```
% ro1.m  -  6-oct-04
% relative orientation by collinearity

nrow=1920;
ncol=2560;
cl=[78;427;1121;1757;1909;33;1486;458;2082;2260;757;1043;72];
rl=[793;262;136;409;1696;1216;1528;1741;948;1507;1690;593;1687];
cr=[155;376;1092;1801;1901;97;1559;297;2297;2505;1089;990;113];
rr=[853;315;106;310;1715;1254;1531;1749;872;1494;1699;579;1698];

r0=nrow/2;
c0=ncol/2;

tcl=cl-c0;
trl=rl-r0;
tcr=cr-c0;
trr=rr-r0;

xl=tcl;
yl=-trl;
xr=tcr;
yr=-trr;

% units: feet, by pacing
Base=25;
Height=70;
B_H=Base/Height;

Xl=0.0;
Yl=0.0;
Zl=0.0;
Xr=Base;
Yr=0.0;
Zr=0.0;
alpha=atan((Base/2) / Height);
wl=0.0;
pl=-alpha;
kl=0.0;
wr=0.0;
pr=alpha;
kr=0.0;

phang_l=[wl;pl;kl];
phang_r=[wr;pr;kr];
phxyz_l=[Xl;Yl;Zl];
phxyz_r=[Xr;Yr;Zr];

% determined by brute force minimization
x0=7;
y0=-55;
foc=2590;
```

```
k1=0.0;
k2=0.0;
k3=0.0;
cam=[x0;y0;foc;k1;k2;k3];
load -ascii delta.dat

% make initial approximations for the obect points

[m,n]=size(cl);
npt=m;
X=zeros(npt,1);
Y=zeros(npt,1);
Z=zeros(npt,1);

for i=1:npt
  B=zeros(4,3);
  f=zeros(4,1);
  ndx=1;

  % first the left photo
  x=xl(i);
  y=yl(i);
  [c1,c2,f1,f2]=int_leq(x,y,phxyz_l,phang_l,cam);
  B(ndx:ndx+1,:)=[-1 0 c1; 0 -1 c2];
  f(ndx:ndx+1)=[f1;f2];
  ndx=ndx+2;

  % next the right photo
  x=xr(i);
  y=yr(i);
  [c1,c2,f1,f2]=int_leq(x,y,phxyz_r,phang_r,cam);
  B(ndx:ndx+1,:)=[-1 0 c1; 0 -1 c2];
  f(ndx:ndx+1)=[f1;f2];

  % ok now solve the LS intersection problem
  N=B'*B;
  t=B'*f;
  del=inv(N)*t;
  X(i)=del(1);
  Y(i)=del(2);
  Z(i)=del(3);
  end

% carry as unknowns: Yr,Zr,wr,pr,kr plus m-ground points
n=npt*2*2;
nu=5 + npt*3;
r=n-nu;

B=zeros(npt*2*2,nu);
f=zeros(m*2*2,1);
b=zeros(2,18);
```

```
converged=0;
threshold=1.0e-06;
prior_sighat=9.99e+09;
% ok now big iteration loop
for iter=1:10
  rndx=1;
  for i=1:npt
    ptxyz=[X(i);Y(i);Z(i)];

    % 2 equations from the left
    x=xl(i);
    y=yl(i);
    b=gencof(x,y,phang_l,phxyz_l,ptxyz,cam,delta);
    %disp('left');
    %keyboard

    cndx=5 + (i-1)*3 + 1;
    B(rndx,cndx)=b(1,15);
    B(rndx,cndx+1)=b(1,16);
    B(rndx,cndx+2)=b(1,17);
    f(rndx)=-b(1,18);
    B(rndx+1,cndx)=b(2,15);
    B(rndx+1,cndx+1)=b(2,16);
    B(rndx+1,cndx+2)=b(2,17);
    f(rndx+1)=-b(2,18);
    rndx=rndx+2;

    % now 2 equations from the right
    x=xr(i);
    y=yr(i);
    b=gencof(x,y,phang_r,phxyz_r,ptxyz,cam,delta);
    %disp('right');
    %keyboard
    B(rndx,1)=b(1,13);
    B(rndx,2)=b(1,14);
    B(rndx,3)=b(1,9);
    B(rndx,4)=b(1,10);
    B(rndx,5)=b(1,11);

    B(rndx,cndx)=b(1,15);
    B(rndx,cndx+1)=b(1,16);
    B(rndx,cndx+2)=b(1,17);
    f(rndx)=-b(1,18);

    B(rndx+1,1)=b(2,13);
    B(rndx+1,2)=b(2,14);
    B(rndx+1,3)=b(2,9);
    B(rndx+1,4)=b(2,10);
    B(rndx+1,5)=b(2,11);

    B(rndx+1,cndx)=b(2,15);
    B(rndx+1,cndx+1)=b(2,16);
    B(rndx+1,cndx+2)=b(2,17);
    f(rndx+1)=-b(2,18);
    rndx=rndx+2;
    end
  N=B'*B;
  t=B'*f;
  del=inv(N)*t;
  % now update the parameters
  phxyz_r(2)=phxyz_r(2) + del(1);
  phxyz_r(3)=phxyz_r(3) + del(2);
  phang_r(1)=phang_r(1) + del(3);
  phang_r(2)=phang_r(2) + del(4);
  phang_r(3)=phang_r(3) + del(5);
  for i=1:npt
    ndx=5 + (i-1)*3 + 1;
    X(i)=X(i) + del(ndx);
    Y(i)=Y(i) + del(ndx+1);
    Z(i)=Z(i) + del(ndx+2);
    end

  v=f - B*del;
  sig2hat=(v'*v/r);
  sighat=sqrt(sig2hat);
  [iter sighat]

  d_sighat=abs(sighat - prior_sighat);
  if((d_sighat/sighat) < threshold)
    disp('we have converged');
    converged=1;
    break;
    end
  prior_sighat=sighat;
  % ok next iteration
  end

if(converged == 0)
  disp('NOT CONVERGED');
  end

disp('results');

% for final results uncomment display of residuals
% disp('residuals');
% v

disp('exp sta left / angles left');
phxyz_l'
phang_l'
disp('exp sta right / angles right');
phxyz_r'
phang_r'
disp('x0 y0 foc');
```

```
[x0 y0 foc]
```

```
% gencof.m  2-jan-03
% generate coefficients of collinearity equations
% for bundle program, by numerical approximation
% result b is 2 x 18 matrix with the following elements
%
% | dFx/dx dFx/dy dFx/dx0 dFx/dy0 dFx/df dFx/dk1 dFx/dk2 dFx/dk3
% | dFy/dx dFy/dy dFy/dx0 dFy/dy0 dFy/df dFy/dk1 dFy/dk2 dFy/dk3
%
% dFx/dw dFx/dp dFx/dk dFx/dXL dFx/dYL dFx/dZL
% dFy/dw dFy/dp dFy/dk dFy/dXL dFy/dYL dFy/dZL
%
% dFx/dX dFx/dY dFx/dZ Fx |
% dFy/dX dFy/dY dFy/dZ Fy |
%

function b=gencof(x,y,phang,phxyz,ptxyz,cam,dl)
p=zeros(17,1);
b=zeros(2,18);
p(1)=x;
p(2)=y;
ndx=3;
for i=1:6
  p(ndx)=cam(i);
  ndx=ndx+1;
  end
for i=1:3
  p(ndx)=phang(i);
  ndx=ndx+1;
  end
for i=1:3
  p(ndx)=phxyz(i);
  ndx=ndx+1;
  end
for i=1:3
  p(ndx)=ptxyz(i);
  ndx=ndx+1;
  end
F=collin(p);
for i=1:17
  pwk=p;
  pwk(i)=p(i)+dl(i);
  Fp=collin(pwk);
  dFxdp=(Fp(1)-F(1))/dl(i);
  dFydp=(Fp(2)-F(2))/dl(i);
  b(1,i)=dFxdp;
  b(2,i)=dFydp;
  end
b(1,18)=F(1);
b(2,18)=F(2);
```

```
% collin.m  2-jan-03
% evaluate collinearity equation for given parameters
% note we are scaling the lens distortion coefficients
% for numerical stability

function F=collin(p)
x=p(1);
y=p(2);
x0=p(3);
y0=p(4);
foc=p(5);
k1=p(6);
k2=p(7);
k3=p(8);
w=p(9);
h=p(10);
k=p(11);
XL=p(12);
YL=p(13);
ZL=p(14);
X=p(15);
Y=p(16);
Z=p(17);

xp=x-x0;
yp=y-y0;
r=sqrt(xp^2 + yp^2);
c1=1/90^3;
c2=1/90^5;
c3=1/90^7;
% note: following gives problem for r=0
% dr=c1*k1*r^3 + c2*k2*r^5 + c3*k3*r^7;
% dx=dr*(xp/r);
% dy=dr*(yp/r);
% revise (equivalently) as
dr=c1*k1*r^2 + c2*k2*r^4 + c3*k3*r^6;
dx=dr*xp;
dy=dr*yp;

xpp=xp+dx;
ypp=yp+dy;
mw=[1 0 0;0 cos(w) sin(w);0 -sin(w) cos(w)];
mp=[cos(h) 0 -sin(h);0 1 0;sin(h) 0 cos(h)];
mk=[cos(k) sin(k) 0;-sin(k) cos(k) 0;0 0 1];
m=mk*mp*mw;
DX=[X-XL;Y-YL;Z-ZL];
UVW=m*DX;
Fx=xpp + foc*(UVW(1)/UVW(3));
Fy=ypp + foc*(UVW(2)/UVW(3));
F=[Fx;Fy];
```

```
% int_leq.m  31-dec-02
% intersection equation coefficients for pseudo-linear
% formulation
function [c1,c2,f1,f2]=int_leq(x,y,phoc,phoa,cm)

om=phoa(1);
ph=phoa(2);
kp=phoa(3);
x0=cm(1);
y0=cm(2);
foc=cm(3);
XL=phoc(1);
YL=phoc(2);
ZL=phoc(3);
mw=[1 0 0;0 cos(om) sin(om);0 -sin(om) cos(om)];
mp=[cos(ph) 0 -sin(ph);0 1 0;sin(ph) 0 cos(ph)];
mk=[cos(kp) sin(kp) 0;-sin(kp) cos(kp) 0;0 0 1];
m=mk*mp*mw;
mt=m';
vc=[x-x0; y-y0; -foc];
uvw=mt*vc;
c1=uvw(1)/uvw(3);
c2=uvw(2)/uvw(3);
f1=c1*ZL - XL;
f2=c2*ZL - YL;
```