MCDB/BCHM 4100/6440 – Microscopy Labs

Lab 3:

Tracking cells in time-lapse microscope images

Lecturer: Jian Wei Tay

Date: 2021-10-28

Before we start, does everyone have code to segment and measure Centroid positions?

Learning objectives

- Understand the concept behind the nearest-neighbor tracking algorithm
- Learn how to write if statements
- Learning how to plan out a data structure to hold time-series data for each cell

if statements

- if statements allow code execution to be controlled depending on the value of a variable

```
if <logical expression>
    %Code here runs if expression is true
end
```

Example

```
V = 10

if V > 5
    output = true;
end
```

Extending the if statement using elseif

- You can add additional logical statements using the elseif keyword

```
if <logical expression 1>

elseif <logical expression 2>
    %Code here runs only if expression 1 is
    %false and expression 2 is true
end
```

Example

```
V = 10

if V > 5
    output = true;
elseif V < 3
    output = false;
end
```

Note: In this example, only the first statement runs (output = true)

Example

```
V = 3

if V > 5
    output = true;
elseif V < 3
    output = false;
end
```

Note: In this example, the second statement runs (output = false)

The else keyword

- Use the else keyword to run some code if all prior statements were false

```
if <logical expression 1>

elseif <logical expression 2>

else
    %Code here runs only if expression 1 is
    %false and expression 2 is false
end
```

Note: The `else` statement has no logical expressions and must be last (before the end keyword)

```
V = 4

if V > 5
    output = true;
elseif V < 3
    output = false;
else
    disp('Value is 4!')
end
```

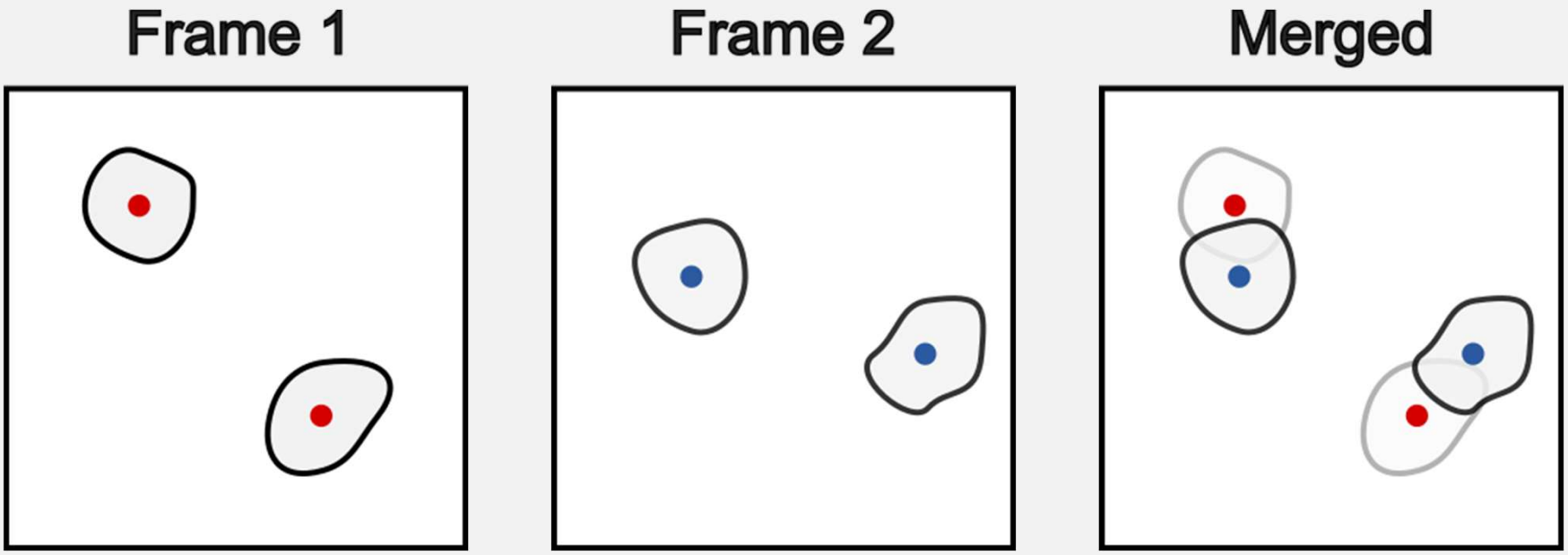
Note: The `else` statement will run here

Questions?

The goal of a tracking algorithm

is to figure out which data belongs to a specific object

The tracking problem



Where have the cells in Frame 1 moved to in Frame 2?

The tracking problem in code

```
for iT = 1:reader.sizeT
      image = getPlane(reader, 1, 1, iT);
      mask = imbinarize(image);

      data = regionprops(mask, 'Centroid',
'Area');

      %How do we link data together?
end
```

The tracking problem

- When we use regionprops, we only get back a list of centroid positions
- Example:

Index	X	Y	Area
1	10.5	5	100
2	15	3.5	120
3	14	10	115

Frame 1

Index	X	Y	Area
1	15.1	3.2	118
2	14.8	11	118
3	11.2	5.2	102

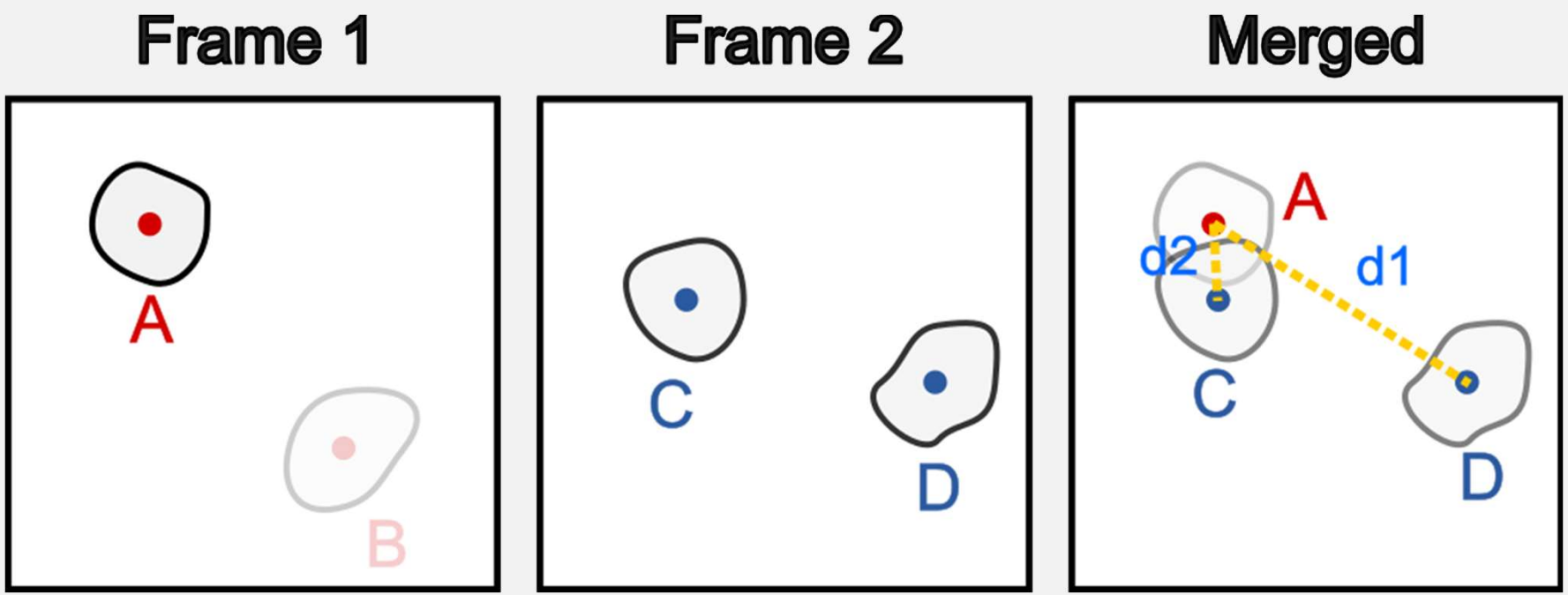
Frame 2

Where have the cells in Frame 1 moved to in Frame 2?

Nearest neighbor algorithm

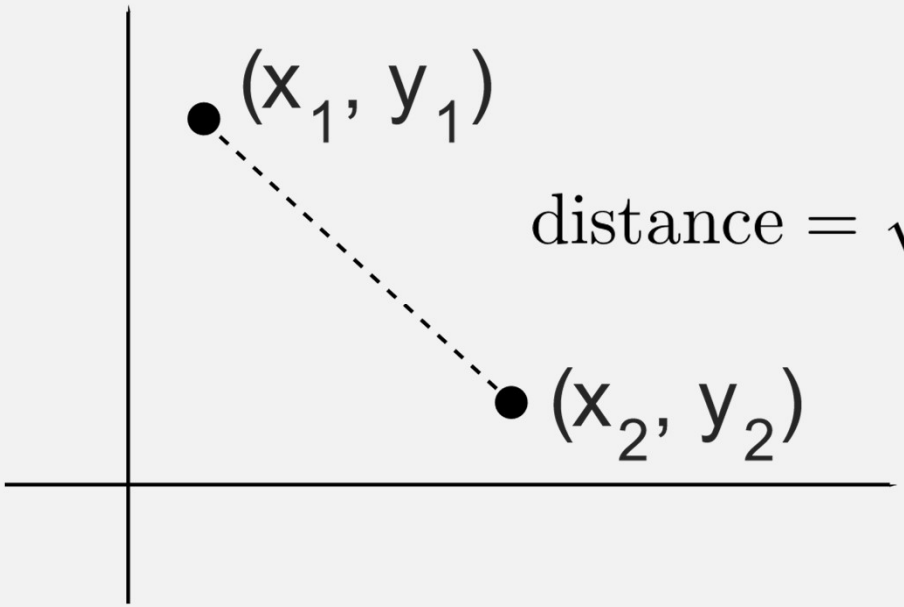
- Measure the distance from the last known (centroid) position of an object to the position of every detected object in the current frame

- Link objects with the shortest distance (i.e., the *nearest-neighbor*)



Since d2 < d1, link A and C

Distance between two points



$$\text{distance} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

The tracking problem

Index	X	Y	Area
1	10.5	5	100
2	15	3.5	120
3	14	10	115

Frame 1

Index	X	Y	Area
1	15.1	3.2	118
2	14.8	11	118
3	11.2	5.2	102

Frame 2

Where have the cells in Frame 1 moved to in Frame 2?

The tracking problem

Frame 1

Index	X	Y	Area
1	10.5	5	100
2	15	3.5	120
3	14	10	115

Frame 2

Index	X	Y	Area
1	15.1	3.2	118
2	14.8	11	118
3	11.2	5.2	102

Index	Frame	X	Y	Area
1	1	10.5	5	100
	2	11.2	5.2	102
2	1	15	3.5	120
	2	15.1	3.2	118
3	1	14	10	115
	2	14.8	11	118

Desired output

Data structures in MATLAB

- We've already used a couple of data structures in MATLAB:
  - Matrices (Arrays of numbers)
  - Structs (Structured data)

- I am going to suggest that you use a struct

Why use a struct?

- You can label different data within a struct using the fieldname
- Structs can hold data of different sizes and types
- Structs can also be multi-element – this way you can hold data from different cells and keep the field names the same

Creating a struct

- You can create a struct be declaring it directly:

```
>> S.Area = 100;
>> S.MajorAxisLength = 10;
```

Note: The basic syntax is `variableName.fieldName`. MATLAB will know that you want to create a struct.

Adding elements to a struct

- You can add elements by indexing just like with a vector

```
>> S(2).Area = 110;
>> S(2).MajorAxisLength = 25;
```

Give the tracking problem a try and let me know when you have questions

You are encouraged to work together