080315a tpsRelw.mcd

Ordinations on Partial & Relative Warps

prepared by
Wm Stein

The TPS series of programs for Windows may be found at:

ORIGIN := 0

http://life.bio.sunysb.edu/morph/

Look under "software" "thin-plate spline"

The TPS series supports several different activities including data collection (tps.Dig2) and subsequent analysis (tpsSplin, tpsRelw, tpsTree, tpsRegr, tpsPLS, tpsSmall). The object of the tpsRelw program is to implement F. James Rohlf & D.E. Slice's (1990 Systematic Zoology 39:40-59) "GPA consensus" configuration of landmark points (LM) to define a "reference" form, followed by calculation of Fred Bookstein's (1991 Cambridge Univ. Press) "partial warps" and "relative warps". Graphs produced by tpsRelw show objects *as single points* and are to be considered "ordinations" much in the sense of Principal Components Analysis (PCA). Thin-plate spline ordinations involve the same style of matrix decomposition using eigenvalues and eigenvectors as PCA. Partial Warp decomposition involves "spectral decomposition" of Bookstein's matrix L_K^{-1} followed by projection with Bookstein's vectors V to produce "Partial Warps". A separate calculation is also made of "scores" (or "weights") describing the general affine component to shape variation. Following this, a simple PCA is done (by decomposition of the variance-covariance matrix, or equivalently by singular value decomposition) of a combined matrix of "scores" (or Rolf's "weights"), as shown below. The ordination approach described here is statistically "guilt free", and exploratory, in the sense that no specific model of shape change is assumed, and no statistical hypothesis is formed or tested.

Reading the Data:

The program tpsRelw allows input of multiple sets of LM points from which it calculates a "consensus" based on Rolf & Slice's "GPA method" involving alignment of the original specimens by translation, rotation and scaling into a common reference system. The program then allows these values to be saved into text datafiles in either *.tps or *.nps format. GPA concensus will not be evaluated here. To work with the consensus configuration, however, tpsRelw output was edited to strip out labels leaving nothing but LM locations in the common GPA reference system. We'll use this refernce set of LM as if originally digitized by the researcher.

Fr := READPRN("c:/2008Morphometrics/fossilconsensus.dta") **< "reference" form calculated as a GPA consensus in tpsRelw**

Fd := READPRN("c:/2008Morphometrics/fossilalligned.dta") **< 55 "data" forms in a single file**

LMnum := rows(Fr) LMnum = 8 **< number of landmarks (LM) per object (OBJ)**

$$\text{OBJnum} := \frac{\text{rows(Fd)}}{\text{LMnum}} \qquad \text{OBJnum} = 55$$

< number of objects (OBJ)

$$\text{Fr} = \begin{pmatrix} -0.3789487 & 0.1375346 \\ 0.0784491 & 0.115048 \\ 0.6592233 & 0.118948 \\ 0.1414199 & -0.0888444 \\ -0.1906962 & -0.2409879 \\ -0.4248481 & -0.0506075 \\ -0.1092035 & -0.0235232 \\ 0.2246042 & 0.0324325 \end{pmatrix}$$

< LM for reference OJB

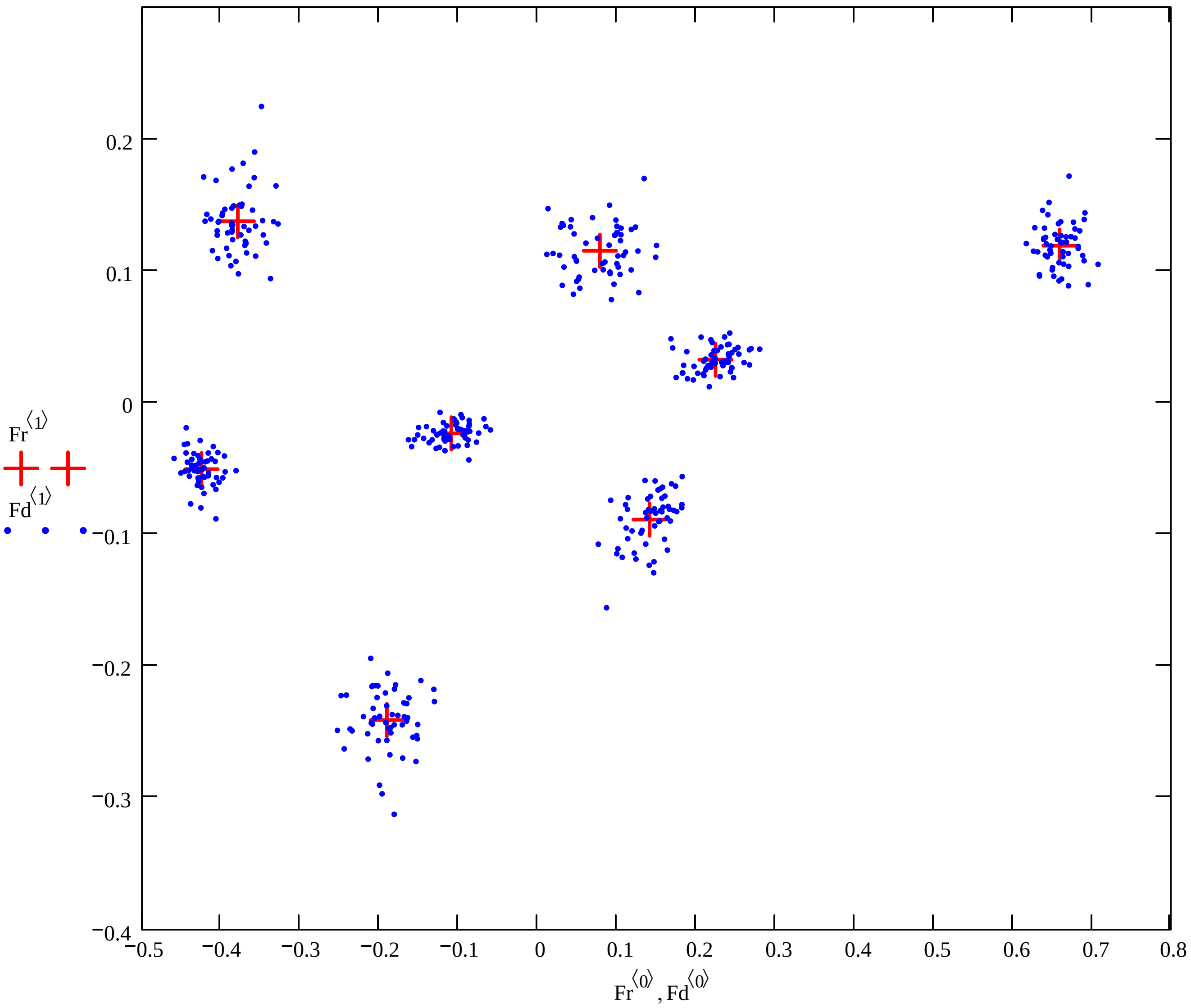
55 OBJ with 8 LM each in consensus allignment >

Fd =

	0	1
0	-0.3954	0.1466
1	0.0486	0.1079
2	0.6417	0.1253
3	0.1241	-0.1189
4	-0.1919	-0.243
5	-0.4265	-0.0453
6	-0.068	-0.0124
7	0.2674	0.04
8	-0.3766	0.1498
9	0.0308	0.1357
10	0.6505	0.1023
11	0.1757	-0.0829
12	-0.1907	-0.2565
13	-0.4225	-0.0561
14	-0.1012	-0.0202
15	0.234	0.028

Plotting Landmarks:

points show reference and scatter about each reference LM. GPA minimizes the total spread of "data" points Fd around the reference Fr.



Calculating Matrix P_K (Bookstein 1991, p. 27, 32 & 294):

$i := 0 .. \text{rows}(Fr) - 1 \quad j := 0 .. \text{rows}(Fr) - 1$ **< index variables**

$x_i := \left(Fr^{\langle 0\rangle}\right)_i \qquad y_i := \left(Fr^{\langle 1\rangle}\right)_i$ **< obtaining x & y coordinates from the reference form Br**

$r_{i,j} := \sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2}$ **< $r_{i,j}$ are the Euclidean distances between LM ponts i & j in reference form Br**

$$r = \begin{pmatrix} 0 & 0.458 & 1.038 & 0.567 & 0.423 & 0.194 & 0.314 & 0.613 \\ 0.458 & 0 & 0.581 & 0.213 & 0.446 & 0.53 & 0.233 & 0.168 \\ 1.038 & 0.581 & 0 & 0.558 & 0.923 & 1.097 & 0.782 & 0.443 \\ 0.567 & 0.213 & 0.558 & 0 & 0.365 & 0.568 & 0.259 & 0.147 \\ 0.423 & 0.446 & 0.923 & 0.365 & 0 & 0.302 & 0.232 & 0.497 \\ 0.194 & 0.53 & 1.097 & 0.568 & 0.302 & 0 & 0.317 & 0.655 \\ 0.314 & 0.233 & 0.782 & 0.259 & 0.232 & 0.317 & 0 & 0.338 \\ 0.613 & 0.168 & 0.443 & 0.147 & 0.497 & 0.655 & 0.338 & 0 \end{pmatrix}$$

$$P_{K_{i,j}} := \left| \begin{array}{ll} \left(r_{i,j}\right)^2 \cdot \ln\left[\left(r_{i,j}\right)^2\right] & \text{if } i \neq j \\ 0 & \text{otherwise} \end{array} \right.$$

< spline matrix P_K. Note that since ln(0) is undefined as a function, P_K has to be explicitly programmed here using a conditional ...

$$P_K = \begin{pmatrix} 0 & -0.328 & 0.081 & -0.365 & -0.308 & -0.123 & -0.229 & -0.368 \\ -0.328 & 0 & -0.367 & -0.141 & -0.321 & -0.357 & -0.158 & -0.101 \\ 0.081 & -0.367 & 0 & -0.363 & -0.137 & 0.223 & -0.301 & -0.32 \\ -0.365 & -0.141 & -0.363 & 0 & -0.269 & -0.365 & -0.181 & -0.083 \\ -0.308 & -0.321 & -0.137 & -0.269 & 0 & -0.218 & -0.157 & -0.345 \\ -0.123 & -0.357 & 0.223 & -0.365 & -0.218 & 0 & -0.231 & -0.363 \\ -0.229 & -0.158 & -0.301 & -0.181 & -0.157 & -0.231 & 0 & -0.248 \\ -0.368 & -0.101 & -0.32 & -0.083 & -0.345 & -0.363 & -0.248 & 0 \end{pmatrix}$$

^ Matrix P_K is an explicit calculation of "bending energy" between LM points in the reference form Fr.

Matrix Q of 1's and x,y Coordinates of Landmark Points (Bookstein 1991, p. 32 & 320):

$ONE_i := 1$ $\quad k := 2$ $\quad ZERO_{k,2} := 0$

$Q := \text{augment}(ONE, Fr)$

< In addition to "bending energy" P_K, a thin plate spline requires information about the (x,y) location of the "armature" points. This information is supplied by the LM points in the reference form Fr.

Partioned Matrix L containing both affine and non-affine elements (Bookstein 1991, p. 32 & 320):

$L_{TOP} := \text{augment}\left(P_K, Q\right)$ $\quad L_{BOT} := \text{augment}\left(Q^T, ZERO\right)$ $\quad L := \text{stack}\left(L_{TOP}, L_{BOT}\right)$

$$L = \begin{pmatrix} 0 & -0.328 & 0.081 & -0.365 & -0.308 & -0.123 & -0.229 & -0.368 & 1 & -0.379 & 0.138 \\ -0.328 & 0 & -0.367 & -0.141 & -0.321 & -0.357 & -0.158 & -0.101 & 1 & 0.078 & 0.115 \\ 0.081 & -0.367 & 0 & -0.363 & -0.137 & 0.223 & -0.301 & -0.32 & 1 & 0.659 & 0.119 \\ -0.365 & -0.141 & -0.363 & 0 & -0.269 & -0.365 & -0.181 & -0.083 & 1 & 0.141 & -0.089 \\ -0.308 & -0.321 & -0.137 & -0.269 & 0 & -0.218 & -0.157 & -0.345 & 1 & -0.191 & -0.241 \\ -0.123 & -0.357 & 0.223 & -0.365 & -0.218 & 0 & -0.231 & -0.363 & 1 & -0.425 & -0.051 \\ -0.229 & -0.158 & -0.301 & -0.181 & -0.157 & -0.231 & 0 & -0.248 & 1 & -0.109 & -0.024 \\ -0.368 & -0.101 & -0.32 & -0.083 & -0.345 & -0.363 & -0.248 & 0 & 1 & 0.225 & 0.032 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ -0.379 & 0.078 & 0.659 & 0.141 & -0.191 & -0.425 & -0.109 & 0.225 & 0 & 0 & 0 \\ 0.138 & 0.115 & 0.119 & -0.089 & -0.241 & -0.051 & -0.024 & 0.032 & 0 & 0 & 0 \end{pmatrix}$$

< Matrix L provides information about *potential* (i.e., before knowledge about any specific deformation) "bending energy" and LM locations in the reference form Br.

"Charging the Spline" with Vectors V & Y containing LM locations in the second (data) form (Bookstein 1991, p. 33 & 320):

$n := 0 .. OBJnum - 1$ $\quad J_n := n \cdot LMnum$

$Vx_{i,n} := \left(Fd^{\langle 0 \rangle}\right)_{(J_n)+i}$

$Vy_{i,n} := \left(Fd^{\langle 1 \rangle}\right)_{(J_n)+i}$ $\qquad Z := \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$

$Yx^{\langle n \rangle} := \text{stack}\left(Vx^{\langle n \rangle}, Z\right)$

$Yy^{\langle n \rangle} := \text{stack}\left(Vy^{\langle n \rangle}, Z\right)$

< Since the thin-plate spline models warping of a thin metal plate in the z direction given (x,y) information about position on the plate, Bookstein's procedure considers each (V_x, V_y) separately and stacks them with vector Z to make vectors (Yx, Yy).

Yx =

	0	1	2	3	4
0	-0.3954	-0.3766	-0.3679	-0.3467	-0.365
1	0.0486	0.0308	0.111	0.0964	0.0848
2	0.6417	0.6505	0.6394	0.6607	0.6849
3	0.1241	0.1757	0.1362	0.1355	0.1117
4	-0.1919	-0.1907	-0.237	-0.2346	-0.1769
5	-0.4265	-0.4225	-0.434	-0.4325	-0.4174
6	-0.068	-0.1012	-0.097	-0.1113	-0.1318
7	0.2674	0.234	0.2494	0.2324	0.2097
8	0	0	0	0	0
9	0	0	0	0	0
10	0	0	0	0	0

Yy =

	0	1	2	3	4
0	0.1466	0.1498	0.1135	0.1271	0.1306
1	0.1079	0.1357	0.1142	0.0898	0.1066
2	0.1253	0.1023	0.1246	0.137	0.1302
3	-0.1189	-0.0829	-0.0836	-0.0593	-0.0954
4	-0.243	-0.2565	-0.2479	-0.2492	-0.2375
5	-0.0453	-0.0561	-0.0518	-0.0478	-0.0445
6	-0.0124	-0.0202	-0.0092	-0.028	-0.0214
7	0.04	0.028	0.0401	0.0303	0.0313
8	0	0	0	0	0
9	0	0	0	0	0
10	0	0	0	0	0

Calculating the thin-plate spline (Bookstein 1991, p. 33 & 321):

$$Wx := L^{-1} \cdot Yx \qquad Wy := L^{-1} \cdot Yy$$

Wx =

	0	1	2	3	4
0	-0.0676	0.1042	-0.0823	0.0337	0.0402
1	-0.6838	-0.3956	-0.0705	-0.0406	0.1956
2	-0.0728	-0.0173	-0.0371	-0.01	0.0236
3	-0.6188	0.107	-0.0824	0.061	-0.0767
4	-0.1081	-0.1642	-0.0717	-0.0226	0.1785
5	-0.01	-0.0338	0.0607	-0.0204	-0.0546
6	0.635	0.2056	0.1229	-0.0036	-0.2244
7	0.9262	0.1941	0.1604	0.0025	-0.0823
8	-0.0071	-0.0043	-0.0012	0.0012	0.0065
9	0.996	1.0063	0.9669	0.9786	0.9952
10	-0.0253	-0.1195	0.1602	0.1699	0.0473

Wy =

	0	1	2	3	4
0	-0.019	0	-0.11	0.011	-0.037
1	-0.216	0.128	-0.076	-0.092	-0.052
2	-0.015	0.003	-0.004	0.015	0.005
3	-0.321	0.149	-0.054	0.237	-0.073
4	0.031	-0.011	-0.11	-0.152	-0.021
5	-0.029	-0.002	0.102	0.069	0.035
6	0.207	-0.071	0.179	0.034	0.072
7	0.362	-0.196	0.072	-0.122	0.071
8	-0.002	0.001	-0.007	-0.006	-0.003
9	-0.007	-0.015	0.027	0.032	0.012
10	1.058	1.07	0.967	0.91	0.978

^ weights for the thin-plate spline. Each column represents an OBJ in Fd. Last three rows represent affine coefficients, the remaining rows are the non-affine coefficients.

Principal Warps & Partial Warps:

Bookstein (1991) describes (and Rohlf impliments in TPS) an further spectral decomposition of the non-affine part of the thin-plate spline very much in the spirit of PCA. The mathematics, involving eigenvectors and eigenvalues, is nearly identical to PCA although based on the very different matrix L_K^{-1} described below.

Now Bookstein defines the sub-block matrix L_K^{-1}:

$K := LMnum \qquad invL_K := submatrix\left(L^{-1}, 0, K-1, 0, K-1\right)$

$$invL_K = \begin{pmatrix} 3.69008 & -1.69909 & -0.04093 & 0.91724 & 1.89597 & -3.63019 & -1.50591 & 0.37283 \\ -1.69909 & 7.60142 & 0.24904 & 1.84287 & 1.46112 & 1.10091 & -4.35085 & -6.20542 \\ -0.04093 & 0.24904 & 0.4983 & 0.29288 & -0.00426 & 0.21403 & 0.12966 & -1.33871 \\ 0.91724 & 1.84287 & 0.29288 & 8.15978 & -2.30126 & 0.62735 & -2.4749 & -7.06397 \\ 1.89597 & 1.46112 & -0.00426 & -2.30126 & 3.54471 & -2.79226 & -2.47739 & 0.67337 \\ -3.63019 & 1.10091 & 0.21403 & 0.62735 & -2.79226 & 4.8875 & 0.16886 & -0.57622 \\ -1.50591 & -4.35085 & 0.12966 & -2.4749 & -2.47739 & 0.16886 & 8.16701 & 2.34353 \\ 0.37283 & -6.20542 & -1.33871 & -7.06397 & 0.67337 & -0.57622 & 2.34353 & 11.79459 \end{pmatrix}$$

^ (K X K) upper left sub-block of L^{-1}

Eigenanalysis of L_K^{-1}:

$\lambda := reverse\left(sort\left(eigenvals\left(invL_K\right)\right)\right)$

$k := 0 .. K-4$

$\lambda\lambda_k := \lambda_k$

$$\lambda\lambda = \begin{pmatrix} 22.1509 \\ 11.4901 \\ 8.4579 \\ 5.2279 \\ 1.0165 \end{pmatrix} \qquad \lambda = \begin{pmatrix} 22.1509 \\ 11.4901 \\ 8.4579 \\ 5.2279 \\ 1.0165 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

< Eigenvalues sorted from largest to smallest

< there are always K-3 non-zero eigenvectors for L_K^{-1}

$\varepsilon^{\langle k \rangle} := eigenvec\left(invL_K, \lambda\lambda_k\right)$

$$\varepsilon = \begin{pmatrix} 0.0214 & -0.394 & 0.4687 & -0.051 & 0.1573 \\ -0.4541 & -0.1502 & -0.51 & 0.2661 & -0.4485 \\ -0.0522 & 0.0375 & 0.0224 & 0.1109 & 0.5895 \\ -0.4658 & 0.1883 & 0.4879 & -0.3995 & -0.3126 \\ 0.0155 & -0.5165 & -0.1031 & 0.2597 & 0.1754 \\ -0.0729 & 0.4564 & -0.4257 & -0.3658 & 0.3769 \\ 0.3308 & 0.5347 & 0.2399 & 0.6081 & -0.1941 \\ 0.6772 & -0.1561 & -0.1801 & -0.4286 & -0.3439 \end{pmatrix}$$

< columns are eigenvectors standardized to length 1 and corresponding in order to each eigenvalue in λλ.

Bookstein calls these eigenvectors "Principal Warps"

Partial Warps:

Partial warps are calculated using the spectral decomposition of L_K^{-1}:

$$invL_K = \begin{pmatrix} 3.69008 & -1.69909 & -0.04093 & 0.91724 & 1.89597 & -3.63019 & -1.50591 & 0.37283 \\ -1.69909 & 7.60142 & 0.24904 & 1.84287 & 1.46112 & 1.10091 & -4.35085 & -6.20542 \\ -0.04093 & 0.24904 & 0.4983 & 0.29288 & -0.00426 & 0.21403 & 0.12966 & -1.33871 \\ 0.91724 & 1.84287 & 0.29288 & 8.15978 & -2.30126 & 0.62735 & -2.4749 & -7.06397 \\ 1.89597 & 1.46112 & -0.00426 & -2.30126 & 3.54471 & -2.79226 & -2.47739 & 0.67337 \\ -3.63019 & 1.10091 & 0.21403 & 0.62735 & -2.79226 & 4.8875 & 0.16886 & -0.57622 \\ -1.50591 & -4.35085 & 0.12966 & -2.4749 & -2.47739 & 0.16886 & 8.16701 & 2.34353 \\ 0.37283 & -6.20542 & -1.33871 & -7.06397 & 0.67337 & -0.57622 & 2.34353 & 11.79459 \end{pmatrix} \qquad \lambda\lambda = \begin{pmatrix} 22.1509 \\ 11.4901 \\ 8.4579 \\ 5.2279 \\ 1.0165 \end{pmatrix}$$

Spectral Decomposition:

$$P_k := \lambda\lambda_k \cdot \varepsilon^{\langle k \rangle} \cdot \varepsilon^{\langle k \rangle^T}$$

for $\lambda\lambda_0$:

$$P_0 = \begin{pmatrix} 0.0101 & -0.2152 & -0.0247 & -0.2208 & 0.0074 & -0.0346 & 0.1568 & 0.321 \\ -0.2152 & 4.568 & 0.5249 & 4.6853 & -0.1564 & 0.7335 & -3.3275 & -6.8125 \\ -0.0247 & 0.5249 & 0.0603 & 0.5384 & -0.018 & 0.0843 & -0.3823 & -0.7828 \\ -0.2208 & 4.6853 & 0.5384 & 4.8057 & -0.1604 & 0.7523 & -3.413 & -6.9874 \\ 0.0074 & -0.1564 & -0.018 & -0.1604 & 0.0054 & -0.0251 & 0.1139 & 0.2332 \\ -0.0346 & 0.7335 & 0.0843 & 0.7523 & -0.0251 & 0.1178 & -0.5343 & -1.0939 \\ 0.1568 & -3.3275 & -0.3823 & -3.413 & 0.1139 & -0.5343 & 2.4239 & 4.9625 \\ 0.321 & -6.8125 & -0.7828 & -6.9874 & 0.2332 & -1.0939 & 4.9625 & 10.1598 \end{pmatrix}$$

for $\lambda\lambda_1$:

$$P_1 = \begin{pmatrix} 1.7835 & 0.6801 & -0.1697 & -0.8523 & 2.3383 & -2.0661 & -2.4204 & 0.7066 \\ 0.6801 & 0.2594 & -0.0647 & -0.325 & 0.8917 & -0.7879 & -0.923 & 0.2695 \\ -0.1697 & -0.0647 & 0.0161 & 0.0811 & -0.2225 & 0.1966 & 0.2303 & -0.0672 \\ -0.8523 & -0.325 & 0.0811 & 0.4073 & -1.1174 & 0.9873 & 1.1567 & -0.3377 \\ 2.3383 & 0.8917 & -0.2225 & -1.1174 & 3.0656 & -2.7088 & -3.1733 & 0.9264 \\ -2.0661 & -0.7879 & 0.1966 & 0.9873 & -2.7088 & 2.3935 & 2.8039 & -0.8185 \\ -2.4204 & -0.923 & 0.2303 & 1.1567 & -3.1733 & 2.8039 & 3.2848 & -0.9589 \\ 0.7066 & 0.2695 & -0.0672 & -0.3377 & 0.9264 & -0.8185 & -0.9589 & 0.2799 \end{pmatrix}$$

for $\lambda\lambda_2$:

$$P_2 = \begin{pmatrix} 1.8577 & -2.0214 & 0.0888 & 1.9338 & -0.4085 & -1.6873 & 0.9508 & -0.714 \\ -2.0214 & 2.1995 & -0.0966 & -2.1042 & 0.4445 & 1.836 & -1.0346 & 0.7769 \\ 0.0888 & -0.0966 & 0.0042 & 0.0924 & -0.0195 & -0.0806 & 0.0454 & -0.0341 \\ 1.9338 & -2.1042 & 0.0924 & 2.0131 & -0.4253 & -1.7564 & 0.9898 & -0.7432 \\ -0.4085 & 0.4445 & -0.0195 & -0.4253 & 0.0898 & 0.371 & -0.2091 & 0.157 \\ -1.6873 & 1.836 & -0.0806 & -1.7564 & 0.371 & 1.5325 & -0.8636 & 0.6485 \\ 0.9508 & -1.0346 & 0.0454 & 0.9898 & -0.2091 & -0.8636 & 0.4867 & -0.3654 \\ -0.714 & 0.7769 & -0.0341 & -0.7432 & 0.157 & 0.6485 & -0.3654 & 0.2744 \end{pmatrix}$$

for $\lambda\lambda_3$:

$$P_3 = \begin{pmatrix} 0.0136 & -0.0709 & -0.0296 & 0.1065 & -0.0692 & 0.0975 & -0.1621 & 0.1142 \\ -0.0709 & 0.3701 & 0.1543 & -0.5557 & 0.3613 & -0.5088 & 0.8459 & -0.5961 \\ -0.0296 & 0.1543 & 0.0643 & -0.2316 & 0.1506 & -0.2121 & 0.3526 & -0.2485 \\ 0.1065 & -0.5557 & -0.2316 & 0.8344 & -0.5424 & 0.7639 & -1.2701 & 0.8951 \\ -0.0692 & 0.3613 & 0.1506 & -0.5424 & 0.3527 & -0.4966 & 0.8257 & -0.5819 \\ 0.0975 & -0.5088 & -0.2121 & 0.7639 & -0.4966 & 0.6994 & -1.1628 & 0.8195 \\ -0.1621 & 0.8459 & 0.3526 & -1.2701 & 0.8257 & -1.1628 & 1.9333 & -1.3625 \\ 0.1142 & -0.5961 & -0.2485 & 0.8951 & -0.5819 & 0.8195 & -1.3625 & 0.9602 \end{pmatrix}$$

for $\lambda\lambda_4$:

$$P_4 = \begin{pmatrix} 0.0251 & -0.0717 & 0.0943 & -0.05 & 0.028 & 0.0603 & -0.031 & -0.055 \\ -0.0717 & 0.2045 & -0.2688 & 0.1425 & -0.08 & -0.1718 & 0.0885 & 0.1568 \\ 0.0943 & -0.2688 & 0.3533 & -0.1873 & 0.1051 & 0.2259 & -0.1163 & -0.2061 \\ -0.05 & 0.1425 & -0.1873 & 0.0993 & -0.0557 & -0.1198 & 0.0617 & 0.1093 \\ 0.028 & -0.08 & 0.1051 & -0.0557 & 0.0313 & 0.0672 & -0.0346 & -0.0613 \\ 0.0603 & -0.1718 & 0.2259 & -0.1198 & 0.0672 & 0.1444 & -0.0744 & -0.1318 \\ -0.031 & 0.0885 & -0.1163 & 0.0617 & -0.0346 & -0.0744 & 0.0383 & 0.0679 \\ -0.055 & 0.1568 & -0.2061 & 0.1093 & -0.0613 & -0.1318 & 0.0679 & 0.1203 \end{pmatrix}$$

$$\sum_k P_k = \begin{pmatrix} 3.6901 & -1.6991 & -0.0409 & 0.9172 & 1.896 & -3.6302 & -1.5059 & 0.3728 \\ -1.6991 & 7.6014 & 0.249 & 1.8429 & 1.4611 & 1.1009 & -4.3509 & -6.2054 \\ -0.0409 & 0.249 & 0.4983 & 0.2929 & -0.0043 & 0.214 & 0.1297 & -1.3387 \\ 0.9172 & 1.8429 & 0.2929 & 8.1598 & -2.3013 & 0.6274 & -2.4749 & -7.064 \\ 1.896 & 1.4611 & -0.0043 & -2.3013 & 3.5447 & -2.7923 & -2.4774 & 0.6734 \\ -3.6302 & 1.1009 & 0.214 & 0.6274 & -2.7923 & 4.8875 & 0.1689 & -0.5762 \\ -1.5059 & -4.3509 & 0.1297 & -2.4749 & -2.4774 & 0.1689 & 8.167 & 2.3435 \\ 0.3728 & -6.2054 & -1.3387 & -7.064 & 0.6734 & -0.5762 & 2.3435 & 11.7946 \end{pmatrix}$$

Partials sum to original L_K^{-1} matrix >

$$invL_K = \begin{pmatrix} 3.6901 & -1.6991 & -0.0409 & 0.9172 & 1.896 & -3.6302 & -1.5059 & 0.3728 \\ -1.6991 & 7.6014 & 0.249 & 1.8429 & 1.4611 & 1.1009 & -4.3509 & -6.2054 \\ -0.0409 & 0.249 & 0.4983 & 0.2929 & -0.0043 & 0.214 & 0.1297 & -1.3387 \\ 0.9172 & 1.8429 & 0.2929 & 8.1598 & -2.3013 & 0.6274 & -2.4749 & -7.064 \\ 1.896 & 1.4611 & -0.0043 & -2.3013 & 3.5447 & -2.7923 & -2.4774 & 0.6734 \\ -3.6302 & 1.1009 & 0.214 & 0.6274 & -2.7923 & 4.8875 & 0.1689 & -0.5762 \\ -1.5059 & -4.3509 & 0.1297 & -2.4749 & -2.4774 & 0.1689 & 8.167 & 2.3435 \\ 0.3728 & -6.2054 & -1.3387 & -7.064 & 0.6734 & -0.5762 & 2.3435 & 11.7946 \end{pmatrix}$$

And Partials multiplied by LM "data" points are the same as the non-affine weights:

$\sum_k P_k \cdot Vx =$

	0	1	2	3	4	5	6	7	8
0	-0.0676	0.1042	-0.0823	0.0337	0.0402	-0.0464	-0.1943	0.0848	0.1795
1	-0.6838	-0.3956	-0.0705	-0.0406	0.1956	-0.8546	0.3903	-0.6216	0.8398
2	-0.0728	-0.0173	-0.0371	-0.01	0.0236	-0.088	0.0241	-0.0209	0.0863
3	-0.6188	0.107	-0.0824	0.061	-0.0767	-0.9158	0.083	-0.1383	0.7543
4	-0.1081	-0.1642	-0.0717	-0.0226	0.1785	-0.0302	-0.0077	-0.1765	0.218
5	-0.01	-0.0338	0.0607	-0.0204	-0.0546	-0.082	0.2045	-0.0561	-0.1072
6	0.635	0.2056	0.1229	-0.0036	-0.2244	0.7688	-0.1973	0.4551	-0.8304
7	0.9262	0.1941	0.1604	0.0025	-0.0823	1.2481	-0.3027	0.4737	-1.1402

$Wx =$

	0	1	2	3	4	5	6	7	8
0	-0.0676	0.1042	-0.0823	0.0337	0.0402	-0.0464	-0.1943	0.0848	0.1795
1	-0.6838	-0.3956	-0.0705	-0.0406	0.1956	-0.8546	0.3903	-0.6216	0.8398
2	-0.0728	-0.0173	-0.0371	-0.01	0.0236	-0.088	0.0241	-0.0209	0.0863
3	-0.6188	0.107	-0.0824	0.061	-0.0767	-0.9158	0.083	-0.1383	0.7543
4	-0.1081	-0.1642	-0.0717	-0.0226	0.1785	-0.0302	-0.0077	-0.1765	0.218
5	-0.01	-0.0338	0.0607	-0.0204	-0.0546	-0.082	0.2045	-0.0561	-0.1072
6	0.635	0.2056	0.1229	-0.0036	-0.2244	0.7688	-0.1973	0.4551	-0.8304
7	0.9262	0.1941	0.1604	0.0025	-0.0823	1.2481	-0.3027	0.4737	-1.1402
8	-0.0071	-0.0043	-0.0012	0.0012	0.0065	-0.0055	-0.0028	-0.0089	0.0139
9	0.996	1.0063	0.9669	0.9786	0.9952	1.007	0.9889	1.0471	1.0115
10	-0.0253	-0.1195	0.1602	0.1699	0.0473	0.1028	-0.1607	-0.158	0.144

^ shown here for the Vx, but $\Sigma P_k V_y$ works the same way.

Projecting as in PCA:

$$PRJx := \varepsilon^{T} \cdot Vx$$

$$PRJy := \varepsilon^{T} \cdot Vy$$

$$\varepsilon^{\langle 0\rangle} = \begin{pmatrix} 0.0214 \\ -0.4541 \\ -0.0522 \\ -0.4658 \\ 0.0155 \\ -0.0729 \\ 0.3308 \\ 0.6772 \end{pmatrix} \quad Vx^{\langle 0\rangle} = \begin{pmatrix} -0.3954 \\ 0.0486 \\ 0.6417 \\ 0.1241 \\ -0.1919 \\ -0.4265 \\ -0.068 \\ 0.2674 \end{pmatrix}$$

$$\varepsilon^{\langle 0\rangle T} \cdot Vx^{\langle 0\rangle} = (0.0649)$$

^ for first projection

PRJx =

	0	1	2	3	4	5	6
0	0.0649	0.015	0.0097	-0.0003	-0.008	0.0868	-0.0229
1	0.0223	0.0163	0.0114	0.0003	-0.0246	0.0144	0.0064
2	0.0017	0.0412	-0.0073	0.009	-0.018	-0.0044	-0.0388
3	0.0049	-0.0275	-0.0039	-0.0076	0.0092	0.0164	0.0031
4	-0.0176	0.002	-0.0454	-0.0131	0.0385	-0.0042	0.0007

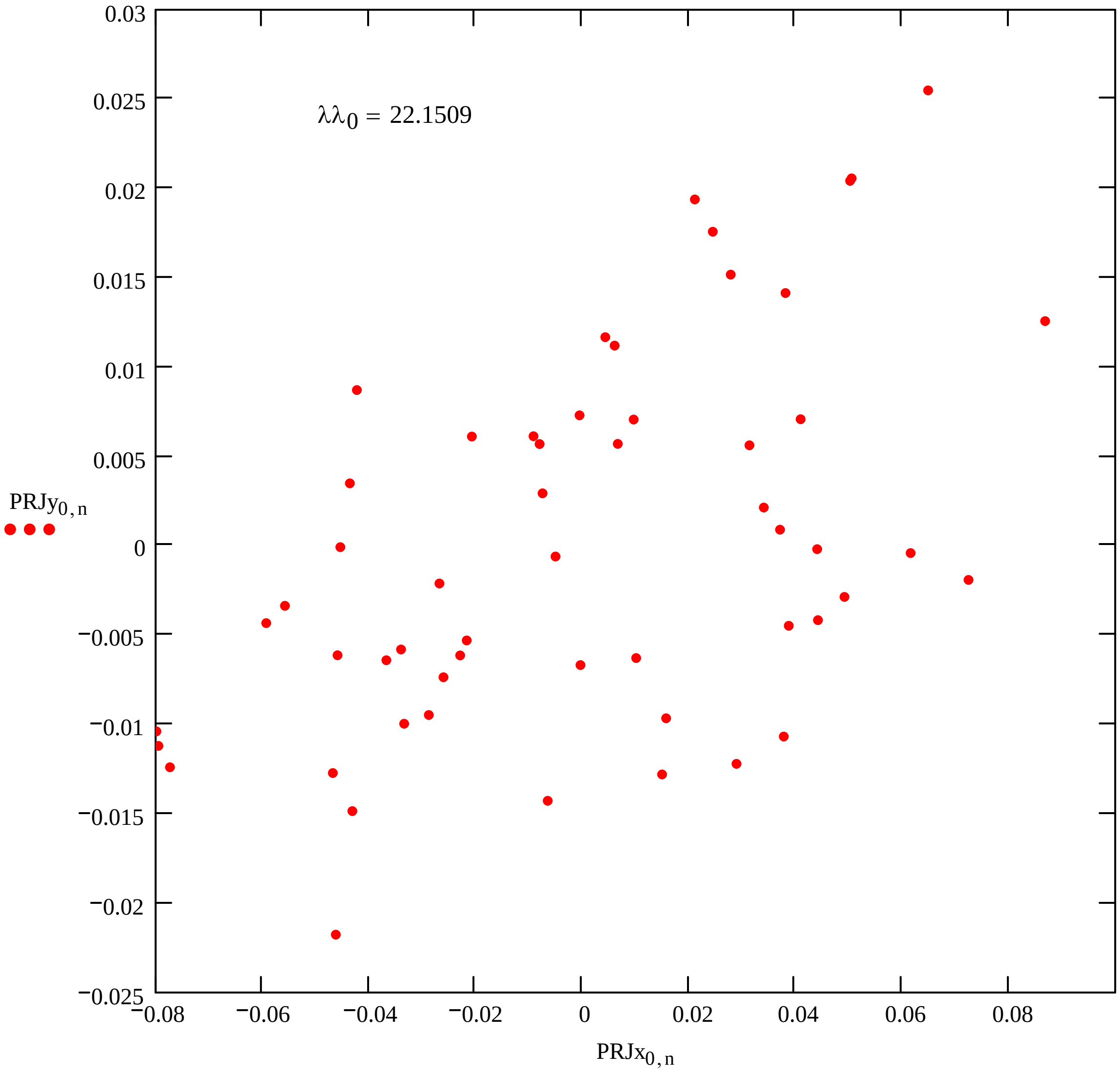
PRJy =

	0	1	2	3	4	5	6
0	0.0255	-0.0128	0.0071	-0.0067	0.0057	0.0126	-0.0061
1	0.0003	0.0005	0.0203	0.0176	0.0055	0.0089	-0.0192
2	-0.0073	0.0033	-0.0049	0.0218	-0.0041	0.0032	-0.0114
3	0.0114	0.0026	0.0035	-0.0209	0.0024	-0.0063	0.0243

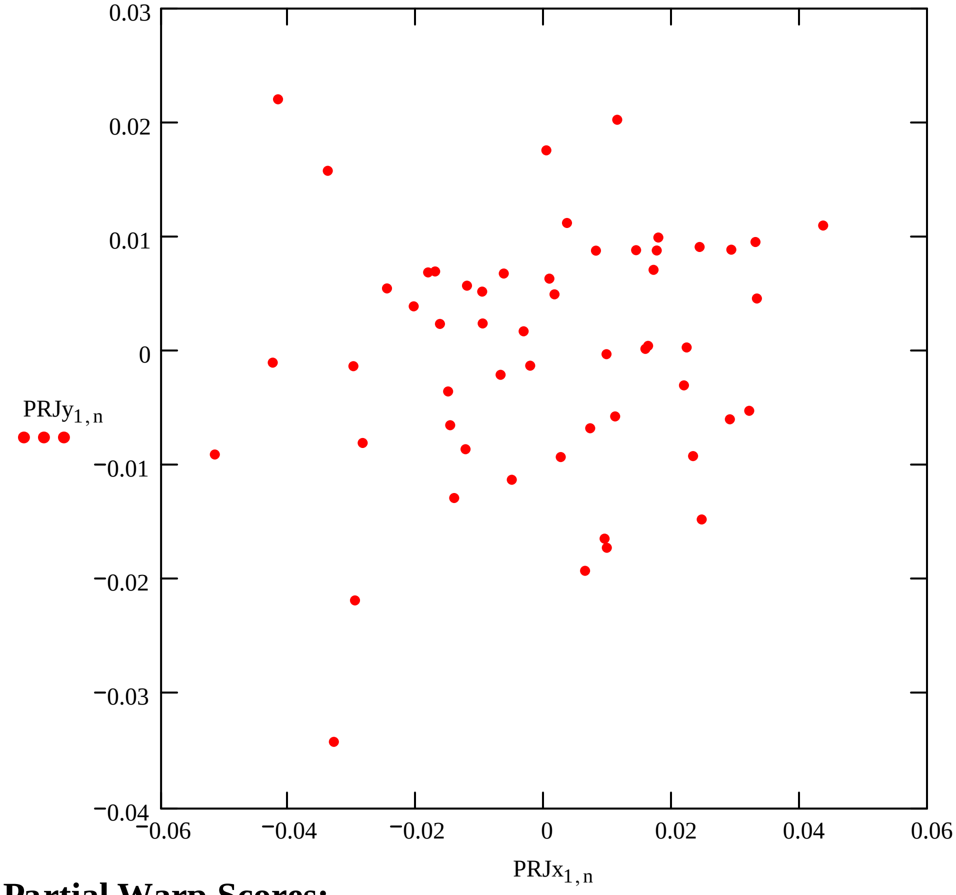
^ Projections of data LM points onto eigenvectors
Rows = K-3 eigenvectors
Columns = Objects

Separately for x and for y, the K-dimensional vectors in V_x or V_y for each OBJ are projected onto the K-3 eigenvectors of the PCA-like Partial Warp space... Points represent "scores" in Partial Warp space.

First Partial Warp Scores:

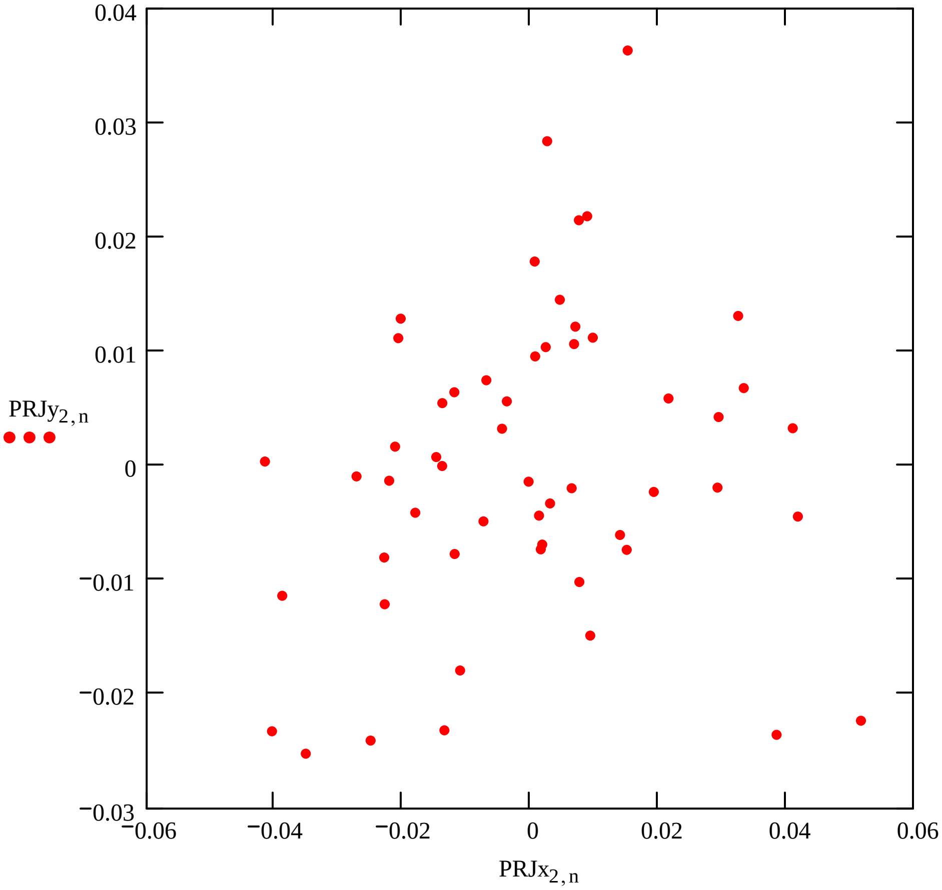


Second Partial Warp Scores:



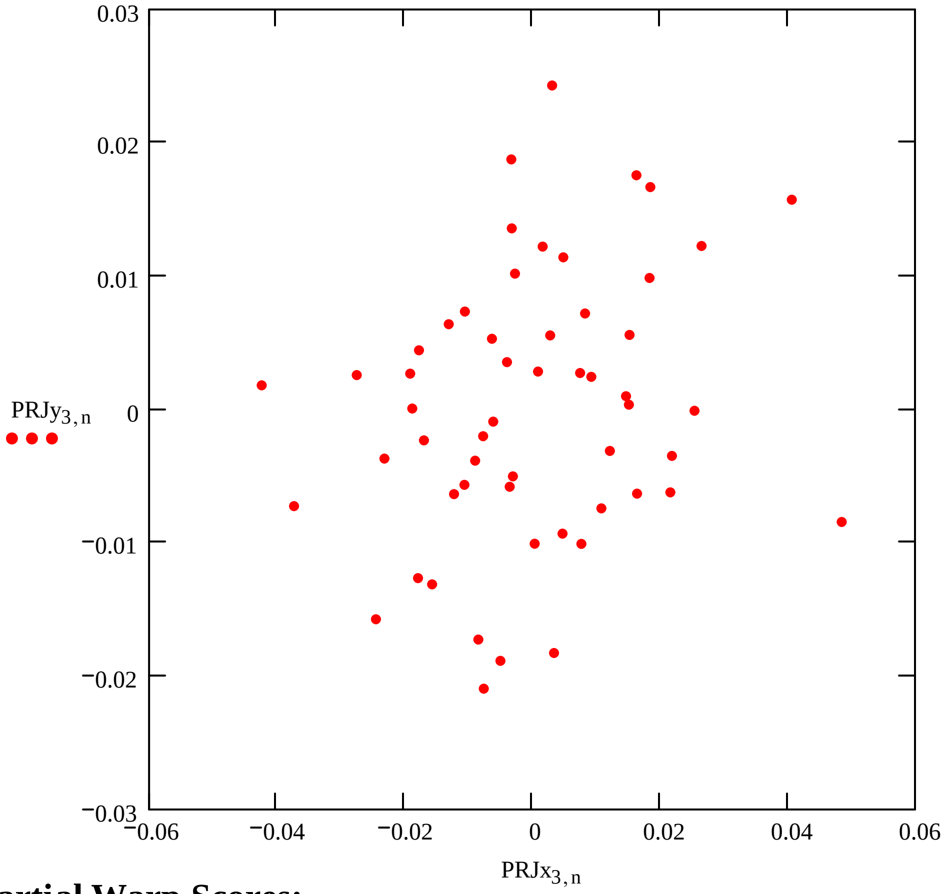
$\lambda\lambda_1 = 11.4901$

Third Partial Warp Scores:



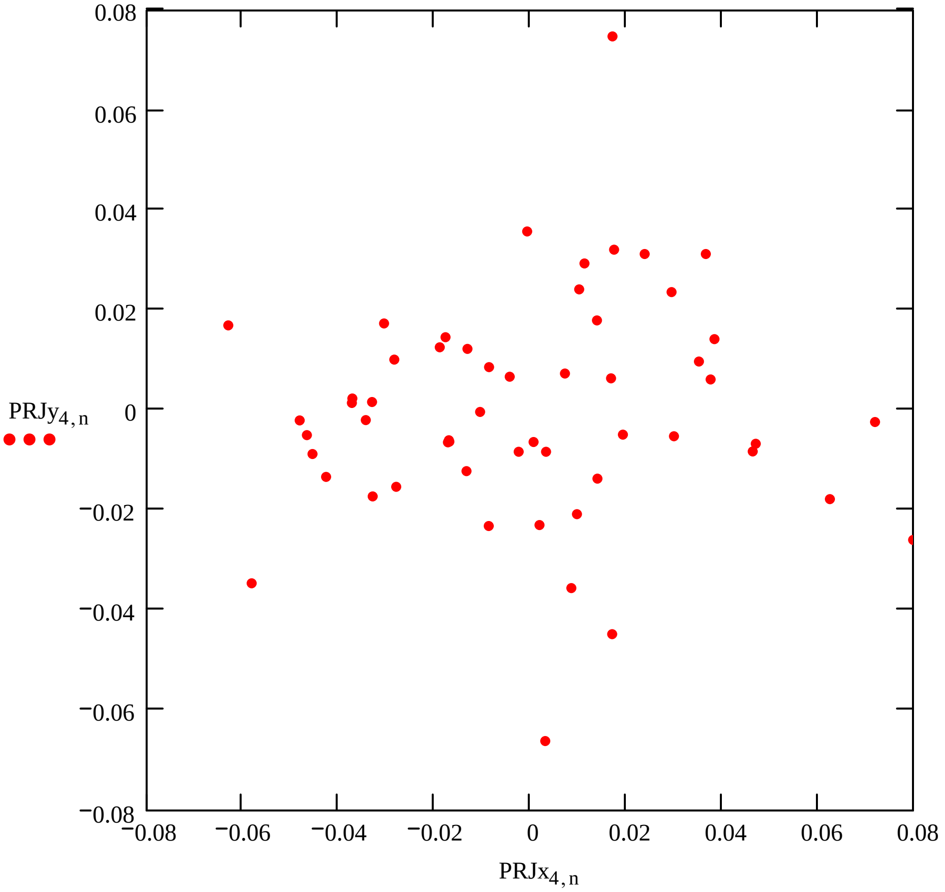
$\lambda\lambda_2 = 8.4579$

Fourth Partial Warp Scores:



$\lambda\lambda_3 = 5.2279$

Fifth Partial Warp Scores:



$\lambda\lambda_4 = 1.0165$

Relative Warp Analysis:

Relative warps are calculated by simple PCA of Rohlf's "weight matrix" which includes the above PCA partial warp "scores" (projections) along with projections of an affine components calculated as described in the Technical details section of his Help notes. Good luck interpreting it.

Assembling the Projections:

The non-affine Partial Warp projections, calculated above, reside in the following matrices for x & y separately:

PRJx =

	0	1	2	3	4	5	6	7	8
0	0.0649	0.015	0.0097	-0.0003	-0.008	0.0868	-0.0229	0.0371	-0.0799
1	0.0223	0.0163	0.0114	0.0003	-0.0246	0.0144	0.0064	0.0233	-0.0417
2	0.0017	0.0412	-0.0073	0.009	-0.018	-0.0044	-0.0388	0.0419	0.0065
3	0.0049	-0.0275	-0.0039	-0.0076	0.0092	0.0164	0.0031	-0.0131	0.0004
4	-0.0176	0.002	-0.0454	-0.0131	0.0385	-0.0042	0.0007	0.0194	0.0176

PRJy =

	0	1	2	3	4	5	6	7	8
0	0.0255	-0.0128	0.0071	-0.0067	0.0057	0.0126	-0.0061	0.0009	-0.0104
1	0.0003	0.0005	0.0203	0.0176	0.0055	0.0089	-0.0192	-0.0092	0.0221
2	-0.0073	0.0033	-0.0049	0.0218	-0.0041	0.0032	-0.0114	-0.0045	-0.002
3	0.0114	0.0026	0.0035	-0.0209	0.0024	-0.0063	0.0243	0.0064	-0.0101
4	0.0146	-0.0229	-0.0087	0.0123	0.0142	0.0067	-0.0063	-0.0048	0.0321

Each row represents the "score" (or "weight") for each Partial Warp. For K landmark points, there are K-3 Partial Warps.

To this, Bookstein & Rohlf append a pair of "scores" ("weights) for the affine component comparing each object with the Procrustes reference in Fr. At this point in time, I have been unable to de-gookify (as in gobbbly-gook) Rohlf's "Technical details" section in tpsRelw or Rohlf & Bookstein paper (2003 - Systematic Biology 52:66-69). In the latter reference, these authors give three ways to calculate the affine components, but no examples to allow verification by prototype. They are also needlessly elliptial about the relationship between the GPA consensus methodology and calculation forumlas they present - notably Eq. (2). I may get back to working on this somtime but, frankly, I've lost patience with them. Things need not be so opaque.

tpsRelw calculates the following affine "scores" for each object:

FU := READPRN("c:/2008Morphometrics/fossiluniformcomponent.dta")

FU^T =

	0	1	2	3	4	5	6	7	8
0	0.0161	0.0224	-0.0136	-0.0196	-0.0066	-0.0517	0.0524	-0.0254	-0.0627
1	-0.0154	-0.0269	0.0493	0.0582	0.0145	0.0184	-0.0498	-0.0465	0.0495

All this may now be assembled into a combined description of "scores" (or "weights") for each object. Total number of weights per object = 2(K-3)+2, given K LM points.

$$T := \text{stack}\left(\text{PRJx}, \text{stack}\left(\text{PRJy}, \text{FU}^T\right)\right)$$

^ this matrix corresponds to the "weight matrix" *.NTS file produced by tpsRelw. See also next page.

T =

	0	1	2	3	4	5	6	7	8
0	0.0649	0.015	0.0097	-0.0003	-0.008	0.0868	-0.0229	0.0371	-0.0799
1	0.0223	0.0163	0.0114	0.0003	-0.0246	0.0144	0.0064	0.0233	-0.0417
2	0.0017	0.0412	-0.0073	0.009	-0.018	-0.0044	-0.0388	0.0419	0.0065
3	0.0049	-0.0275	-0.0039	-0.0076	0.0092	0.0164	0.0031	-0.0131	0.0004
4	-0.0176	0.002	-0.0454	-0.0131	0.0385	-0.0042	0.0007	0.0194	0.0176
5	0.0255	-0.0128	0.0071	-0.0067	0.0057	0.0126	-0.0061	0.0009	-0.0104
6	0.0003	0.0005	0.0203	0.0176	0.0055	0.0089	-0.0192	-0.0092	0.0221
7	-0.0073	0.0033	-0.0049	0.0218	-0.0041	0.0032	-0.0114	-0.0045	-0.002
8	0.0114	0.0026	0.0035	-0.0209	0.0024	-0.0063	0.0243	0.0064	-0.0101
9	0.0146	-0.0229	-0.0087	0.0123	0.0142	0.0067	-0.0063	-0.0048	0.0321
10	0.0161	0.0224	-0.0136	-0.0196	-0.0066	-0.0517	0.0524	-0.0254	-0.0627
11	-0.0154	-0.0269	0.0493	0.0582	0.0145	0.0184	-0.0498	-0.0465	0.0495

Standard PCA analysis of the combined projections:

Forget all the authors' gobbly-gook, this part is a straight-forward PCA analysis of the "scores" matrix T above:

Sizing data array:

$n := \mathrm{cols}(T) \quad n = 55 \qquad p := \mathrm{rows}(T) \qquad p = 12 \qquad i := 0 .. n-1 \qquad j := 0 .. p-1$

$l_i := 1$

< vector of 1's

Variance/covariance (S):

$I := \mathrm{identity}(n)$

$$S := \frac{1}{n-1} \cdot T \cdot \left(I - \frac{1}{n} \cdot l \cdot l^T \right) \cdot T^T$$

< Note variance-covariance matrix is used here, NOT correlation.

$$S = \begin{pmatrix} 0.0017 & 0.0003 & 0.0002 & 0.0002 & 0.0002 & 0.0002 & -0 & -0.0002 & 0 & 0.0002 & -0 & -0.0005 \\ 0.0003 & 0.0005 & 0 & -0.0002 & -0.0002 & 0 & 0 & -0 & -0 & -0 & -0 & -0 \\ 0.0002 & 0 & 0.0005 & -0.0002 & 0.0002 & -0.0001 & 0 & 0.0001 & -0 & 0.0001 & -0.0001 & -0.0001 \\ 0.0002 & -0.0002 & -0.0002 & 0.0003 & 0.0001 & 0.0001 & -0 & -0.0001 & 0 & -0 & 0.0001 & -0 \\ 0.0002 & -0.0002 & 0.0002 & 0.0001 & 0.001 & -0.0001 & -0.0001 & -0 & 0.0001 & 0.0001 & 0.0001 & -0.0003 \\ 0.0002 & 0 & -0.0001 & 0.0001 & -0.0001 & 0.0001 & 0 & -0.0001 & 0 & 0.0001 & 0 & 0 \\ -0 & 0 & 0 & -0 & -0.0001 & 0 & 0.0001 & -0 & -0 & 0.0001 & -0.0002 & 0.0001 \\ -0.0002 & -0 & 0.0001 & -0.0001 & -0 & -0.0001 & -0 & 0.0002 & -0.0001 & -0 & -0 & -0 \\ 0 & -0 & -0 & 0 & 0.0001 & 0 & -0 & -0.0001 & 0.0001 & -0 & 0.0001 & -0 \\ 0.0002 & -0 & 0.0001 & -0 & 0.0001 & 0.0001 & 0.0001 & -0 & -0 & 0.0005 & -0.0001 & 0.0001 \\ -0 & -0 & -0.0001 & 0.0001 & 0.0001 & 0 & -0.0002 & -0 & 0.0001 & -0.0001 & 0.0015 & 0 \\ -0.0005 & -0 & -0.0001 & -0 & -0.0003 & 0 & 0.0001 & -0 & -0 & 0.0001 & 0 & 0.0011 \end{pmatrix}$$

Calculate eigenvalues & eigenvectors of variance/covariance matrix (S):

$\Lambda := \mathrm{reverse}(\mathrm{sort}(\mathrm{eigenvals}(S)))$

$E^{\langle j \rangle} := \mathrm{eigenvec}\left(S, \Lambda_j\right)$

NOTE:
Be certain that Eigenvalues (Λ) are in rank order and that Eigenvectors (E) are in columns in the same rank order. Otherwise calculations below will take into account the wrong entry in the variance / covariance matrix S.

$$\Lambda = \begin{pmatrix} 2.1513 \times 10^{-3} \\ 1.5537 \times 10^{-3} \\ 1.249 \times 10^{-3} \\ 9.4778 \times 10^{-4} \\ 6.4203 \times 10^{-4} \\ 4.2366 \times 10^{-4} \\ 2.5446 \times 10^{-4} \\ 1.8716 \times 10^{-4} \\ 1.0283 \times 10^{-4} \\ 6.6678 \times 10^{-5} \\ 2.8924 \times 10^{-5} \\ 2.6203 \times 10^{-5} \end{pmatrix}$$

E =

	0	1	2	3	4	5
0	0.82731	-0.12013	-0.33645	0.21305	-0.06293	0.09414
1	0.13781	-0.13174	-0.34486	-0.17426	0.37224	0.38683
2	0.11294	-0.08698	0.19969	0.20734	0.61021	-0.10779
3	0.08107	0.09176	-0.01911	0.06525	-0.5748	-0.11185
4	0.24415	0.25129	0.68283	0.40346	-0.03749	0.37386
5	0.08282	-0.01382	-0.15463	0.06015	-0.22228	-0.02918
6	-0.0288	-0.13708	-0.04094	0.05383	-0.00447	-0.0419
7	-0.07476	-0.00395	0.06764	-0.06216	0.22205	-0.06073
8	0.03345	0.10913	-0.00612	0.01184	-0.07919	0.06709
9	0.06699	-0.08903	-0.04111	0.43999	0.11429	-0.73263
10	0.02631	0.91899	-0.30132	0.01423	0.18563	-0.09071
11	-0.44503	-0.07553	-0.37239	0.71407	-0.02828	0.34904

Calculating Principal Component Scores (P) as linear combinations of T and E:

$P := T^{T} \cdot E$

< "loadings" or "scores" of original variables T on new principal components (eigenvectors) E...

P =

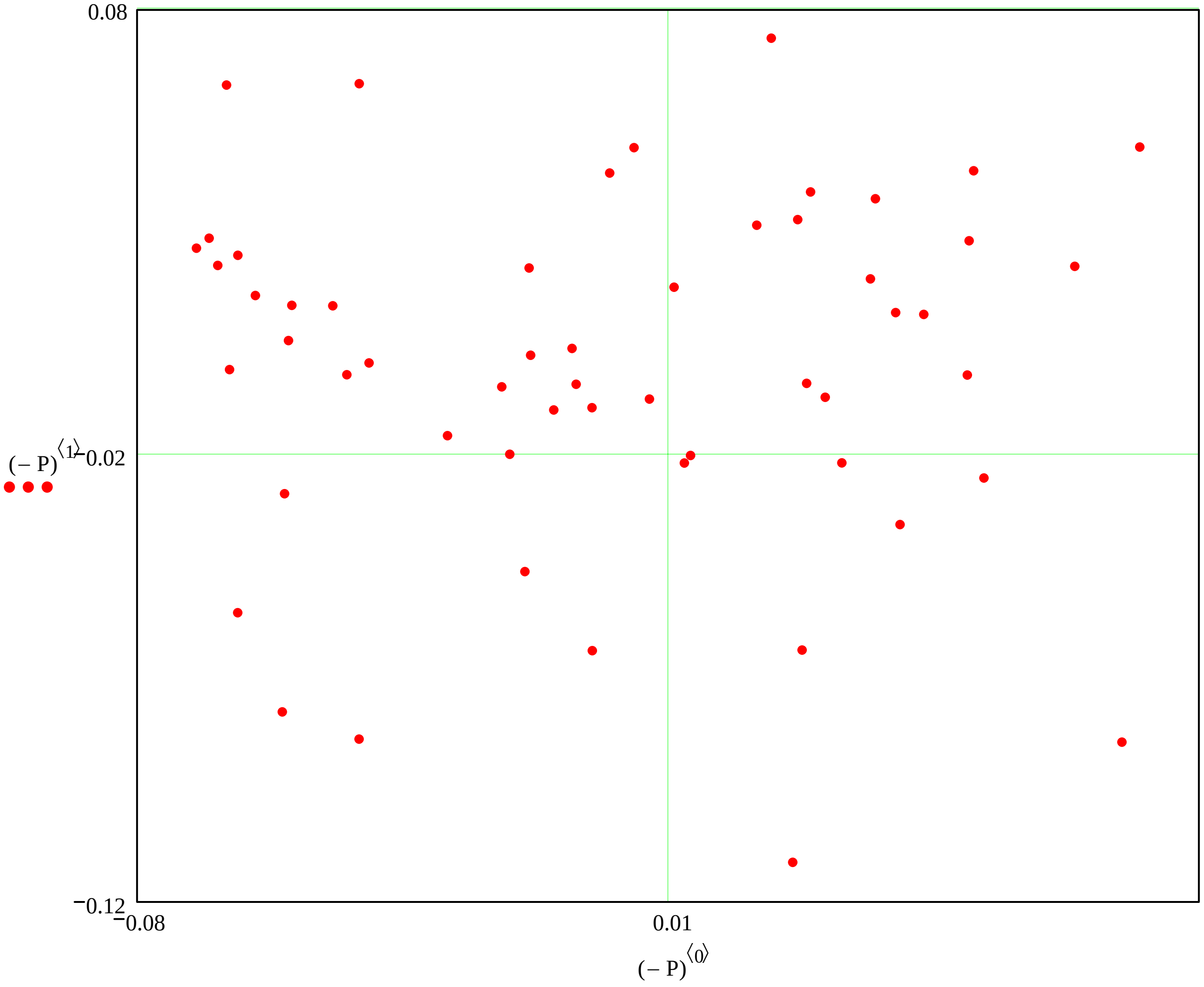
	0	1	2	3	4	5	6	7	8	9	10	11
0	0.0643	0.0006	-0.0455	0.0013	0	-0.0097	0.0026	-0.0079	-0.0038	0	0.008	0.0115
1	0.0274	0.0155	0.0058	-0.022	0.0516	0.0128	-0.0193	-0.0094	0.0039	-0.0081	-0.0013	0.0003
2	-0.025	-0.0317	-0.0558	0.013	-0.0048	0.0139	-0.0125	-0.0033	0.0055	-0.0128	0.0073	-0.0011
3	-0.032	-0.033	-0.0213	0.0416	0.0145	0.0049	-0.0094	0.0278	0.0091	-0.0046	0.0004	-0.002
4	-0.0068	0.0073	0.0283	0.0325	-0.0288	0.0003	0.0147	0.0047	-0.0014	-0.008	0.0038	0.0002
5	0.0648	-0.0631	-0.0319	0.0308	-0.023	0.0156	-0.0094	0.0189	-0.0011	0.0015	0.0046	-0.0021
6	0.0028	0.0637	0.002	-0.0515	-0.0142	-0.0098	0.0287	-0.0152	-0.0119	0.0055	0.0074	-0.013
7	0.0629	-0.025	0.0264	-0.0163	0.0331	0.0074	-0.0055	-0.0179	-0.0031	0.0096	-0.0007	0.0075
8	-0.09	-0.0492	0.0543	0.0478	-0.0142	-0.0196	0.0072	-0.0046	0.011	-0.006	-0.0011	-0.0026
9	-0.0451	-0.0376	0.0131	-0.0331	0.0038	0.0061	0.0194	-0.0253	0.0181	0.0281	0.0055	0.0028
10	-0.0275	-0.0737	0.0288	0.0107	-0.0082	0.009	0.0154	0.0095	-0.015	0.0004	-0.0028	0.0079
11	0.0424	0.0835	-0.0677	0.0448	-0.0121	-0.0002	0.0269	-0.0098	0.0041	0.0013	0.0019	-0.0017
12	-0.061	-0.0282	-0.0294	-0.0534	-0.0295	0.0094	0.0005	0.006	0.0071	-0.0015	-0.0034	-0.0016
13	-0.0327	0.0635	-0.0429	0.0094	-0.0188	-0.0208	-0.0251	0.0148	-0.0299	0.0087	-0.0037	-0
14	0.0663	-0.0227	0.0098	0.0202	0.011	0.0439	0.0198	0.0079	-0.0091	-0.0021	-0.0025	-0.0051
15	-0.0342	-0.0392	-0.0035	-0.0226	-0.0171	-0.0189	-0.0024	0.0015	-0.0016	0.0012	-0.0051	-0.0029
16	0.0599	-0.016	0.0401	0.0202	-0.0133	0.0142	-0.0189	0.0042	0.004	-0.0021	-0.0001	-0.0027

^ Each row represents Objects, columns are the Relative Warp "scores" (PCA projections).

Note that similar PCA results can be acheived, as Rohlf suggests, by Singular Value Decomposition of matrix T. Although perhaps more direct (since one need not calculate the variance/covariance matrix), this alternative approach offers nothing else. SVD is a useful multivariate eigen technique, but is not necessary for understanding thin-plate splines or PCA. As expected, for K landmarks, there are 2(K-3)+2 "Relative Warps"

Plots:

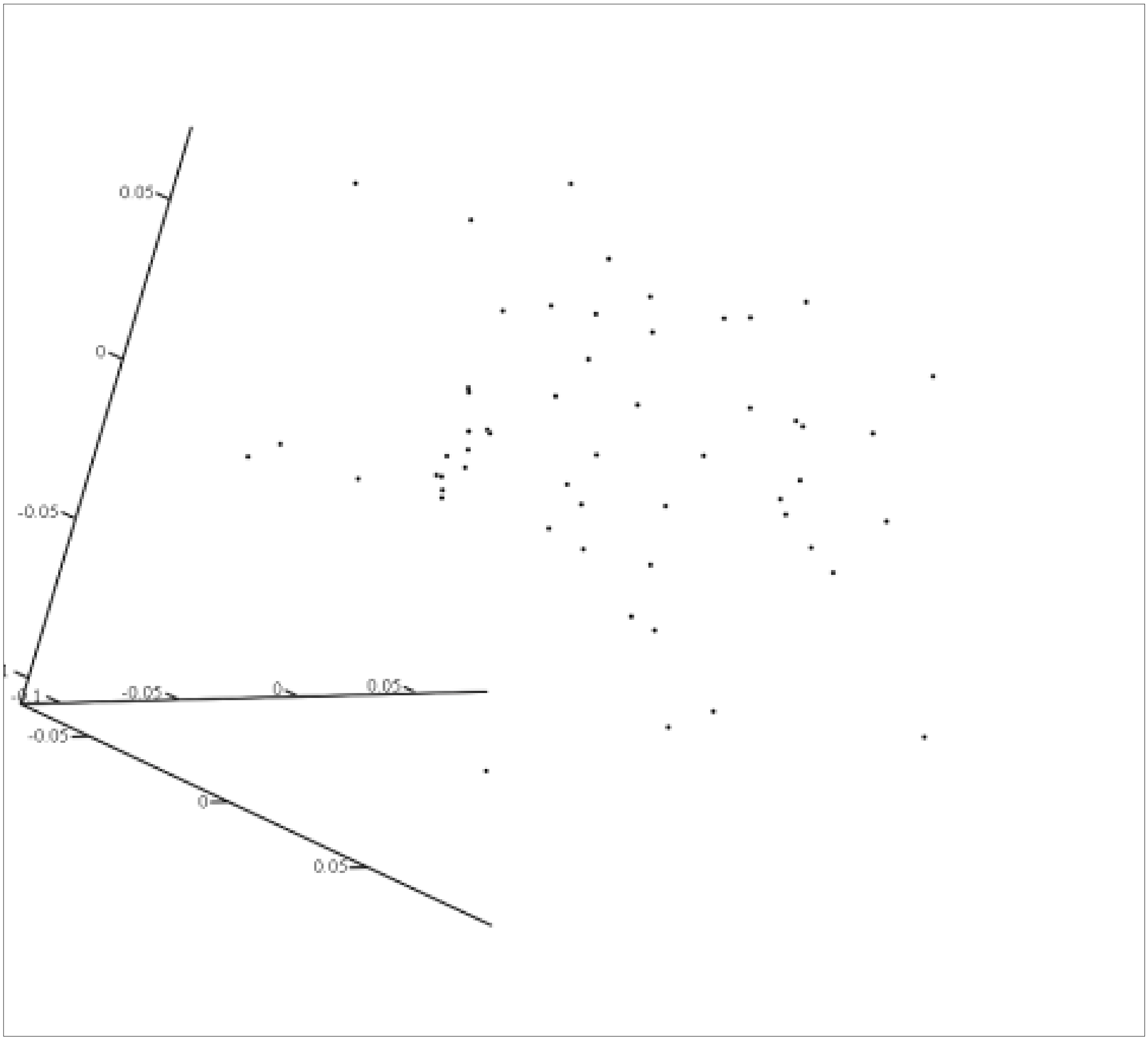
Relative Warps 1 vs 2:



^ Since Eigenvectors can reverse direction, plots may may be "turned around" by using negative numbes, as here. This plot matches tpsRelw Relative Warp 1 vs 2 output.

Plot in 3D for first three Relative Warps:

$\mathrm{Pf} := \mathrm{submatrix}(\mathrm{P}, 0, \mathrm{n} - 1, 0, 2)$



$\left[(-\mathrm{Pf})^{\langle 0\rangle}, (-\mathrm{Pf})^{\langle 1\rangle}, (-\mathrm{Pf})^{\langle 2\rangle}\right]$

080316a tpsRelw.mcd

Problems in Calculating the Uniform Part of Relative Warps:

Above, I am forced to take the uniform components in matrix FU as output from tpsRelw on faith as I cannot duplicate tpsRelw's calculation. The best I can do in interpreting the Rohlf & Bookstein (2003 Eq. 2) formula is this:

Columns in Fr are (x,y) "reference" landmarks from tpsRelw GPA Procrustes fitting.
From Fr, I calculate: α and γ as sums of squares using matrix algebra. I see that $\alpha + \gamma = 1$ as they describe.
x_i's and y_i's are (x,y) coordinates of the "reference" form as stated in the paper for Eq.2.

$$\mathrm{Fr} = \begin{pmatrix} -0.3789 & 0.1375 \\ 0.0784 & 0.115 \\ 0.6592 & 0.1189 \\ 0.1414 & -0.0888 \\ -0.1907 & -0.241 \\ -0.4248 & -0.0506 \\ -0.1092 & -0.0235 \\ 0.2246 & 0.0324 \end{pmatrix}$$

$$\alpha := \left(\mathrm{Fr}^{\langle 0\rangle^{T}} \cdot \mathrm{Fr}^{\langle 0\rangle}\right)_0 \qquad \alpha = 0.8836 \qquad x := \mathrm{Fr}^{\langle 0\rangle}$$

$$\gamma := \left(\mathrm{Fr}^{\langle 1\rangle^{T}} \cdot \mathrm{Fr}^{\langle 1\rangle}\right)_0 \qquad \gamma = 0.1164 \qquad y := \mathrm{Fr}^{\langle 1\rangle}$$

$$\alpha + \gamma = 1$$

$$x = \begin{pmatrix} -0.3789 \\ 0.0784 \\ 0.6592 \\ 0.1414 \\ -0.1907 \\ -0.4248 \\ -0.1092 \\ 0.2246 \end{pmatrix} \qquad y = \begin{pmatrix} 0.1375 \\ 0.115 \\ 0.1189 \\ -0.0888 \\ -0.241 \\ -0.0506 \\ -0.0235 \\ 0.0324 \end{pmatrix}$$

From my presentation above, each data object resides in columns of Vx and Vy for x & y separately:

Vx =

	0	1	2	3	4	5	6	7	8
0	-0.3954	-0.3766	-0.3679	-0.3467	-0.365	-0.3814	-0.4222	-0.392	-0.3376
1	0.0486	0.0308	0.111	0.0964	0.0848	0.0532	0.0907	0.0115	0.1277
2	0.6417	0.6505	0.6394	0.6607	0.6849	0.6681	0.6439	0.6707	0.6914
3	0.1241	0.1757	0.1362	0.1355	0.1117	0.0921	0.1469	0.1635	0.1547
4	-0.1919	-0.1907	-0.237	-0.2346	-0.1769	-0.2096	-0.1529	-0.1798	-0.2052
5	-0.4265	-0.4225	-0.434	-0.4325	-0.4174	-0.4375	-0.3979	-0.4271	-0.4426
6	-0.068	-0.1012	-0.097	-0.1113	-0.1318	-0.0655	-0.1176	-0.0859	-0.1633
7	0.2674	0.234	0.2494	0.2324	0.2097	0.2805	0.2091	0.2391	0.1749

Vy =

	0	1	2	3	4	5	6	7	8
0	0.1466	0.1498	0.1135	0.1271	0.1306	0.107	0.1711	0.1287	0.094
1	0.1079	0.1357	0.1142	0.0898	0.1066	0.0867	0.1496	0.1124	0.0834
2	0.1253	0.1023	0.1246	0.137	0.1302	0.1215	0.1105	0.1032	0.1438
3	-0.1189	-0.0829	-0.0836	-0.0593	-0.0954	-0.0742	-0.1209	-0.0877	-0.0656
4	-0.243	-0.2565	-0.2479	-0.2492	-0.2375	-0.2155	-0.2527	-0.2143	-0.2149
5	-0.0453	-0.0561	-0.0518	-0.0478	-0.0445	-0.0476	-0.0572	-0.0508	-0.0314
6	-0.0124	-0.0202	-0.0092	-0.028	-0.0214	-0.0184	-0.022	-0.0219	-0.0283
7	0.04	0.028	0.0401	0.0303	0.0313	0.0404	0.0216	0.0304	0.0189

Calculations of u_1 and u_2 as I interpret Rohlf & Bookstein (2003) Eq. 2:

$$i := 0..7 \qquad j := 0..54$$

$$u1_j := \frac{\left(\alpha \cdot \sum \overrightarrow{\left[y \cdot \left[\left(Vx^{\langle j\rangle}\right) - x\right]\right]} + \gamma \cdot \sum \overrightarrow{\left[x \cdot \left[\left(Vy^{\langle j\rangle}\right) - y\right]\right]}\right)}{\sqrt{\alpha \cdot \gamma}}$$

$$u2_j := \frac{-\gamma \cdot \sum \overrightarrow{\left[x \cdot \left[\left(Vx^{\langle j\rangle}\right) - x\right]\right]} + \alpha \cdot \sum \overrightarrow{\left[y \cdot \left[\left(Vy^{\langle j\rangle}\right) - y\right]\right]}}{\sqrt{\alpha \cdot \gamma}}$$

< vectorization function here allows calculation of sum of each $y_i\Delta y_j$ etc. This corresponds to Rohlf & Bookstein's description that x_i & y_i are based on the "reference" form wheras the Δx_j and Δy_j are based on subtraction of (x,y) coordinates each data object j from the reference, or vise versa.

$$u := \mathrm{stack}\left(u_1{}^T, u_2{}^T\right)$$

my numbers:

u =

	0	1	2	3	4	5
0	-0.017000242	-0.028795295	0.048357076	0.05761333	0.014629511	0.025478326
1	0.012615576	0.016690275	-0.004895186	-0.009148192	-0.003913132	-0.045525594

tpsRelw output:

$FU^T =$

	0	1	2	3	4	5
0	0.01607983	0.02238233	-0.013580967	-0.019621985	-0.006632972	-0.051723556
1	-0.015424154	-0.026948002	0.049321812	0.058179717	0.014493729	0.018436326

Difference in sign of the numbers probably reflects the order of subtraction in calculating Δx_i's & Δy_i's. The remining descrepencies MAY be related to calculating the GPA fit for the "reference" form but, unfortunately, I can only guess.

However, perhaps I have misinterpreted x_i & y_i in their formula:

$$u_{1_j} := \frac{\left(\alpha \cdot \sum \overrightarrow{\left[Vy^{\langle j\rangle} \cdot \left[\left(Vx^{\langle j\rangle}\right) - x\right]\right]} + \gamma \cdot \sum \overrightarrow{\left[Vx^{\langle j\rangle} \cdot \left[\left(Vy^{\langle j\rangle}\right) - y\right]\right]}\right)}{\sqrt{\alpha \cdot \gamma}}$$

$$u_{2_j} := \frac{-\gamma \cdot \sum \overrightarrow{\left[Vx^{\langle j\rangle} \cdot \left[\left(Vx^{\langle j\rangle}\right) - x\right]\right]} + \alpha \cdot \sum \overrightarrow{\left[Vy^{\langle j\rangle} \cdot \left[\left(Vy^{\langle j\rangle}\right) - y\right]\right]}}{\sqrt{\alpha \cdot \gamma}}$$

$$u := \mathrm{stack}\left(u_1{}^T, u_2{}^T\right)$$

u =

	0	1	2	3	4	5
0	-0.013104693	-0.030776161	0.049168437	0.055728294	0.015855323	0.026239733
1	0.014238869	0.018641538	-0.003944762	-0.004760982	-0.003967897	-0.041240677

In this calculation, I used each "data" form in Vx & Vy for the x_i's and y_i's. However, I still don't retreive tpsRelw's numbers. This approach is seemingly suggested by Rohlf in his Technical details section, but it would have been helpful if i or j had actually been defined in this document.

$FU^T =$

	0	1	2	3	4	5
0	0.01607983	0.02238233	-0.01358097	-0.01962198	-0.00663297	-0.05172356
1	-0.01542415	-0.026948	0.04932181	0.05817972	0.01449373	0.01843633