

## 3D Shape Recognition

### =====

### 3D Shape Classification Data

### =====

```
('vase-data.mat')
```

```
num_shapes = 200;
```

```
-----  
Category 1  
-----
```

```
category_number = 1;
```

```
for j = 1 : num_shapes
```

```
radius = 2 + rand();
```

```
t = 0:pi/11:2*pi;
```

```
[A,B,C] = cylinder(radius + t.*cos(t));
```

```
data_array = shape_to_data(A,B,C);
```

```
X = [];
```

```
Y = [];
```

```
Z = [];
```

```
num_items = size(data_array,2);
```

```
for i = 1 : num_items
```

```
temp_vec = data_array{i};
```

```
x = temp_vec(1);
```

```
y = temp_vec(2);
```

```
z = temp_vec(3);
```

```
X = [X x];
```

```
Y = [Y y];
```

## 3D Shape Recognition

```
Z = [Z z];
endfor

shape_vector = [X Y Z];
N = size(shape_vector,2);
shape_vector(N + 1) = category_number;
data_set1{j} = shape_vector;
endfor

-----
Category 2
-----

category_number = 2;
for j = 1 : num_shapes
radius = 4 + rand();

t = 0:pi/11:2*pi;
[A,B,C] = cylinder(radius + t.*cos(t));
data_array = shape_to_data(A,B,C);
X = [];
Y = [];
Z = [];
num_items = size(data_array,2);

for i = 1 : num_items
temp_vec = data_array{i};
x = temp_vec(1);
y = temp_vec(2);
```

### 3D Shape Recognition

```
z = temp_vec(3);
X = [X x];
Y = [Y y];
Z = [Z z];
endfor

shape_vector = [X Y Z];
N = size(shape_vector,2);
shape_vector(N + 1) = category_number;
data_set2{j} = shape_vector;
endfor

-----
Category 3
-----

category_number = 3;
for j = 1 : num_shapes
radius = 6 + rand();

t = 0:pi/11:2*pi;
[A,B,C] = cylinder(radius + t.*cos(t));
data_array = shape_to_data(A,B,C);
X = [];
Y = [];
Z = [];
num_items = size(data_array,2);
```

### 3D Shape Recognition

```
for i = 1 : num_items
temp_vec = data_array{i};
x = temp_vec(1);
y = temp_vec(2);
z = temp_vec(3);
X = [X x];
Y = [Y y];
Z = [Z z];
endfor

shape_vector = [X Y Z];
N = size(shape_vector,2);
shape_vector(N + 1) = category_number;
data_set3{j} = shape_vector;
endfor

-----
GENERATES DATA TREES
-----

data_array = {data_set1{:} data_set2{:} data_set3{:}};

[category_tree delta_tree anchor_tree] = generate_data_tree_N(data_array,N);

-----
TESTS PREDICTIONS
-----

[success_rate success rejection_rate rejection fail_rate fail] =
generate_random_trials(300, 1, 1200, N, anchor_tree, delta_tree)
```