The mathematics of Panorama photography

Oliver Knill, July 11, 2007

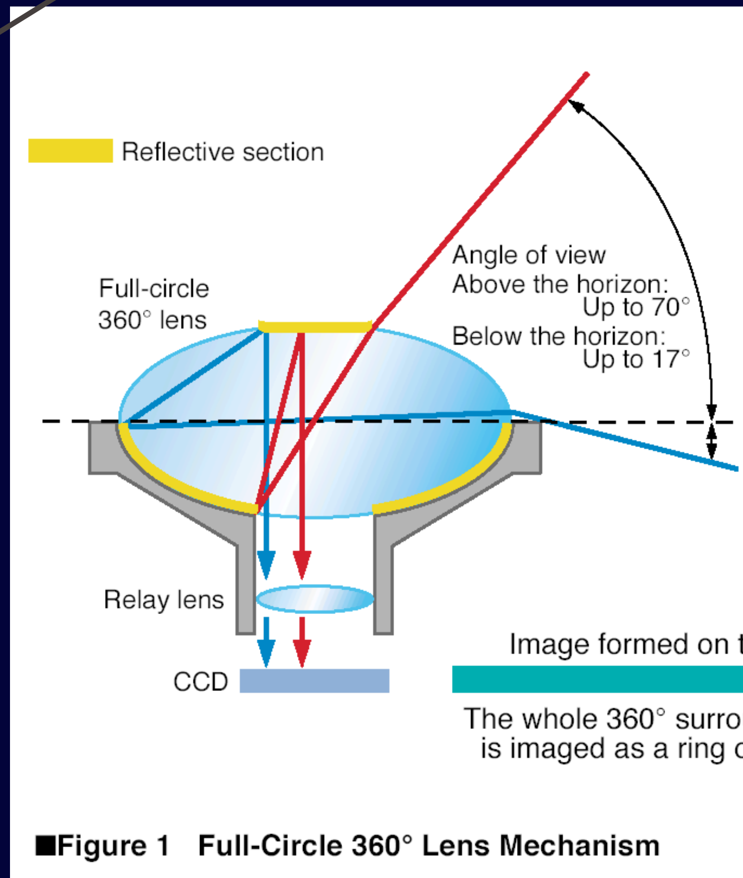
1. Panorama Cameras
2. Math NOW
3. Demonstration
4. Structure from motion
5. Orthographic cameras
6. Omnidirectional cams
7. History pointers
8. Synthetic panoramas
9. Applications

Panorama Cameras

- Catadiopdric systems
- Polydioptric systems
- Rotating cameras
- Traditional cameras and stitching

Catadioptric systems

0-360 panoramic optic



Sony full circle

id mind

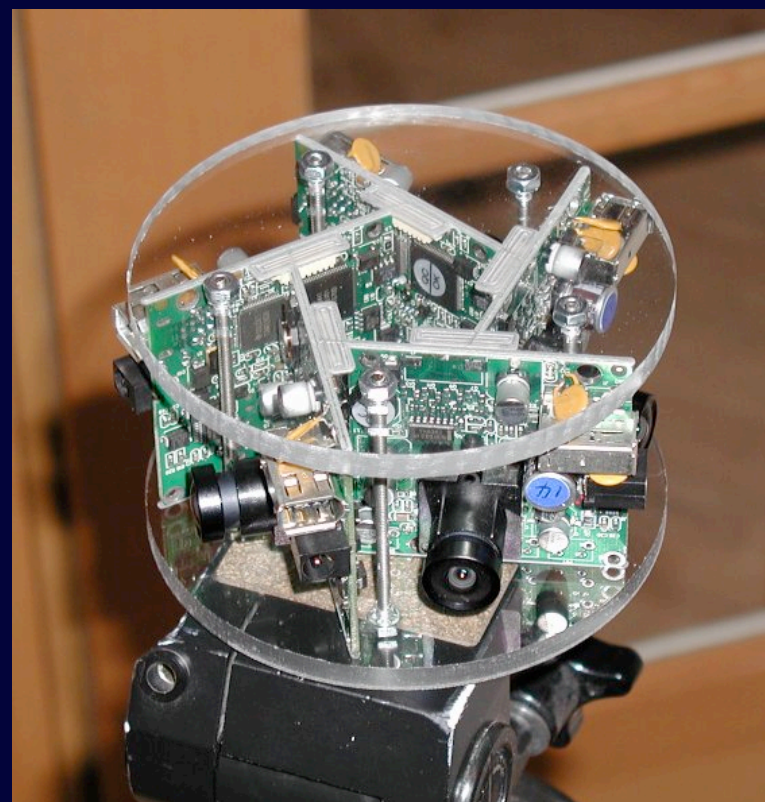
crucial: vertical field of view

Example

Arlington
jan 2006

Polydioptric systems

Microsoft



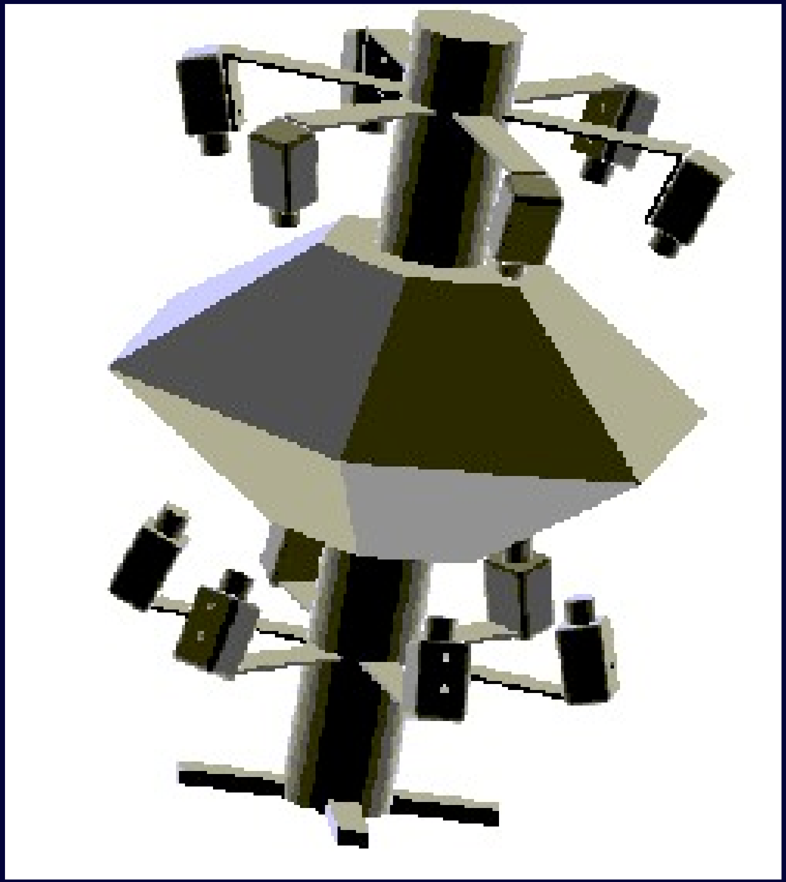
ring cam

google

U-Penn

Sphere cam

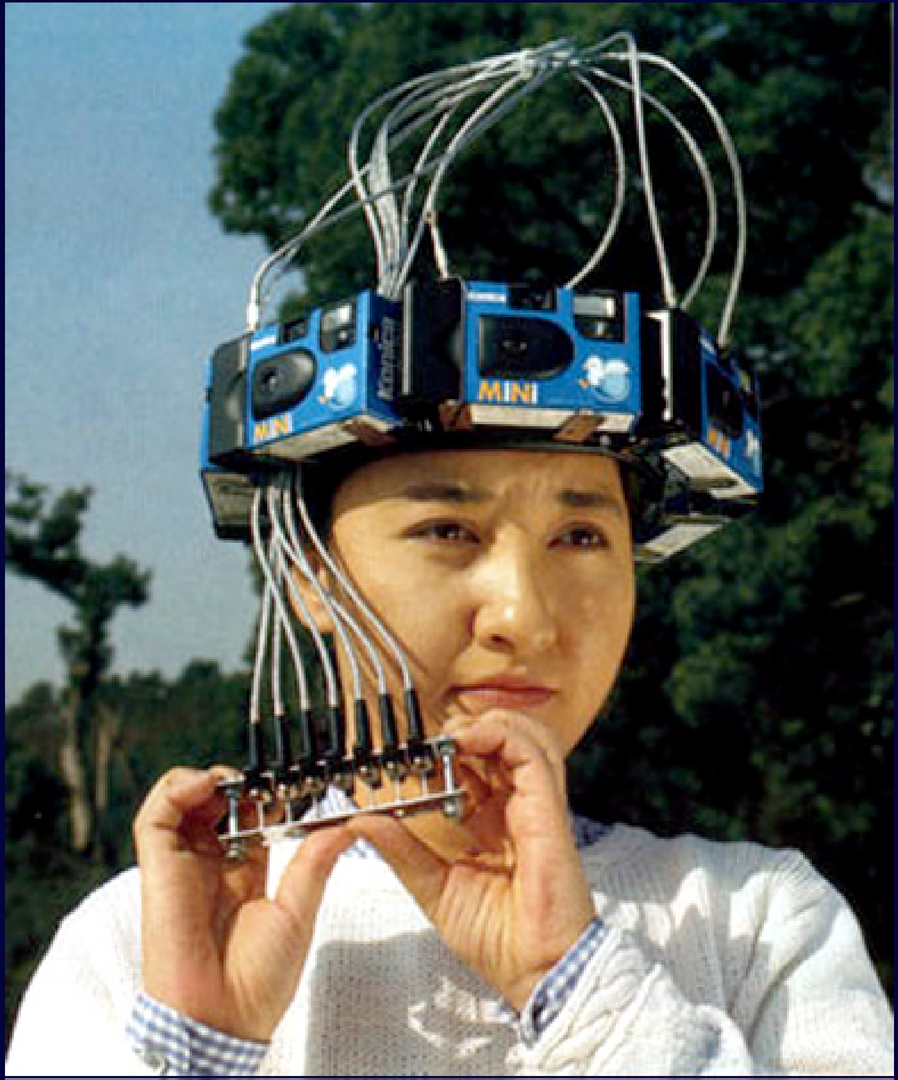
Mixed systems



mirrors and
multiple cameras

university of arizona

My favorate multicam:



wacky

Rotating cameras

Seitz

cedric

ipix

Expensive, but can produce fantastic quality pictures with professional cameras

Panorama

Airbus example

Panorama


Bergrestaurant Sillerenbühl
Lohner
Altels
Rinderhorn
Wildstrubel
Plaine Morte
Hahnenmoospass
Adelboden
©virtualpanorama.ch

2006 www.erikleeman.com

Google street map


19th Ave
N
S

Last, but not at least:

Omni cameras in nature:

damselfly


dragonfly

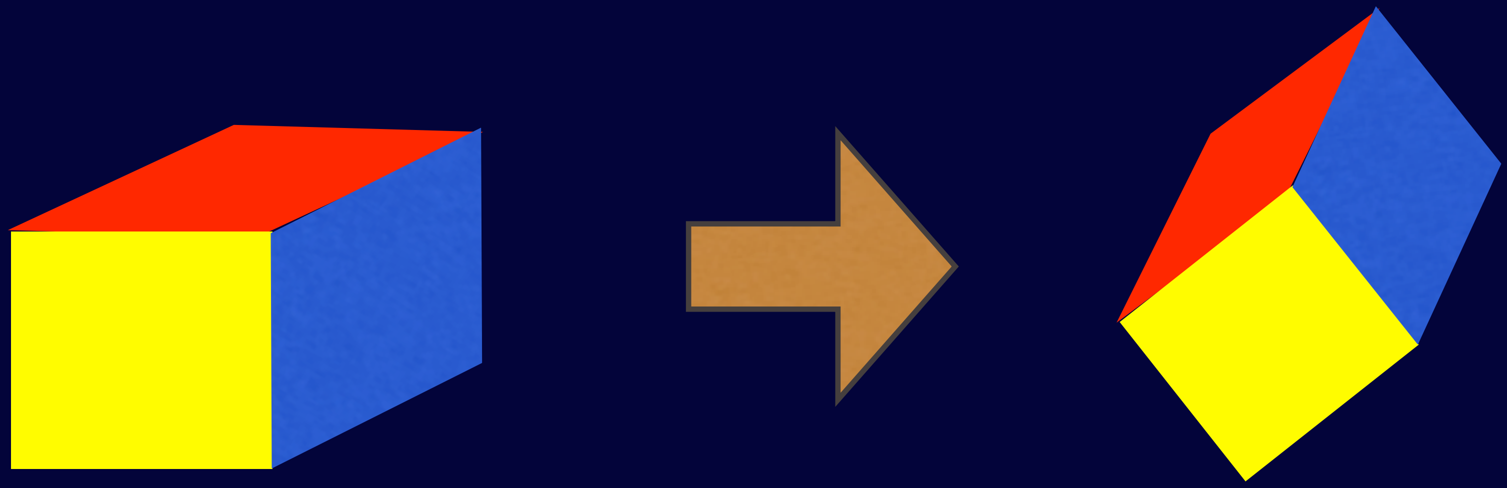

damselfly

Math NOW!

For the following, it is important to understand something about the mathematics of rotation translation, etc. These operations are central in geometry.

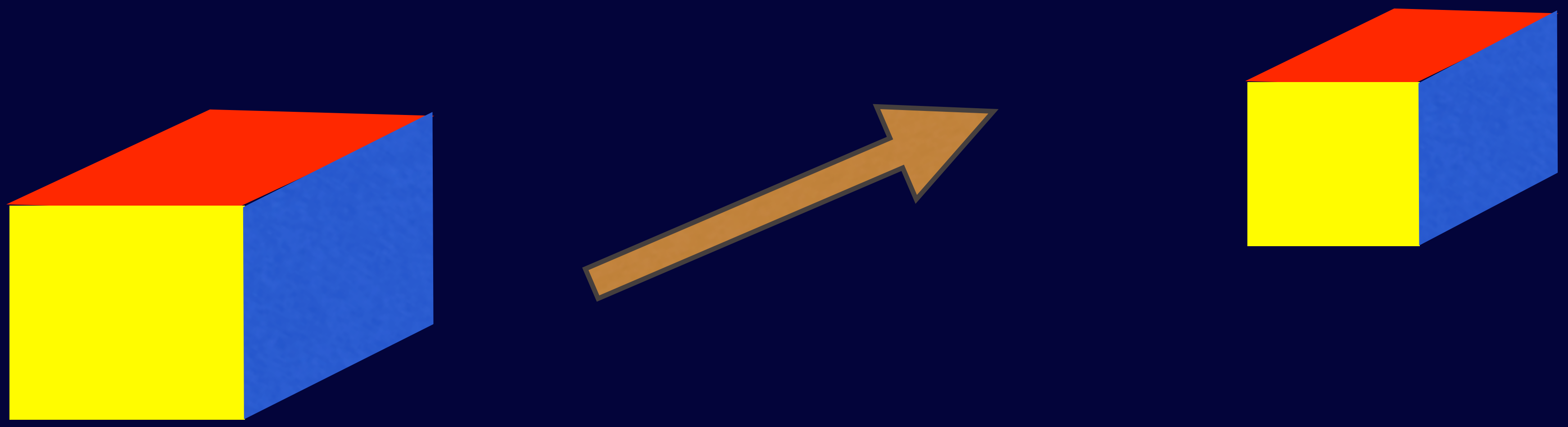
Rotation

A rotation in space is described by 3 angles. Rotations form a group, one can compose rotations to get a new rotation.



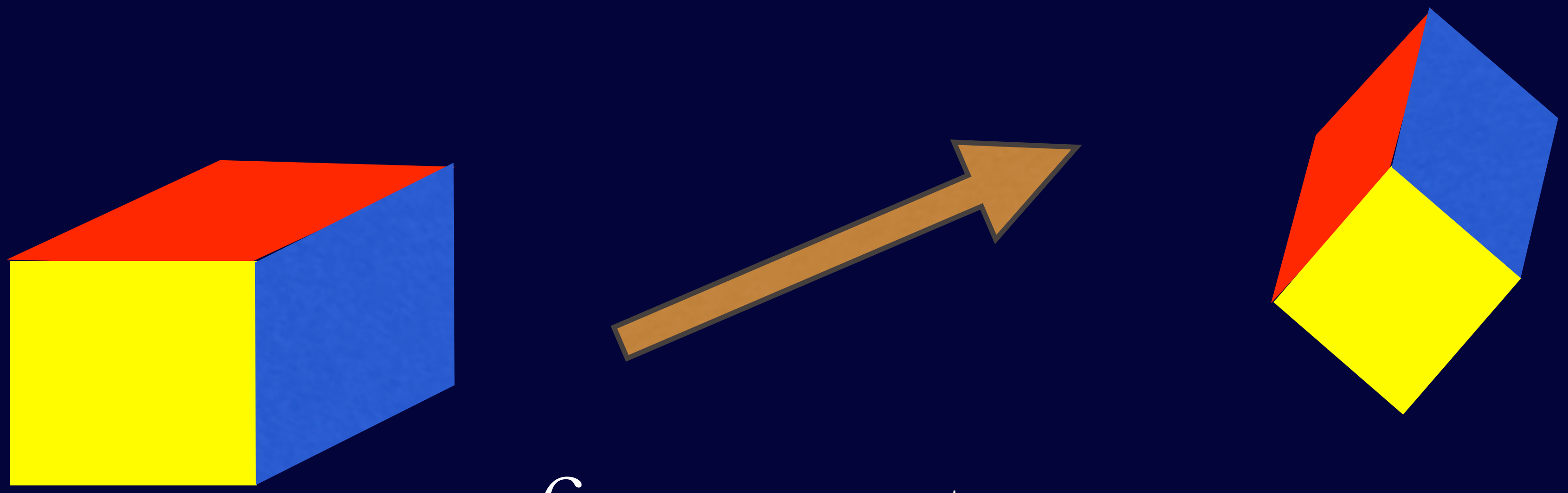
Translation

Also a translation is described by 3 paramaeters. Translations form a group too.



Euclidean transforms

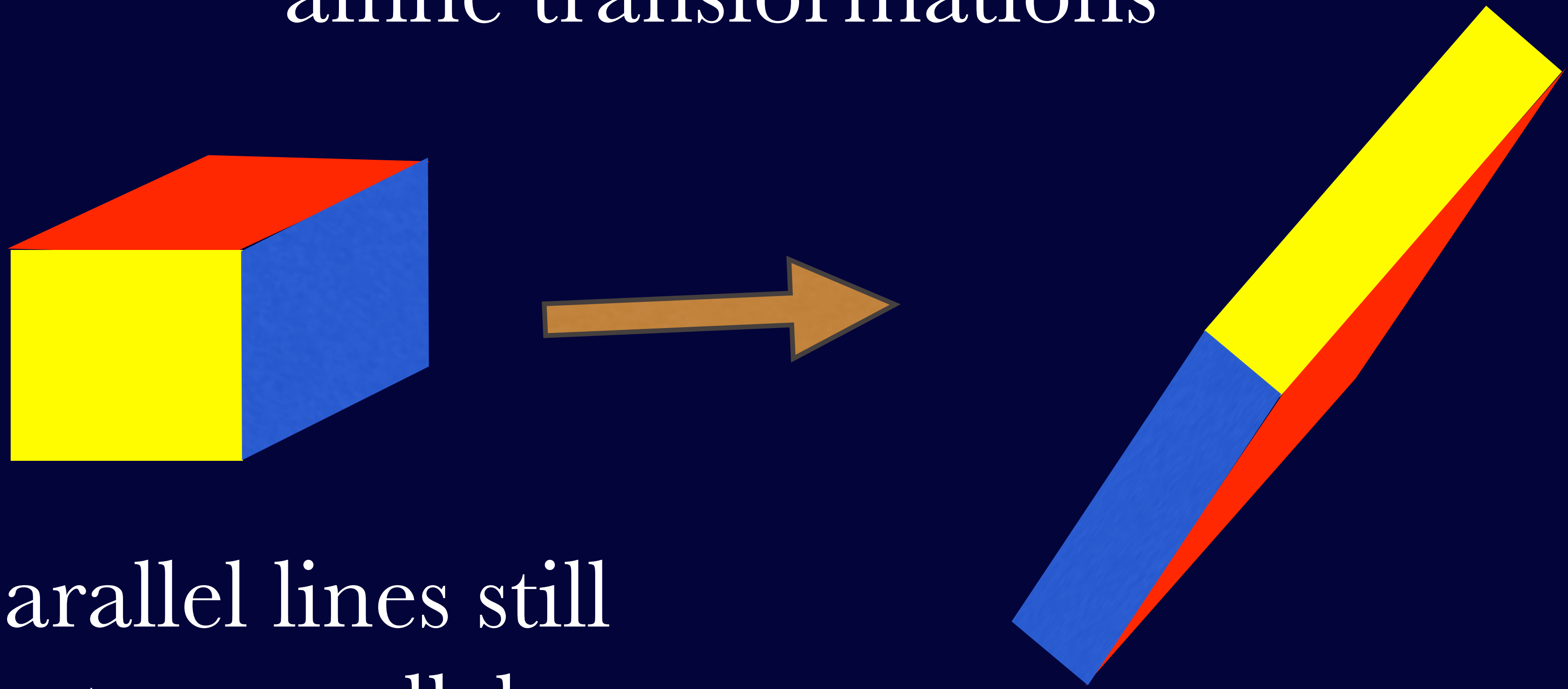
Translations and rotations generate the group of Euclidean transformations



6 parameters

Affine transforms

If we also can shear and scale, we get affine transformations

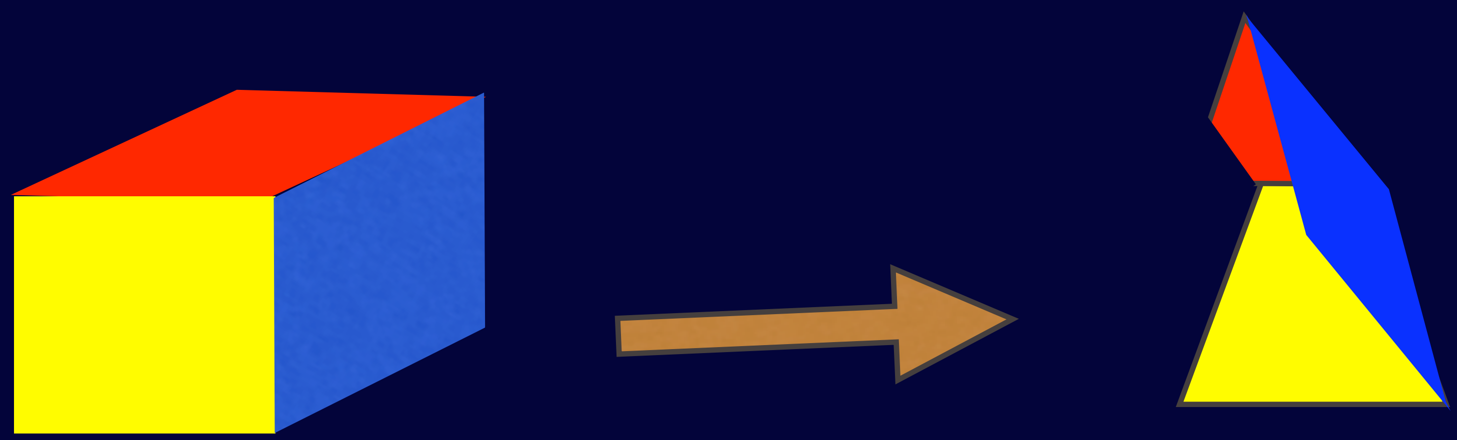


parallel lines still stay parallel

12 parameters

Projective transforms

We even allow perspective transformations



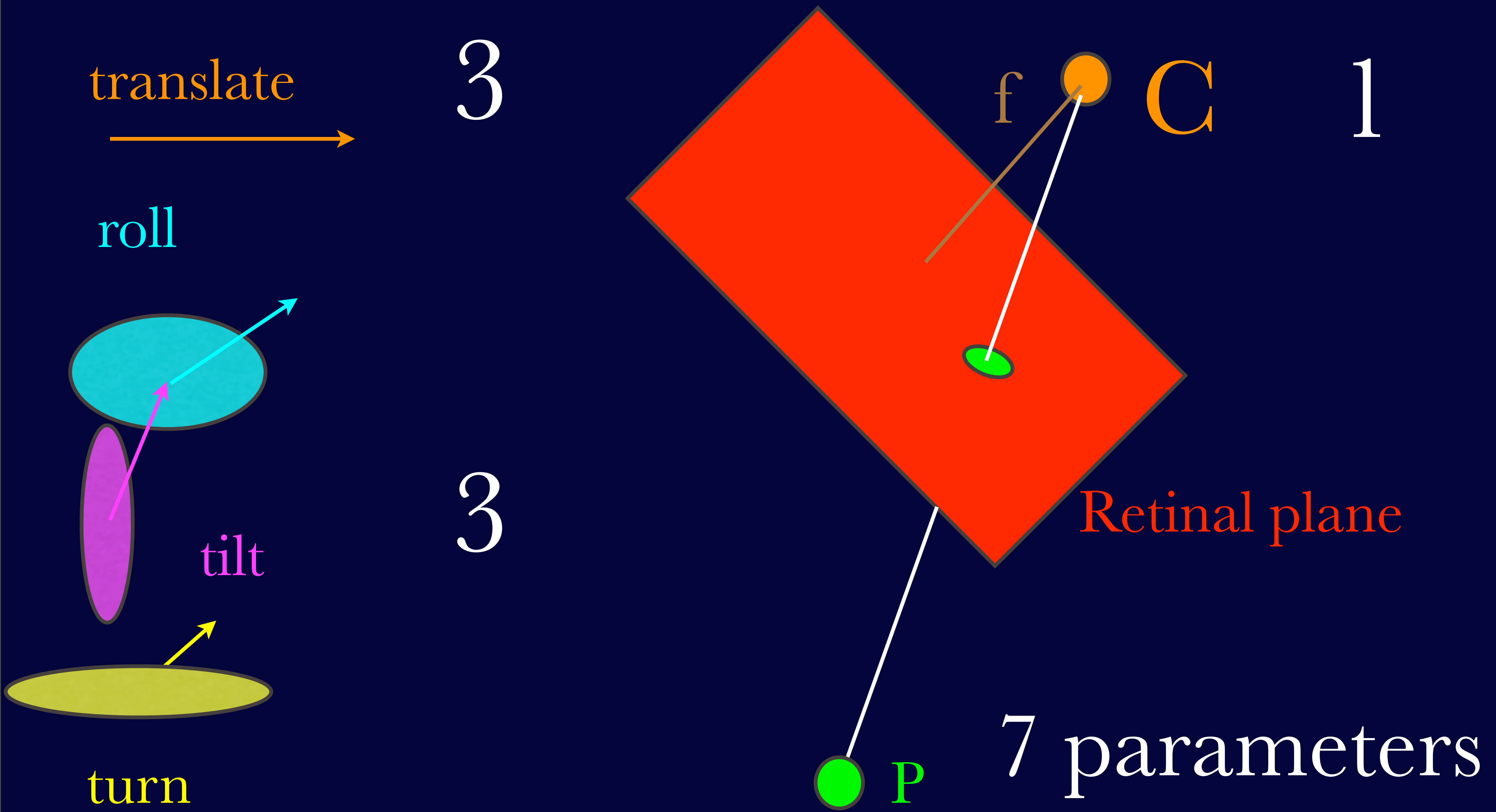
parallel lines stay no more parallel in general

15 parameters

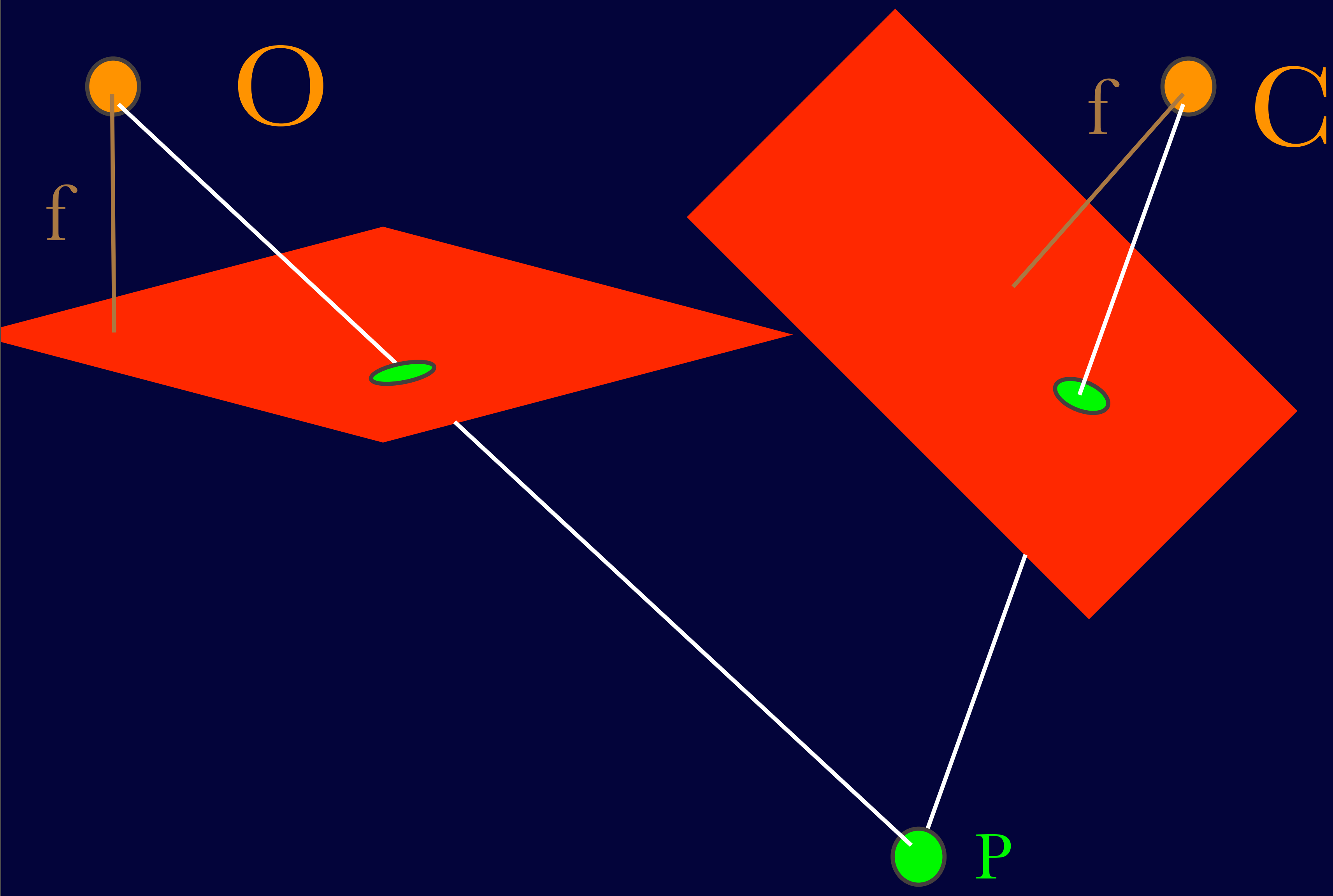
Coordinates and camera types

The mathematics of photography is most developed for perspective cameras, because most cameras can be modeled as perspective cameras

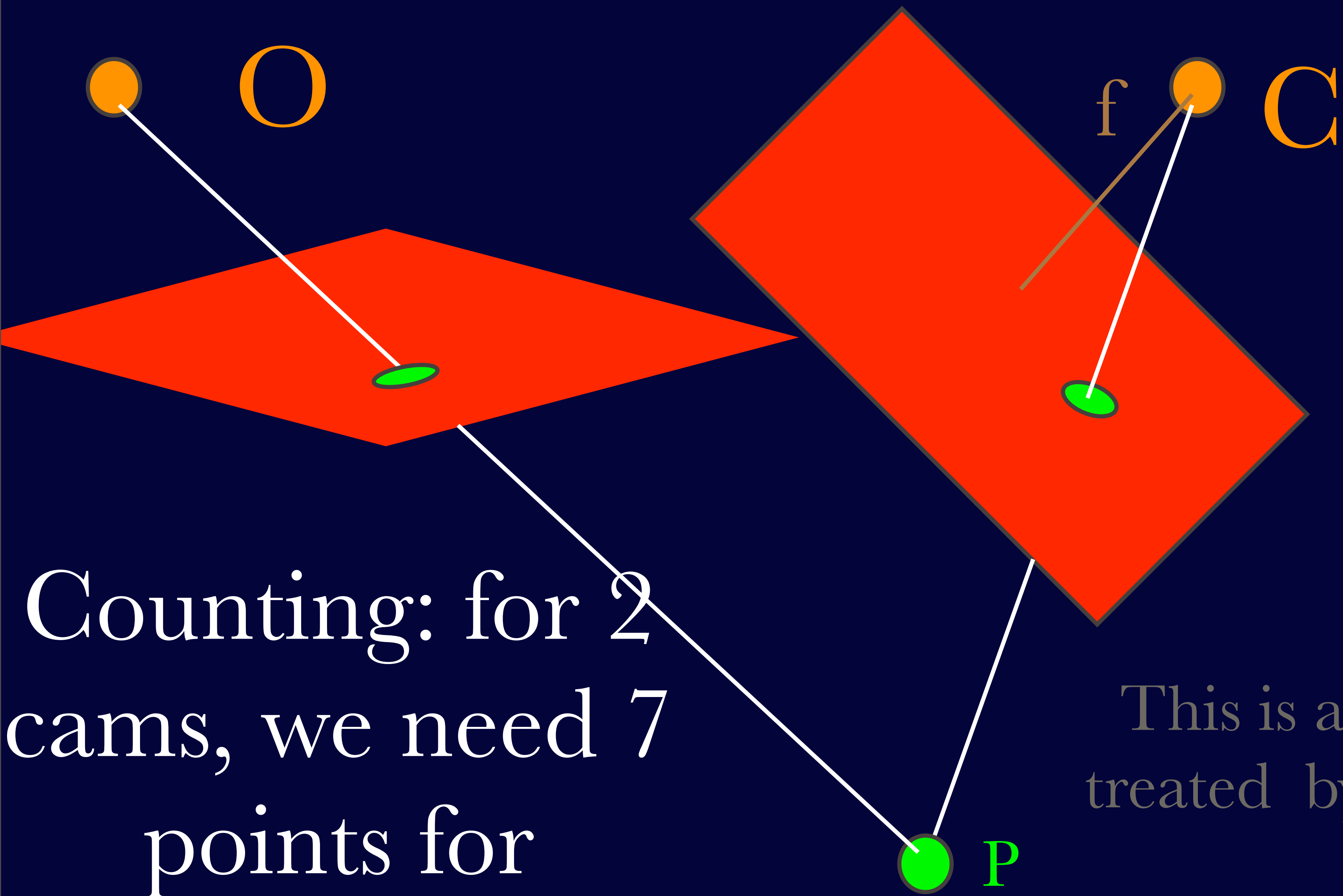
Traditional camera



2 cams how many pts?



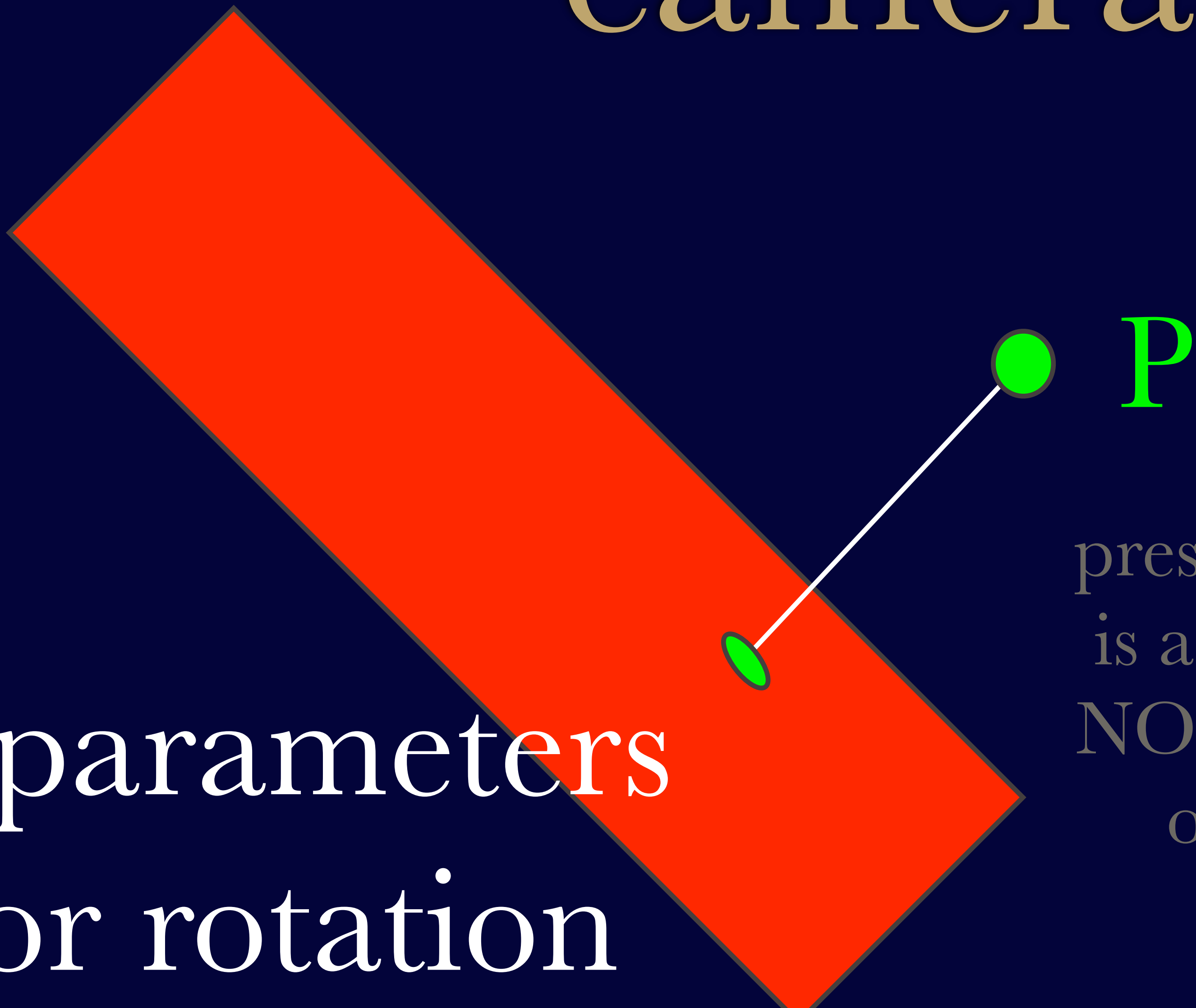
Every point introduces 3 unknowns and 4 picture coordinates



Counting: for 2 cams, we need 7 points for reconstruction

This is a situation treated by Darboux

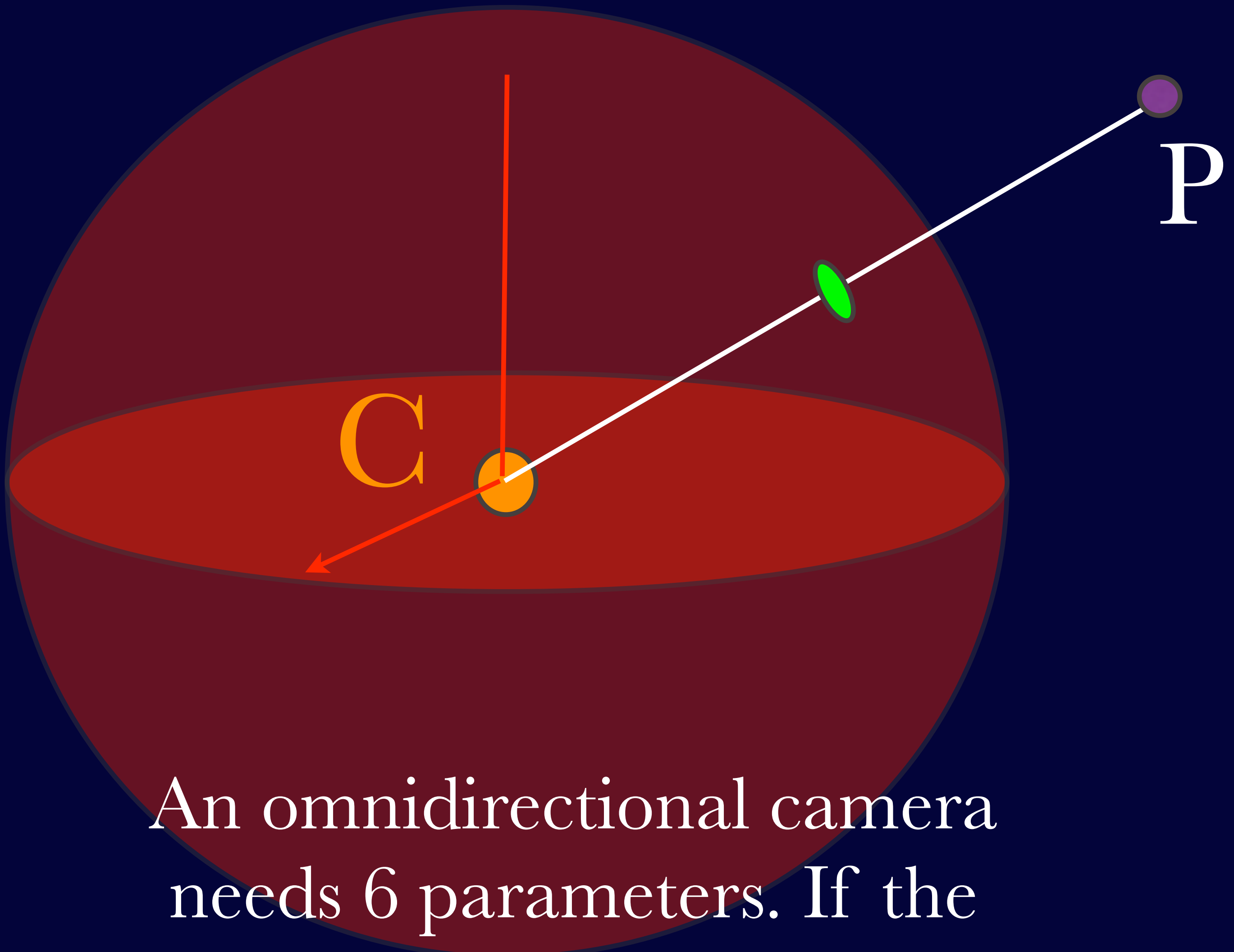
Orthographic camera



preserves parallelism
is an affine camera.
NO definite location
of the camera!

3 parameters
for rotation

Spherical camera



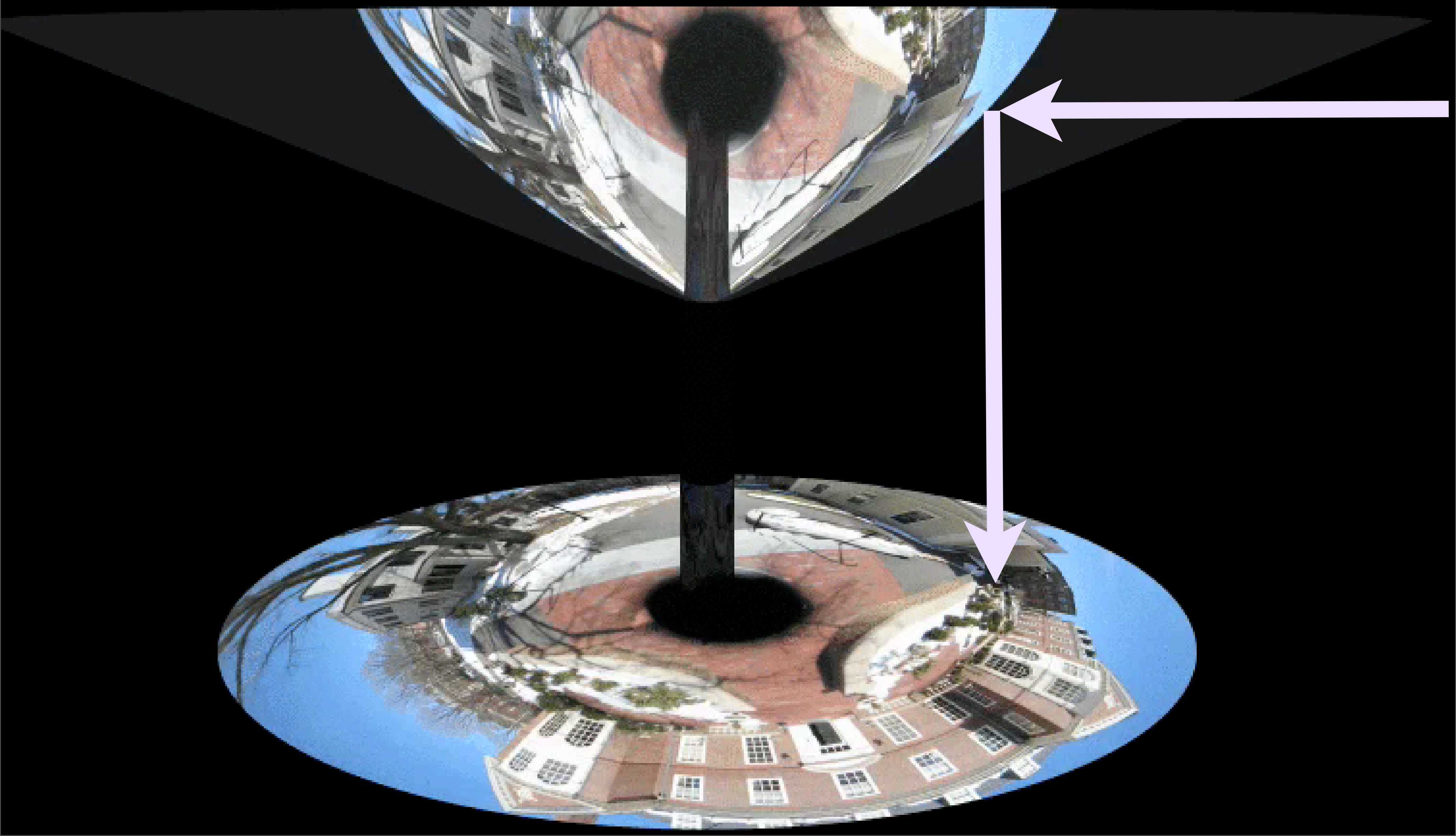
An omnidirectional camera needs 6 parameters. If the orientation is known, then only 3 parameters are needed: the position

Image unwrapping

Camera picture

This is what the camera sees!

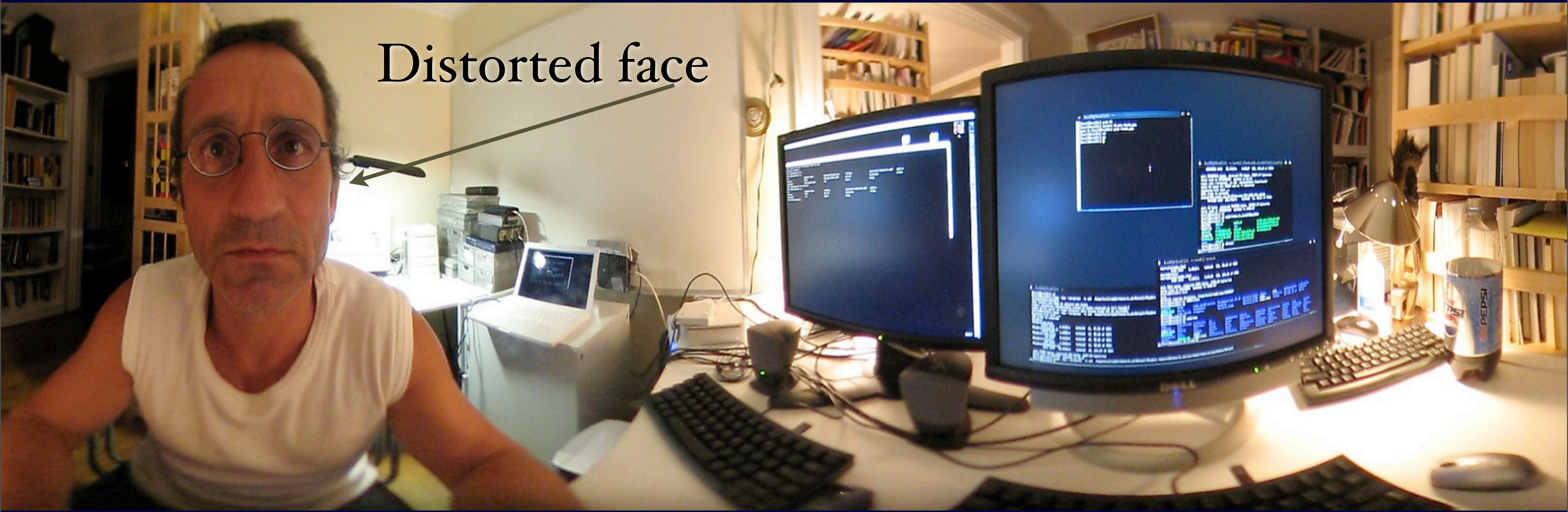
How do we unwrap it?



to get pictures like :

Olivers Office

Grandmothers kitchen


Distorted face

The key are polar coordinates

$$x = r\cos(\theta)$$

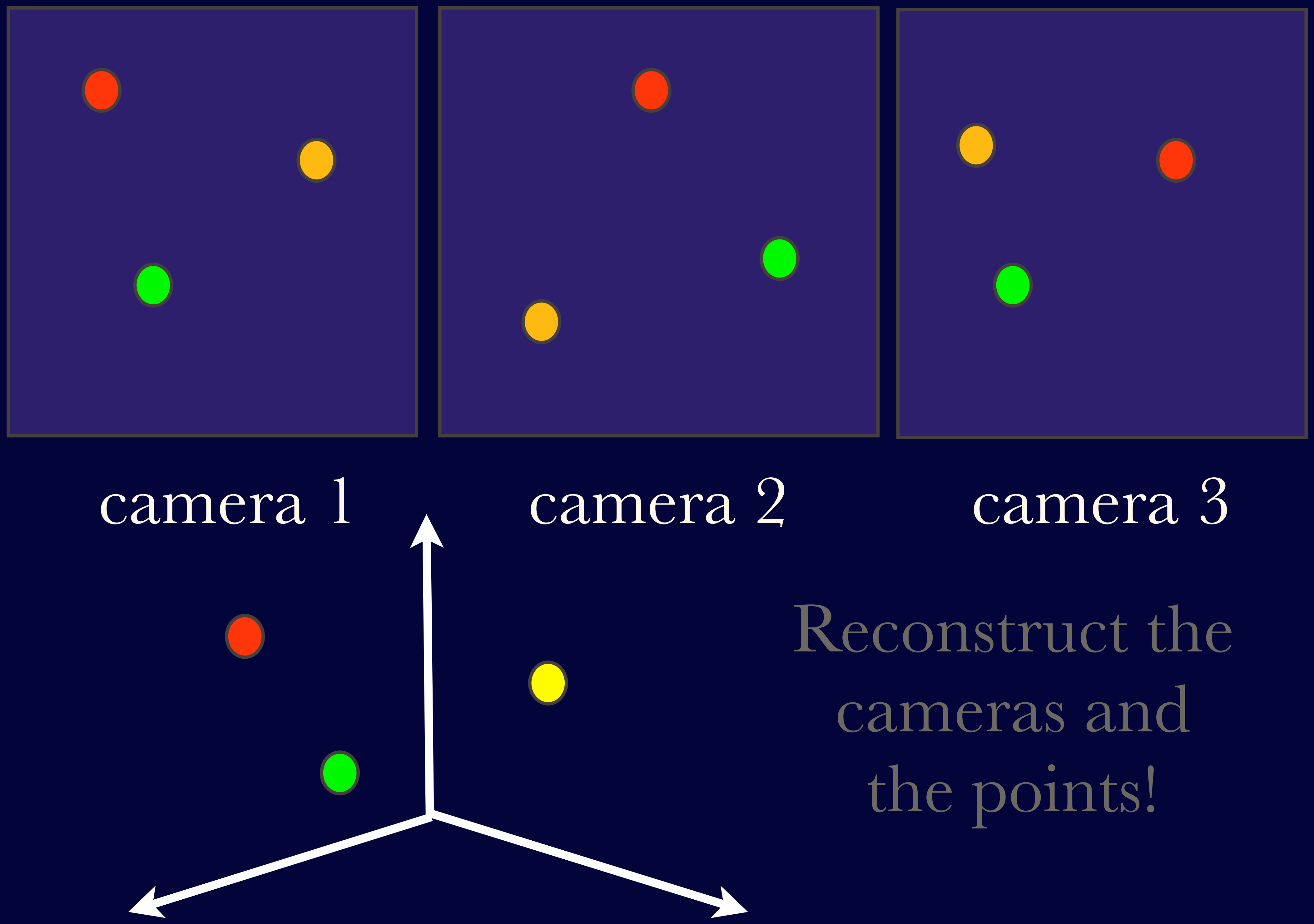
$$y = r\sin(\theta)$$

Demonstration

Viewing panoramas Virtual Reality

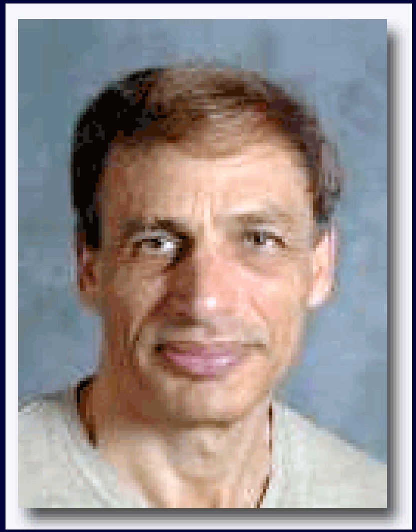
Building a cube:

Correspondence problem



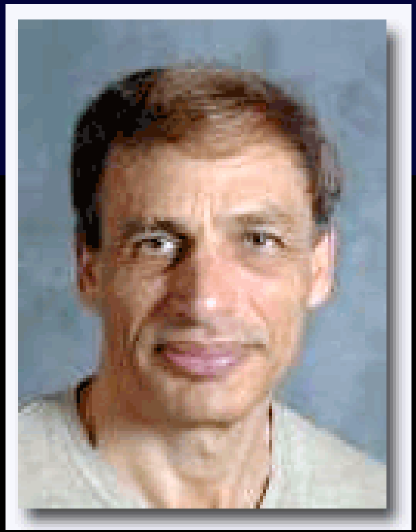
Reconstruct the cameras and the points!

Orthographic cameras



Have 3 cameras
How many
points
are needed?

3 cameras 4 points



193

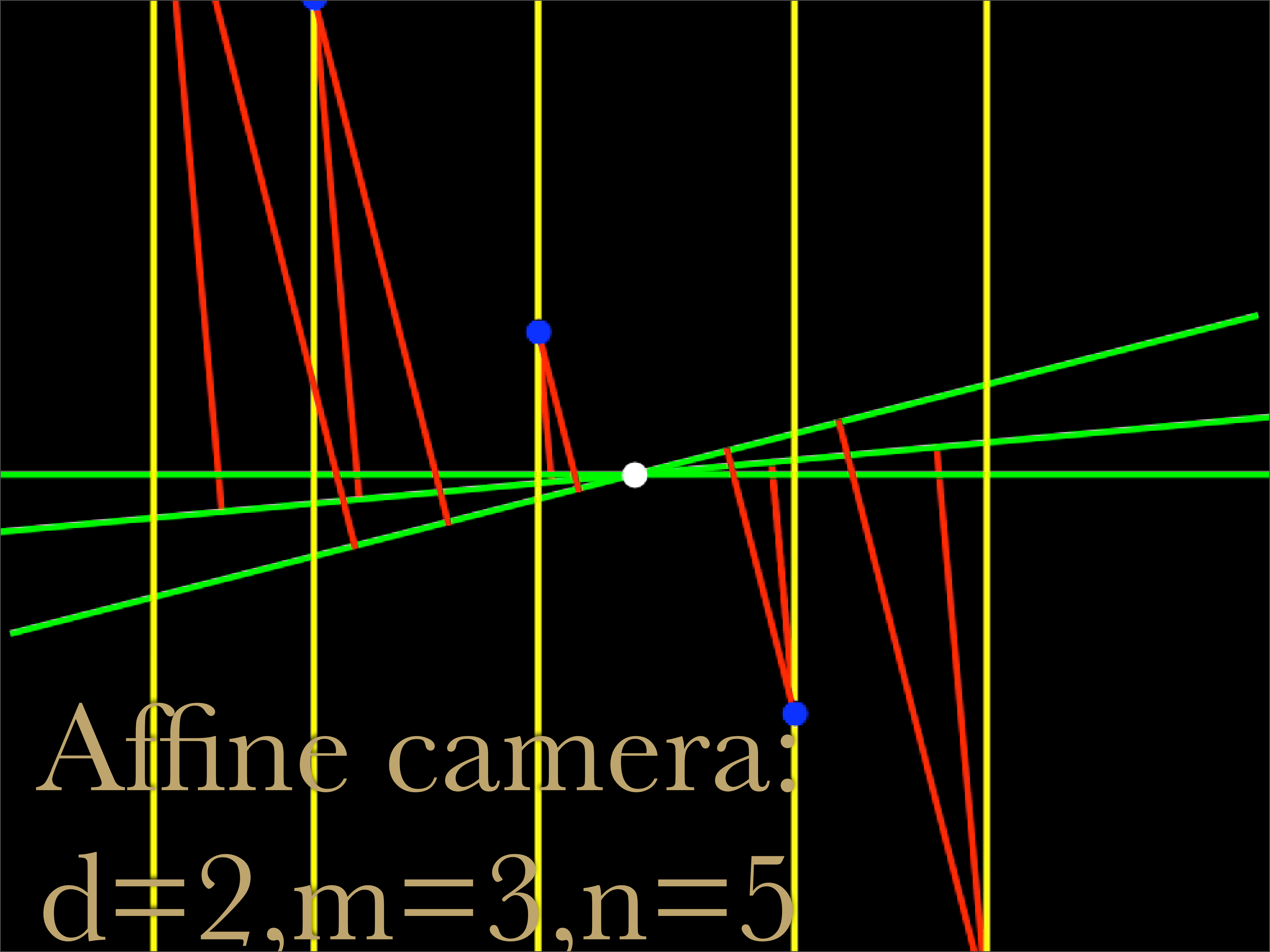
APPENDIX 1

THE STRUCTURE FROM MOTION THEOREM

The structure from motion theorem:

Given three distinct orthographic projections of four non-coplanar points in a rigid configuration, the structure and motion compatible with the three views are uniquely determined up to a reflection about the image plane.

1979

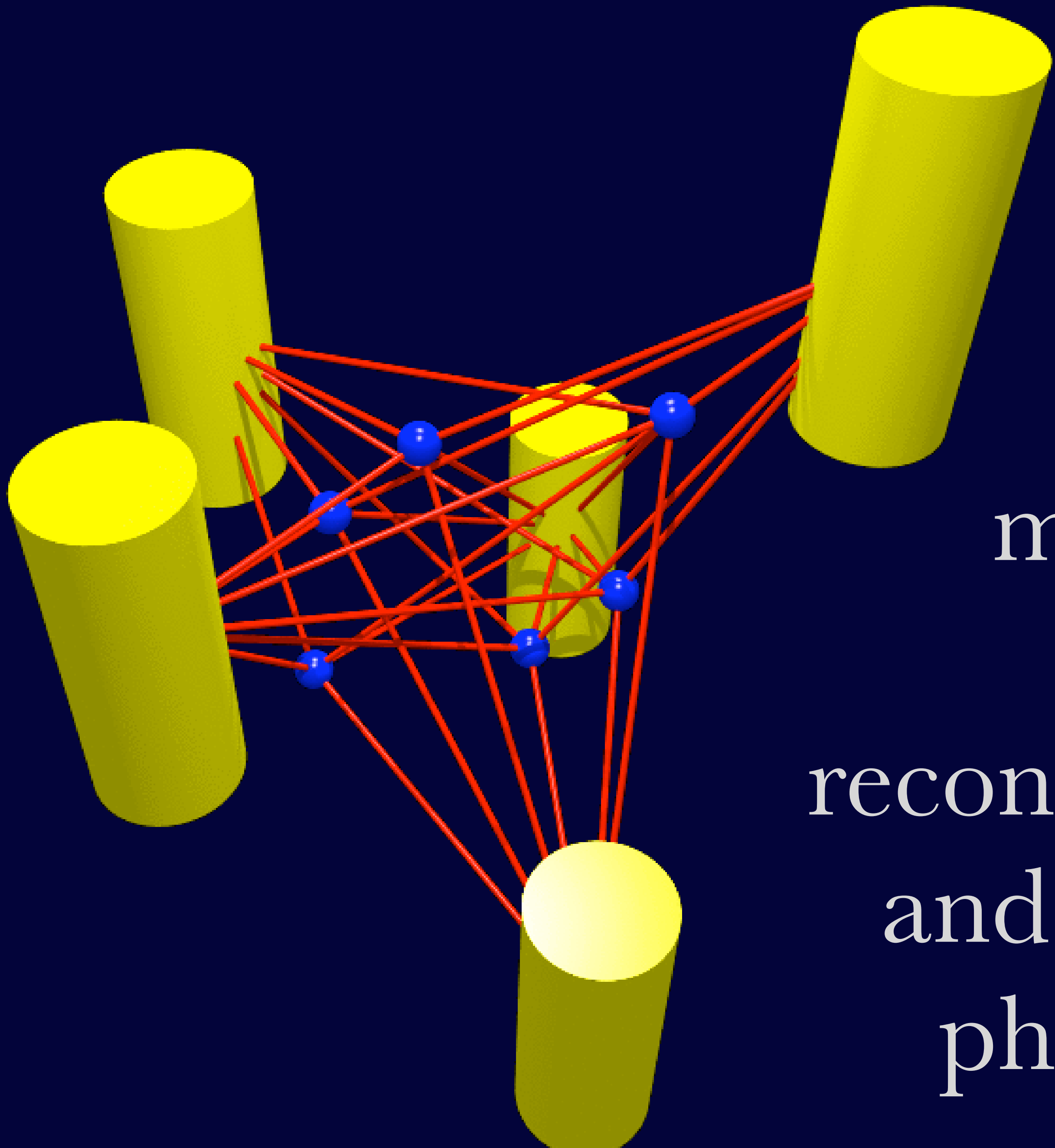

Affine camera:
d=2,m=3,n=5

Given m photos of n points. When is this realizable in 3D?

This questions seems have been unexplored so far. Jose Ramirez and I have obtained results this summer.

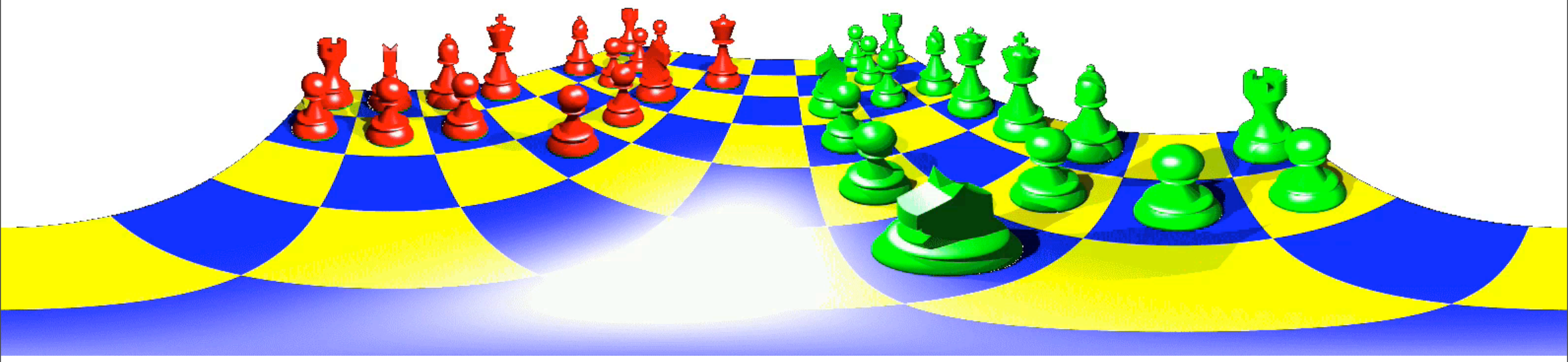
Omnidirectional cameras

Structure from motion



m cameras
n points
reconstruct camera and points from photographs!

From movie



Reconstruction

Uses linear algebra.

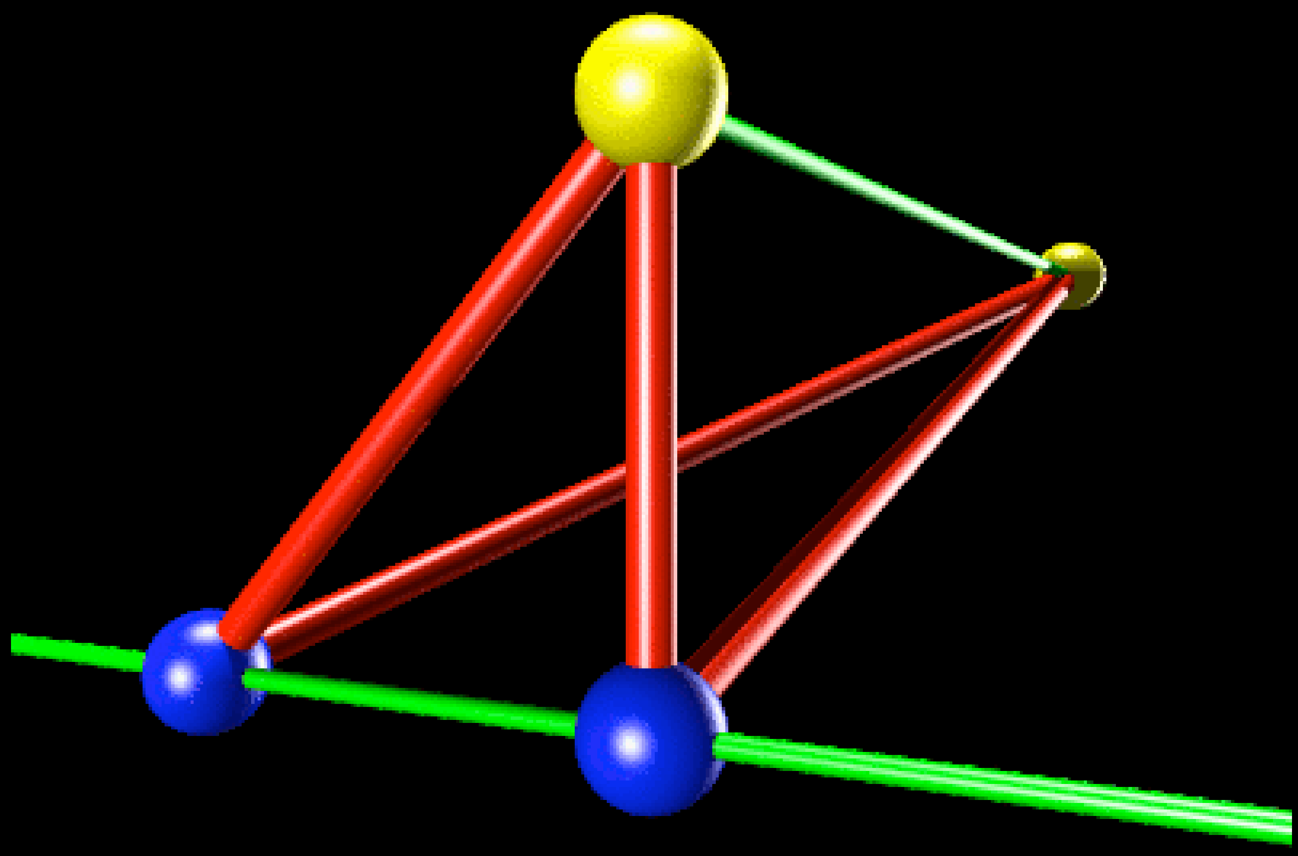
Uses elementary geometry like Desargues theorem

Details would go too far but here are some elementary considerations

What about 2 cameras and 2 points?

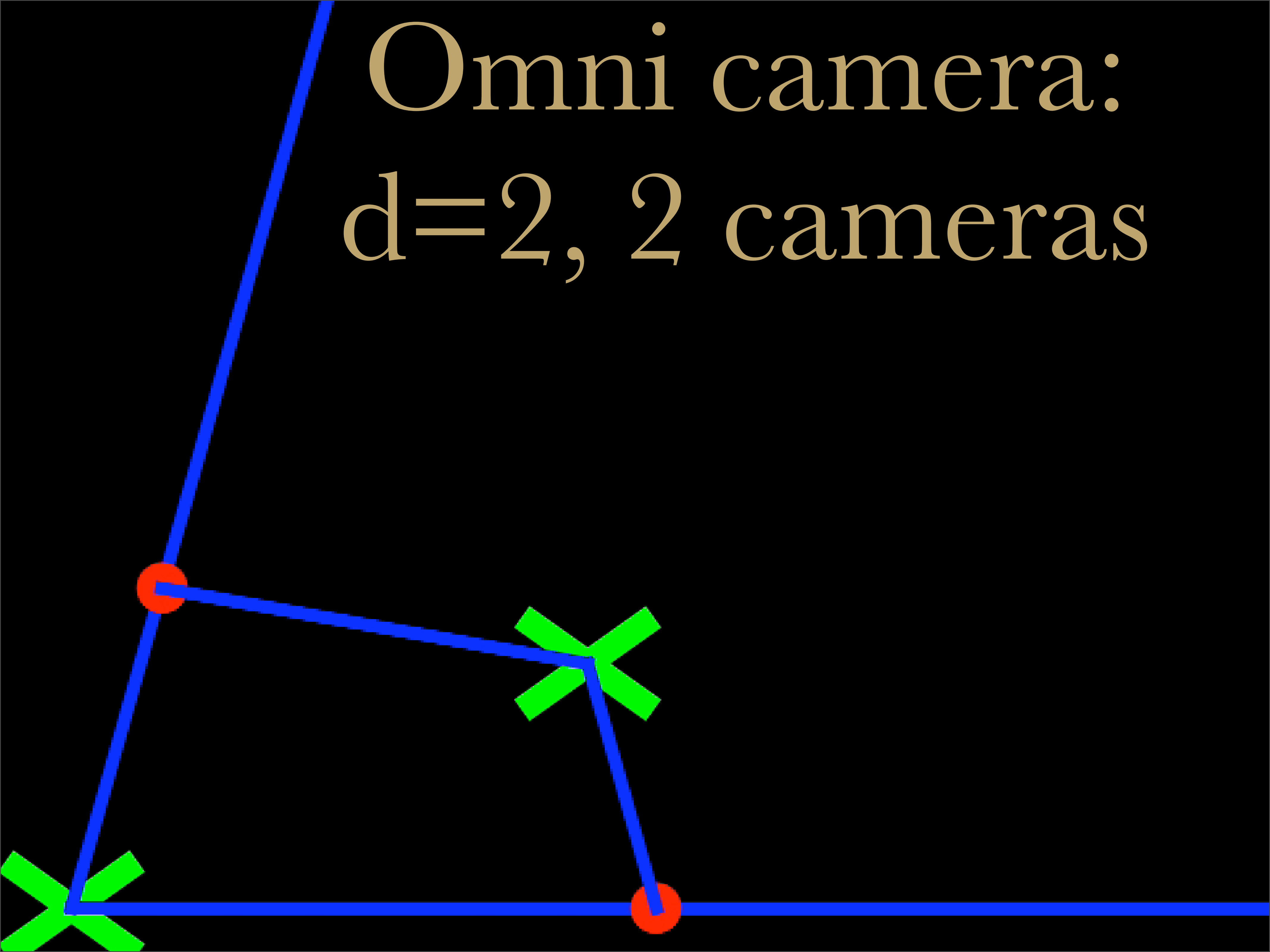
Omni cameras in space:

Omni:d=3,m=2,n=2

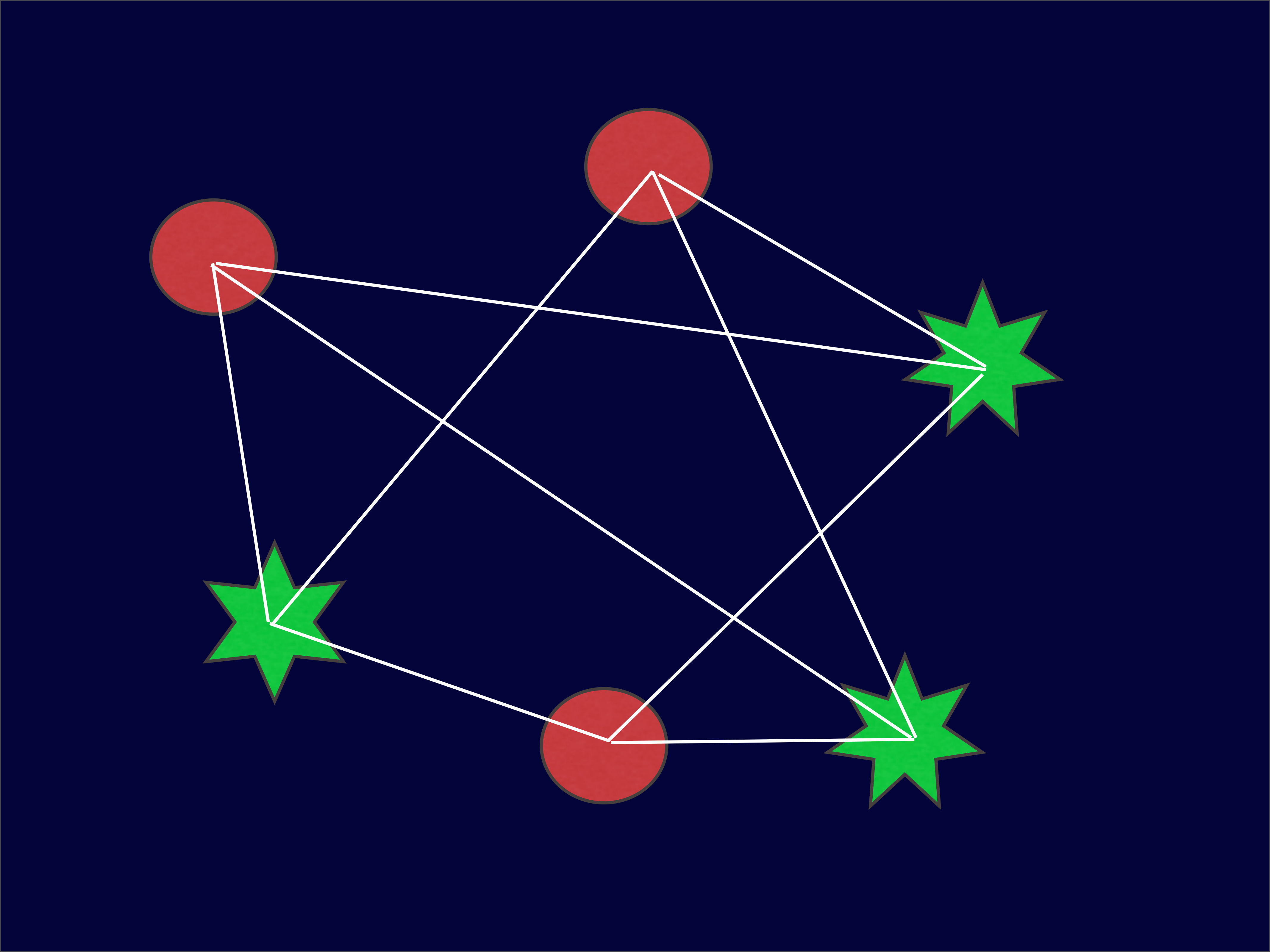


If configuration is flat, things can be deformed

Omni camera:
d=2, 2 cameras

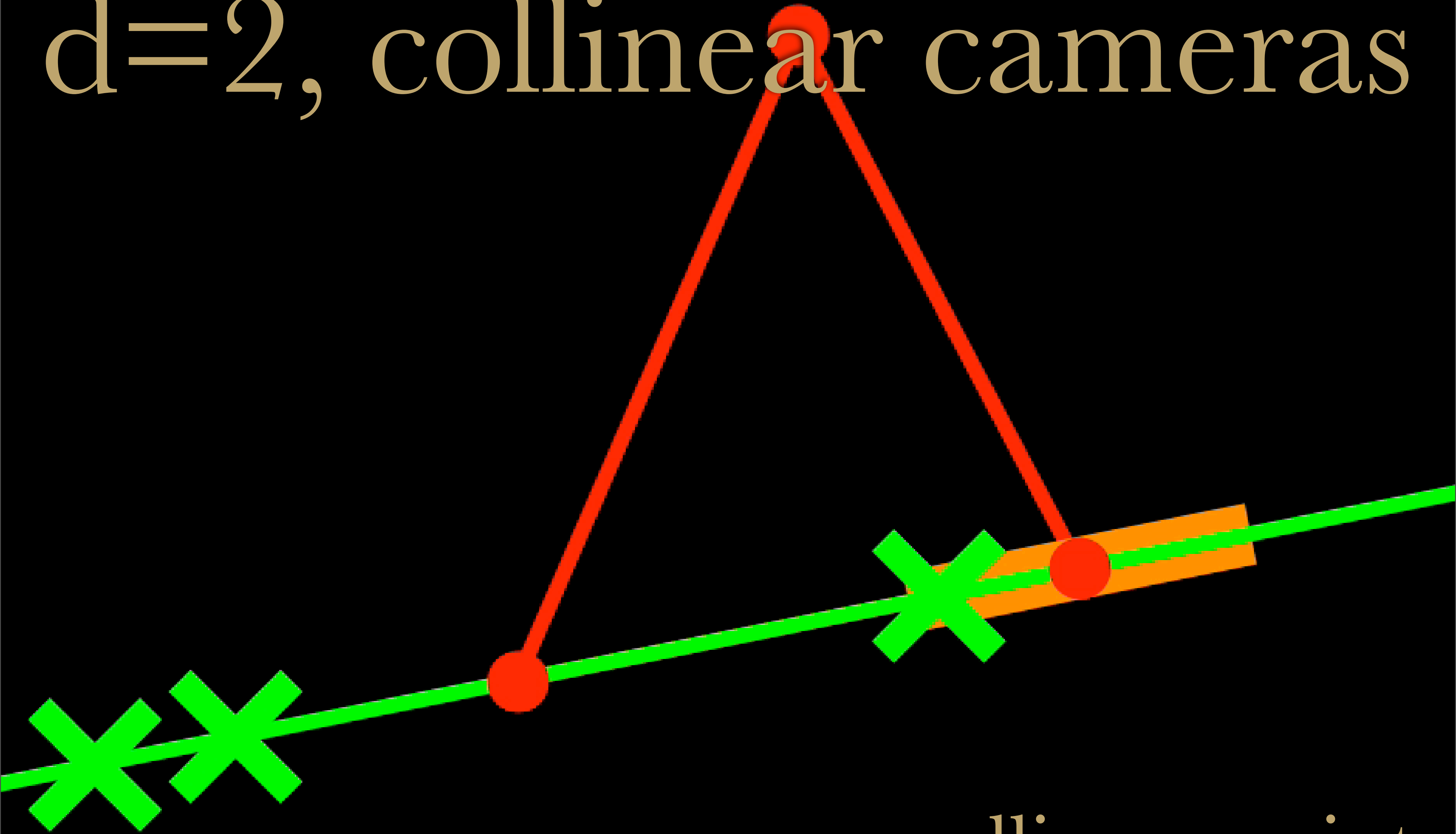


Are three cameras enough?



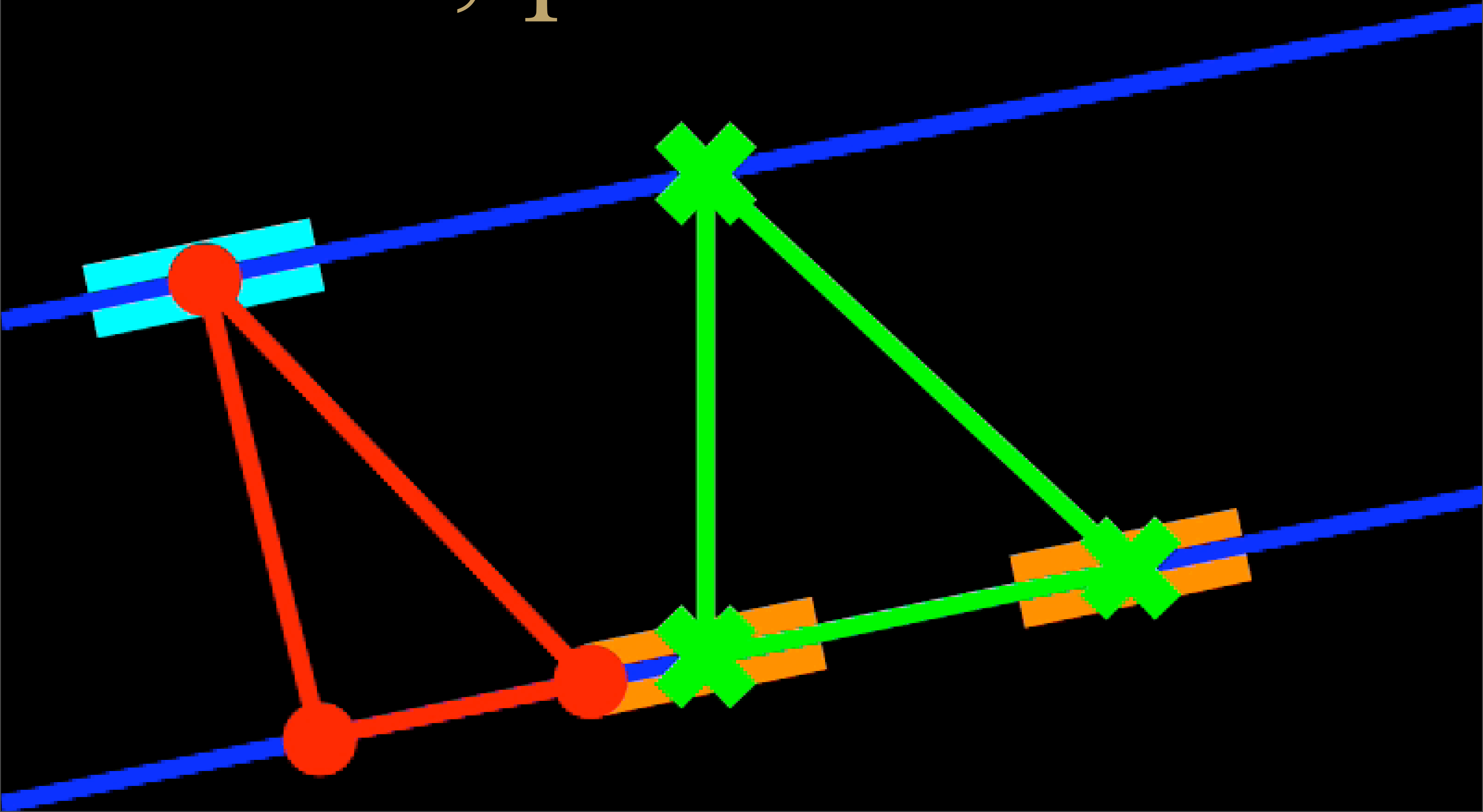
Examples:

Omni camera: d=2, collinear cameras

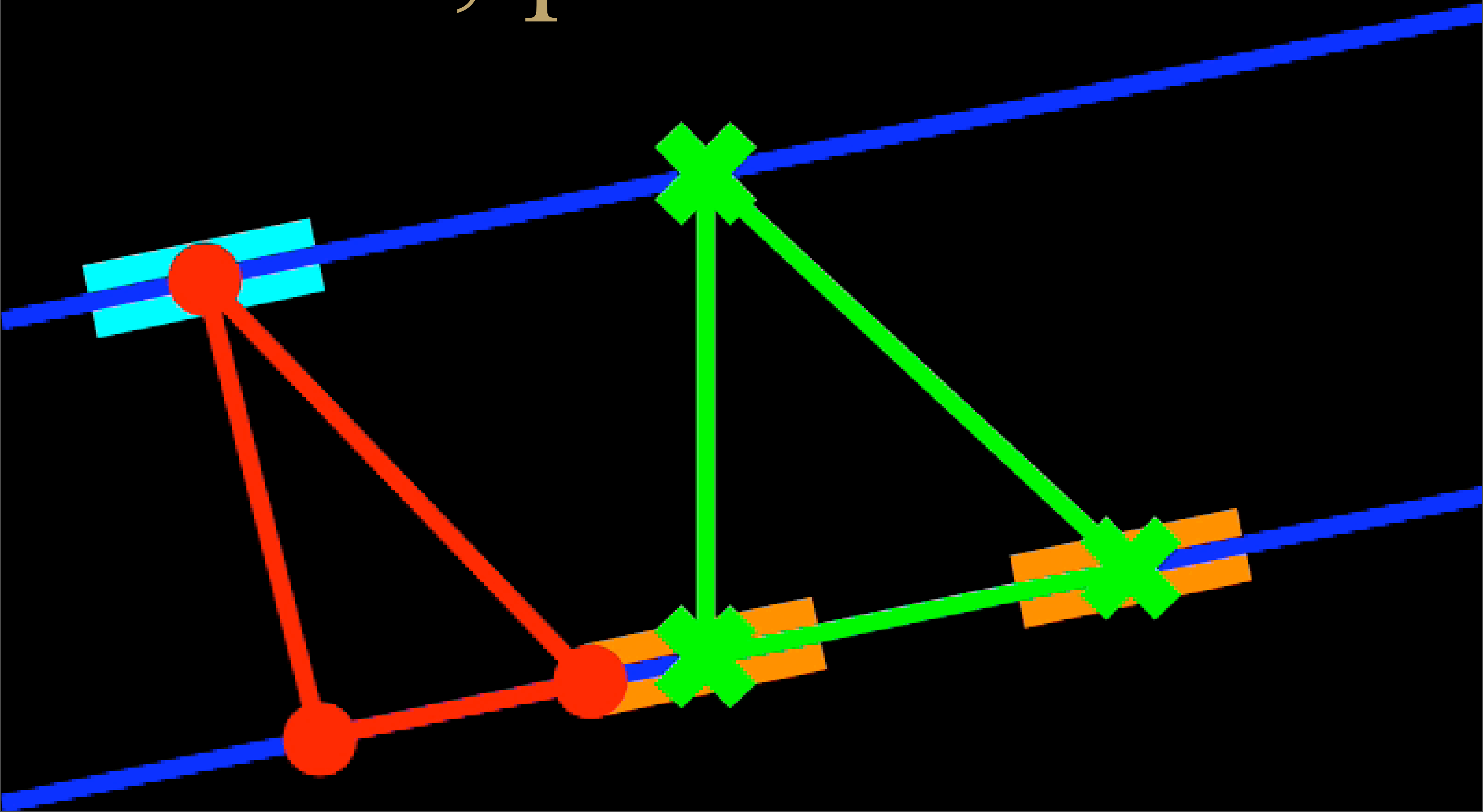


or collinear points

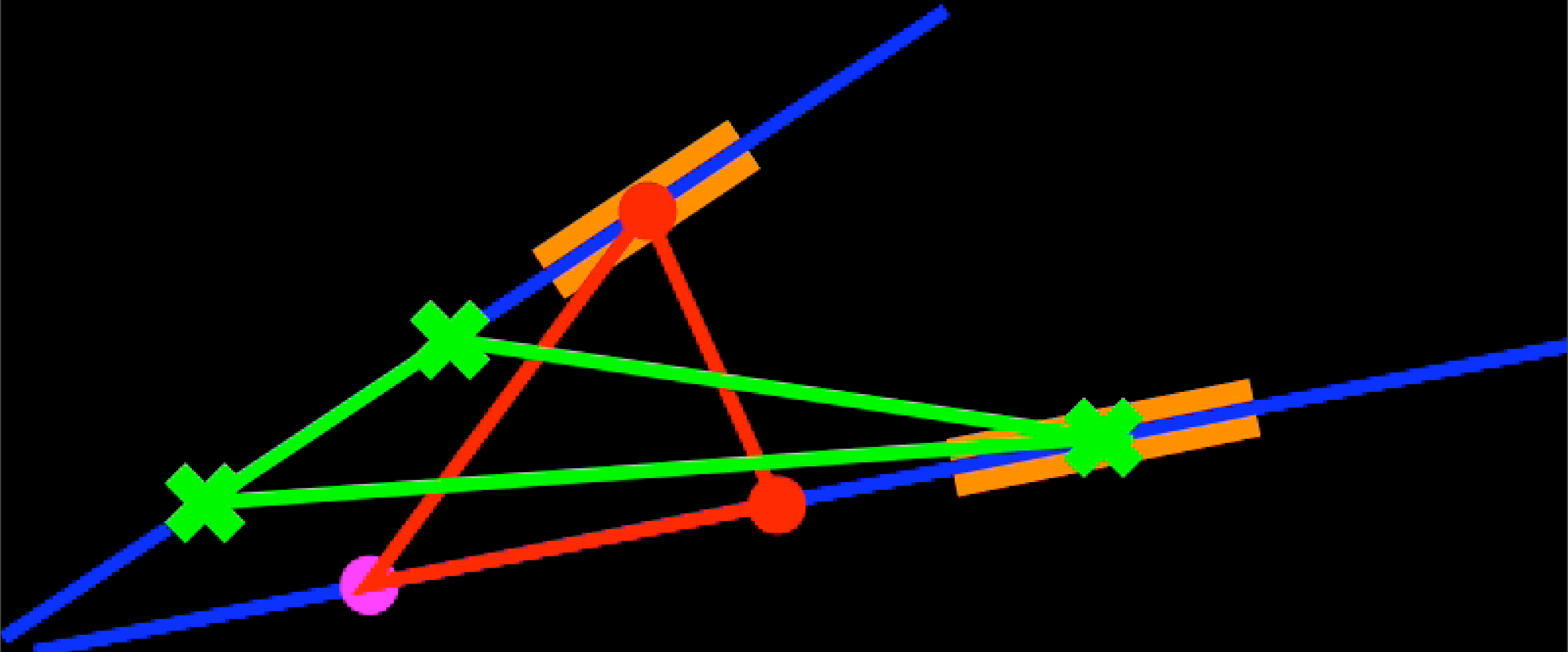
Omni camera: d=2, parallel lines



Omni camera: d=2, parallel lines



Omni camera:
d=2,m=3,n=3

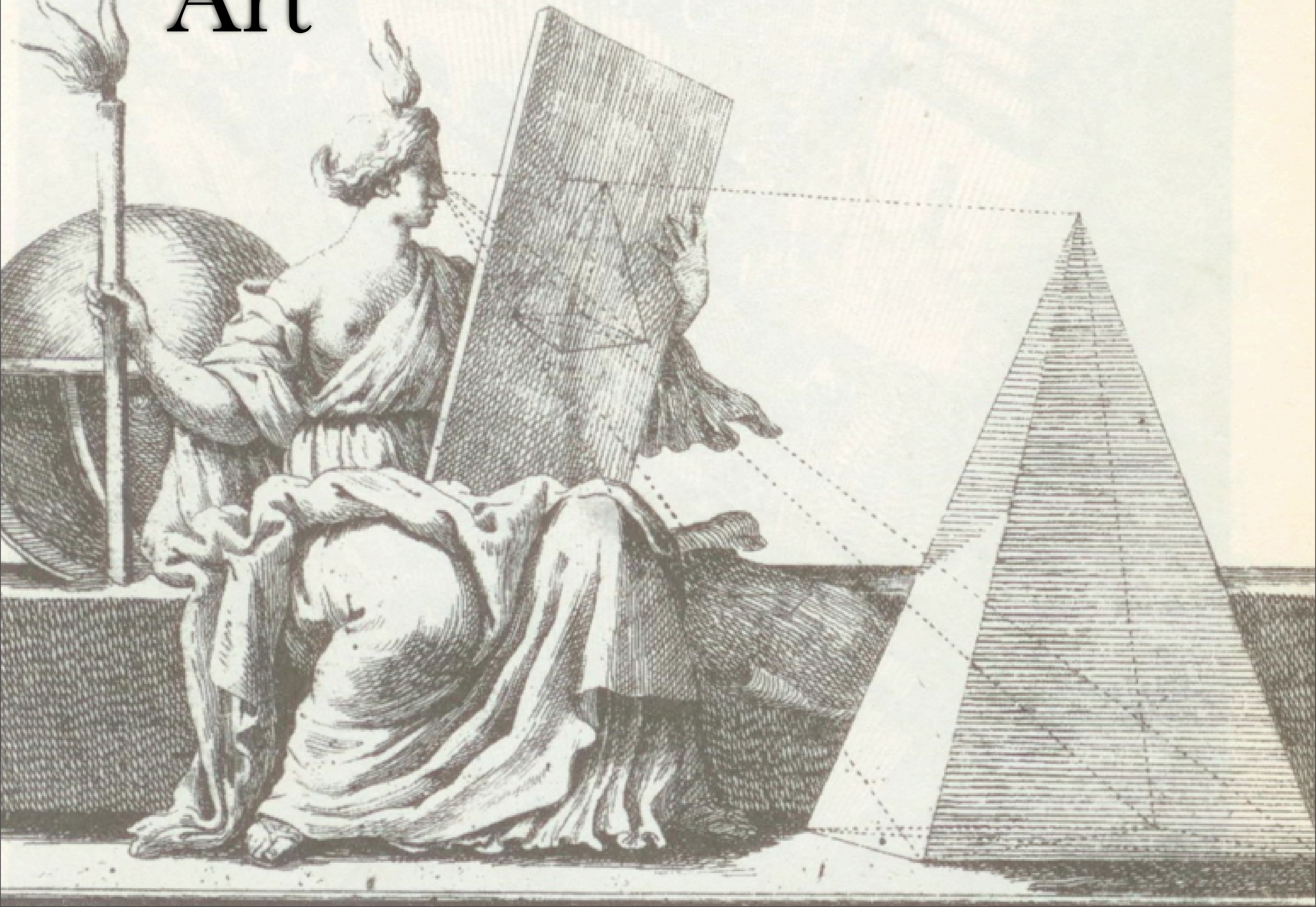


Some history pointers for the mathematics of photography


Albrecht Duerer, 1471-1528
Ant
Art

Art

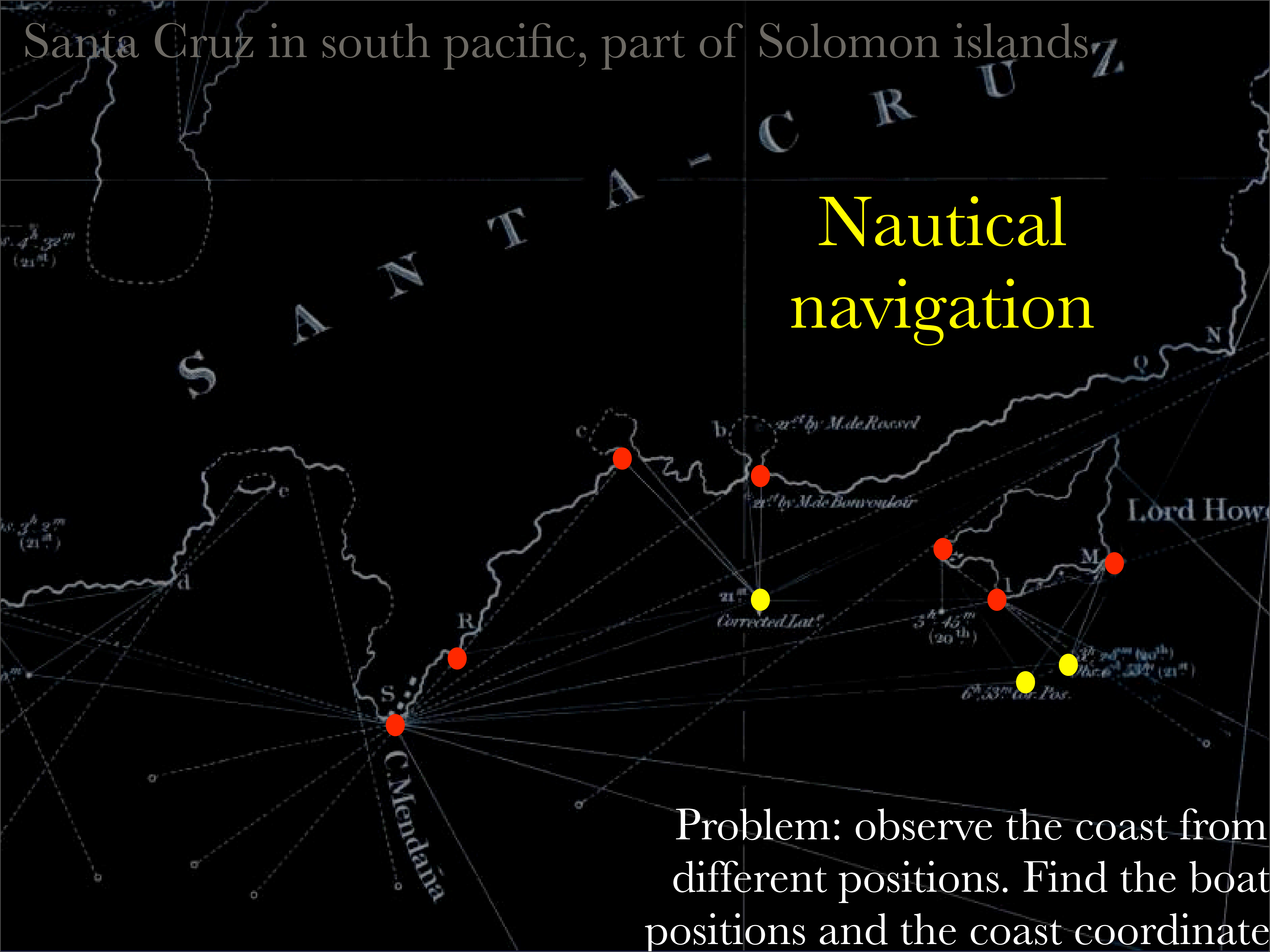
Edme-Sebastien Jeaurat, Traite de perspective a l'usage des artistes,
Paris: Jombert, 1750



Omnidirectional problems appeared in nautical navigation

Santa Cruz in south pacific, part of Solomon islands

Nautical navigation



Problem: observe the coast from different positions. Find the boat positions and the coast coordinate

Chasles, 1855

marquée dans tous les cours industriels où il est appelé à rendre de grands services au professeur et aux auditeurs.

QUESTION.

296. On donne dans le même plan deux systèmes de sept points chacun et qui se correspondent. Faire passer par chacun de ces systèmes un faisceau de sept rayons, de telle sorte que les deux faisceaux soient homographiques. Démontrer qu'il n'y a que trois solutions. (CHASLES.)

EXERCICES SUR DE GRANDS NOMBRES.

$$2^{m} = 63382\,533001\,14114\,70074\,83516\,02688 = a,$$

$$\log 2 = 0,30102\,99956\,63981\,19521\,37388\,94724\,49,$$

$a \log 2$ donne pour caractéristique

$$19080\,04273\,45073\,52812\,21794\,13680 = b,$$

ainsi, $b + 1$ est le nombre de chiffres de 2^{a}.

Ce calcul a été fait par Clausberg et se trouve dans son ouvrage *Démonstrativer Rechenkunst*, Arithmétique démonstrative, IIIe partie, § 1474; Leipsig, 1782; in-8°. On y donne les logarithmes de Briggs de 1 à 100 avec 32 décimales. Les Tables de Callet renferment de tels logarithmes de 1 à 1097 avec 61 décimales; mais il faut prendre les dix décimales à gauche dans la Table des 20 décimales.

THÉORIE DE LA DIVISION ARITHMÉTIQUE DES NOMBRES ENTIERS;

PAR M. L.-E. FAUCHEUX.

Traitée avec la simplicité convenable, la division ne présentera pas aux élèves plus de difficultés que les autres règles arithmétiques.
(*Nouveau Programme de l'École Polytechnique.*)

Lemme. Soit à multiplier deux nombres, par exemple 7436 par 48. On pourra toujours obtenir le produit de la manière suivante. On multipliera d'abord 48 par 6, ce qui donnera 288; on écrira 8 et on retiendra 28. On multipliera ensuite 48 par 3, ce qui donnera 144, à quoi on ajoutera 28 du produit partiel précédent, ce qui donnera 172; on écrira 2 et on retiendra 17. On multipliera 48 par 4, et au produit 192 on ajoutera 17, ce qui donnera 209; on écrira 9 et on retiendra 20. Enfin on multipliera 48 par 7, et au produit 336 on ajoutera 20; on aura 356, qu'on écrira entièrement, et le produit sera 356928.

On pourra toujours considérer le produit d'une multiplication effectuée comme ayant été obtenu par ce procédé-là.

Dans cette multiplication, il n'y a qu'un produit partiel entièrement écrit, c'est le dernier, c'est-à-dire le produit du multiplicateur par le chiffre des plus hautes unités du multiplicande. Des autres produits partiels on n'a écrit que le chiffre des unités; et ce chiffre exprime des unités de même ordre que celles du chiffre du multiplicande qui a donné le produit partiel.

Les retenues de chaque produit partiel n'égaleront jamais le multiplicateur 48; en effet, le premier produit

Photogrammetry

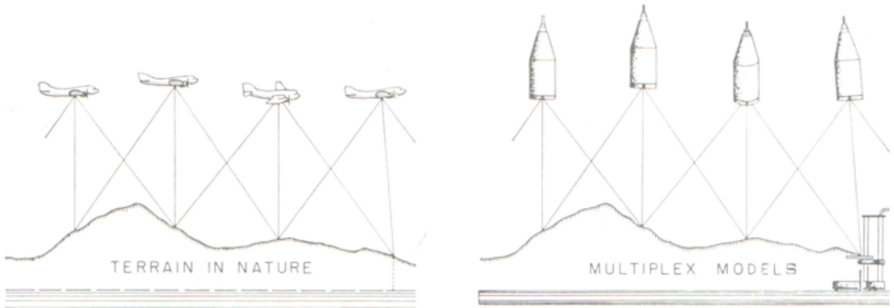
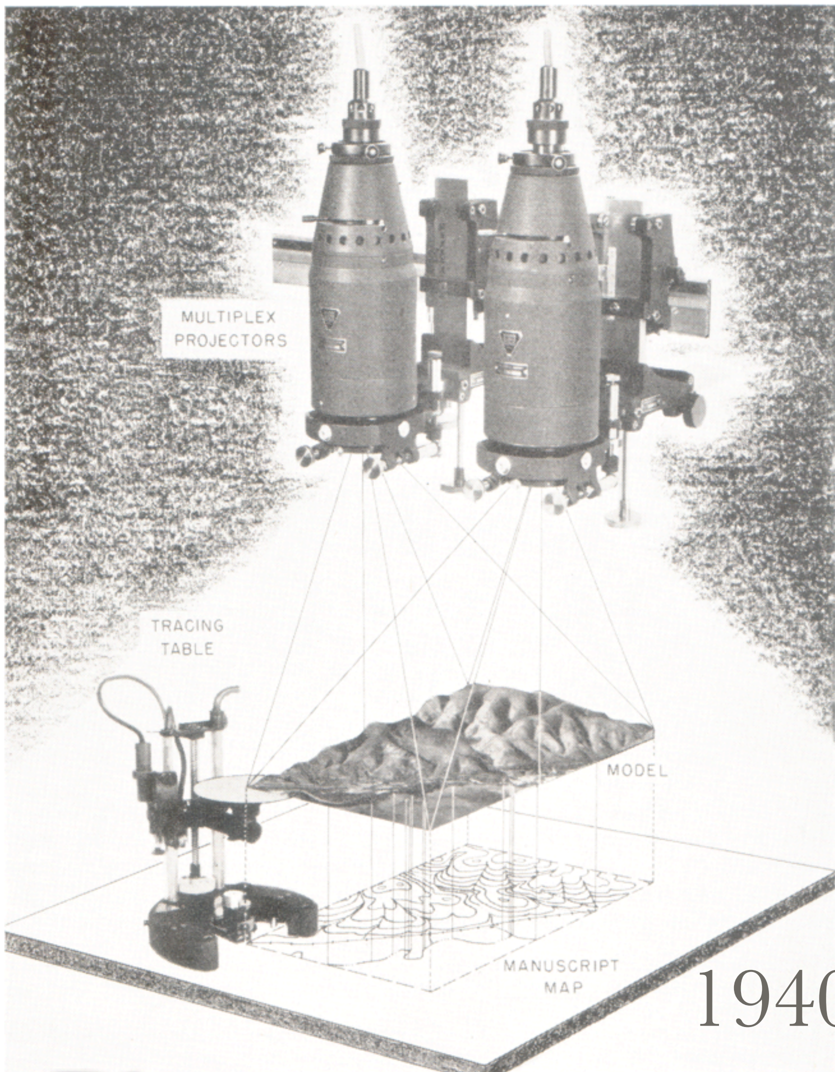


FIG. 1. Geometric parallelism between aerial photography and multiplex reconstruction.



1940

FIG. 1A. Diagrammatic sketch of multiplex model.

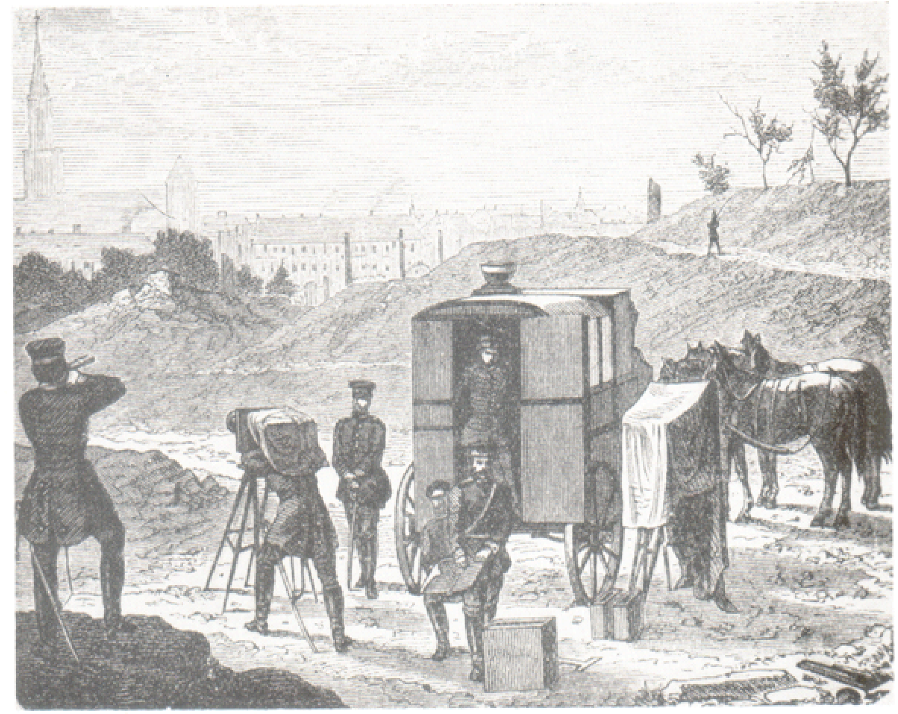
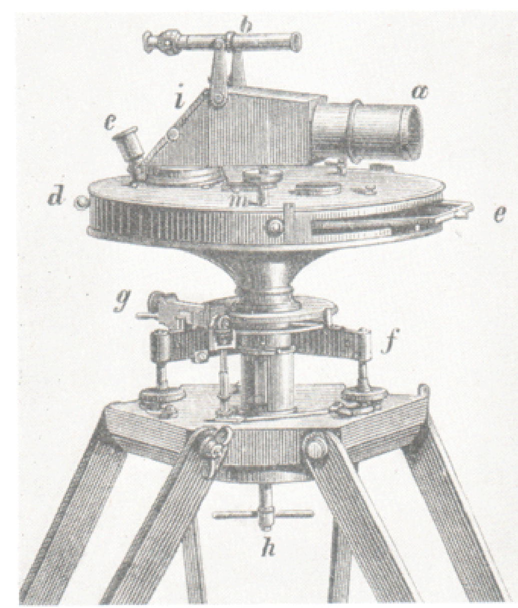


FIG. 4. Prussian mobile photographic unit, 1870. (Laussedat, vol. 2, pt. 2, p. 8.)

1870



1858

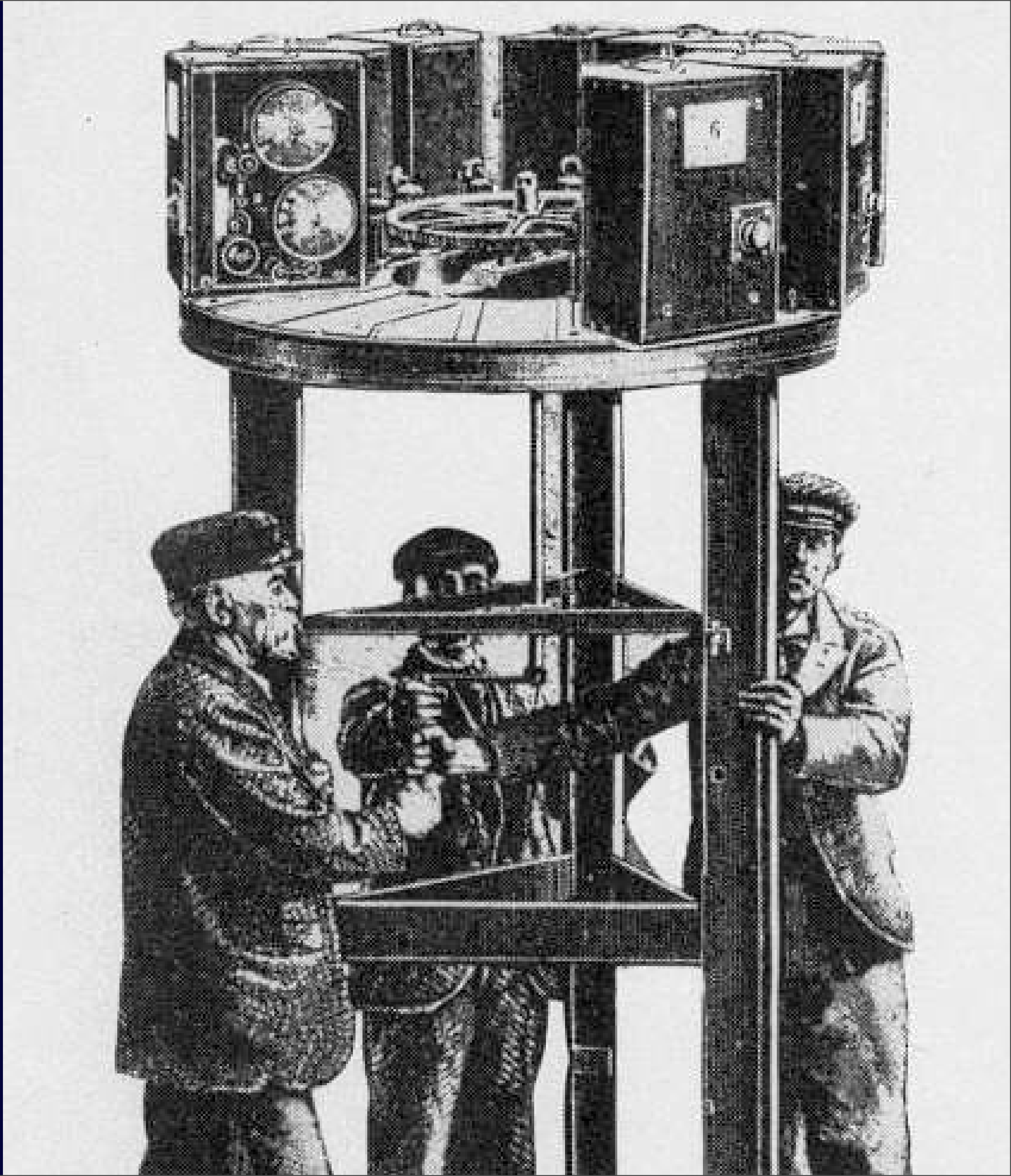
FIG. 5. Photographic plane table of Chevallier, 1858. (Laussedat, vol. 2, pt. 1, p. 30.)

Raoul
Grimoin-Sanson
(1860-1940)

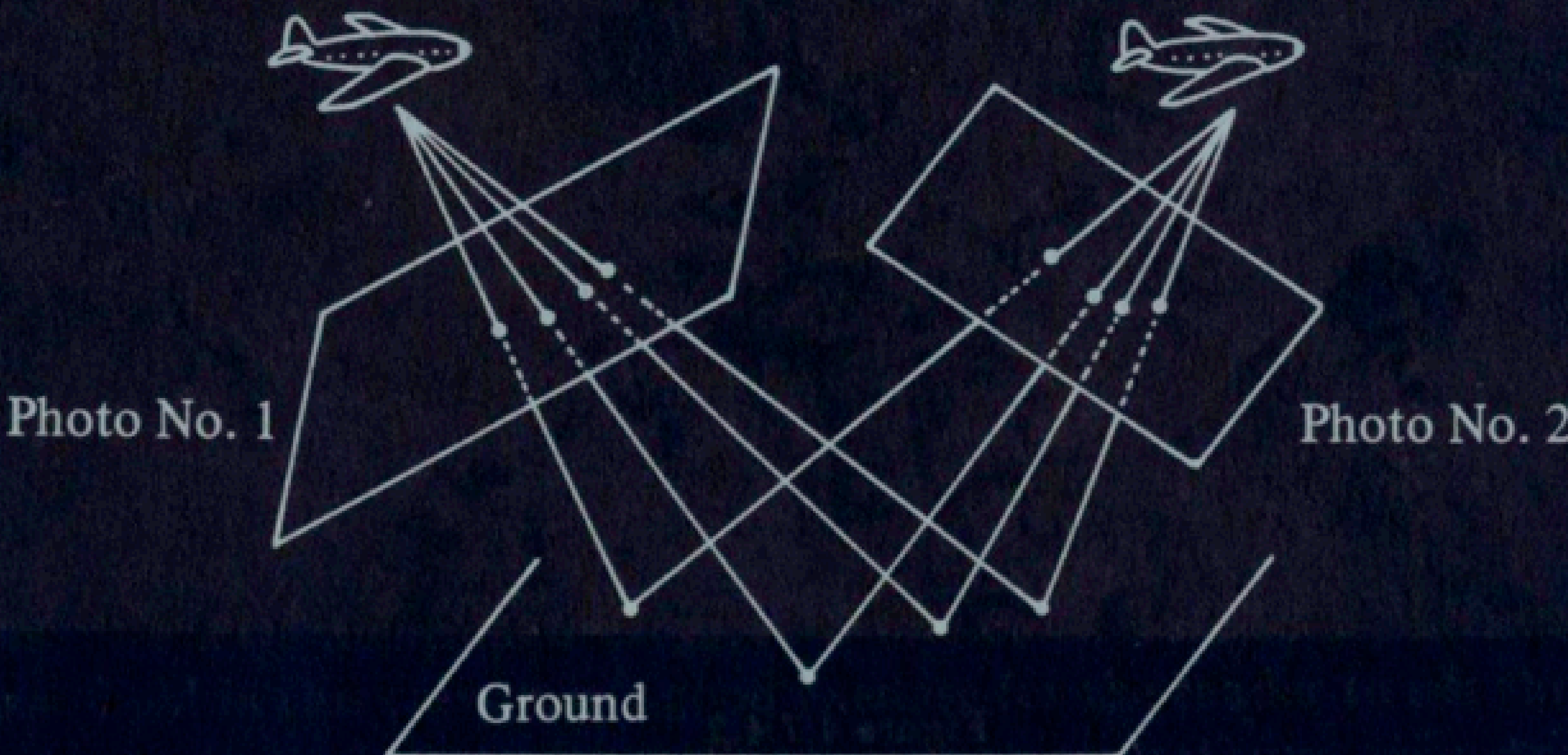
10 synchronised cameras filmed a balloon ascent from a balloon basket.

Cineorama

1900



In aerial photography, the problem arises of matching partial shots of an area which is too large to fit in a single photo. Assuming the area to be flat, it is possible to use perspectives to make overlapping parts coincide perfectly: the correspondence between matching points in two photographs is a homography, being the composition of the perspectives of the two photos (figure 4.7.3.1), and thus can be composed with another homography to give the identity. By proposition 4.5.10, it is enough to match four points in the two images to obtain a perfect correspondence. See [BUR, 36–51].



Photogrammetry

Figure 4.7.3

Berger, Geometry I

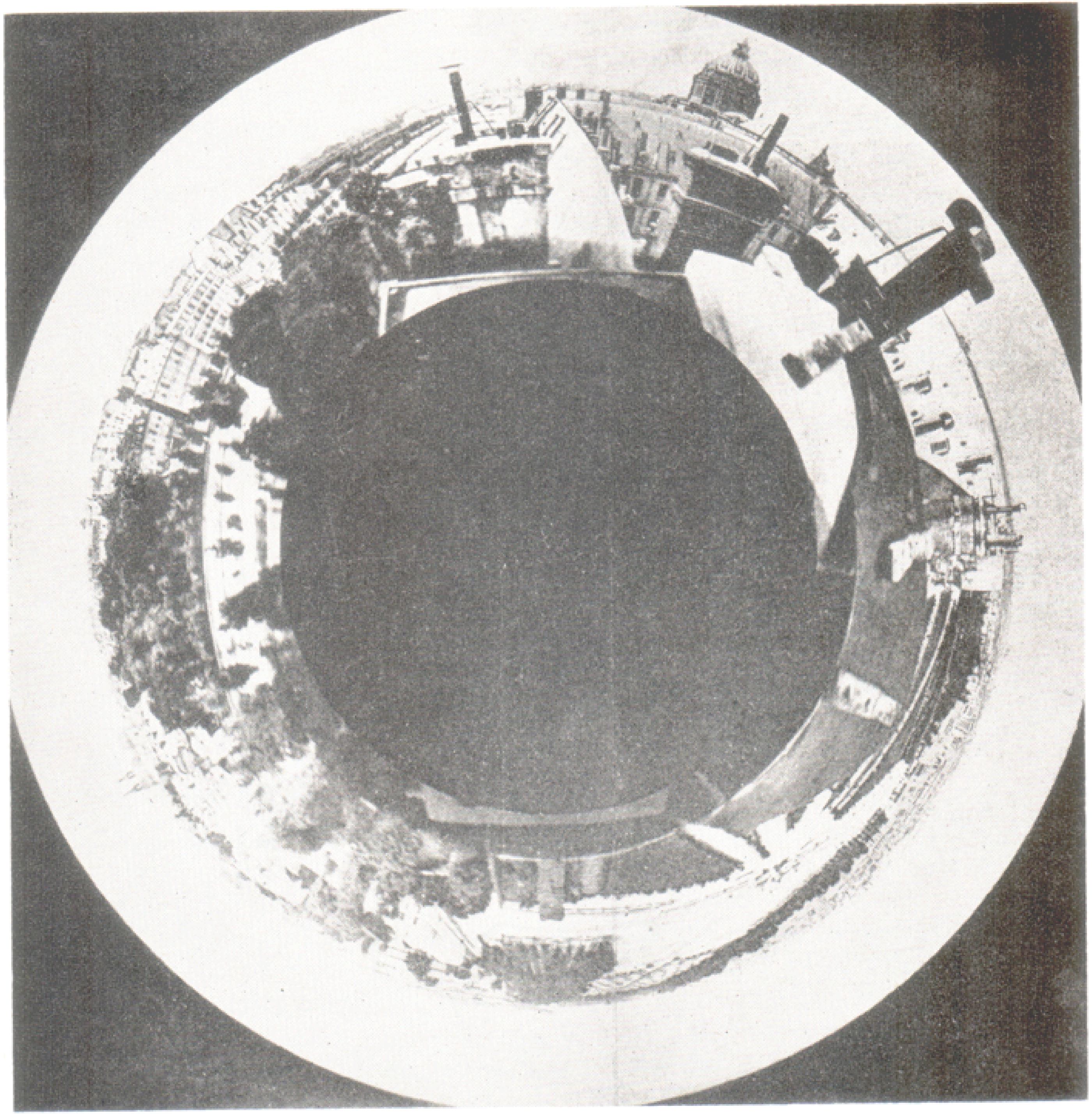


FIG. 7, B. Horizon photograph taken with Mangin's improved apparatus based on Chevallier's photographic plane table. (Laussedat, vol. 2, pt. 1, pl. v.)

MANUAL OF PHOTOGRAMMETRY

(Second Edition)

AMERICAN SOCIETY OF PHOTOGRAMMETRY

Chevaliers photographic table

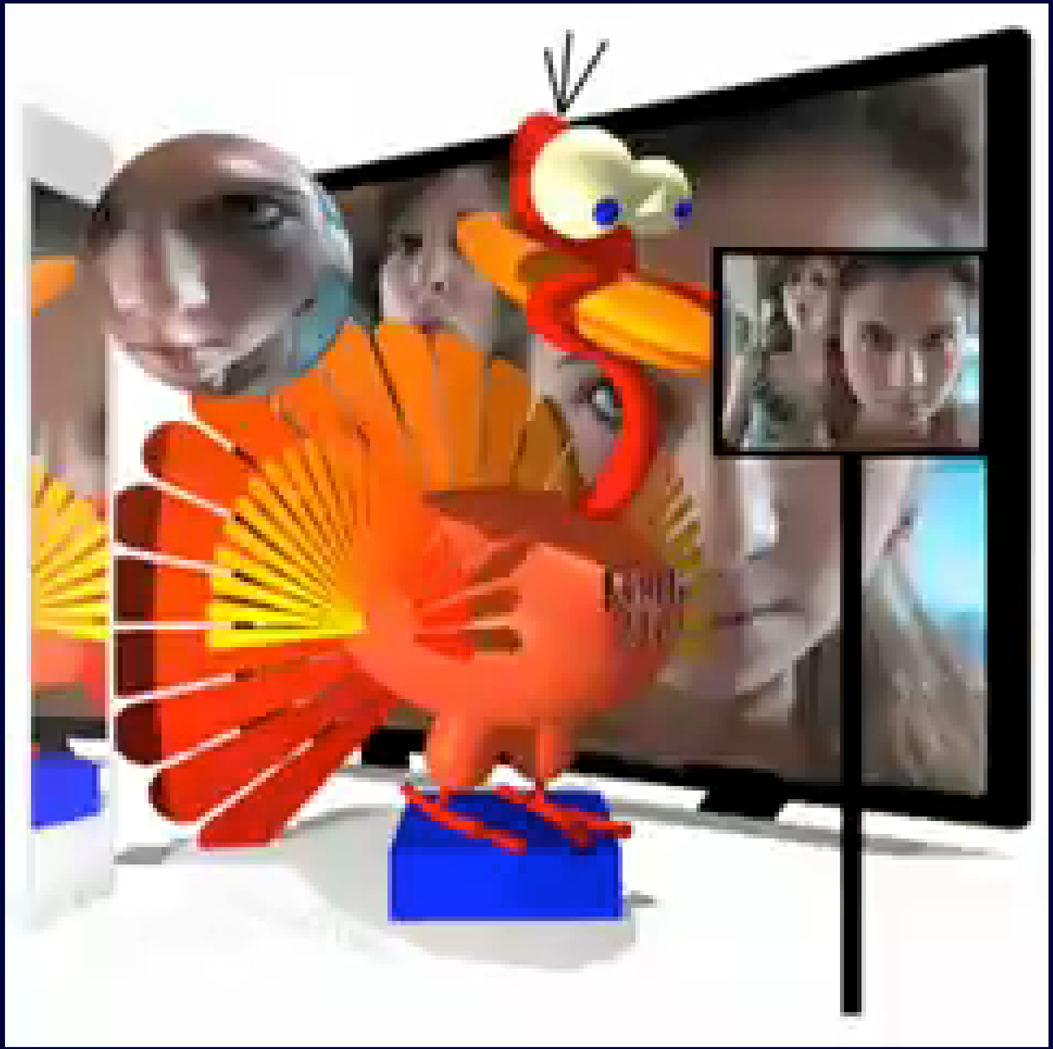
Synthetic Panoramas

Computer generated panoramas

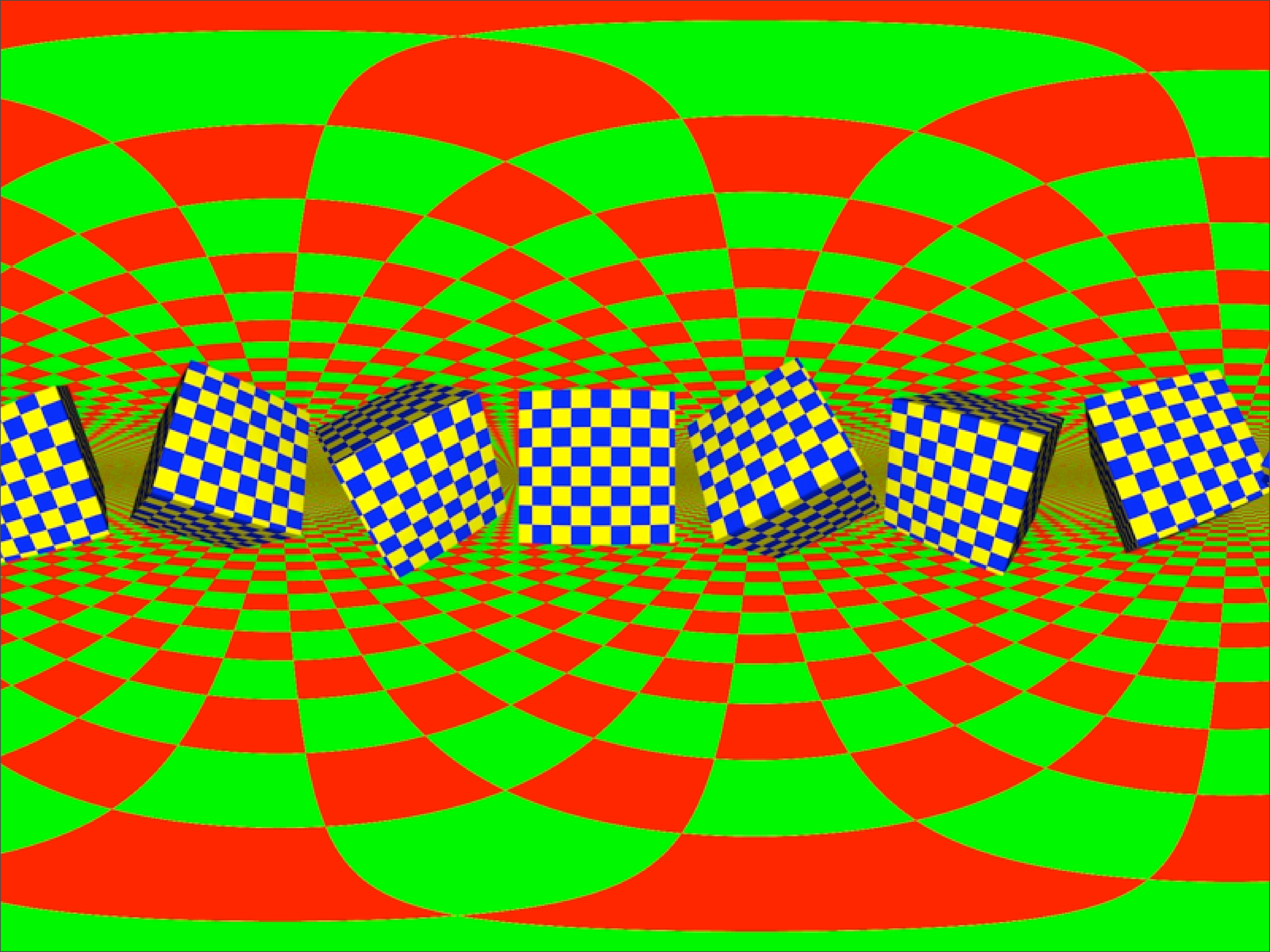
The next examples were made with the open source ray tracer Povray (Persistence of vision ray tracer). This software can produce pictures like this:

This is a picture from a ray tracing competition (not done by me) but rendered in Povray.

I use it mostly for illustration purposes or for fun

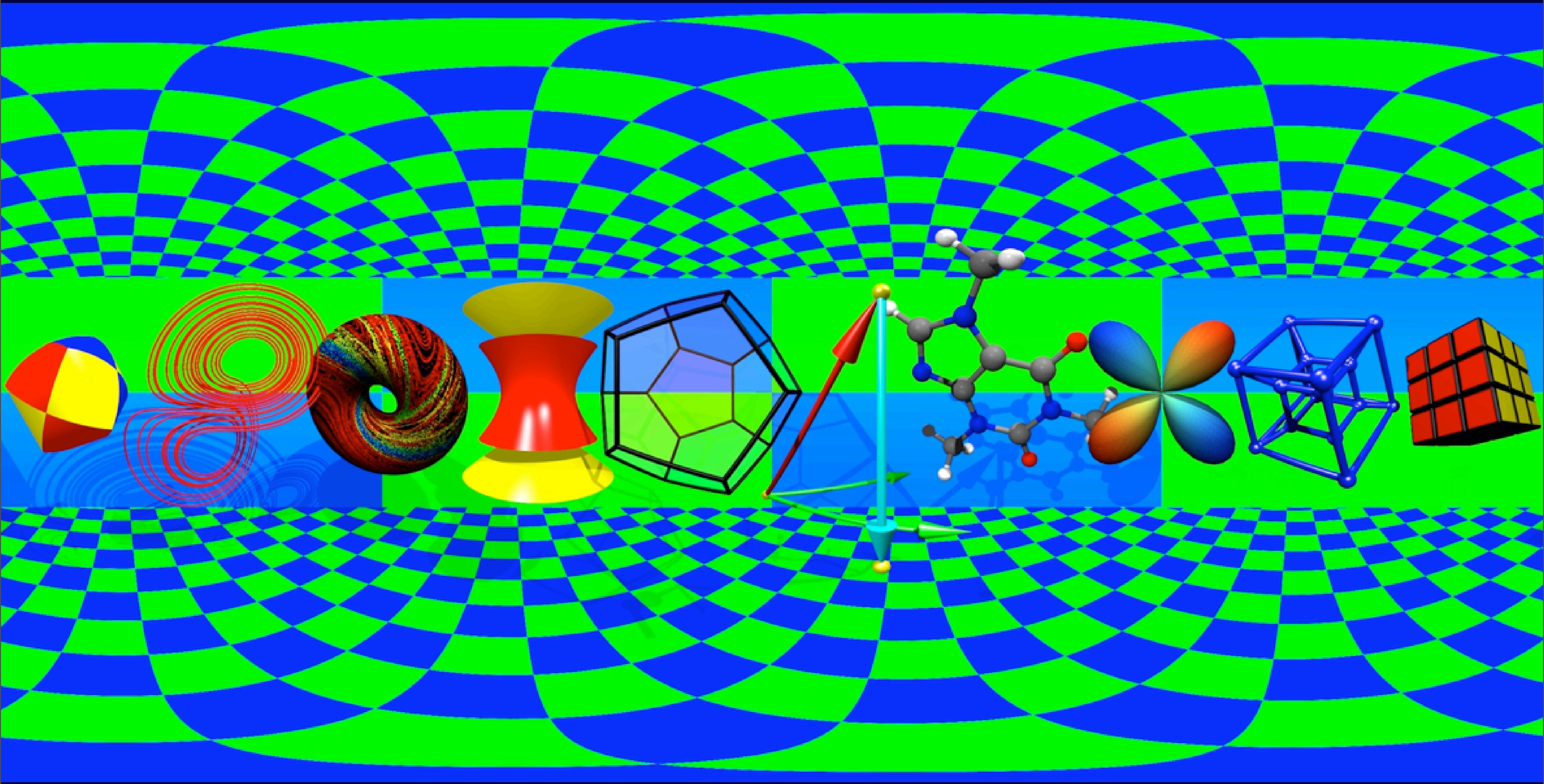


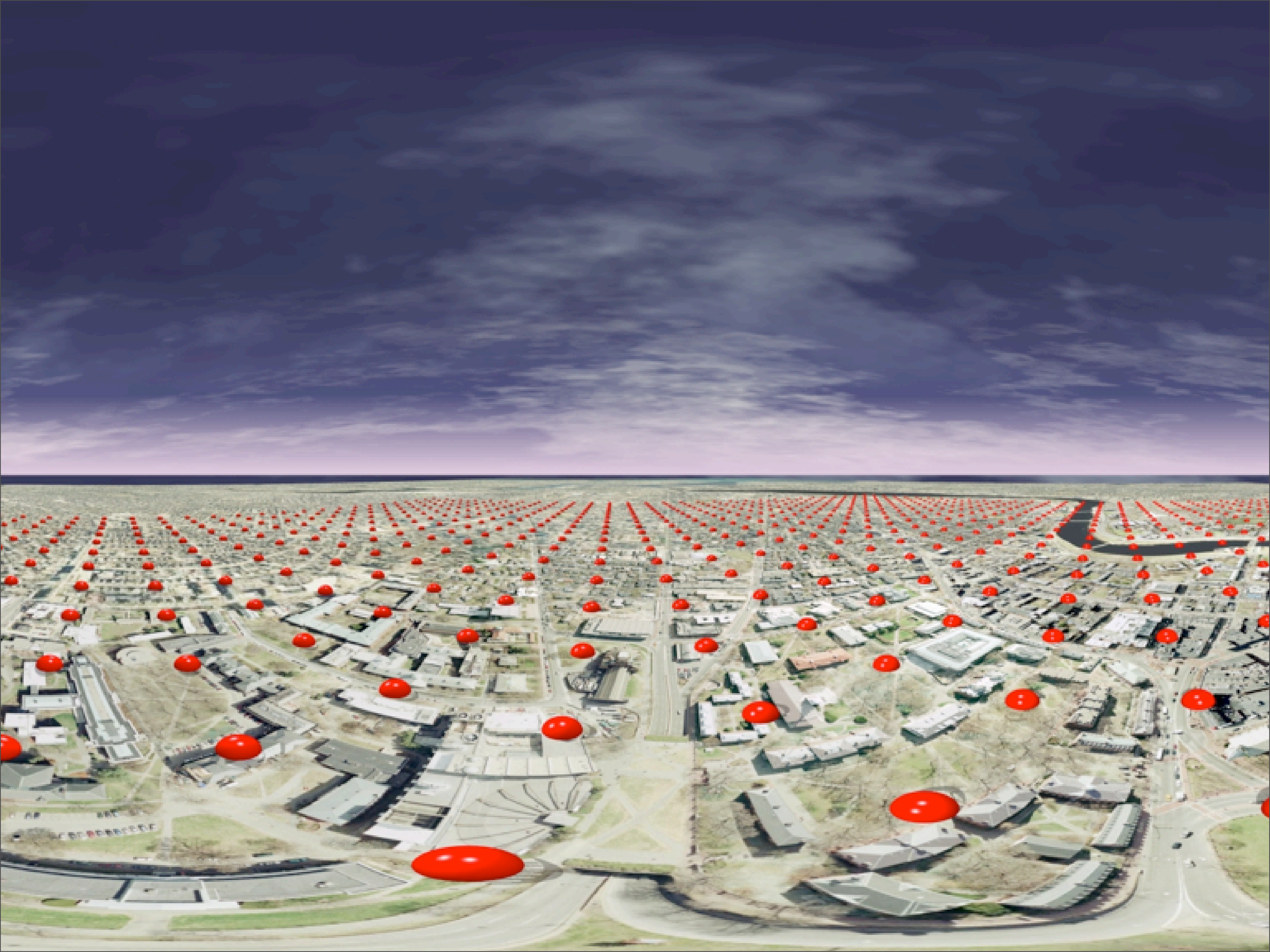
Next comes an example of a panorama:

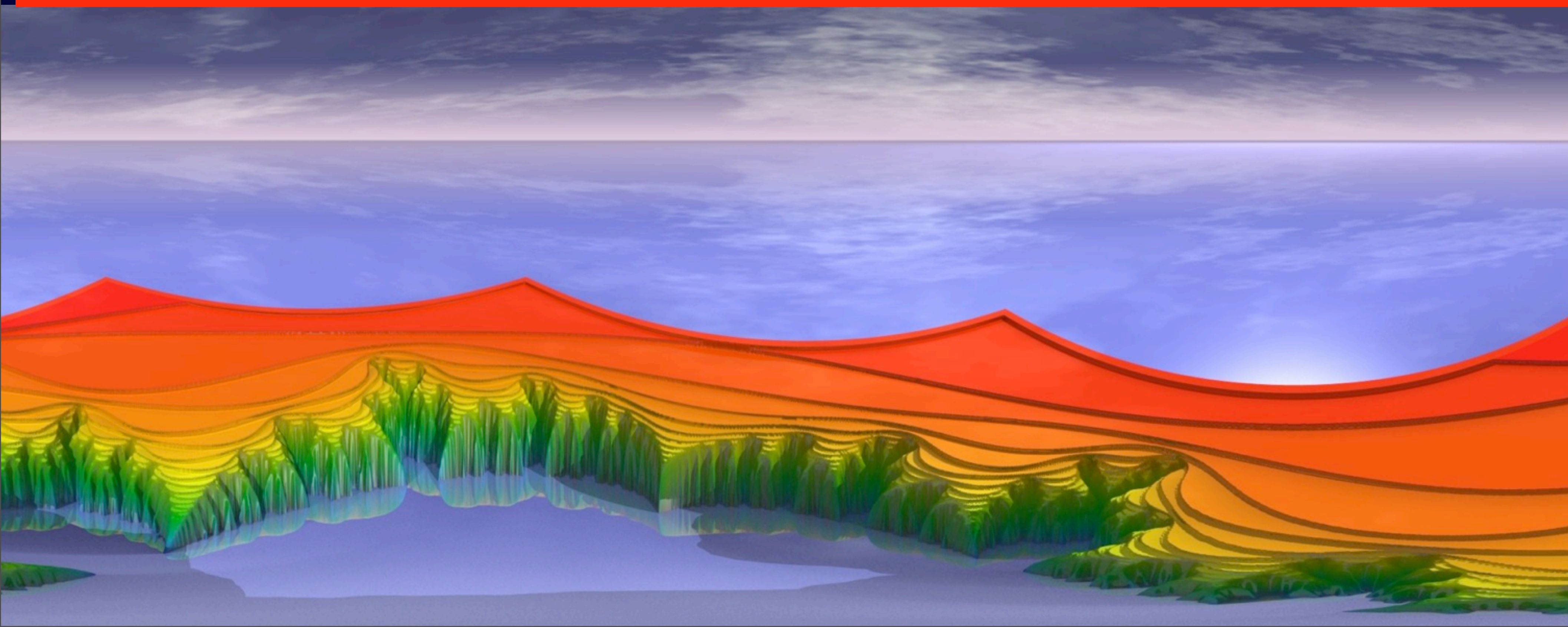


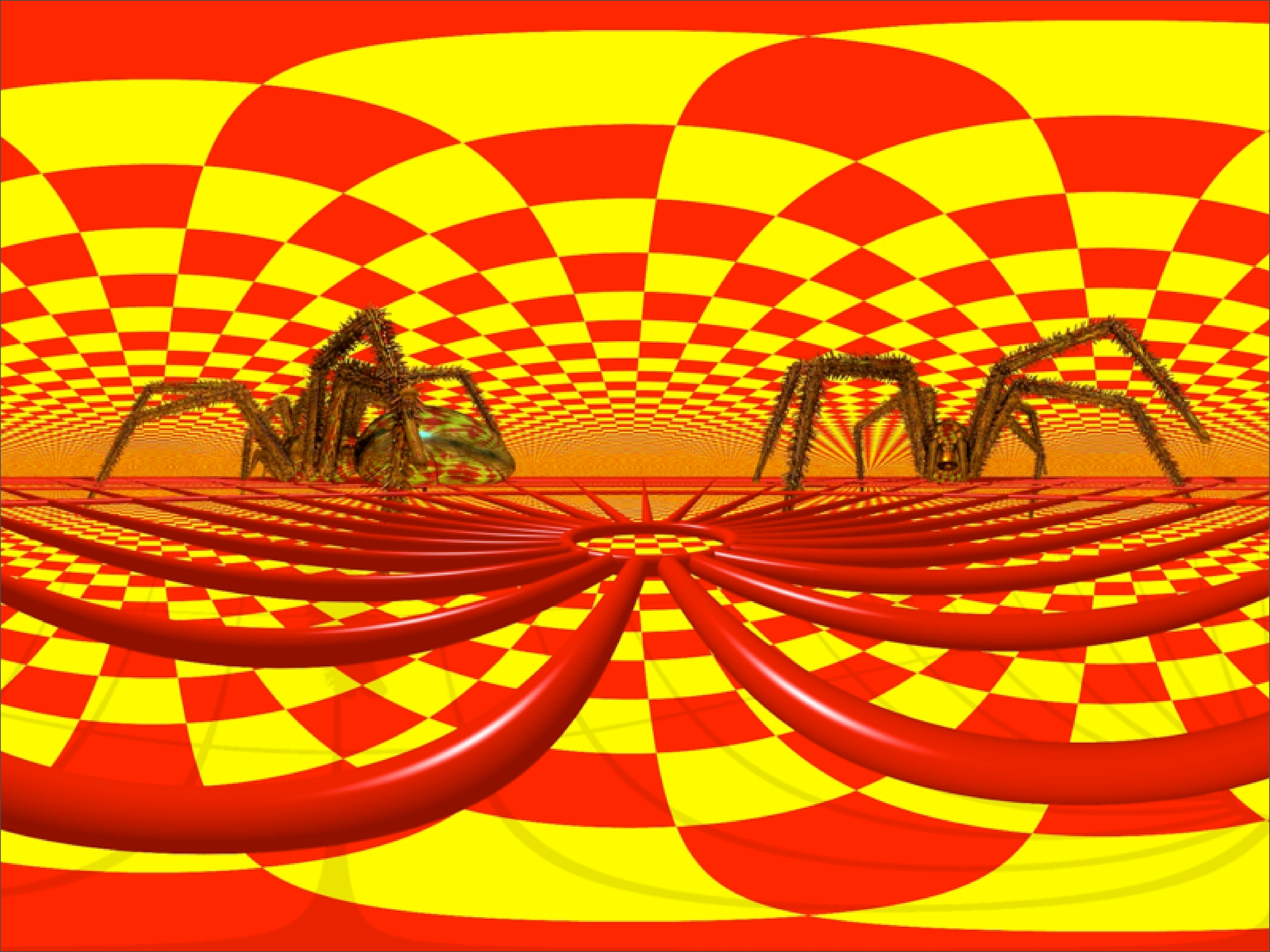
Show me the code:

```
light_source { <0,2,0> colour rgb <1,1,1> }
background { rgb<1,1,1> }

#macro r(c) pigment{ rgb c } finish { phong 1.0 ambient 0.5 diffuse 0.5 } #end

plane { <0,1,0>, -3
  pigment{checker colour rgb<1,0,0> color <0,1,0> translate <0.2,0,0.3>}
  finish{ specular 0.25 ambient 0.9 diffuse 0.8 reflection metallic 1.0} rotate <0,30,0>
}

plane { <0,1,0>, 3
  pigment{checker colour rgb<0,1,0> color <1,0,0> translate <0.2,0,0.3> }
  finish{ specular 0.25 ambient 0.9 diffuse 0.8 reflection metallic 1.0 } rotate <0,30,0>
}

#declare i=0; #declare M=10;
#while (i<M)
     #declare si=4*sin(2*pi*i/M); #declare co=4*cos(2*pi*i/M);
     #declare rr=1.2; #declare yy=0.1;
     object{  box{ <-1,-1,-1>,<1,1,1>}
         pigment{checker colour rgb<1,1,0> color <0,0,1> translate <0.2,0,0.3> scale 1/4}
         rotate 360*(i/M)*<0,1,1> translate rr*<co,yy,si> }
   #declare i=i+1;
#end
```



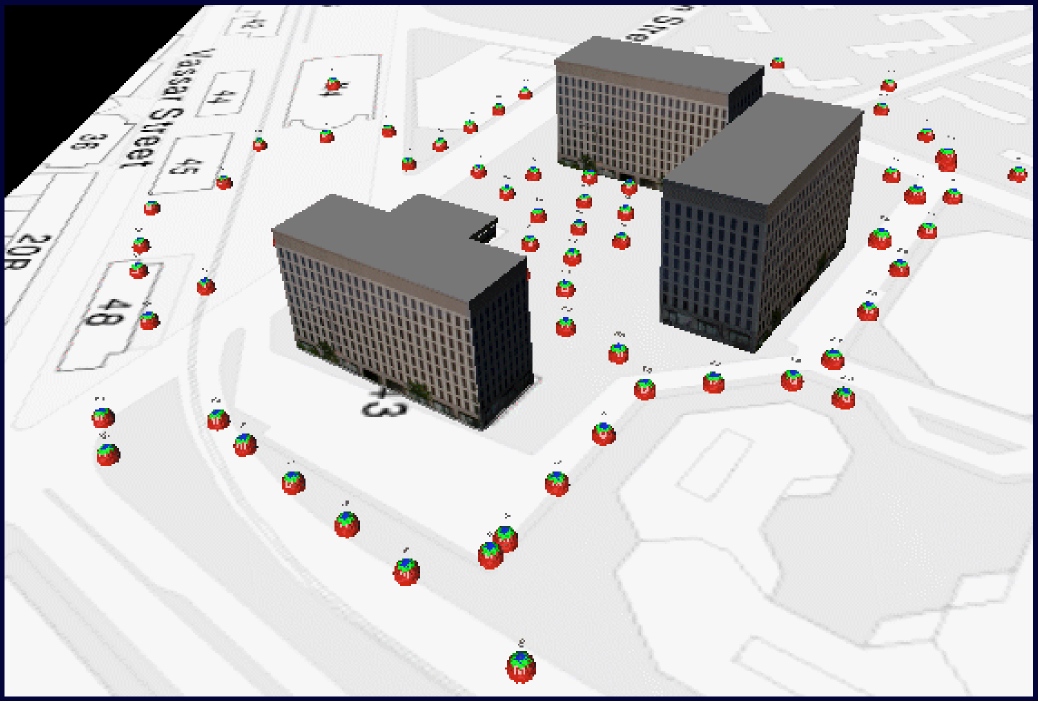




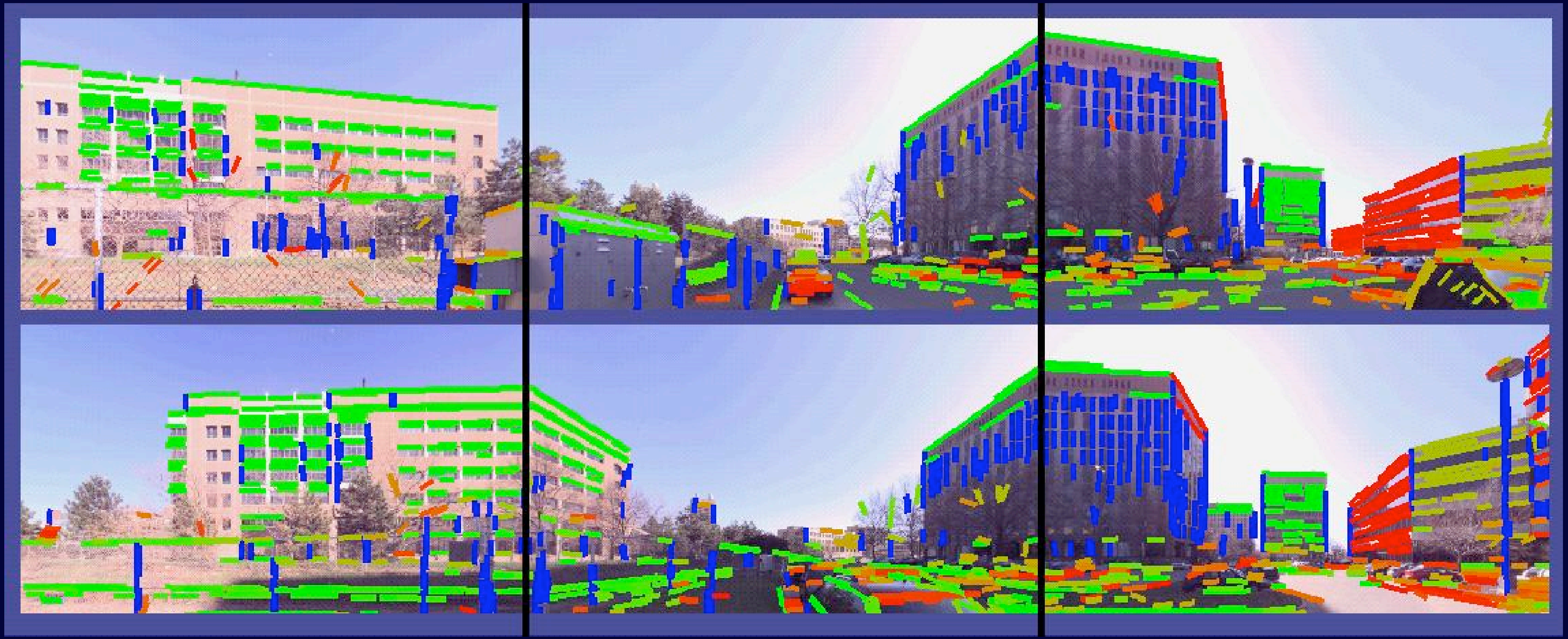


Applications

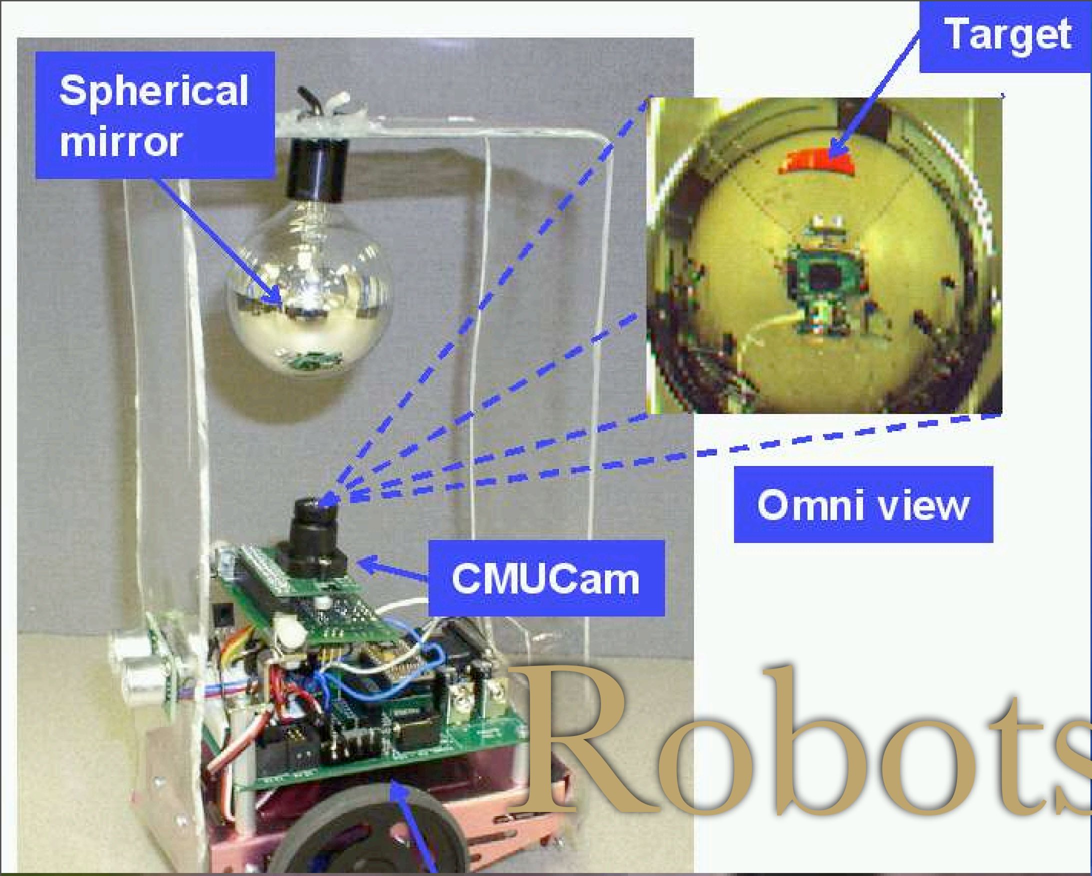
City scanning



MIT



Robots



Google Street maps

University of Fukui