Visual Inspection Planning: Research Update

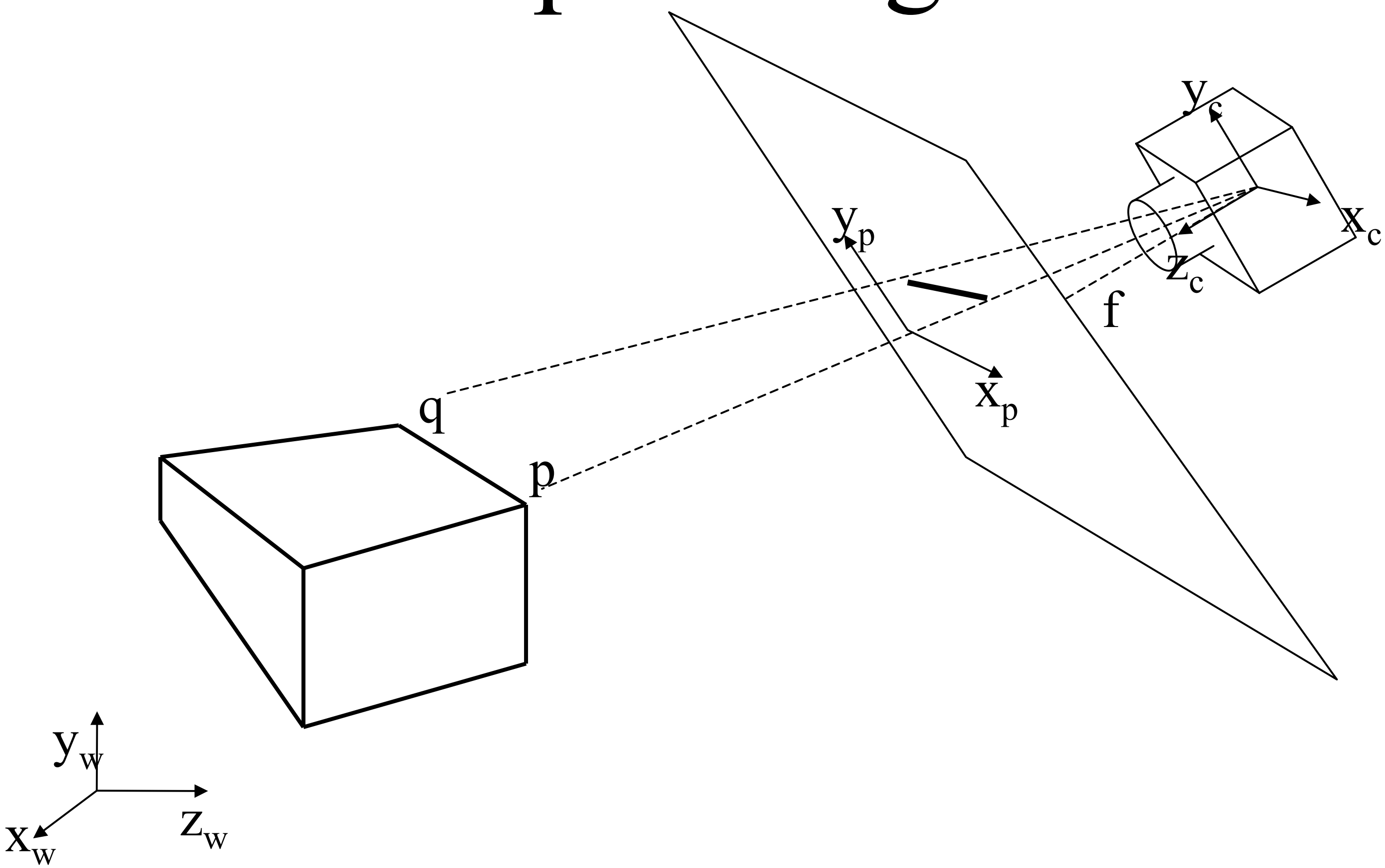
Alexis H Rivera-Rios

Fai Lung Shih

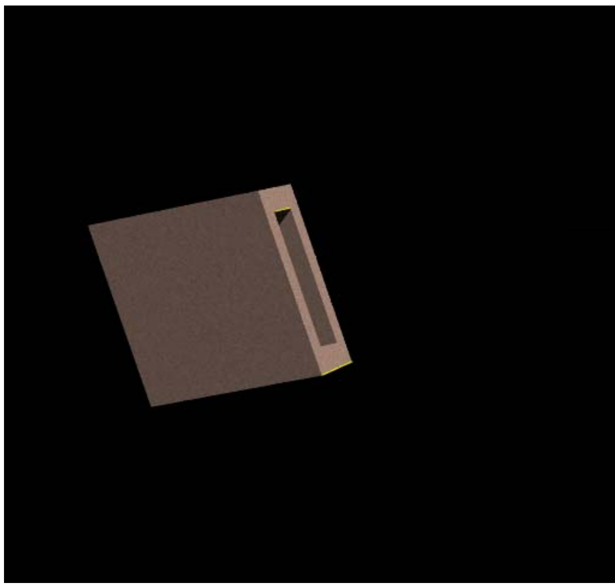
Outline

- Very brief summary of what is visual inspection planning
- Camera models
- Incidence Angle
- Feasible Pose Determination
- Issues with intrinsic errors model
- Total Percent Error
- Robustness Index

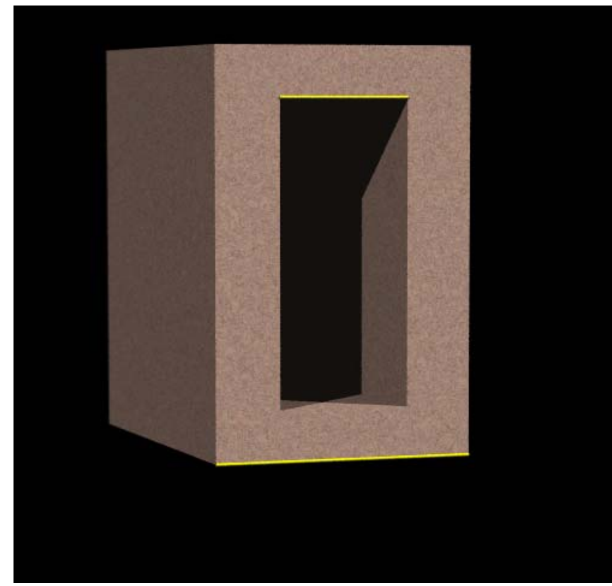
What is visual inspection planning?



Example

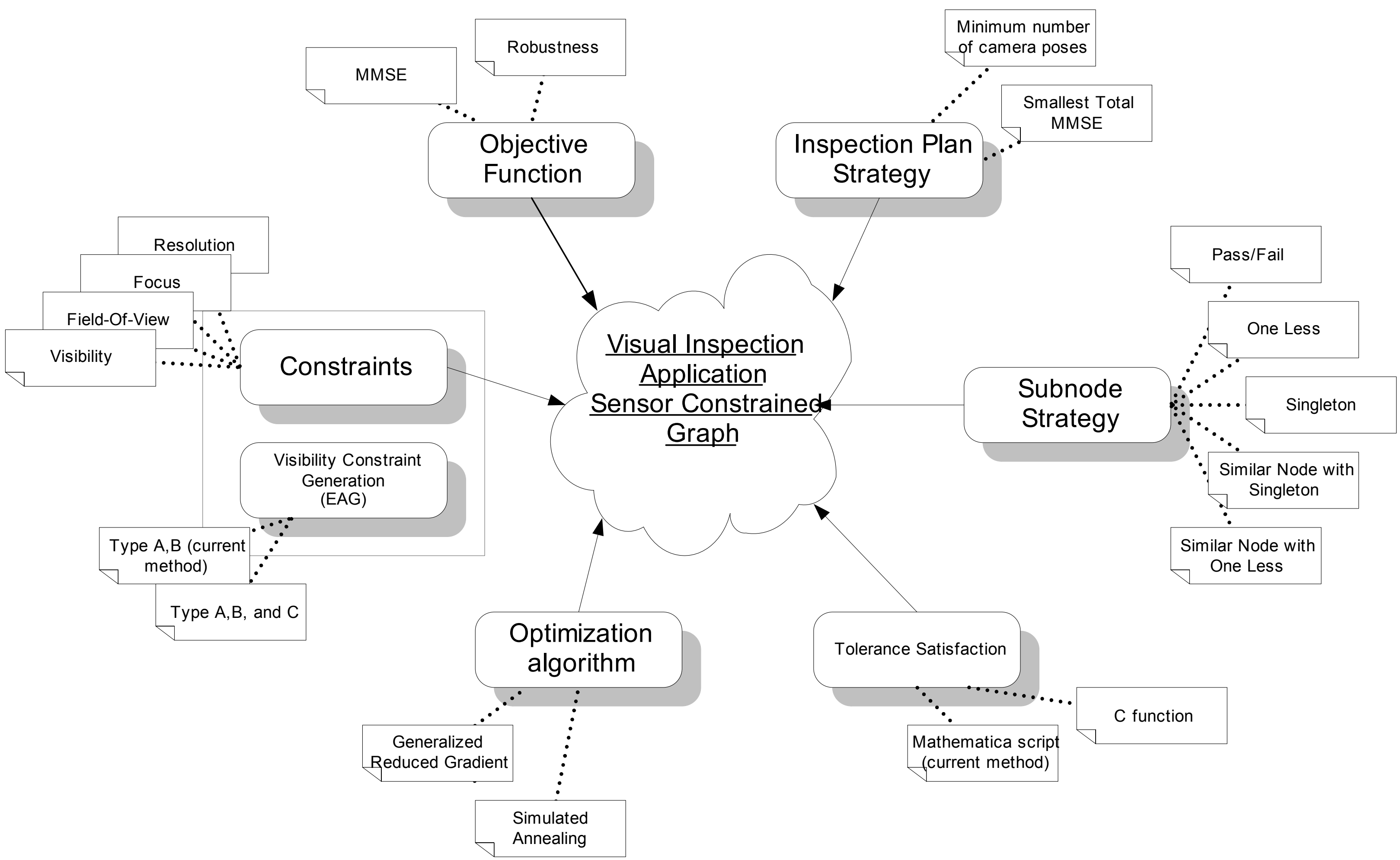


Optimal Value: 19.993025



Optimal Value: 0.356250

(Goal) Automated Plan Generation



Visual Inspection Problem

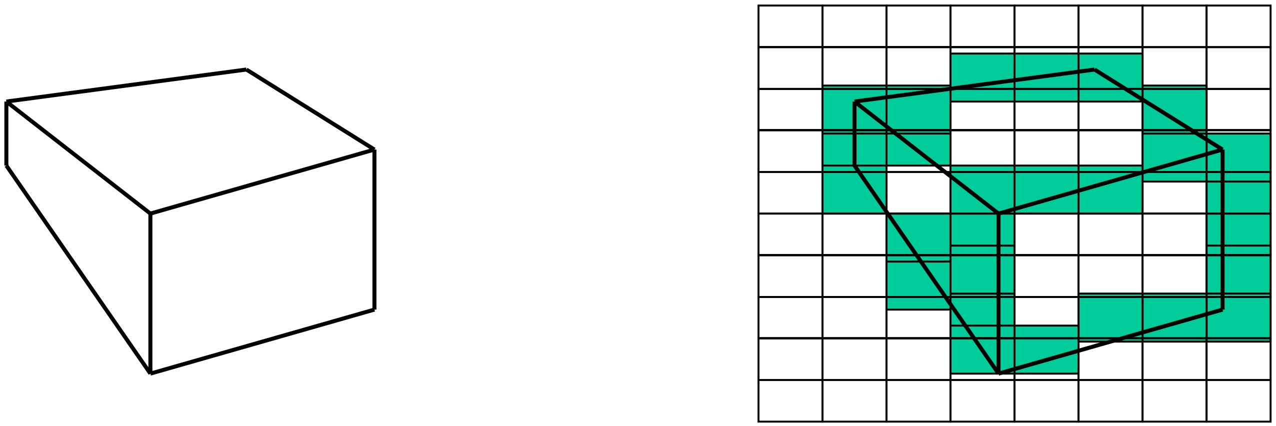
- Finding optimal camera pose:
  - For a given set of entities, S
  - minimize Obj (t_x, t_y, t_z, Φ, θ, Ψ, S)
  - subject to:
    - g1j <= 0 (resolution), for j=1 to k
    - g2a <= 0 (focus)
    - g2b <= 0
    - g3 <= (field of view)
    - g4i <= 0 (visibility) for i=1 to m

Objective Function

- Indicates the goodness of the camera pose
  - Mean Square Error of intrinsic camera errors (Crosby)
  - Robustness criteria (Gu)
  - Robustness criteria (Tarabanis)

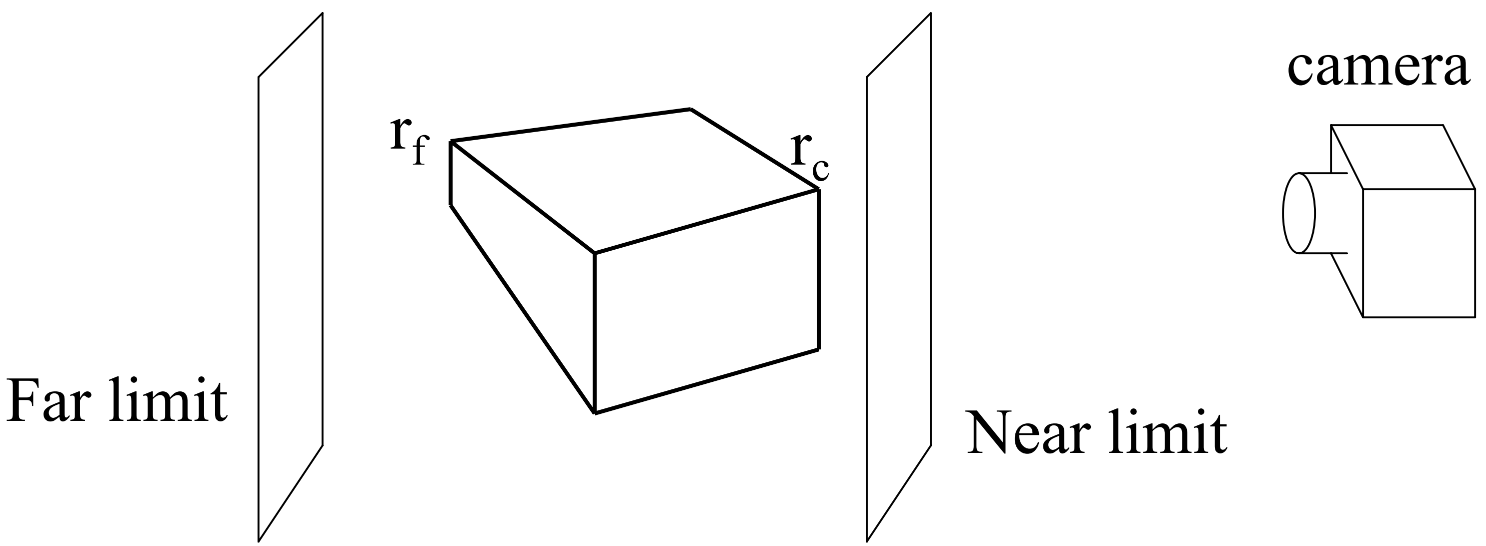
Resolution

- For each entity j, there is a constraint g1j()
- Projects a line of *l* millimeters to a line of w millimeters



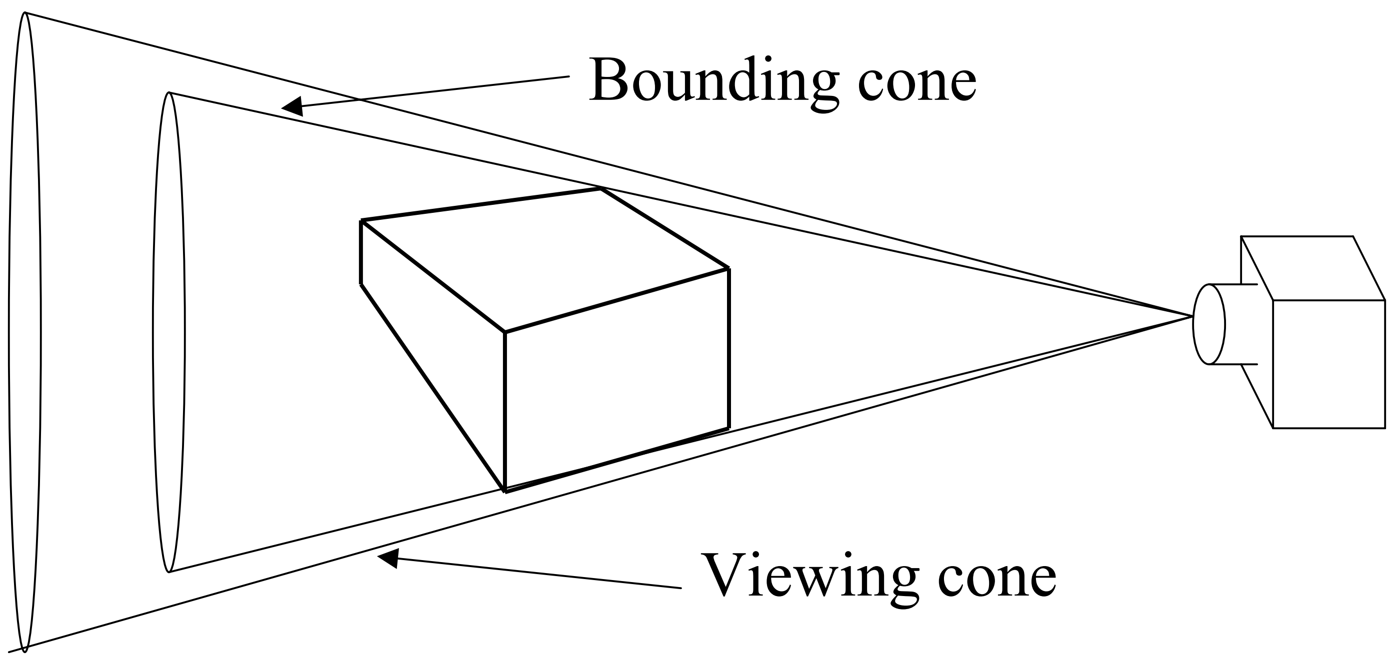
Focus

- Two constraints, g2a(), g2b()
- Require closest and furthest entity vertices from the camera position to be within the far and near limits of the depth of field



Field Of View

- One constraint: g3()
- Bounding cone must be contained within the viewing cone

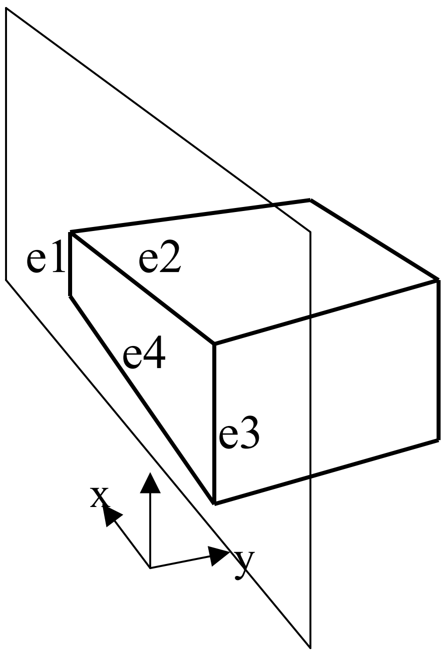


Visibility

- Many equations: g4i() for i=1 to m
- Plane equations that bound the visibility of the desired entities

Example:

To see entities e1, e2, e3, e4, the camera must satisfy equation y < 0



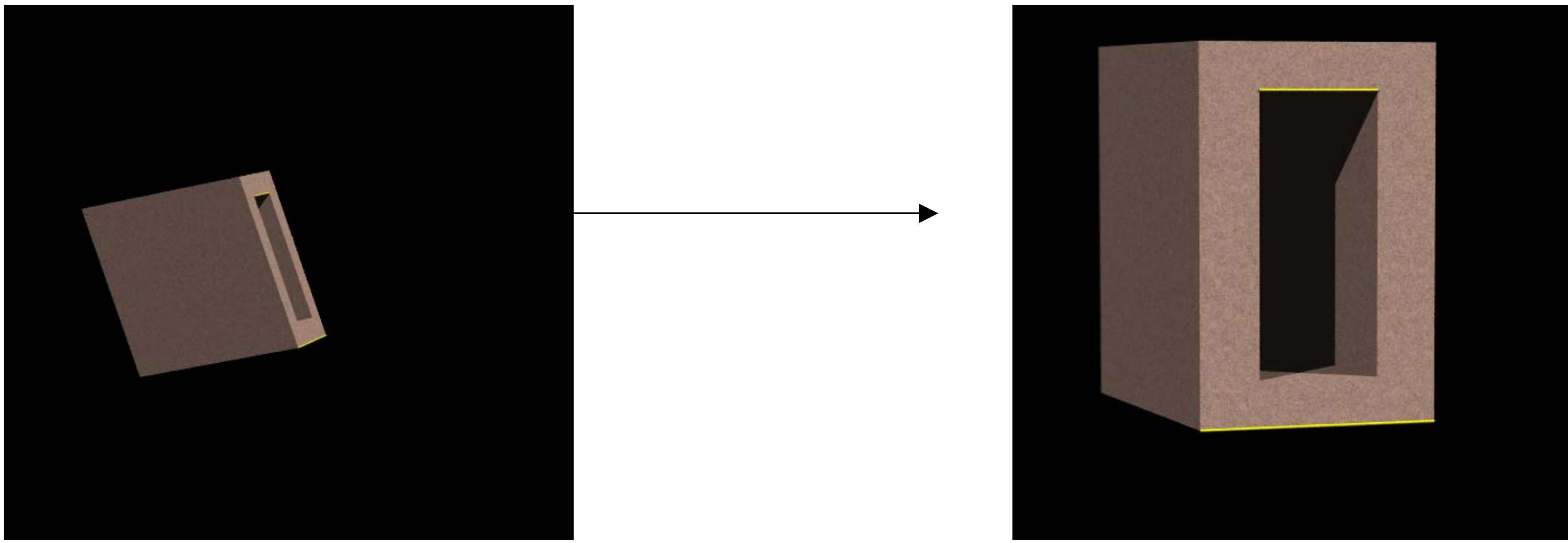
Dimensional Tolerances

- Dimensional Tolerance is satisfied if

$$\int_{-\Delta L}^{\Delta L} f_{\varepsilon}(\varepsilon)\,d\varepsilon \geq Threshold$$

- Ie. The probability that the measurement is within the tolerance is greater than a specified threshold
- $f_{\varepsilon}(\varepsilon)$ is the probability density function of dimensional inspection error

Example



Optimal Value: 19.993025
Prob: 0.102

Optimal Value: 0.356250
Prob: 0.997

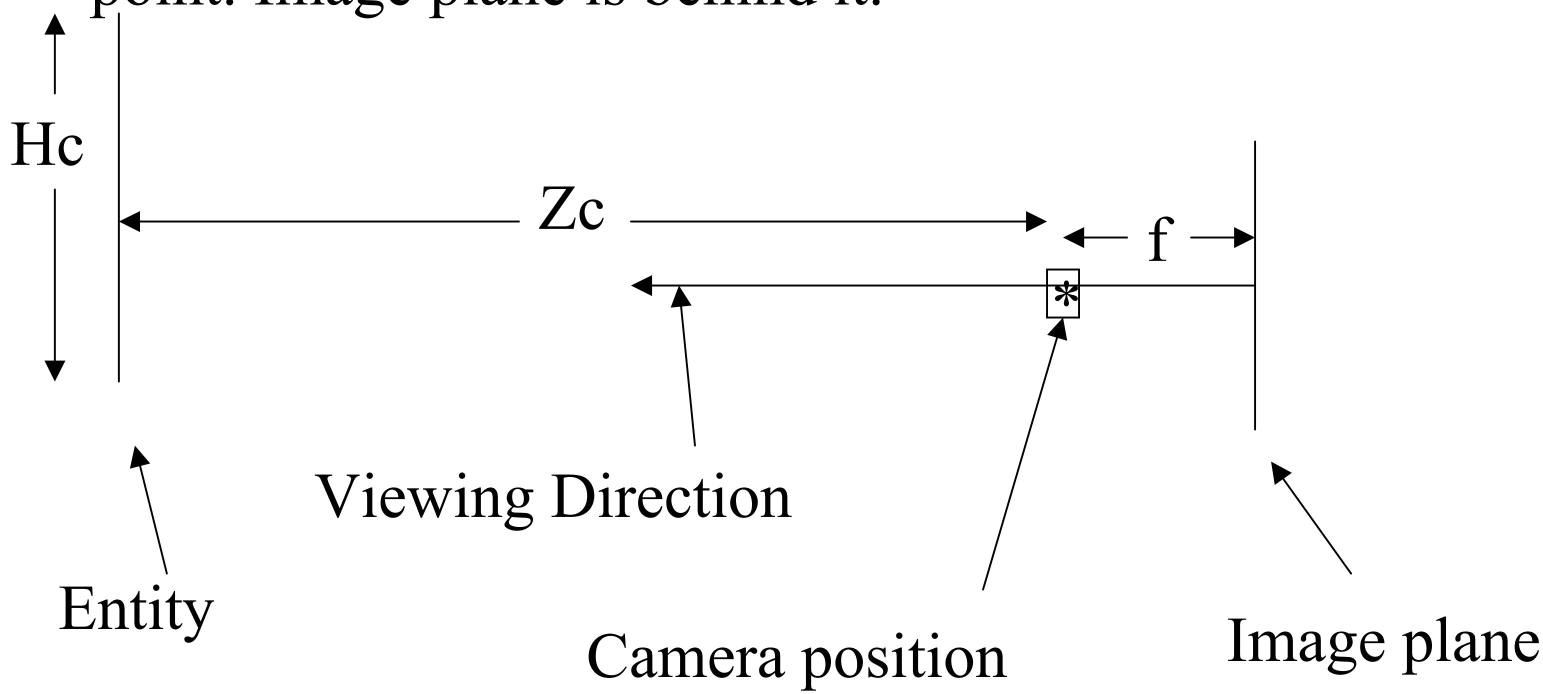
Making constraints and objective functions consistent

Alexis Rivera

Camera models (Tarabanis)

Illustrating imaging process (2D case)

NOTE: Camera position is the position of the focal point. Image plane is behind it.



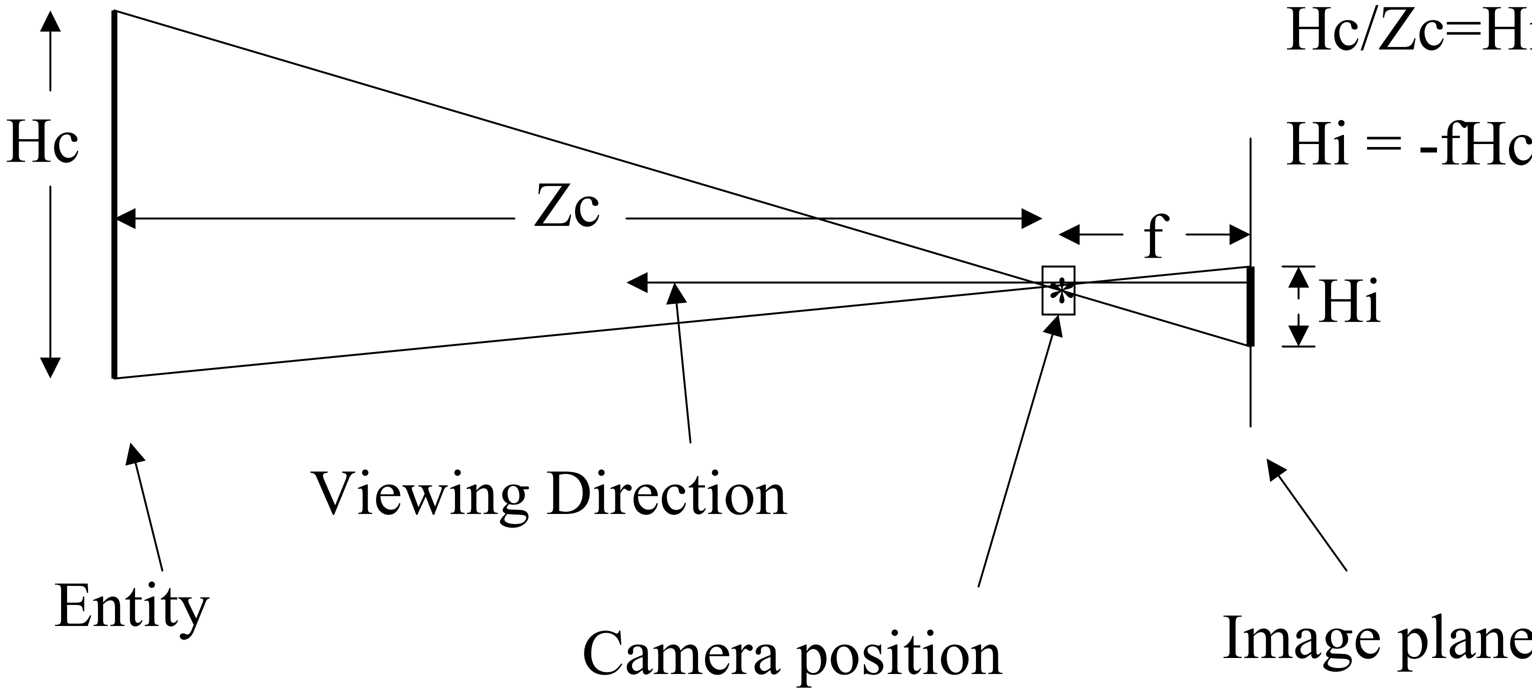
Camera models (Tarabanis)

Illustrating imaging process (2D case)

By similar triangles:

$Hc/Zc = Hi/-f$

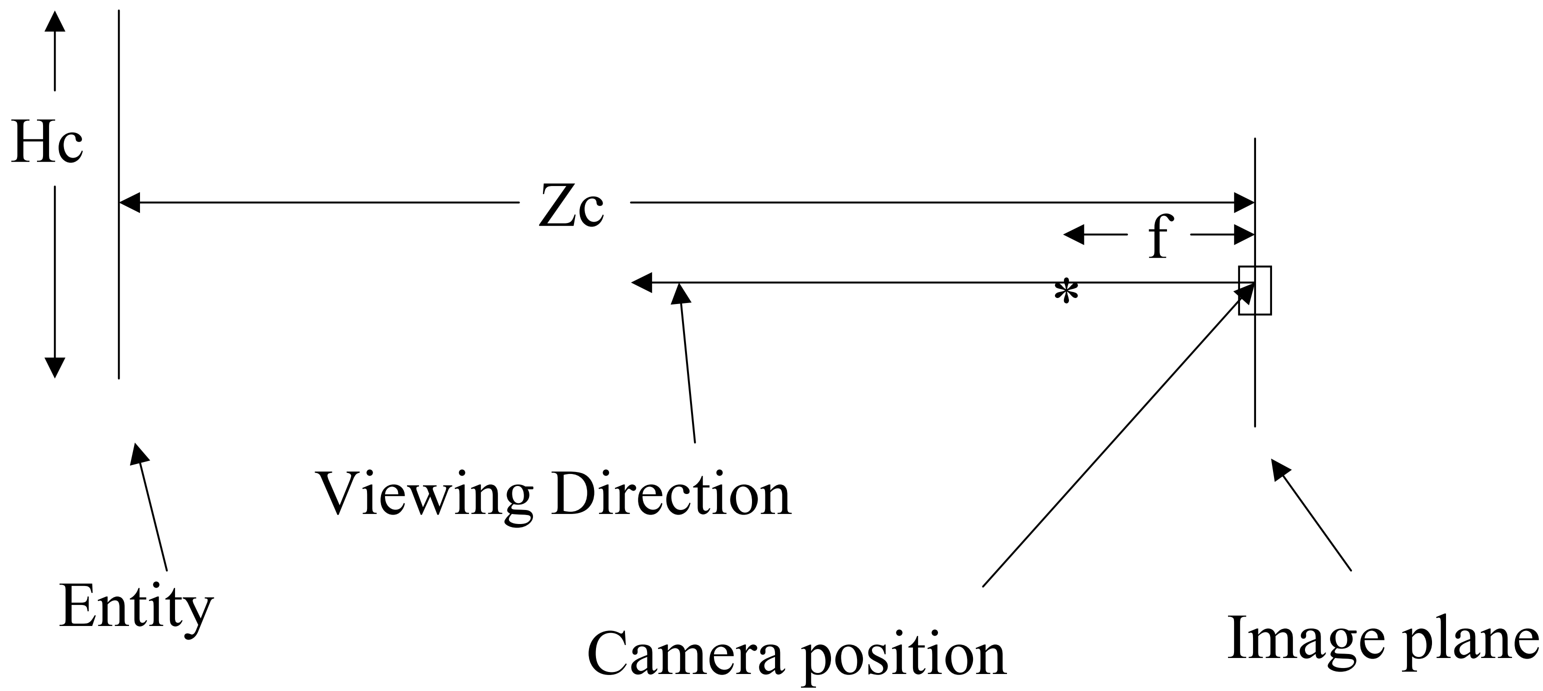
$Hi = -fHc/Zc$



Camera models (Crosby)

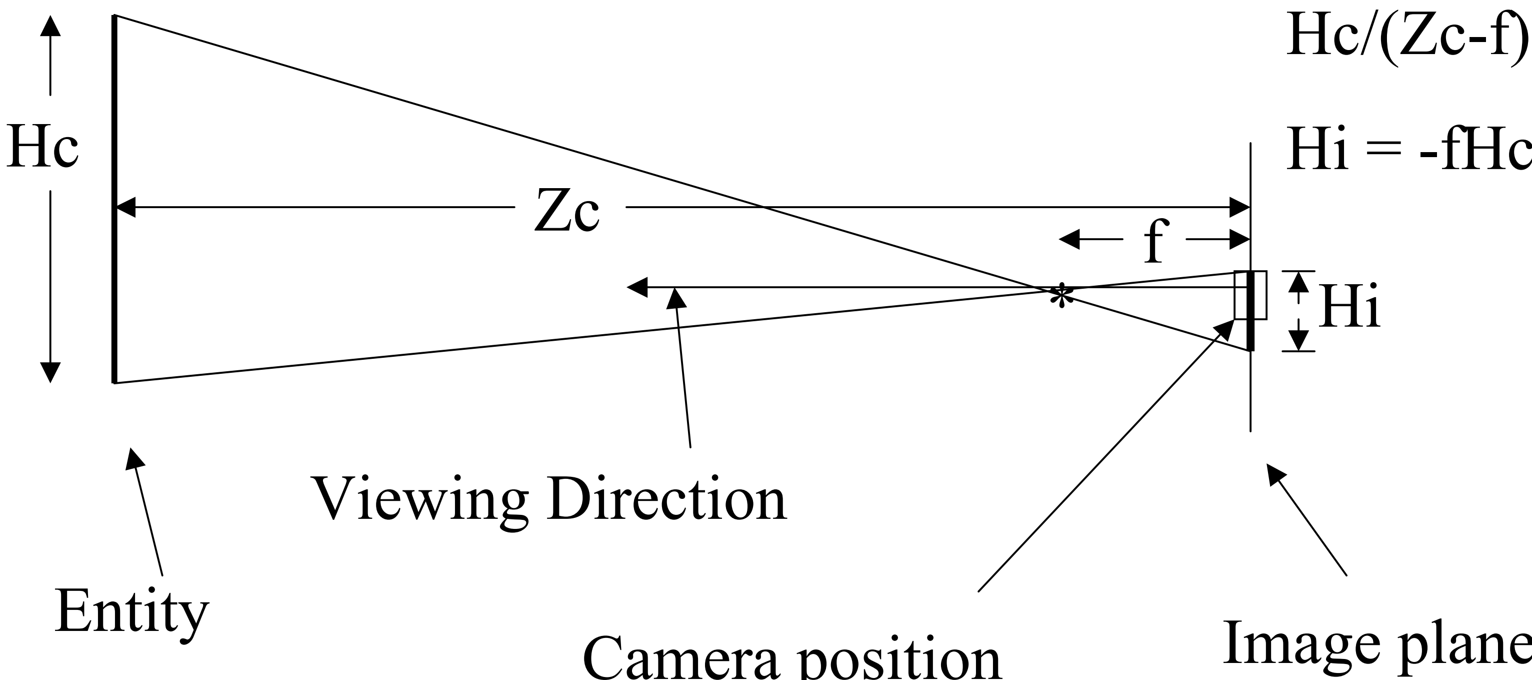
Illustrating imaging process (2D case)

NOTE: Camera position is the position of the image plane



Camera models (Crosby)

Illustrating imaging process (2D case)



By similar triangles:

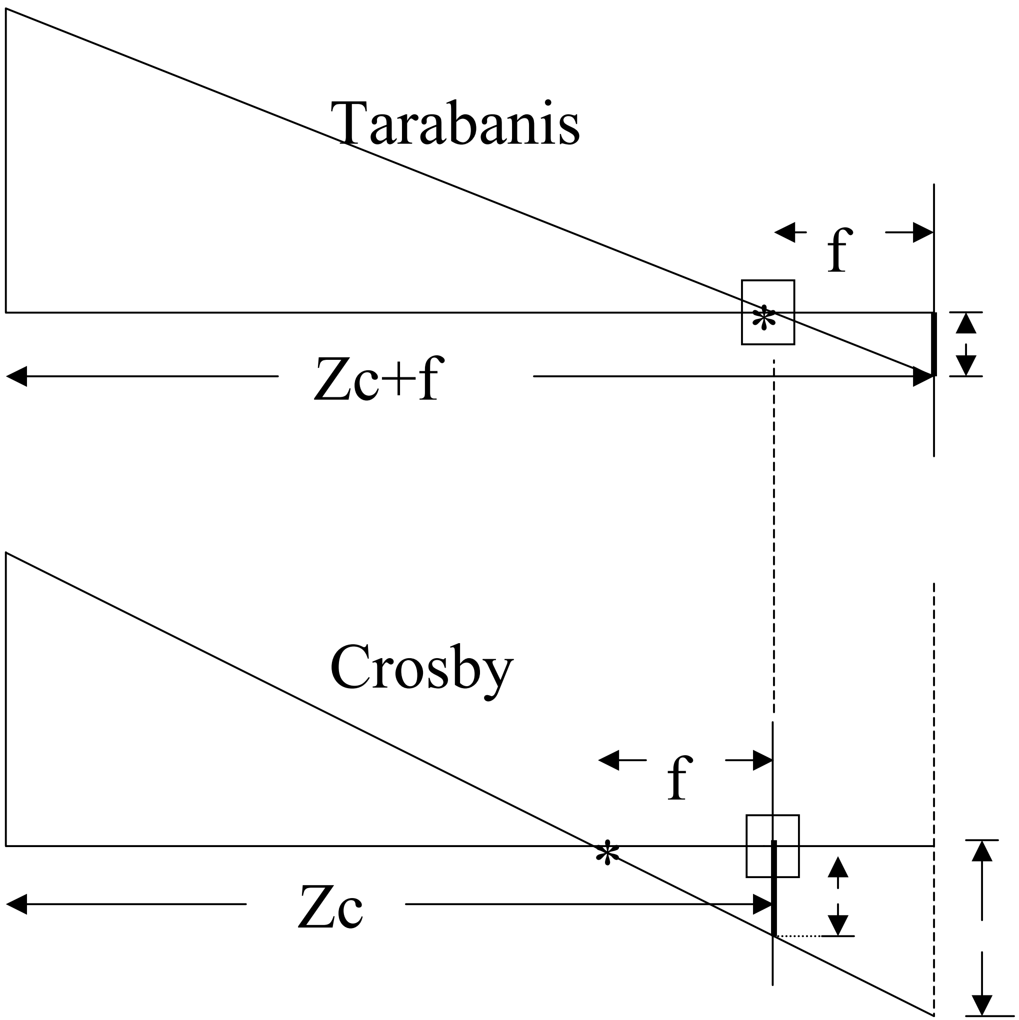
$Hc/(Zc-f)=Hi/-f$

$Hi = -fHc/(Zc-f)$

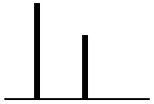
Why is this important?

- The objective functions use Crosby’s model
- The constraint functions use Tarabanis’ model
- CONSEQUENCE:
  - Given the same camera pose, you get different projected entities
  - The constraints and the objective functions are not in sync

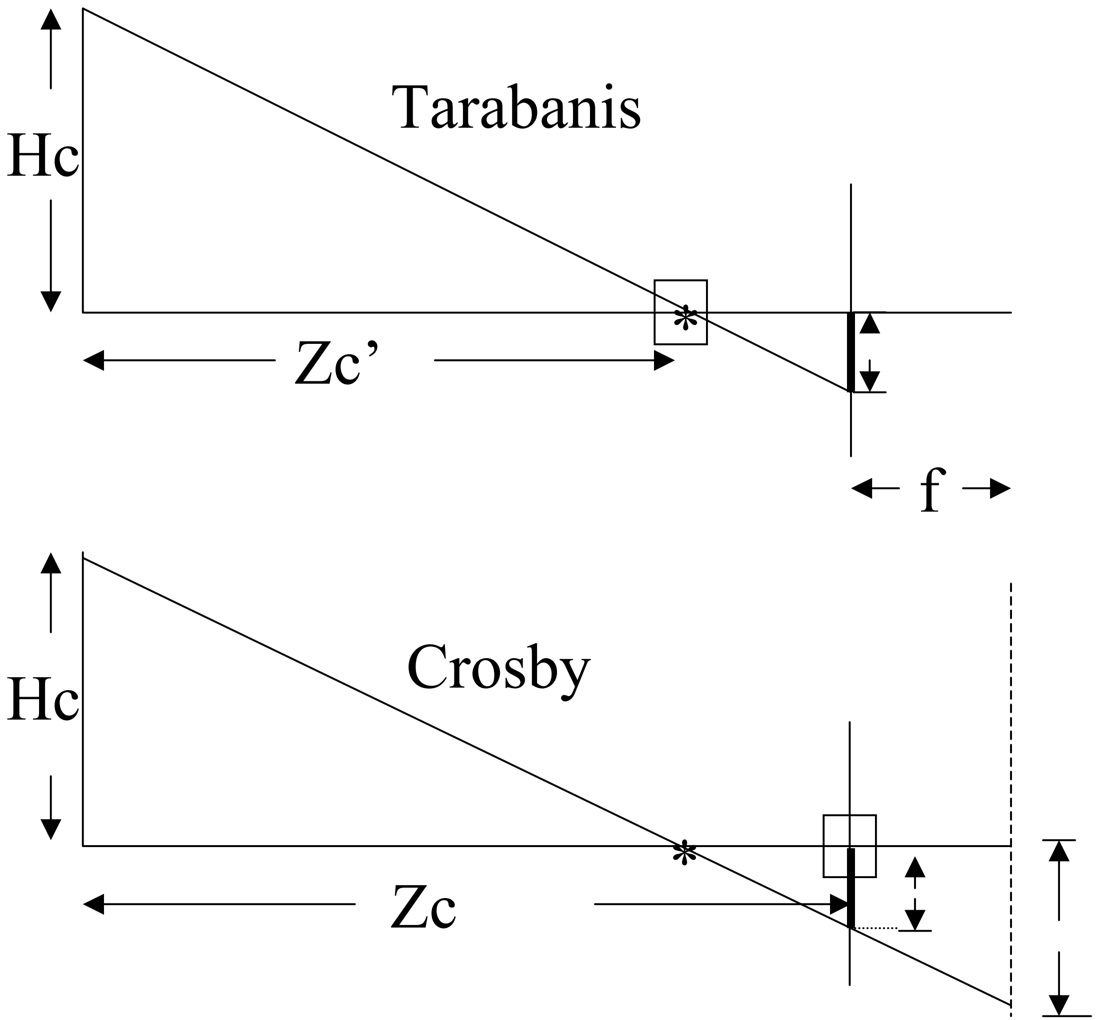
Example



Square is the camera position. Notice the difference between the projections



Example: fixing the problem



Align the image planes by moving Tarabanis' camera pose by the focal length along the viewing direction

Substitute

$$Zc' = Zc - f$$

in Tarabanis model

$Hc/Zc' = Hi/-f$

$Hi = -fHc/Zc'$

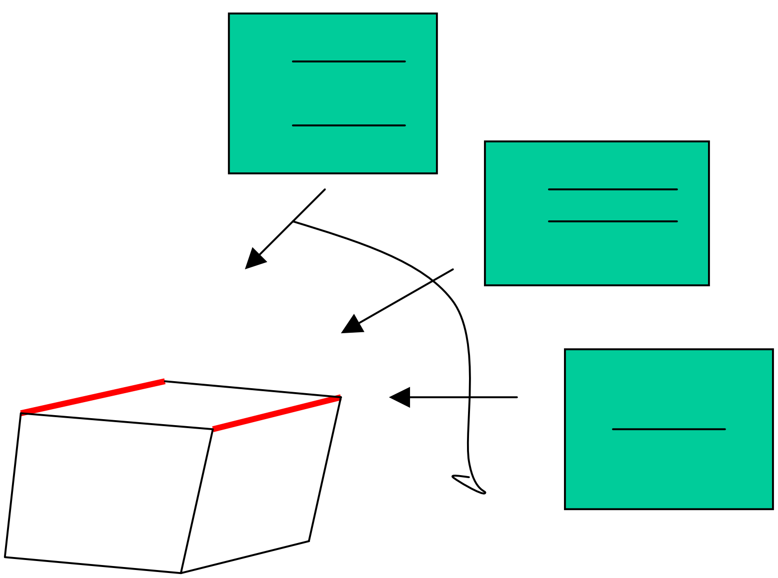
$Hi = -fHc/(Zc-f) =$ Crosby's model

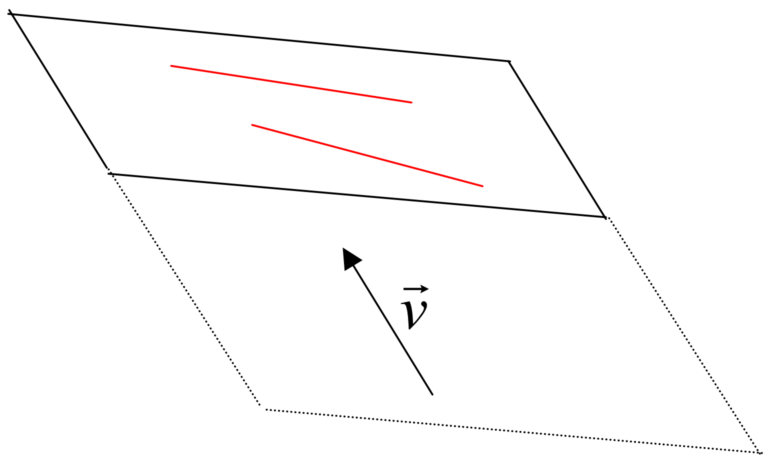
Incidence angle between viewing direction and plane

Fai Lung Shih

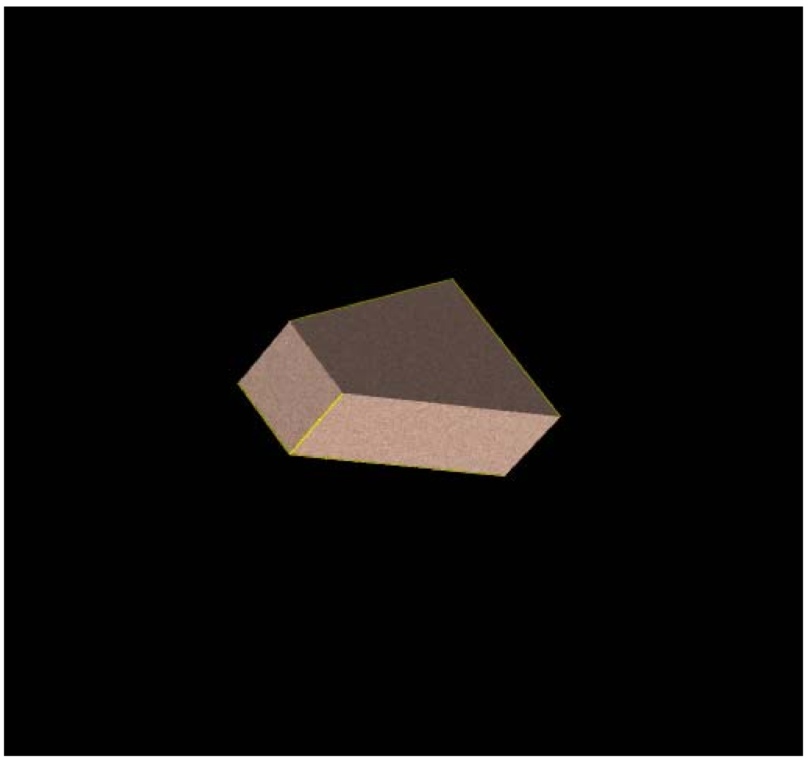
Problem

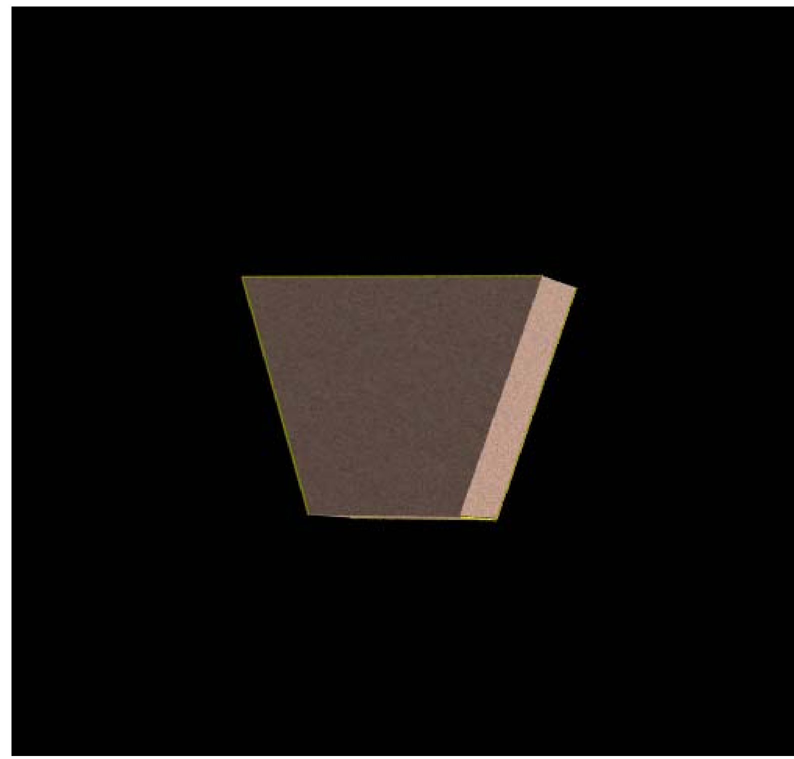
- Object features cannot be measured when viewed from directions approaching coplanar





Example



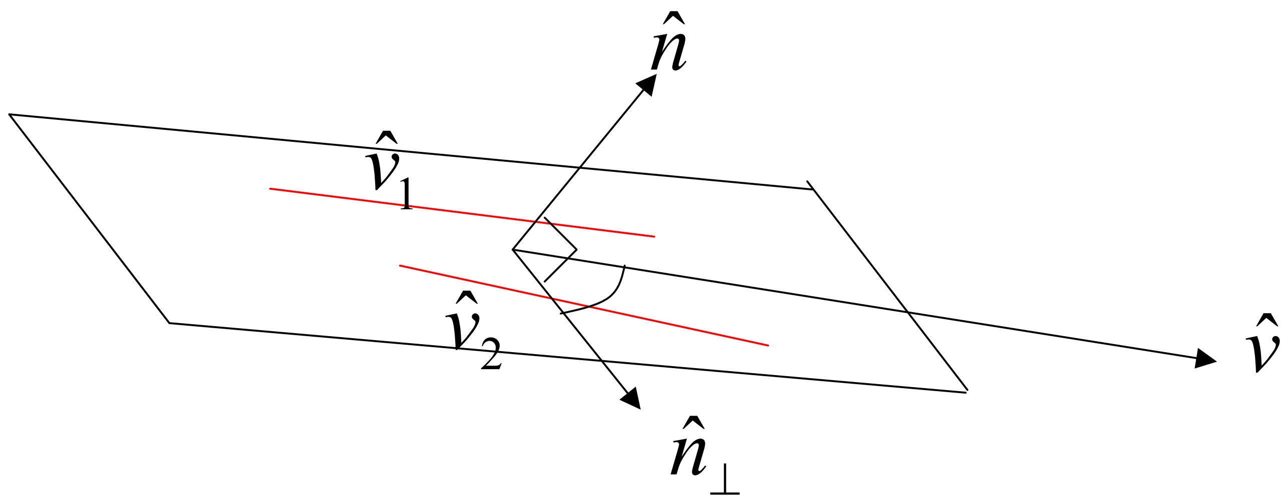


Minimum angle

$$\hat{n} = \hat{v}_1 \times \hat{v}_2$$

$$\sin^{-1}(\hat{n} \cdot \hat{v}) >= \delta_{inc}$$

Where δ_{inc} is the minimum angle between the plane and the viewing direction

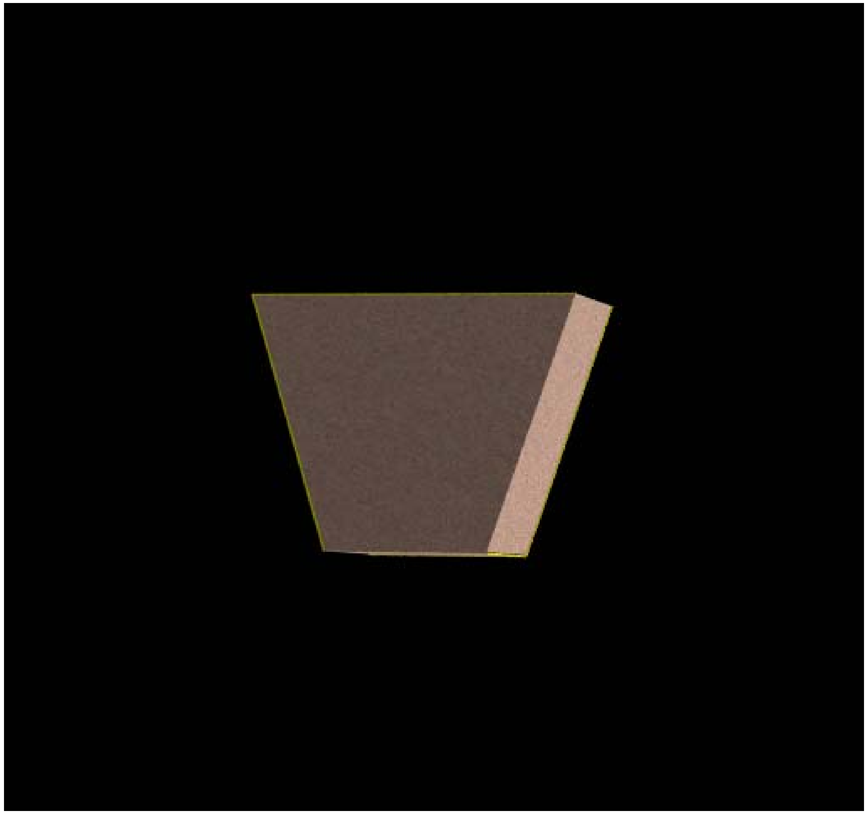


New Nonlinear Program

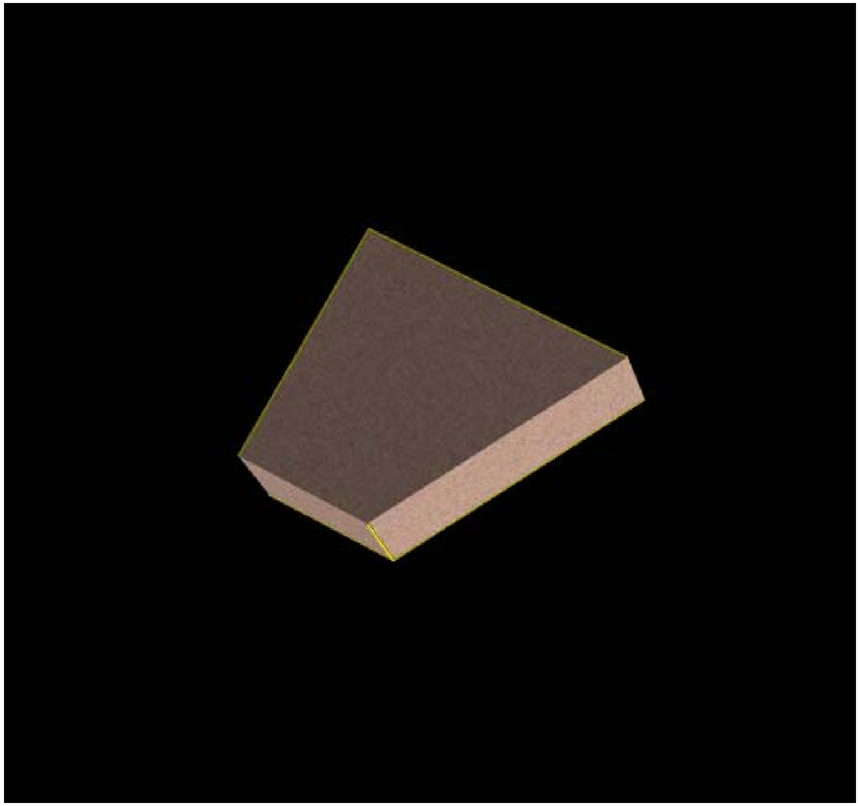
- minimize $E[\varepsilon^2] = f(t_x, t_y, t_z, \Phi, \theta, \Psi)$
  subject to:
  - g1j <= 0 (pixel resolution), for j=1 to k
  - g2a <= 0 (focus)
  - g2b <= 0
  - g3 <= (field of view)
  - g4i <= 0 (visibility), for i=1 to m
  - **g5j <= 0 (incidence angle constraints),**
    for j=1 to $_kC_2$($\frac{k!}{2!(k-2)!}$)

Example

Without incidence angle constraint



With incidence angle constraint



Finding initial feasible pose

Fai Lung Shih

Problem

- In non-linear programming, different starting points result in different optimal solutions

Example

- Ex1
  - Initial pose
    - camera position/orientation: 1216.193661 -954.187115 -1220.729648 0.513804 -0.456819 -0.811894
  - Optimal pose
    - camera position/orientation: 282.725838 -499.690340 -1469.377994 0.927833 0.151262 -0.342271
    - Optimal Value: 0.242909
- Ex2
  - Initial pose
    - camera position/orientation: 499.146112 -680.528149 -1361.056298 0.244979 -0.237941 -0.510331
  - Optimal pose
    - camera position/orientation: 492.216499 -666.899532 -1368.051894 -1.212159 -0.525523 0.145824
    - Optimal Value: 0.245218
- Ex3
  - Initial pose
    - camera position/orientation: 70.213576 -960.212503 -2105.580459 0.026566 -0.026556 -0.428443
  - Optimal pose
    - camera position/orientation: 470.466854 -641.647819 -1384.297776 -1.134582 -0.511611 0.101482
    - Optimal Value: 0.242944

Initial Feasible camera pose

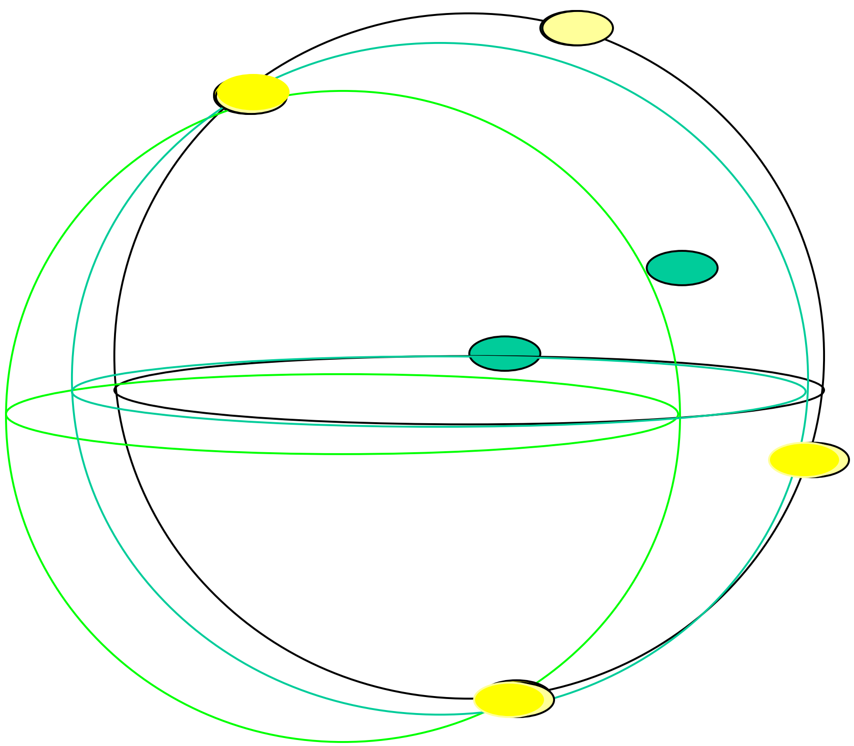
- Outline to determine initial feasible camera pose
  1. Construct the minimum bounding sphere
  2. Determine the minimum distance that satisfies focus and field of view constraints
  3. Determine the viewing direction
  4. Find camera pose that is closest to the viewing direction
  5. Optimize the camera pose

Minimum bounding sphere

- Algorithm(modification of Elzinga-Hearn Algorithm)
  - Input: A set P of 3D points
  - Output: The center and radius of a sphere that covers the points in P, and points Pi, Pj, Pk, and Pl that are on the sphere

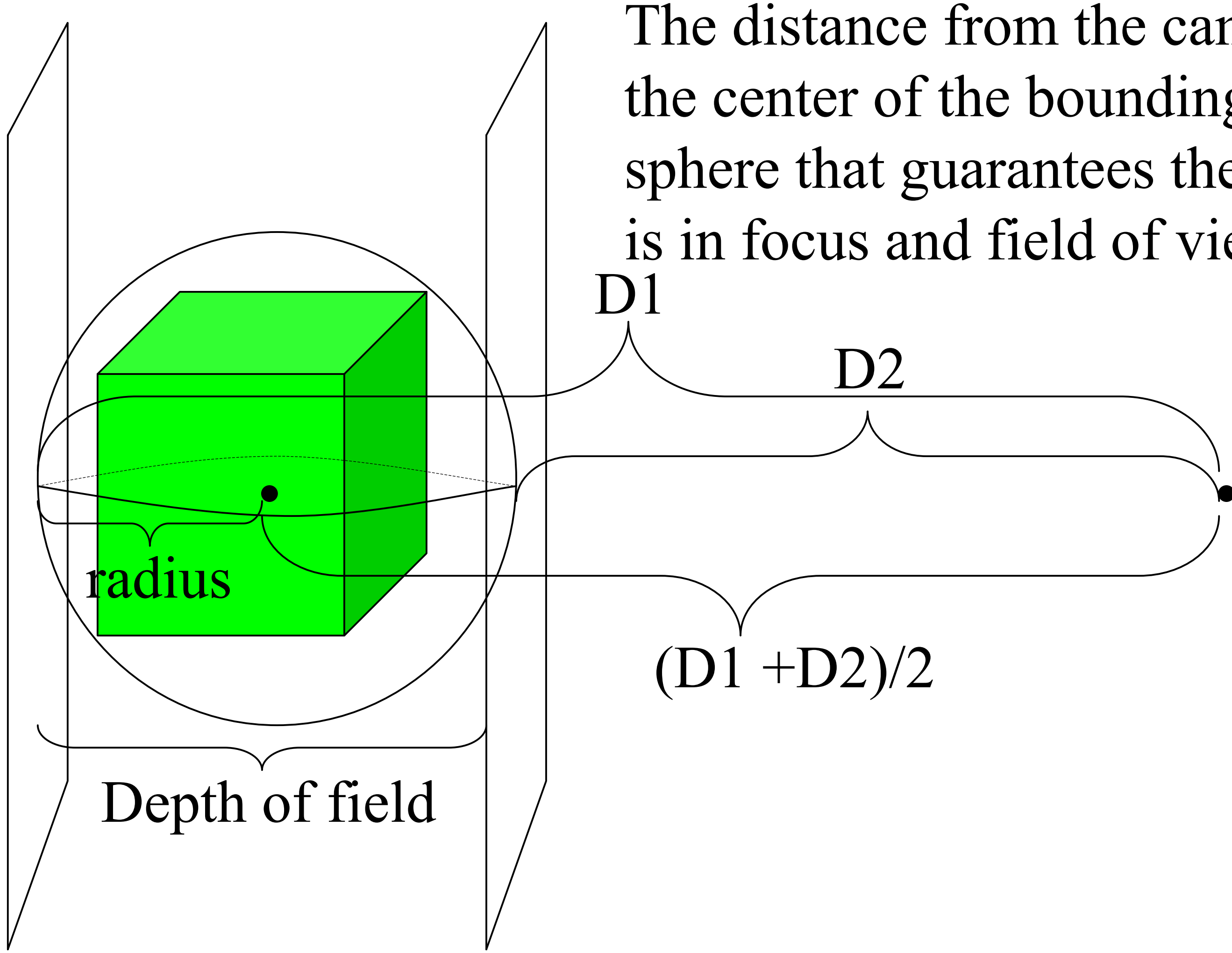
1. Choose two points, P_i and P_j that are farthest apart
2. Construct the sphere whose center $C_2((P_i + P_j)/2)$
   1. If the sphere contain all points, then the center of the sphere is C_2
   2. Else, find a point P_k is the farthest away from C_2
3. Use the three points (P_i ,P_j, P_k) to determine the center C_3
   1. If the sphere contain all points, then the center of the sphere is C_3
   2. Else, find a point P_l is the farthest away from C_3
4. Use the there points (P_i ,P_k, P_l) to determine the center C_{3a}
   1. If the sphere contain all points, then the center of the sphere is C_{3a}
   2. Else, use the there points (P_j ,P_k, P_l) to determine the center C_{3b}
      1. If the sphere contain all points, then the center of the sphere is C_{3b}
      2. Else, use the four points (P_i , P_j ,P_k, P_l) to construct the sphere whose center is C_4

Example



Minimum Distance

The distance from the camera to the center of the bounding sphere that guarantees the object is in focus and field of view



Minimum focus distance

$$D_1 = \frac{Daf}{af - c(D - f)}$$

$$D_2 = \frac{Daf}{af + c(D - f)}$$

$$Depth \quad of \quad field(DOF) = D_1 - D_2$$
$$= 2r$$

Where:

D : focus distance

a : lens's aperture

f : focal length

c : blur circle radius

r : radius of minimum bounding sphere

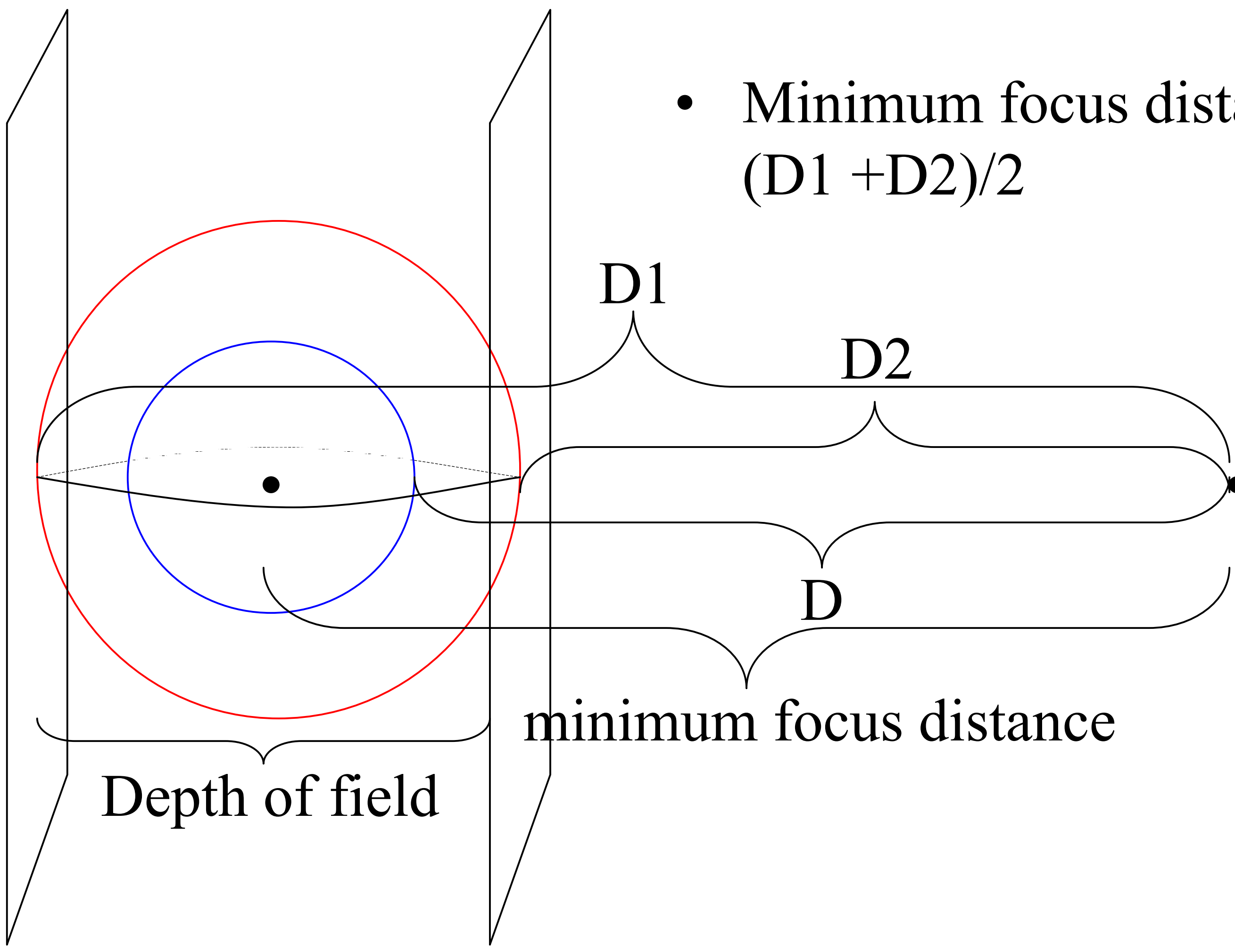
$$2r = Daf\left(\frac{1}{af - c(D - f)} - \frac{1}{af + c(D - f)}\right)$$

$$D^2(2afc + 2rc^2) + D(-2af^2c - 4rfc^2) + 2rf^2(c^2 - a^2) = 0$$

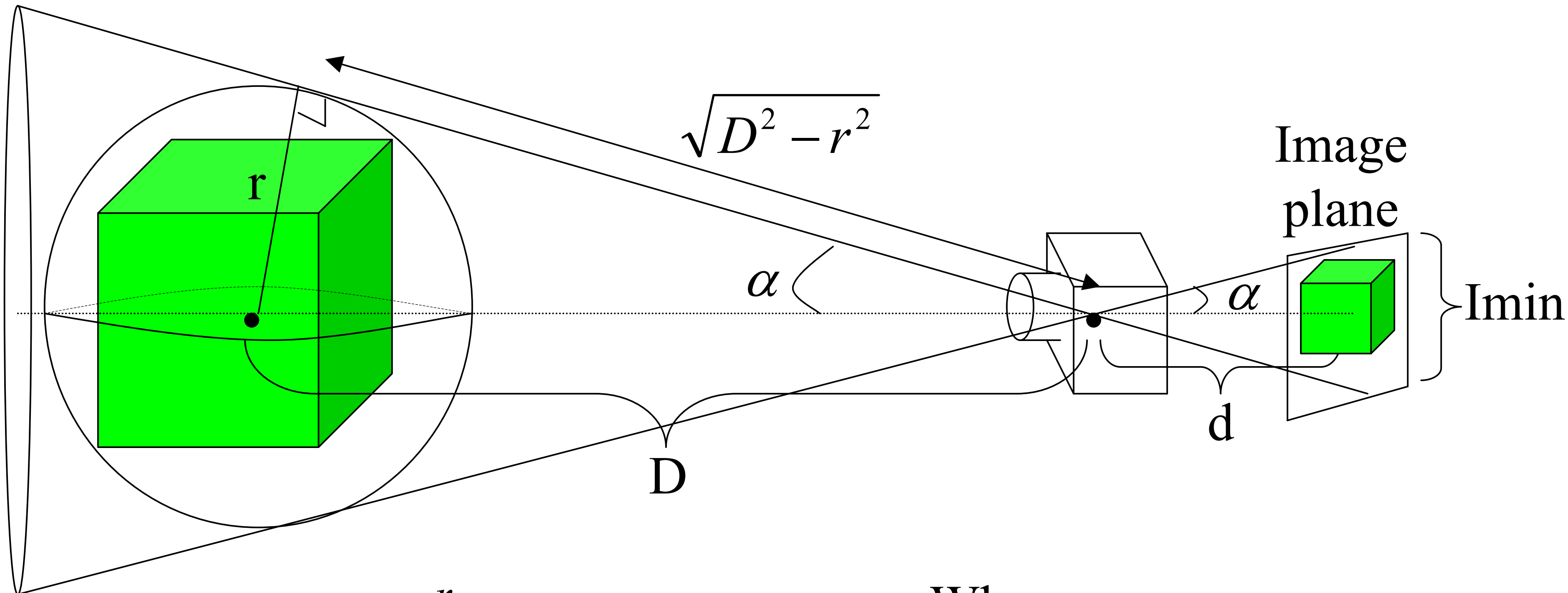
$$D = \frac{2acf^2 + 4rfc^2 + \sqrt{(-2acf^2 - 4rfc^2)^2 - 16rcf^2(af + rc)(c^2 - a^2)}}{4afc + 4rc^2}$$

Minimum focus distance Cont.

- Minimum focus distance = (D1 +D2)/2



Minimum field of view distance



$$\tan\alpha = \frac{r}{\sqrt{D^2 - r^2}}$$

$$\tan\alpha = \frac{I\min}{2d}$$

Where:

d : effective focal length

α : field of view angle

Minimum field of view distance Cont.

$$\frac{r}{\sqrt{D^2 - r^2}} = \frac{\mathrm{I}\min}{2d}$$

$$\frac{1}{d} + \frac{1}{D} = \frac{1}{f}$$

$$\frac{r}{\sqrt{D^2 - r^2}} = \frac{\mathrm{I}\min(D - f)}{2fD}$$

$$(D^2 - r^2)(D - f)^2 \mathrm{I}\min^2 = 4f^2 D^2 r^2$$

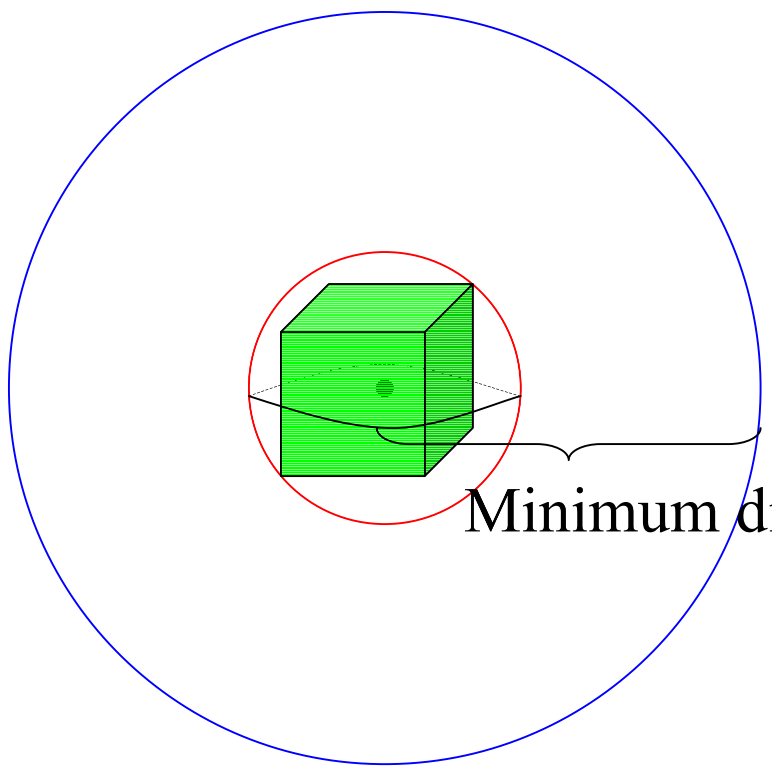
$$D^4 - D^3 2f - D^2 (r^2 + \frac{4f^2 r^2}{\mathrm{I}\min^2} - f^2) + D2fr^2 - f^2 r^2 = 0$$

Minimum field of view distance = D

Minimum Distance

Minimum distance = max {minimum focus distance,
minimum field of view distance}

Any camera position outside the blue circle, satisfies the field of view and focus constraints



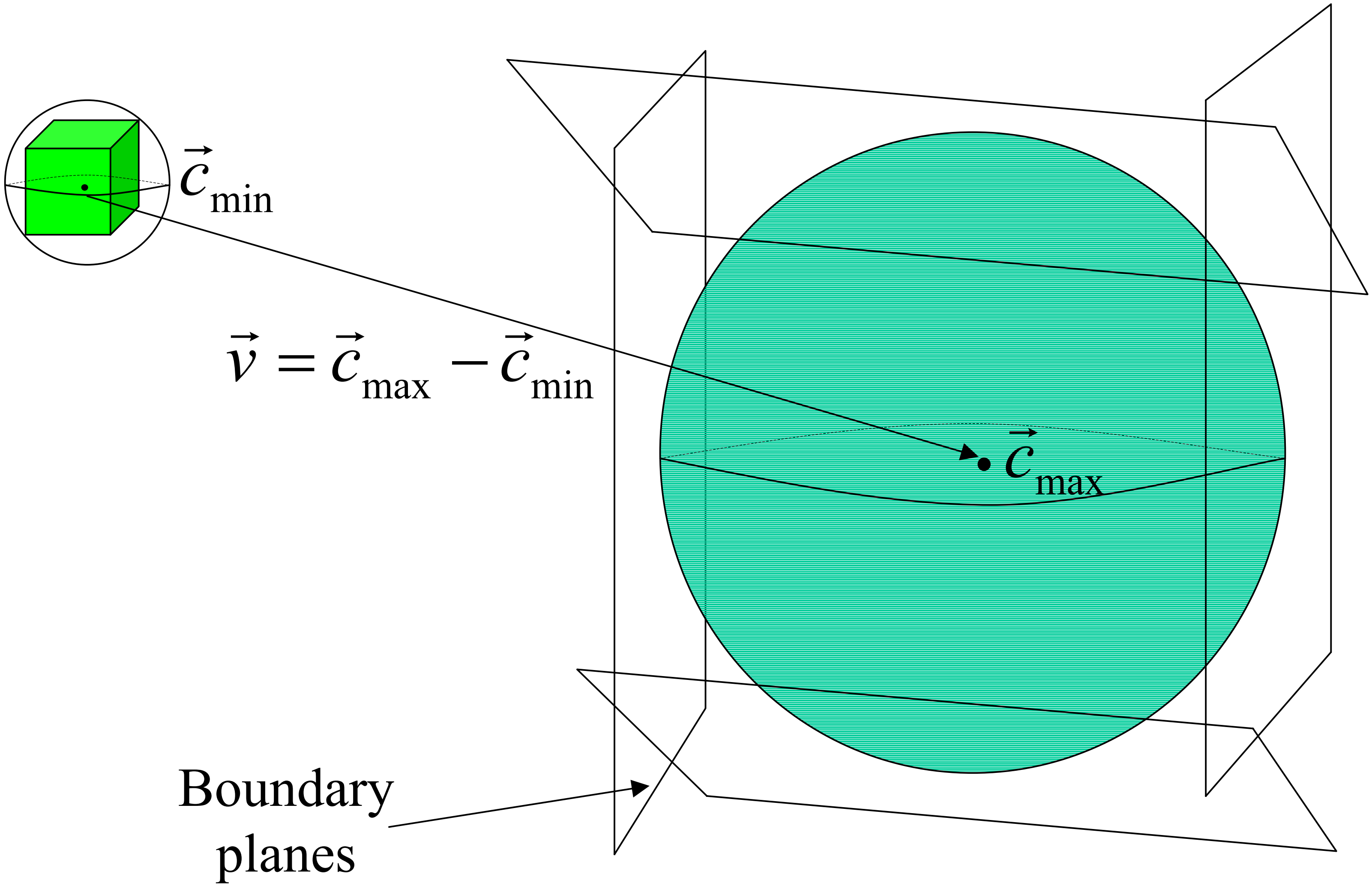
Viewing Direction

- Three approaches
  - Center-center
  - Extreme vertices
  - Average vertices

Center-center approach

- Construct a maximum sphere that is bounded by the visibility planes
- The vector from the maximum sphere center and the minimum sphere center is the viewing direction

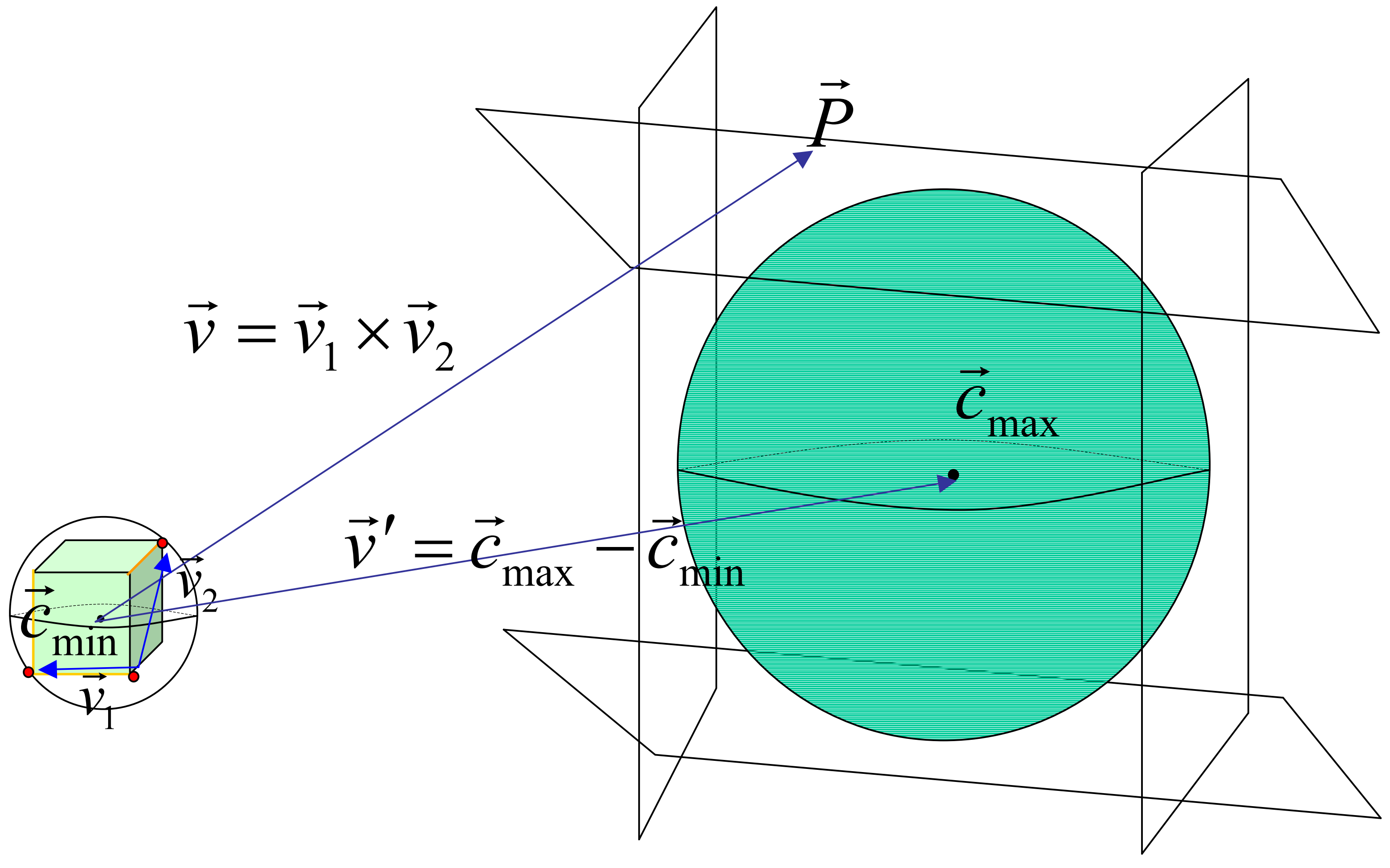
Center-center approach



Extreme vertices approach

- Construct a maximum sphere that is bounded by the visibility planes
- Use the points (Pi, Pj, Pk, Pl) on the minimum sphere to create a viewing direction by calculating the normal of the plane formed by those points

Extreme vertices approach



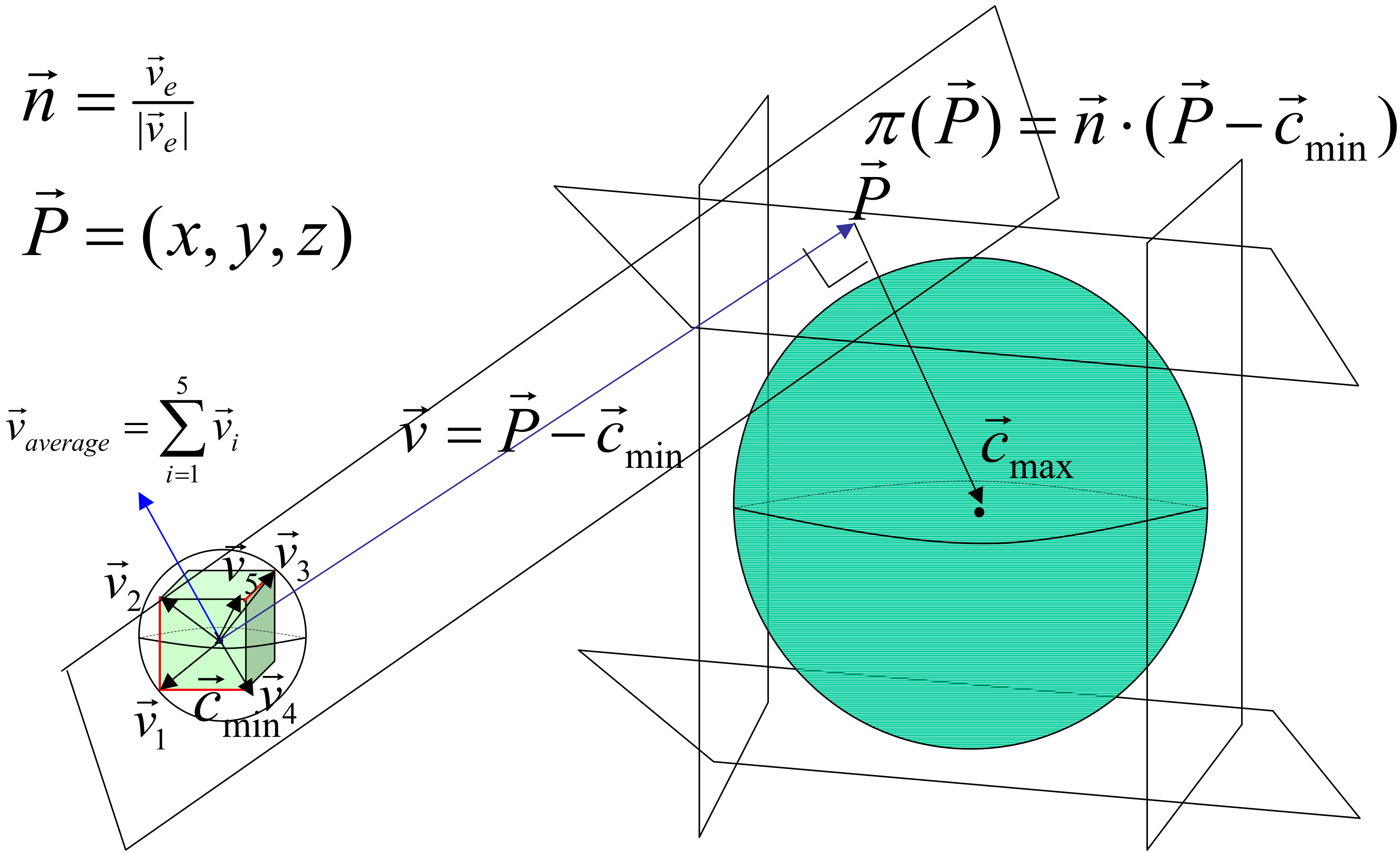
Average vertices approach

- Construct a maximum sphere that is bounded by the visibility planes
- Calculate a vector Vavg that is the sum of the entities to be observed
- Define a plane whose normal is Vavg
- The viewing direction is the vector that is formed by the projection of the maximum sphere center point on that plane

Average vertices approach

$$\vec{n} = \frac{\vec{v}_e}{|\vec{v}_e|}$$

$$\vec{P} = (x, y, z)$$



Average vertices approach cont.

$$\left.\begin{aligned}\vec{v}_{average} \cdot (\vec{P} - \vec{c}_{\min}) = 0 \\ \vec{v}_{average} \times (\vec{c}_{\max} - \vec{P}) = 0\end{aligned}\right\} \vec{P}$$

$$\vec{v} = \vec{P} - \vec{c}_{\min}$$

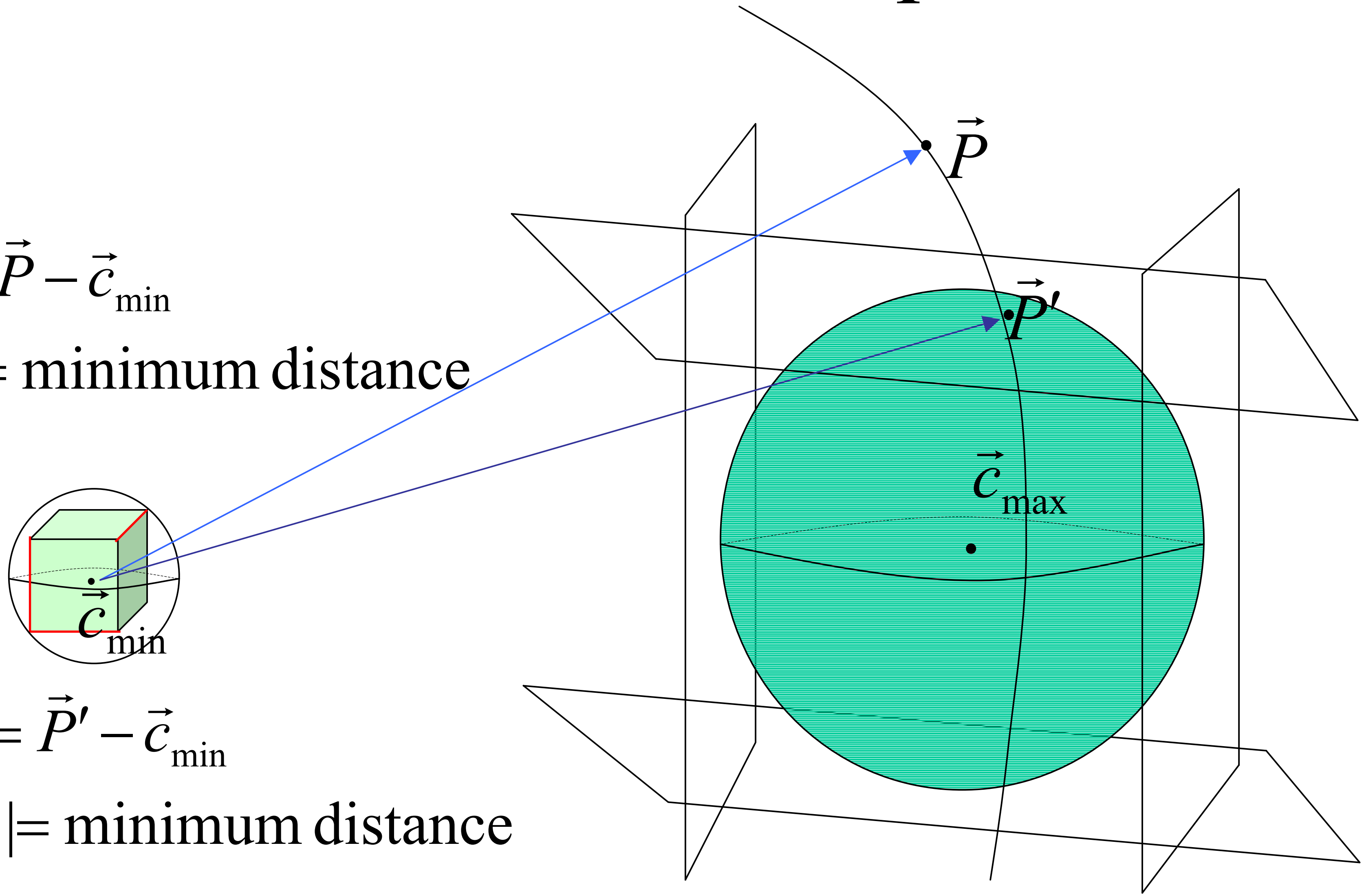
Effective camera pose

- The intersection of the minimum distance and the viewing direction gives you a feasible camera pose. We assume this is the best pose.
- Sometimes this pose is outside the visibility volume
- Need a way to find a new pose as close as possible to the original pose that is inside the visibility volume

Effective camera pose

$$\vec{v} = \vec{P} - \vec{c}_{\min}$$

$$|\vec{v}| = \text{minimum distance}$$



$$\vec{v}' = \vec{P}' - \vec{c}_{\min}$$

$$|\vec{v}'| = \text{minimum distance}$$

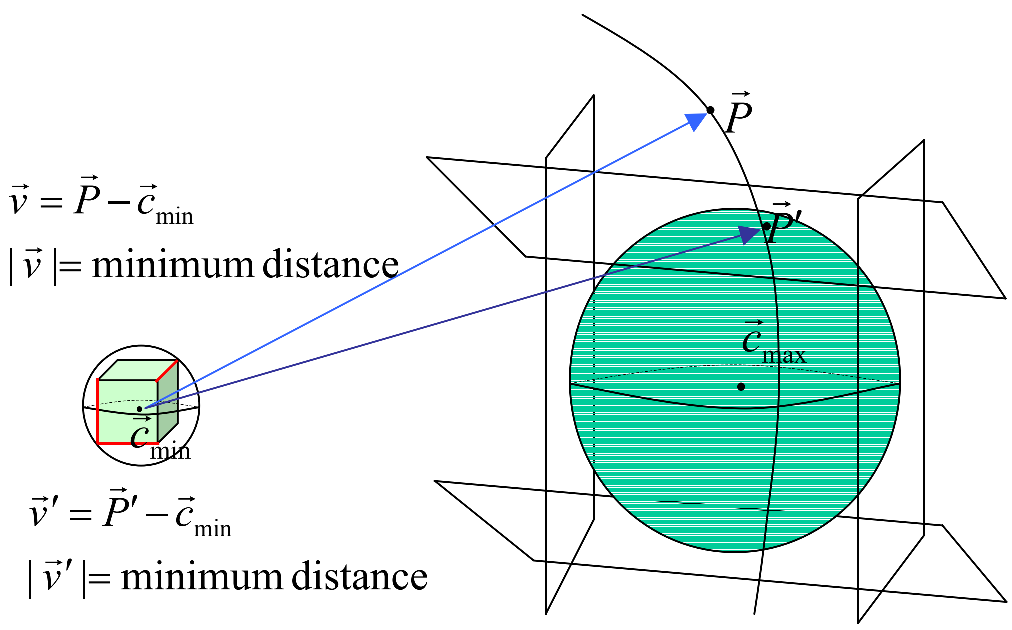
Effective camera pose cont.

- Find camera pose ($\vec{P}'$) is most close the viewing direction
- Two approaches:
  - – Nonlinear program
  - – Midpoint subdivision

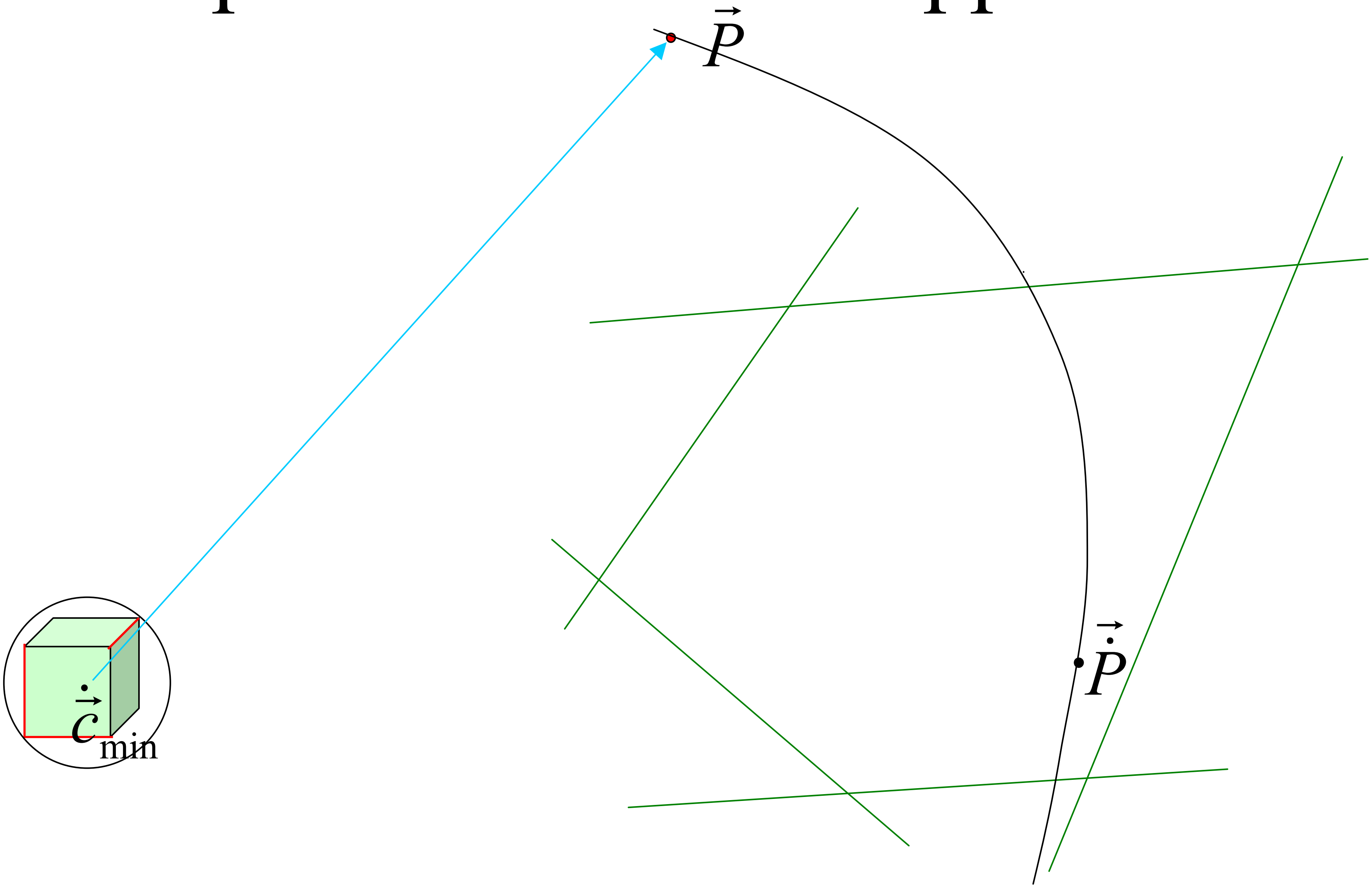
Nonlinear program approach

- minimize $|\vec{P} - \vec{P}'| = f(\vec{P}, \vec{P}')$

  subject to:
  - $P_i \cdot \vec{P}' >= 0$ Where $P_i(x,y,z) = A_i x + B_i y + C_i z + D_i = 0$
  - $|\vec{c}_{\min} - \vec{P}'| >=$ minimum distance



Midpoint subdivision approach

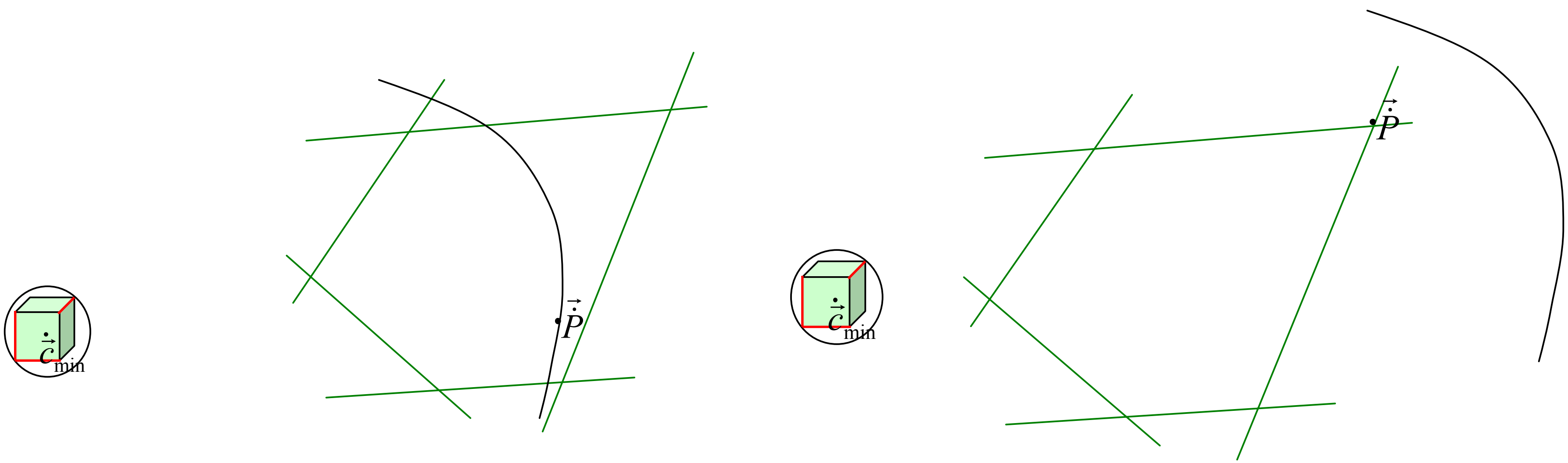


Midpoint subdivision approach cont.

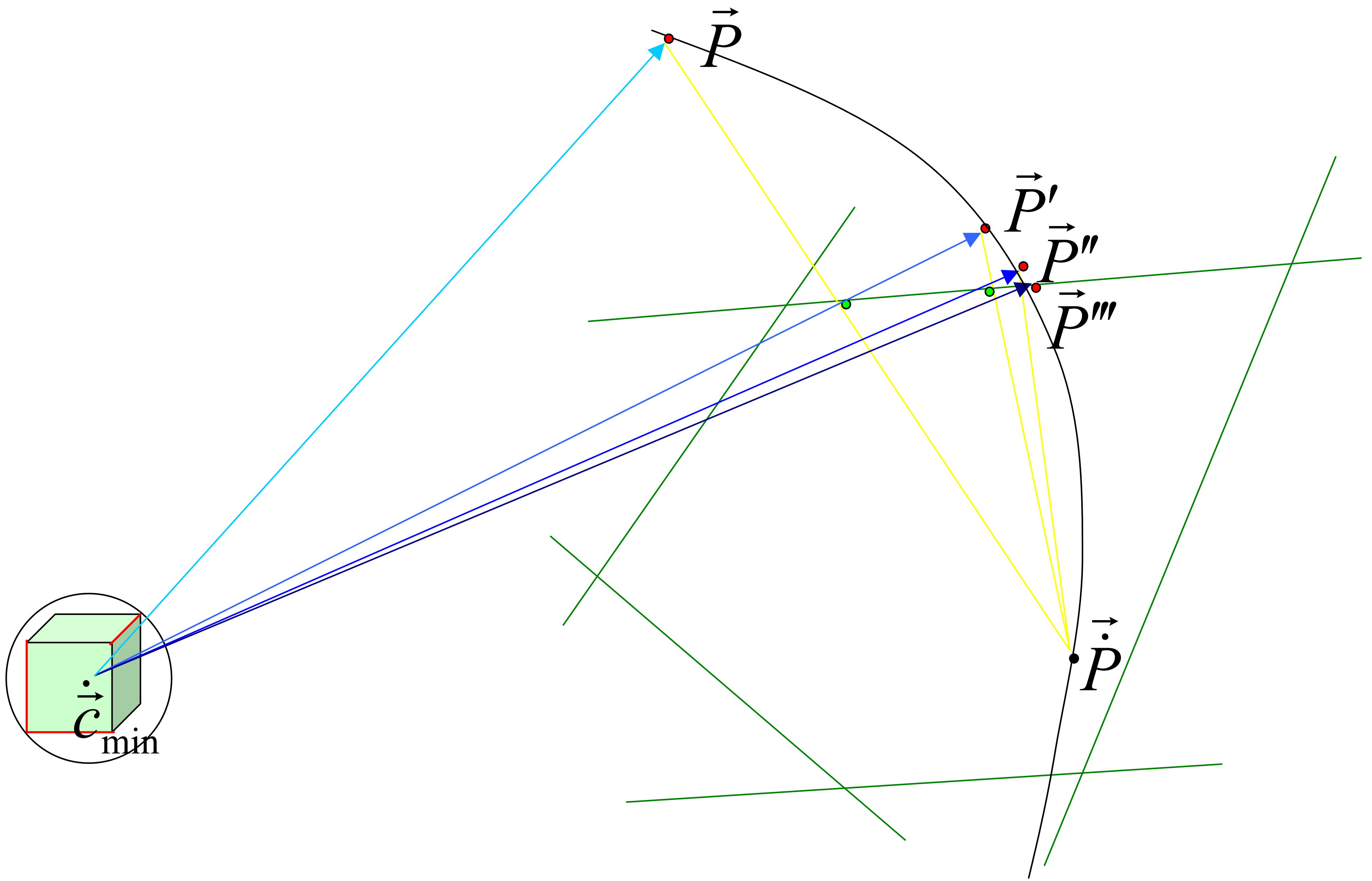
- maximize $|\dot{\vec{P}} - \vec{c}_{\min}| = f(\dot{\vec{P}}, \vec{c}_{\min})$

  subject to:

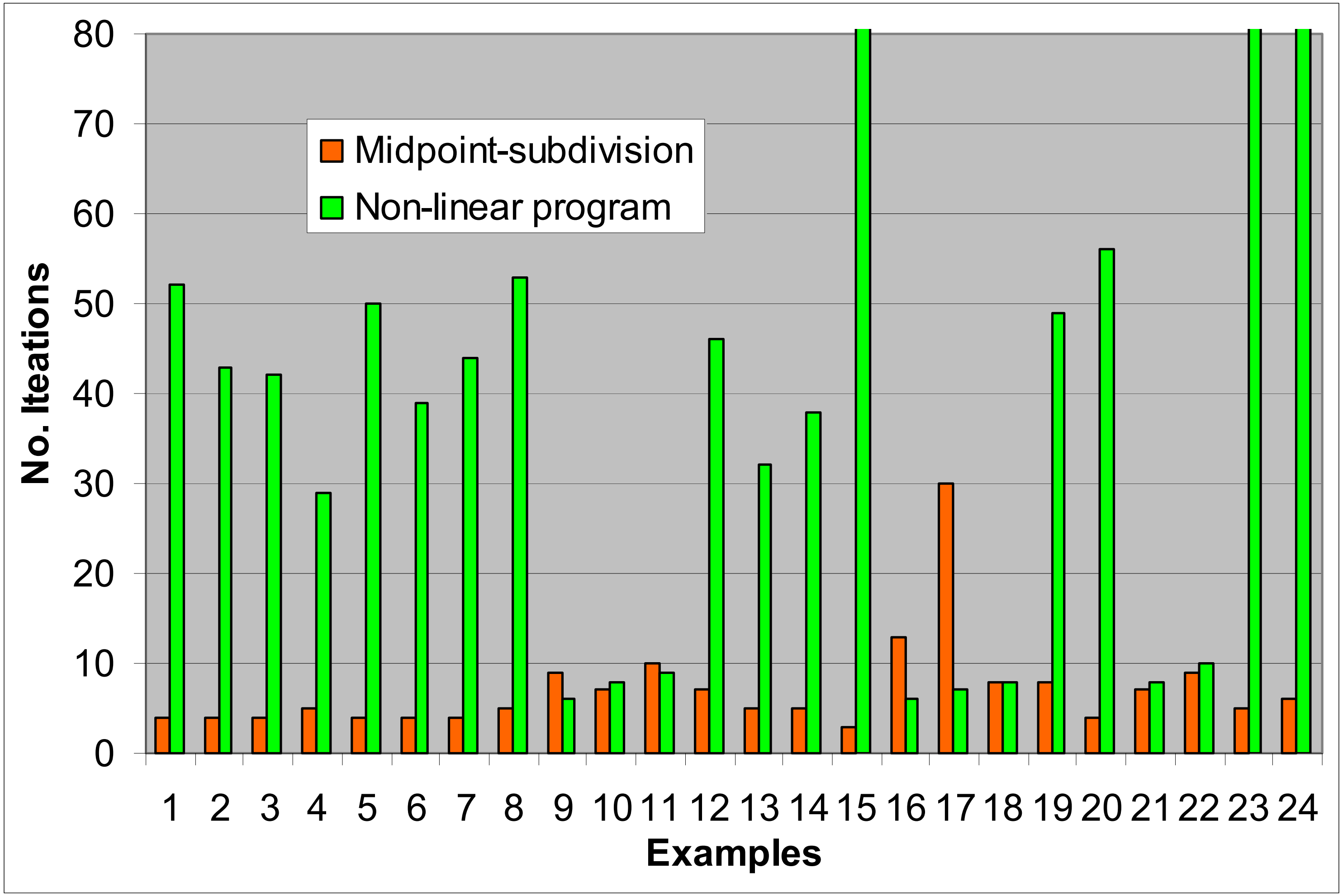
  - $P_i \cdot \dot{\vec{P}} >= 0$ Where $P_i(x,y,z) = A_i x + B_i y + C_i z + D_i = 0$
  - minimum distance - $|\vec{c}_{\min} - \dot{\vec{P}}| >= 0$



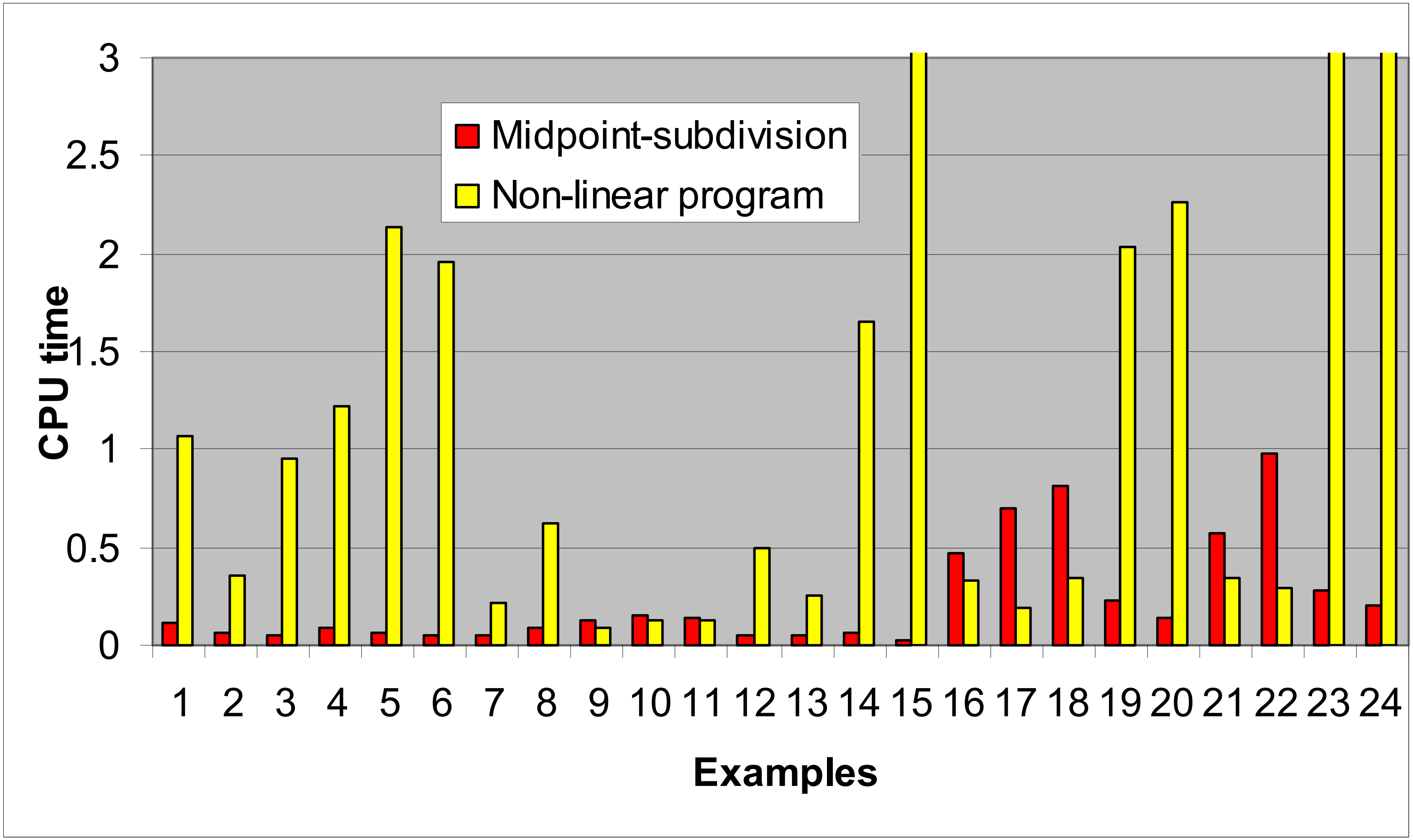
Midpoint subdivision approach Cont.



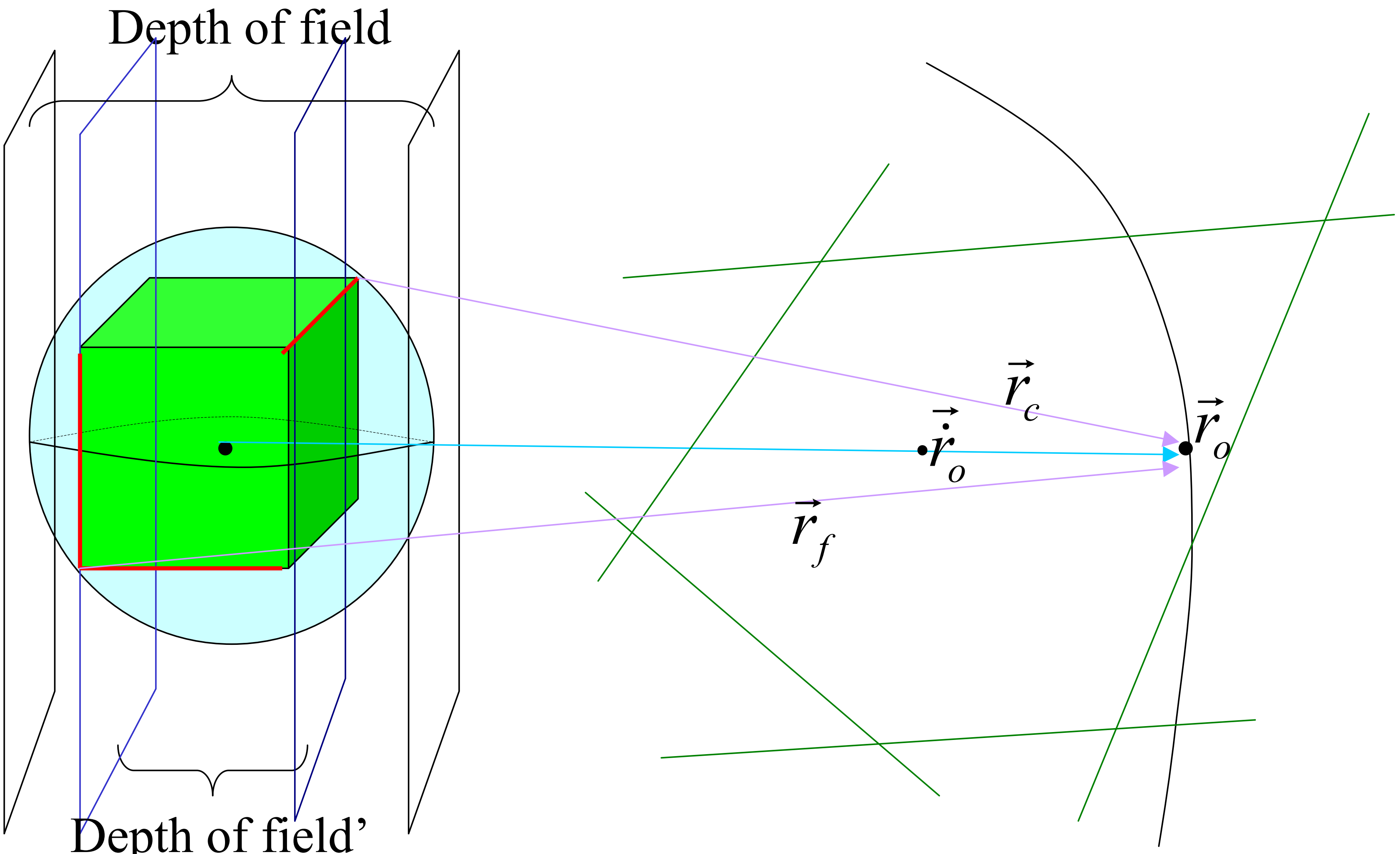
Comparison between the two approaches



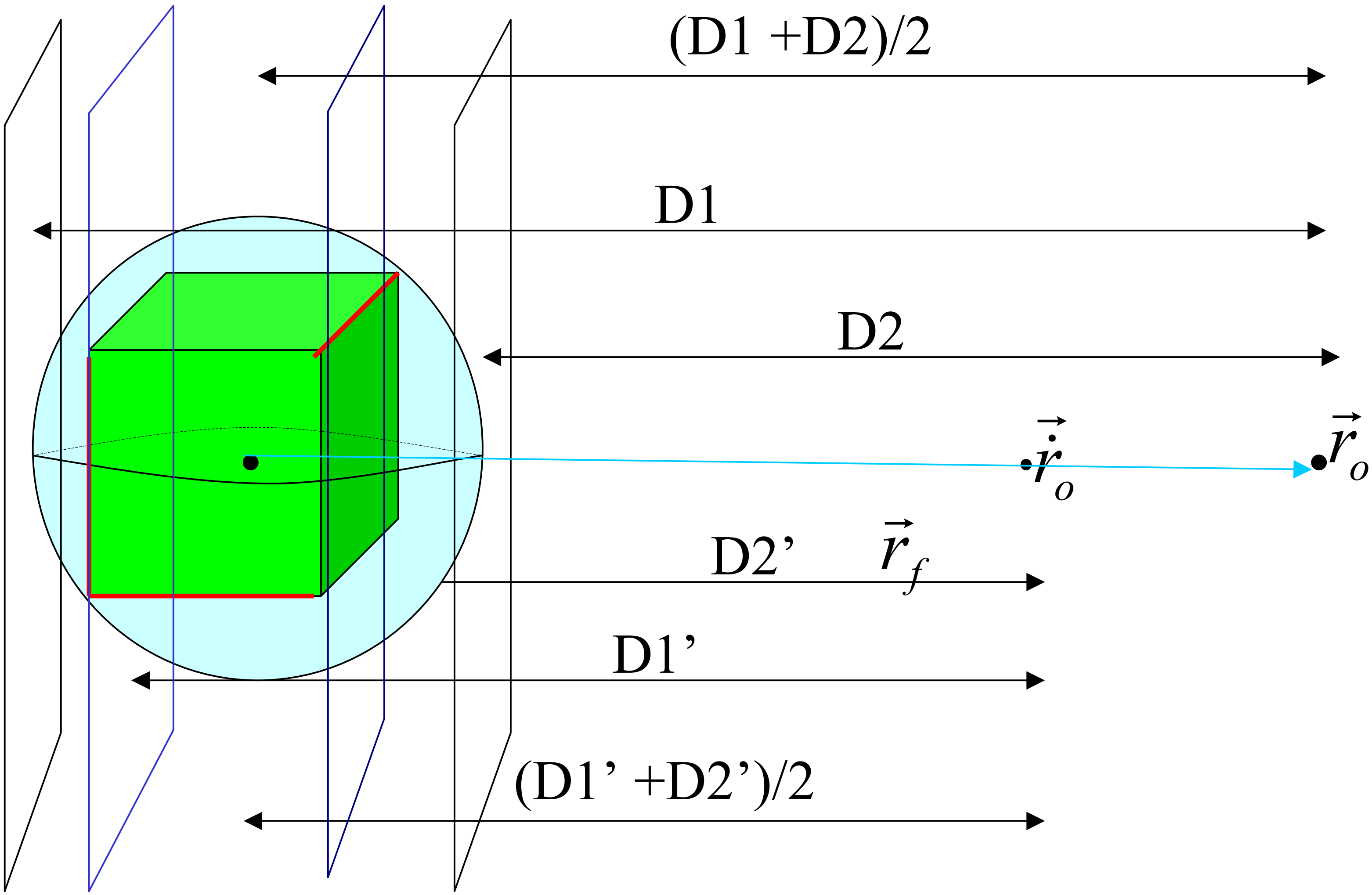
Comparison between the two approaches cont.



Optimize camera pose



Optimize camera pose



Optimize camera pose Cont.

$$f_a : D_2 - (\vec{r}_c - \vec{r}_o) \cdot \vec{v}$$

$$f_b : (\vec{r}_f - \vec{r}_o) \cdot \vec{v} - D_1$$

$$f = \min(f_a, f_b)$$

$$\text{Depth of field}' = 2(r - f)$$

$$D' = \frac{2acf^2 + 4(r-f)fc^2 + \sqrt{(-2acf^2 - 4(r-f)fc^2)^2 - 16(r-f)cf^2(af + (r-f)c)(c^2 - a^2)}}{4afc + 4(r-f)c^2}$$

Example (without optimization)

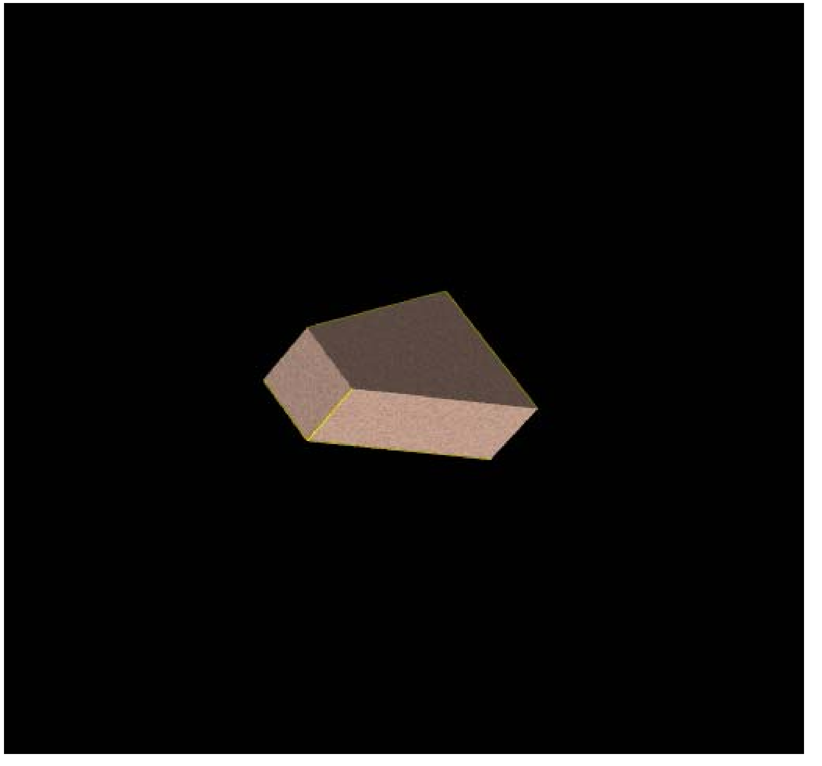
- camera position/orientation:

1426.663466 -1117.801611 -1430.048201
0.513804 -0.456819 -0.811894
viewing direction: -0.6210 0.4827 0.6176
Optimal Value: 0.588711
fov angle: 11.6934
Dmax: 2.3313e+003
effective focal length: 25.2783
aperture: 25
depth of field: 2355.460024 2192.053602
Constraints:

……

-21.318054 } focus constraints
-24.128032 }
-0.004440 } field of view constraint

……



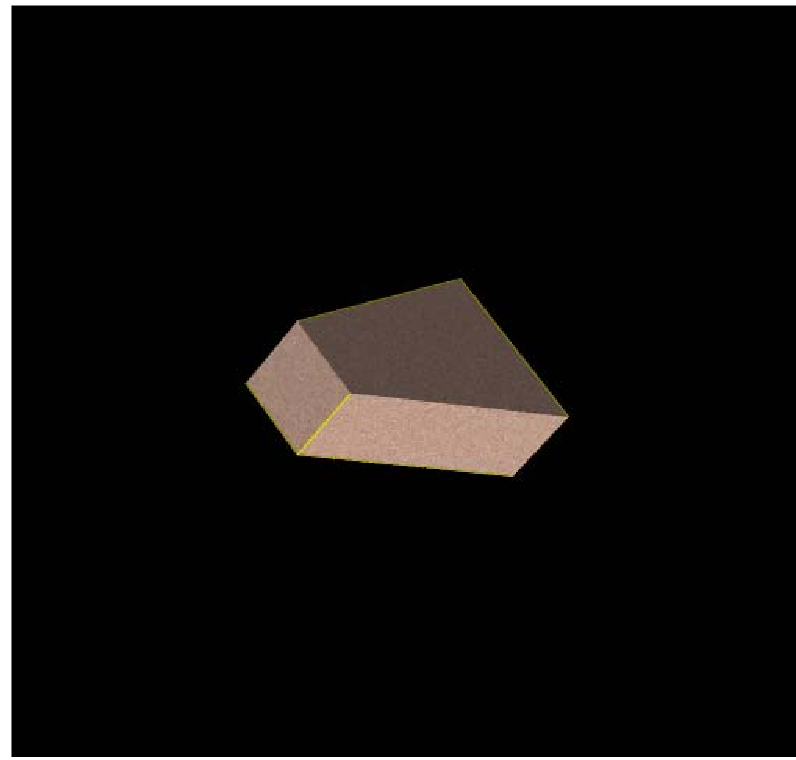
Example (with optimization)

- camera position/orientation: 1216.193661 -954.187115 -1220.729648 0.513804 -0.456819 -0.811894
  viewing direction: -0.6210 0.4827 0.6176
  Optimal Value: 0.431100
  fov angle: 11.6934
  Dmax: 1.9924e+003
  effective focal length: 25.3278
  aperture: 25
  depth of field: 1992.390260 1874.429919
  Constraints:

  ……

  -0.000002 } focus constraints
  -0.000002
  -0.003666 } field of view constraint

  ……



Simulation Results

- Trapezoid
- One-hole object

So far

- **Very brief summary of what is visual inspection planning**
- **Camera models**
- **Incidence Angle**
- **Feasible Pose Determination**
- Issues with intrinsic errors model
- Total Percent Error
- Robustness Index

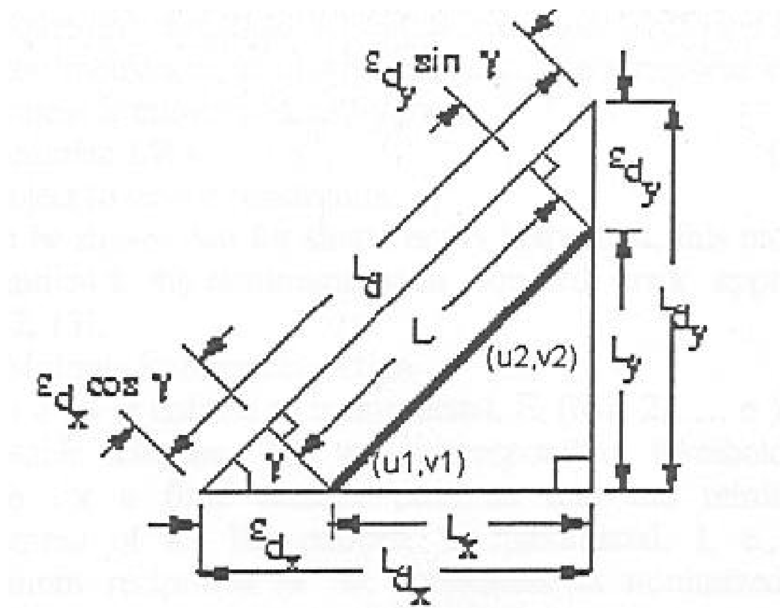
Issues with the error model

Alexis Rivera

Displacement Errors

- Two types:
  - Rotation errors -change the viewing direction
  - Translation errors -change the camera position
- Represented by independent Gaussian variables with zero mean and some variance
- Introduce errors in the projected length of the entity. The error is Gaussian with some mean and variance (Crosby)

Approximate Displacement error



- Dimensional error is geometrically approximated:

  $\varepsilon_d \approx \varepsilon_{dx}\cos(\gamma) + \varepsilon_{dy}\sin(\gamma)$

  γ = angle between line

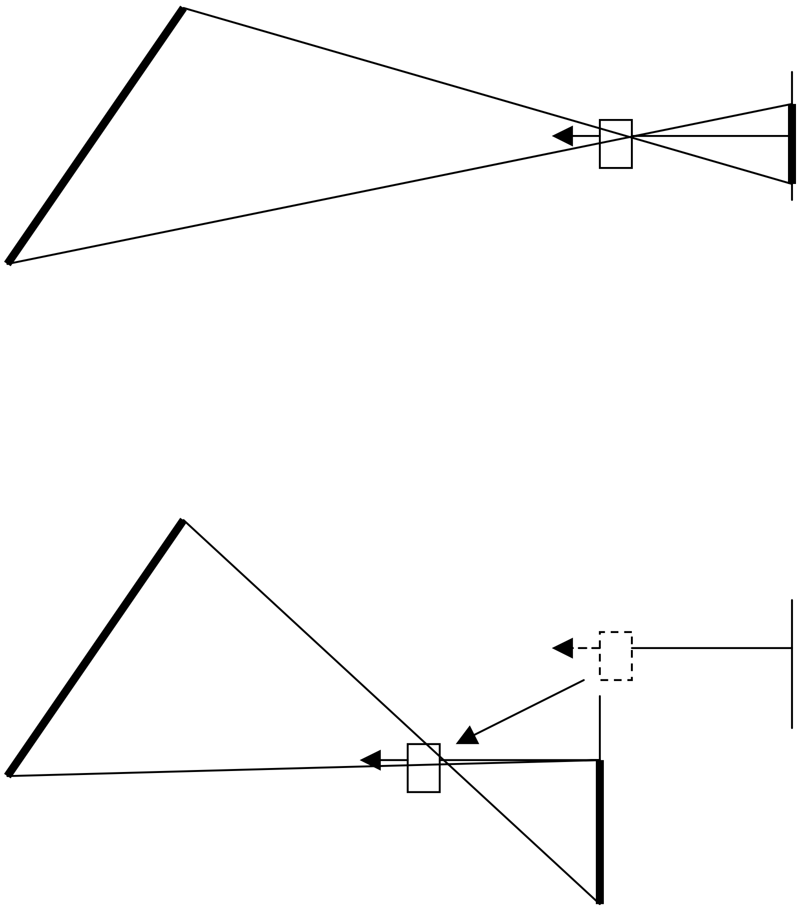
Displacement error of k lines

- Total dimensional error for k lines is:

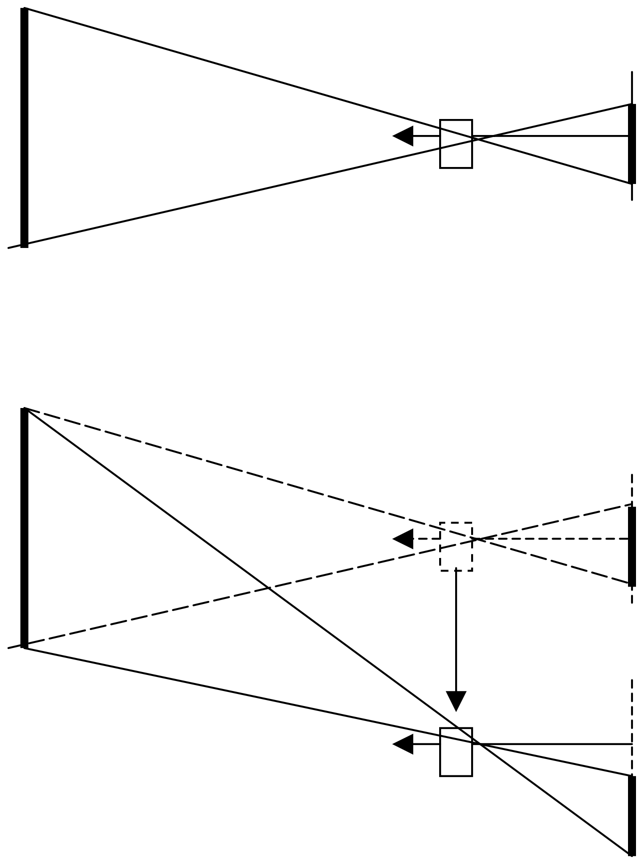
$$\varepsilon_d = \sum_{j=1}^{k} \varepsilon_{d_j}$$

$$E[\varepsilon_d^2] = \sigma_{\varepsilon_d}^2 + \eta_{\varepsilon_d}^2$$

Translation errors

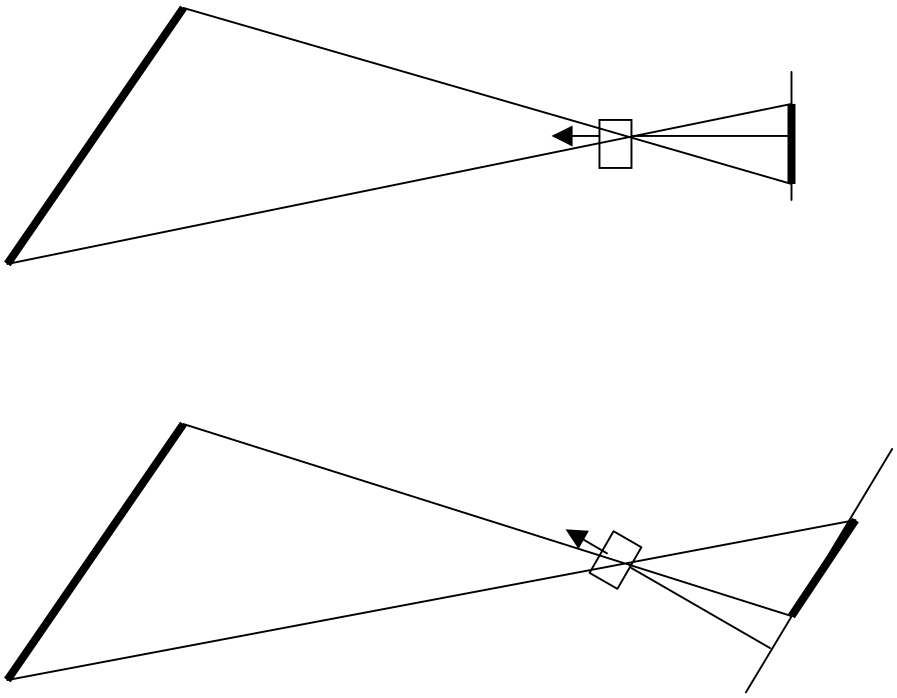


Translation errors

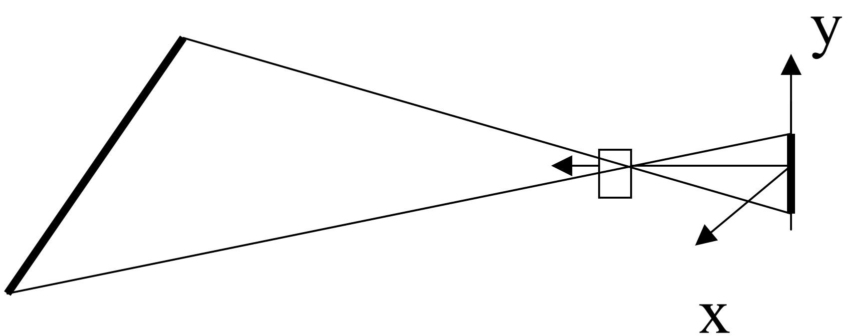


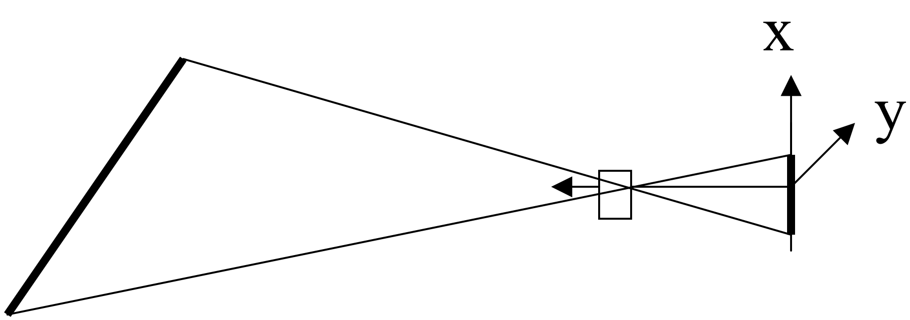
- Translations that are parallel to the entity direction preserve the projected length!!
- The projected length error is zero!!! Independently of the variance of the position error
- Ie. The variance of the projected length error is zero

Rotational errors



Rotational errors





- Rotations around the viewing direction preserve the projected length!!
- The projected length error is zero!! Independently of the variance in the rotation error
- Ie. The variance of the projected length error is zero

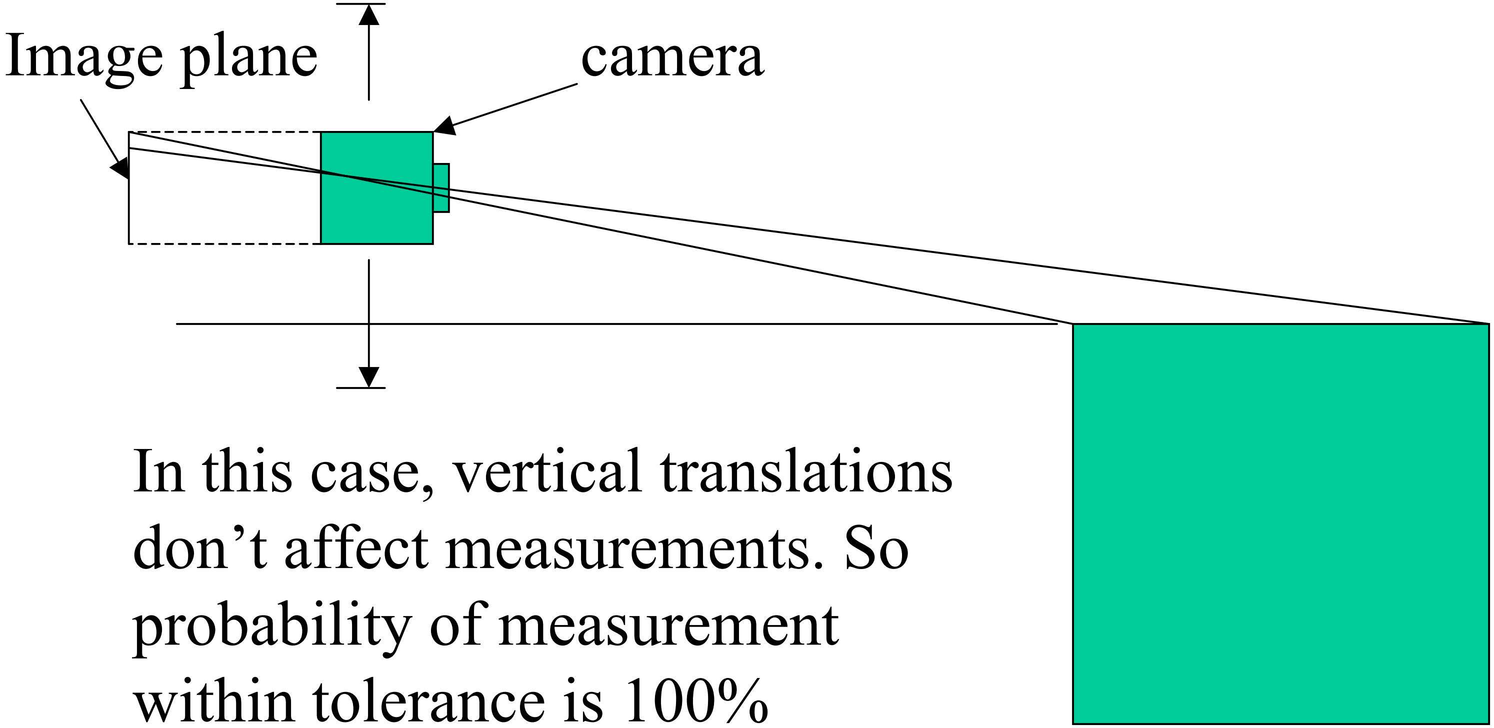
Euclidean Transformations

$$\begin{bmatrix} r11 & r12 & r13 & tx \\ r21 & r22 & r23 & ty \\ r31 & r32 & r33 & tz \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Preserve the length, area, and volume

(R. Hartley and A. Zisserman, Multiple View Geometry in computer vision)

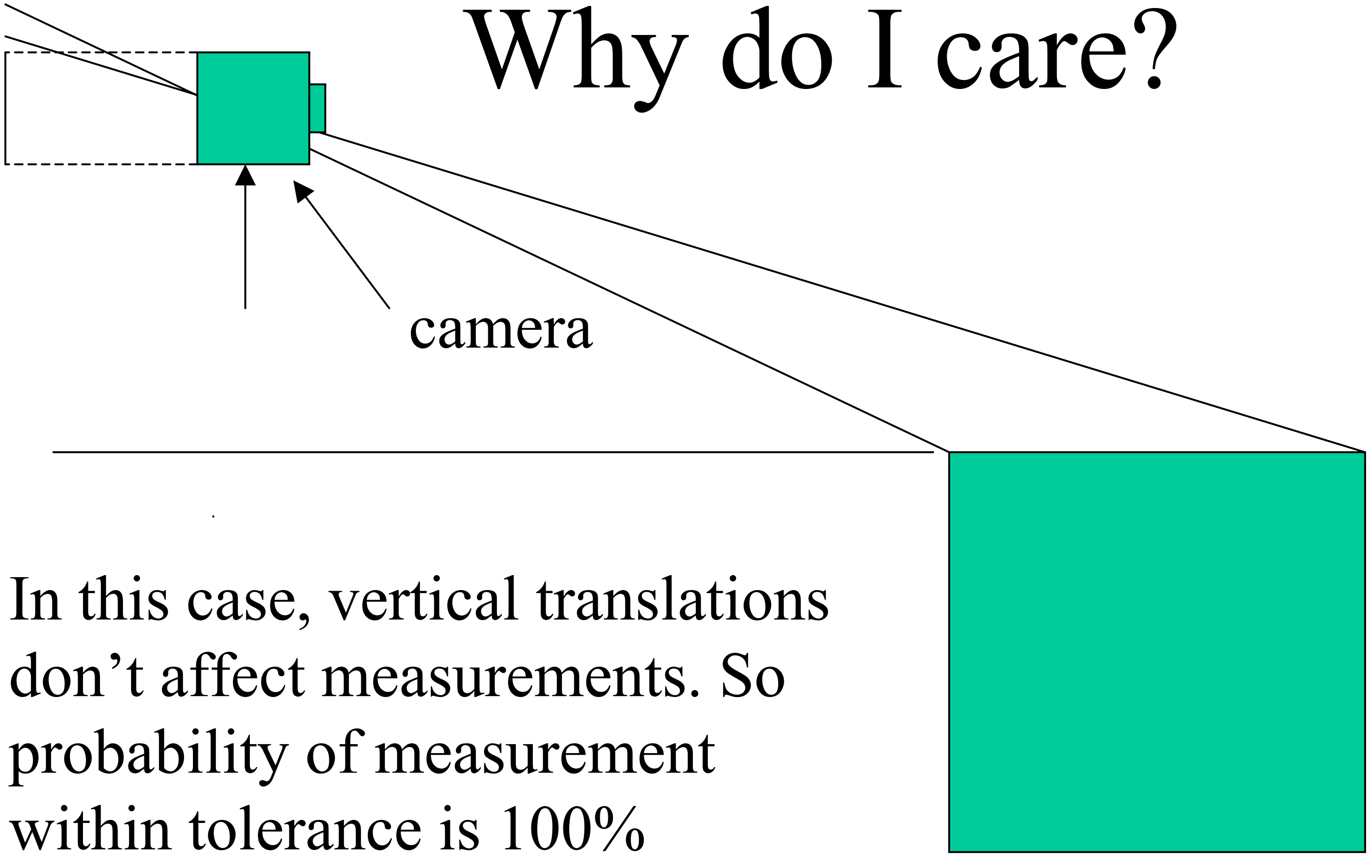
Why do I care?



In this case, vertical translations don't affect measurements. So probability of measurement within tolerance is 100%

But a variance in that pose may violate the visibility, resolution, or field of view constraint

Why do I care?



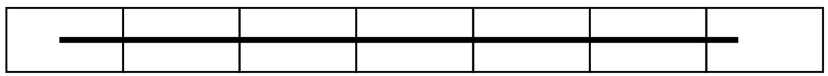
In this case, vertical translations don't affect measurements. So probability of measurement within tolerance is 100%

But a variance in that pose may violate the visibility, resolution, or field of view constraint

In summary

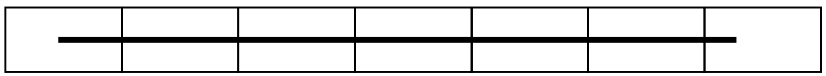
- The probability that the measurement is within the tolerance, is not a complete measure of robustness

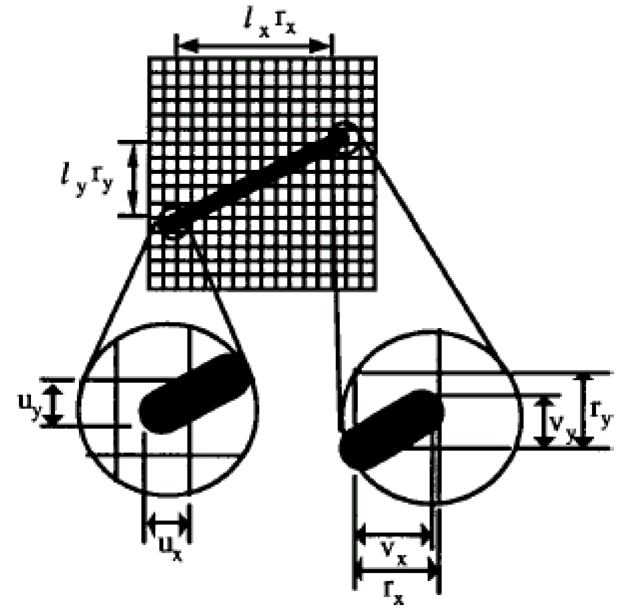
Quantization errors



- Represented by independent uniform variables with zero mean and variance equal to the pixel resolution
- Introduce errors in the projected length of the entity. The error has a triangular distribution with zero mean and some variance (Crosby)

Quantization Error 1D





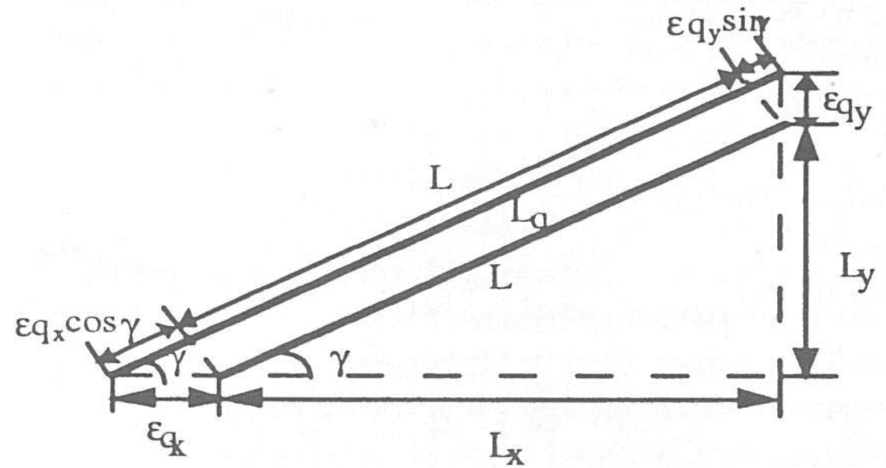
- Actual Length:

  $L = lr_x + u + v$, where u,v uniform random variables

- Quantized Length

$$Lq = \begin{cases} lr_x & u \leq .5 \cap v \leq .5 \\ (l+2)r_x & u > .5 \cap v > .5 \\ (l+1)r_x & (u \leq .5 \cap v > .5) \cup (u > .5 \cap v \leq .5) \end{cases}$$

Quantization Error for a line

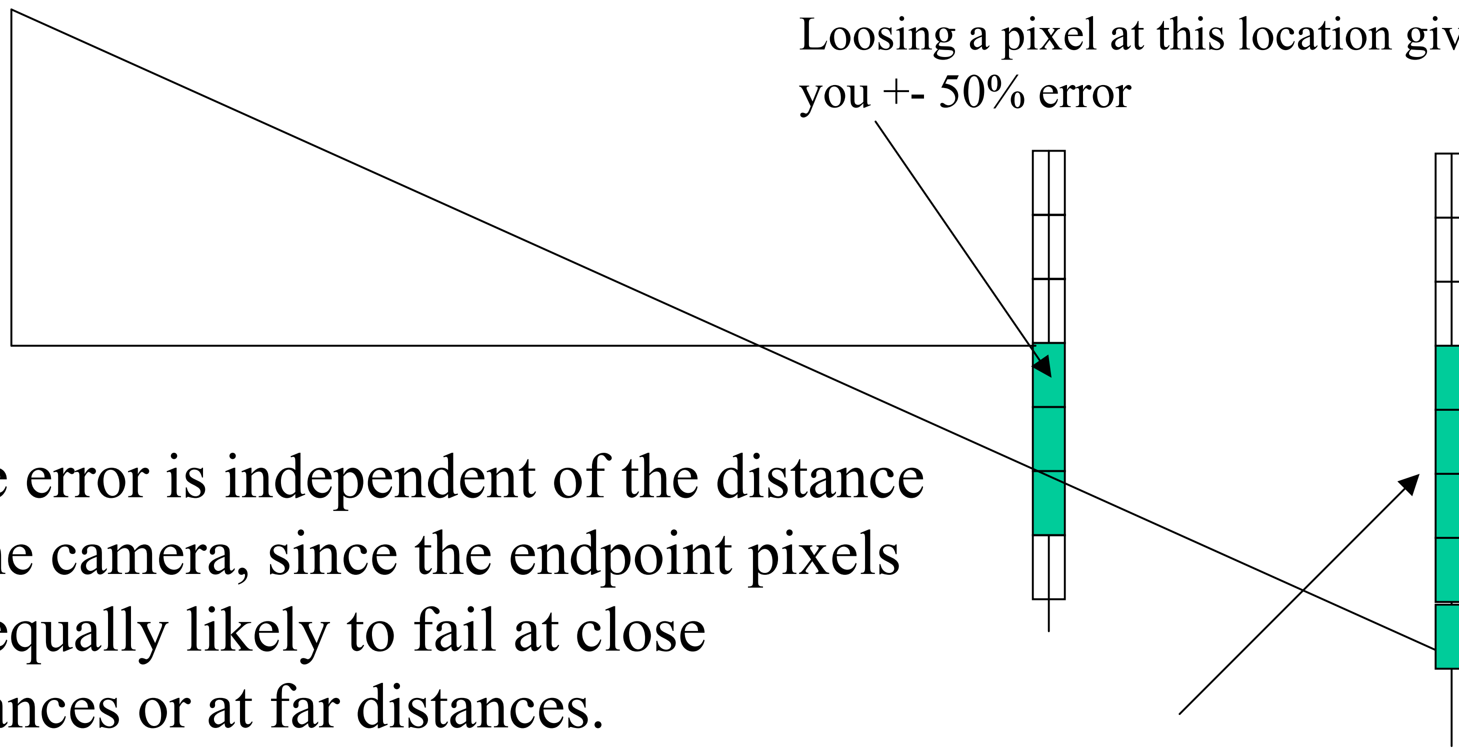


- Total quantization determined by geometric approximation,

$$\varepsilon_q \approx \varepsilon_{qx}\cos(\gamma) + \varepsilon_{qy}\sin(\gamma)$$

- zero mean
- $E[\varepsilon_q{}^2]=\sigma_{\varepsilon q}{}^2 \approx (1/6)(r_x{}^2\cos^2\gamma + r_y{}^2\sin^2\gamma)$

Issues with the quantization error model



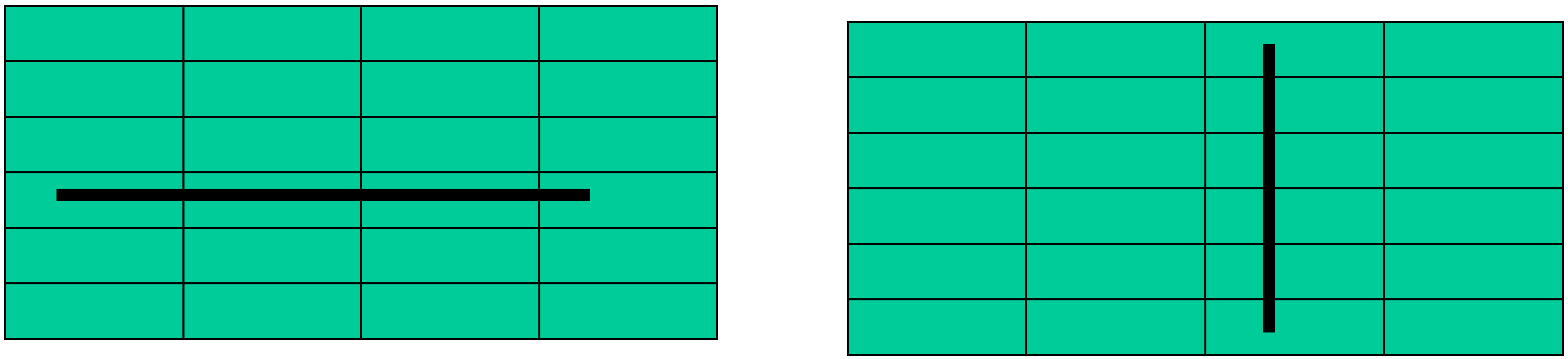
- The error is independent of the distance of the camera, since the endpoint pixels are equally likely to fail at close distances or at far distances.

Therefore, the variance of the line error will be the same. This is wrong.

Issues (cont)

- The measurement error variance changes with the rotation of the entities. This makes sense

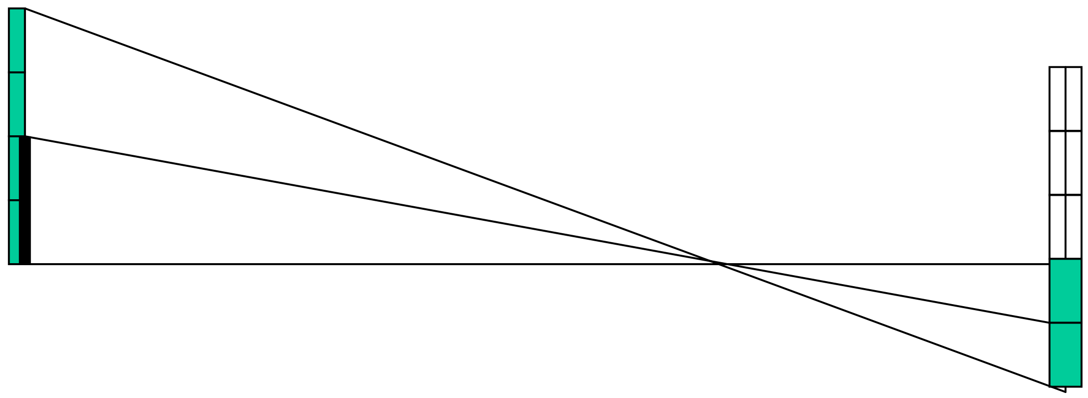
Error is bigger because pixel resolution is bigger along horizontal direction than along the vertical direction



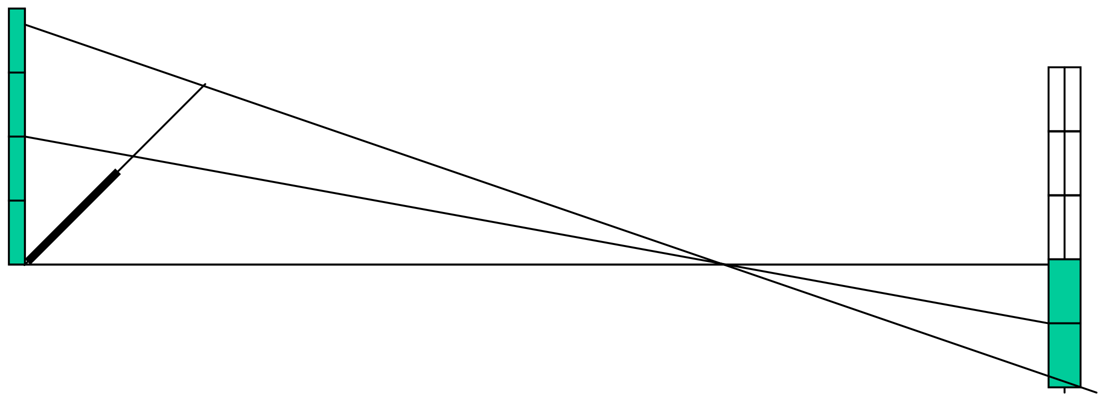
Issues (cont)

- The model assumes that each pixel covers the same amount of world units
- This is only true if the entities to be measured are perpendicular to the viewing direction
- Otherwise, the error in the image is different than the error in the world

Example



Loosing a pixel here is equivalent to a 50% error



Loosing a pixel here is more than 50% error in world, but still 50% in image

Summary intrinsic errors issues

- Some camera displacements may result in 100% probability of the measurement satisfying the dimensional tolerances, yet in practice violate constraints
  - The current model doesn't take into account the probability that the final pose will still satisfy the constraints, given the displacement error variances
- Measurements are done in image plane without taking into consideration how much of the world each pixel represent
  - On some cases, the probability that the measurement is within the specified tolerances (robustness) will be an overestimate

Summary intrinsic errors issues

- Measuring length of 3D lines using pixels only works when the entities are perpendicular to the viewing direction
  - it can be done using multiple views
  - it can be done using a single view by analyzing geometric invariants of the perspective projections.
  - This model doesn’t represent these scenarios

TO DO

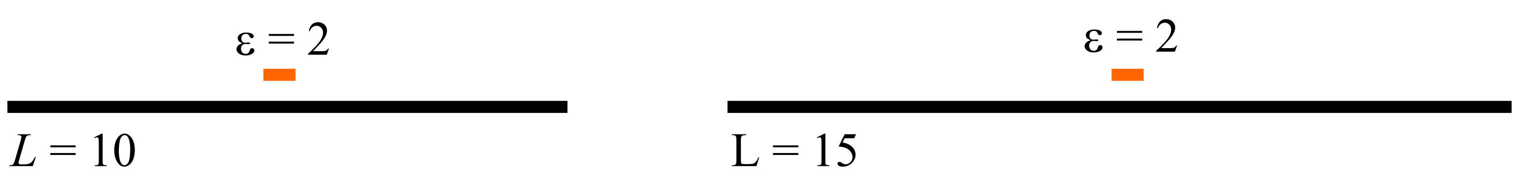
- Come up with experiments that illustrate these claims
- How frequently do these cases occur? How much different is this model from the experiments?
- Evaluate possible fixes
  - add constraints to restrict angle between lines and viewing direction so they are perpendicular to each other
  - incorporate pixel covering into the quantization error
  - incorporate probability that final pose will still satisfy the constraints

Total Percent Mse

Fai Lung Shih

Problem

- In the error model, the displacement and quantization are measured in absolute error
- Those errors are not related to the size of the object



Absolute error = 2	Absolute error = 2
Relative error = 20%	Relative error = 13.3%

New Modeling Errors

- Displacement Error

$\varepsilon_{dx} = (\varepsilon_{du1} - \varepsilon_{du2})$

$\varepsilon_{dy} = (\varepsilon_{dv1} - \varepsilon_{dv2})$

$\varepsilon_d \approx (\varepsilon_{dx}\cos(\gamma) + \varepsilon_{dy}\sin(\gamma))$

- Total percent dimensional error for k lines is:

$$\varepsilon_d = \sum_{j=1}^{k} \varepsilon_{d_j} \Big/ \sum_{j=1}^{k} L_j$$

$$E[\varepsilon_d^2] = (\sigma_{\varepsilon_d}^2 + \eta_{\varepsilon_d}^2) \Big/ \left(\sum_{j=1}^{k} L_j \right)^2$$

New Modeling Errors Cont.

- Quantization Error

$$\varepsilon_q = (L_q - L)$$
$$= (\sqrt{(L\cos\gamma + \varepsilon_{qx})^2 + (L\sin\gamma + \varepsilon_{qy})^2}) - L$$

$\varepsilon_{qx} = (L_{qx} - L_x)$

$\varepsilon_{qy} = (L_{qy} - L_y)$

$\varepsilon_q \approx \varepsilon_{qx}\cos(\gamma) + \varepsilon_{qy}\sin(\gamma)$

$$E[\varepsilon_q{}^2] = \sigma_{\varepsilon q}{}^2 \approx (1/6)(r_x{}^2\cos^2\gamma + r_y{}^2\sin^2\gamma)$$

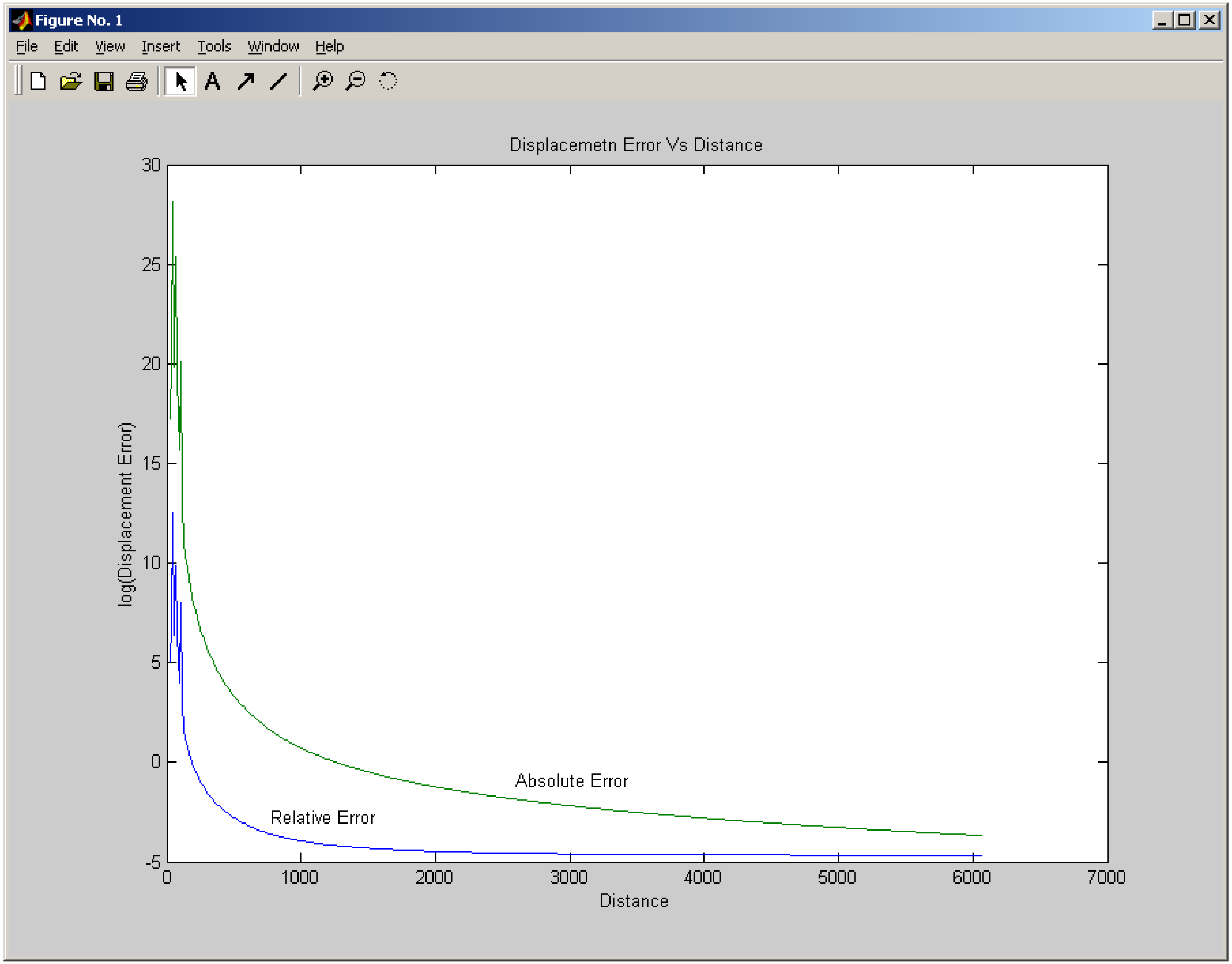
New Modeling Errors Cont.

- Total quantization error for k lines is:

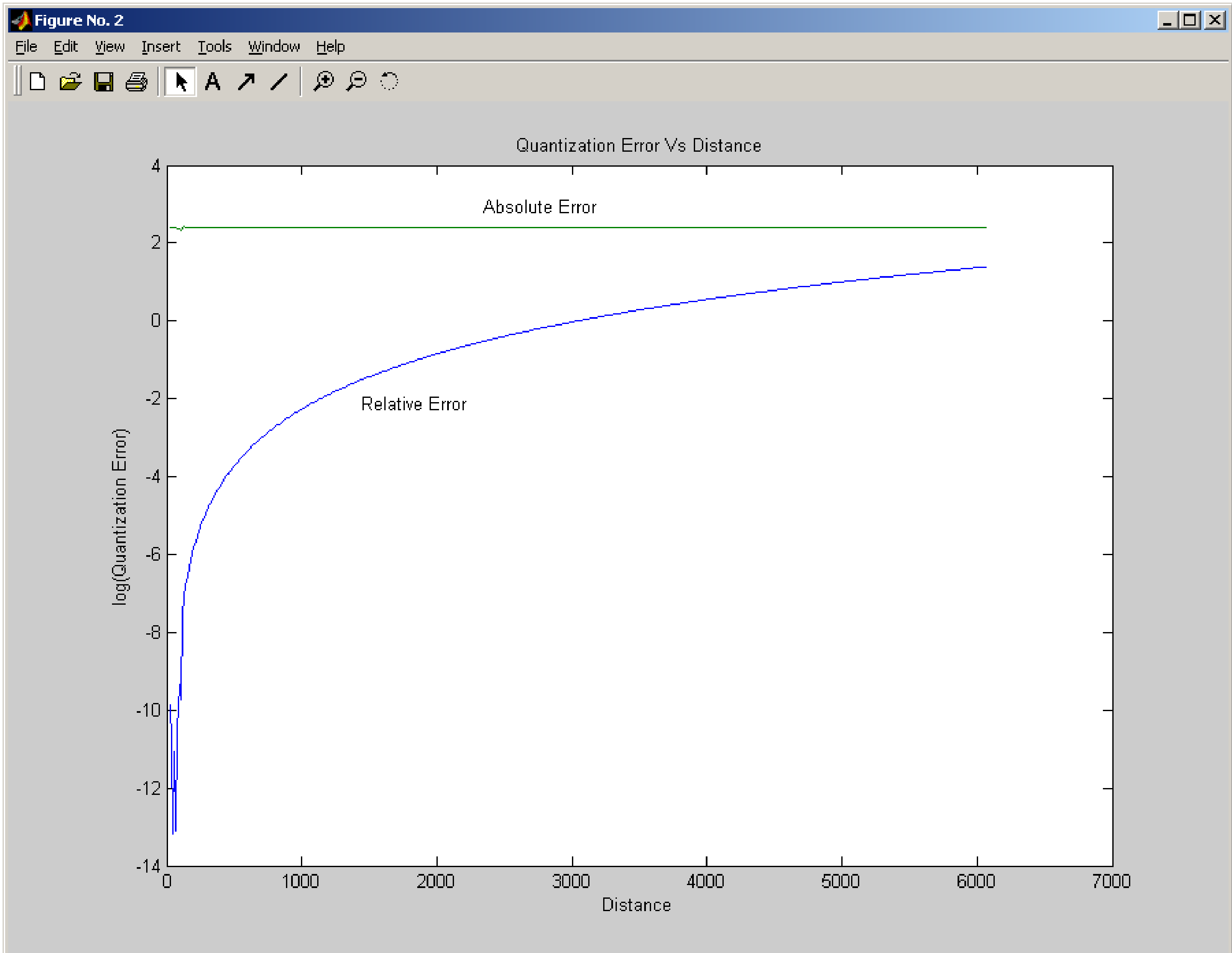
$$\varepsilon_q = \sum_{j=1}^{k} \varepsilon_{q_j} \Bigg/ \sum_{j=1}^{k} L_j$$

$$E[\varepsilon_q^2] = \sigma_{\varepsilon_q}^2 \Bigg/ \left(\sum_{j=1}^{k} L_j \right)^2$$

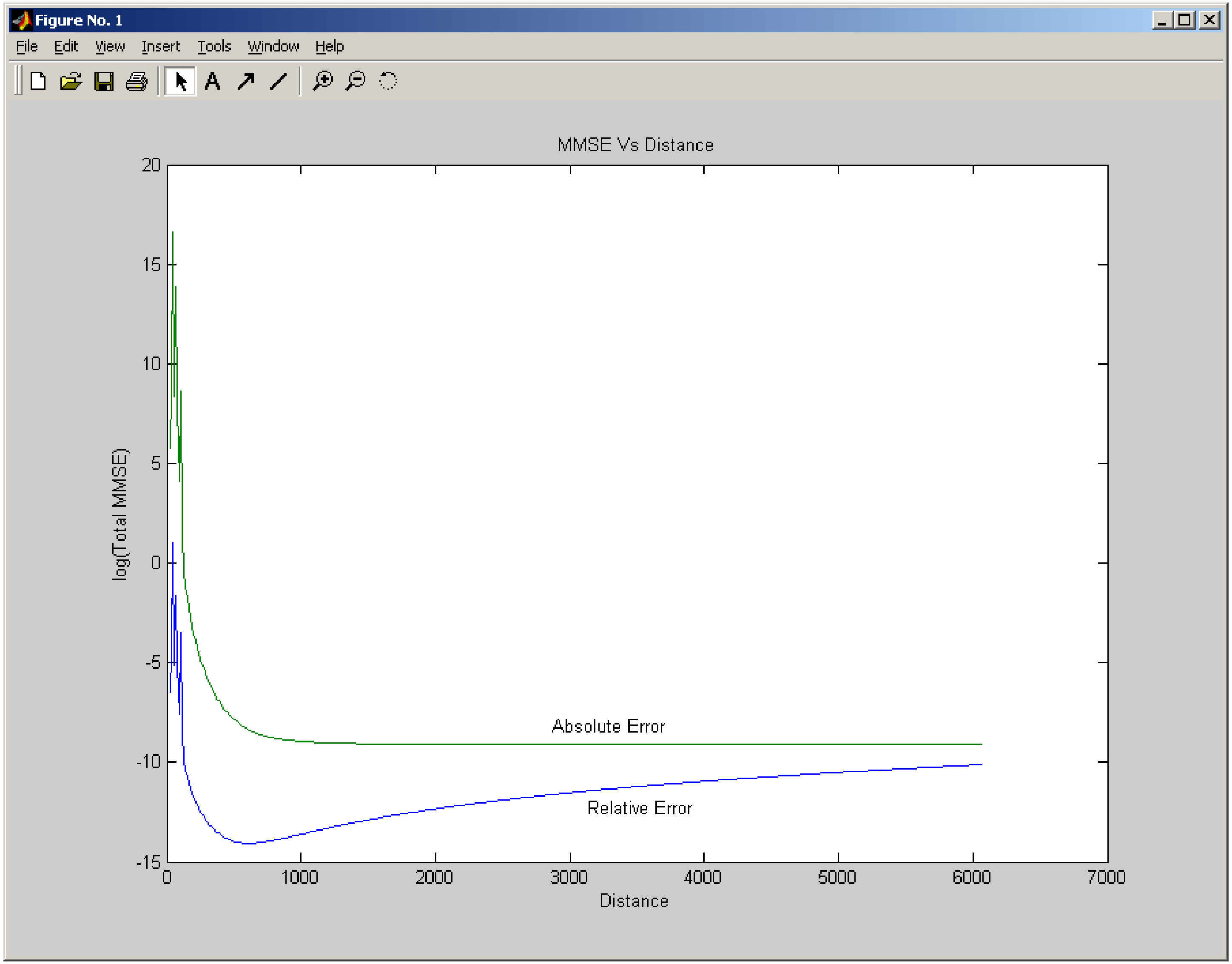
Comparison between the two models



Comparison between the two models



Comparison between the two models



Robustness Index

Alexis Rivera

Robustness Index

- Crosby's work gives the mean and variance and distribution of the displacement error.
- From that, we can calculate the statistics of the measured line
- Robustness Index

= P(tolerance) P (resolution) P(field of view) P(visibility)

Probability resolution is satisfied

- Let,
  - w = minimum required length
  - l’ = RV measured projected length
  - l = true projected length
  - ε_d = RV displacement error in projected line. Gaussian RV
  - $l' = l + \varepsilon_d$
- $E[l'] = l + E[\varepsilon_d]$
- $Var[l'] = Var[\varepsilon_d]$
- Probability resolution is satisfied:
- $P(l' > w) = 1 - CDF(w, E[l'], Var[l'])$
  - Where CDF, is the Gaussian Cumulative Density function

Probability FOV is satisfied

- Let
  - u_half = size of half of the horizontal image plane
  - v_half = size of half of the vertical image plane
  - u', v' = RV measured horizontal coordinate and vertical coordinates of an endpoint
  - ε_u = RV displacement error along horizontal axis
  - ε_v = RV displacement error along vertical axis
  - u, v = true horizontal true vertical coordinates of an endpoint
- $u' = u + \varepsilon_u$
- $E[u'] = u + E[\varepsilon_u]$
- $Var[u'] = Var[\varepsilon_u]$
- $v' = v + \varepsilon_v$
- $E[v'] = v + E[\varepsilon_v]$
- $Var[v'] = Var[\varepsilon_v]$
- Probability FOV is satisfied
- P(-u_half <= u' <= u_half) = CDF(u_half, E[u'], Var[u']) - CDF(-u_half, E[u'], Var[u'])
- P(-v_half <= v' <= v_half) = CDF(v_half, E[v'], Var[v']) - CDF(-v_half, E[v'], Var[v'])

Probability visibility is satisfied

- Let
  - dx, dy, dz = gaussian RVs of the position's displacement error
  - f(dx, dy, dz) = distance from the camera position to the visibility plane given by
  - ax + by + cz + d + adx + ady + cdz
- E[f(dx, dy, dz)] = ax + by + cz + d
- Var[f(dx, dy, dz)]= a^2Var(dx) + b^2Var(dy) + c^2Var(dz)
- Probability point inside visibility plane:
- P(f(dx,dy,dz) <= 0) = CDF (0, E[f(dx,dy,dz)], Var[f(dx,dy,dz)])
- Note: displacement errors are given in camera coordinates, so the plane equations have to be converted in terms of the camera coordinates.
- Let Q = transformation matrix from world to camera world
- Let H = matrix defines the visibility volume
- Let H' = plane equations in terms of the camera coordinate system
- H' = inv(Q)'*H

Dimensional Tolerances

- Let

  ε = RV dimensional error
- The probability that the measurement is within the tolerance
- P(-deltaL <= ε <= deltaL) =

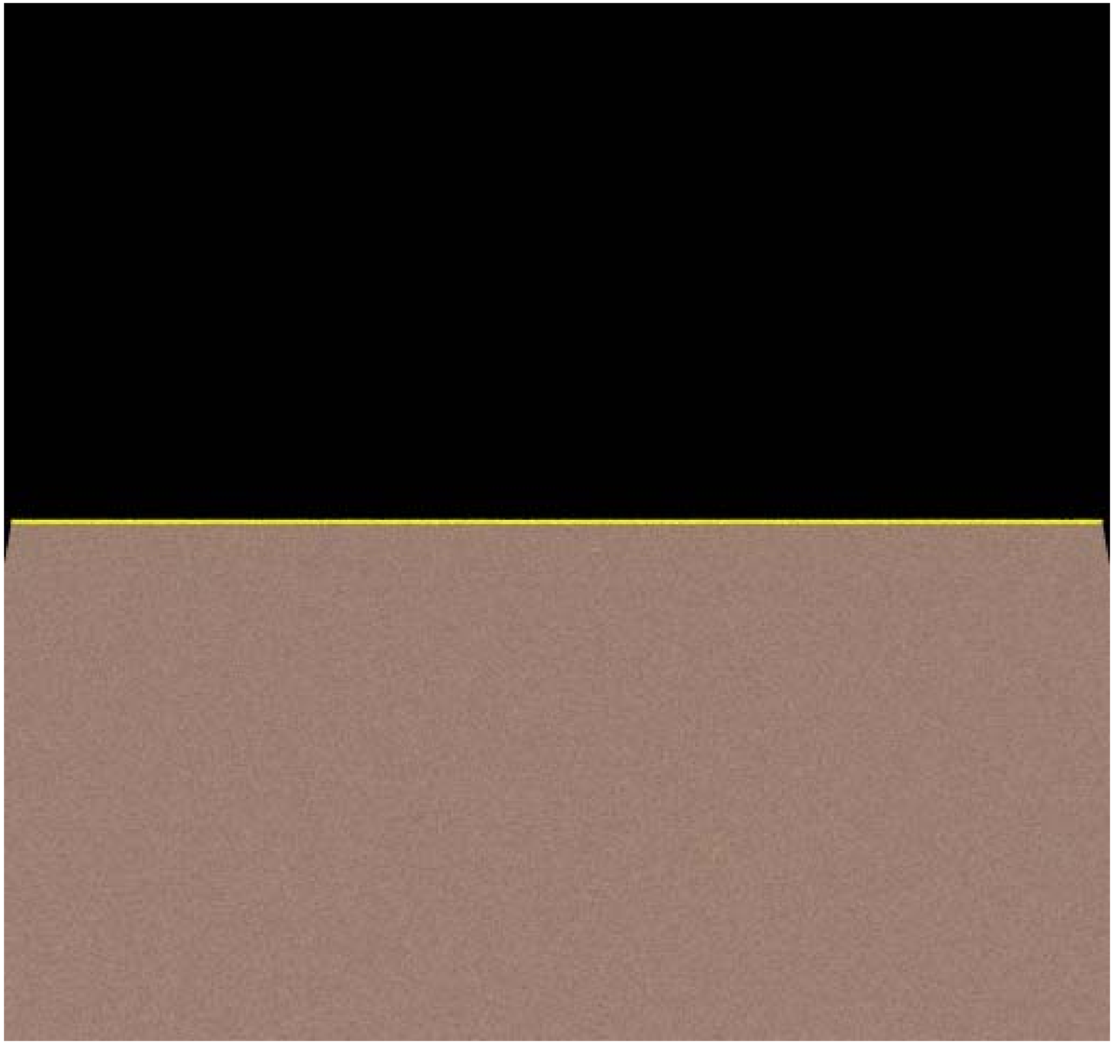
  CDF(deltaL,E[ε], Var[ε]) - CDF(-deltaL,E[ε], Var[ε])

Robustness Index

- Robustness Index

= P(tolerance) P (resolution) P(field of view) P(visibility)

Example



Example (cont)

Optimal Pose	2443.004882 -2060.480192 0.000045 0.663994 -1.570779 -1.571883				
Objective Function	Robustness	Total MSE	Total Percent MSE	Robustness Index	Robustness Index Minimax
Distribution	Displacement	Displacement, Quantization	Displacement, Quantization	Displacement	Displacement
Obj Val	1029342.605	0.000348	0.000014	-10.835789	-100 (tol) -49.208272(fov) -66.894771(fov) -49.208268(fov) -66.89477 (fov) -100(res) -100(vis) -100(vis) -100(vis) -100(vis)

Summary

- **Very brief summary of what is visual inspection planning**
- **Camera models**
- **Incidence Angle**
- **Feasible Pose Determination**
- **Issues with intrinsic errors model**
- **Total Percent Error**
- **Robustness Index**

Future Work

- Implement image processing algorithms
- Perform real experiments to verify hypothesis
- Implement illumination models