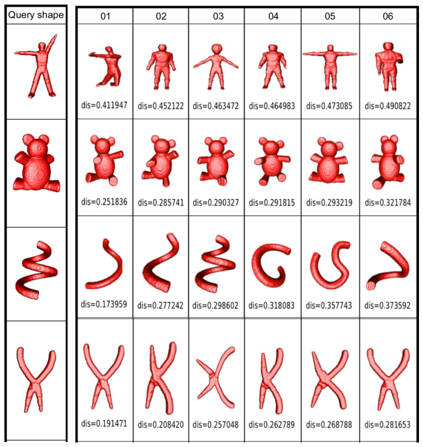
Technical tips: Step 5

1. Presenting and assessing the results of spatial search

Step 5 of the assignment asks you to implement a spatial search mechanism to accelerate searching for a given query shape in a given shape database. How to present and `demonstrate' the results of this implementation?

(1) Presentation: The simplest (and best) is to present a table organized as follows (see also snapshot below)



Query shape	01	02	03	04	05	06
	dis=0.411947	dis=0.452122	dis=0.463472	dis=0.464983	dis=0.473085	dis=0.490822
	dis=0.251836	dis=0.285741	dis=0.290327	dis=0.291815	dis=0.293219	dis=0.321784
	dis=0.173959	dis=0.277242	dis=0.298602	dis=0.318083	dis=0.357743	dis=0.373592
	dis=0.191471	dis=0.208420	dis=0.257048	dis=0.262789	dis=0.268788	dis=0.281653

- Each *row* indicates a shape. Think of a few (4..5) such rows, each for a shape in a different class.
- The *leftmost column* shows the query shapes. Select one such query shape in each of the abovementioned classes.
- The *other columns* (6 of them in the above example) show the retrieved shapes for the respective queries, sorted left to right based on distance. That is, they show the K-nearest-neighbors of the query shape (in our example, K=6). We exclude in the presentation here the query shape itself (which should always be the first nearest-neighbor of the query shape).
- Each *cell* shows the actual shape. For the retrieval columns, we also show the distance between the retrieved shape and the query shape as a label.

You can use *exactly* the same type of table for illustrating the results obtained in step 4 of the assignment using your 'custom' distance metric. The *difference* is that, in step 4, you do retrieval based on the computation of your 'custom' distance metric, whereas in step 5, you do retrieval based on the 'built in' distance of the spatial search mechanism you are using to accelerate search. For instance, if you use Approximate Nearest Neighbors (ANN) as this spatial search mechanism, the distance can be any of the L_p metrics (Minkowski norms), the one most used in practice being the Euclidean (L_2) distance.

(2) Assessment: Given the above, the table (query results) using your own distance metric will likely *differ* from the table (query results) using the spatial search mechanism. Comparing the two tables you should be able to see and reason about the following points:

- The two tables are very **similar**: This means that your 'custom' distance metric doesn't perform that differently from the L_2 metric used by ANN. Why is this so? Try to explain this in writing.
- The table for your 'custom' results looks **better** than the one from ANN. What does this mean? Hint: It means your distance function is modelling similarities better than L_2. Try to explain this in writing.

- The table for your ‘custom’ results looks **worse** than the one from ANN. What does this mean? Hint: Probably there’s a problem with the design or implementation of your ‘custom’ metric. Replace this metric in your code with the simple L_2 metric. Then, the two tables should be identical. If not, there is likely a coding problem somewhere.

2. Presenting and assessing the results of dimensionality reduction

The bonus step asks you to create a ‘visual map’ of all the shapes in your database using dimensionality reduction (DR). Several points are important when doing this:

- **Be minimal:** First, create the simplest possible visualization: A scatterplot where each database shape (feature vector) corresponds to a point, colored by the class the shape belongs to. You should only try to add extra things, like brushing the point to show its ID, or feature vector, or even thumbnail in a tooltip, *later on*, after the basic colored scatterplot is created, and only if you have additional time. Implementing such interactive features typically costs quite a lot of energy and tweaking.

- **What is a good plot:** The DR plot reflects how well separated the feature vectors are. Hence, a good plot will place *similar* shapes *close* to each other, and *different* shapes *far away* from each other. Do you see that same-color clusters are well separated in the plot? This is a good sign of separation, since shapes in the same class are typically quite similar among themselves, and also different from shapes in other classes. Do you see a plot in which colors are heavily mixed? This is a sign of poor separation, which very likely indicates that your feature vectors don’t capture well the similarity of the 3D shapes.

- **Technical settings:** The t-SNE projection method you have to use for creating the DR plot has two main parameters to set: perplexity and number of iterations. Good default settings are perplexity=30..40 and a number of iterations of at least 500. You should experiment a bit with these values since t-SNE is not a deterministic algorithm, and its results are quite sensitive on the parameter setting. That is: If you get a bad plot, in which colors are heavily mixed (see above), don’t directly assume your feature vectors are bad. Try a few more t-SNE parameter values before reaching that conclusion on the feature vectors.