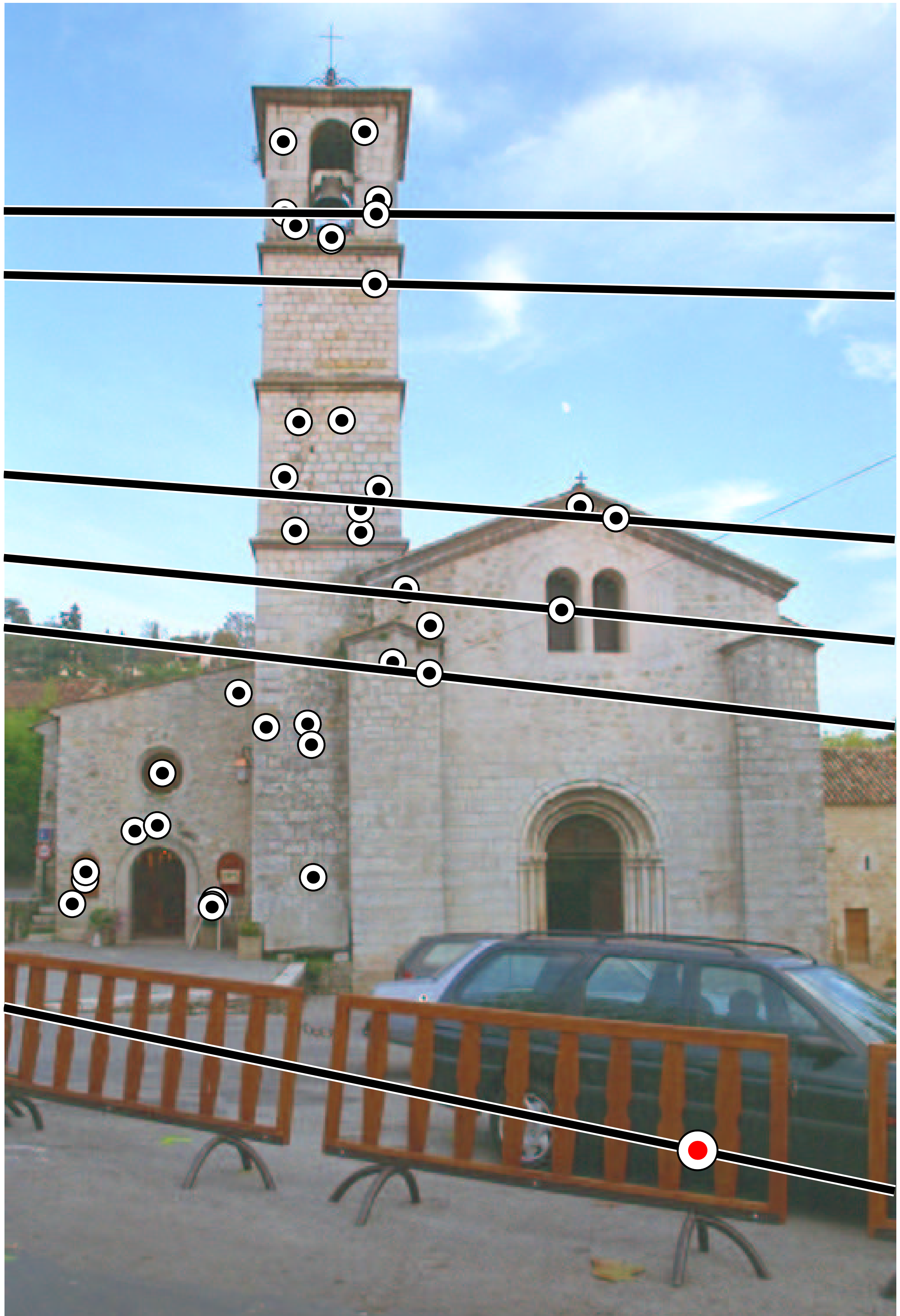
$$
\mathcal{B}_k = Span\left(\begin{bmatrix} 0 & x_1^1 & \lambda_2^1 x_2^1 & \lambda_3^1 x_3^1 & \lambda_4^1 x_4^1 & 0 & 0 & 0 \\ 0 & y_1^1 & \lambda_2^1 y_2^1 & \lambda_3^1 y_3^1 & \lambda_4^1 y_4^1 & 0 & 0 & 0 \\ 0 & w_1^1 & \lambda_2^1 w_2^1 & \lambda_3^1 w_3^1 & \lambda_4^1 w_4^1 & 0 & 0 & 0 \\ \lambda_1^2 x_1^2 & 0 & \lambda_2^2 x_2^2 & \lambda_3^2 x_3^2 & \lambda_4^2 x_4^2 & 0 & 0 & 0 \\ \lambda_1^2 y_1^2 & 0 & \lambda_2^2 y_2^2 & \lambda_3^2 y_3^2 & \lambda_4^2 y_4^2 & 0 & 0 & 0 \\ \lambda_1^2 w_1^2 & 0 & \lambda_2^2 w_2^2 & \lambda_3^2 w_3^2 & \lambda_4^2 w_4^2 & 0 & 0 & 0 \\ \vdots & & & & & & & \\ \lambda_1^m x_1^m & 0 & \lambda_2^m x_2^m & \lambda_3^m x_3^m & 0 & \mathbf{1} & 0 & 0 \\ \lambda_1^m y_1^m & 0 & \lambda_2^m y_2^m & \lambda_3^m y_3^m & 0 & 0 & \mathbf{1} & 0 \\ \lambda_1^m w_1^m & 0 & \lambda_2^m w_2^m & \lambda_3^m w_3^m & 0 & 0 & 0 & \mathbf{1} \end{bmatrix} \right)
$$

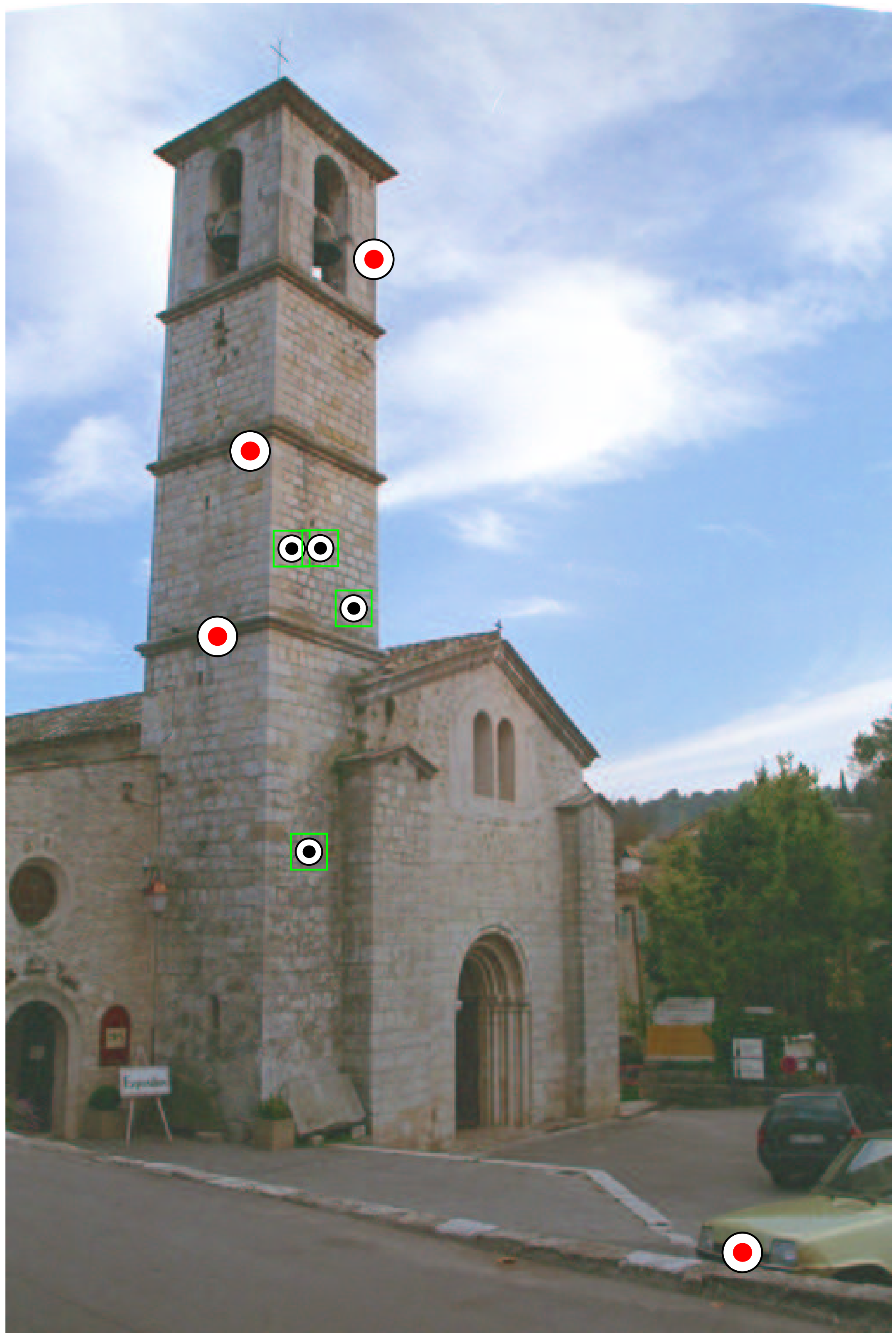


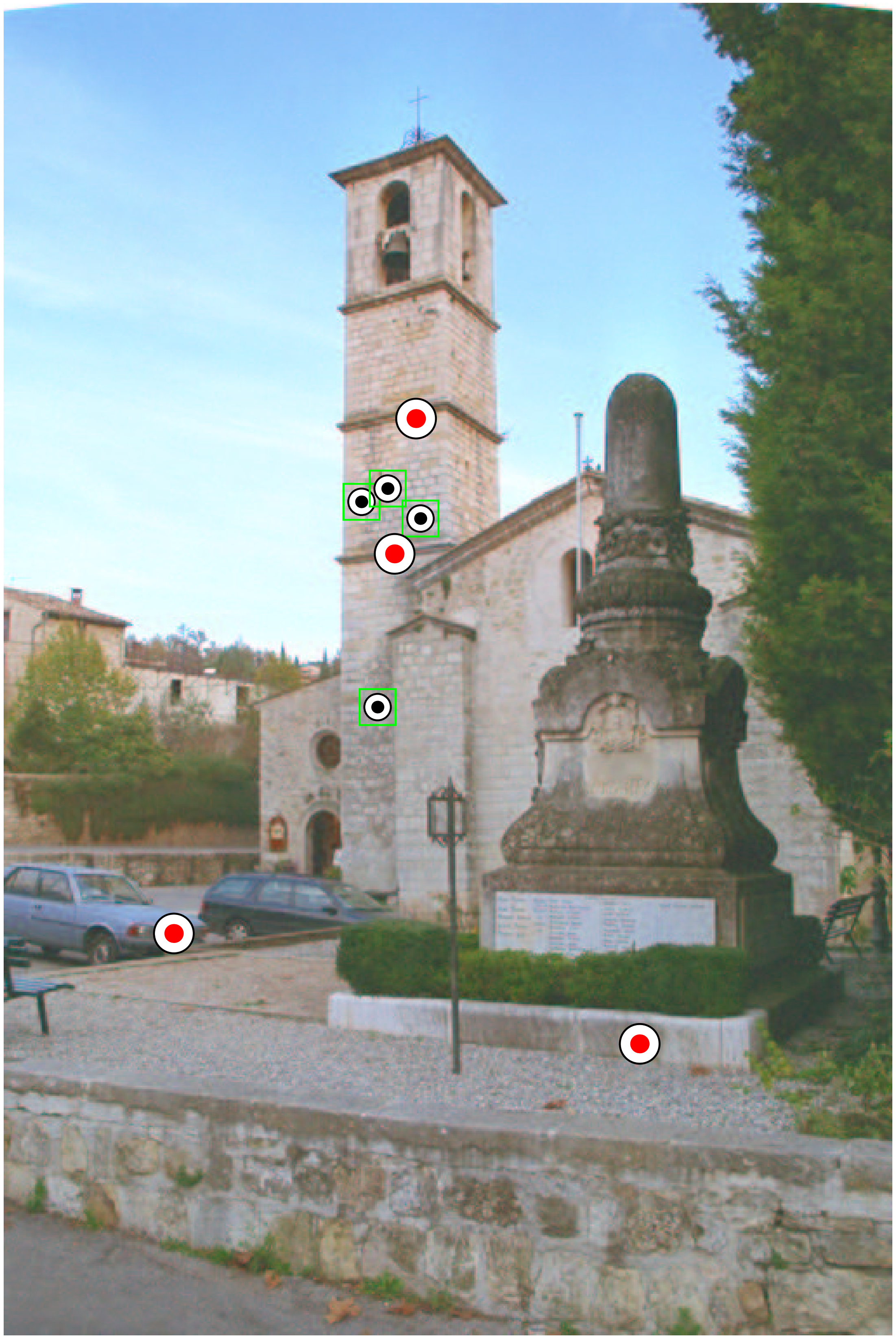


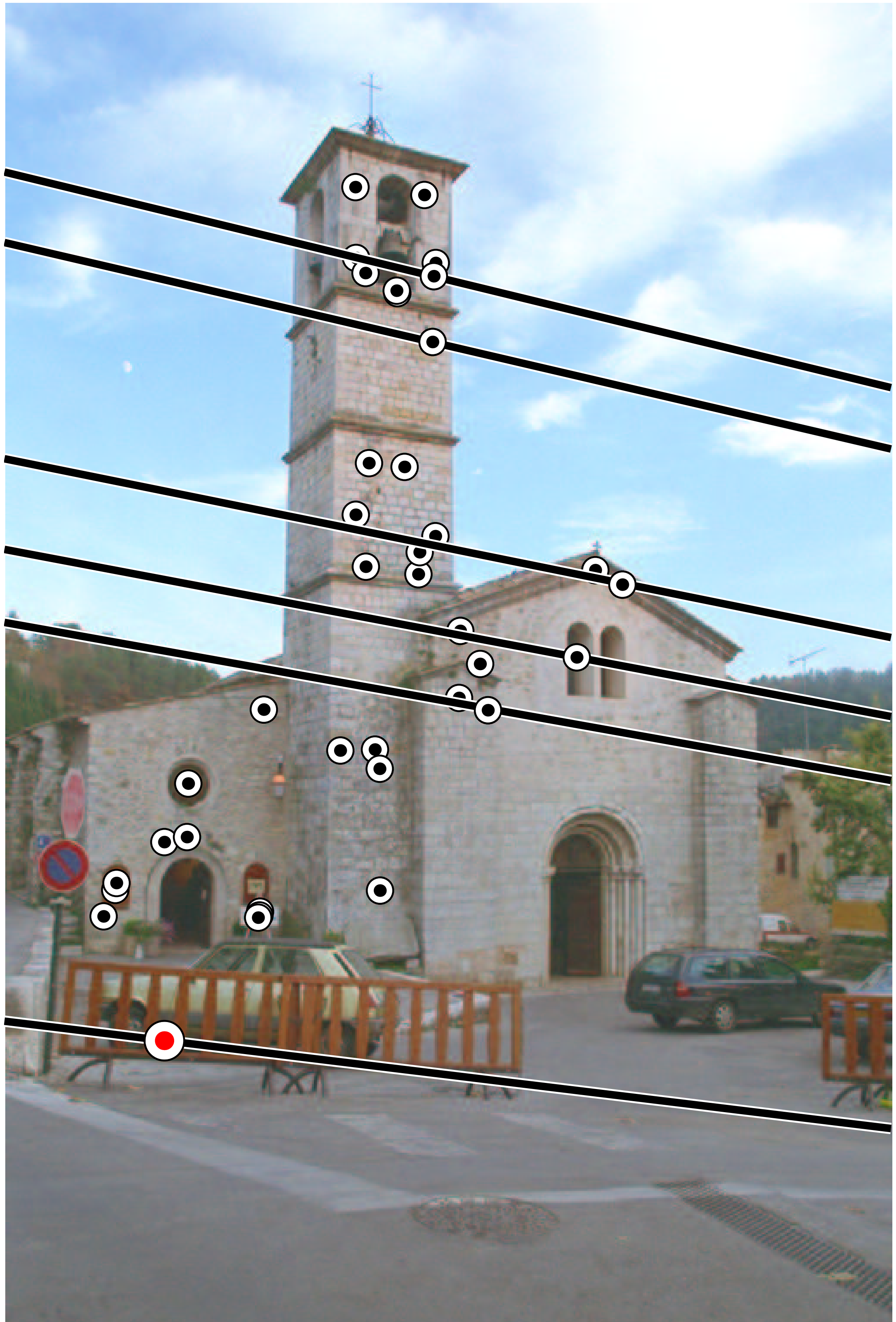


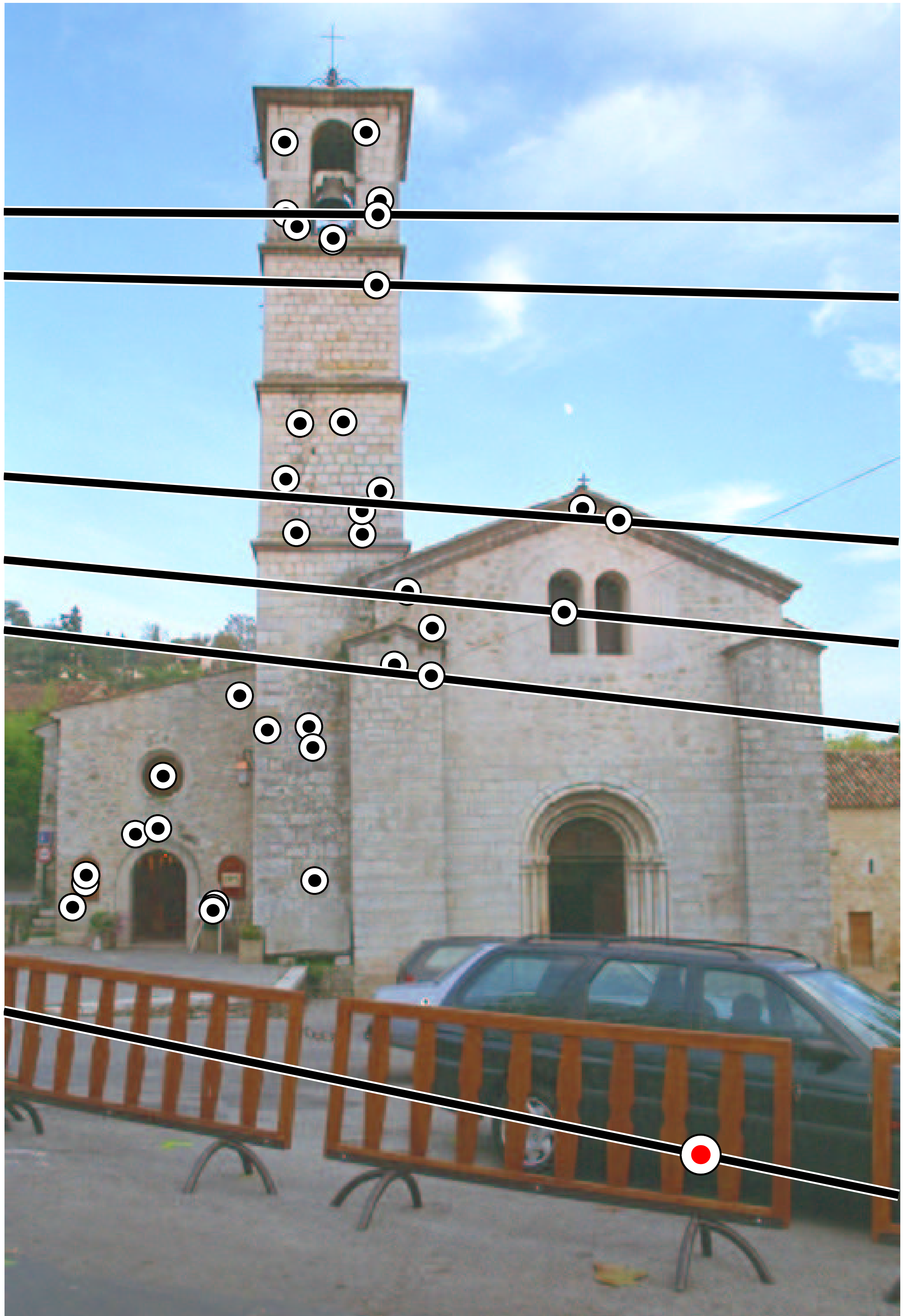


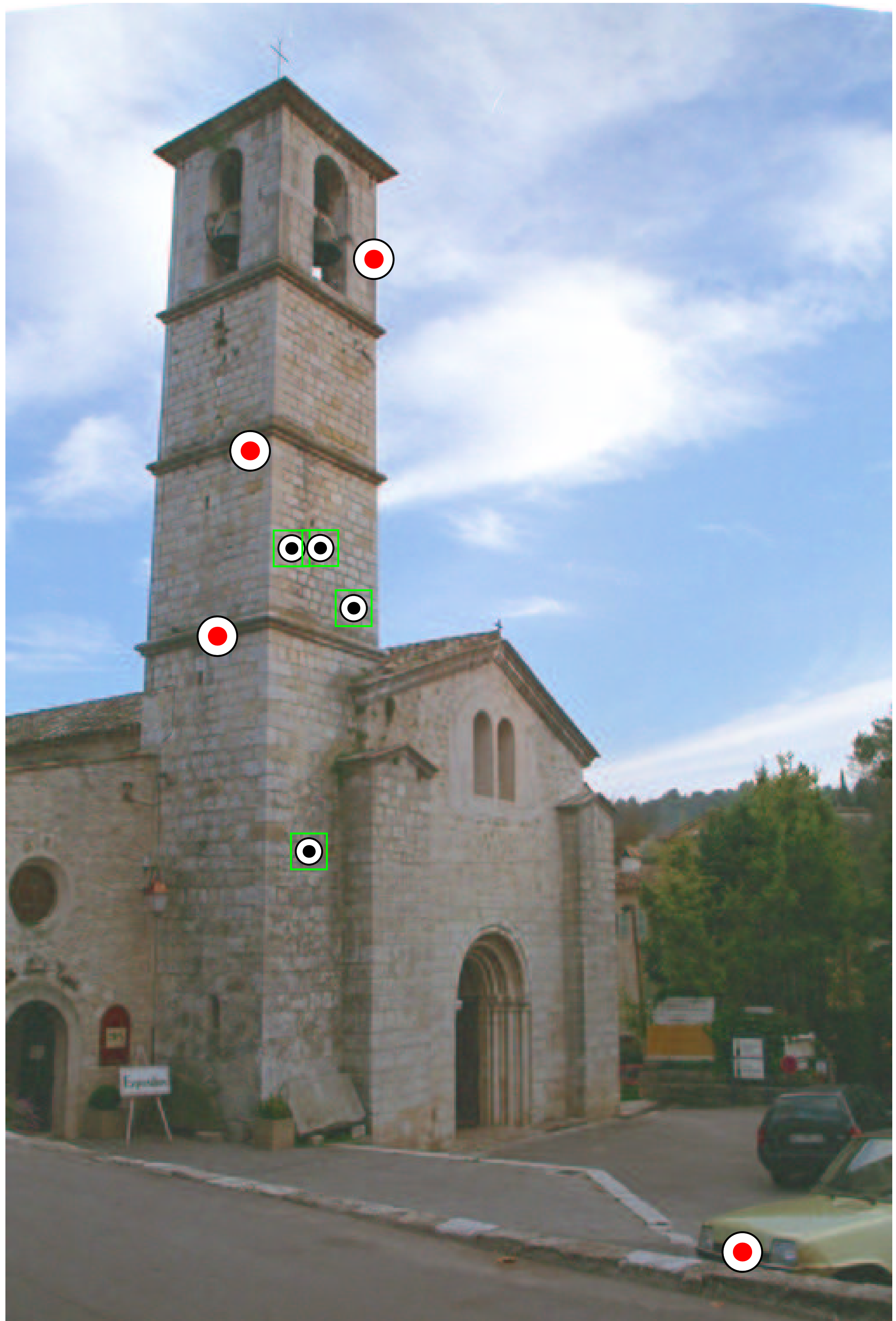


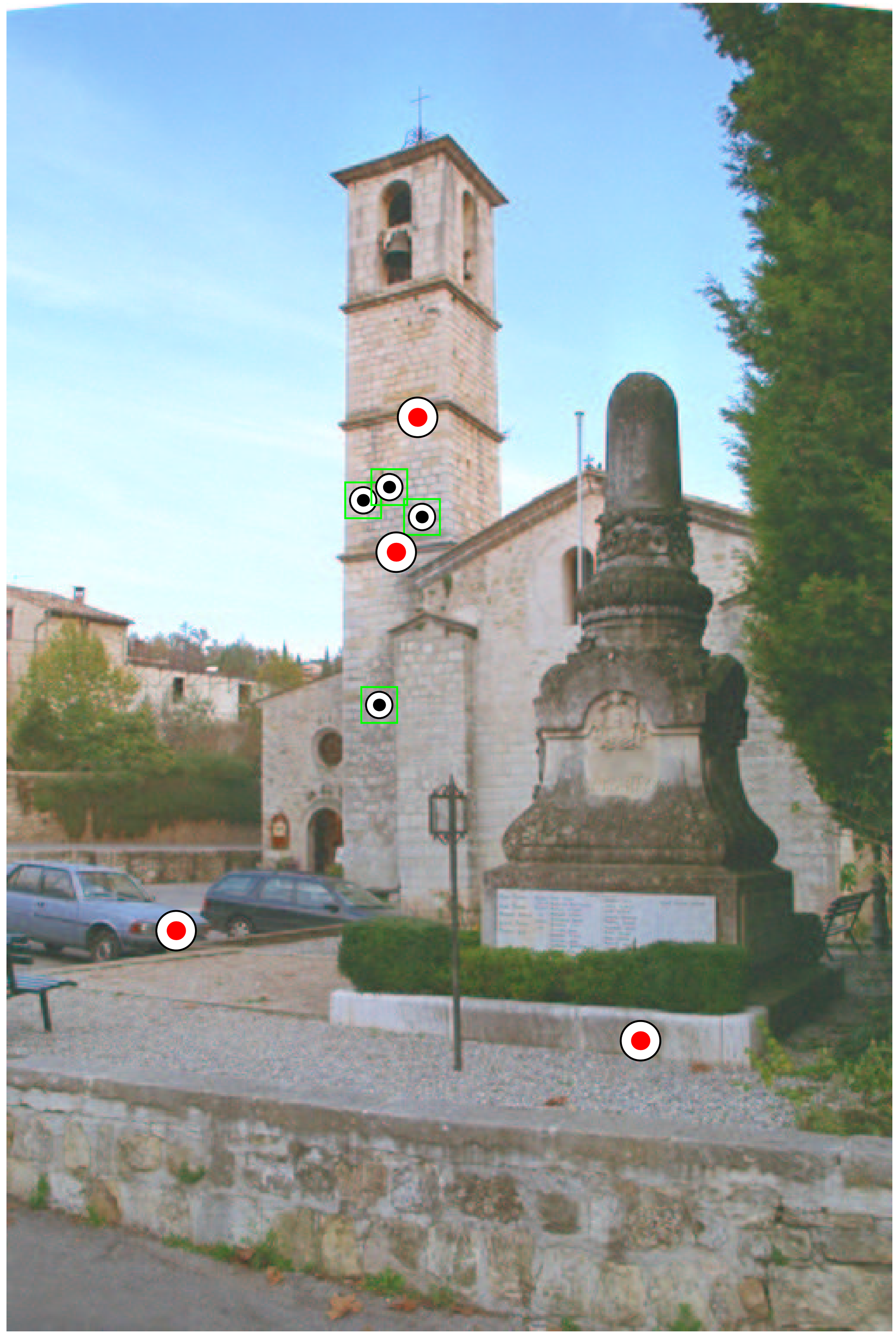


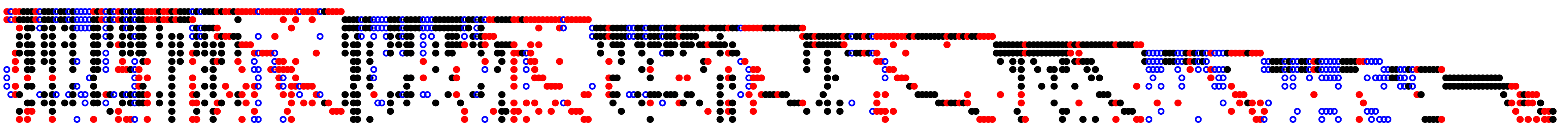


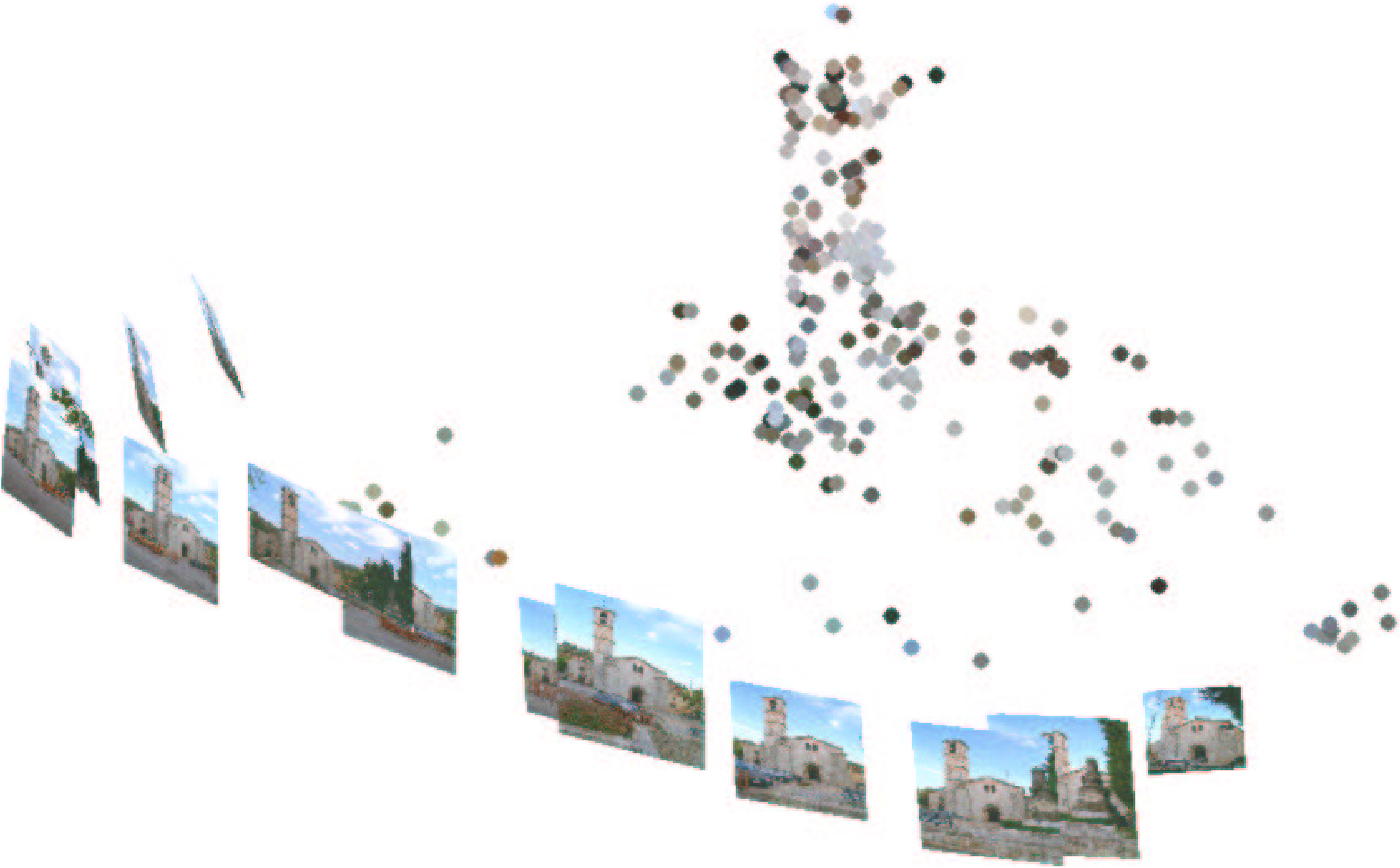


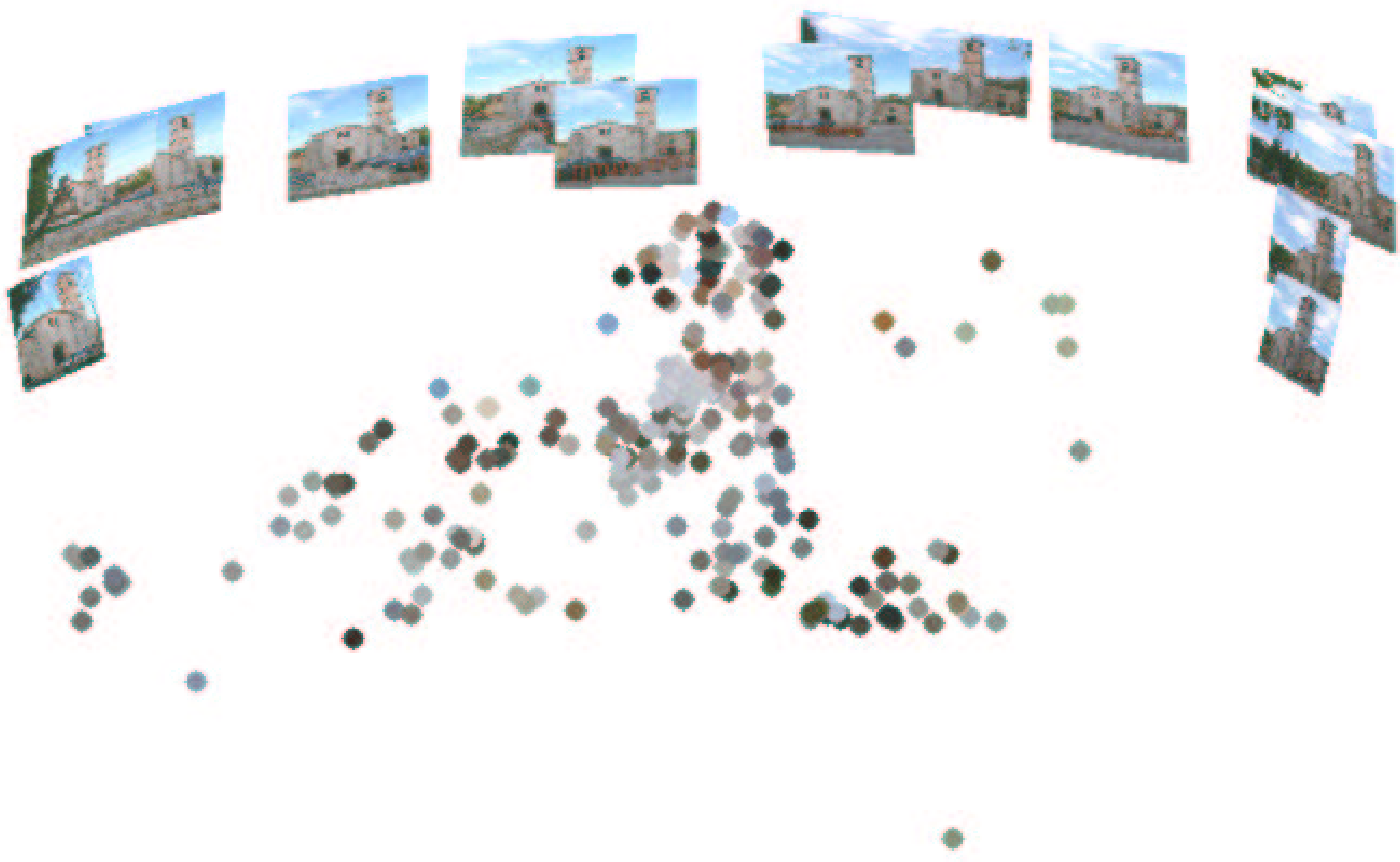




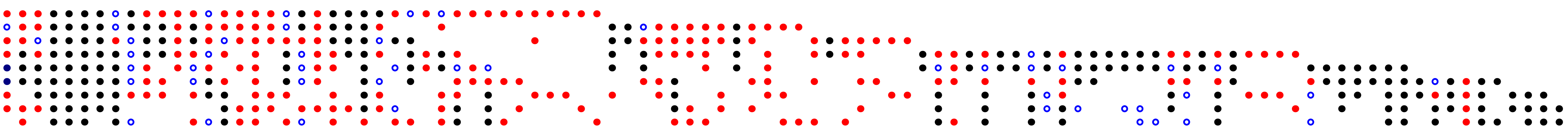









LIFIA




MATTONI