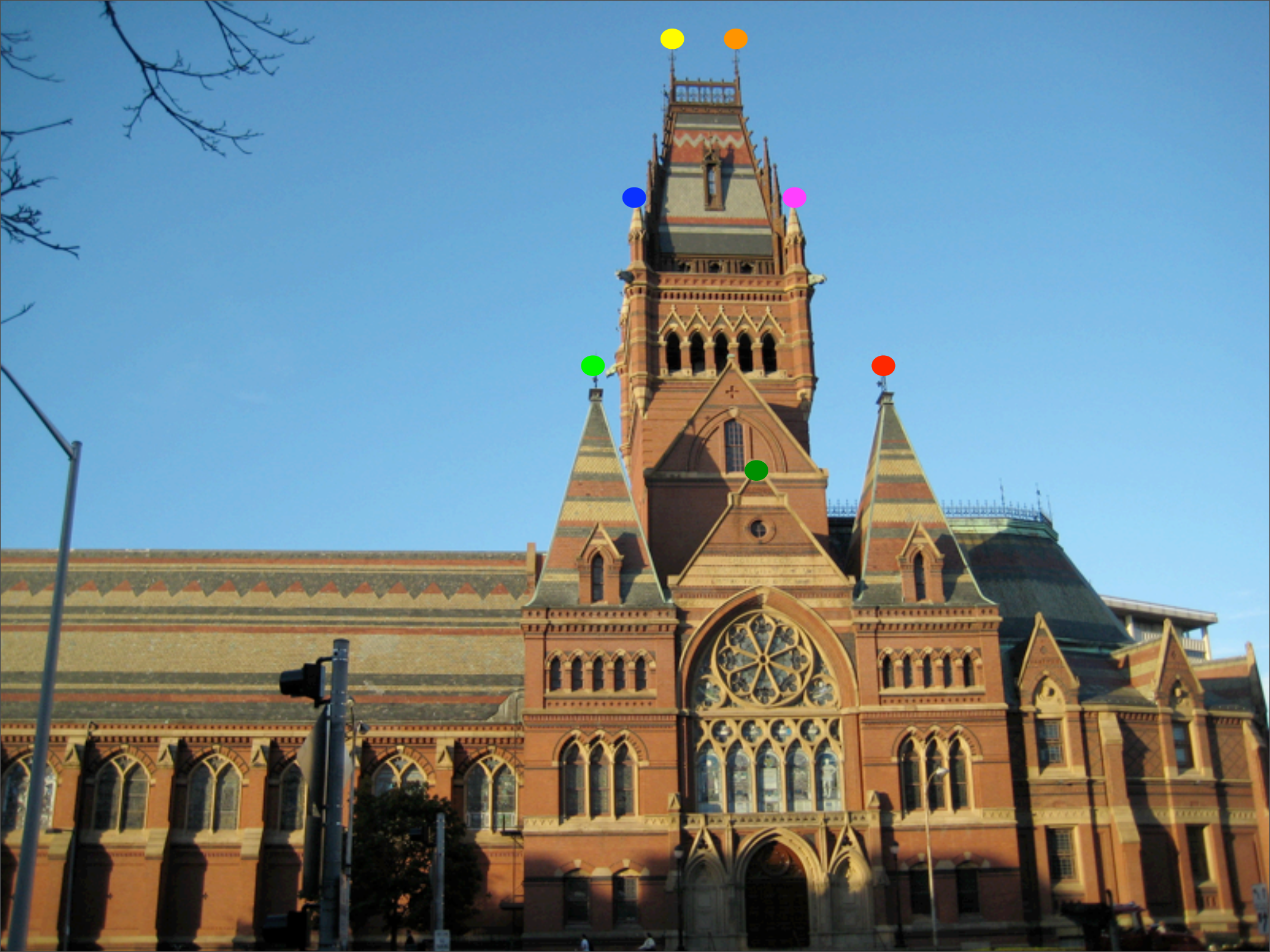
The Structure From Motion problem

Oliver Knill, October 29, 2007, Colby College

or “The

Mathematics of Panorama photography”

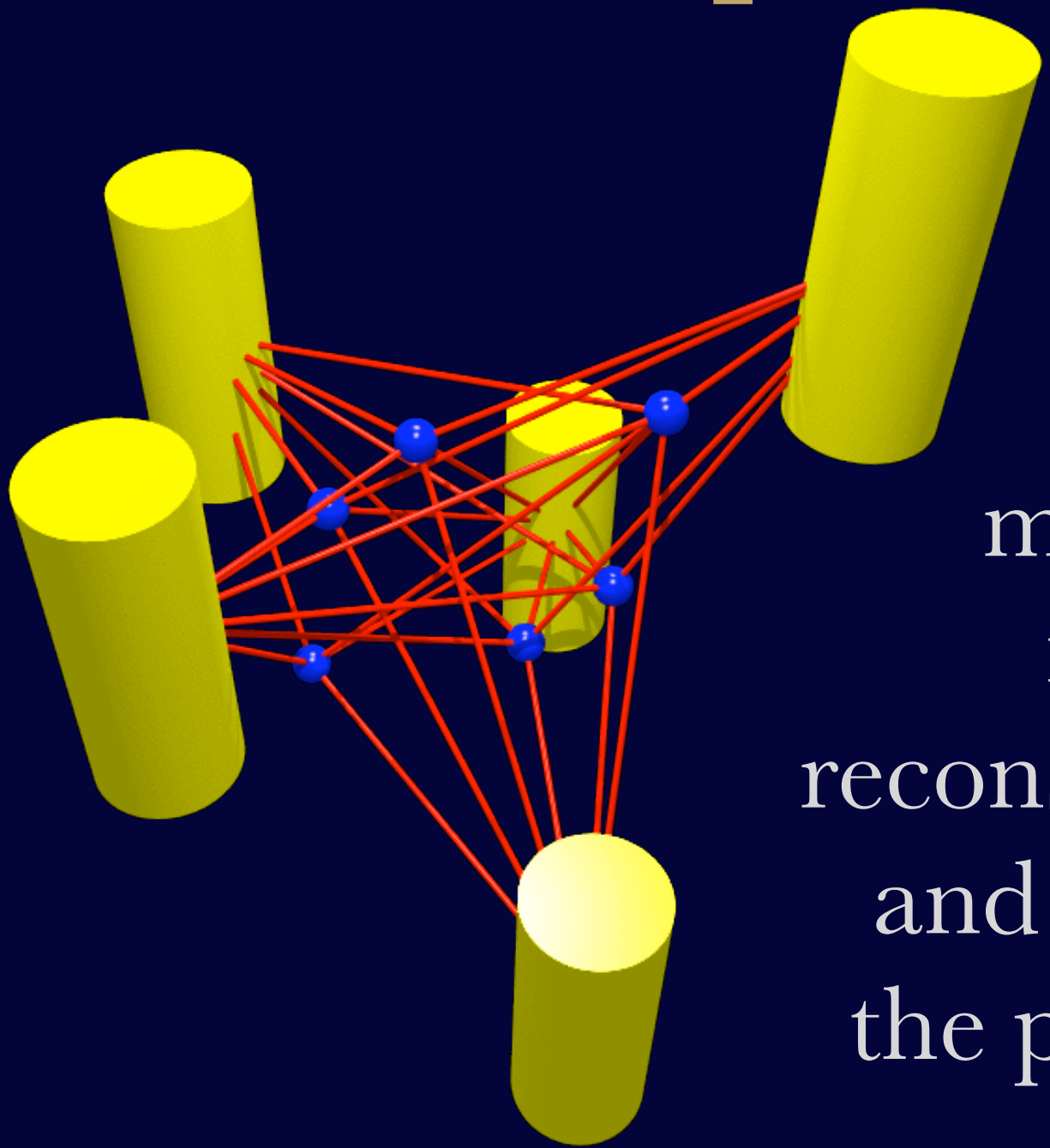
Given three photographs of an object. Can we reconstruct the points and the camera positions and parameters? How many points do we have to see? How do we reconstruct cameras and point positions?



How much can we reconstruct from a movie with many frames?


Schwarzsee
Z'mutt

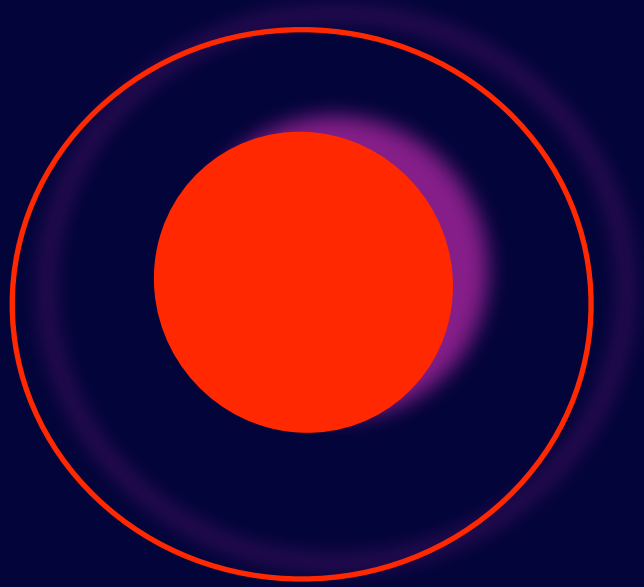
in space



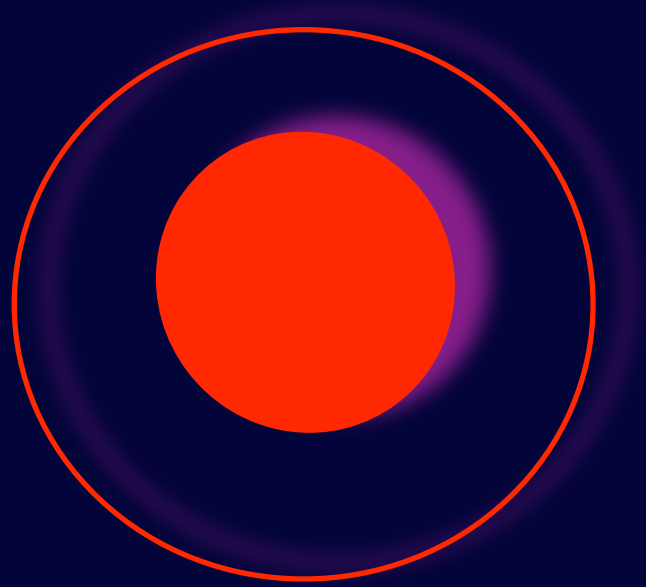
m cameras
n points
reconstruct camera
and points from
the photographs.

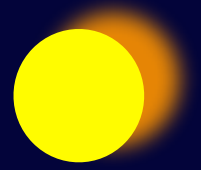
Here is a Structure from Motion problem in 2D:

We assume
an omnidirectional
camera in the plane.

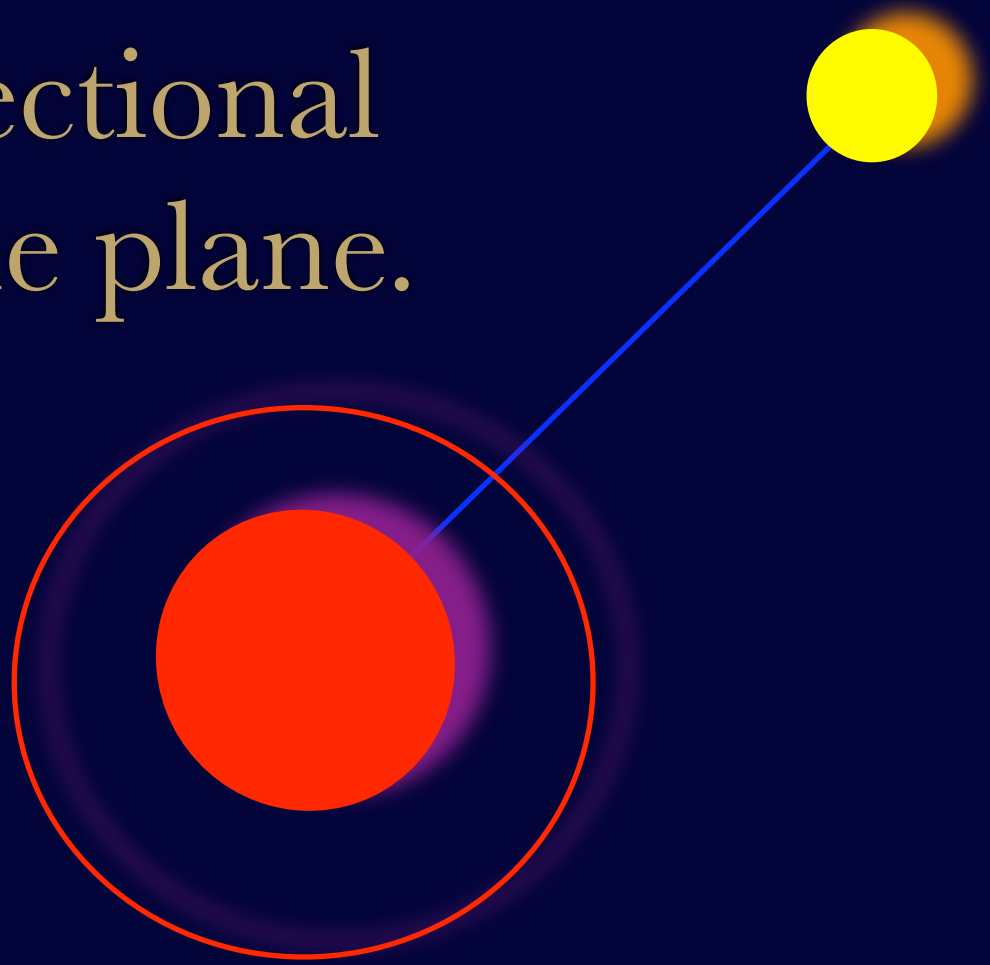


We assume
an omnidirectional
camera in the plane.

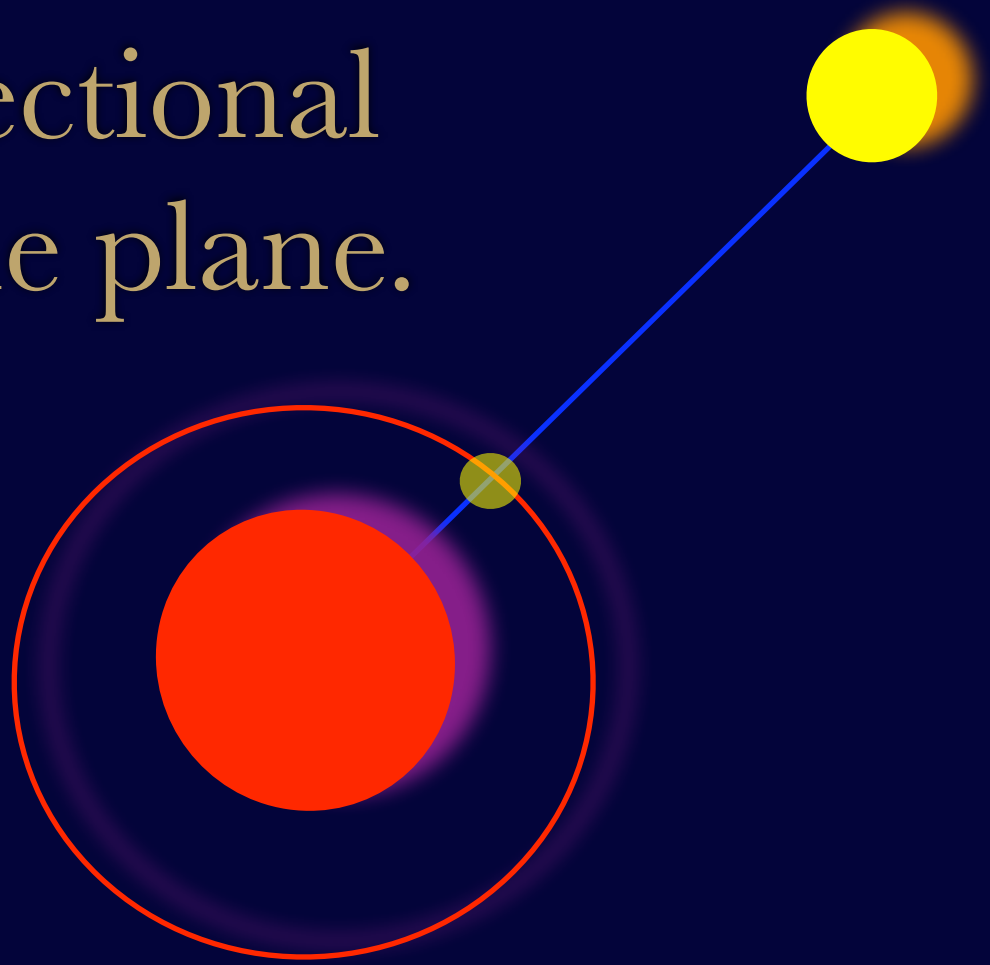




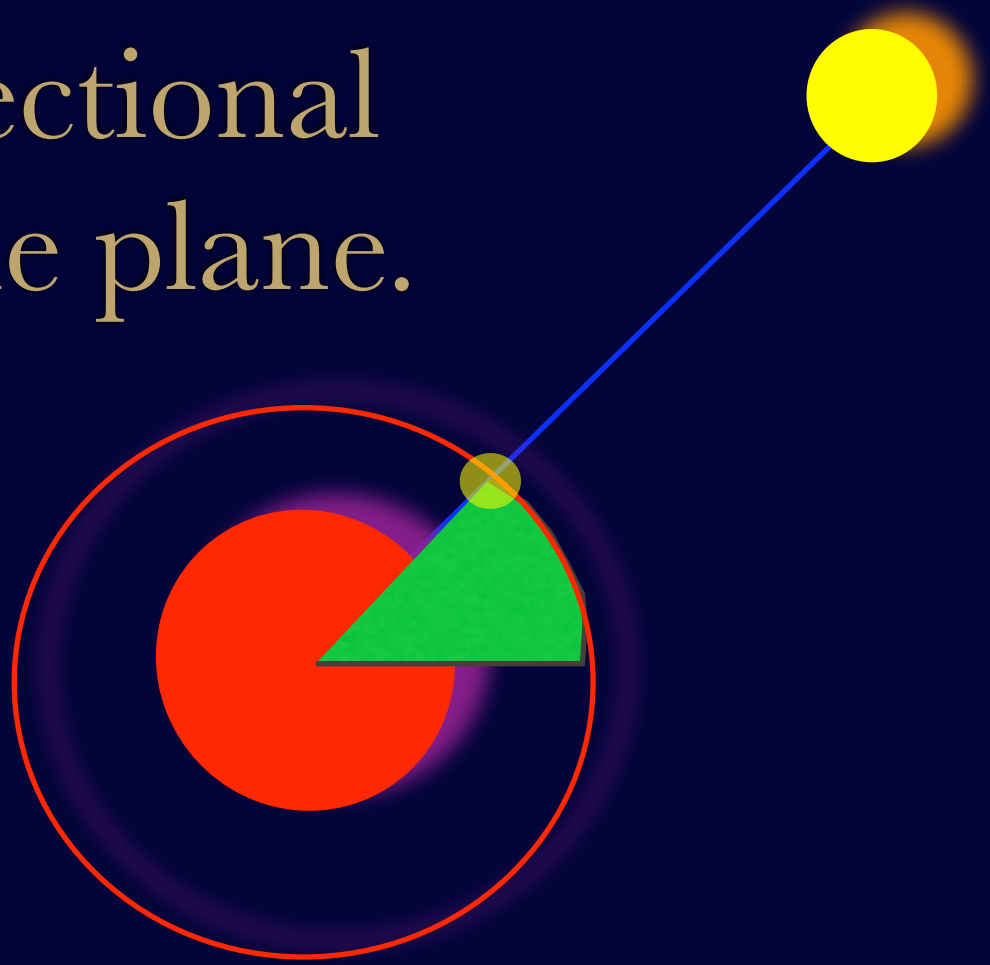
We assume
an omnidirectional
camera in the plane.



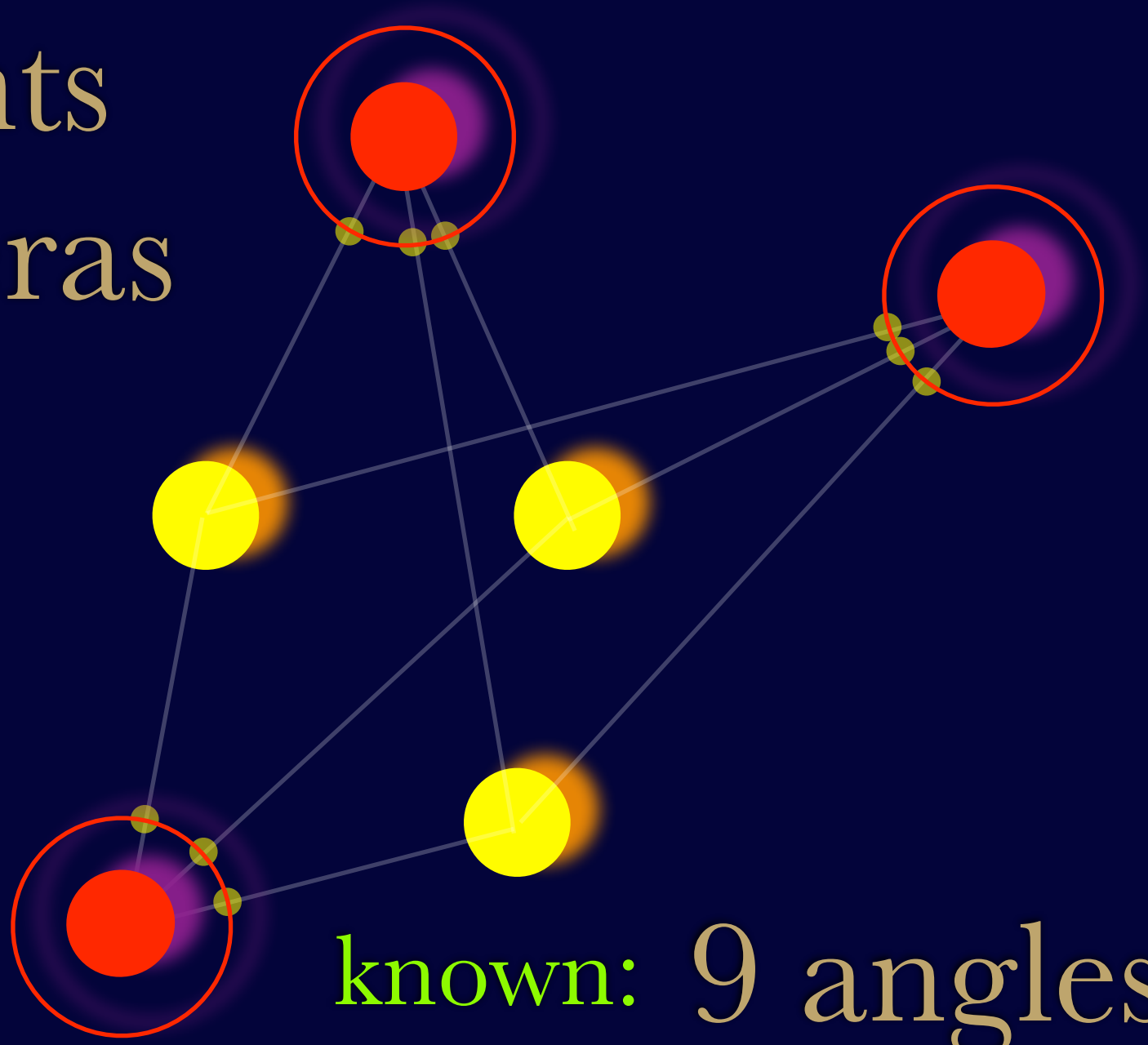
We assume
an omnidirectional
camera in the plane.



We assume
an omnidirectional
camera in the plane.



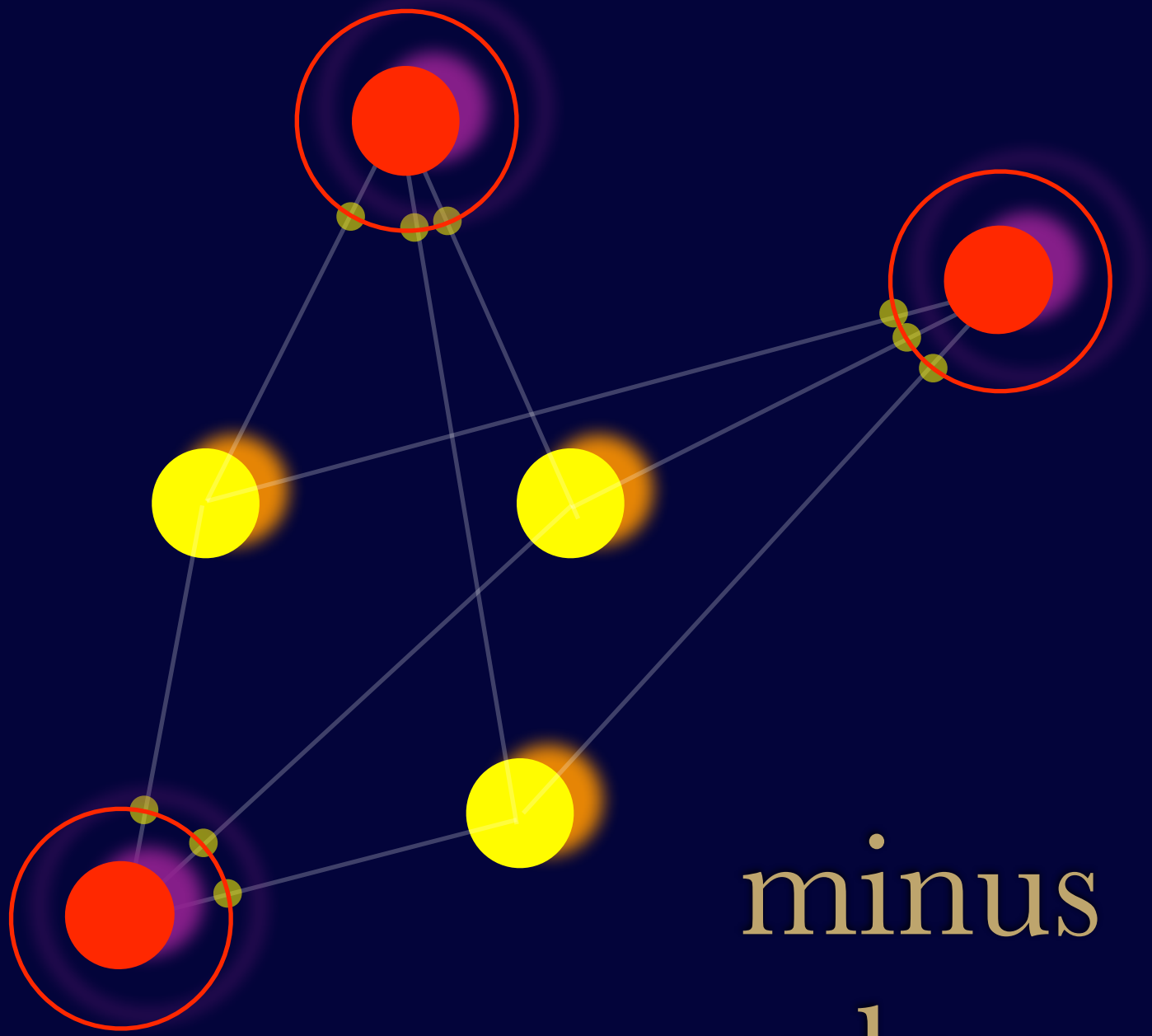
3 points
3 cameras



known: 9 angles

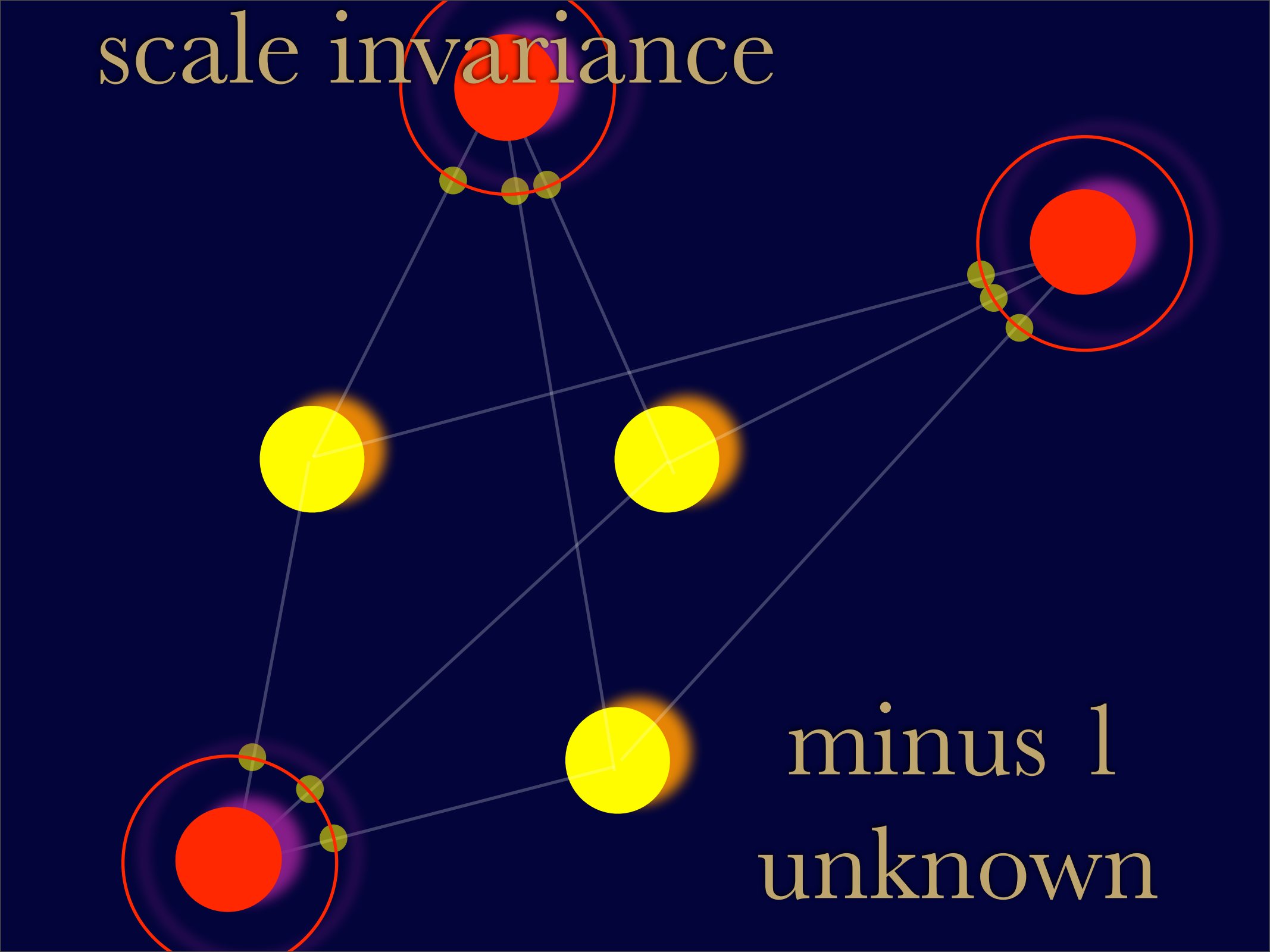
wanted: 2*6=12 coordinates

scale invariance

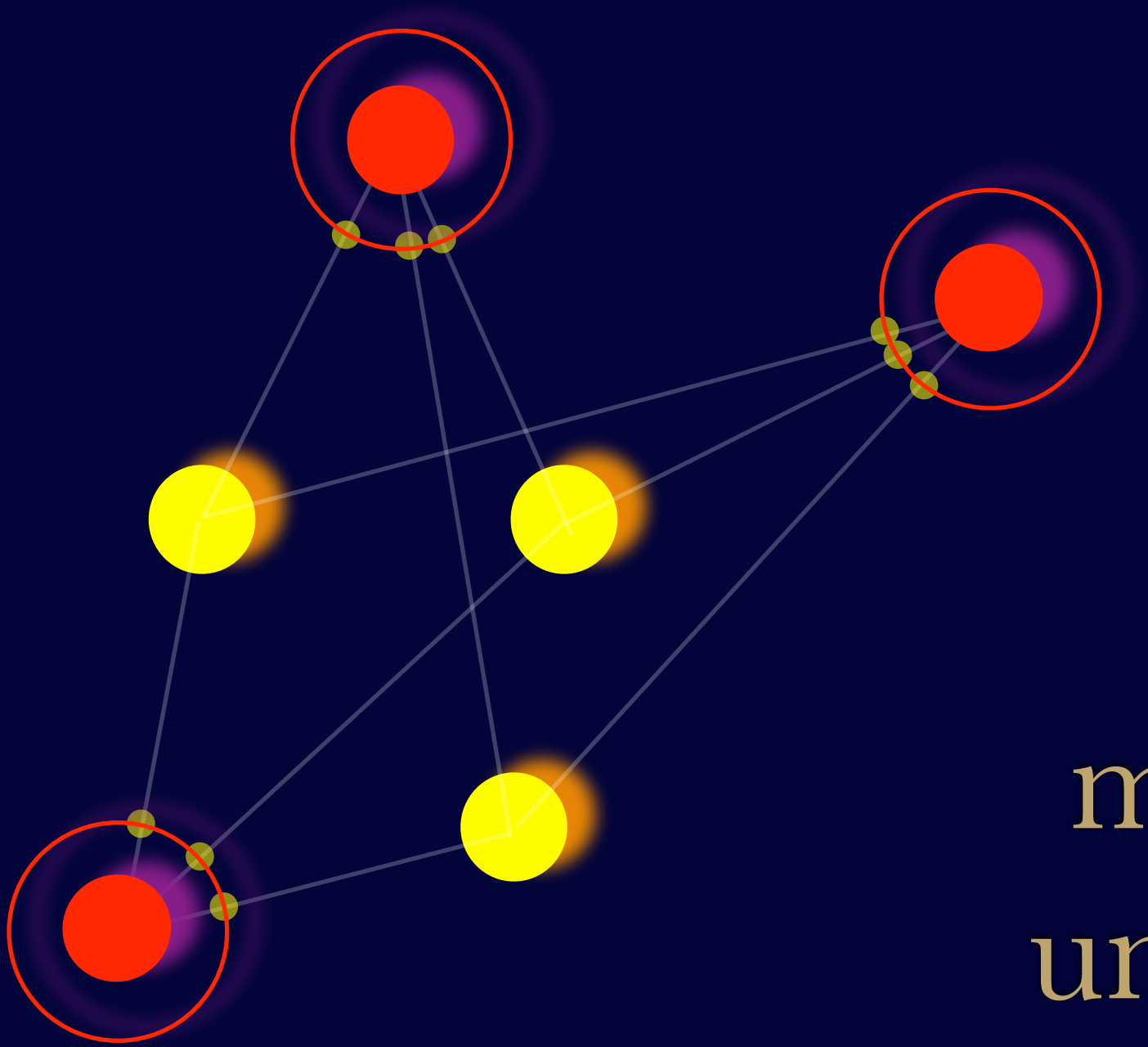


minus 1

unknown


scale invariance
minus 1
unknown

translational invariance



minus 2

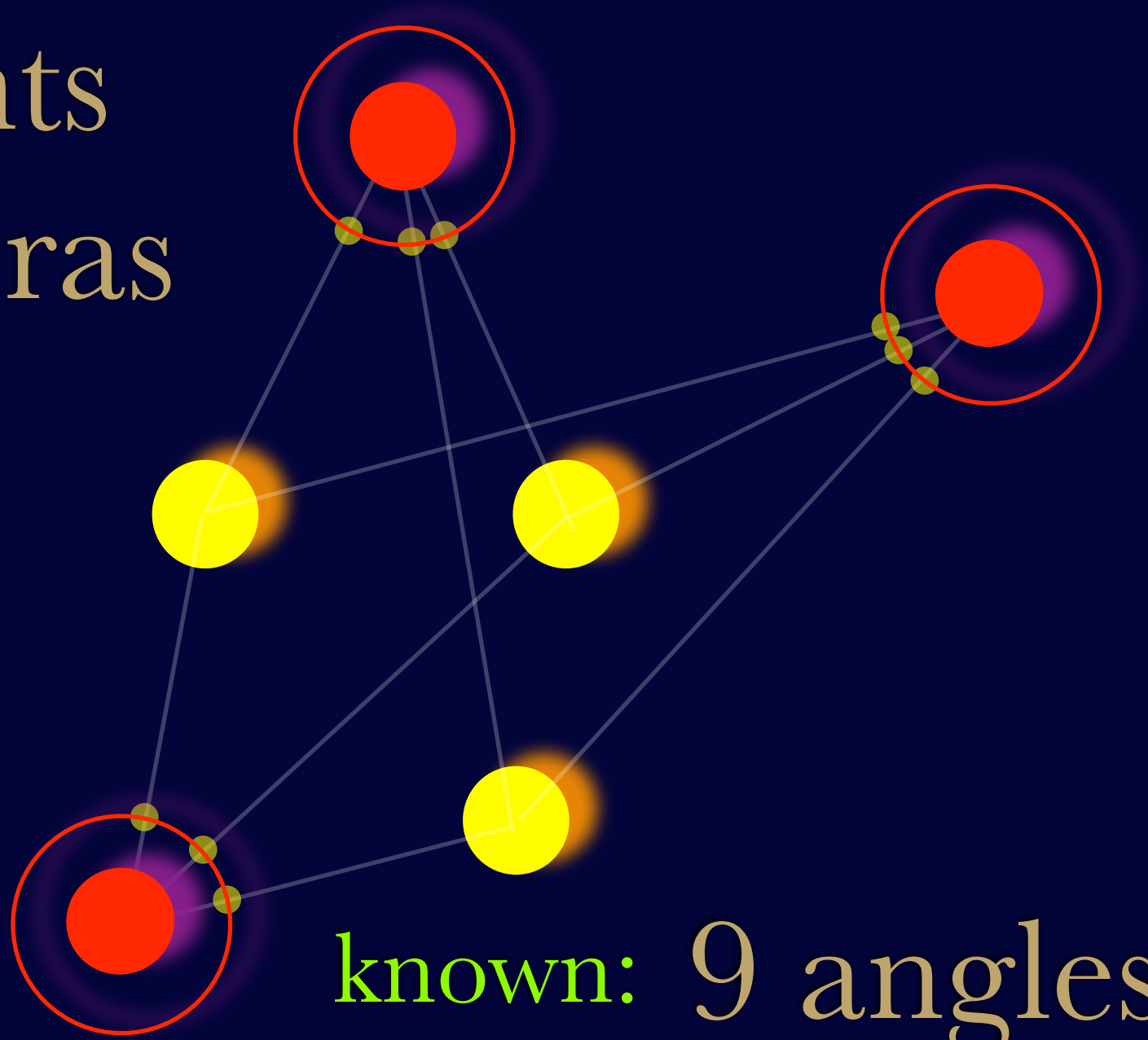
unknown

translational invariance

minus 2

unknown

3 points
3 cameras

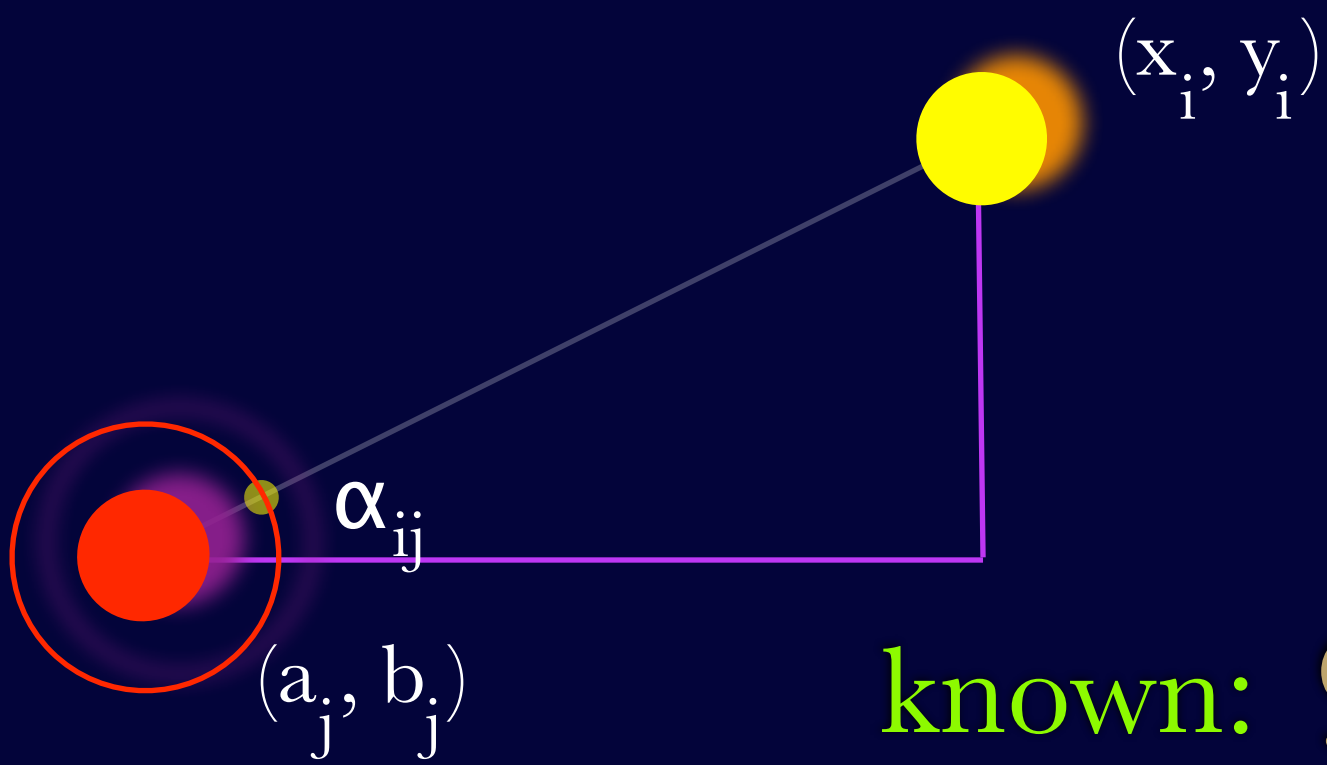


known: 9 angles

wanted: 9 coordinates

9 linear equations

$$\sin(\alpha_{ij})\,(x_i - a_j) = \cos(\alpha_{ij})\,(y_i - b_j)$$



known: 9 angles

wanted: 9 coordinates

When can it be solved?

$r_{ij}(x_i - a_j) = (y_i - b_j)$ $\qquad x_1 = y_1 = 0,\ x_2 = 1$

$$A\,x = b$$

$$\det(A) =$$

$$\begin{aligned}
& r_{21}(r_{13}r_{22}(r_{31} - r_{32})(r_{23} - r_{33}) \\
+\ & r_{12}(-(r_{23}(r_{22} - r_{32})(r_{31} - r_{33})) \\
+\ & r_{13}(r_{22}r_{31} - r_{23}r_{31} + r_{23}r_{32} \\
-\ & r_{31}r_{32} - r_{22}r_{33} + r_{31}r_{33}))) \\
+\ & r_{11}(r_{22}(r_{23}(r_{21} - r_{31})(r_{32} - r_{33}) \\
+\ & r_{13}(-(r_{23}r_{31}) - r_{21}r_{32} + r_{23}r_{32} + r_{31}r_{32} + r_{21}r_{33} - r_{32}r_{33})) \\
+\ & r_{12}(r_{13}(-(r_{22}r_{31}) + r_{23}r_{31} + r_{21}r_{32} - r_{23}r_{32} - r_{21}r_{33} + r_{22}r_{33}) \\
+\ & r_{23}(r_{22}r_{31} - r_{21}r_{32} + r_{21}r_{33} - r_{22}r_{33} - r_{31}r_{33} + r_{32}r_{33})))
\end{aligned}$$

no insight....

Algebra ⟷ Geometry

Decartes teaching Queen Christine

This is an example, where geometry will give more insight than algebra.

Decartes teaching Queen Christine

Theorem (Knill-Ramirez)

In the plane, if the 3 omni cameras are noncollinear, the 3 points are noncollinear and the union of cameras and points is not contained in the union of two lines, then the structure of motion problem has a unique solution.

lemmas:

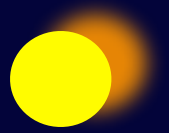
fixed point deformable point

1. the problem is linear. The space of possible solutions is an affine linear space.

2. if three cameras are fixed, then every point has to fixed.

3. if three points are fixed, then every camera has to fixed.

lemmas:



fixed point deformable point

1. the problem is linear. The space of possible solutions is an affine linear space.

2. if three cameras are fixed, then every point has to fixed.

3. if three points are fixed, then every camera has to fixed.

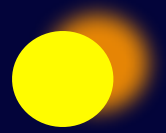
lemmas:

1. the problem is linear. The space of possible solutions is an affine linear space.

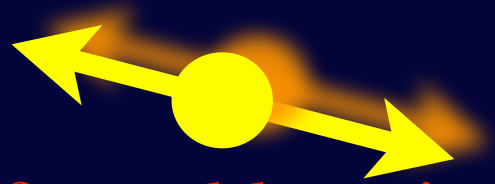
2. if three cameras are fixed, then every point has to fixed.

3. if three points are fixed, then every camera has to fixed.

lemmas:



fixed point



deformable point

1. the problem is linear. The space of possible solutions is an affine linear space.

2. if three cameras are fixed, then every point has to fixed.

3. if three points are fixed, then every camera has to fixed.

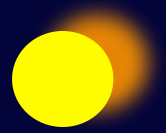
lemmas:

1. the problem is linear. The space of possible solutions is an affine linear space.

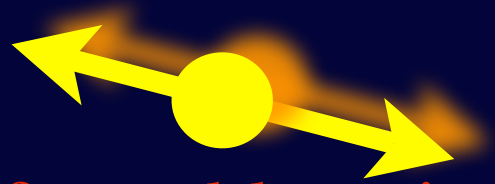
2. if three cameras are fixed, then every point has to fixed.

3. if three points are fixed, then every camera has to fixed.

lemmas:

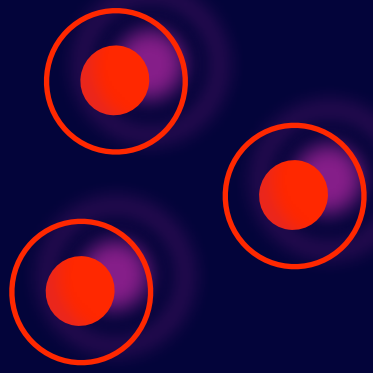


fixed point



deformable point

1. the problem is linear. The space of possible solutions is an affine linear space.



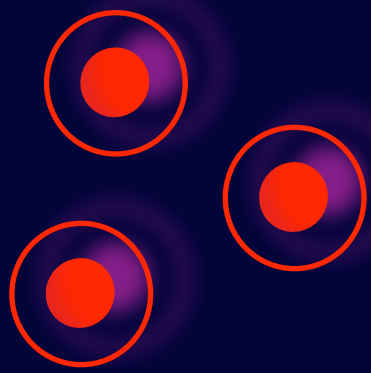
2. if three cameras are fixed, then every point has to fixed.

3. if three points are fixed, then every camera has to fixed.

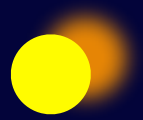
lemmas:

1. the problem is linear. The space of possible solutions is an affine linear space.

2. if three cameras are fixed, then every point has to fixed.



3. if three points are fixed, then every camera has to fixed.

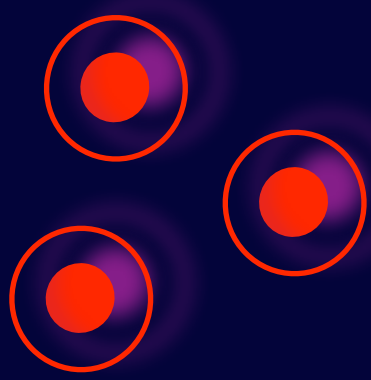


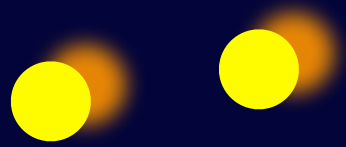
lemmas:

1. the problem is linear. The space of possible solutions is an affine linear space.

2. if three cameras are fixed, then every point has to fixed.

3. if three points are fixed, then every camera has to fixed.



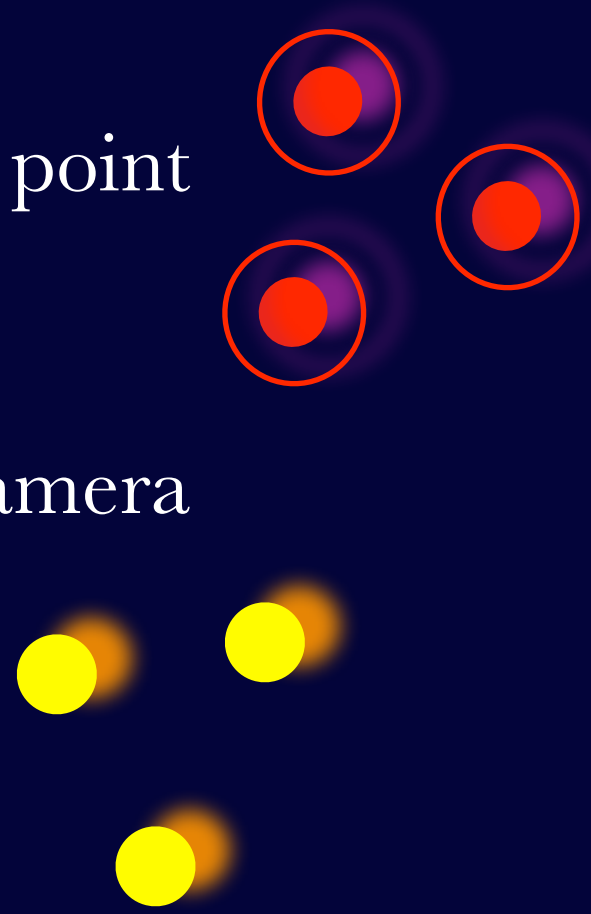


lemmas:

1. the problem is linear. The space of possible solutions is an affine linear space.

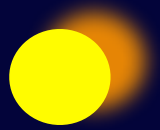
2. if three cameras are fixed, then every point has to fixed.

3. if three points are fixed, then every camera has to fixed.



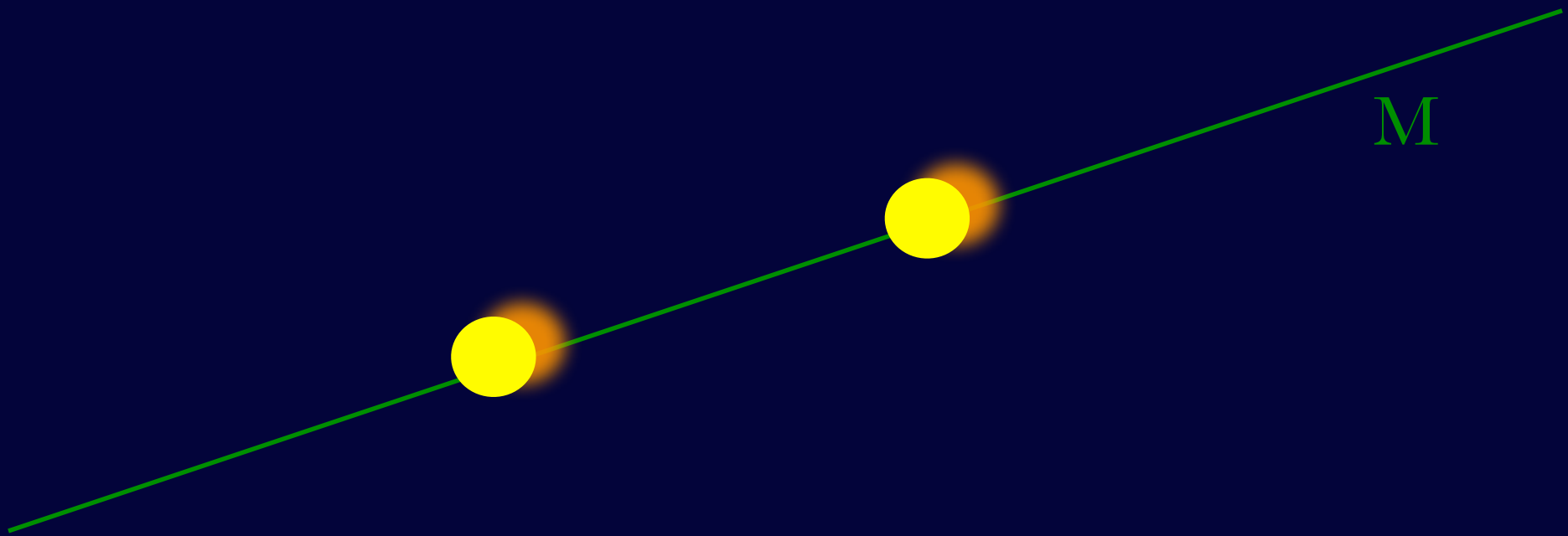
Proof:

Proof:

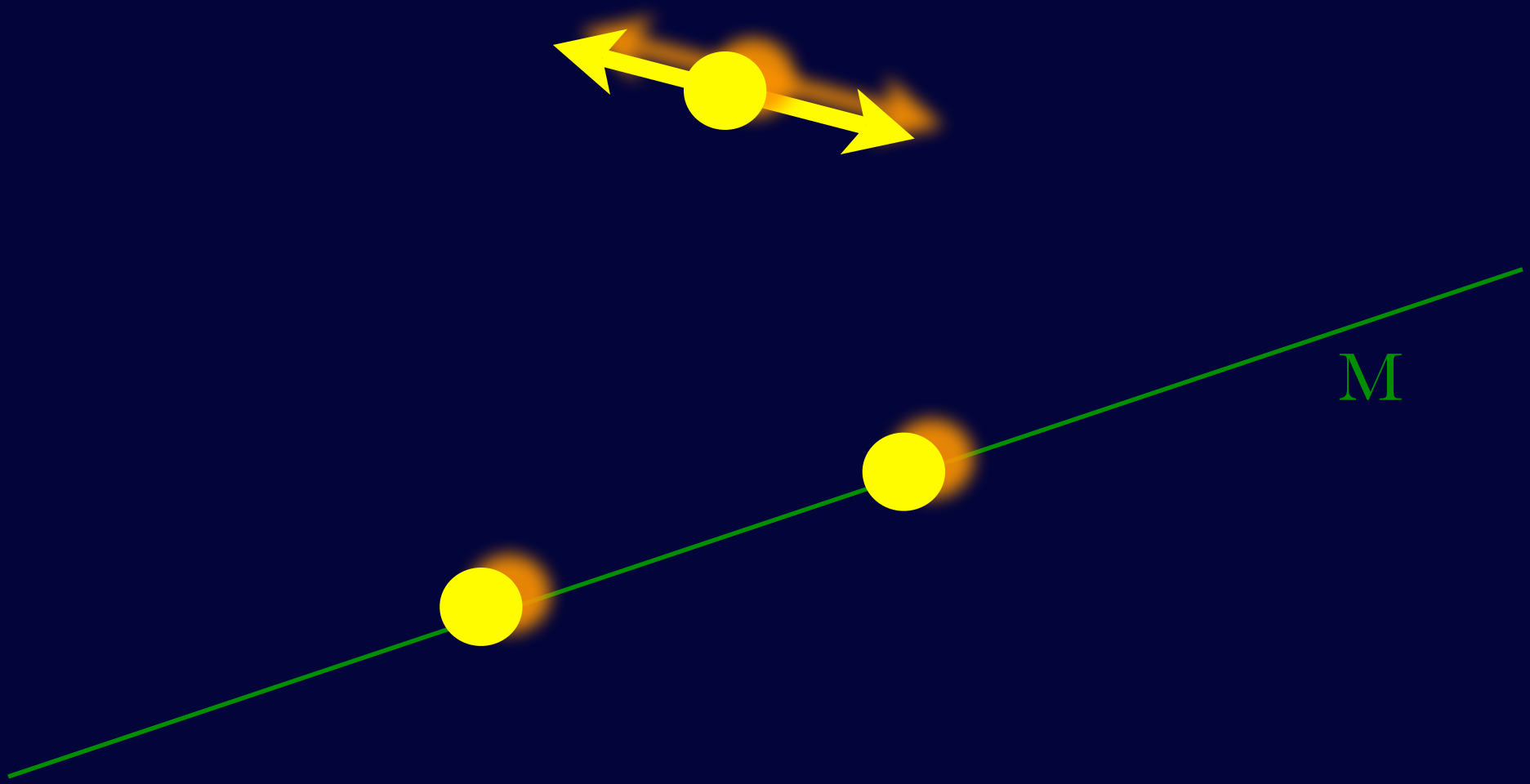


Proof:

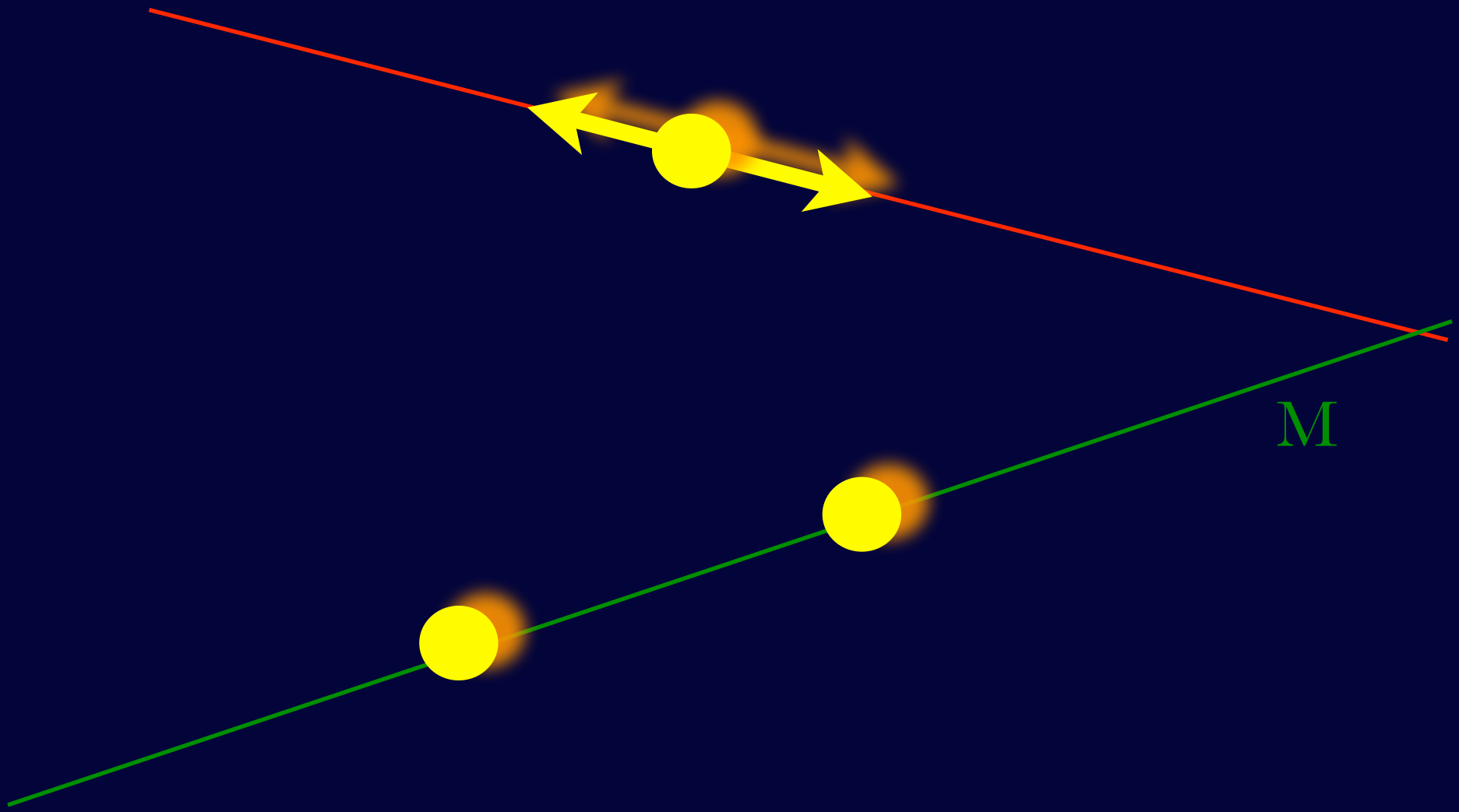
Proof:



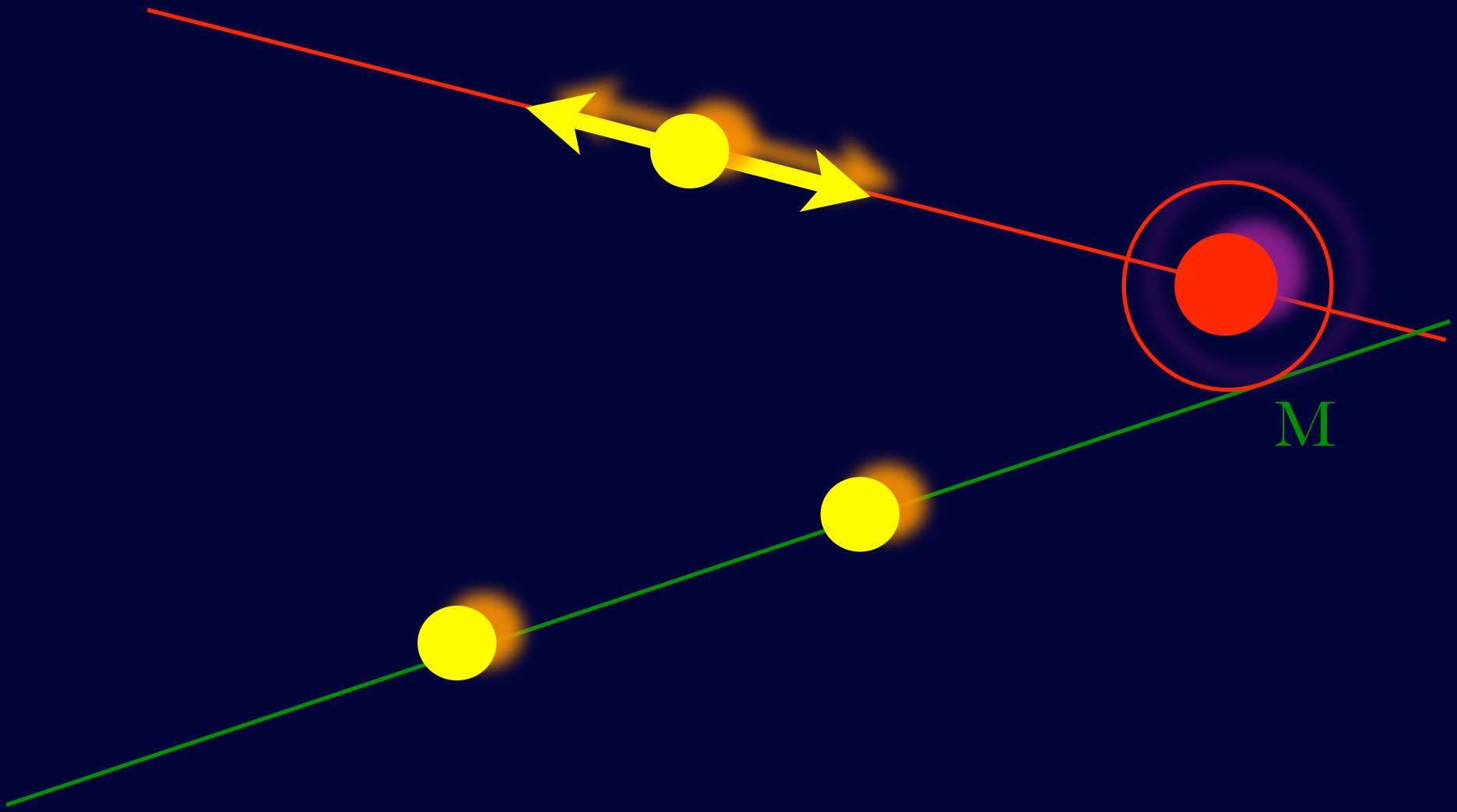
Proof:



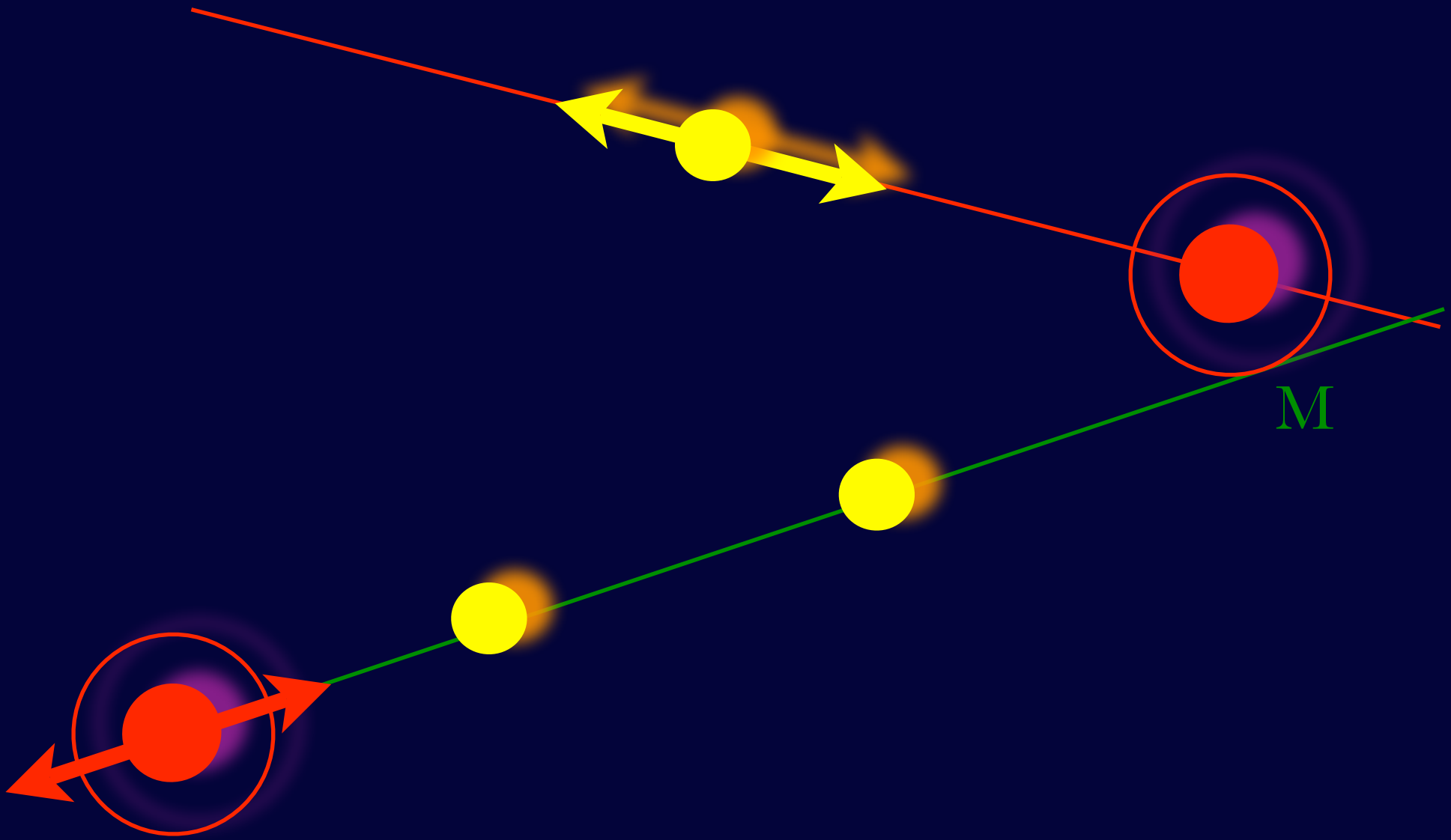
Proof:



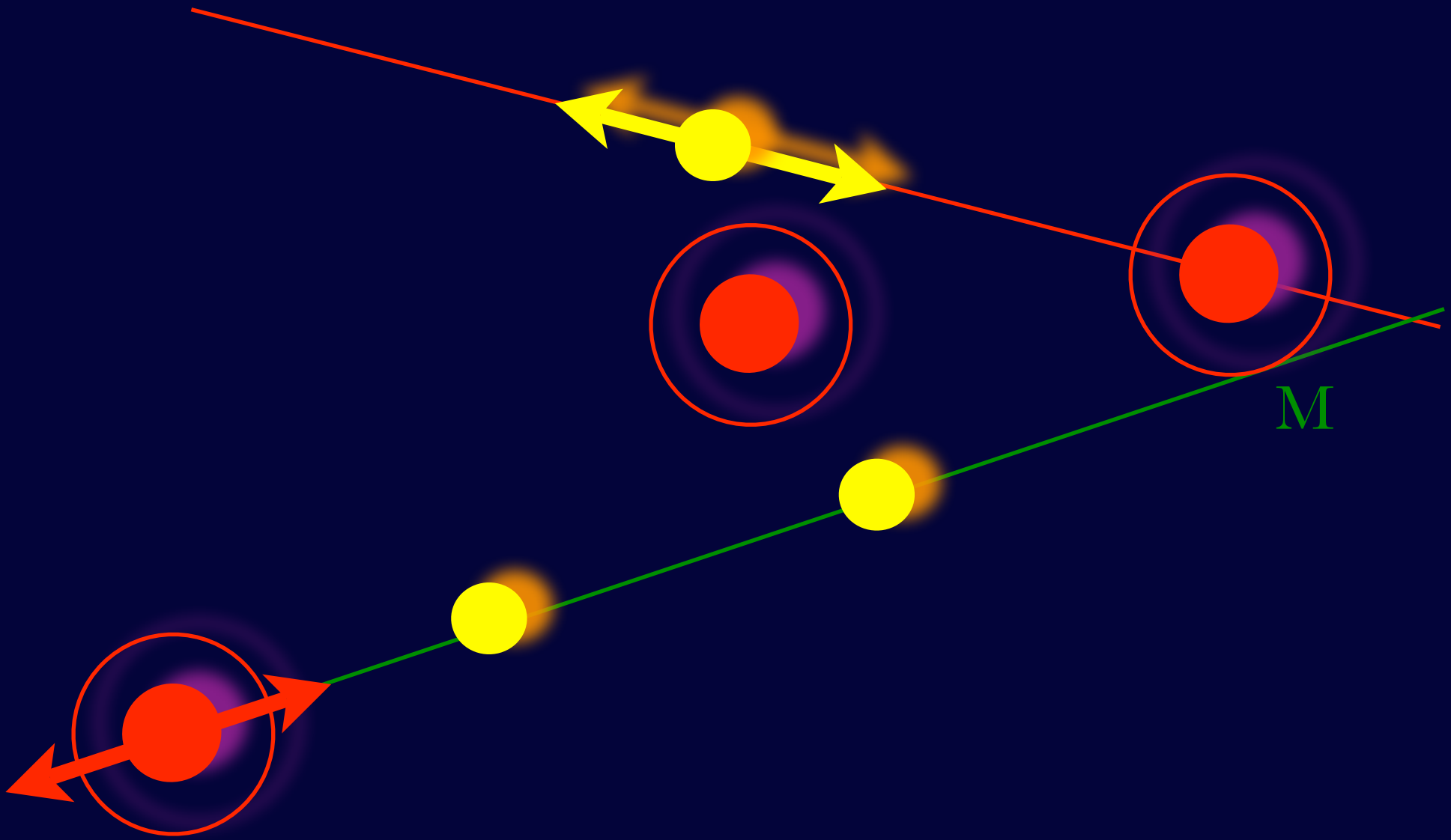
Proof:



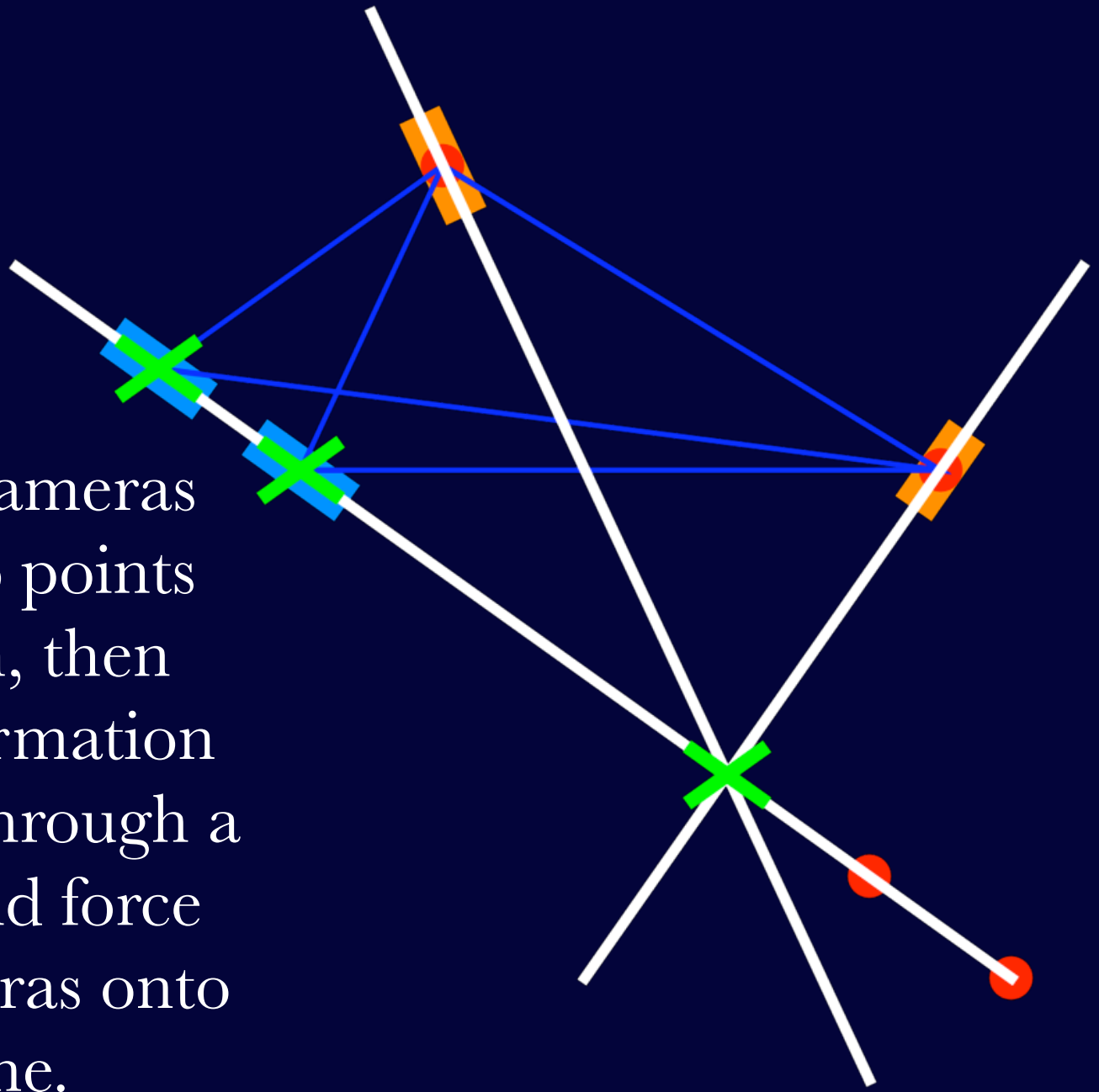
Proof:



Proof:

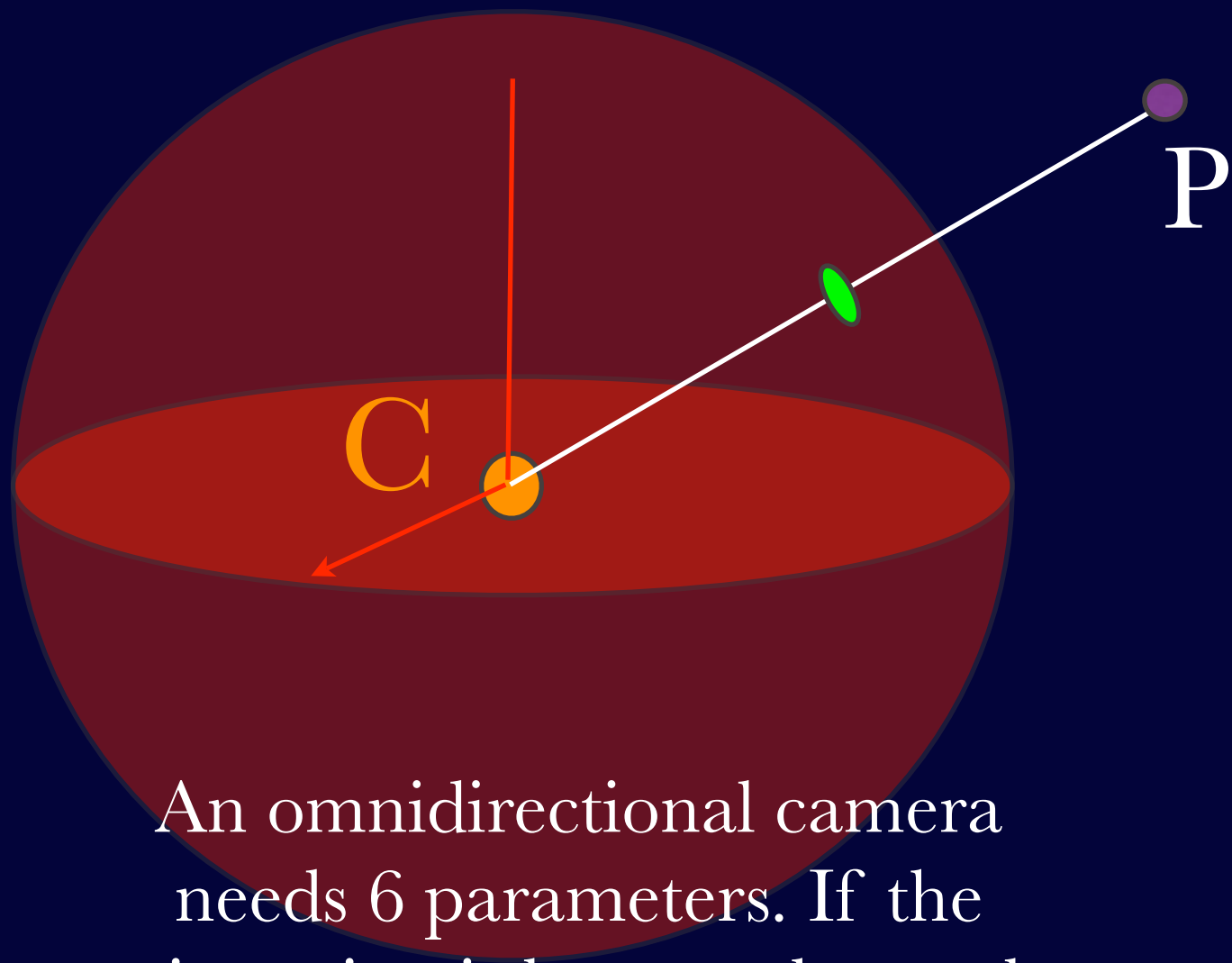


Desargues:



if two cameras and two points deform, then the deformation lines go through a point and force the cameras onto a line.

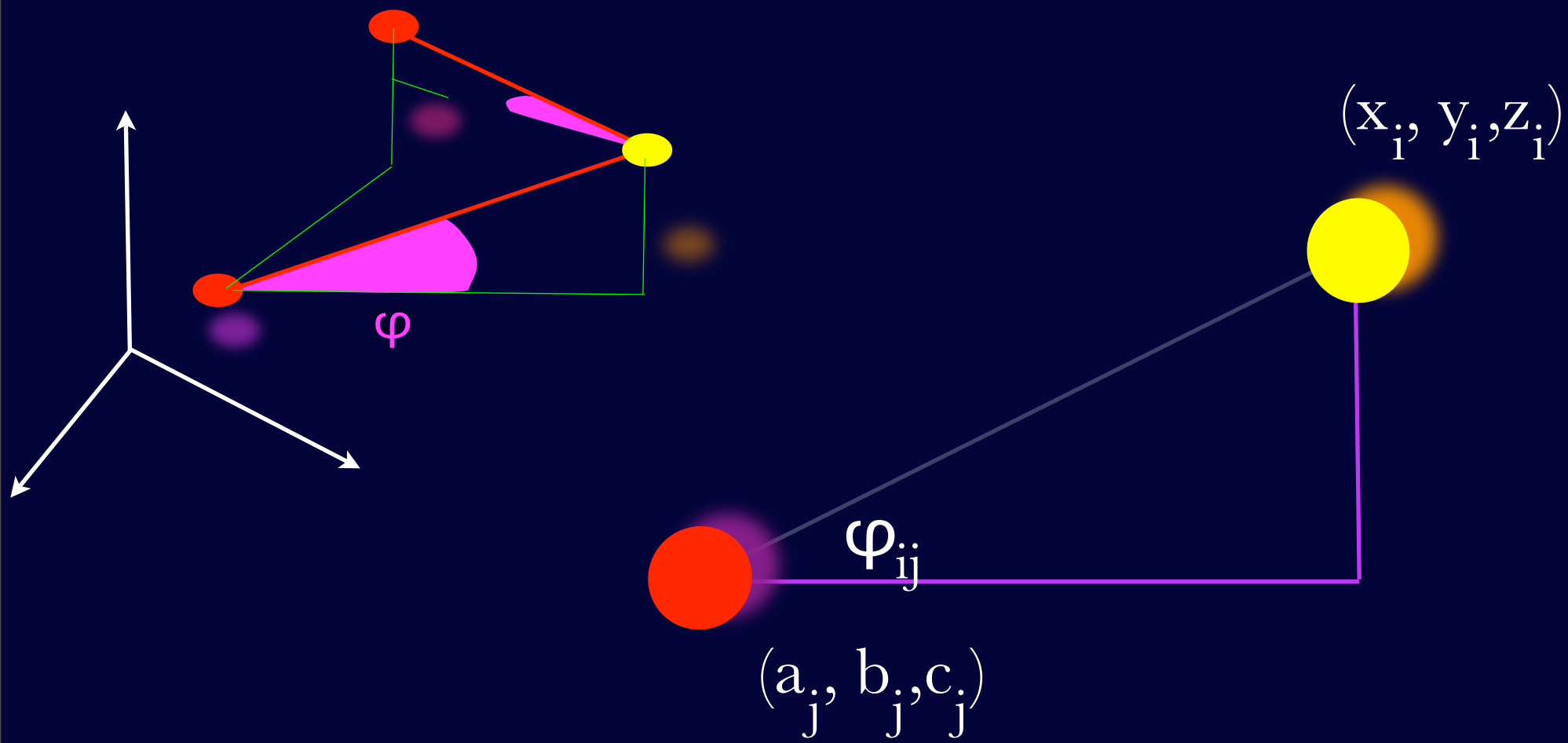
Spherical camera



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

From the plane to space:

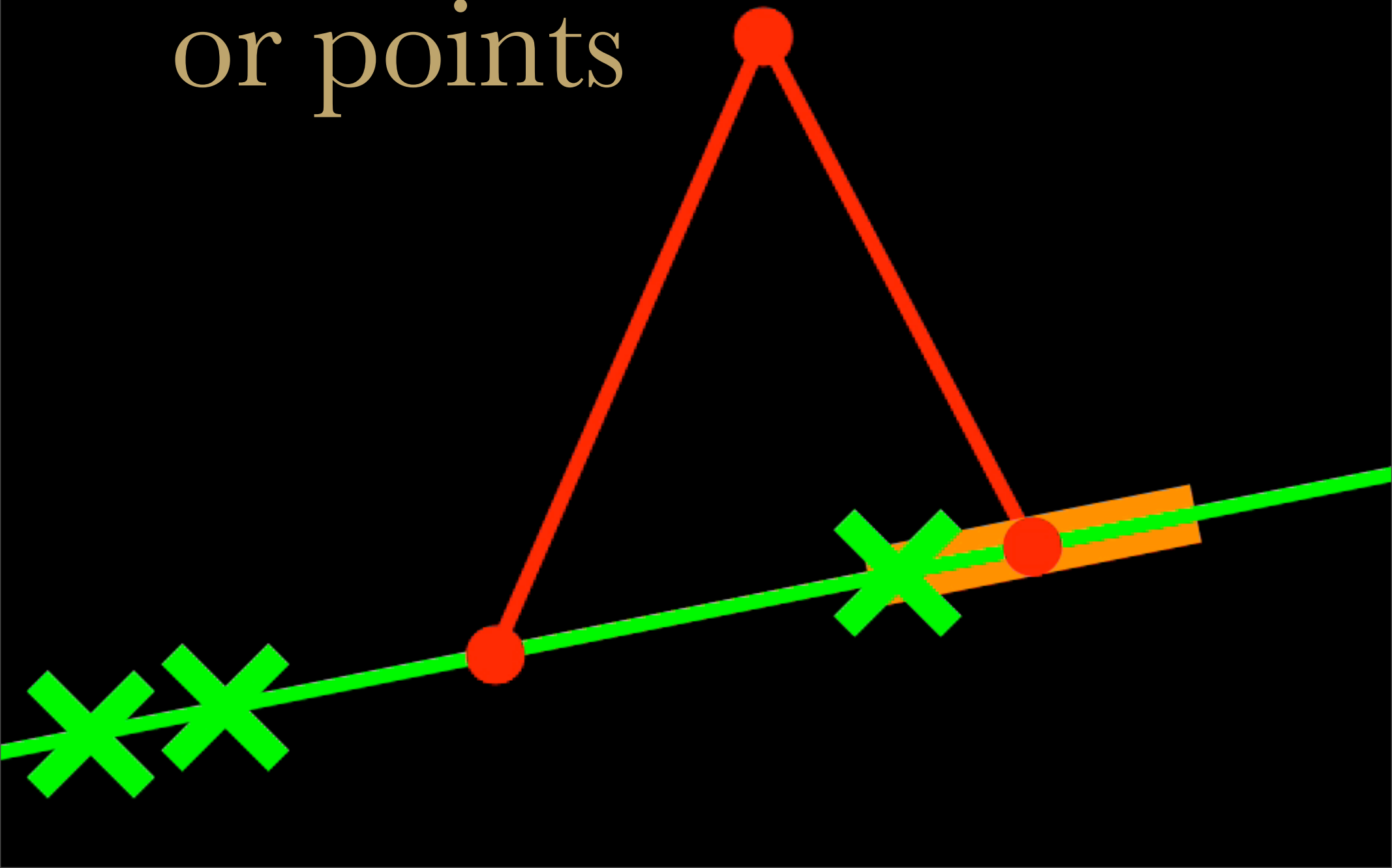
we know the (x,y) coordinates already.
we can get the z coordinates from the angle φ



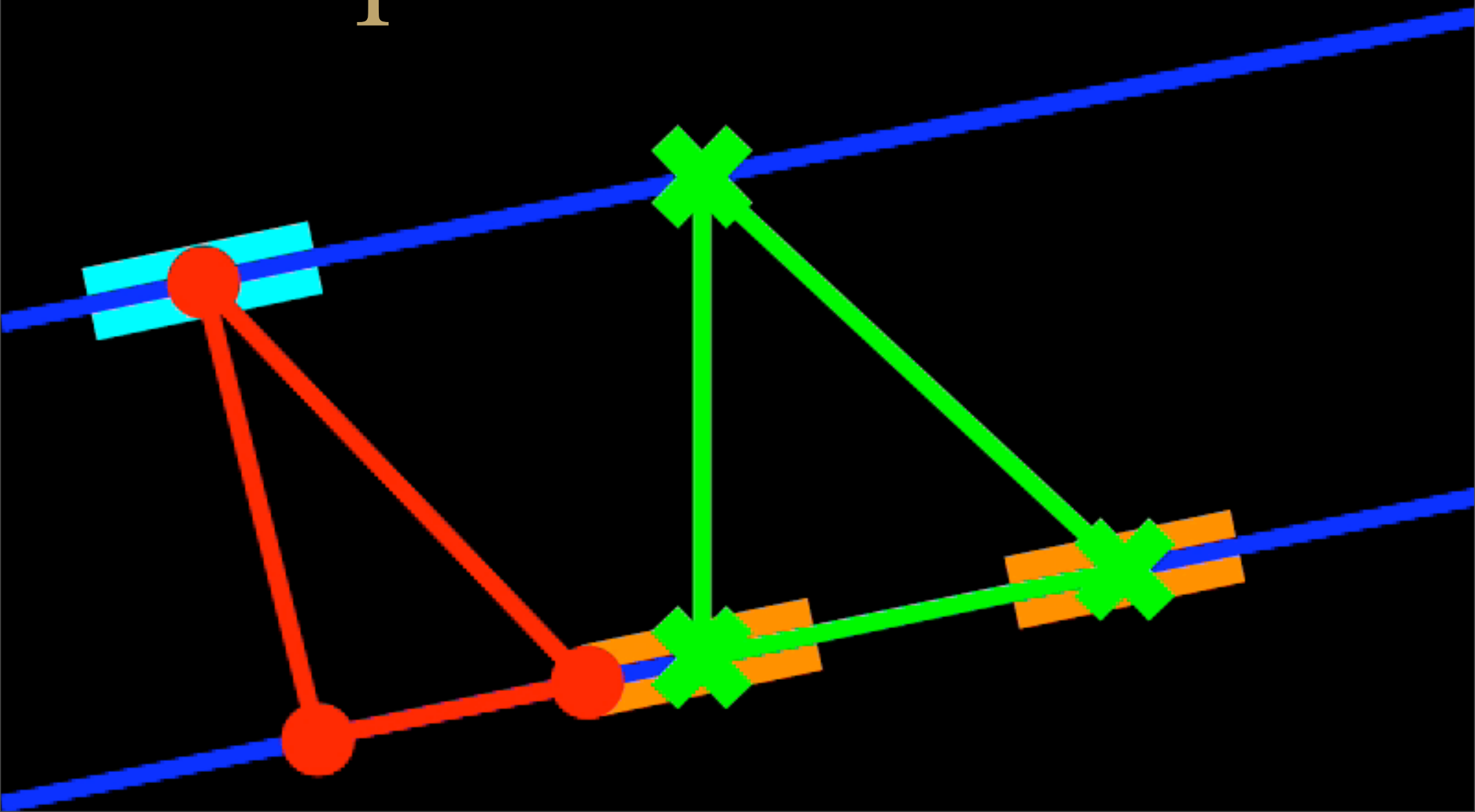
The result is sharp

Examples:

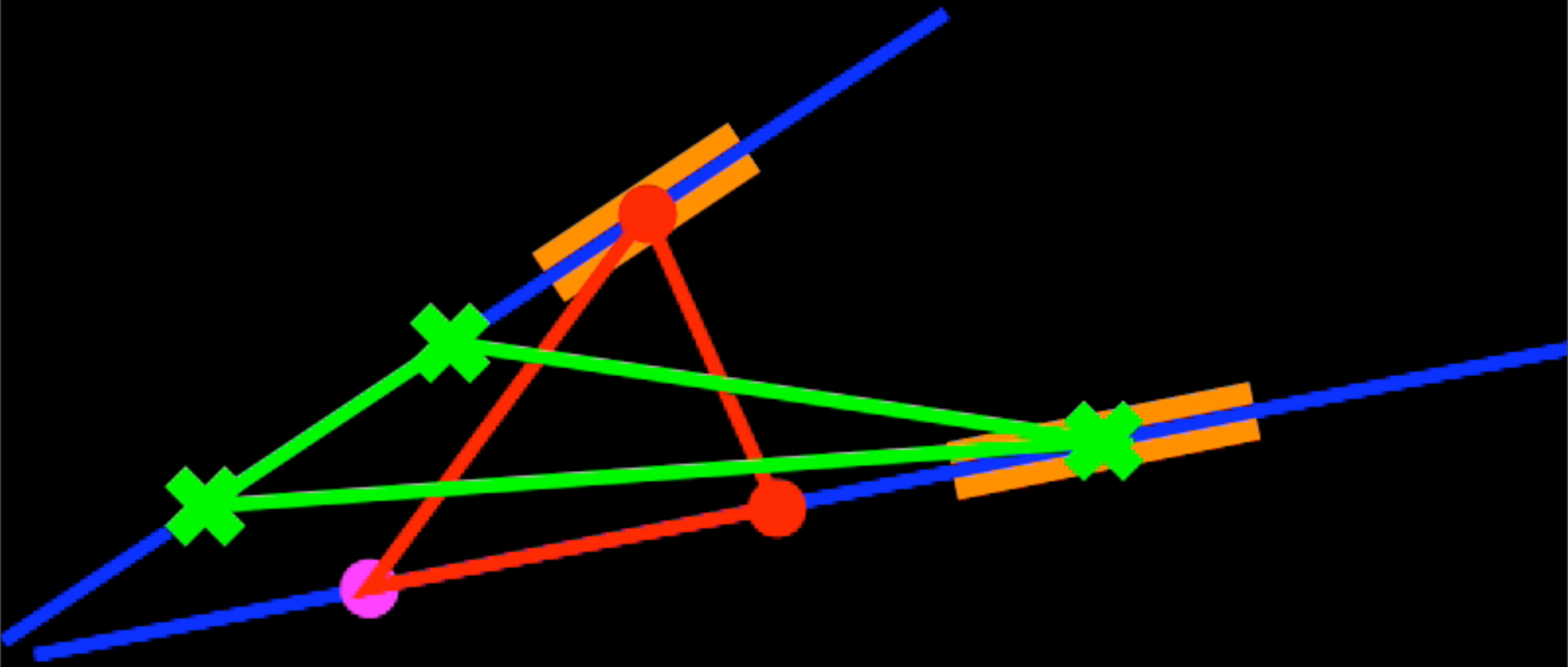
collinear cameras
or points



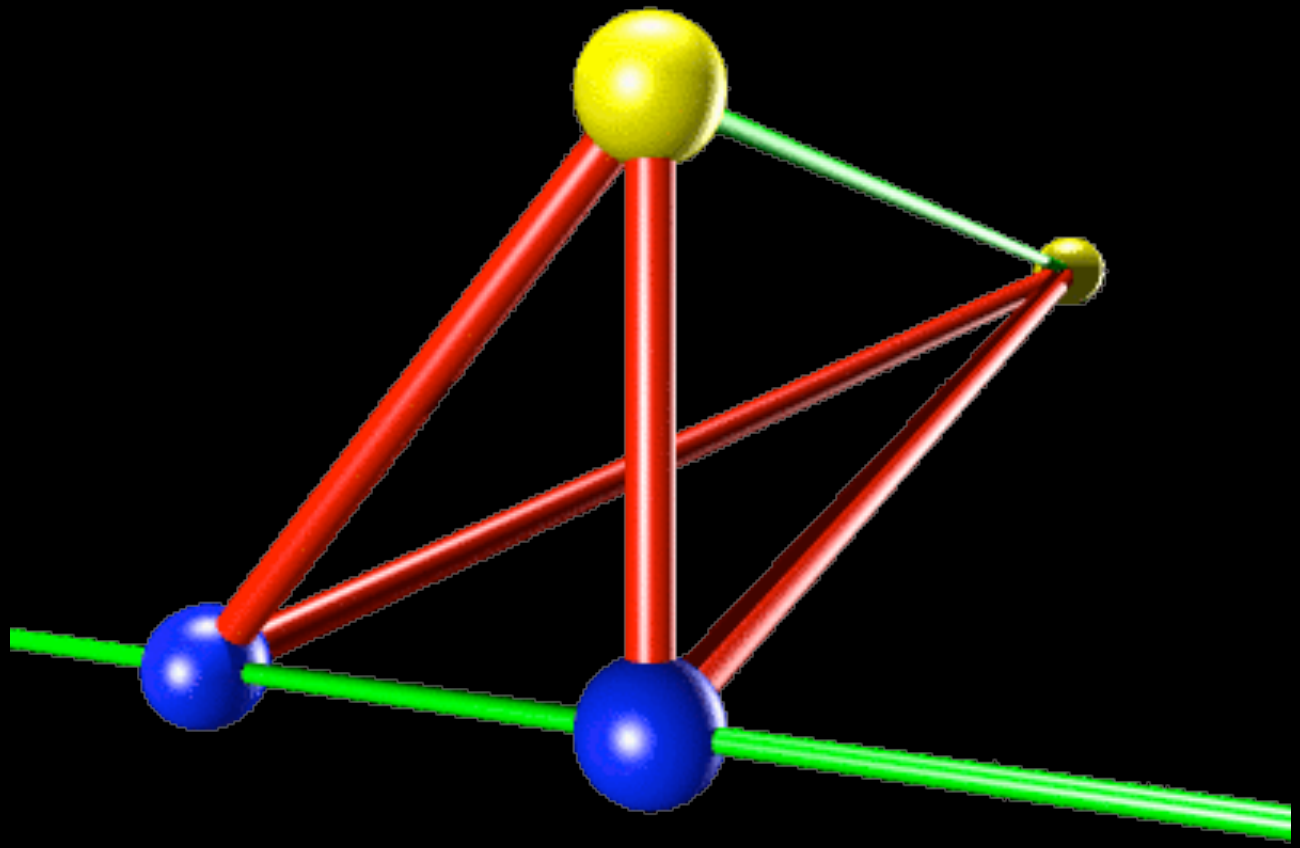
Union of two parallel lines



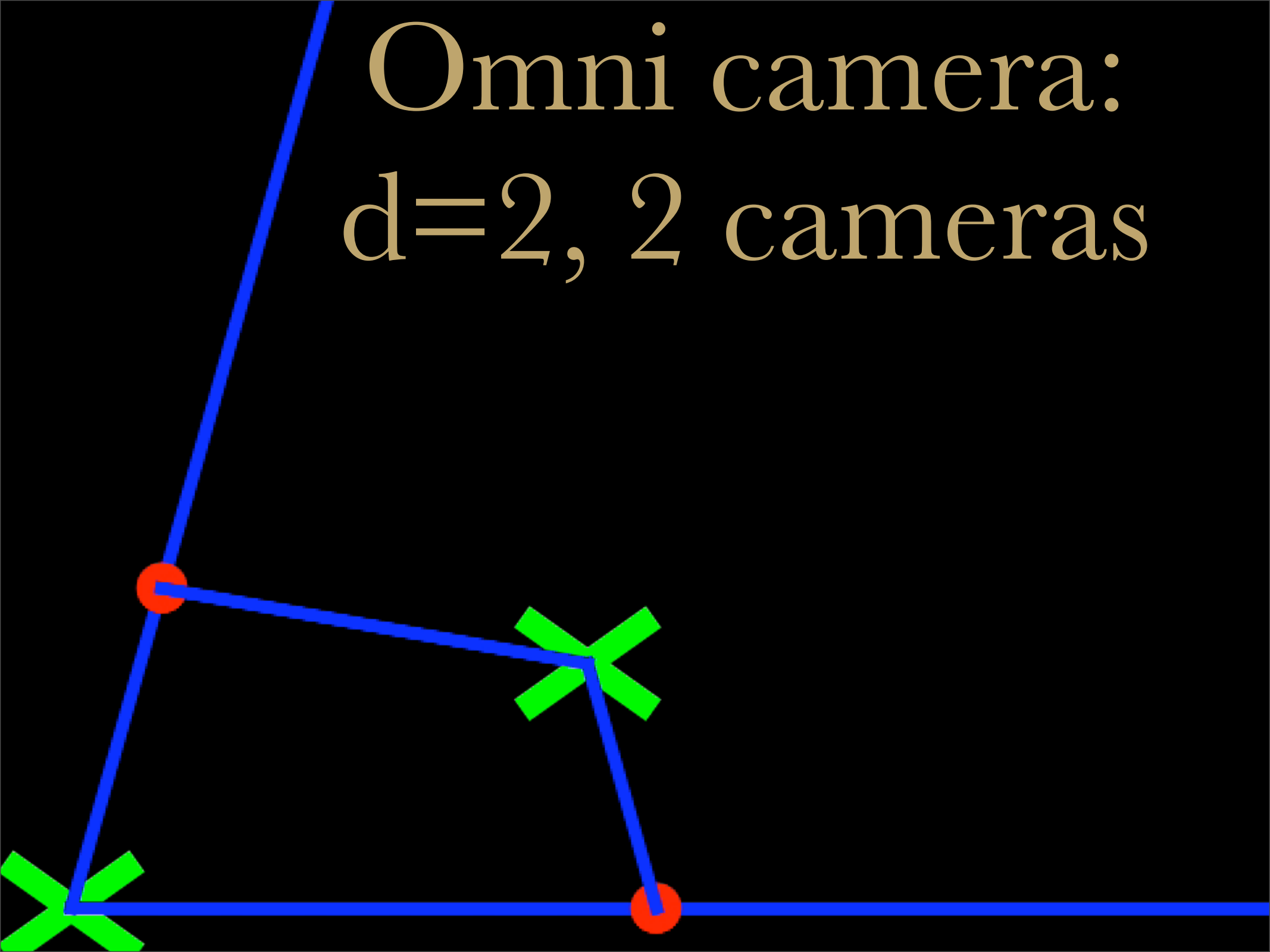
Union of two lines



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

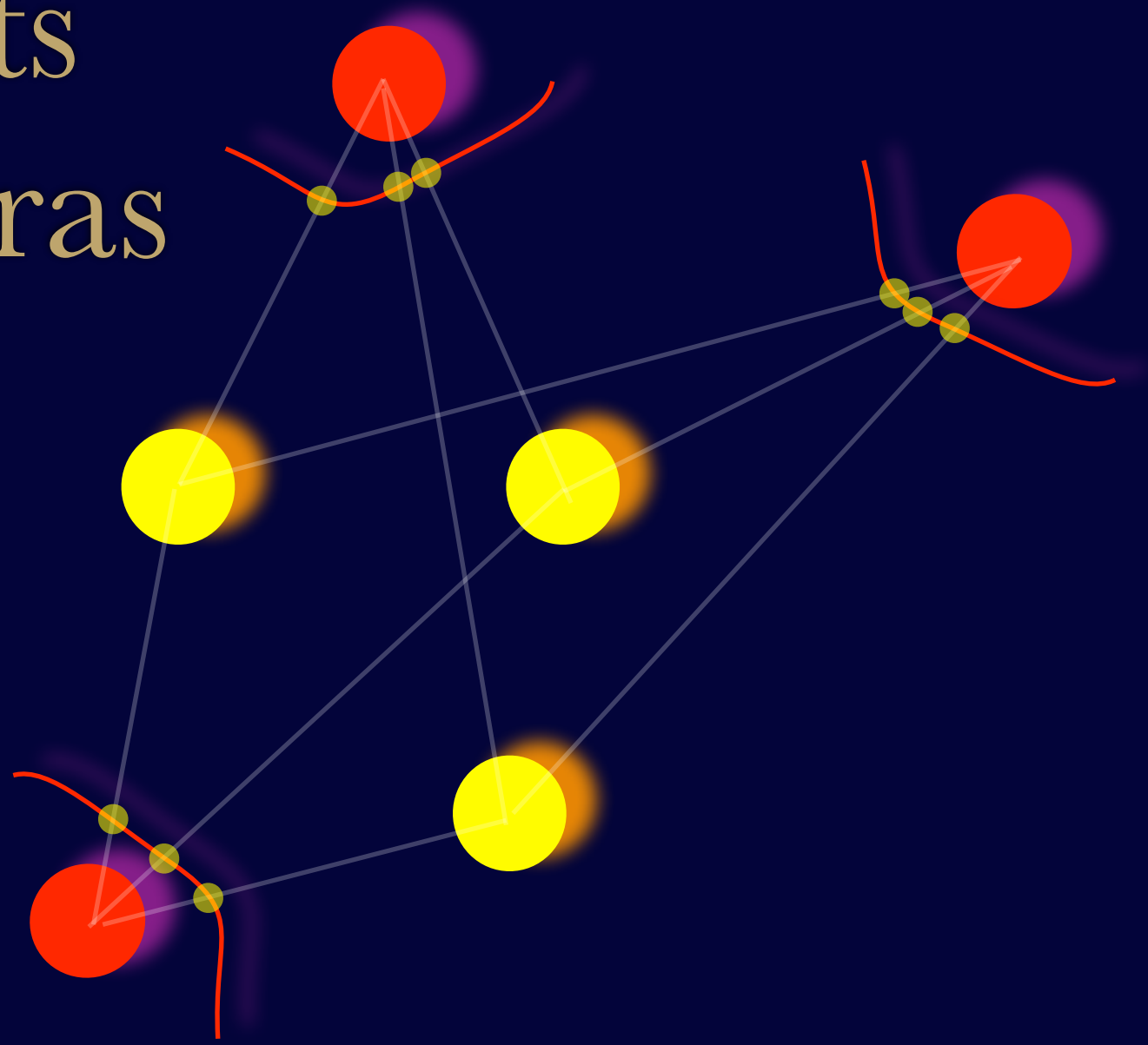


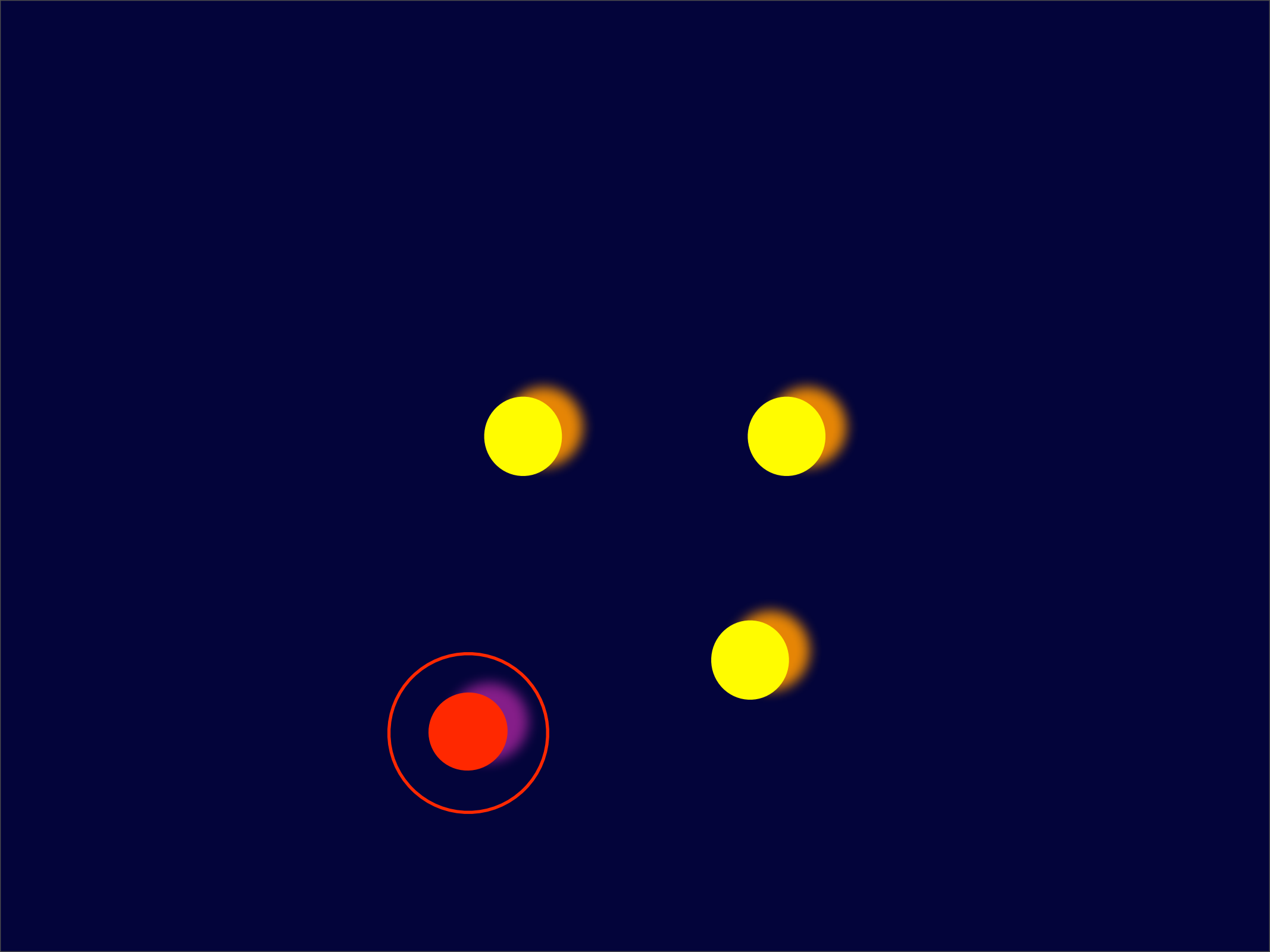
Omni camera: d=2, 2 cameras

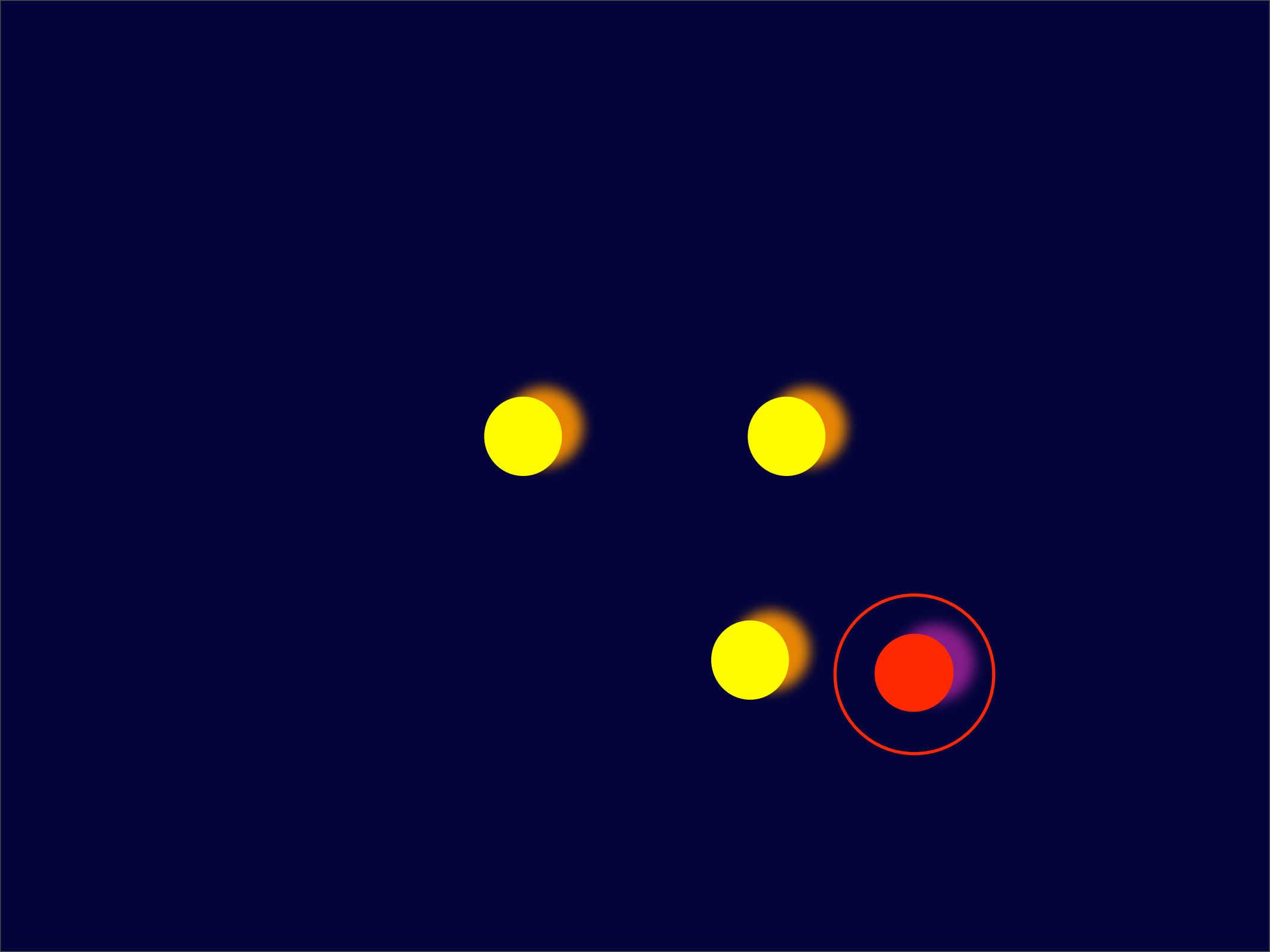


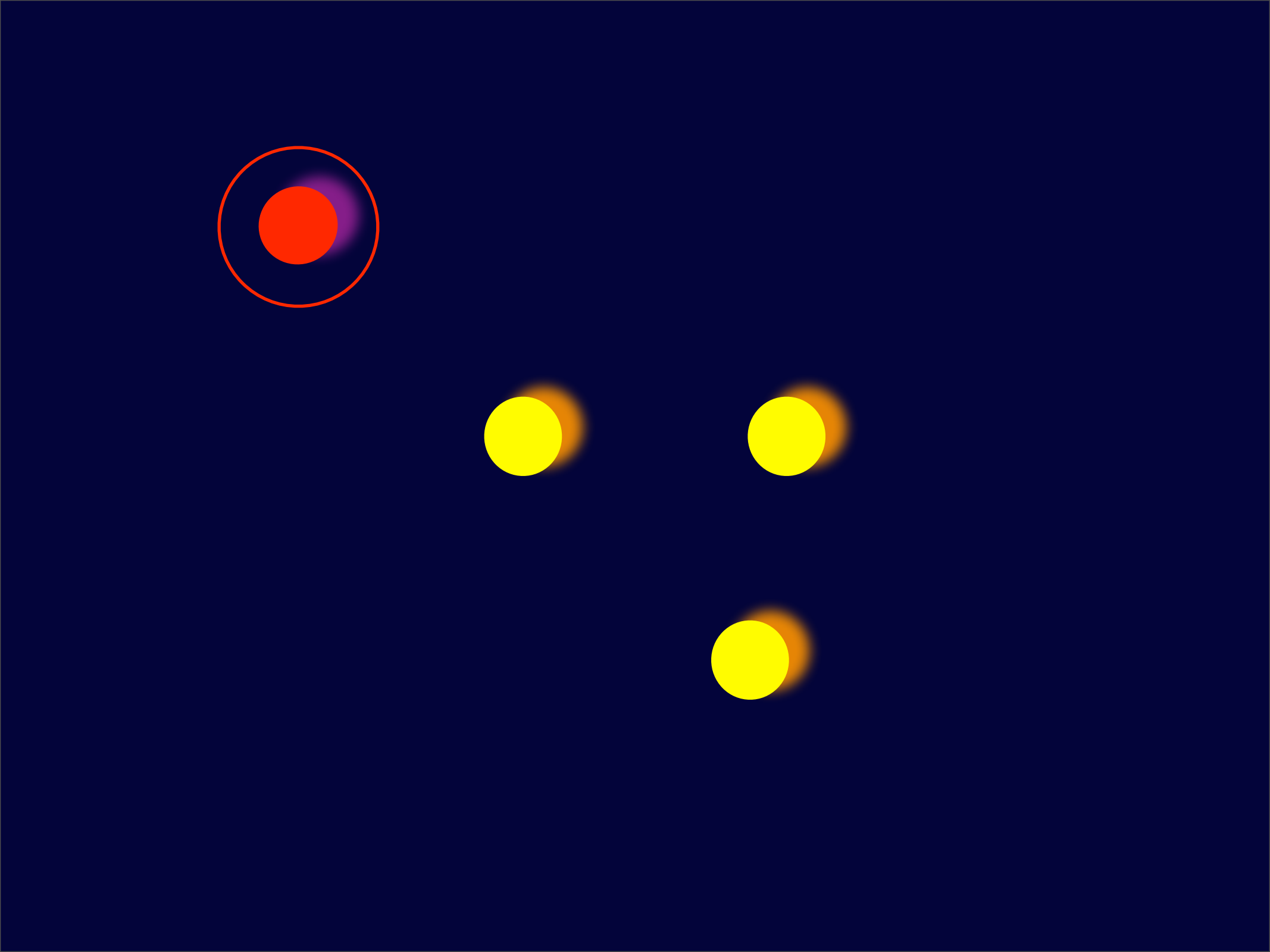
There are two point of views for the SFM problem:

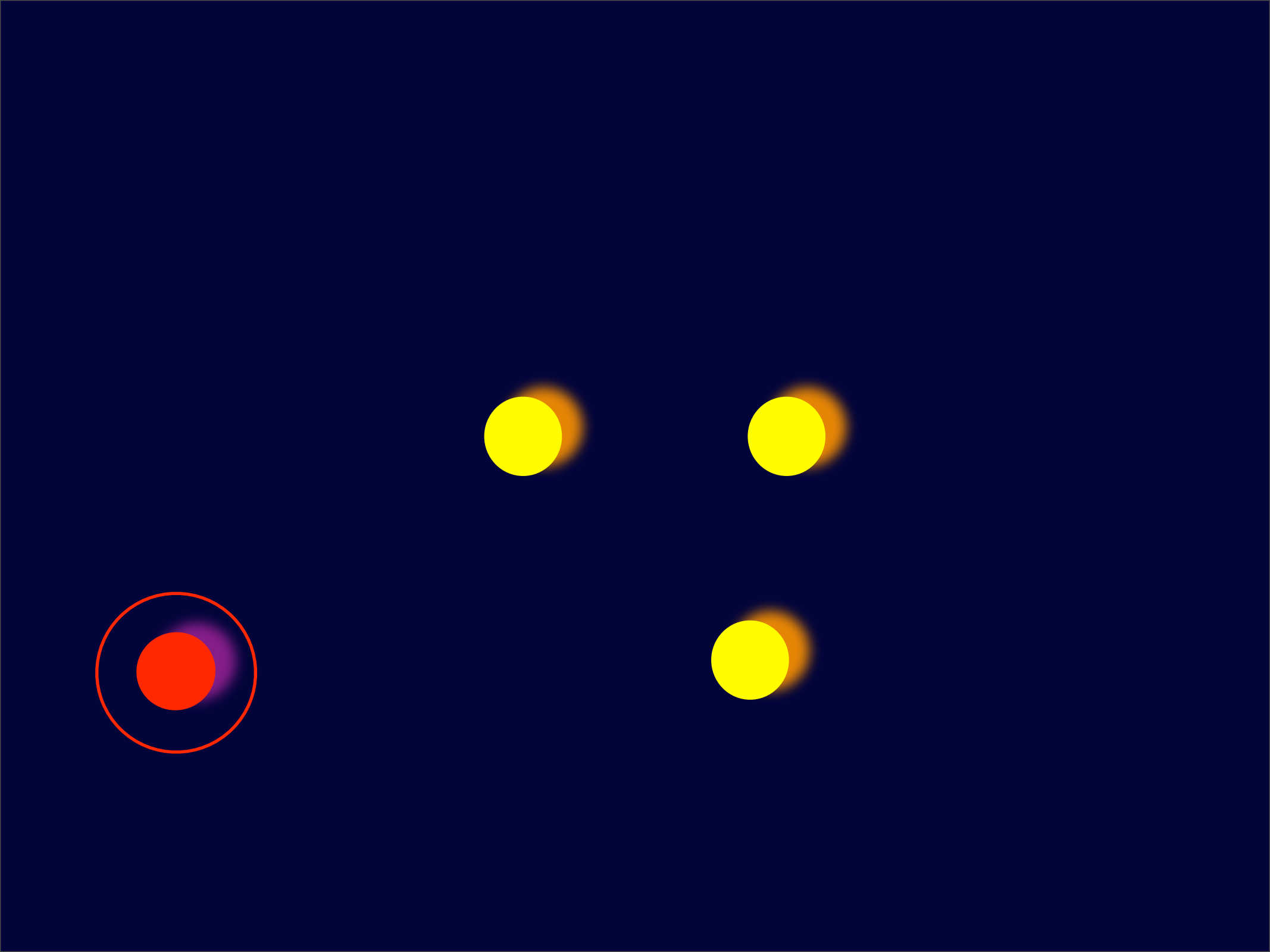
n points
m cameras



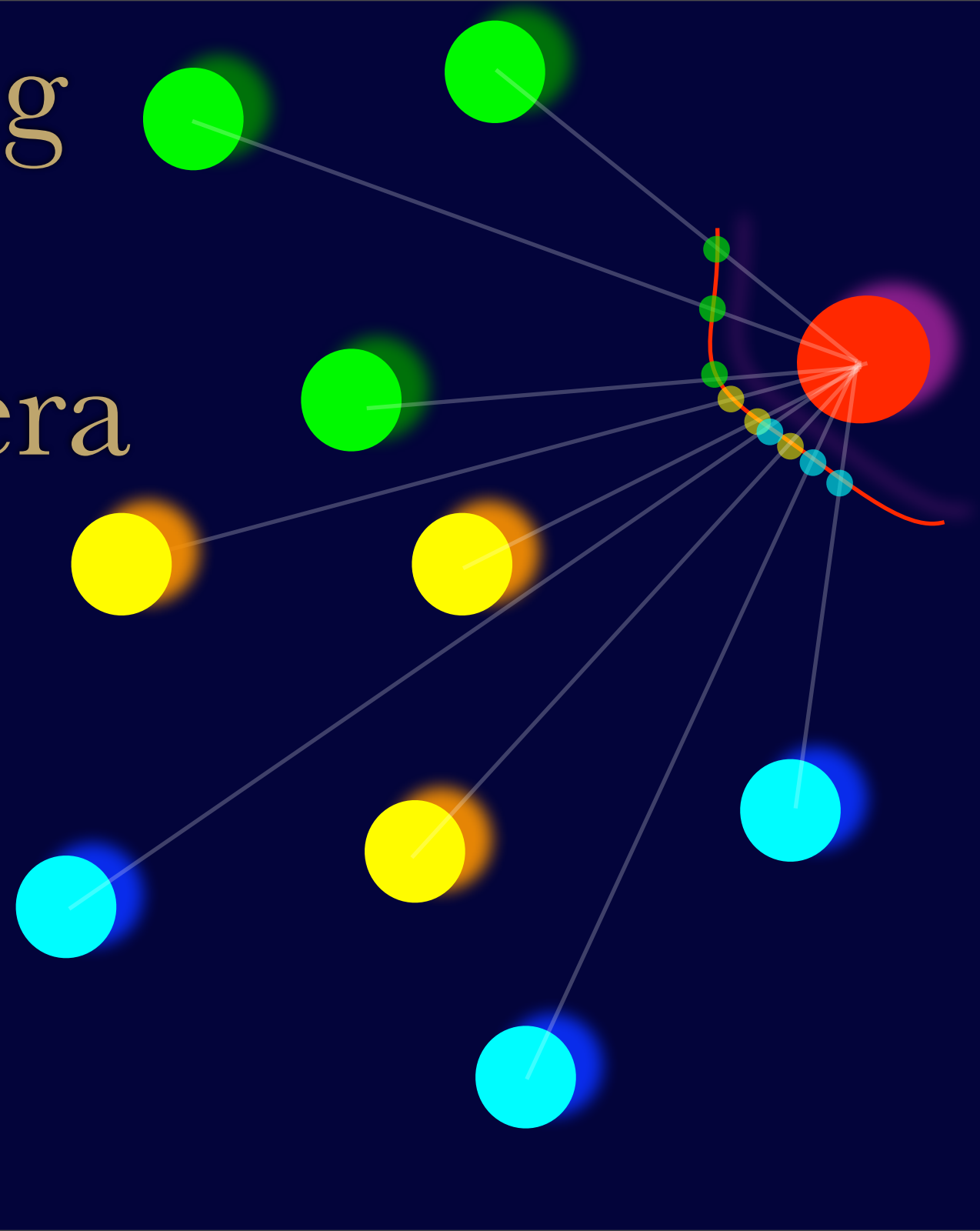


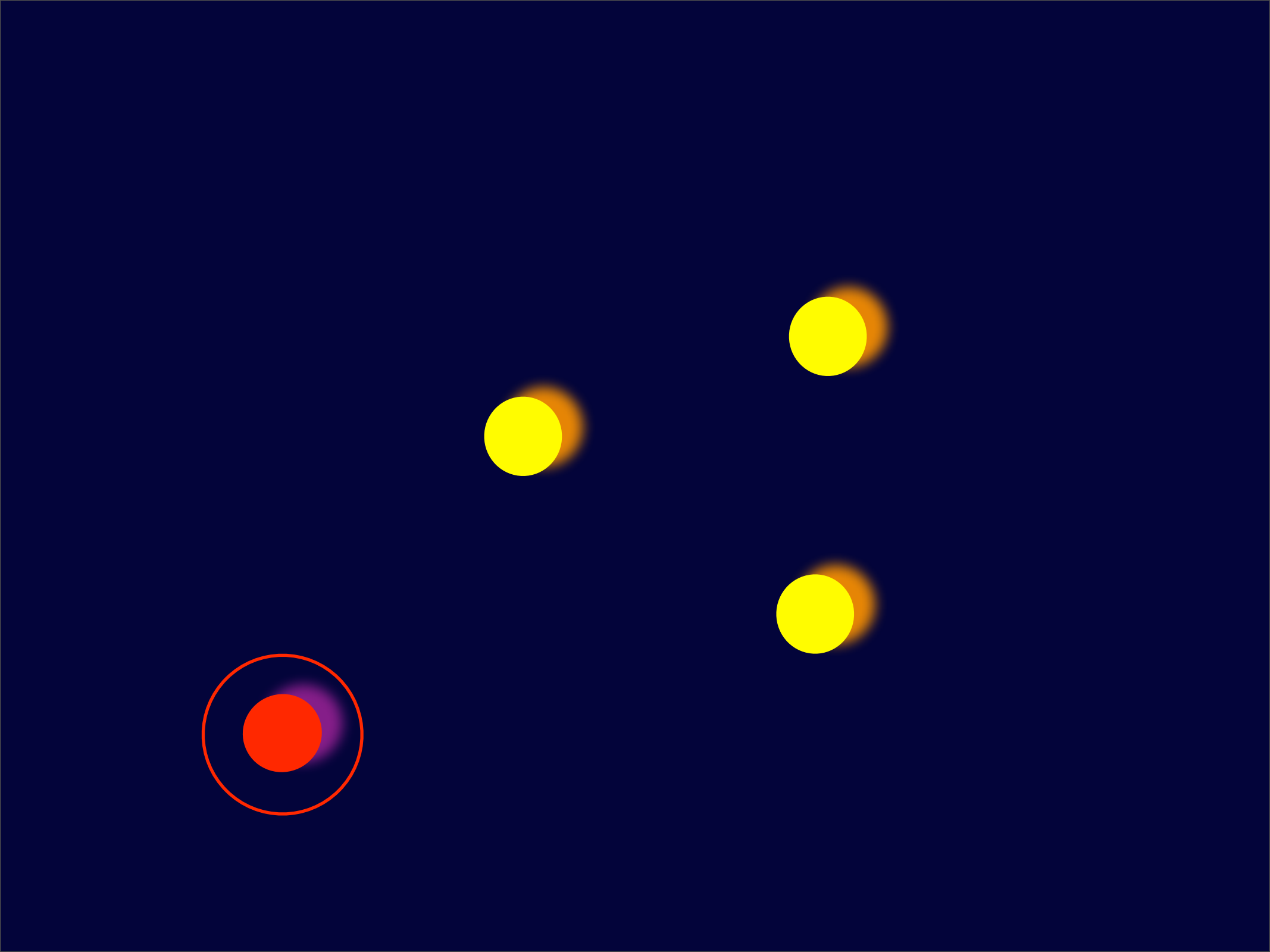


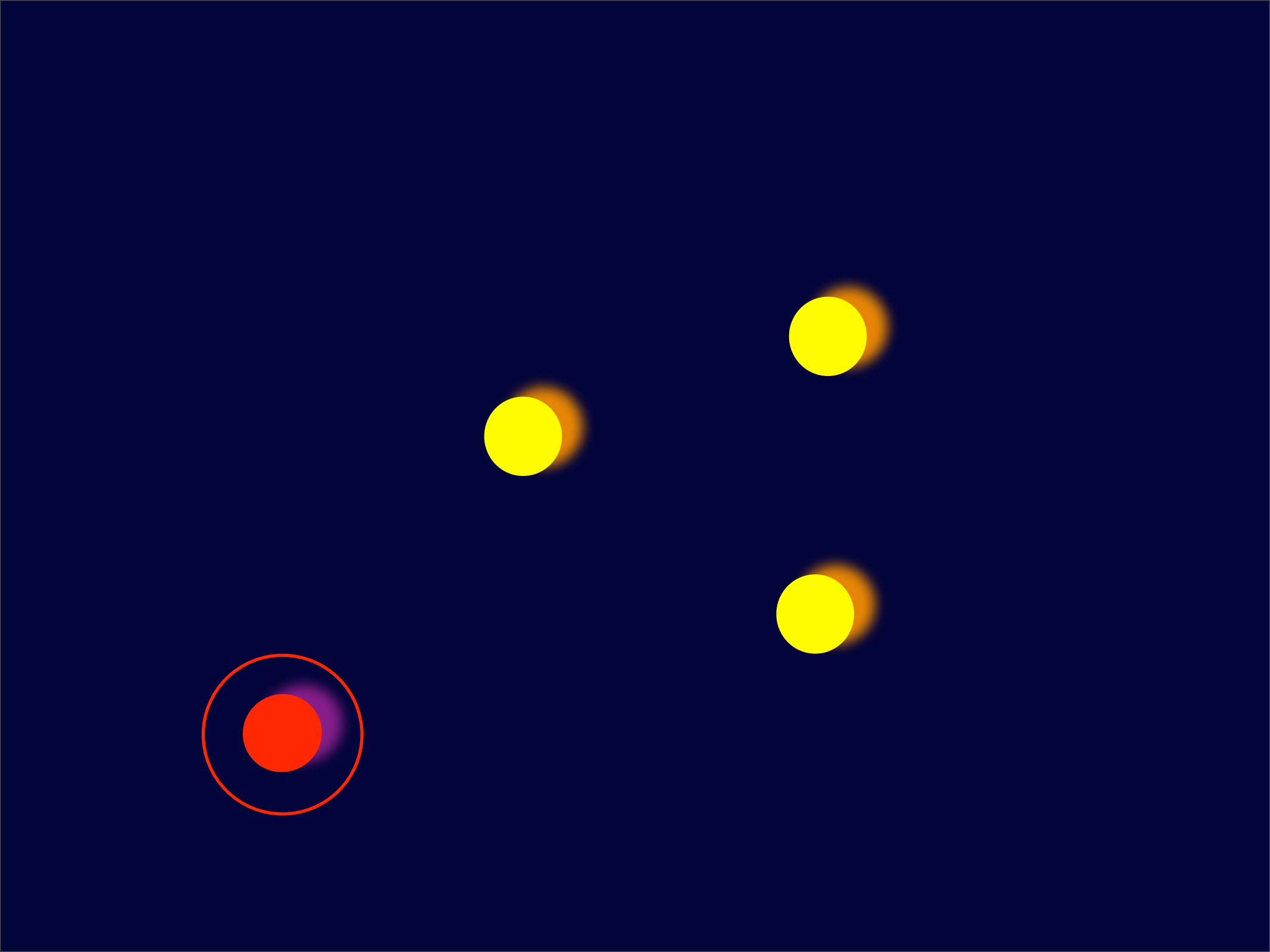




n moving
points
one camera







Reconstruction with many points and cameras

$$\sin(\alpha_{11})(x_1 - a_2) = \cos(\alpha_{11})(y_1 - b_1)$$

$$\sin(\alpha_{12})(x_1 - a_2) = \cos(\alpha_{12})(y_1 - b_2)$$

$$\vdots$$

$$\sin(\alpha_{nm})(x_n - a_m) = \cos(\alpha_{nm})(y_n - b_m)$$

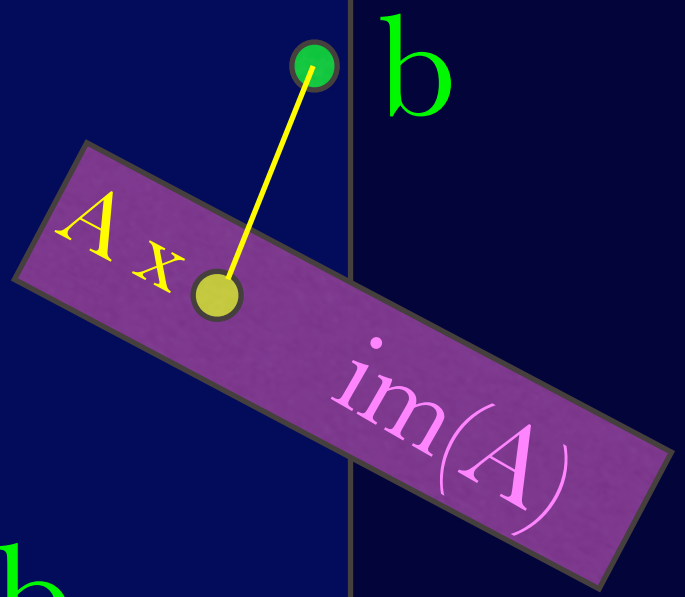
$$x_1 = 0$$
$$y_1 = 0$$
$$x_2 = 1$$

nm+3 equations
2n+2m unknowns

Least square:

$$A\,x = b$$

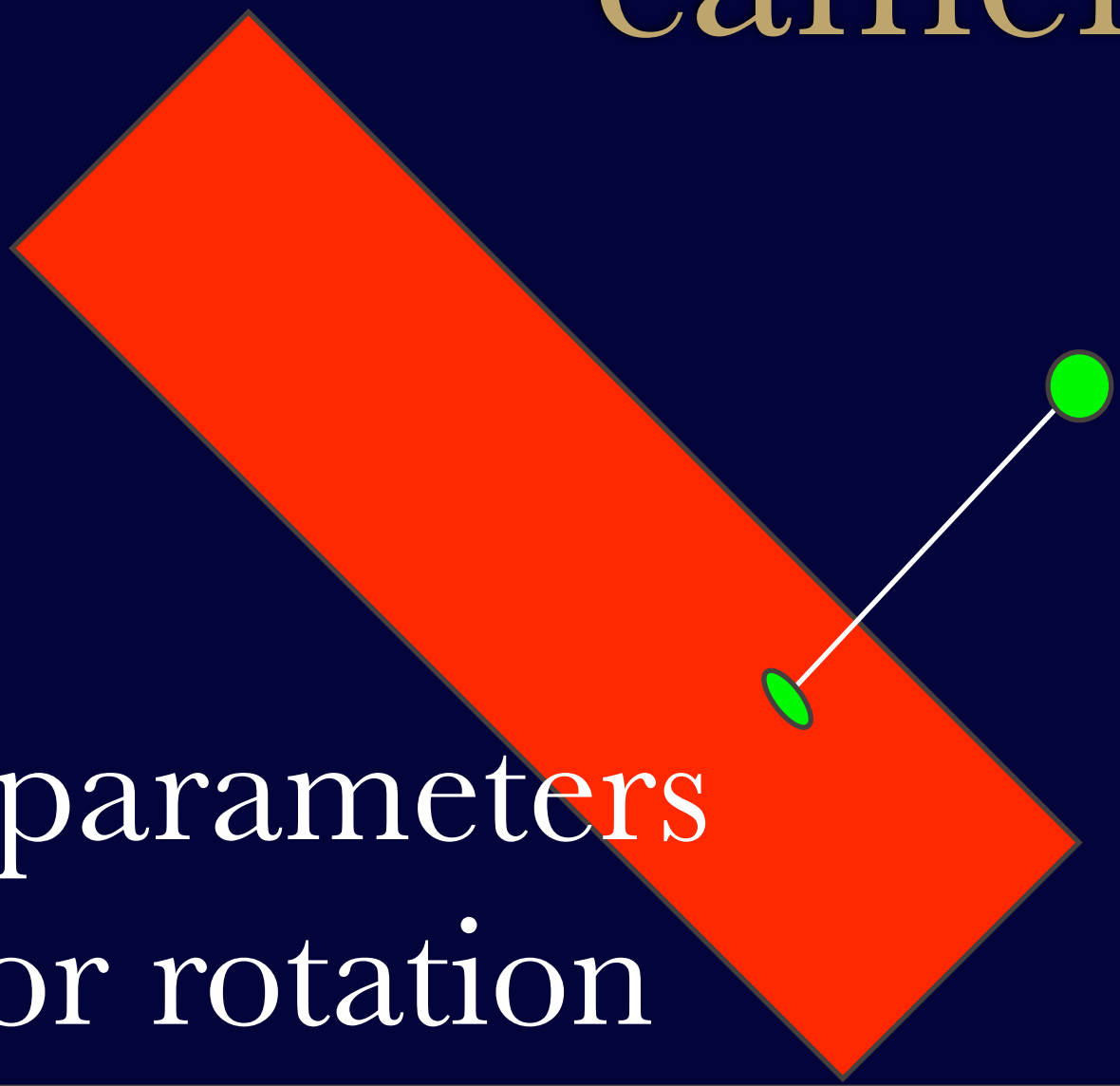
$$x = (A^T A)^{-1} A^T b$$



This finishes the discussion on omnidirectional cameras

Affine orthographic cameras

Orthographic camera

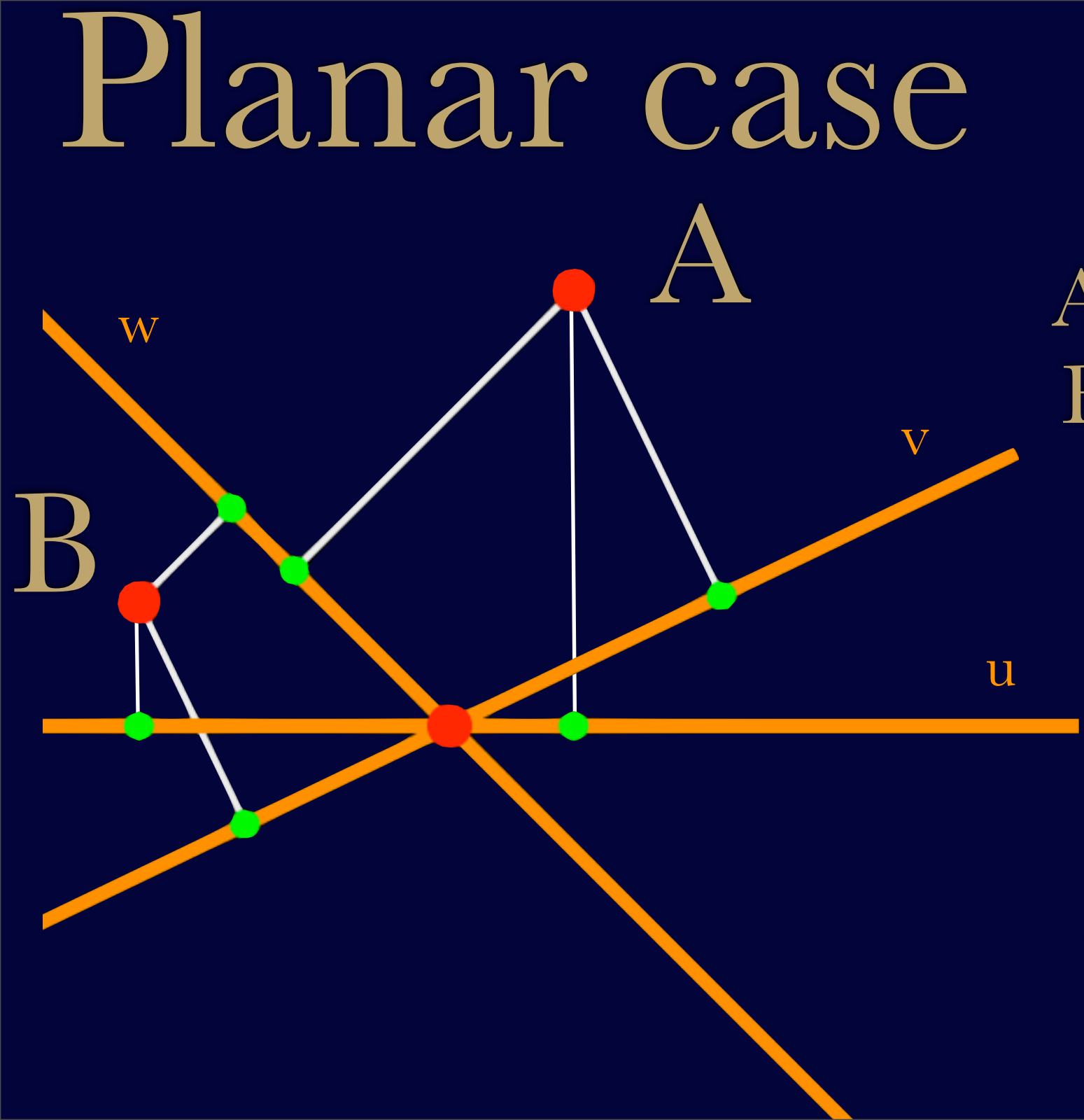


preserves parallelism
is an affine camera.
NO definite location
of the camera!

3 parameters
for rotation

Have 3 cameras
How many
points
are needed?

Planar case



know

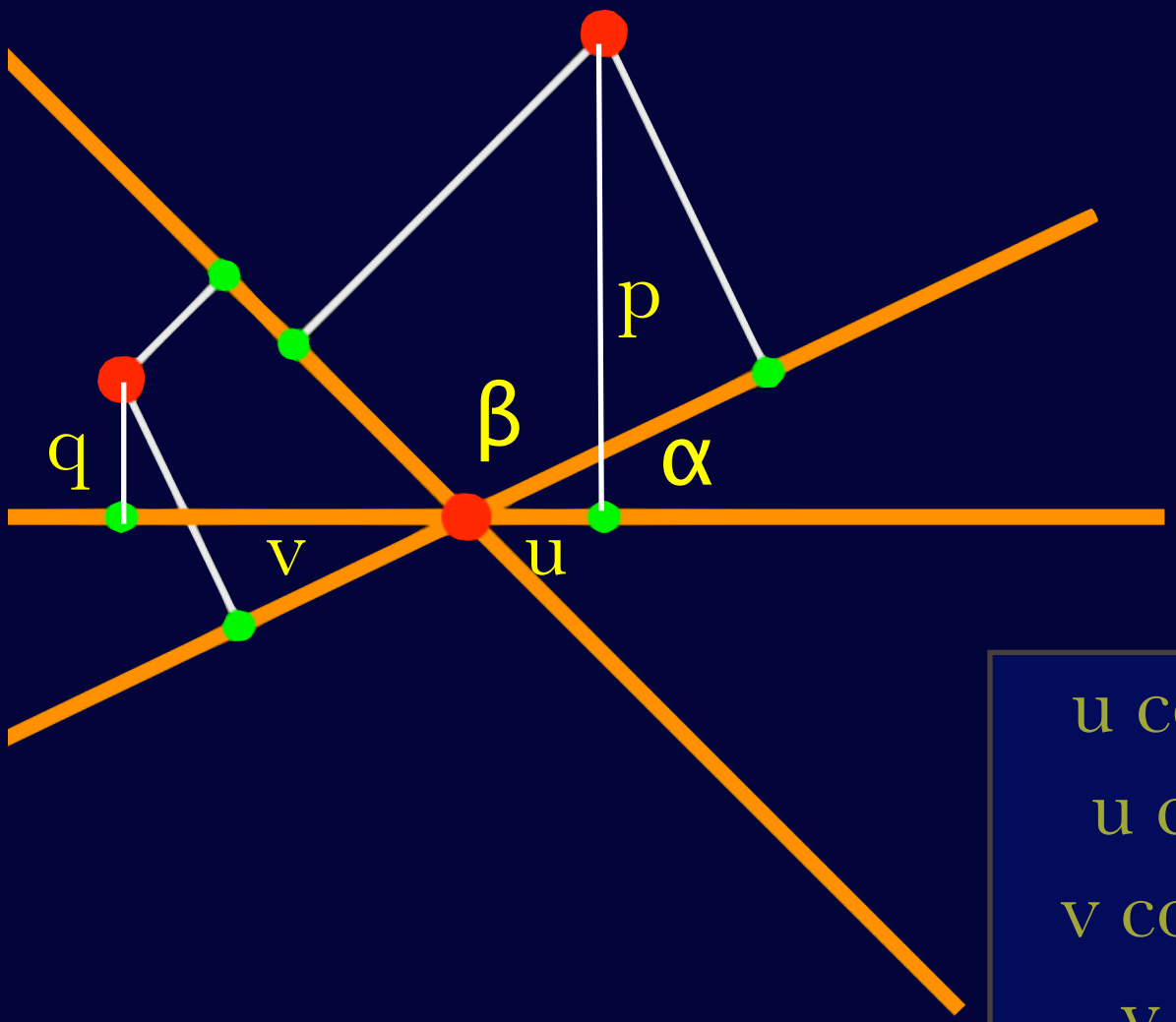
A.u A.v A.w

B.u B.v B.w

find

u,v,w,A,B

Equations



solve for p,q, α, β

$$
\begin{aligned}
u\cos(\alpha) + p\sin(\alpha) &= a \\
u\cos(\beta) + p\sin(\beta) &= c \\
v\cos(\alpha) + q\sin(\alpha) &= b \\
v\cos(\beta) + q\sin(\beta) &= d
\end{aligned}
$$

Nonlinear map

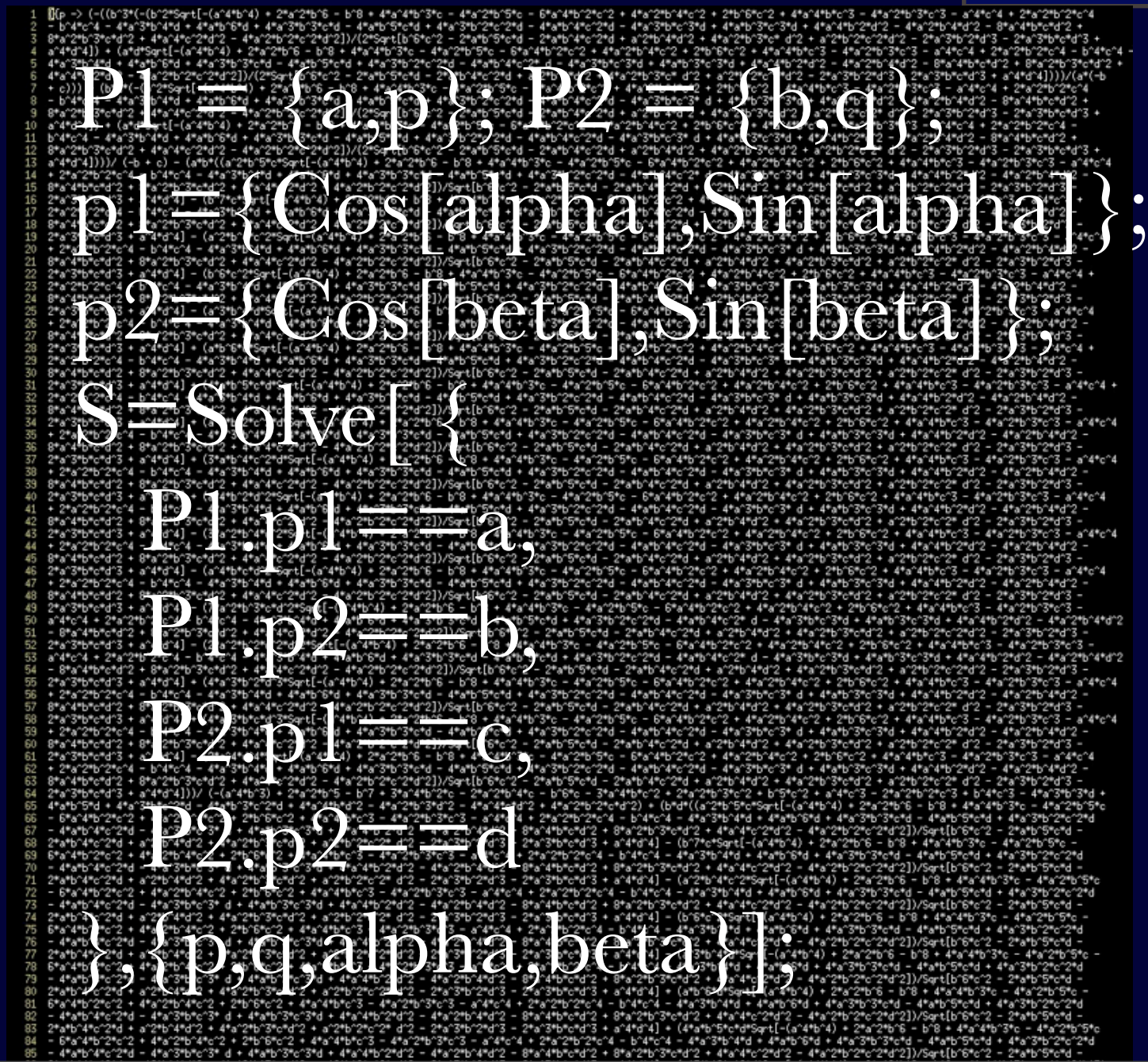
$$
\begin{aligned}
u\cos(\alpha) + p\sin(\alpha) &= a \\
u\cos(\beta) + p\sin(\beta) &= c \\
v\cos(\alpha) + q\sin(\alpha) &= b \\
v\cos(\beta) + q\sin(\beta) &= d
\end{aligned}
$$

$$F(x) = y$$

det(DF(x)) not zero assures locally unique solution

Inversion

$$y = F^{-1}(x)$$



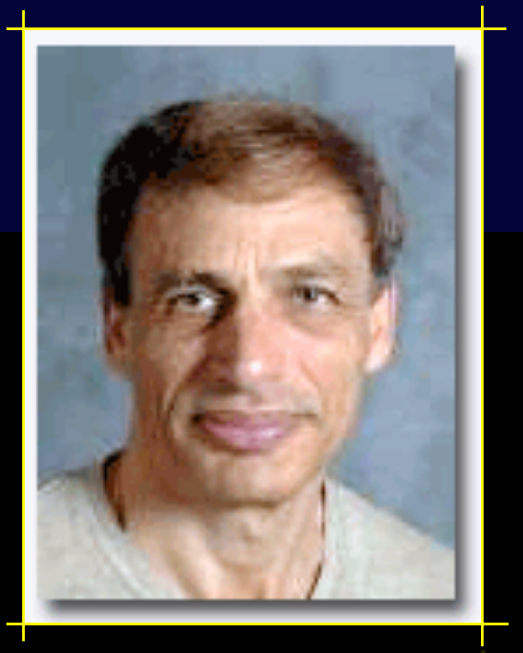
```
P1={a,p};P2={b,q};
p1={Cos[alpha],Sin[alpha]};
p2={Cos[beta],Sin[beta]};
S=Solve[{
P1.p1==a,
P1.p2==b,
P2.p1==c,
P2.p2==d
},{p,q,alpha,beta}];
```

start of the solution formula computed in mathematica (40'000 summands)

```
P1 = {a,p}; P2 = {b,q};
p1={Cos[alpha],Sin[alpha]};
p2={Cos[beta],Sin[beta]};
S=Solve[ {
  P1.p1==a,
  P1.p2==b,
  P2.p1==c,
  P2.p2==d
},{p,q,alpha,beta}];
```

start of the solution formula computed in mathematica (40'000 summands)

3 cameras 4 points



193

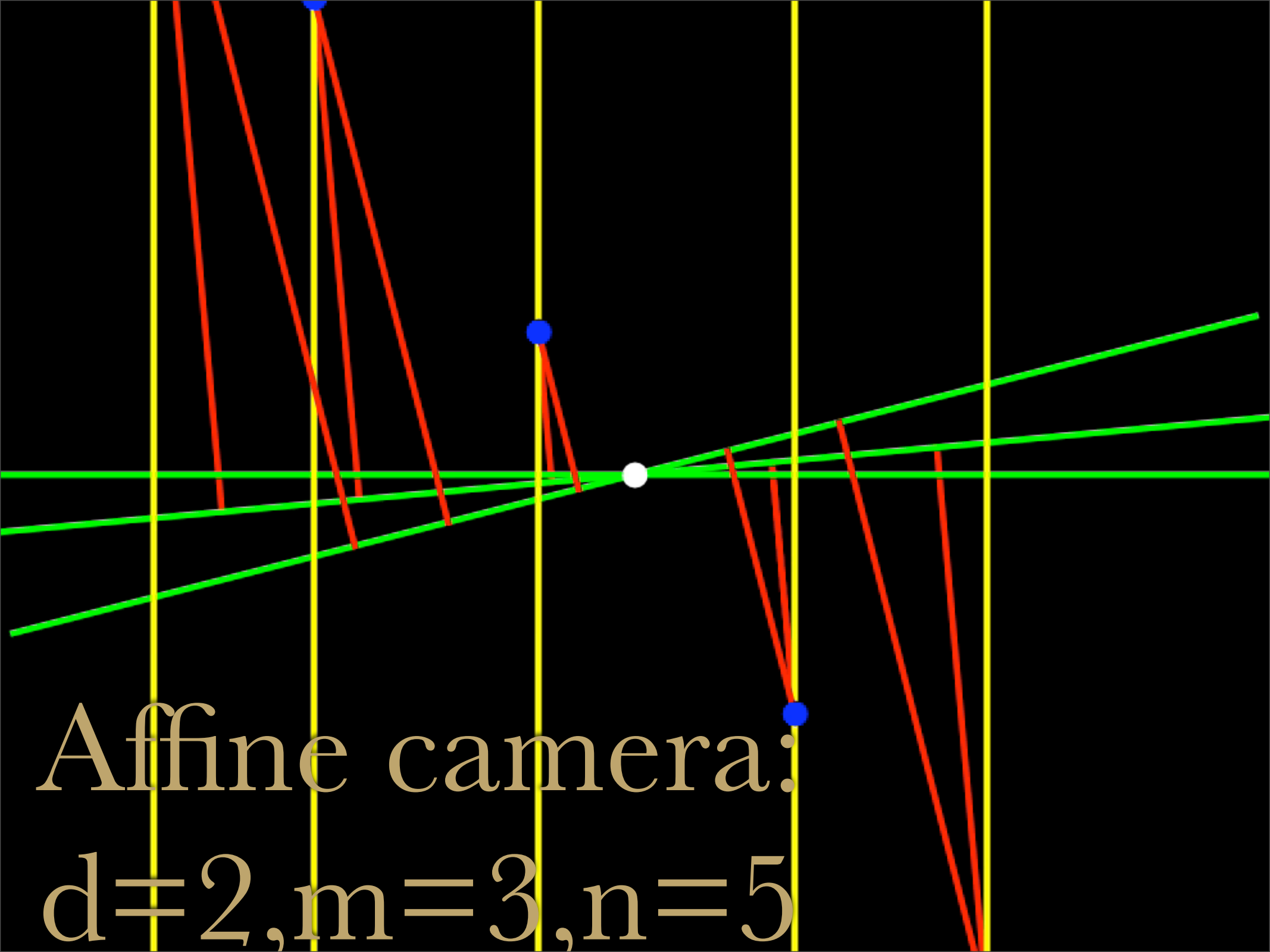
APPENDIX 1

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

Theorem (Knill-Ramirez)

(Addition to Ullmans theorem)

Both in space and in the plane, three cameras and three points allow a

locally

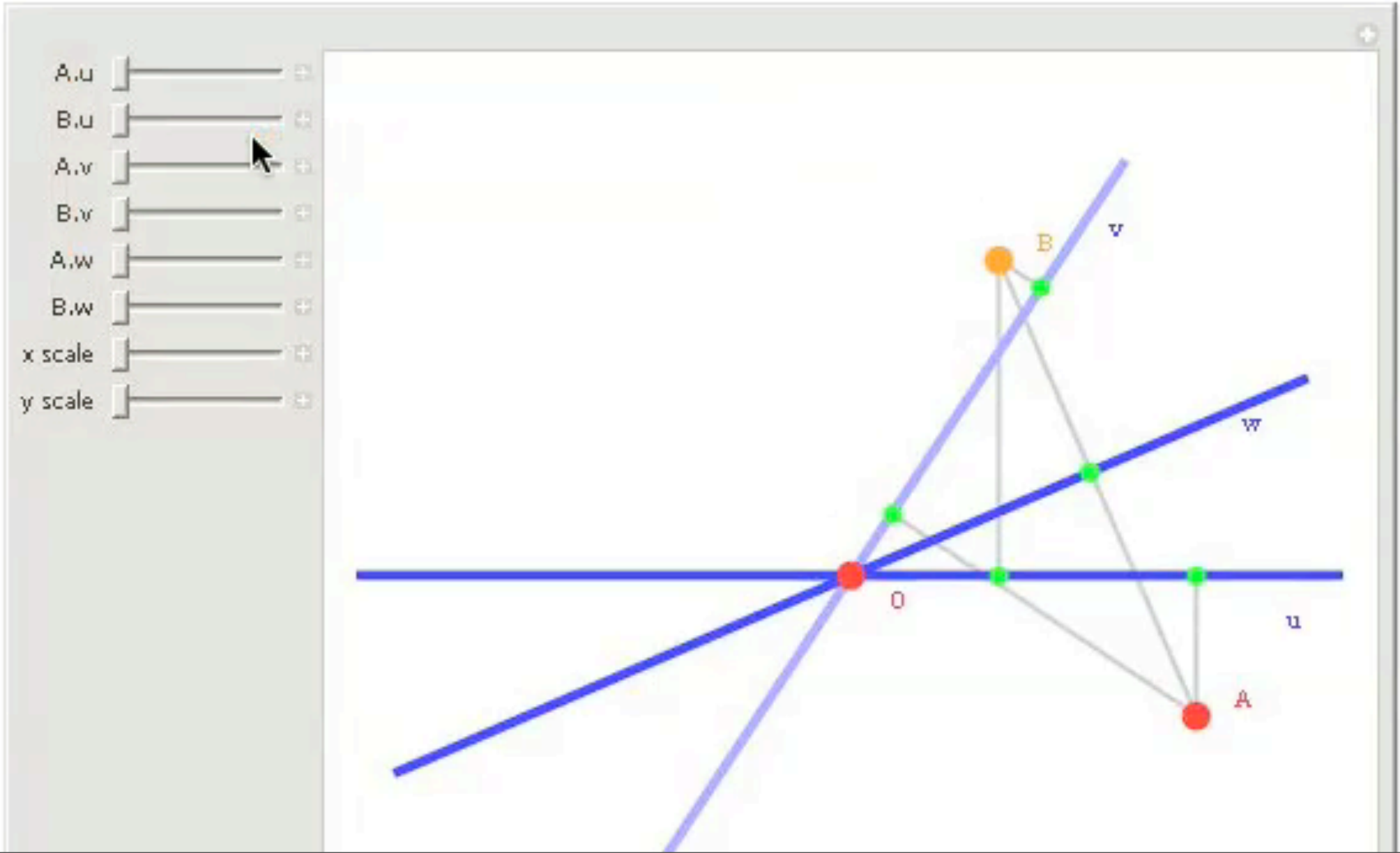
unique reconstruction.

Ullman theorem assumes 4 points and gets uniquness up to a reflection, with 3 points we have uniqueness modulo finitely many reflections.

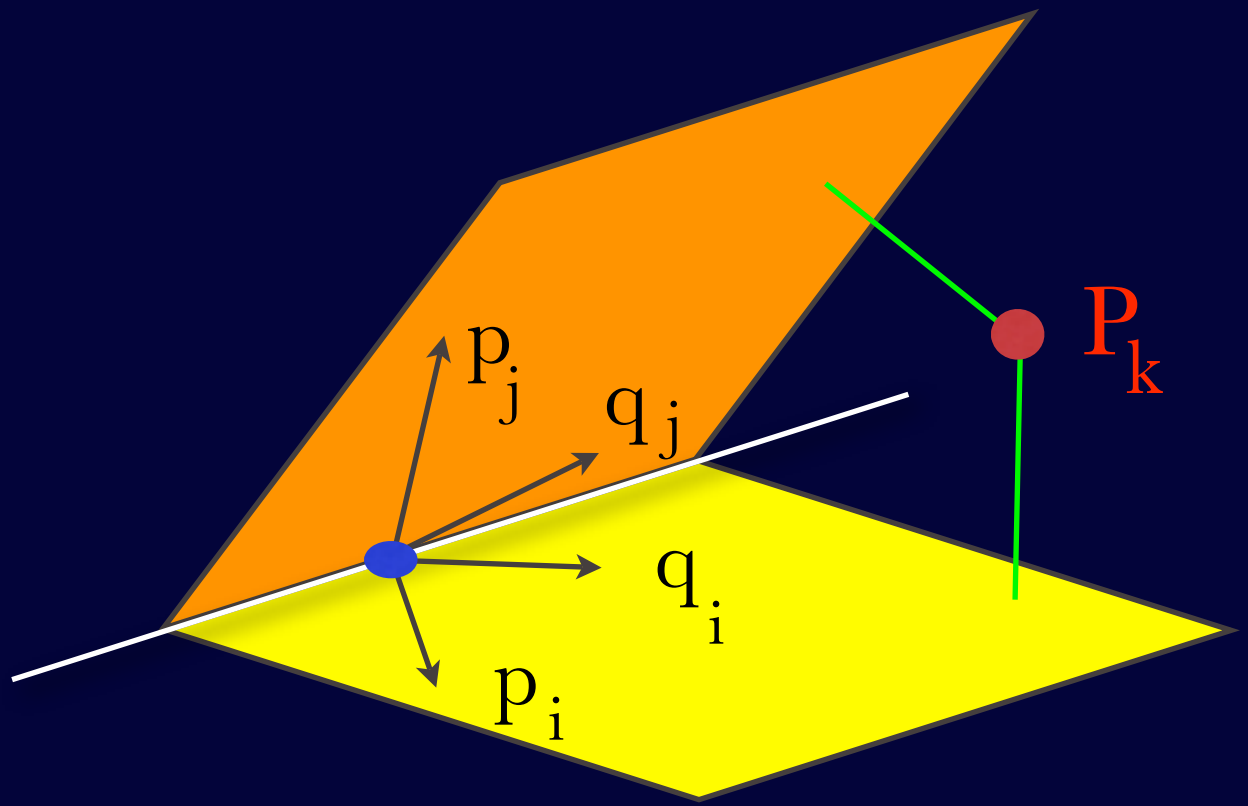
Ullman's Theorem in Two Dimensions

DOWNLOAD LIVE VERSION >>

watch web preview >>



$$\alpha_{ij} p_i + \beta_{ij} q_i = \alpha_{ij} p_j + \beta_{ij} q_j$$



in three dimensions, a new idea is needed.
And this has been done by Ullman.

$$\alpha_{ij} u_{i1} + \beta_{ij} v_{i1} = \gamma_{ij} u_{j1} + \delta_{ij} v_{j1}$$
$$\alpha_{ij} u_{i2} + \beta_{ij} v_{i2} = \gamma_{ij} u_{j2} + \delta_{ij} v_{j2}$$
$$\alpha_{ij}^2 + \beta_{ij}^2 = 1 \quad , \quad \gamma_{ij}^2 + \delta_{ij}^2 = 1$$

This is a system of nonlinear equations for each i,j. Have 4 equations and 4 unknowns. Computer algebra systems have problems solving it, humans not....

In[6]:=

```
Manipulate[FF[cc1,cc2,cc3,cc4,cc5,cc6,cc7,cc8,cc9,cc10,cc11,cc12],{cc1,-0
{cc4,-0.06,0.06},{cc5,-0.06,0.06},{cc6,-0.06,0.06},{cc7,-0.06,0.06},{cc8,
{cc10,-0.03,0.03},{cc11,-0.03,0.03},{cc12,-0.03,0.03}]
```

Out[6]=

cc1

cc2

cc3

cc4

cc5

cc6

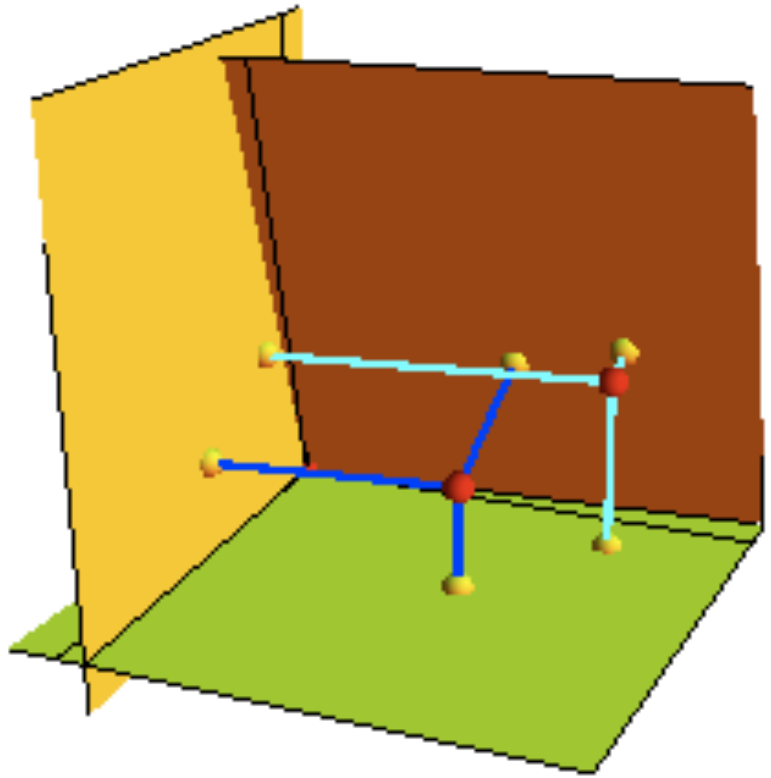
cc7

cc8

cc9

cc10

cc11

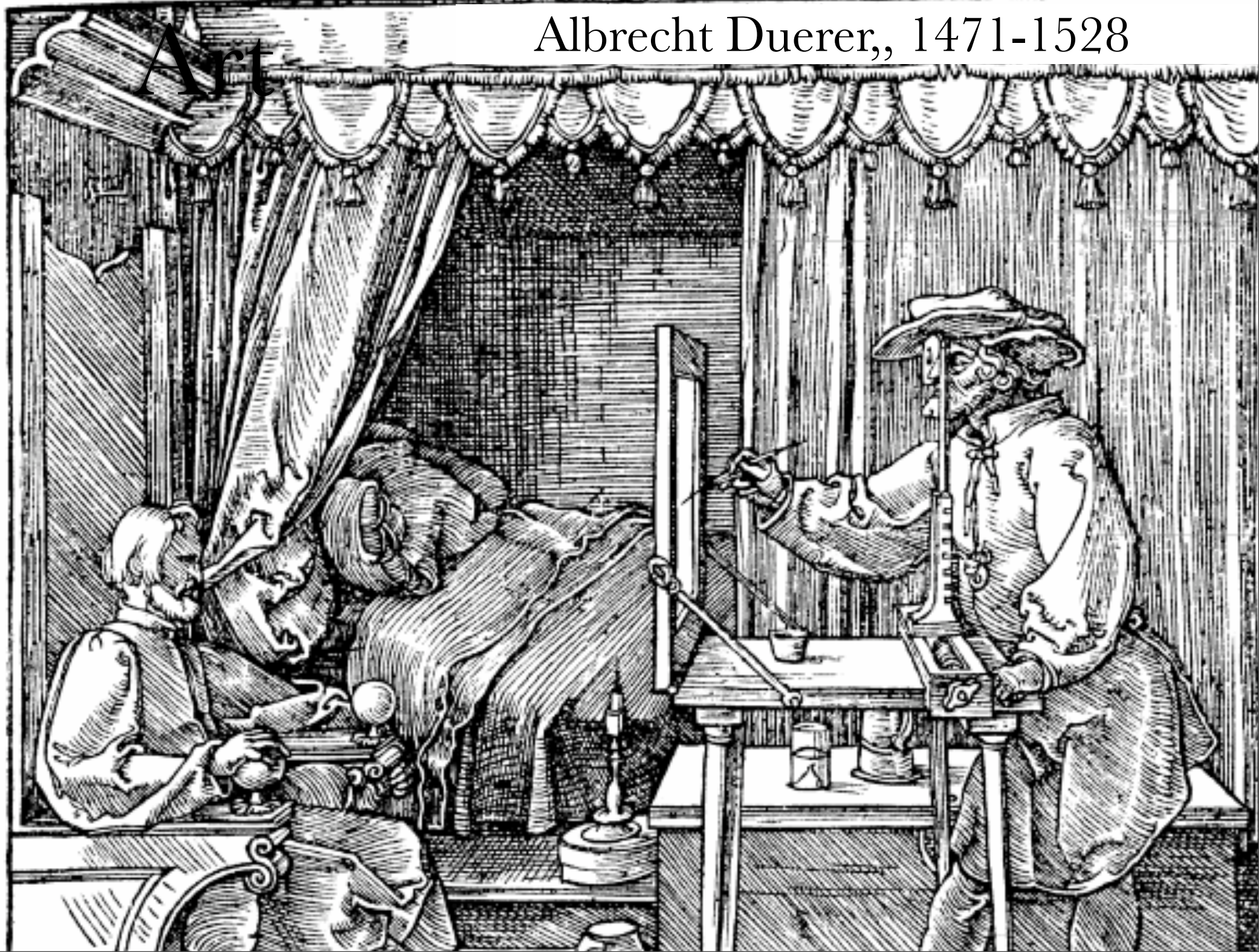


Perspective Cameras

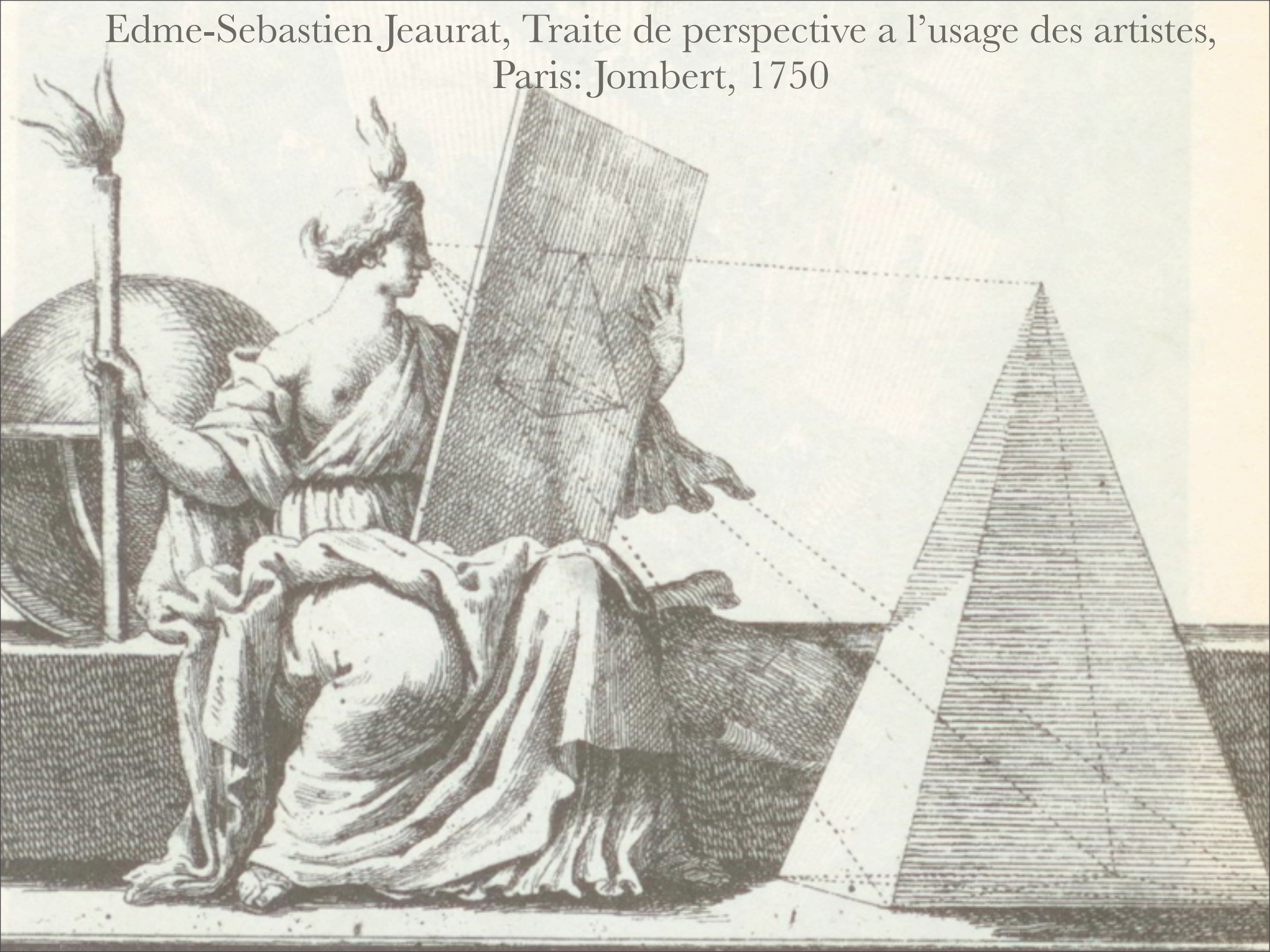
This is the main case considered in the computer vision literature.

Art

Albrecht Duerer,, 1471-1528

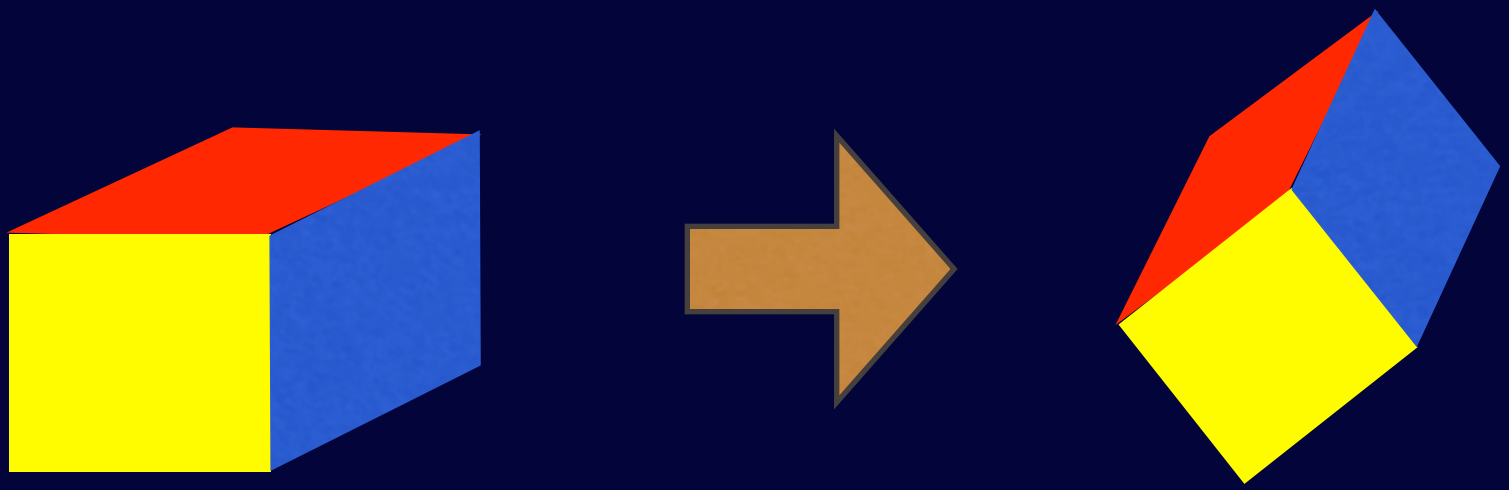


Edme-Sebastien Jeaurat, Traite de perspective a l'usage des artistes, Paris: Jombert, 1750



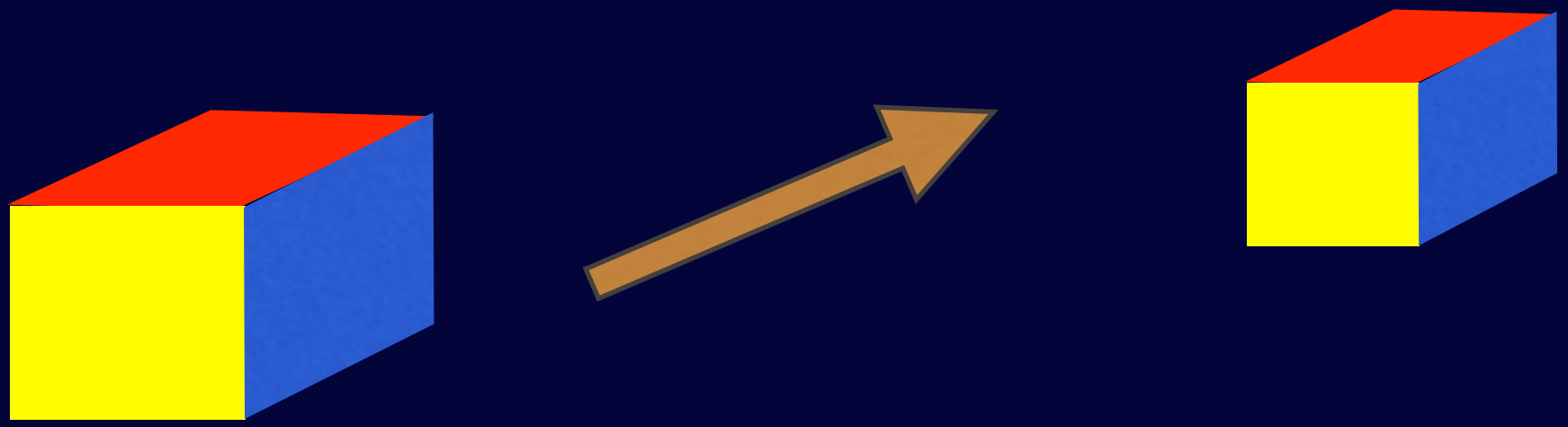
Rotation

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



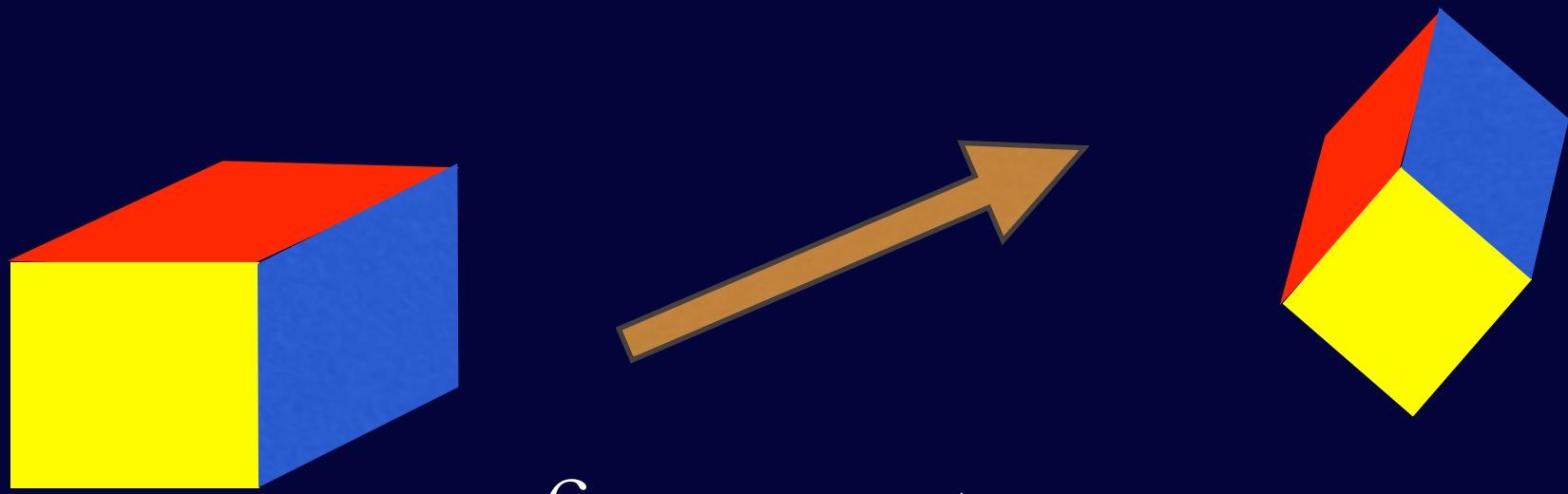
Translation

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



Euclidean transforms

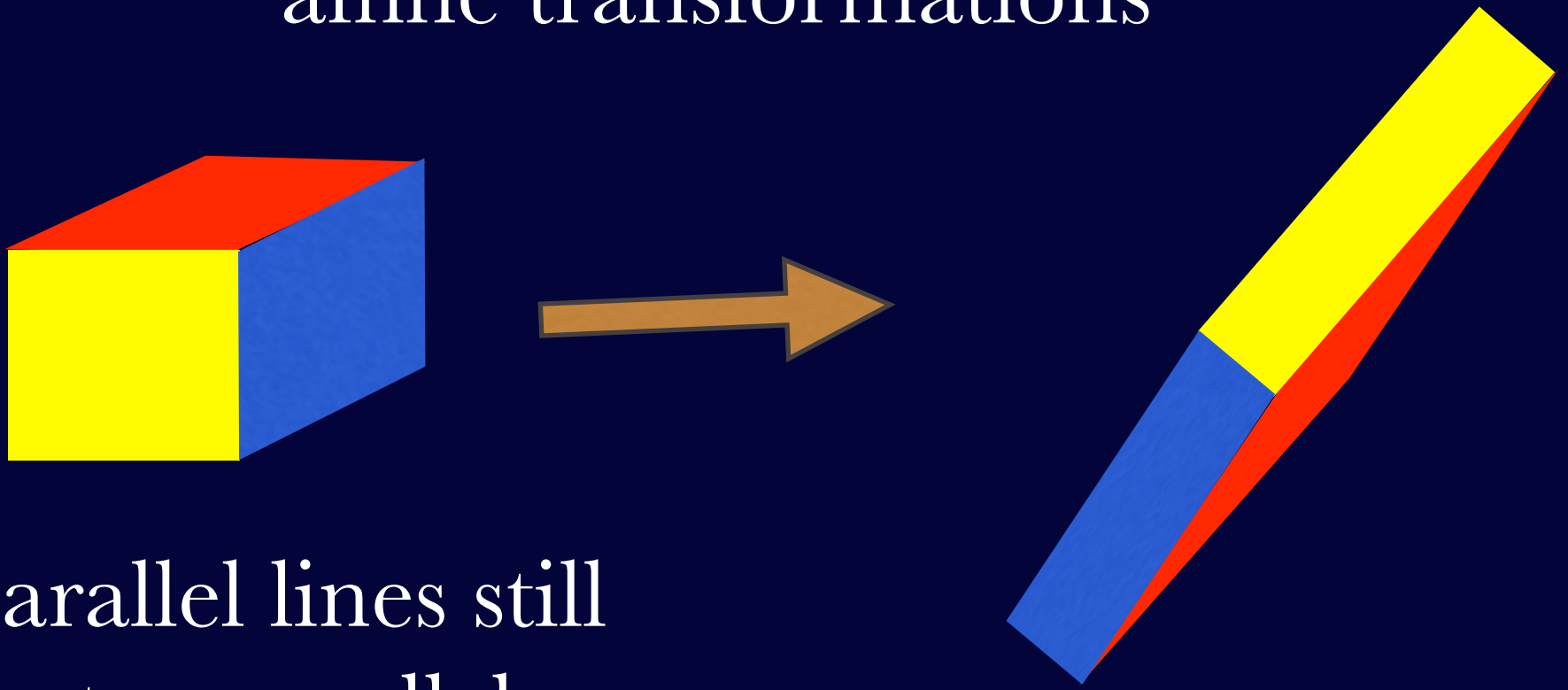
Translations and rotations generate the group of Euclidean transformations



6 parameters

Affine transforms

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

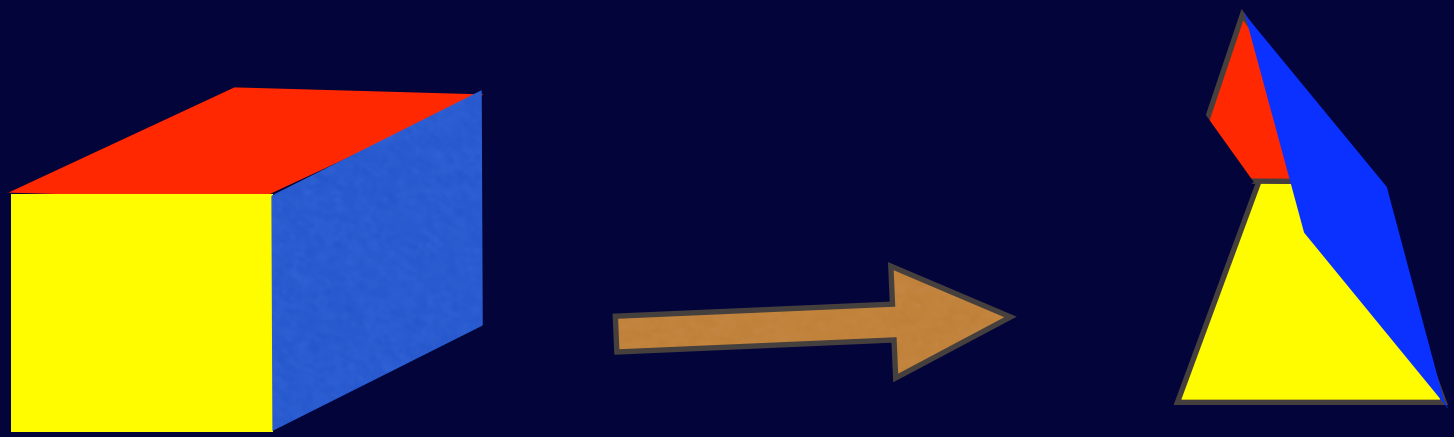


parallel lines still
stay parallel

12 parameters

Projective transforms

We even allow perspective transformations



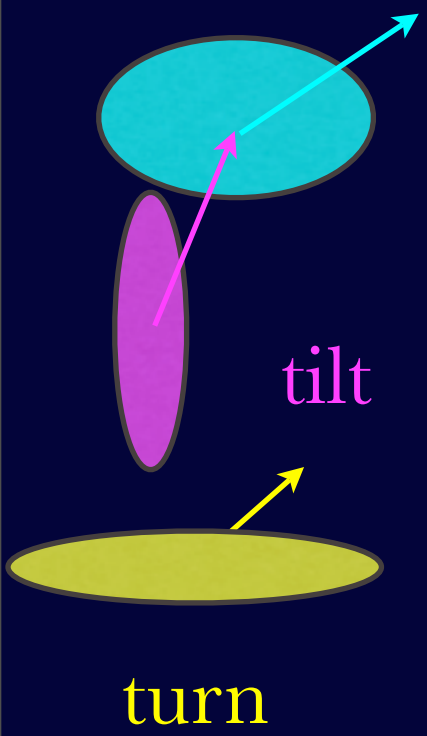
parallel lines stay
no more parallel
in general

15 parameters

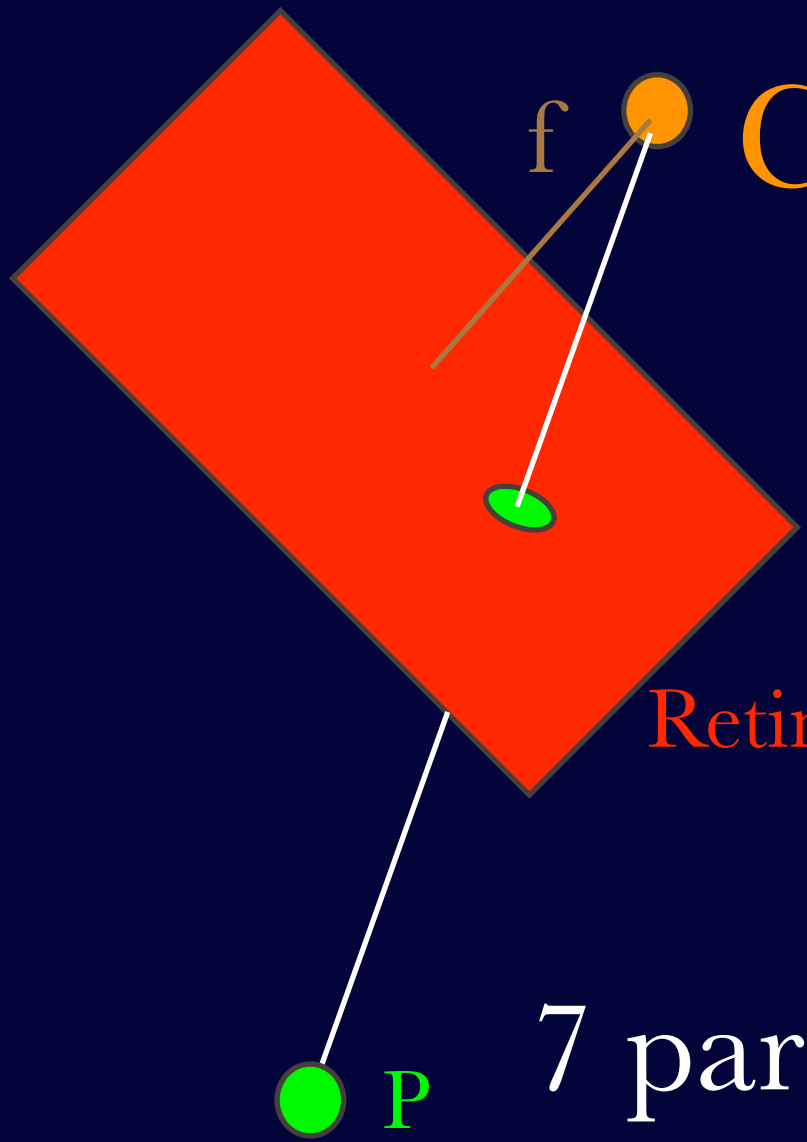
Traditional camera

translate

3



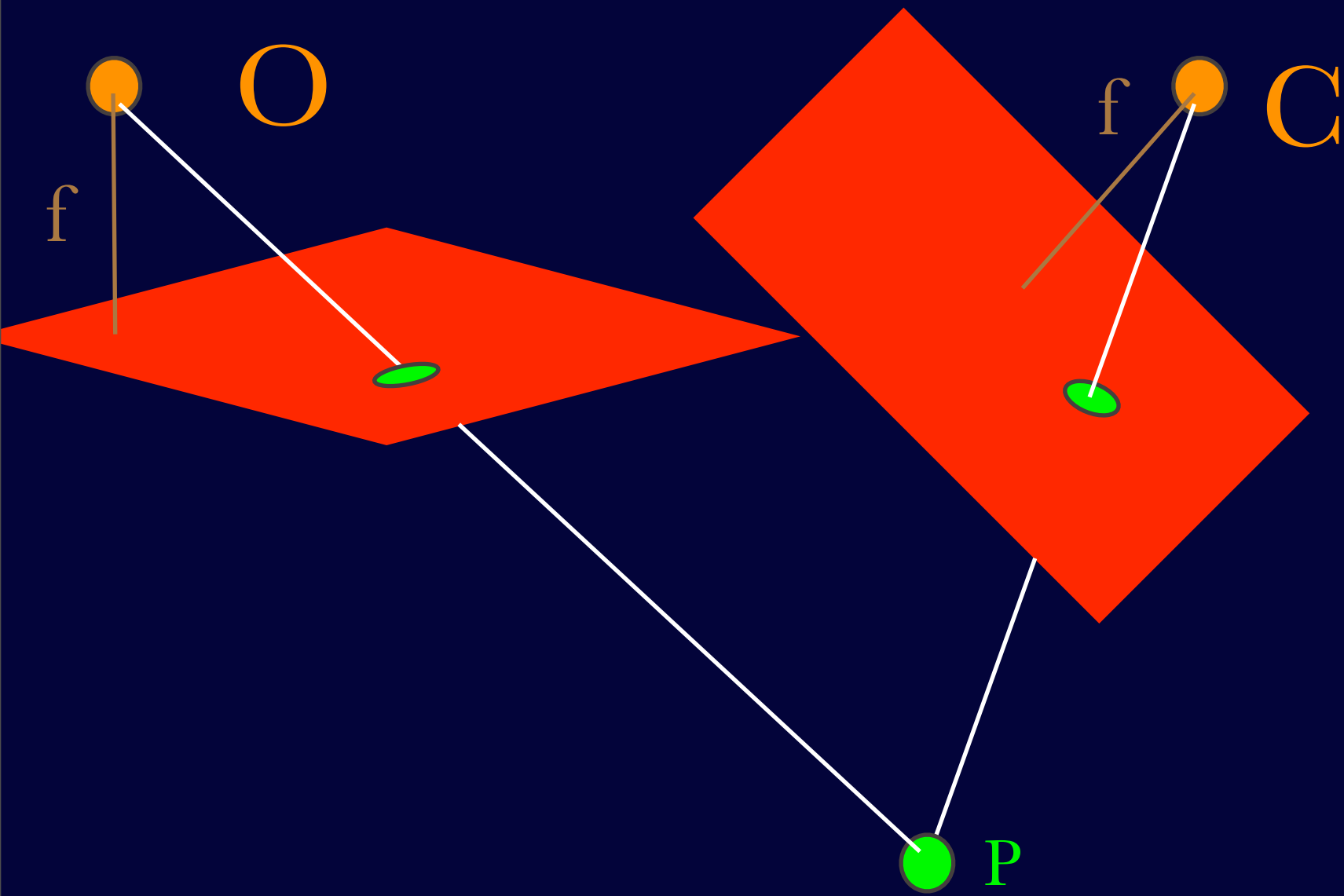
3



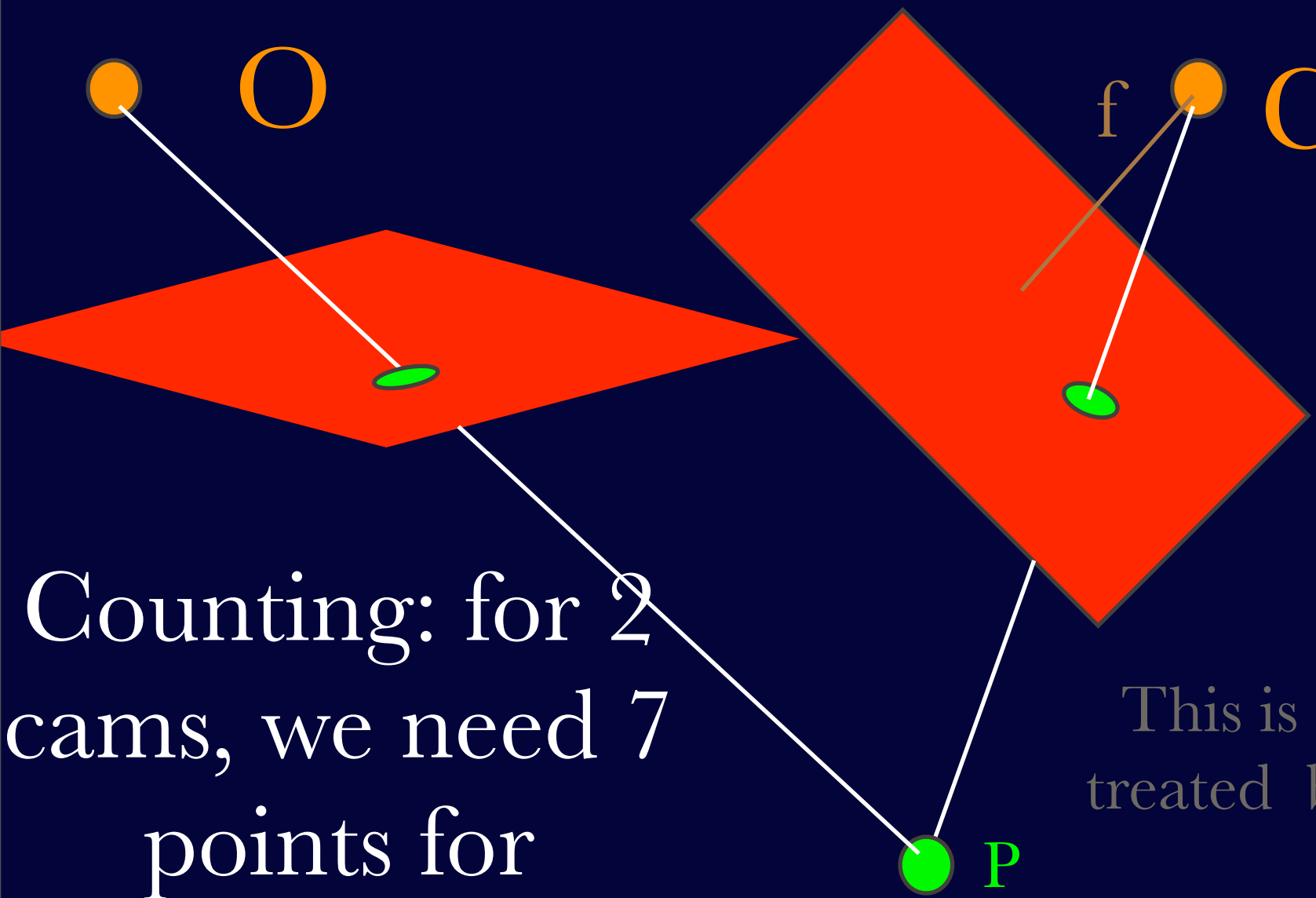
1

7 parameters

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

The general problem

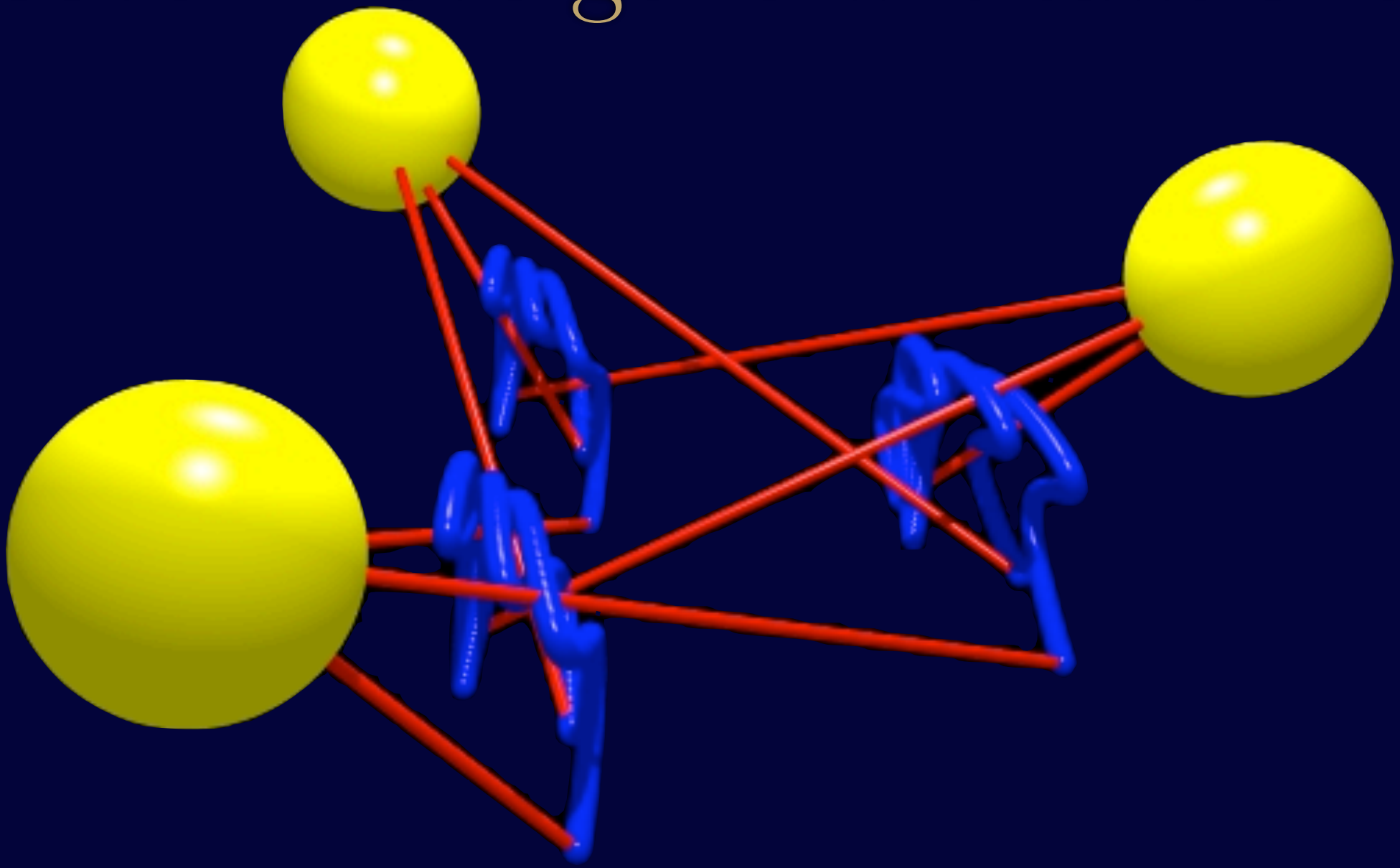
Space is a manifold N. A camera is a map Q: N -> N where Q(N) is a lower dimensional surface diffeomorphic to S and $Q^2 = Q$.
Given a manifold M of cameras.

$P_1, P_2, \ldots, P_n$ points in N
$Q_1, Q_2, \ldots, Q_m$ cameras in M

Reconstruct points and cameras from the data matrix $Q_j(P_i)$.

General critics on mathematics: Why such f&@#$*%o generality?

cover move general situations



The SFM inequality

d dimension of space

n number of points

f number of camera parameters

m number of cameras

h global camera parameters

s dimension of retinal surface

g global symmetries

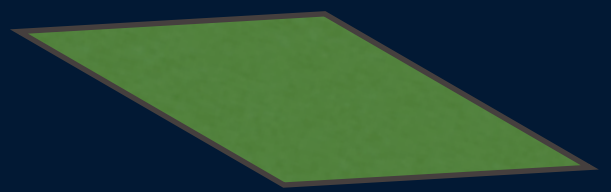
$$d\,n + f\,m + h \leq s\;\; n\,m + g$$

Example: orthographic camera

d=3, m=3, n=3, s=2

f=5, h=0

3 angles, 2 translations in plane



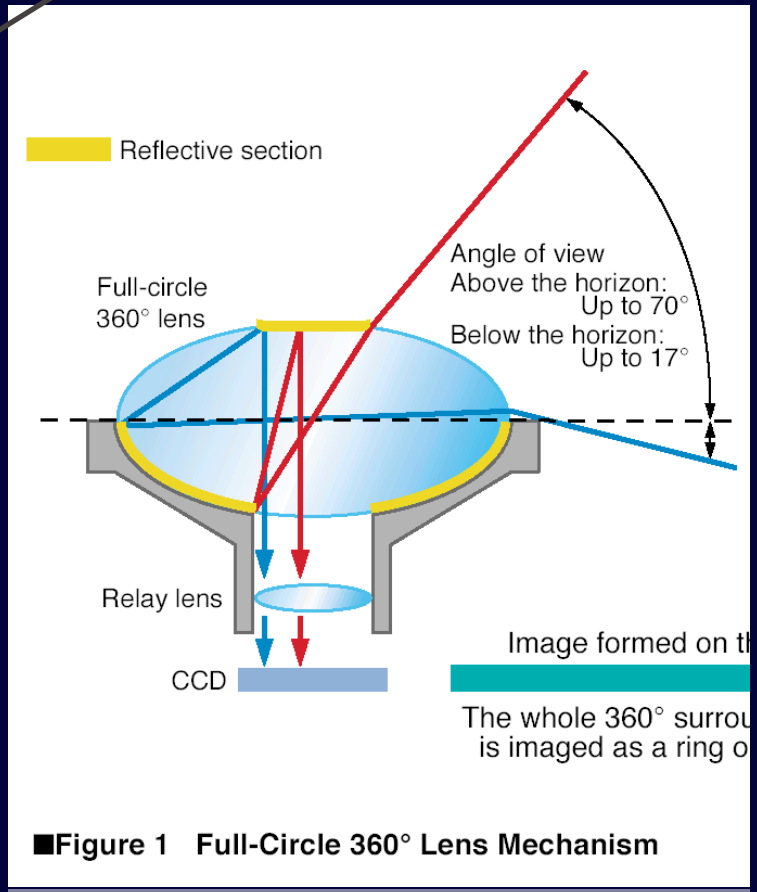
g=6: translations and rotations

Panorama Photography

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

Catadioptric systems

0-360 panoramic optic



■Figure 1 Full-Circle 360° Lens Mechanism

Sony full circle

id mind

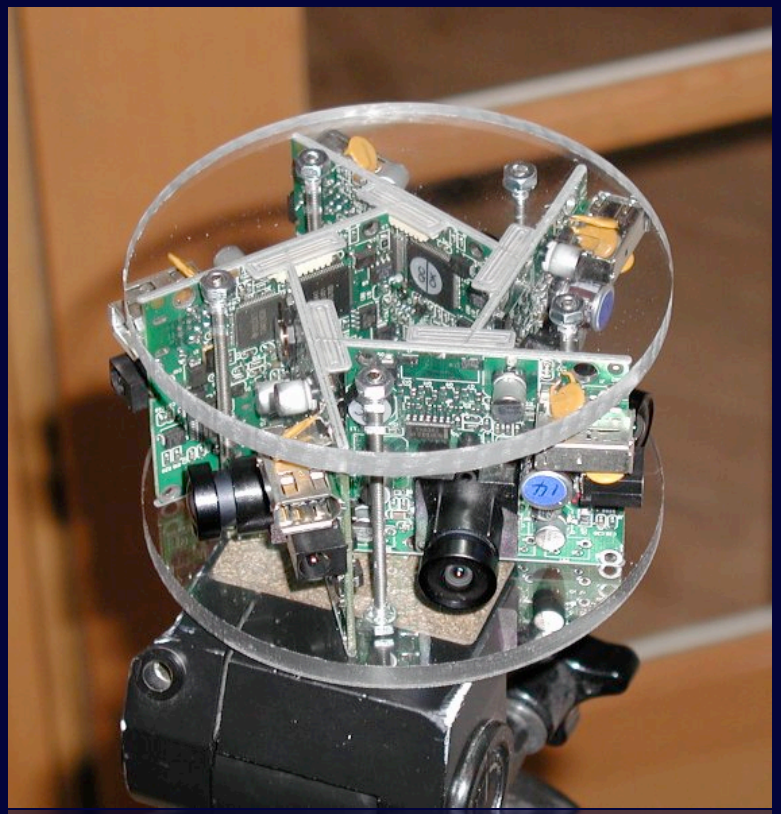
crucial: vertical field of view

Example

Arlington
jan 2006

Polydioptric systems

Microsoft



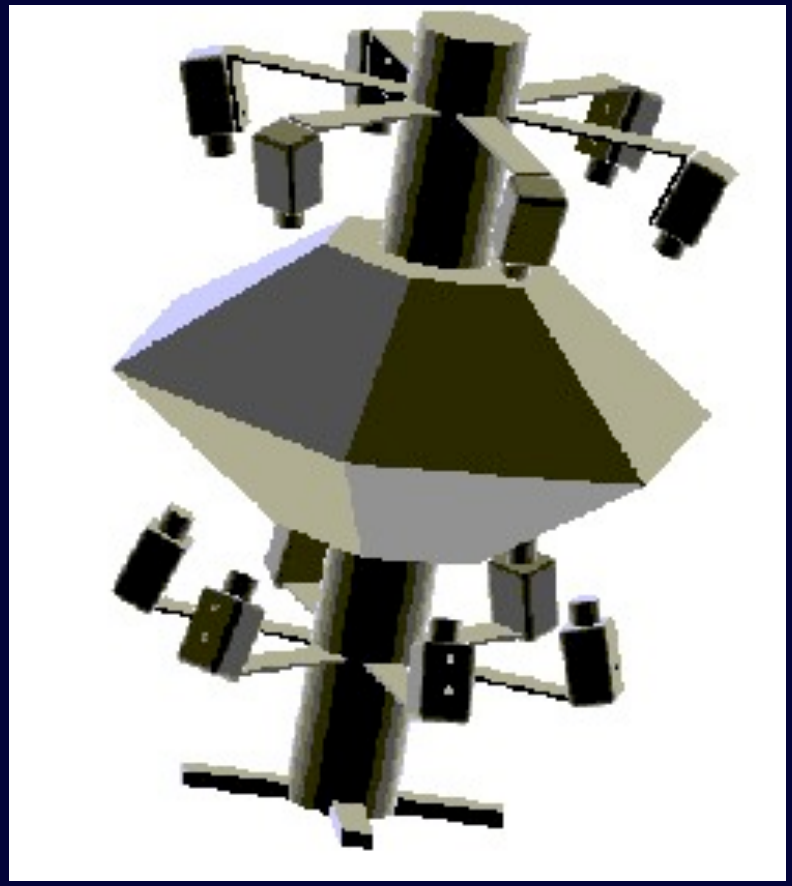
ring cam

google

U-Penn

Sphere cam

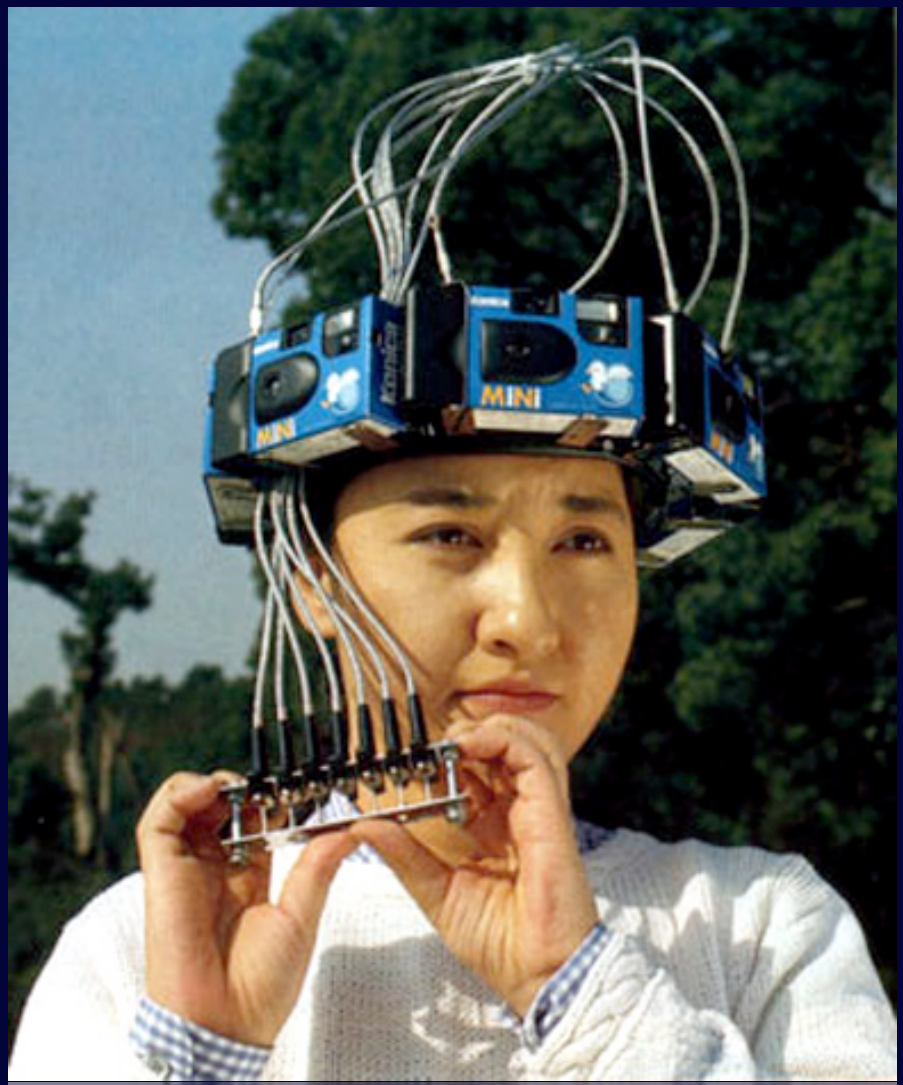
Mixed systems



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


2006 www.ephkleeman.com

Google street map


19th Ave
S
N


19th Ave
S
N


S
N


N


N
S


N
19th Ave
S


19th Ave
N
S


N
19th Ave

Last, but
not at least:

Omni cameras in nature:

damselfly


dragonfly

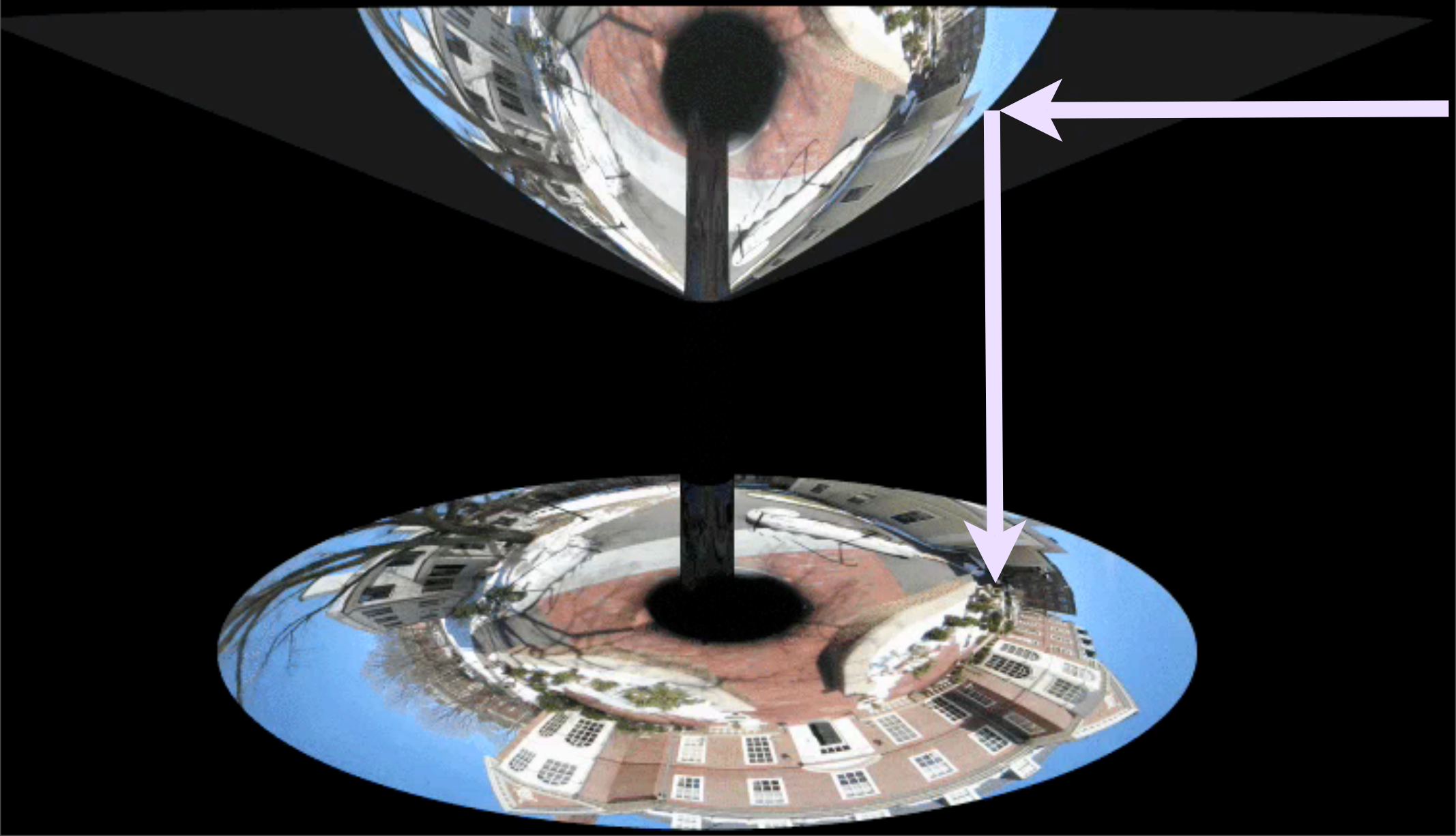

damselfly

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

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

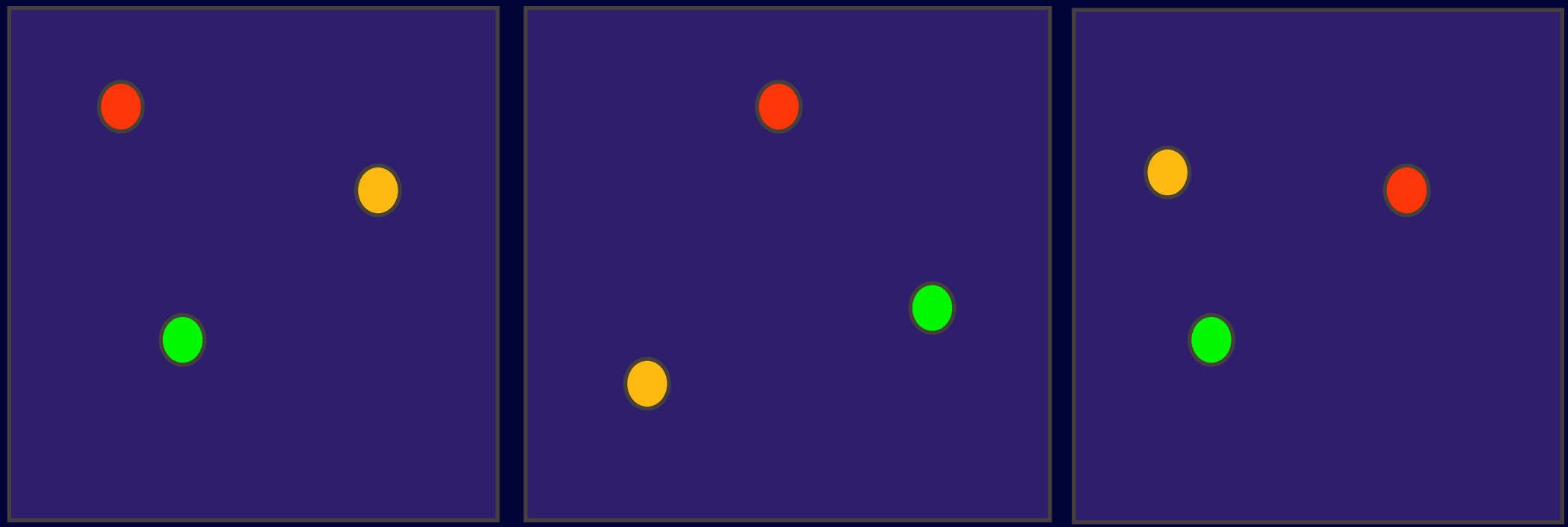
$$y = r\sin(\theta)$$

r

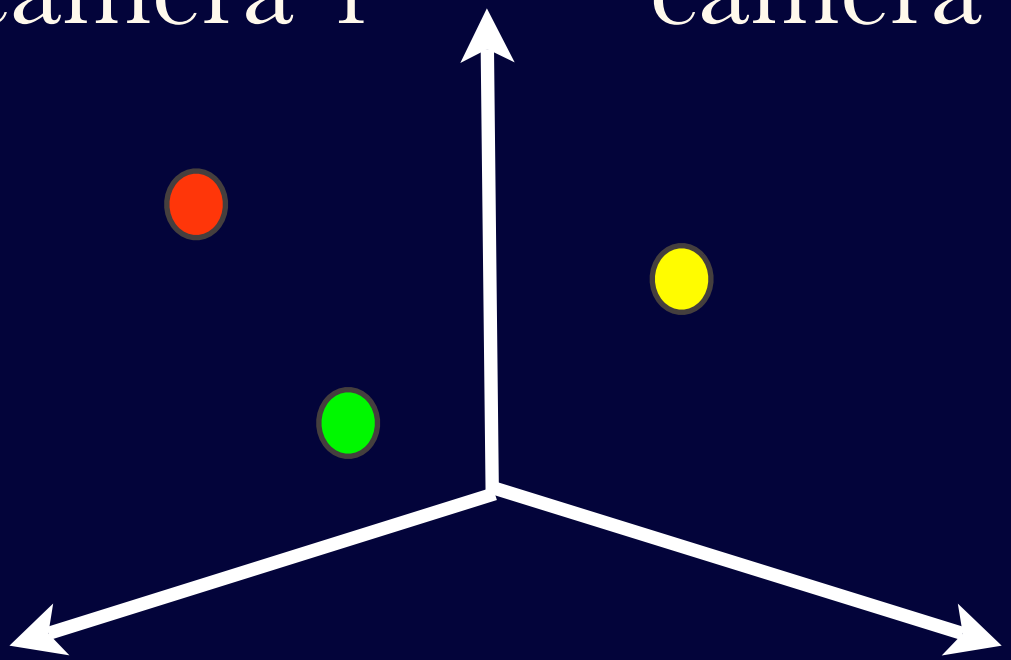
θ

Demonstration

Correspondence problem

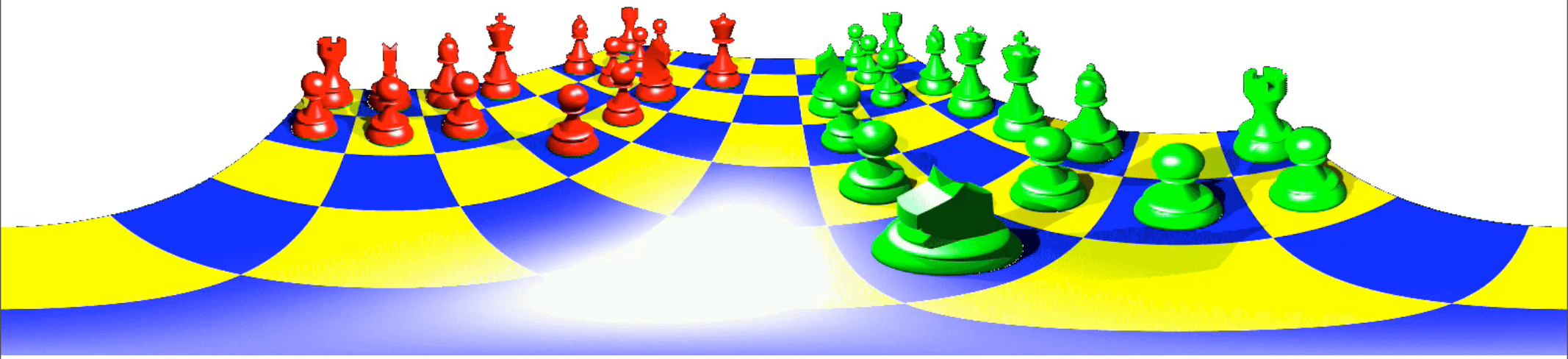


camera 1 camera 2 camera 3



Reconstruct the cameras and the points!

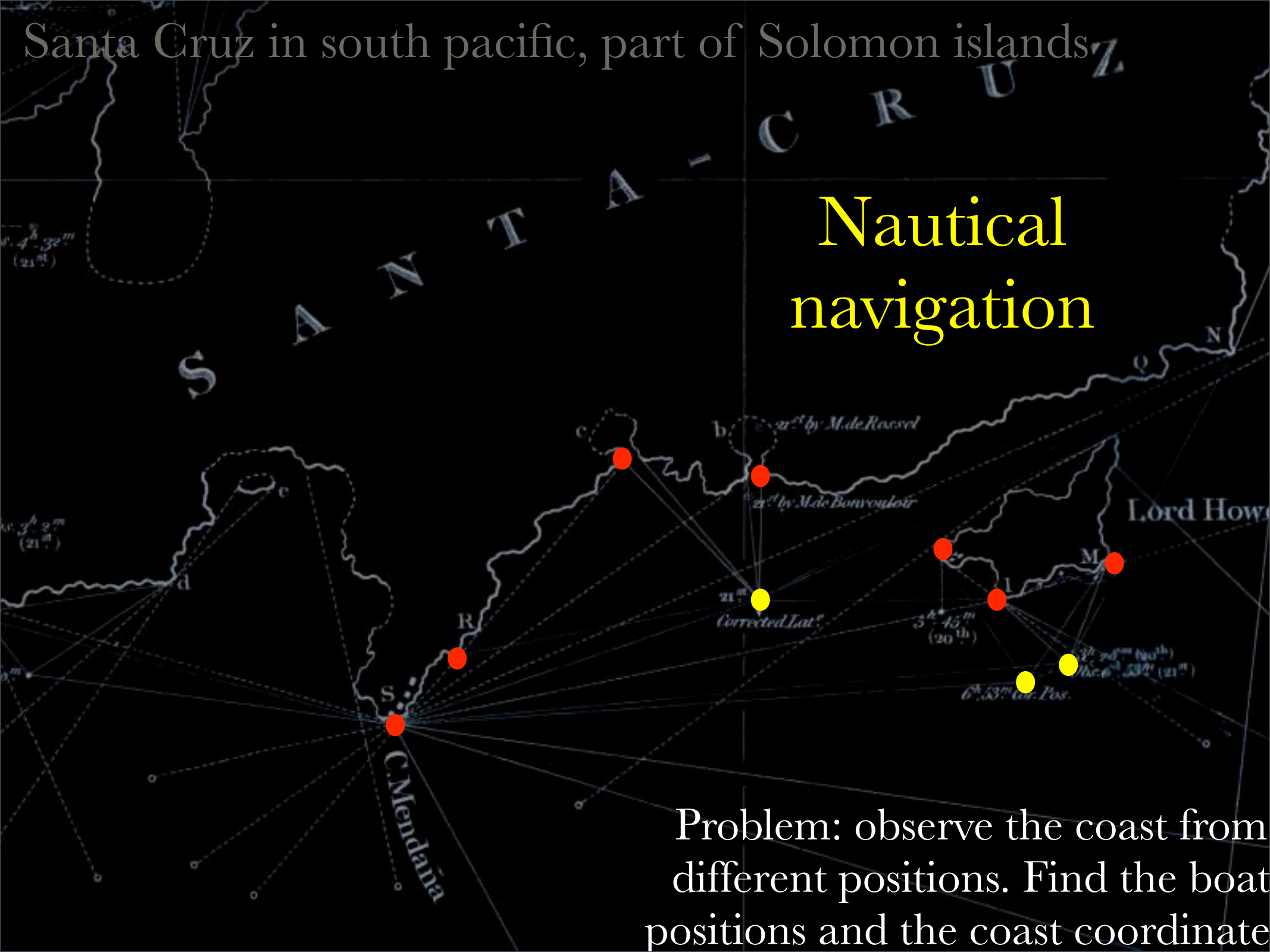
From movie



Omnidirectional problems appeared in nautical navigation


M46
Crater
M93
Pyxis
Antlia
Puppis
S

Santa Cruz in south pacific, part of Solomon islands



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.

$$2^{99} = 63382\,533001\,14114\,70074\,83516\,02688 = a,$$

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

$a \log 2$ 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, III^e^ 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.

Chasles, 1855

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

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].

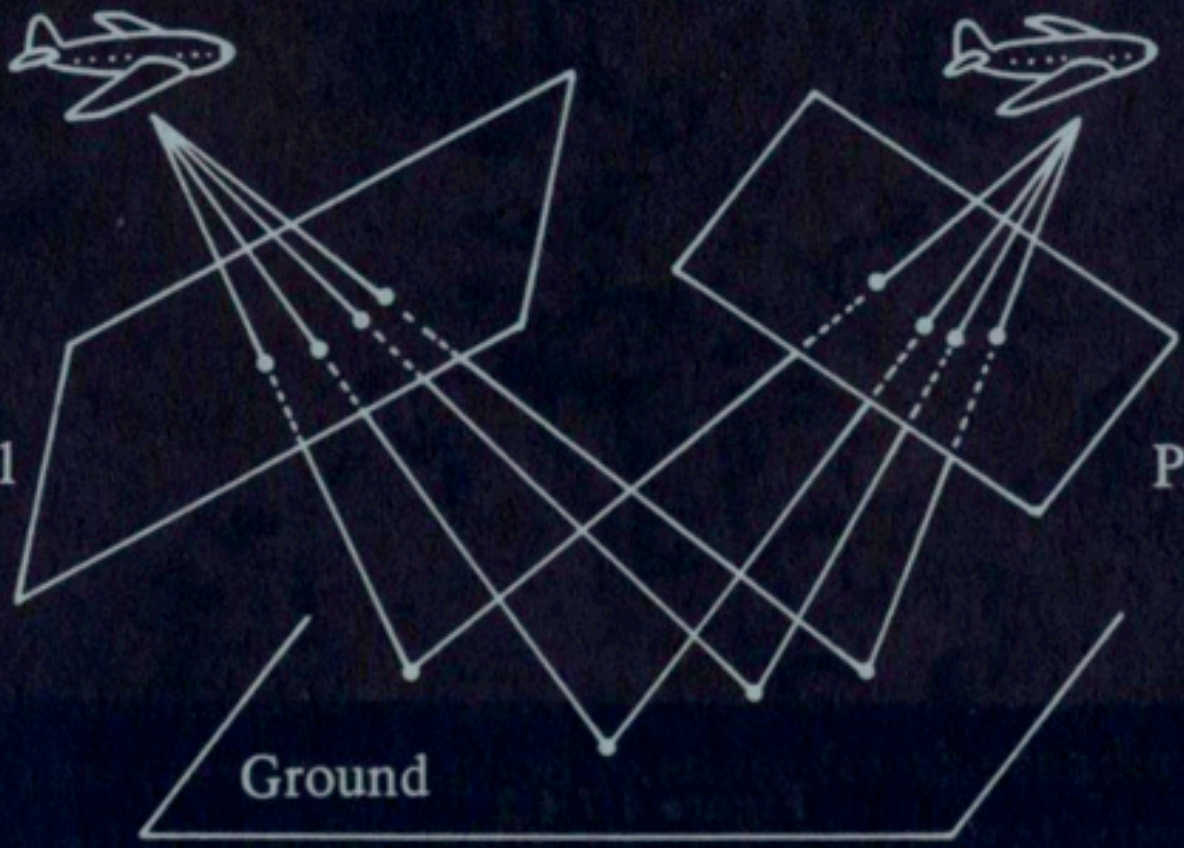


Figure 4.7.3

Photogrammetry

Berger, Geometry I

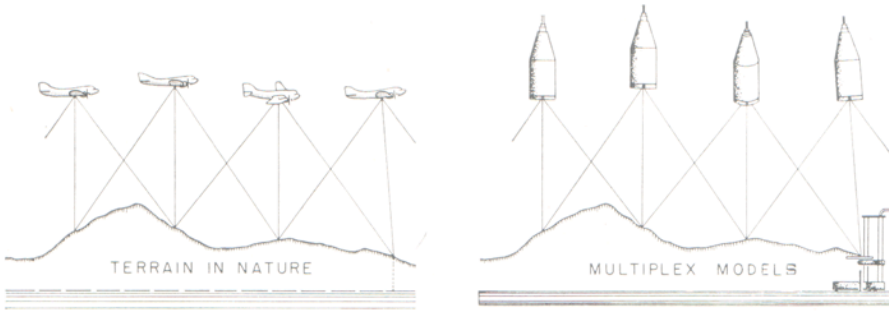
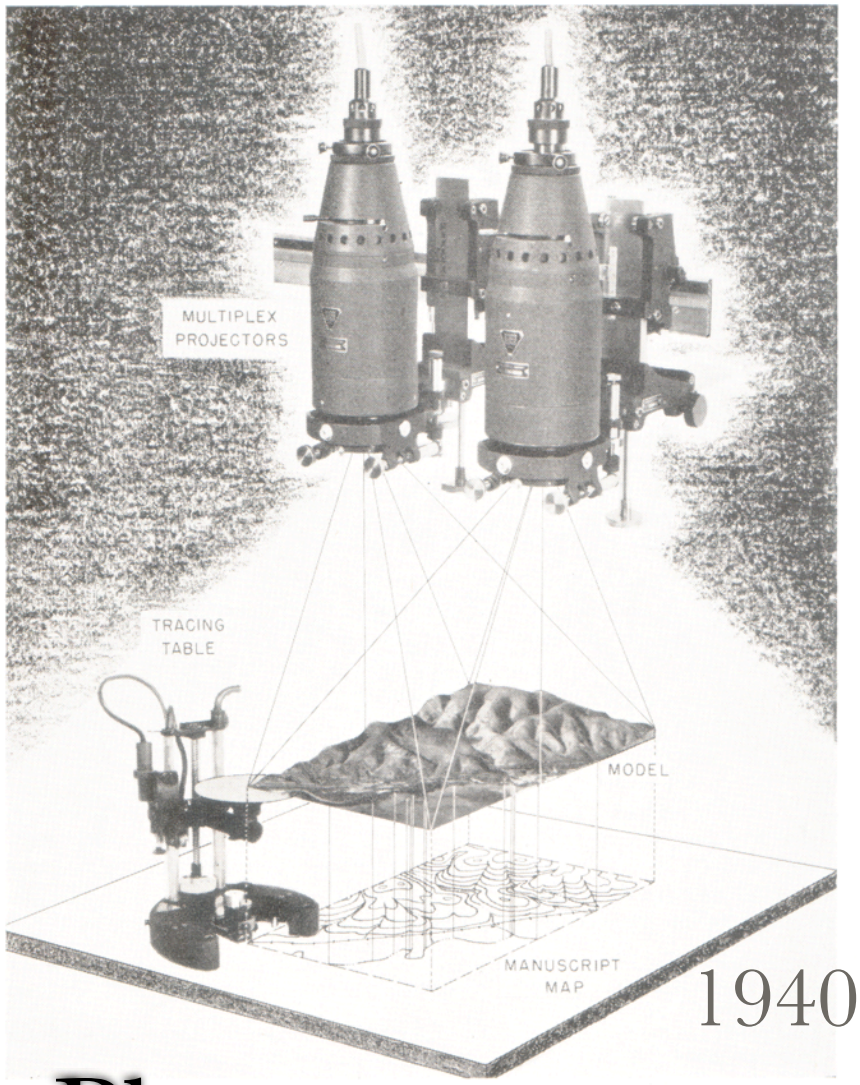


FIG. 1. Geometric parallelism between aerial photography and multiplex reconstruction.



1940

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

1870

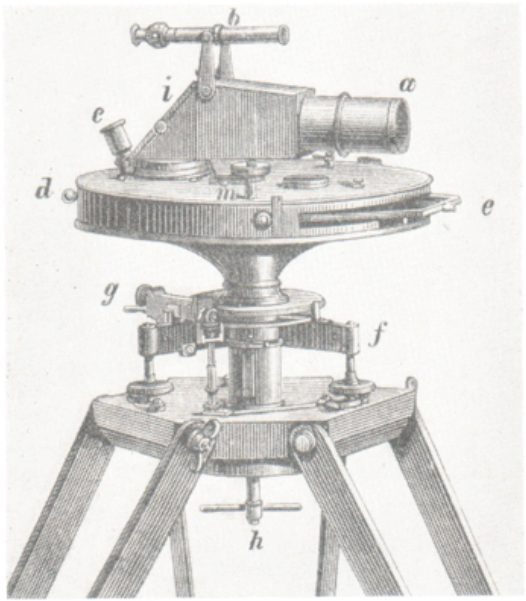


FIG. 5. Photographic plane table of Chevallier, 1858. (Laussedat, vol. 2, pt. 1, p. 30.)

1858

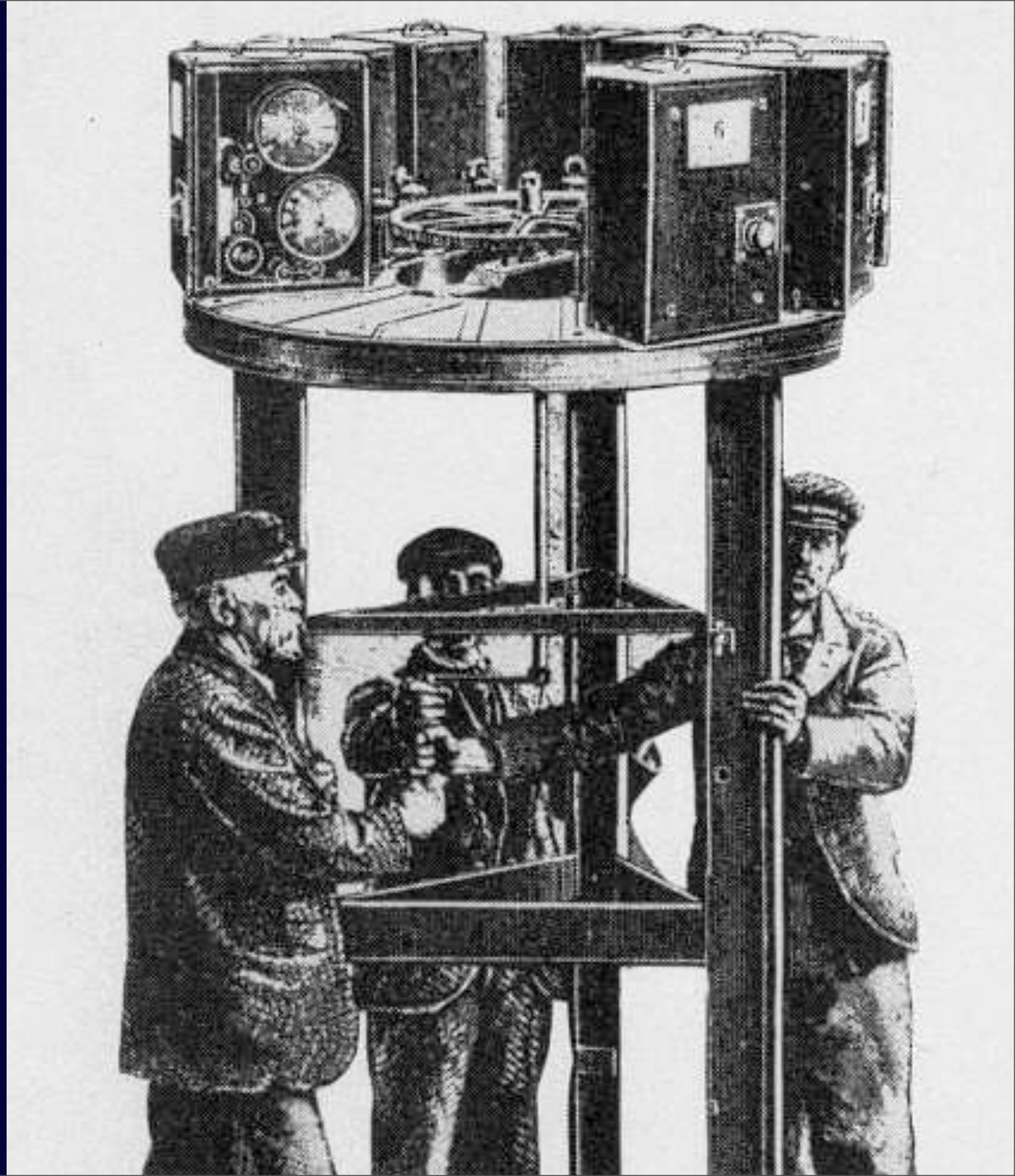
Photogrammetry

Raoul
Grimoin-Sanson
(1860-1940)

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

Cineorama

1900



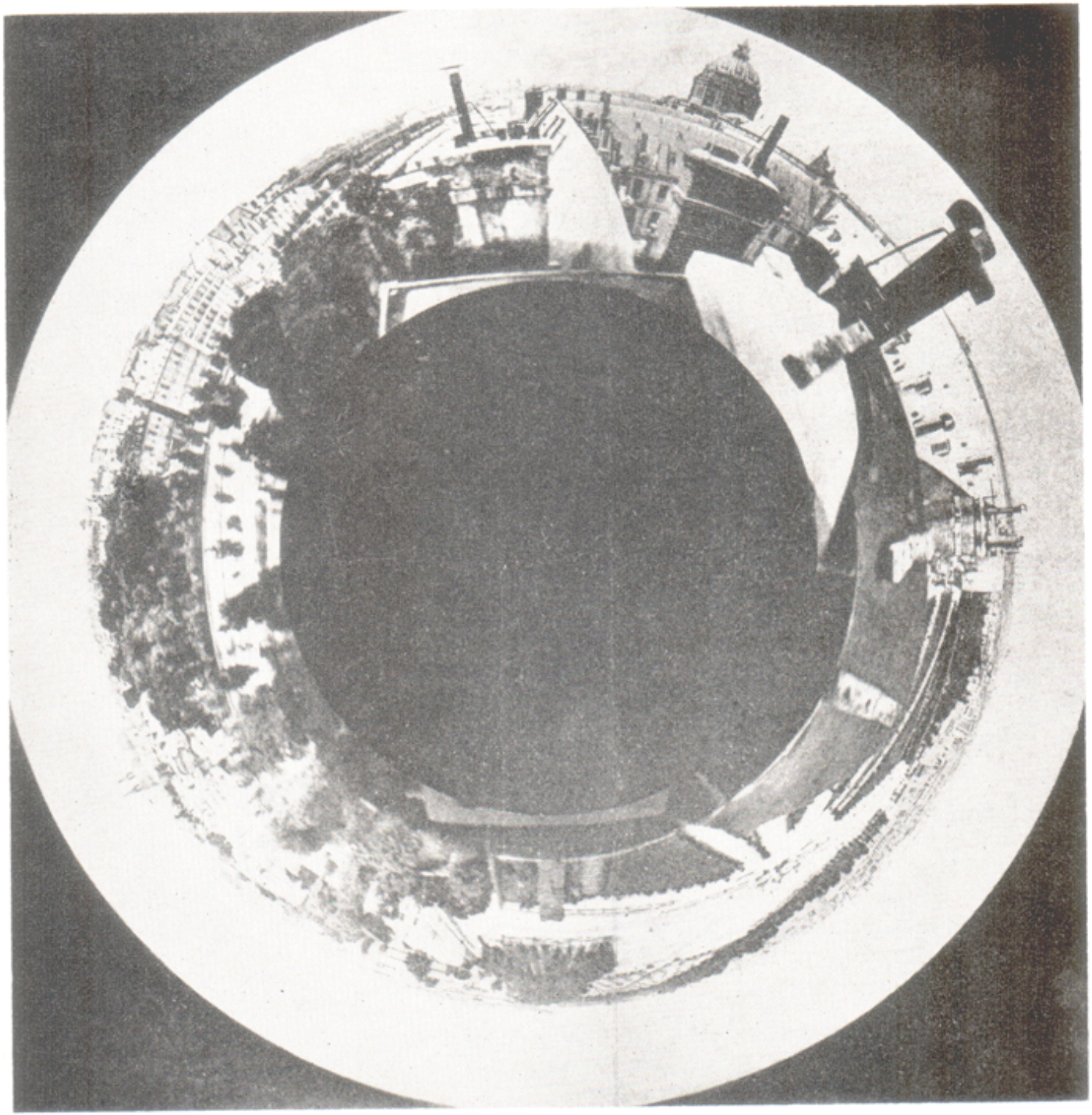


FIG. 7, B. Horizon photograph taken with Mangin's improved apparatus based on Chevallier's photographic plane table. (Laussedat, vol. 2, pt. 1, pl. v.)

TA
593
A63
1952

MANUAL
OF
PHOTOGRAMMETRY

(*Second Edition*)

AMERICAN SOCIETY
OF
PHOTOGRAMMETRY

Chevaliers photographic table

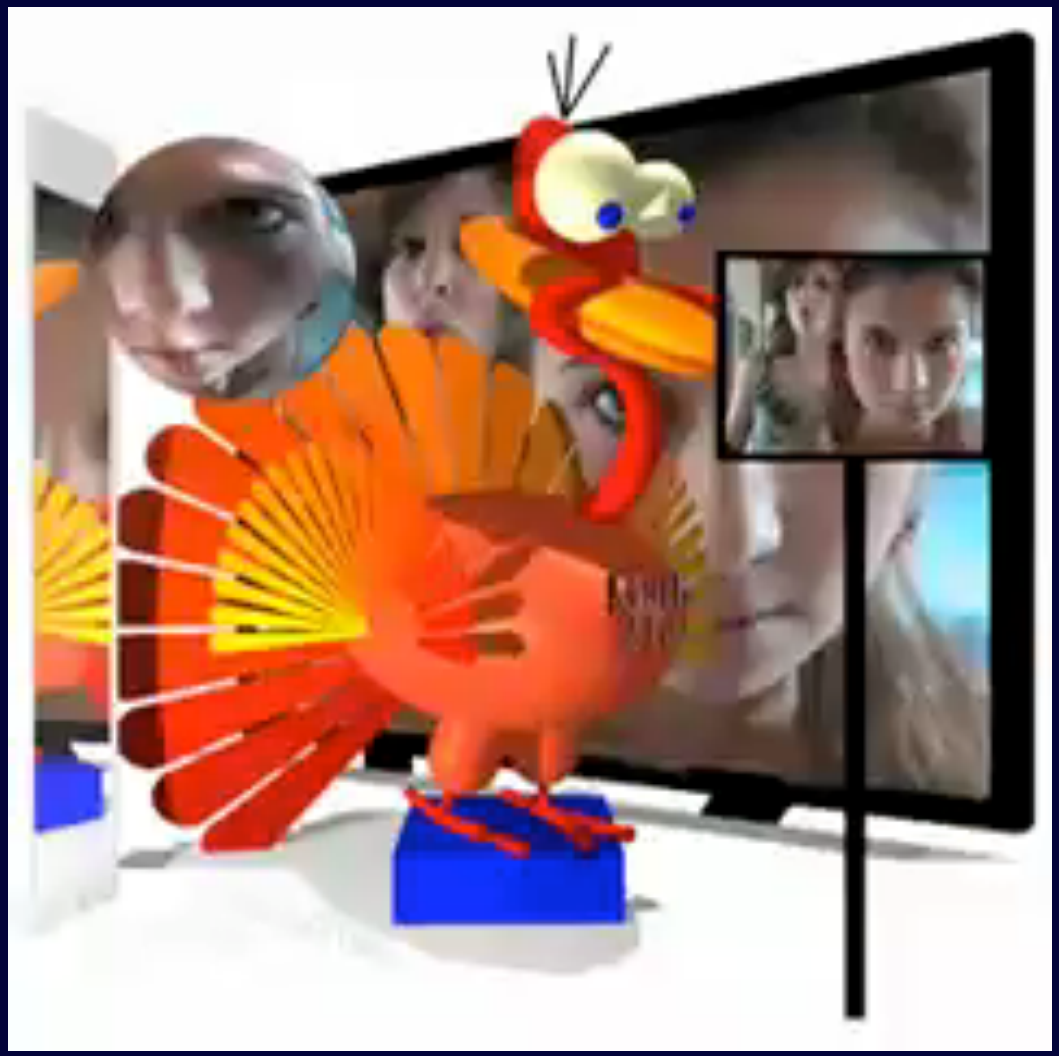
Synthetic Panoramas

Computer generated panoramas

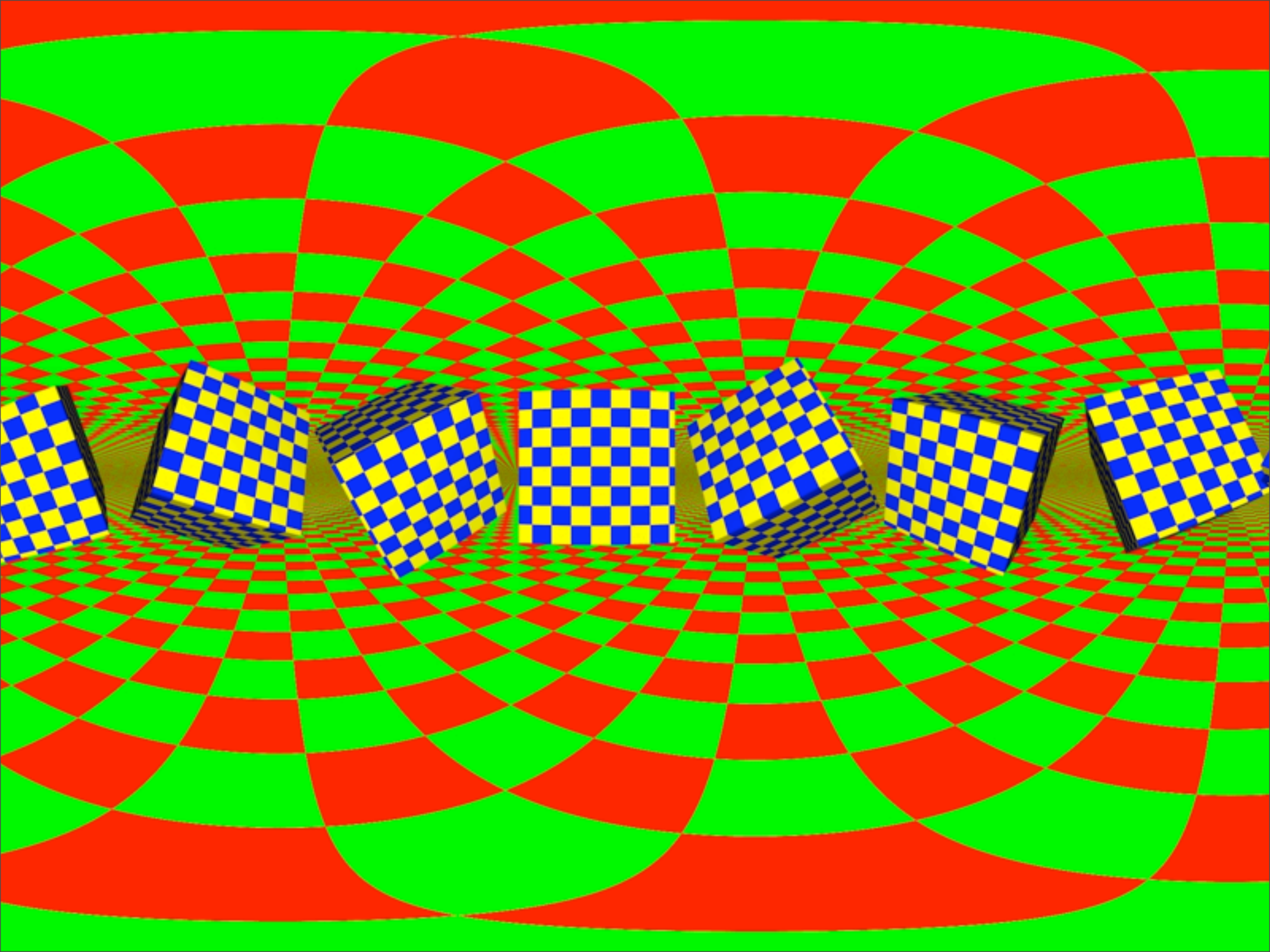
The next examples were made with the open source ray tracer Povray (Persistence of vision ray tracer). This software can produce pictures like this:


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

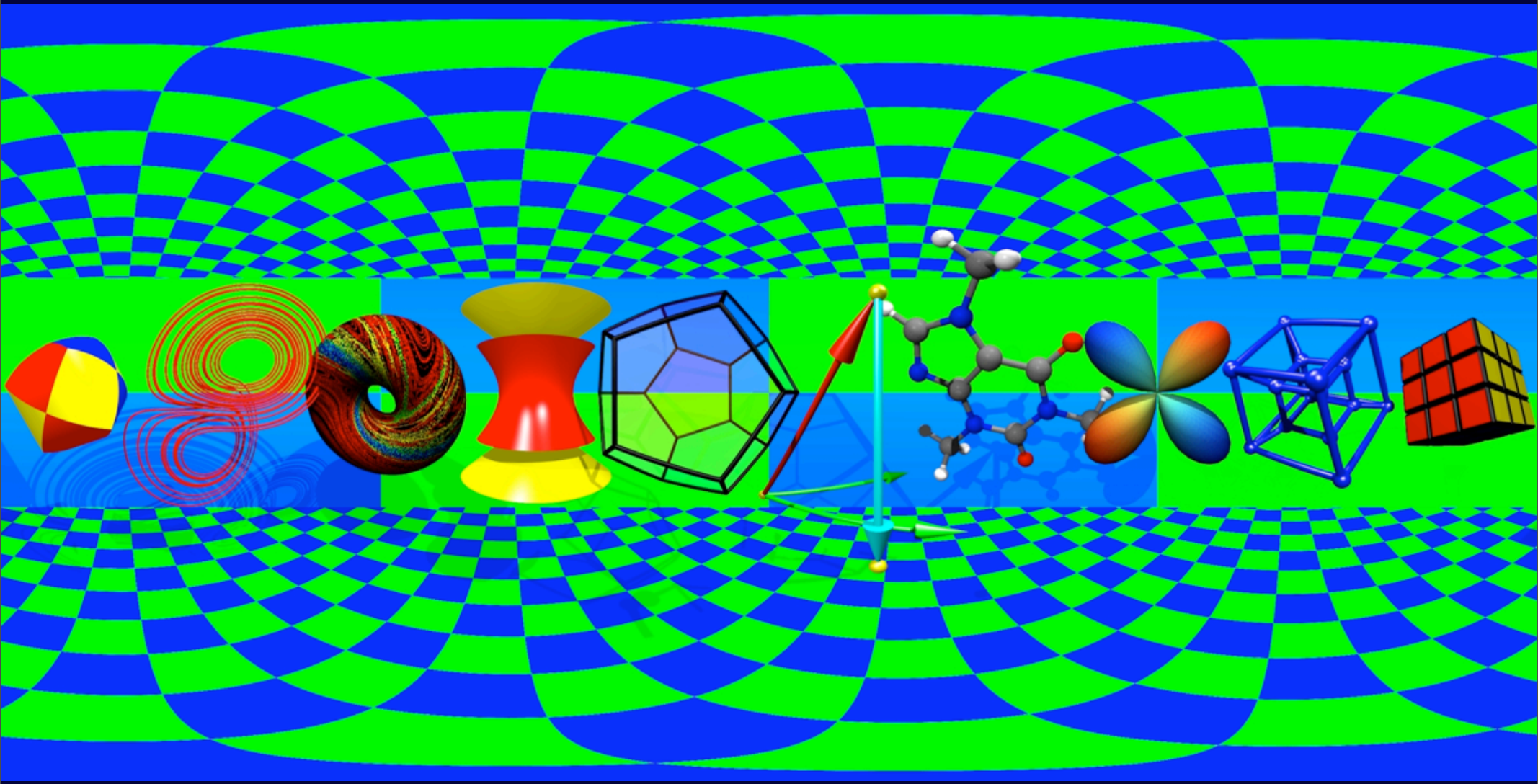


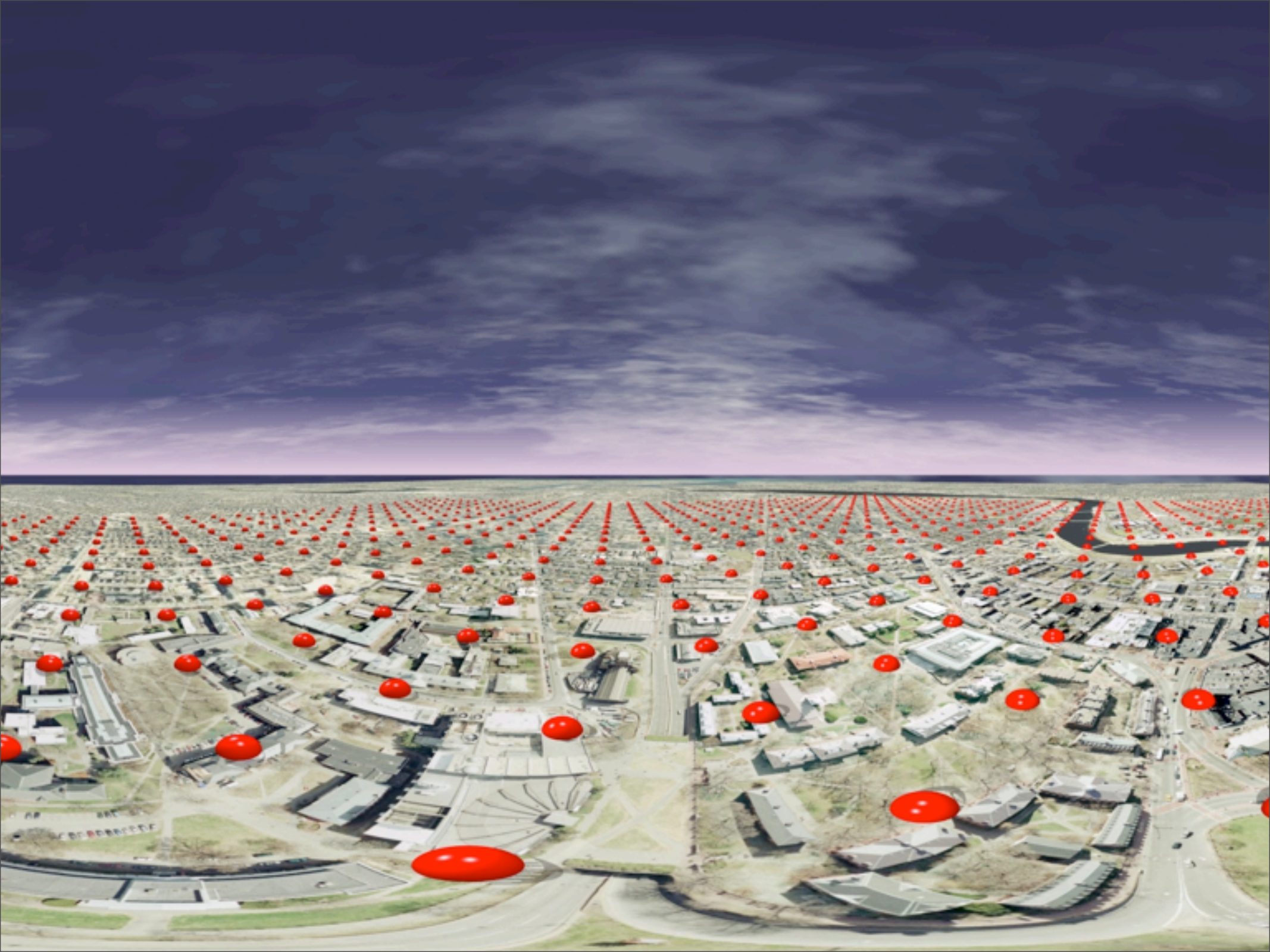
Next comes an example of a panorama:

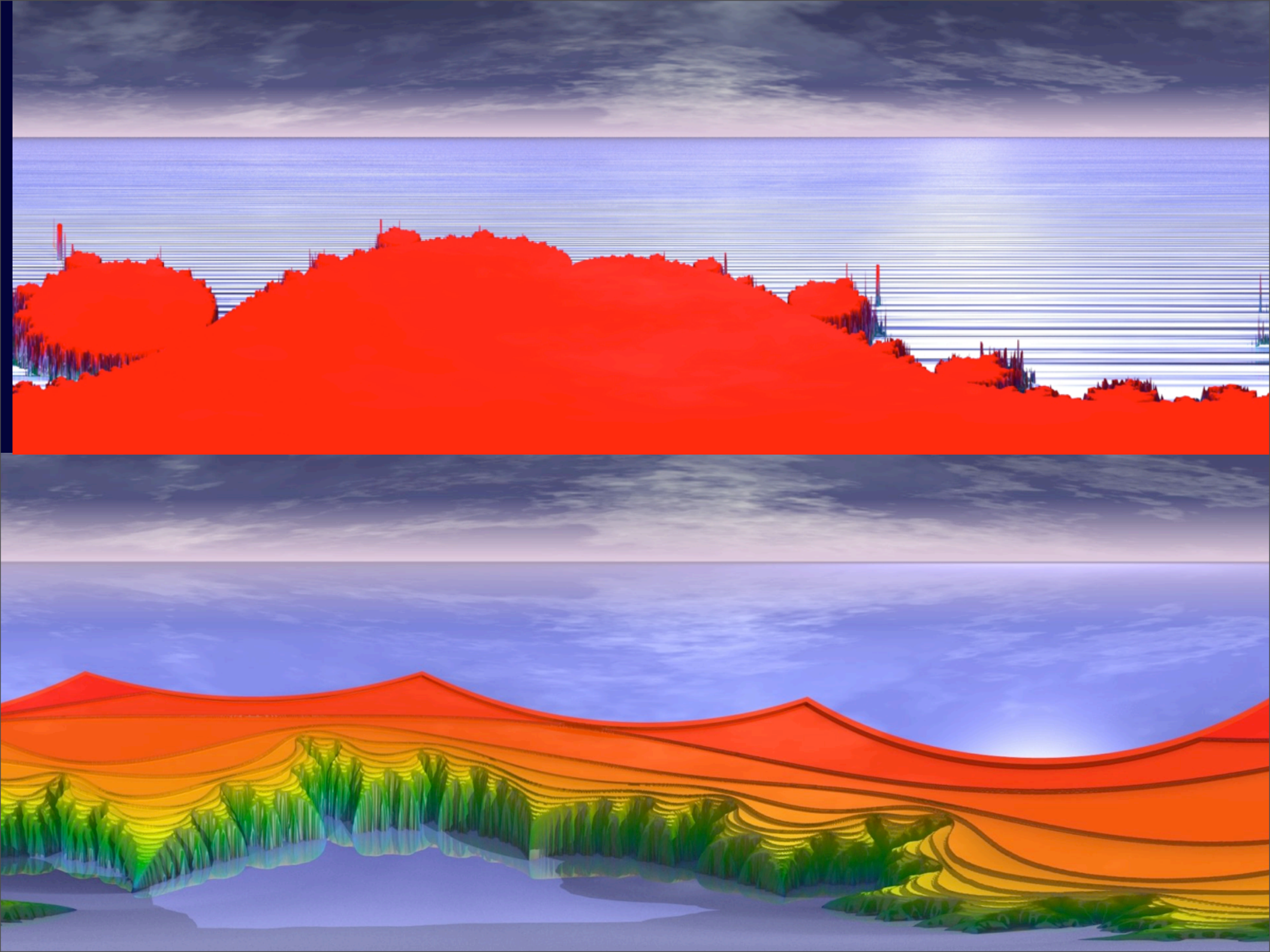


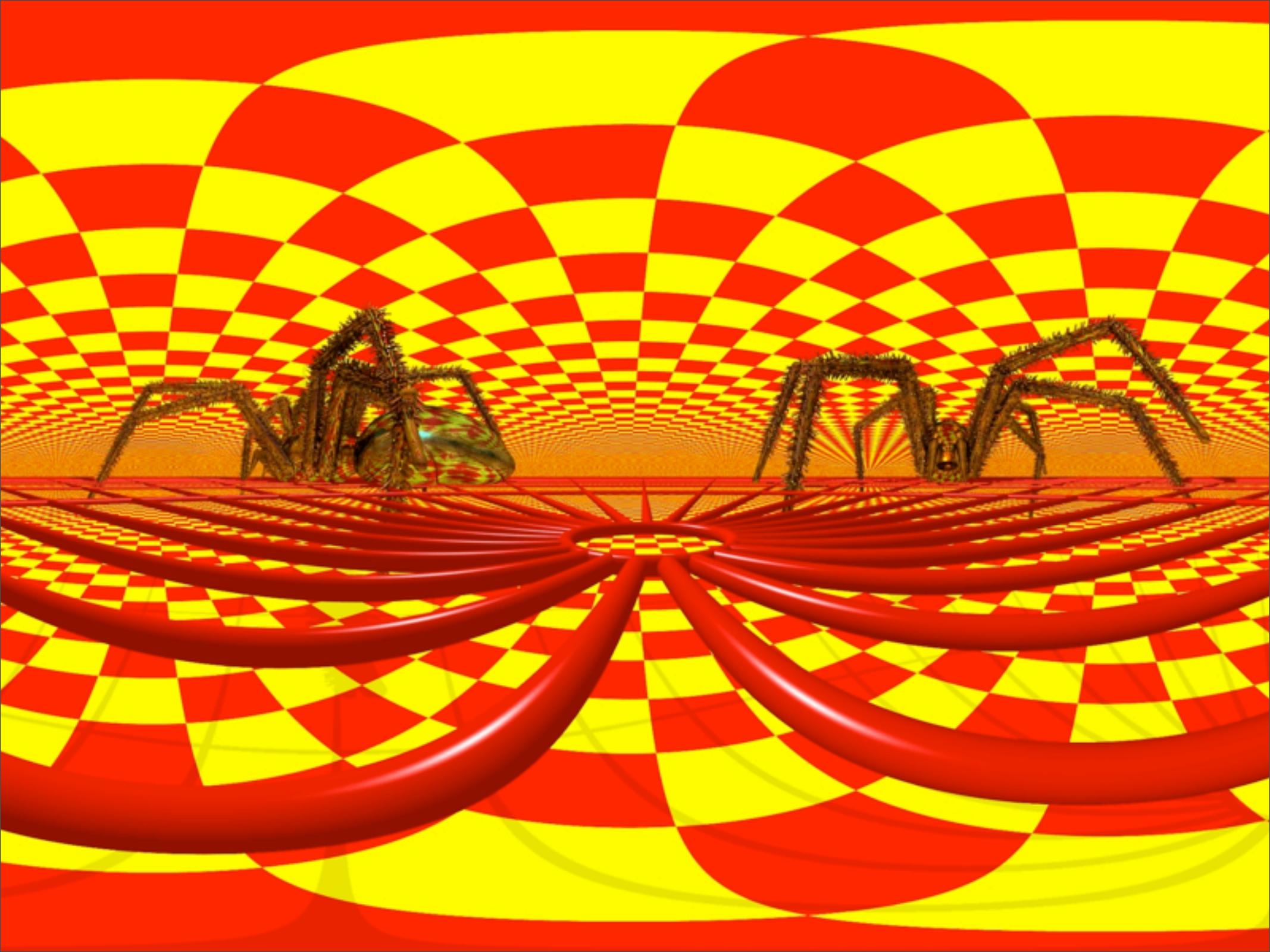
Show me the code:

```
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
```





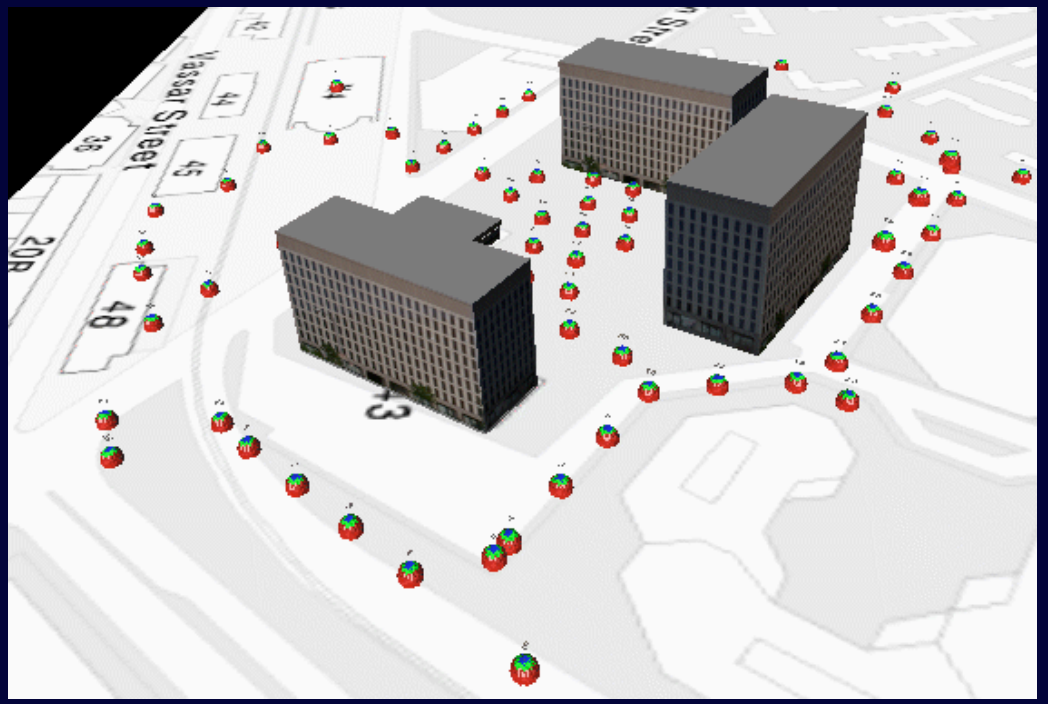


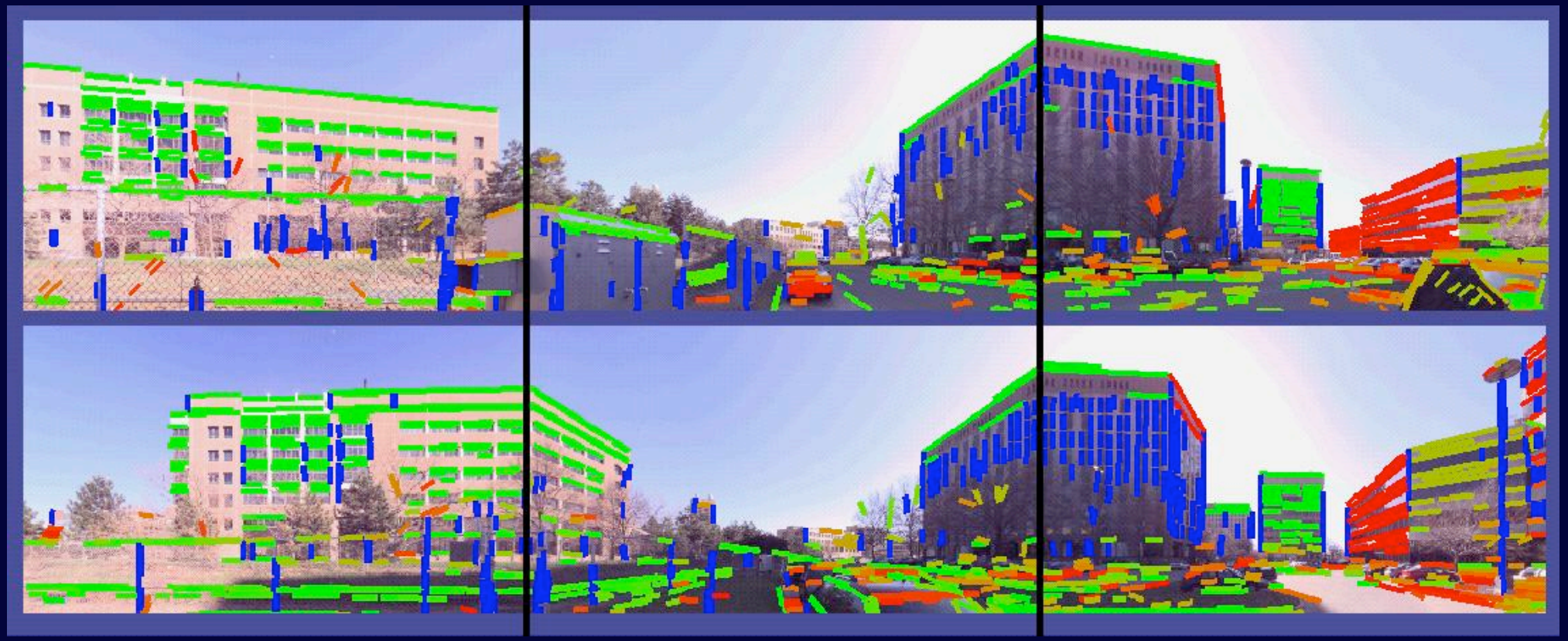


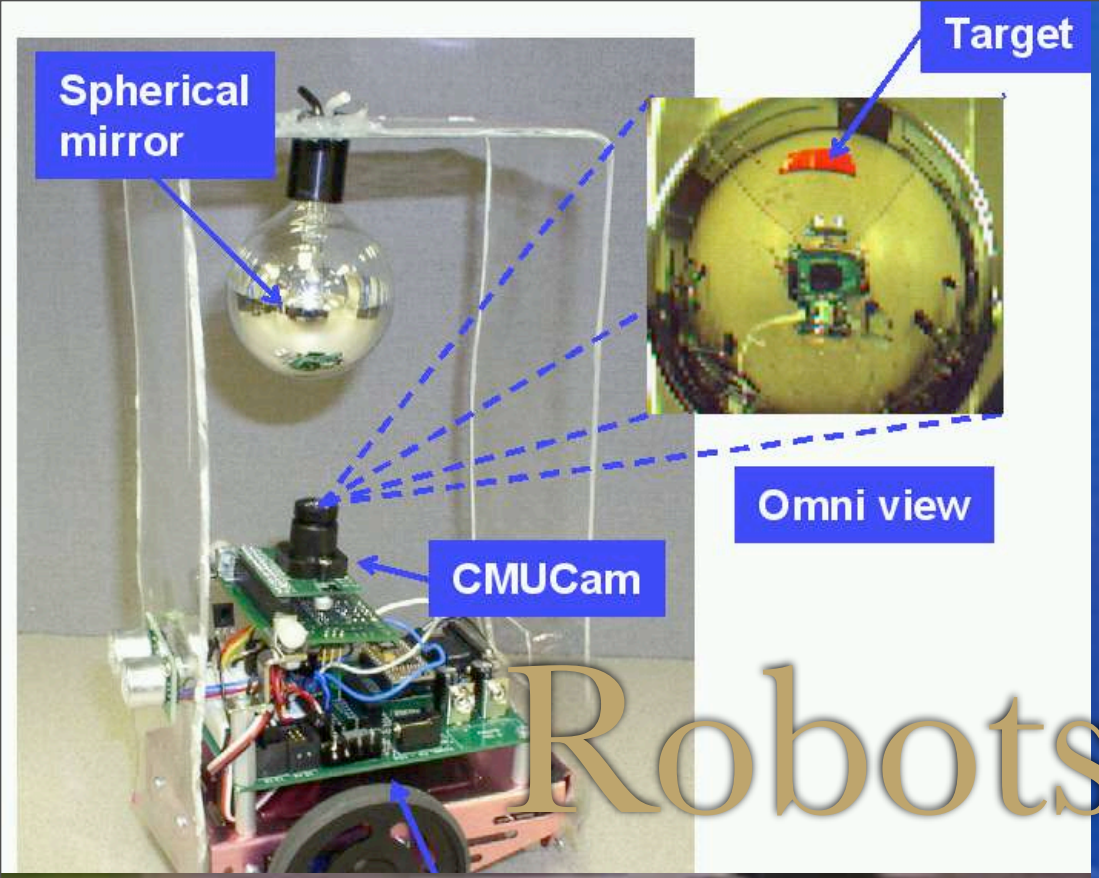
Applications

City scanning

MIT







Robots

Google Street maps


University of Fukui

The end.