The Chain Rule, revisited

How to structure complicated derivatives

Niclas Börlin

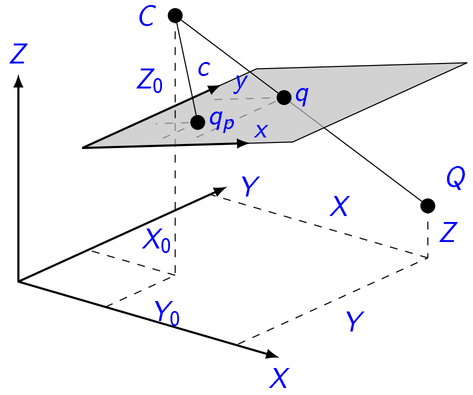
5DA001 Non-linear Optimization

The collinearity equations

- The *collinearity equations*

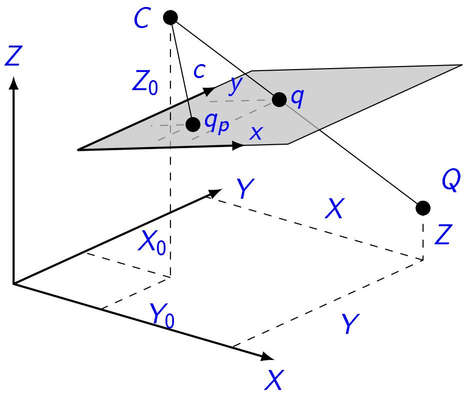
$$\begin{pmatrix} x - x_p \\ y - y_p \\ -c \end{pmatrix} = kR \begin{pmatrix} X - X_0 \\ Y - Y_0 \\ Z - Z_0 \end{pmatrix}$$

describe the relationship between the object point $(X, Y, Z)^T$, the position $C = (X_0, Y_0, Z_0)^T$ of the camera center and the orientation R^T of the camera.



The collinearity equations (2)

- The distance c is known as the *principal distance* or *camera constant*.
- The point $q_p = (x_p, y_p)^T$ is called the *principal point*.
- The ray passing through the camera center C and the principal point q_p is called the *principal ray*.



The collinearity equations (3)

- From

$$\begin{pmatrix} x - x_p \\ y - y_p \\ -c \end{pmatrix} = kR \begin{pmatrix} X - X_0 \\ Y - Y_0 \\ Z - Z_0 \end{pmatrix}, \text{ and } R = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix},$$

we can solve for k and insert:

$$x = x_p - c\frac{r_{11}(X - X_0) + r_{12}(Y - Y_0) + r_{13}(Z - Z_0)}{r_{31}(X - X_0) + r_{32}(Y - Y_0) + r_{33}(Z - Z_0)},$$
$$y = y_p - c\frac{r_{21}(X - X_0) + r_{22}(Y - Y_0) + r_{23}(Z - Z_0)}{r_{31}(X - X_0) + r_{32}(Y - Y_0) + r_{33}(Z - Z_0)}.$$

The rotation matrix

- The rotation in $\Re^3$ may be described as a sequence of 3 elementary rotations, by the so called *Euler angles*.
- Each elementary rotation takes place about a cardinal axis, x, y, or z.
- The sequence of axis determines the actual rotation.
- A common example is the $\omega - \varphi - \kappa$ (omega-phi-kappa or x-y-z) convention that correspond to sequential rotations about the x, y, and z axes, respectively.

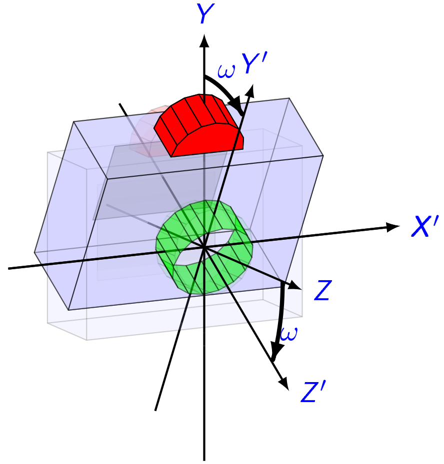
Elementary rotations (1)

- The first elementary rotation (ω, omega) is about the x-axis. The rotation matrix is defined as

$$R_1(\omega) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & c_\omega & -s_\omega \\ 0 & s_\omega & c_\omega \end{pmatrix},$$

where

$$c_\omega = \cos\omega,$$
$$s_\omega = \sin\omega.$$



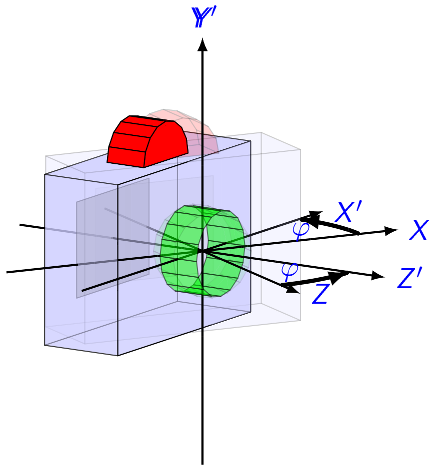
Elementary rotations (2)

- The second elementary rotation (φ, phi) is about the y-axis. The rotation matrix is defined as

$$R_2(\varphi) = \begin{pmatrix} c_\varphi & 0 & s_\varphi \\ 0 & 1 & 0 \\ -s_\varphi & 0 & c_\varphi \end{pmatrix},$$

where

$$c_\varphi = \cos\varphi,$$
$$s_\varphi = \sin\varphi.$$



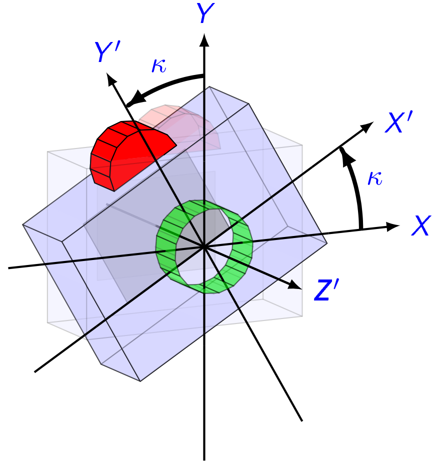
Elementary rotations (3)

- The third elementary rotation (κ, kappa) is about the z-axis. The rotation matrix is defined as

$$R_3(\kappa) = \begin{pmatrix} c_\kappa & -s_\kappa & 0 \\ s_\kappa & c_\kappa & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

where

$$c_\kappa = \cos\kappa,$$
$$s_\kappa = \sin\kappa.$$



Combined rotations

- The combined $\omega - \varphi - \kappa$ rotation matrix is thus

$$
\begin{aligned}
R(\omega,\varphi,\kappa) &= R_3(\kappa)R_2(\varphi)R_1(\omega) \\
&= \begin{pmatrix} c_\kappa & -s_\kappa & 0 \\ s_\kappa & c_\kappa & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} c_\varphi & 0 & s_\varphi \\ 0 & 1 & 0 \\ -s_\varphi & 0 & c_\varphi \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & c_\omega & -s_\omega \\ 0 & s_\omega & c_\omega \end{pmatrix} \\
&= \begin{pmatrix} c_\kappa c_\varphi & -s_\kappa c_\omega + c_\kappa s_\varphi s_\omega & s_\kappa s_\omega + c_\kappa s_\varphi c_\omega \\ s_\kappa c_\varphi & c_\kappa c_\omega + s_\kappa s_\varphi s_\omega & -c_\kappa s_\omega + s_\kappa s_\varphi c_\omega \\ -s_\varphi & c_\varphi s_\omega & c_\varphi c_\omega \end{pmatrix}.
\end{aligned}
$$

Summary

- Thus, the collinearity equations are

$$
\begin{aligned}
x &= x_p - c\frac{r_{11}(X - X_0) + r_{12}(Y - Y_0) + r_{13}(Z - Z_0)}{r_{31}(X - X_0) + r_{32}(Y - Y_0) + r_{33}(Z - Z_0)}, \\
y &= y_p - c\frac{r_{21}(X - X_0) + r_{22}(Y - Y_0) + r_{23}(Z - Z_0)}{r_{31}(X - X_0) + r_{32}(Y - Y_0) + r_{33}(Z - Z_0)}
\end{aligned}
$$

with

$$
R = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} = \begin{pmatrix} c_\kappa c_\varphi & -s_\kappa c_\omega + c_\kappa s_\varphi s_\omega & s_\kappa s_\omega + c_\kappa s_\varphi c_\omega \\ s_\kappa c_\varphi & c_\kappa c_\omega + s_\kappa s_\varphi s_\omega & -c_\kappa s_\omega + s_\kappa s_\varphi c_\omega \\ -s_\varphi & c_\varphi s_\omega & c_\varphi c_\omega \end{pmatrix}.
$$

- Now, imagine you need to derive the Jacobian of this...

Deriving the Jacobian

- Solution 1: Use the numerical approximation.
- Solution 2: Derive expressions using a computer algebra system, e.g. Maple or Mathematica.
- Solution 3: Exploit the structure and the chain rule to divide-and-conquer.

The Chain Rule

Scalar version

- Consider a function f of two parameters x and y

$$f(x,y).$$

- If the function is part of an optimization, we may need the Jacobian of f with respect to x and y

$$J^f_{x,y} = \begin{pmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} \end{pmatrix}.$$

- If the function expression is complex, we may benefit from splitting the expression into a *composition* of functions.
- Introduce the intermediate functions $u(x,y)$ and $v(x,y)$ such that

$$f(u(x,y), v(x,y)).$$

- The partial derivatives are given by the *chain rule*

$$
\begin{aligned}
\frac{\partial f}{\partial x} &= \frac{\partial f}{\partial u}\frac{\partial u}{\partial x} + \frac{\partial f}{\partial v}\frac{\partial v}{\partial x}, \\
\frac{\partial f}{\partial y} &= \frac{\partial f}{\partial u}\frac{\partial u}{\partial y} + \frac{\partial f}{\partial v}\frac{\partial v}{\partial y}.
\end{aligned}
$$

The Chain Rule

Vector version

- Notice that if the variables (x, y) are collected in a vector X and a vector-valued function U is contructed

$$U(X) = \begin{pmatrix} u(X) \\ v(X) \end{pmatrix},$$

then

$$J_X^U = \nabla_X U^T = \begin{pmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} \end{pmatrix}$$

is the Jacobian of U with respect to X.

- Similarly, as

$$J_U^f = \begin{pmatrix} \frac{\partial f}{\partial u} & \frac{\partial f}{\partial v} \end{pmatrix},$$

the total Jacobian is just the *product* of the intermediate Jacobians

$$J_X^f = J_U^f J_X^U = \begin{pmatrix} \frac{\partial f}{\partial u} & \frac{\partial f}{\partial v} \end{pmatrix} \begin{pmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} \end{pmatrix} = \begin{pmatrix} \frac{\partial f}{\partial u}\frac{\partial u}{\partial x} + \frac{\partial f}{\partial v}\frac{\partial v}{\partial x} & \frac{\partial f}{\partial u}\frac{\partial u}{\partial y} + \frac{\partial f}{\partial v}\frac{\partial v}{\partial y} \end{pmatrix}.$$

The collinearity equations

- Notice that the collinearity equations

$$x = x_p - c\frac{r_{11}(X - X_0) + r_{12}(Y - Y_0) + r_{13}(Z - Z_0)}{r_{31}(X - X_0) + r_{32}(Y - Y_0) + r_{33}(Z - Z_0)},$$
$$y = y_p - c\frac{r_{21}(X - X_0) + r_{22}(Y - Y_0) + r_{23}(Z - Z_0)}{r_{31}(X - X_0) + r_{32}(Y - Y_0) + r_{33}(Z - Z_0)}$$

may be rewritten using the substitutions

$$\begin{aligned} U &= r_{11}(X - X_0) + r_{12}(Y - Y_0) + r_{13}(Z - Z_0) \\ V &= r_{21}(X - X_0) + r_{22}(Y - Y_0) + r_{23}(Z - Z_0) \\ W &= r_{31}(X - X_0) + r_{32}(Y - Y_0) + r_{33}(Z - Z_0) \end{aligned}$$

into

$$\begin{aligned} x &= x_p - c\frac{U}{W}, \\ y &= y_p - c\frac{V}{W}. \end{aligned}$$

The collinearity equation

Composed version

- Using the following vectors

$$p = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}, p_0 = \begin{pmatrix} X_0 \\ Y_0 \\ Z_0 \end{pmatrix}, \alpha = \begin{pmatrix} \omega \\ \phi \\ \kappa \end{pmatrix}, k = \begin{pmatrix} c \\ x_p \\ y_p \end{pmatrix}$$

the collinearity equations may be rewritten using intermediate functions

Rotation matrix	$M(\alpha) = R(\omega, \varphi, \kappa),$
3D translate, rotate	$q(p, p_0, \alpha) = \begin{pmatrix} U \\ V \\ W \end{pmatrix} = M(\alpha)(p - p_0),$
Project 3D→2D	$f(q, p, p_0, \alpha, k) = \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x_p - c\frac{U}{W} \\ y_p - c\frac{V}{W} \end{pmatrix}.$

The collinearity equations

Partial Jacobians

- Now most partial Jacobians are fairly straightforward:

$$\begin{aligned} J_p^q &= M(\alpha), \\ J_{p_0}^q &= -M(\alpha), \\ J_k^f &= \begin{pmatrix} 1 & 0 & -\frac{U}{W} \\ 0 & 1 & -\frac{V}{W} \end{pmatrix}, \\ J_q^f &= \begin{pmatrix} -\frac{c}{W} & 0 & \frac{cU}{W^2} \\ 0 & -\frac{c}{W} & \frac{cV}{W^2} \end{pmatrix}, \\ J_p^f &= J_q^f J_p^q = \begin{pmatrix} -\frac{c}{W} & 0 & \frac{cU}{W^2} \\ 0 & -\frac{c}{W} & \frac{cV}{W^2} \end{pmatrix} M(\alpha), \\ J_{p_0}^f &= J_q^f J_{p_0}^q = \begin{pmatrix} -\frac{c}{W} & 0 & \frac{cU}{W^2} \\ 0 & -\frac{c}{W} & \frac{cV}{W^2} \end{pmatrix} (-M(\alpha)). \end{aligned}$$

The collinearity equations

Implementation

- If the intermediate functions also compute the respective Jacobians, the complexity of each function can be significantly reduced.
- ```
  function [M,dMdo,dMdp,dMdk]=M(alpha)
  % Compute the rotation matrix.
  M=...
  % Compute the partial derivatives.
  dMdo=...
  dMdp=...
  dMdk=...
  ```
- ```
  function [qq,dqdp,dqdp0,dqda]=q(p,p0,alpha)
  % Get the rotation matrix and partials.
  [Ma,dMdo,dMdp,dMdk]=M(alpha);
  % Compute q.
  qq=Ma*(p-p0);
  % Compute partials.
  dqdp=Ma;
  dqdp0=-Ma;
  dqda=[dMdo*(p-p0),dMdp*(p-p0),dMdk*(p-p0)];
  ```


The collinearity equations

Implementation

- ```
  function [r,J]=f(x)
  % Unpack x vector.
  k=x(1:3); p=x(4:6); p0=x(7:9); alpha=x(10:12);
  % Get q and partials.
  [qq,dqdp,dqdp0,dqda]=q(p,p0,alpha);
  UV=qq(1:2); W=qq(3);
  % Compute f.
  c=k(1); xyp=k(2:3);
  r=xyp-c/W*UV;
  % Compute partials.
  dfdk=[eye(2),-UV/W];
  qfdq=[-c/W*eye(2),c/(W*W)*UV];
  % Combine partials to form complete Jacobian.
  J=[dfdk,dfdq*[dqdp,dqdp0,dqda]];
  ```
  ```