MEAM 620 Advanced Robotics
Vijay Kumar and Kostas Daniilidis
Introduction (1/14)
Quadrotor – experimental platform from KMel Robotics

References: **both online**
A Mathematical Introduction to Robotic Manipulation
Robotics, Vision and Control (comes with Matlab toolbox)

Motivational video on canvas

Approximate Cell Decomposition & Searching for the best plan (1/14)
Convert motion planning in continuous space into discrete graph search by identifying cells of free space that you can move through

Vertical Cell Decomposition: extend vertical lines from every vertex
Approximate Cell Decomposition: overlay grid; never exactly computing boundaries of free space
Have to figure out how to handle partially occupied grid cells / grid cell size
Planning by Searching a Tree – breadth first search, depth first search, best first search (Djikstra / A*)

More edges between nodes = finer resolution of paths
More nodes (vertices) = finer resolution of paths as well
Convert partially observable cells into untraversable cells = conservative and incomplete (may not find a path that actually exists)

Questions to ask yourself: Resolution? Adaptive? Which Search Algorithm?
Search Algorithm: cost to go from intermediate point to destination is key estimate h(s)
Heuristic: remaining number of edges to goal from a given node
Better solution typically involves finding a better heuristic for your problem

Rigid Body Transformations (1/21)
Determine position / orientations from reference frames
Rigid body displacement:

- Lengths are preserved (distance invariance)
- Cross products are preserved (angles are preserved)

R is orthogonal, det(R) = 1 (special orthogonal)
Group
Beyond rotation matrices

- The group of rotations, SO(3)
- SO(3) is a Lie group, properties of a differentiable manifold
- Coordinates for SO(3)
    - Rotation matrices – only 3 of the 9 numbers are independent

- Euler angles / yaw, pitch, roll
- Axis / angle
- Exponential coordinates
- quaternions

The group of rotations

$$SO(3) = \{R \epsilon R^{3x3} | R^T R = RR^T = I, detR = 1\}$$

SO(3) satisfies the 4 axioms that must be satisfied by the elements of an algebraic group:

- closure under the binary operation
- associativity
- SO(3) includes the identity element
- SO(3) includes the inverse of every element

SO(3) is a continuous group.

The binary operation above is a continuous operation

The inverse of any element is a continuous function of that element

The norm of RR'^T should be close to the norm of the identity matrix SO RT and R'T should be very similar
$(RT)(R'T)^T = RTT^TR'^T = RR'^T$

SO(3) is a smooth manifold

Manifold: a manifold of dimension n is a set M which is locally homeomorphic to R^n
Homeomorphism: a map f, from M to N, and its inverse, f^{-1}, are both continuous
SO(3) is locally homeomorphic to R^3

Euler Angles: any rotation can be described by 3 successive rotations about linearly independent axes

Hard to go from rotation matrix to Euler angles

Control System Design (1/26)
LTI Systems – can write as linear dynamic system
Controller Design – Gain

Control of a simple first-order system
Have control over velocity

General Approach

Define error: $e(t) = x^{des}(t) - x(t)$

Want e(t) to converge exponentially to 0

Find u such that: $\dot{e} + K_p e = 0, K_p > 0$

$u(t) \ = \dot{x}^{des}(t) + K_p e(t)$, proportional feed forward

Control for a second-order system

$$\ddot{e} + K_v\,\dot{e} + K_p e = 0, K_p, K_v > 0$$

$u(t) = \ddot{x}^{des}(t) + K_v\dot{e}(t) + K_p e(t)$, PD controller (proportional derivative feed forward)

Similar to mass, spring, damper system

Large derivative gain makes the system overdamped and the system converges slowly

PID control – add in integral term. Advantageous in the presence of disturbance or modeling errors

$$u(t) = \ddot{x}^{des}(t) + K_v\dot{e}(t) + K_p e(t) + \mathrm{K_I}\int_0^t e(\tau)d\tau$$

PID control generates a third-order closed loop system

Integral control makes the steady-state error go to 0

Best to start with PD and only add integral if necessary (3^{rd} order systems are difficult to analyze / design)

Disadvantages of PID / PD control schemes

- Performance will depend on the model
- Need to tune gains to maximize performance

Model based control law: $f(t) = m(\ddot{x}_{d(t)} + k_p e(t) + k_v\dot{e}(t)) + b\,\dot{x}(\mathrm{t}) + \mathrm{kx(t)}$ = model (feed-forward + PD feedback) + model based

Separates model independent and dependent components of control law

But does not drive error exponentially to 0, actually drives the error away from 0 (though it bounds the error)

Gain Tuning – simple systems can find with trial / error, but usually require some kind of systematic way of doing so

Manual Tuning: marginally stable (error isn't 0 but isn't growing to infinity)

Rise Time: time to go from 10% to 90% of max value

Overshoot: max value exceeding target value

Settling Time: time for error to be 2% of max overshoot

Steady-State: signal does not change over time

Can use MATLAB PID tool to train controller using a single step

K_d = damping

K_I = remove SS error but increase overshoot / settling time

Ziegler-Nichols Method: Heuristic for tuning gains; gives aggressive gains (will have to tweak more)

Application to Quadrotors

$\phi = attitude\ angle$

Non-linear system (cos / sine terms) but also an affine system x and u are related $x = \mathrm{f(x)} + \mathrm{g(x)u}$

State space notation is used

Linearized Dynamic Model
Select equilibrium configurations to solve the dynamic model (first order approximation to simplify nonlinear model into a linear one)

Want our command angles / angular velocities to NOT deviate much from our equilibrium point

Use step inputs where you zero 2/3 variables and change / observe / tune the set of gains of a particular variable

Phase 2 = use full 3D controls system and tune / test in simulator
IDEAL GAIN PARAMETERS PRODUCE: Fast rise time, small over-shoot, small settling time
High K_p = lots of oscillations (ringing)
Low K_p = slow to respond
High K_d = slow to respond

Trajectory Tracking:
Avoid breaking our hover assumptions
If too aggressive (bad time parameterization – too fast, bad model parameters, saturated inputs), want to follow the path but not as fast

Quadrotor Dynamics (1/28)

Each motor rotating at different speeds

2 rotors spin in 1 direction; 2 in the opposite direction (different pitches of blades)

Avoid singular configurations when hovering

Newton's Second Law for a System of Particles
Center of mass: $r_c = position = \frac{1}{m}\sum_{i=1,N} m_i \boldsymbol{p}_i$ (weight position with mass)
The center of mass for a system of particles, S, accelerates in an inertial frame (A) as if it were a single particle with mass m (equal to the total mass of the system) acted upon by a force equal to the net external force)
$F = \sum_i^N F_i = m^A \frac{d^A v^C}{dt}$ (must be in an inertial frame to do this calculation)
Euler equations done in a body-fixed frame
This law holds true for rigid body too

Rotational equations of motion for a rigid body (A is a inertial frame)
C = center of mass AND origin
Angular counterpart to Newton's Second Law

Principal Axes and Principal Moments
Where does the angular momentum vector point?

Principal axes = eigenvectors of inertial tensor (eigenvalues = principal moments)

Euler's Equations – written in the moving frame
Coordinate system is attached to body AND along principal axes
Write angular velocity vector in moving coordinate system
$\frac{d\boldsymbol{x}^A}{dt} = \frac{d\boldsymbol{x}^B}{dt} + \omega^{AB} \times \boldsymbol{x}$
Moving frame simplifies some calculations in exchange for needing to compensate for putting it back into the inertial frame

R = rotation matrix to push applied forces in correct inertial frame
I = diagonal matrix

Rotations, Angular Velocity (2/2)
3D manifold looks like a volume; 2D manifold looks like a plane
Manifold – a manifold of dimension n is a set M which is locally homeomorphic to R^n
Homeomorphism: a map f from M to N and its inverse, f^{-1}, are both continuous.

Diffeomorphism: (continuous AND smooth)
A map f from U belonging to R^m to V belonging to R^n is smooth if all partial derivatives of f, of all orders, exist and are continuous.
A smooth map f from U in R^n to V in R^n is a diffeomorphism if all partial derivatives of f^{-1}, of all orders exist and are continuous

Homeomorphism focuses on continuity between manifolds, Diffeomorphism wants to map calculus operations (derivatives) between manifolds

Example: inverse starts to fall apart around origin for y_2
So it's not a diffeomorphism UNLESS you exclude the origin (a is a diffeomorphism and b is NOT)

Smooth Manifold
Differentiable manifold is locally homeomorphic to R^n
DEPENDING on the area on the manifold, use a different parameterization (mapping) to R^n
Parameterize the manifold using a set of local coordinate charts - $(U, \phi), (V, \psi)$, …
Requires compatibility on overlaps $C^\infty related$- $\phi(x)$and $\psi(x)$ARE NOT necessarily the same spot in R^n
MUST have some relationship between the two though
Collection of charts covering M with differentiable transition functions

Sphere in 3 dimensions (S_2)
- Differentiable manifold is locally homeomorphic to R^2
- Parametrize using a set of local coordinate charts (latitude and longitude)
- Collection of charts covering the surface of the earth with differentiable transition functions

- Minimum is 2
  - Cover northern hemisphere using xy system along equator
  - Create a similar mapping for southern hemisphere (note: must still handle the overlap)

What is the minimum # of charts you need to cover SO(3)?

$$SO(3) = \{R \in R^{3x3} | R^T R = RR^T = I, detR = 1\}$$

Euler Angles (another set of charts to cover SO(3)):

Axis / Angle Representation:

Euler's Theorem

Rotations: any displacement of a rigid body such that a point on the rigid body, say O, remains fixed, is equivalent to a rotation about a fixed axis through the point O.

Chasles' Theorem for General Displacements: the most general rigid body displacement can be produced by a translation along a line followed (or preceded) by a rotation about that line.

Proof of Euler's Theorem

q = Rp

is there a point p that maps onto itself?

p = Rp

Eigenvalue problem: $Rp = \lambda p$

$\lambda = 1$ turns out to be a eigenvalue of ANY rotation matrix R

τ is at most 3 (since each column / row is normalized) and at least 1

$$\lambda_1 = 1 \; \lambda_{2,3} = \cos(\phi) \pm \mathrm{i}\sin(\phi) = e^{\pm i\phi}$$

Real eigenvector (u) and 2 complex conjugate eigenvectors (can convert into 2 real vectors spanning space orthogonal to u: v and w)

Define our axis/angle as axis of rotation u and rotation angle ϕ

Are the axis and angle always uniquely defined for a rotation? – at R = I, no axis. U and ϕ can point in the positive / negative direction.

Linear Systems

$$\frac{d\boldsymbol{x}}{dt} = A_{nxn}\boldsymbol{x}$$
$$\boldsymbol{x}(t_0) = \boldsymbol{x}_0$$

Exponential of a matrix, A: $exp\boldsymbol{A} = e^{\boldsymbol{A}} = I + A + \frac{1}{2}A^2 + \frac{1}{3!}A^3 + \cdots + \frac{1}{n!}A^n + \cdots$

$A^2 = AA$, $A^3 = A^2A$, $A^k = A^{k-1}A$

Solution: $x(t) = \exp\big((t - t_0)\boldsymbol{A}\big)\,\boldsymbol{x}(t_0)$

Opposite points must be identified for SO(3) representation

$R^T\dot{R}\ or\ \dot{R}R^T$: $\frac{d}{dt}(R^T R = I) \rightarrow\ \dot{R^T}R + R^T\dot{R} = 0 \rightarrow\ \widehat{w}^T + \widehat{w} = 0 \rightarrow \widehat{w} =\ -\widehat{w}^T$ yields a skew symmetric matrix

2/4
Calculus of Variations
x* = optimal trajectory
functional = function of functions
Fermat's principle: light passing through different materials
Compute first / second derivative and set to 0

Euler Lagrange Equation: necessary condition satisfied by the "optimal" function

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}}\right) - \frac{\partial L}{\partial x} = 0$$

Robot Perception 2/11/15
Manual

- Shutter speed
- Aperture
- White balance
- Sensitivity
- Depth of field (aperture / focal length)
- Zoom (f)
- Focus (b)

Moving the image plane is what we call (de-)focusing $\frac{1}{f} \neq \frac{1}{a} + \frac{1}{b}$

Point at distance a is in focus
$\frac{1}{f} = \frac{1}{a} + \frac{1}{b}$
Sharpen image by focusing OR by making aperture smaller

Perspective projection: pinhole camera model
$\frac{y}{a} = \frac{y'}{b}, where\ f \approx b, y = f\frac{y}{z}$
Depth from de-focus: $f = \frac{ab}{a+b} = \frac{b}{1+b/a}, \lim_{a\to\infty} f = b$
Calibration = f [pixels]
Pixels ← y = f (mm) Y / Z
Y [pixels] = f/d (Y/Z)
d is μm
place image plane in front of lens in order to avoid inversion

the optical axis is the z-axis
the image plane (u, v) is perpendicular to the optical axis
intersection of the image plane with the optical axis is the image center (u_0. v_0)

f is the distance of the image plane from the origin (in pixels)

$$u = fX_c/Z_c + u_0; \;\; v = fY_c/Z_c + v_0$$

Camera Model 2/16/15

$X_c = (r_1, r_2 r_3)X_W + t$

$R = \begin{bmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix}$ in example

R^2 euclidean plane (and all points at infinity) = P^2 projective plane

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} \in P^2 \rightarrow \begin{pmatrix} \frac{u}{w} \\ \frac{v}{w} \end{pmatrix} \in R^2 (if\ w \neq 0)$$

Line through 2 points: $l \sim p \times q$

Intersection of 2 lines: $p \sim l \times m$

2/18 Nonlinear Control

- Underactuated
- Saturation
- Large angles

$$\frac{\partial h}{\partial x} = \nabla \mathrm{h} = \left[\frac{\partial h}{\partial x_1}, \frac{\partial h}{\partial x_2}, \frac{\partial h}{\partial x_3}\right]$$

$$L_f h = (\nabla h)) \cdot f$$

$$L_g h = (\nabla h) \cdot g$$

Hanging pendulum with length l and mass m with input torque u

$$ml^2\ddot{\theta} + mgsin\theta + c\dot{\theta} = u$$

$$x = \binom{\theta}{\dot{\theta}}$$

$$\dot{x} = \begin{bmatrix} x_2 \\ \frac{-cx_2 - mgsinx_1}{ml^2} \end{bmatrix} + \begin{bmatrix} \theta \\ \frac{1}{ml^2} \end{bmatrix} u = f(x) + g(x)u$$

$$f(x) = hx, g(x) = \ hx$$

$\frac{\partial h}{\partial x} = (1, 0), L_g h = 0$

Torque has no direct effect on velocity?

Original (affine) system with input u and output y

Input-output linearization: use nonlinear feedback that's based on the Lie derivatives to produce a linear system (instead of using linear approximations)

$$\frac{d}{dt}(\dot{y}) = \frac{d}{dt}(L_f h)$$

$$\frac{\partial L_f h}{\partial x} = f + gu = L_f L_f h + L_g L_f u$$

Want to control some derivative of h: this derivative corresponds to the relative degree (r) – the index of the first nonzero term in the sequence – controllable by differentiating at that level

Multiple Input / Multiple Output Systems
n = m? (# inputs = # outputs)
assume each output has same relative degree

$$\left(L_g L_f h\right) = 0\,?$$

$$\left(L_g h\right)_{mxm}$$

$$x = f(x) + [g_1 g_2 \dots g_m]\begin{pmatrix} u_1 \\ \vdots \\ u_m \end{pmatrix}$$

$$(\partial h/\partial x) = \begin{pmatrix} \partial h_1/\partial x_1 & \dots & \partial h_1/\partial x_n \\ \vdots & \ddots & \vdots \\ \partial h_m/\partial x_1 & \dots & \partial h_m/\partial x_n \end{pmatrix}$$

M inputs for u → $\alpha(x) + \beta(x)u$ → m artificial inputs v → $v = \dot{y}$ → y = h(x) m outputs

Visual Servoing:
2 types of approaches

- Image-based: move a camera in such a way that the image becomes my desired image
- Position-based: estimate pose based on features and try to achieve desired pose

Image-based Visual Servoing for 3 features

$$f(x) = 0$$

Relative degree is not 1 IF using point O as point of interest instead of point P (points on the axis lose the ability to control θ)

Quadrotor:

- Under-actuated
- Try to control center of mass

Feature Detectors
Edge is a point which is a maximum of $||\nabla y||$ along the gradient direction
Corner $E(\Delta x) = \sum_{x\in N}(I(x+\Delta x) - I(x)) \rightarrow pick\ \max_{\Delta x}$

$$0 \le \lambda_{min}(A) \le \frac{\Delta x^T A \Delta x}{\Delta x^T \Delta x} \le \lambda_{max}(A)$$

$$I(x,y) + \frac{\partial I}{\partial x}\delta x + \frac{\partial I}{\partial y}\delta y = I(x) + \nabla I^T \Delta x, \Delta x = \begin{pmatrix} \delta x \\ \delta y \end{pmatrix}$$

Good point features are features with $\lambda_{min}, \lambda_{max}$ very large

$$\lambda_{min} \gg 0 \rightarrow \lambda_{max} \gg 0$$

$$\lambda_{min} = 0 \rightarrow \begin{matrix} \lambda_{max} = 0, homogenous\ region \\ \lambda_{max} \gg 0, edge \end{matrix}$$

$$\lambda_{max}\ is\ large\ because\ it\ representes\ the\ \max change$$

2x2 matrix: $tr(A) = \lambda_{min} + \lambda_{max}$, $\det(A) = \lambda_{min}\lambda_{max}$

Cornerness function: R = det(A)-k tr(A)^2 , k = .06 (equation / k found from experiments)

Cornerness function of trace / determinant R(x, y)

Is it rotation invariant? $R(x, y) = R(xcos\theta + ysin\theta, \rightarrow xsin\theta + ycos\theta)$

$$A = \sum \nabla I \nabla I^T = \sum \begin{pmatrix} I_x^2 & I_x I_y \\ Ix\, I_y & I_y^2 \end{pmatrix}$$

$\nabla I_\theta = Q \nabla I$ (rotating an image will rotate the gradient, $Q = \begin{matrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{matrix}$

$$A_\theta = \sum Q \nabla I \nabla I^T Q^T = Q \sum \nabla I \nabla I^T Q^T = QAQ^T$$

Trace and determinant are invariant to rotation (eigenvalues are invariant to rotation)

Is cornerness function scale invariant? No (not same window)!

$$f_s(x) = f(sx), f_s'(x) = f'(sx)s$$

3/2 Homework Review / Review

Things that stop optical flow

- Occlusion
- No texture
- Illumination difference
- Aperture problem
- Too large motion

Find closest window correspondence as local search

Assumption: $I_1(x - u, y - v) = \alpha I_2(x, y)$ (alpha = gain – sudden change in intensity), currently treat alpha = 1 (Brightness change constraint equation)

Neighborhood: best fit (u,v) that satisfies:

$$\int\int \left(I_1(x - u, y - v) - I_2(x, y)\right)^2 dx\, dy\ over\ (x, y) \in Neighborhood$$

(u,v) is locally constant

Things which are closer, move much faster (would have higher optical flow)

$I_1(x - u, y - v) = I_1(x, y) + \nabla I_1^T \begin{pmatrix} u \\ v \end{pmatrix}$, small motion approximation ONLY

Overconstrained system for a neighborhood of M pixels

$$I_1(x, y) + \nabla I_1^T \begin{pmatrix} u \\ v \end{pmatrix} = I_2(x, y)$$

$$\frac{\partial I}{\partial x} u + \frac{\partial I}{\partial y} v = -(I_1 - I_2) = -\Delta I = -\frac{\partial I}{\partial t}$$

$$\begin{pmatrix} I_{x,1} & I_{y,1} \\ \vdots & \vdots \\ I_{x,M} & I_{y,M} \end{pmatrix}_{Mx2} \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} -I_{t,1} \\ \vdots \\ -I_{t,M} \end{pmatrix}, A \begin{pmatrix} u \\ v \end{pmatrix} = b$$

Uniqueness: Rank(A) < 2 (infinite solutions); Rank(A) = 2 (1 solution)

Existence: Rank(A) ≤ 2
No texture: Rank(A) = 0
Edge: Rank(A) = 1

- Large motions
  - Can be reduced by shrinking the image (sub-sampling)

$$I(x,t) = \sin w(x-ut)$$

$$I(2x,t) = sinw(2x-ut) = sin2w(x-\frac{u}{2}t)$$

- Different optical flow model omit bigger windows

$I_x(a_{11}x + a_{12}y + a_{10}) + I_y(a_{21}x + a_{22}y + a_{20}) = -I_t, a_{11}, a_{12}, \ldots$ = affine flow

Robot Perception: Projective Geometry
2D-3D pose

$$(x,y) \in R^2 \to \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \in P^2$$

Points at infinity $\begin{pmatrix} x \\ y \\ 0 \end{pmatrix} \in P^2$

$$Ax + By + Cw = 0 \to \begin{pmatrix} A \\ B \\ C \end{pmatrix} \in P^2$$

$p \sim (proportional)\ l \times m$ (intersection point is cross product of 2 lines)
$l \sim p \times q$ (line is cross product of 2 points on line)

Projective transformation of lines:
If A maps a point to Ap, then where does a line l map to?
Line equation in original plane: $l^T p = 0$
Line equation in image plane: $p' \sim Ap$; $l^T A^{-1} p' = 0$
Implies that $l' = A^{-T} l$

A has 8 unknowns

$$\mu\lambda p' = \mu A p$$

$$\lambda \begin{pmatrix} p_1' \\ p_2' \\ p_3' \end{pmatrix} = A \begin{pmatrix} p_1 \\ p_2 \\ p_3 \end{pmatrix}, 3\ equations$$

$$\frac{p_1'}{p_3'} = \frac{\mu A_{11}p_1 + A_{12}p_2 + A_{13}p_3}{A_{31}p_1 + A_{32}p_3 + A_{33}p_3}$$

$$a \sim A \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}\ \alpha a = A \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

$$b \sim A \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}\ \beta b = A \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$$

$$c \sim A\begin{pmatrix}0\\0\\1\end{pmatrix}\ \gamma c = A\begin{pmatrix}0\\0\\1\end{pmatrix}$$

$$d \sim A\begin{pmatrix}1\\1\\1\end{pmatrix}$$

$$\delta d = \alpha a + \beta b + \gamma c$$

$$d = \frac{\alpha}{\delta}a + \frac{\beta}{\delta}b + \frac{\gamma}{\delta}c = (a \quad b \quad c)\begin{pmatrix}\frac{\alpha}{\delta}\\ \frac{\beta}{\delta}\\ \frac{\gamma}{\delta}\end{pmatrix}$$

$$(a \quad b \quad c \quad d) \sim A\begin{pmatrix}1 & 0 & 0 & 1\\0 & 1 & 0 & 1\\0 & 0 & 1 & 1\end{pmatrix}, A\ is\ unique$$

Or use the trick that no triple of points can be collinear: (missed this bit)

$$(a' \quad b' \quad c' \quad d') \sim A'\begin{pmatrix}1 & 0 & 0 & 1\\0 & 1 & 0 & 1\\0 & 0 & 1 & 1\end{pmatrix} \sim A'A^{-1}(a \quad b \quad c \quad d)$$

No radial distortion

$$\begin{pmatrix}u\\v\\w\end{pmatrix}_{pixels} \sim K_{3\times 3}(r_1\ r_2\ r_3\ T)\begin{pmatrix}X\\Y\\Z=0\\W\end{pmatrix}$$

$$\begin{pmatrix}u\\v\\w\end{pmatrix}_{pixels} \sim K_{3\times 3}(r_1\ r_2\ \ T)\begin{pmatrix}X\\Y\\W\end{pmatrix}$$

Is the matrix non-singular?

$$\det(K) = f^2$$

$$\det(r_1\ r_2\ T) \neq 0\ (only\ singular\ if\ r_1, r_2, T\ are\ on\ the\ same\ plane)$$

$$K^{-1}H \sim (r_1\ \ r_2\ \ T)$$

First 2 columns must be orthogonal, unit vectors: $||r_1|| = ||r_2||, r_1^T r_2 = 0$

H': for any $h_1', h_2' \in R^3$ find the closest r_1 and r_2 such that $||r_1|| = ||r_2||, r_1^T r_2 = 0$

$$\min_{R\in SO(3)} ||R - (h_1' \quad h_2' \quad h_3')||_F^2$$

3/18

Given: P in world, p in image, image calibration ($p_i = K^{-1}\begin{pmatrix}u\\v\\1\end{pmatrix}_{pixels}$)

Find: R, T – where is the camera in the world?

$$\lambda_i p_i = RP_i + T$$

PnP problem: perspective n points problem

Lighthouse problem:
Can see 2 lighthouses and know the angle formed by you / the two lighthouses → still don't know position / orientation: Locus of 2 points + known angle is a circle

$$\lambda_i \begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix} = R \begin{pmatrix} X_i \\ Y_i \\ Z_i \end{pmatrix} + T, R\ is\ unknown\ (3\ unknowns), \begin{pmatrix} X_1 \\ Y_1 \\ Z_1 \end{pmatrix}\ is\ non-coplanar$$

Have N points, $N\ unknown\ \lambda, 3\ unknown\ R\ terms, 3\ unknown\ T\ terms$

Could try using system of equations (linear algebra)

$$(\quad)_{12\times 12} \begin{pmatrix} r_{11} \\ r_{12} \\ r_{13} \\ r_{21} \\ r_{22} \\ r_{23} \\ r_{31} \\ r_{32} \\ r_{33} \\ t_1 \\ t_2 \\ t_3 \end{pmatrix} = 0, \text{ subject to } R^T R = I, \left|\left|R - R'\right|\right|_f^2$$

Needs 6 points – this is too many / inconvenient to provide
Could try to minimize error – no guaranteed solution within a reasonable amount of time

Pose from 3 points: the perspective 3-point problem (P3P)
Given: $d_{12}, d_{23}, d_{13}, \delta_{12}, \delta_{23}, \delta_{13}$
Find: d_1, d_2, d_3
Apply law of cosines to obtain 3 quadratic equations for 3 unknowns

$$d_{23}^2 = d_2^2 + d_3^2 - 2d_2 d_3 cos\delta_{23}$$
$$d_{13}^2 = d_1^2 + d_3^2 - 2d_1 d_3 cos\delta_{13}$$
$$d_{12}^2 = d_1^2 + d_2^2 - 2d_1 d_2 cos\delta_{12}$$
$$ax^2 + bx + c = 0$$
$$a_1 x^2 + b_1 xy + cy^2 = 0$$
$$a_2 x^2 + b_2 xy + c_2 y^2 + d_2 x + e_2 y + f_2 = 0$$

Can try to remove 1 equation (2 equations, 2 unknowns):

$$d_2 = u\, d_1; d_3 = v d_1$$
$$d_{23}^2 = u^2 d_1^2 + v^2 d_1^2 - 2uv d_1^2 cos\delta_{23}$$

$P_{c,i} = RP_{w,i} + T$ – now we're totally in 3D to 3D translation / rotation (known as 3D-3D registration)

Procrustes Problem: given 2 shapes find the scaling, rotation and translation that fits one into the other

How do we solve for R, T from n point correspondences? – need 3 points

$$argmin_{R\in SO(3),T\in R^3}\sum_{i=1}^{N}\left||A_i - RB_i - T|\right|_F^2, \min_{R\in SO(3)}||A-RB||$$

$$\frac{\partial}{\partial T} = \sum 2(A_i - RB_i - T)(-1) = 0 \rightarrow \sum A_i - RB_i = NT \rightarrow \frac{\sum A_i}{N} - \frac{R\sum B_i}{N} = T \rightarrow \bar{A} - R\bar{B} = T$$

$$\left||A|\right|_F^2 = tr(A^T A)$$

$$\left||A-RB|\right|_B^2 = tr(A-RB)^T(A-RB) = tr(A^T A) + tr(B^T B) - tr(A^T RB) - tr(B^T R^T A)$$

$$argmin_{R\in SO(3)} tr(RBA^T)$$

$$tr(RUSV^T) = tr(V^T RUSV^T V) = tr(ZS) = \sigma_1 z_{11} + \sigma_2 z_{22} + \sigma_3 z_{33} \le \sigma_1 + \sigma_2 + \sigma_3, where\ R = VU^T$$

$$\rightarrow Z \in O(3), z_{11} + z_{22} + z_{33} = 1, S = \begin{pmatrix} \sigma_1 & & \\ & \sigma_2 & \\ & & \sigma_3 \end{pmatrix}$$

2D-3D Pose:

P3P given triangle $P_1P_2P_3$ and its projection, find d_1, d_2, d_3 and R, T

Procrustes: $d_i \frac{p_i}{||p_i||} = RP_i + T$

Using at least 4 points: $P(n \ge 4)P$ -C_1, C_2, C_3, C_4 we can describe any other point X in a barycentric coordinates: $X = \alpha_1 C_1 + \alpha_2 C_2 + \alpha_3 C_3 + \alpha_4 C_4, \sum_{j=1}^{4} \alpha_j = 1$, note: not coplanar

Project (using calibration) $\gamma_1 c_1 = C_1 = RC_1 + T$

$X^{Cam} = \sum_{j=1}^{4} \alpha_j C_j^{Cam}$, in 3D wrt camera

Barycentric coordinates are invariant to rigid body transform

$\lambda x^{im} = \sum_{j=1}^{4} \alpha_j C_j^{Cam} \rightarrow \lambda x = \sum_{j=1}^{4} \alpha_j C_{jx}^{Cam}, \lambda y = \sum_{j=1}^{4} \alpha_j C_{jy}^{Cam}, \lambda = \sum_{j=1}^{4} \alpha_j C_{jz}^{Cam}$

We know a 5th point $X = \sum \alpha_j C_j$ in the world. We know α_j for this point. We also know its projection x.

$$(\alpha_1 C_{1z} + \alpha_2 C_{2z} + \alpha_3 C_{3z} + \alpha_4 C_{4z})x = (\alpha_1 C_{1x} + \alpha_2 C_{2x} + \alpha_3 C_{3x} + \alpha_4 C_{4x})$$

$(\alpha_1 C_{1z} + \alpha_2 C_{2z} + \alpha_3 C_{3z} + \alpha_4 C_{4z}) = \lambda$

$$(\alpha_1 C_{1z} + \alpha_2 C_{2z} + \alpha_3 C_{3z} + \alpha_4 C_{4z})y = (\alpha_1 C_{1y} + \alpha_2 C_{2y} + \alpha_3 C_{3y} + \alpha_4 C_{4y})$$

Known x, $\alpha_{j=1,..,4}$, unknown $C_j^{cam} = \begin{pmatrix} C_{jx} \\ C_{jy} \\ C_{jz} \end{pmatrix}$ → 12 unknowns of a homogeneous linear system

6 points X_i with their barycentric coordinates $\alpha_{j,i}$ (i = 1:6, j = 1:4) yield a solution for $C_{j=1\ldots4}^{cam}$

$X_{i=1\ldots6}^{cam} = RX_{i=1\ldots6}^{W} + T$ → Procrustes

First LINEAR method for $n \ge 6$ points

What if the barycentric coordinates are unknown?

Subset of X can be the control point C

openGV = open geometric vision

Find 3D pose of a camera from the image of a known cylinder

$$X_w^2 + Y_w^2 = \rho^2$$

$$\bar{X}_w = R^T(\bar{X}_{cyl} - T)$$

$$(X_w\ Y_w\ Z_w)\begin{pmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 0\end{pmatrix}\begin{pmatrix}X_w\\ Y_w\\ Z_w\end{pmatrix} = \rho^2$$

$$(X_c - T)^T R(\quad) R^{T(X_c - T)} = \rho^2, X_c = \lambda\begin{pmatrix}x\\ y\\ 1\end{pmatrix}$$

Projection of cylinder is set of rays tangent to the cylinder

Lemma: projection of any quadrant in space (can be a sphere, cylinder, ellipsoid, cone, paraboloid, hyperboloid) can be written as $X^T MX + m^T X + \mu = 0$, X is camera coordinates, $X = \lambda x$ where x is in image coordinates

$\lambda^2 x^T Mx + \lambda m^T x + \mu = 0$ for the ray to be tangential – only 1 solution for λ

$(m^T x)^2 - 4\mu x^T Mx = 0$ (discriminant) corresponding depth $\lambda = -\frac{m^T x}{2x^T Mx}$

Find nearest point on cylinder to camera:

$T = \gamma c; R = (c\ b\ a); b = a \times c\ , a\ is \perp\ to\ c$ (a vector from the camera to through the nearest point on the cylinder) and γ is the distance between the point on the cylinder and the camera

$$X = (c\ b\ a)\begin{pmatrix}x_w\\ y_w\\ z_w\end{pmatrix} + \gamma c$$

$$(x_w y_w z_w)\begin{pmatrix}1 & & \\ & 1 & \\ & & 0\end{pmatrix}\begin{pmatrix}x_w\\ y_w\\ z_w\end{pmatrix} = \rho^2$$

$$\gamma^2(c^T X)^2 + (\gamma^2 - \rho^2)(c^T X)^2 + (b^T X)^2 = 0$$

$(\gamma^2 - \rho^2)(b^T x)^2 - \rho^2(c^T x)^2 = 0$ projection of a cylinder (2 lines in the image, not necessarily parallel)

$$x = \begin{pmatrix}x\\ y\\ 1\end{pmatrix}, l^T x = 0$$

$$\left(\sqrt{\gamma^2 - \rho^2}\, b^T x + \rho c^T x\right)\left(\sqrt{\gamma^2 - \rho^2}\, b^T x - \rho c^T x\right) = 0$$

Line 1: $n_1^T x = 0 \to n_1 = \frac{1}{\gamma}(\sqrt{\gamma^2 - \rho^2}\, b + \rho c)$

Line 2: $n_2^T x = 0 \to n_2 = \frac{1}{\gamma}\left(\sqrt{\gamma^2 - \rho^2}\, b - \rho c\right)$

$n_1 \times n_2 \sim b \times c \sim a =$axis of cylinder

$$n_1 + n_2 \sim b$$

$n_1 - n_2 \sim c$ (unit vectors)

$$n_1^T n_2 = 1 - 2\rho^2/\gamma^2$$

Ackermann steering geometry – depending on how you turn translation direction differs

Cannot get absolute speed purely from optical flow (use additional depth sensing))

3/25

Robot Perception: 3D Velocities from Optical Flow

Closer points move faster than further points when camera is moving

Projection equations for calibrated camera: $p = \frac{1}{Z}P$

Differentiating wrt time yields: $\dot{p} = \frac{\dot{P}}{Z} - \frac{\dot{Z}}{Z}p$

Issues:

Some things don't move when they should / move faster than they should

Ex: Highlights (double-triple reflections of light)

$\dot{P} = -\Omega \times P - V$, V = velocity, Ω = angular velocity?

$$P^W = R_C^W P + T_C^W$$
$$\dot{P}^W = 0 = \dot{R}_C^W P \, R_C^W \dot{P} + \dot{T}_C^W$$
$$R_C^W \dot{P} = -\dot{R}_C^W P - \dot{T}_C^W$$
$$\dot{P} = -R_C^W \dot{R}_C^W P - R_C^W \dot{T}_C^W$$

$\dot{R} = R\widehat{\Omega}_{frame} = \widehat{\Omega}_{frame} R$ (which frame is which?)

$$\dot{P} = \widehat{\Omega} P - V, V = linear\ velocity\ in\ camera\ frame, \widehat{\Omega} = angular\ velocity\ in\ camera\ frame$$

$$\dot{P}_{optical\ flow} = \frac{1}{Z}(-\Omega \times P - V) - \frac{\dot{Z}}{Z} p = -\Omega \times p - \frac{V}{Z} - \frac{\dot{Z}}{Z} p$$

$$\dot{Z} = e_3^T \dot{p}, e_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

$$\dot{P} = \frac{1}{Z} e_3 \times (p \times V) + e_3 \times (p \times (p \times \Omega))$$

$$\dot{p} = \frac{1}{Z}\begin{pmatrix} xV_z - V_x \\ yV_z - V_y \end{pmatrix} + (\quad)_{2\times 3}\Omega = \dot{\mathrm{P}}_{\mathrm{trans}} + \dot{P}_{rot}$$

Efference copy (Corollary discharge) – biological – brain receives copy of eye movement control input (allows our brain to compensate for eye motion) - http://en.wikipedia.org/wiki/Efference_copy

$\dot{P}_{rot}$ depends only on x, y but not on depth

$$\dot{p} = \frac{V_z}{Z}\begin{pmatrix} xV_z - V_x \\ yV_z - V_y \end{pmatrix}, V_z \neq 0$$

Radial

Focus of Expansion = FOE = intersection of all lines of optical flow

$$||\dot{p}|| = \left|\frac{V_z}{Z}\right| \left|\left|\begin{pmatrix} x \\ y \end{pmatrix} - FOE\right|\right|$$
$$z \to \infty, \dot{p}_{trans} \to 0$$

Points at double distance moving with double speed produce the same flow: unknown scale

We can compute $\left|\frac{V_z}{Z}\right|$ [units: inverse time] but NOT Z

$\frac{Z}{V_z}$ is TTC (time to collision)

We cannot always assume $V_z \neq 0$

V is a point on the sphere – continuous epipolar constraints

$$\dot{P}_{trans}^T (p \times V) = 0$$
$$(\dot{p} \times p)^T V = 0$$
$$(\dot{p}_1 \times p_1)^T V = 0$$
$$(\dot{p}_2 \times p_2)^T V = 0$$
$$a^T V = 0, b^T V = 0, V \sim a \times b$$

Overdetermined homogeneous system: want to find the Nullspace

$$\begin{pmatrix} (\dot{p}_1 \times p_1)^T \\ (\dot{p}_2 \times p_2)^T \\ \vdots \\ (\dot{p}_n \times p_n)^T \end{pmatrix}_{n\times 3} \quad V = 0$$

$$A = (u_1\ u_2\ u_3)\begin{pmatrix} \sigma_1 & & \\ & \sigma_2 & \\ & & \sigma_3 \end{pmatrix}(v_1\ \ v_2\ \ v_3)^T$$

Nullspace is eigenvector(s) (v_3) corresponding to the smallest eigenvalue(s), typically the eigenvalue(s) is 0

$\dot{p} = \frac{1}{Z}F(x,y)V + G(x,y)\Omega$, Heeger and Jepson

$$\begin{pmatrix} \dot{p}_1 \\ \dot{p}_2 \\ \vdots \\ \dot{p}_n \end{pmatrix} = \dot{d} = \big(\phi(V)\big)\begin{pmatrix} \dfrac{1}{Z_1} = \lambda_1 \\ \dfrac{1}{Z_2} = \lambda_2 \\ \vdots \\ \dfrac{1}{Z_n} = \lambda_n \\ \Omega \end{pmatrix}_{n+3\times 1} \qquad inverse\ depth$$

$$\begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_n \end{pmatrix} = \phi(V)^+\dot{d}$$

$heat(V) = \left|\left|\dot{d} - \phi(V)\phi^+(V)\dot{d}\right|\right|^2 = argmin_{V\in S^2}\left|\left|(I - \phi(V)\phi^+(V))\dot{d}\right|\right|^2$, could treat as convolution to speed this up but this assumes optical flow is everywhere (which isn't true); this is an exhaustive search though

3/30

Optical flow → V, Ω, Z (up to a scale factor)

$$\mu q = R\lambda q + T$$
$$Q = RP + T$$

Known: p, q

Unknown: R, T, λ, μ

$$Q = RP + T \rightarrow Procrustes\ (no\ depth)$$
$$\mu q = RP + T \rightarrow PnP\ (1\ unknown\ depth)$$
$$T \times\ \mu q = T \times R\lambda p + T \times T \rightarrow q^T(T \times Rp) = 0, coplanar\ (epipolar\ constraint)$$

The plane spanned by the 3 vectors is called the epipolar plane

Pencil of planes

Epipoles: $e_p \sim -R^T T$; $e_q \sim T$

Epipoles → Focus of Expansion (direction of heading)

Epipolar constraint: $q^T(T \times Rp) = 0 \rightarrow q^T E\ p = 0, E = \hat{T}R$

$$q^T E\ p = 0 \rightarrow l_p^T p = 0, l_p = E^T q\ AND\ l_q^T q = 0, l_q = Ep$$

All pass through $e_p \sim -R^T T, l_p^T e_p = 0\ \forall l_p\ AND\ l_q^T e_q = 0\ \forall l_q$

Compute E-matrix from very few points

N correspondences $(p_i, q_i)_{i=1:N}$ $q_i^T E p_i = 0$

$$E = (e_1\ \ e_2\ \ e_3)$$

$$\begin{pmatrix} p_x q^T & p_y q^T & p_z q^T \end{pmatrix} \begin{pmatrix} e_1 \\ e_2 \\ e_3 \end{pmatrix} = 0$$

$$a = \begin{pmatrix} p_x q^T & p_y q^T & p_z q^T \end{pmatrix}$$

$\begin{pmatrix} a_1^T \\ a_2^T \\ \vdots \\ a_n^T \end{pmatrix} E' = 0$, solution to E' is in the nullspace. This works for $N \geq 8$ (general point 3D configuration)

$$E^T T = \left(\hat{T} R\right)^T T = R^T \hat{T}^T T\ = 0$$

$$\det(E) = 0$$

$$E E^T = \hat{T} R R^T \hat{T}^T = T T^T - T^T T I$$

$$\det(E E^T - \lambda I) = 0 \rightarrow \lambda_{min} = 0, \lambda_{1,2} = \left|\left|T\right|\right|^T$$

$$E E^T \rightarrow eigenvalues\ 0, \left|\left|T\right|\right|^2, \left|\left|T\right|\right|^2$$

$$E = U \begin{pmatrix} ||T|| & 0 & 0 \\ 0 & ||T|| & 0 \\ 0 & 0 & 0 \end{pmatrix} V^T$$

Can any 3x3 real matrix be essential (be decomposed into the product of a skew symmetric x orthogonal)? If a matrix is essential then it has 2 equal singular values and the 3rd = 0 (necessary). If a matrix has $\sigma_1 = \sigma_2 > 0$ and $\sigma_3 = 0$ then it can be decomposed to a skew symmetric $\hat{T}$ x orthogonal R

Lemma: if Q is orthogonal ($Q^T Q = I$) then $\widehat{Qa} = Q \hat{a} Q^T$

$$\widehat{Qa} b = Qa \times b = Q(a \times Q^T b) = Q \hat{a} Q^T b$$

$$\frac{E}{\left|\left|T\right|\right|} = U \begin{pmatrix} 1 & & \\ & 1 & \\ & & 0 \end{pmatrix} V^T$$

$$\begin{pmatrix} 1 & & \\ & 1 & \\ & & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} = -\widehat{\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}} R_{z,\frac{\pi}{2}} = -\widehat{e_z}^T R_{z,\pi/2}$$

$$E = U \begin{pmatrix} \sigma & & \\ & \sigma & \\ & & 0 \end{pmatrix} V^T = \sigma U \begin{pmatrix} 1 & & \\ & 1 & \\ & & 0 \end{pmatrix} V^T = \sigma U \hat{e}_z^T R_{z,\frac{\pi}{2}} V^T = \sigma U \hat{e}_z^T I R_{z,\frac{\pi}{2}} V^T$$

$$= \sigma U \hat{e}_z^T U^T U R_{z,\frac{\pi}{2}} V^T = -\sigma \widehat{U e_z} U R_{z,\frac{\pi}{2}} V^T$$

$$E = \sigma U \hat{e}_z^T U^T U R_{z,\frac{\pi}{2}} V^T = -\sigma \widehat{U e_z} U R_{z,\frac{\pi}{2}} V^T \rightarrow T = U e_z, R = U R_{z,\frac{\pi}{2}} V^T$$

$$[U, S, V^T] = svd(E), T = U(:,3); R = U \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} V^T$$

Just doing this will probably end up with your points being behind the camera

Possible ambiguities:

Twisted pair ambiguity

$$\begin{pmatrix} 1 & & \\ & 1 & \\ & & 0 \end{pmatrix} = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}\begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \hat{e}_z\, P_{z,-\pi/2}$$ is an alternative solution

$$E = \sigma U \widehat{e_z} U^T U P_{z,-\pi/2} V^T$$

Mirror Ambiguity

If T is a solution then –T is a solution as well (since T can be computed by a scale factor)

$$q^T(T \times Rp) = 0\ AND\ q^T(-T \times Rp) = 0$$

This gives us 4 possible solutions – check which one gives you positive depths for both μ, λ

4/1

Correspondences

- Noisy (noise follows $e^{-\frac{1}{2}\frac{noise^2}{2}}$)
- Outliers (___ do not __)
- Multiple models (background, other independent vehicle motions)

Hough, RANSAC, EM

$$x cos\theta + y sin\theta - d = 0\ better\ than\ y = ax + b\ (does\ not\ handle\ vertical\ lines)$$

$\theta \in [0,2\pi), d > 0\ \cup \theta \in [0,\pi), d = 0$ (need to allow d = 0 to allow line to pass through 0 but we cannot allow the full θ range in this case)

$$\frac{\partial}{\partial d} = -2\sum(x_1 cos\theta + y_1 sin\theta - d) = 0 \rightarrow d = cos\theta\frac{\sum x_i}{N} + sin\theta\frac{\sum y_i}{N}\ (centroid)$$

$$\sum\big((x_i - \bar{x})cos\theta + (y_i - \bar{y})sin\theta\big)^2 = \begin{pmatrix} cos\theta \\ sin\theta \end{pmatrix}^T \begin{pmatrix} \sum(x_i - \bar{x})^2 & \sum(\ \ )(\ \ ) \\ \sum(\ \ )(\ \ ) & \sum(y_i - \bar{y})^2 \end{pmatrix}\begin{pmatrix} cos\theta \\ sin\theta \end{pmatrix} \rightarrow \min_{\theta,\phi}$$

Auto-correlation; covariance; moment of inertia matrix (would be inertial tensor)

$$\eta = \begin{pmatrix} cos\theta \\ sin\theta \end{pmatrix}, \eta^T C \eta \rightarrow \min_{||\eta||=1}$$

$$\lambda_{min} \le \eta^T C \eta \le \lambda_{max}$$

$$\eta^T C \eta is\ symmetric\ positive\ semidefinite$$

Solution η is the eigenvector to λ_{min} $Cu = \lambda_{min}; u^T C u = \lambda_{min}$

Hough accumulator space – maxima of votes in Hough space

Radon Transform $V(\theta, d) = \int_x \int_y p(x,y)\delta(x cos\theta + y sin\theta - d)dx\, dy$

Parameters of an ellipsoid: 5 (minor, major axis, center, rotation)

Hough transform stops working after 3 dimensions

Probability to be an inlier ϵ

Minimal number of points M (=2 for linear, =3 circle / P3P, =4 collinear; =5 2D ellipse, Structure From Motion)

Probability to obtain an inlier M-tuple is ϵ^M

k-times = $(1 - \epsilon^M)^k$

at least 1 inlier: $1 - (1 - \epsilon^M)^k \ge P$

Visual Odometry Libraries: libviso

How to do HW

$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \sim K(r_1 \;\; r_2 \;\; t)\begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$ $(ground\ to\ img)$, assume K = I

Ground: $x^2 + y^2 - \rho^2 = 0$

$$(x \;\; y \;\; 1)\begin{pmatrix} 1 & & \\ & 1 & \\ & & -\rho^2 \end{pmatrix}\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = 0$$

$$(x \;\; y \;\; 1)(r_1 \;\; r_2 \; t)^{-T}(\quad)(r_1 \;\; r_2 \; t)^{-1}\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = 0$$

$$(r_1 \;\; r_2 \; t)^{-T}(\quad)(r_1 \;\; r_2 \; t)^{-1}$$

$$-r_3^T t = \det((r_1 \;\; r_2 \; t)^{-T}(\quad)(r_1 \;\; r_2 \; t)^{-1}) = (r_1 \times r_2)^T t$$