Sehbau: Matching Vectors

The use of the program for vector matching is explained, called **mvec**. It matches the descriptor vectors of two images (**.vec** files) as generated by the descriptor extraction program (**dscx**).

Read https://www.researchgate.net/publication/360033329
Repository https://github.com/Sehbau/MtchVec
PREVIOUS descriptor extraction with **dscx**, https://github.com/Sehbau/DescExtraction

The program **mvec** matches the two list of descriptors using pairwise distance (similarity) calculation and choosing the nearest neighbor. As we usually have unequal list length, the results are different for matching the first image against the second image, and with order reversed (2nd versus 1st). I provide both matching measurements. The user can then combine the values ad libitum.

For each descriptor and each (pyramid) level, a dissimilarity and a similarity value is calculated. The dissimilarity is the Euclidean distance, the similarity is the count of matches that are below a certain threshold value, set with option **[dsc]TolMtc**.

1 Program Use

Two vector files (**.vec**) are given as arguments:

```
> mvec Desc/img1.vec Desc/img2.vec
```

The vector files are required to have the same number of levels, ie. generated by similarly sized images. More flexibility will be added later.

The output is written to **stdout**.

By default, the program matches all four descriptor types for the entire pyramid. The options allow to subselect descriptors and levels, as well as to set attribute weight values.

1.1 Options

--cntTolMtc: tolerance for contour matches, for the similarity metric. Fixed value for all levels, but will try to provide something more flexible. Default: complicated.

--rsgTolMtc: tolerance for radial descriptor matches. Analogous to option **cntTolMtc**.

--arcTolMtc: tolerance for arc segment matches (see **cntTolMtc**).

--strTolMtc: tolerance for straighter segment matches (see **cntTolMtc**).

More in progress.

1.1.1 Utility:

--prms: displays the parameters used.

2 Output

The program writes the dissimilarity and similarity values to **stdout** and looks as follows. The output is organized in two principal sections, 12 and 21, each of which lists the dissimilarity (Dist) and the similarity (Simi) values for each descriptor type (skl, rsg, arc and str) and each pyramid level.

```
-----  MatchResults  -----
nLevRes 5
-----  12  -----
Dist:
SklDis12 13.175 39.664 45.217 41.913 13.573
RsgDis12 109.845 27.919 12.824 8.965 3.398
ArcGstDis12 114.219 46.689 53.185 21.986 8.605
StrGstDis12 38.233 14.913 32.930 21.659 11.041
ArcDis12 0.000 0.000 0.000 0.000 0.000
StrDis12 0.000 0.000 0.000 0.000 0.000
Simi:
SklSim12 0.000 16.000 108.000 201.000 30.000
RsgSim12 68.000 42.000 10.000 1.000 0.000
ArcGstSim12 2.000 2.000 19.000 2.000 2.000
StrGstSim12 0.000 0.000 35.000 6.000 1.000
ArcSim12 0.000 0.000 0.000 0.000 0.000
StrSim12 0.000 0.000 0.000 0.000 0.000
-----  21  -----
Dist:
SklDis21 53.248 83.856 93.762 80.252 23.530
RsgDis21 223.865 77.948 22.488 7.217 3.611
ArcGstDis21 169.397 54.075 42.516 8.441 2.109
StrGstDis21 99.205 21.655 37.785 17.365 2.734
ArcDis21 0.000 0.000 0.000 0.000 0.000
StrDis21 0.000 0.000 0.000 0.000 0.000
Simi:
SklSim21 0.000 16.000 111.000 236.000 30.000
RsgSim21 68.000 42.000 11.000 1.000 0.000
ArcGstSim21 2.000 2.000 19.000 2.000 2.000
StrGstSim21 0.000 0.000 35.000 6.000 1.000
ArcSim21 0.000 0.000 0.000 0.000 0.000
StrSim21 0.000 0.000 0.000 0.000 0.000
-----  img  -----
disV12 727138240.000
disV21 5543590912.000
```

The very last section contains the (total) dissimilarity value for 1 versus 2, and 2 versus 1:

```
-----  img  -----
disV12 1580528384.000
disV21 41524080640.000
```

The combination of both achieved better results typically. Similarity values and combination of both has never been really evaluated but could potentially improve results significantly.

The two principal sections contain the individual match values for the descriptor types per level. For example for the matching of 1 versus 2:

```
-----  12  -----
Dist:
SklDis12 13.175 39.664 45.217 41.913 13.573
...
Simi:
```

```
SklSim12 0.000 16.000 108.000 201.000 30.000
...
```

the line `SklDis12...` lists the dissimilarity values for the skeleton (selected contours) for each level (5 in this case); the line `SklSim12...` lists the similarity values for each level and are integers as they represent the count of succesful matches (given a fixed threshold).

The syllables summarized:
- `Skl`: (selected) contours
- `Rsg`: radial descriptors
- `ArcGst`: subselected arc segments
- `StrGst`: subselected straighter segments
- `Arc`: full set of arcs, by default *not* matched
- `Str`: full set of arcs, by default *not* matched