

These methods are described in

Ercan I, Ocakoglu G, Guney I, Yazici B.

“Adaptation of Generalizability Theory for Inter-Rater Reliability for Landmark Localization”

International Journal of Tomography & Statistics, 2008, Vol. 9, No. S08:51-58

<http://www.ceser.res.in/isder/ijts/cont/2008/ijts-s08-abs.html>

x_{ijr} : i^{th} subject, j^{th} landmark, r^{th} rater

Table 1: The coordinates of landmark locations of raters on the subjects

subject	Landmark	Rater 1		Rater 2	
		x	y	x	y
1	1	x_{111}	y_{111}	x_{112}	y_{112}
	2	x_{121}	y_{121}	x_{122}	y_{122}
	:	:	:	:	:
	k	x_{1k1}	y_{1k1}	x_{1k2}	y_{1k2}
2	1	x_{211}	y_{211}	x_{212}	y_{212}
	2	x_{221}	y_{221}	x_{222}	y_{222}
	:	:	:	:	:
	k	x_{2k1}	y_{2k1}	x_{2k2}	y_{2k2}
.	1	:	:	:	:
	2	:	:	:	:
	:	:	:	:	:
	k	:	:	:	:
.	1	:	:	:	:
	2	:	:	:	:
	:	:	:	:	:
	k	:	:	:	:
n	1	x_{n11}	y_{n11}	x_{n12}	y_{n12}
	2	x_{n21}	y_{n21}	x_{n22}	y_{n22}
	:	:	:	:	:
	k	x_{nk1}	y_{nk1}	x_{nk2}	y_{nk2}

The Matlab program for inter-rater reliability for landmark localization in the case of two or three dimension landmark locations. Data matrixes are constructed for rater 1 (Table 2) and rater 2 (Table 3) before running the program.

After constructing the data matrixes the following program is run:

```
% adaptation of generalizability theory for inter-rater
% reliability for landmark localization
% for 2 or 3 dimensions
db=input('input the dimension:')
n=input('input the number of the subjects:')
k=input('input the number of the landmarks:')
a=0; rr=0; ka=0; kb=0; sn1=1; tlsr=0; tlr2=0; tr2=0; tr1a=0; tr1b=0;
tl2=0; ts2=0; tlsb2=0; td=0; tlre=0; tsre=0; sn2=0; d=0; trs2=0; srkt=0;
C = nchoosek(k,2); sn2=(n*(C)); s=C*n
%Constructing coding matrix for (R) landmark pair(l) x rater(r) x subject(s)
for a=1:2
    for b=1:n
        for c=1:C
            d=d+1; R(d,1)=a; R(d,2)=b; R(d,3)=c;
        end
    end
end
end
% Reading the matrixes A and B for rater 1 and Rater 2
for i=1:(n*k)
    for j=1:db
        A(i,j)
        B(i,j)
    end
end
end
%Calculation of the Euclidean distances between landmark pairs for raters
if (db<3)
% calculations for two dimensions
    for p=1:k:(n*k)
        t=k+p; m=t-k;
        for j=m:(t-1)
```

```

for i=m:(t-1)
    if (j<i) & (i~=j)
        acia1=sqrt(((A(j,1)-A(i,1))^2)+((A(j,2)-A(i,2))^2));
        R(sn1,4)=acia1; sn2=sn2+1;
        acib1=sqrt(((B(j,1)-B(i,1))^2)+((B(j,2)-B(i,2))^2));
        R(sn2,4)=acib1;
    % calculation of sum of squares for lxrxs
        tlsr=tlsr+(acia1^2)+(acib1^2);
    % calculation of sum of squares for r (cont. Outside the loop)
        tr1a=tr1a+acia1; tr1b=tr1b+acib1;
    % calculation of adjustment factor (cont. Outside the loop)
        td=td+acia1+acib1;
        sn1=sn1+1;
    end
end
end
end
else
%calculations for three dimensions
for p=1:k:(n*k)
    t=k+p; m=t-k;
    for j=m:(t-1)
        for i=m:(t-1)
            if (j<i) & (i~=j)
                acia1=sqrt(((A(j,1)-A(i,1))^2)+((A(j,2)-A(i,2))^2)+((A(j,3)-A(i,3))^2));
                R(sn1,4)=acia1; sn2=sn2+1;
                acib1=sqrt(((B(j,1)-B(i,1))^2)+((B(j,2)-B(i,2))^2)+((B(j,3)-B(i,3))^2));
                R(sn2,4)=acib1;
            % calculation of sum of squares for lxrxs
                tlsr=tlsr+(acia1^2)+(acib1^2);
            % calculation of sum of squares for r (cont. Outside the loop)
                tr1a=tr1a+acia1; tr1b=tr1b+acib1;
            % calculation of adjustment factor (cont. Outside the loop)

```



```

        end
    end
    tl2=tl2+(tl1^2);
end
lkt=(tl2/(n*2))-df
% calculation of sum of squares for s
for sb=1:n
    ts1=0;
    for i=1:s
        if (sb>(P(i,2)-1)) & (sb<(P(i,2)+1))
            ts1=ts1+P(i,4);
        end
        if (sb>(V(i,2)-1)) & (sb<(V(i,2)+1))
            ts1=ts1+V(i,4);
        end
    end
end
ts2=ts2+(ts1^2);
end
skt=(ts2/(C*2))-df;
% calculation of sum of squares for lxs
for l=1:C
    for sb=1:n
        tls1=0;
        for i=1:s
            if (l>(P(i,3)-1)) & (l<(P(i,3)+1))
                if (sb>(P(i,2)-1)) & (sb<(P(i,2)+1))
                    tls1=tl1+P(i,4);
                end
            end
            if (l>(V(i,3)-1)) & (l<(V(i,3)+1))
                if (sb>(V(i,2)-1)) & (sb<(V(i,2)+1))
                    tls1=tl1+V(i,4);
                end
            end
        end
    end
end

```

```

        end
    end
    tlsb2=tlsb2+(tls1^2);
end
end
lskt=(tlsb2/2)-df-lkt-skt;
% calculation of sum of squares for lxr
for l=1:C
    tlrp=0;
    tlrsv=0;
    for i=1:s
        if (l>(P(i,3)-1)) & (l<(P(i,3)+1))
            tlrp=tlrp+P(i,4); tlrsv=tlrsv+V(i,4);
        end
    end
    tlr=tlrp+(tlrp^2+tlrsv^2);
end
lrkt=(tlr/n)-df-lkt-rkt;
% calculation of sum of squares for rx
for r=1:2
    for sb=1:n
        trsp=0; trsv=0;
        for i=1:s
            if (r>(P(i,1)-1)) & (r<(P(i,1)+1))
                if (sb>(P(i,2)-1)) & (sb<(P(i,2)+1))
                    trsp=trsp+P(i,4);
                end
            end
            if (r>(V(i,1)-1)) & (r<(V(i,1)+1))
                if (sb>(V(i,2)-1)) & (sb<(V(i,2)+1))
                    trsv=trsv+V(i,4);
                end
            end
        end
    end
end
end

```

```

        end
        trs2=trs2+(trsp^2)+(trsv^2);
    end
end
srkt=(trs2/C)-df-rkt-skt;
% calculation of sum of squares for lrxs
tlsrkt=tlsr-df-lkt-rkt-skt-lrkt-lskt-srkt
% calculation of Mean Squares
ms_lrs=tlsrkt/((n-1)*(C-1));
ms_lr=lrkt/(C-1);
ms_ls=lskt/((n-1)*(C-1));
ms_rs=srkt/(n-1);
ms_l=lkt/(C-1);
ms_s=skt/(n-1);
% estimates of the variance components
var_lrs=ms_lrs;
var_lr=(ms_lr-ms_lrs)/n;
var_ls=(ms_ls-ms_lrs)/2;
var_rs=(ms_rs-ms_lrs)/C;
var_l=(ms_l-ms_lr-ms_ls+ms_lrs)/(2*n);
var_r=(ms_r-ms_lr-ms_rs+ms_lrs)/(C*n);
var_s=(ms_s-ms_ls-ms_rs+ms_lrs)/(2*C);
var_rel=(var_lr/2)+(var_ls/n)+(var_lrs/n);
if (var_lrs<0)
    var_lrs=0;
end
if (var_lr<0)
    var_lr=0;
end
if (var_ls<0)
    var_ls=0;
end
if (var_rs<0)

```

```

    var_rs=0;
end
if (var_l<0)
    var_l=0;
end
if (var_r<0)
    var_r=0;
end
if (var_s<0)
    var_s=0;
end
% calculation of G coefficient
G=var_l/(var_l+var_rel);
display('Rater-->r, Subject-->s, Landmark pairs-->l')
display(' r s l Euclidean distances'); R
display('r sum of square'), rkt
display('l sum of square'), lkt
display('s sum of square'), skt
display('lxs sum of square'), lskt
display('lxr sum of square'), lrkt
display('rxs sum of square'), srkt
display('lxsxr sum of square'), tlsrkt
display('r mean square'), ms_r
display('l mean square'), ms_l
display('s mean square'), ms_s
display('lxs mean square'), ms_ls
display('lxr mean square'), ms_lr
display('rxs mean square'), ms_rs
display('lrxs mean square'), ms_lrs
display('variance of r'), var_r
display('variance of l'), var_l
display('variance of s'), var_s
display('variance of lxs'), var_ls

```

```
display('variance of lxr'), var_lr  
display('variance of rxs'), var_rs  
display('variance of lrxs'), var_lrs  
display('variance of rel'), var_rel  
display('G coefficient'), G
```