Lecture 28:
Nearest-neighbor Tracking (I)

University of Colorado Boulder

MCDB/BCHM 4312/5312
Fall 2020

Announcements about midterm 2

- Next **Wednesday, November 11**
- You have **2 hours and 15 minutes** to complete the exam between **10:00 am and 2:00 pm**
- A **sample exam (Canvas quiz) and instructions** has been added to the Week 12 module – I encourage you to check it out so you know how to submit your answers and how timing works
- **Bring questions to review session (Monday's lecture)**

Changes from midterm 1

- Since we are unable to grade the exam on Canvas without using the "assignment" feature and we cannot time the exam without the "quiz" feature, we have to use **both**

Changes from midterm 1

- You will **start your exam using the quiz**
- When you are done or the timer is up, the quiz will be **submitted**
- A message containing a link will appear under the question
- Click on the link and **upload your answer to the assignment**
- We will check that the quiz end time and the assignment submission time is similar

Lecture 28:
Nearest-neighbor Tracking (I)

University of Colorado Boulder

MCDB/BCHM 4312/5312
Fall 2020

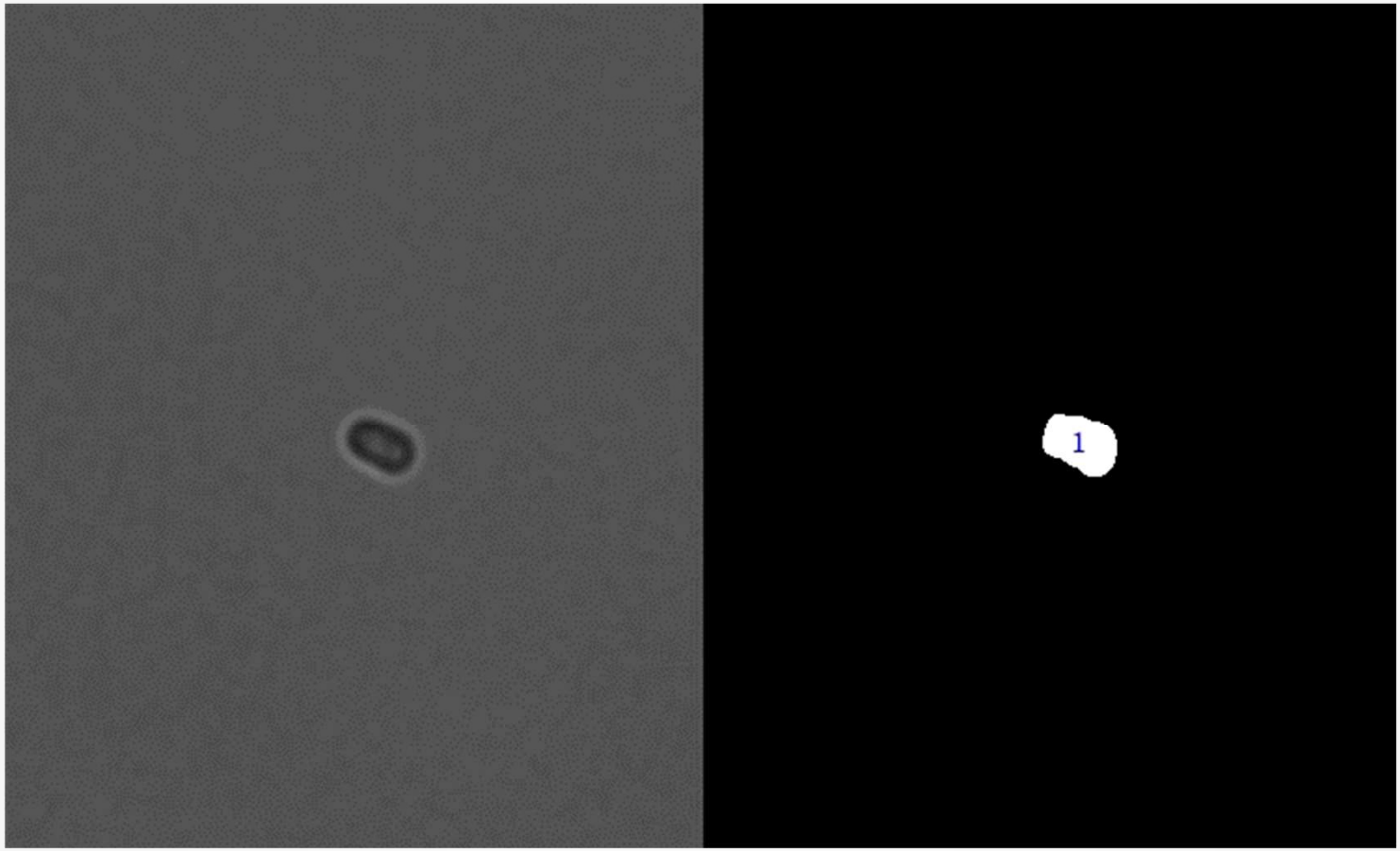
Analyzing time-lapse movies

- Read in image
- Segment cells
- Measure data using `regionprops`
- Link data between objects (tracking)
- Repeat process for each frame

Will be split over two lectures

The goal of a tracking algorithm

is to collect data belonging to a single object


1

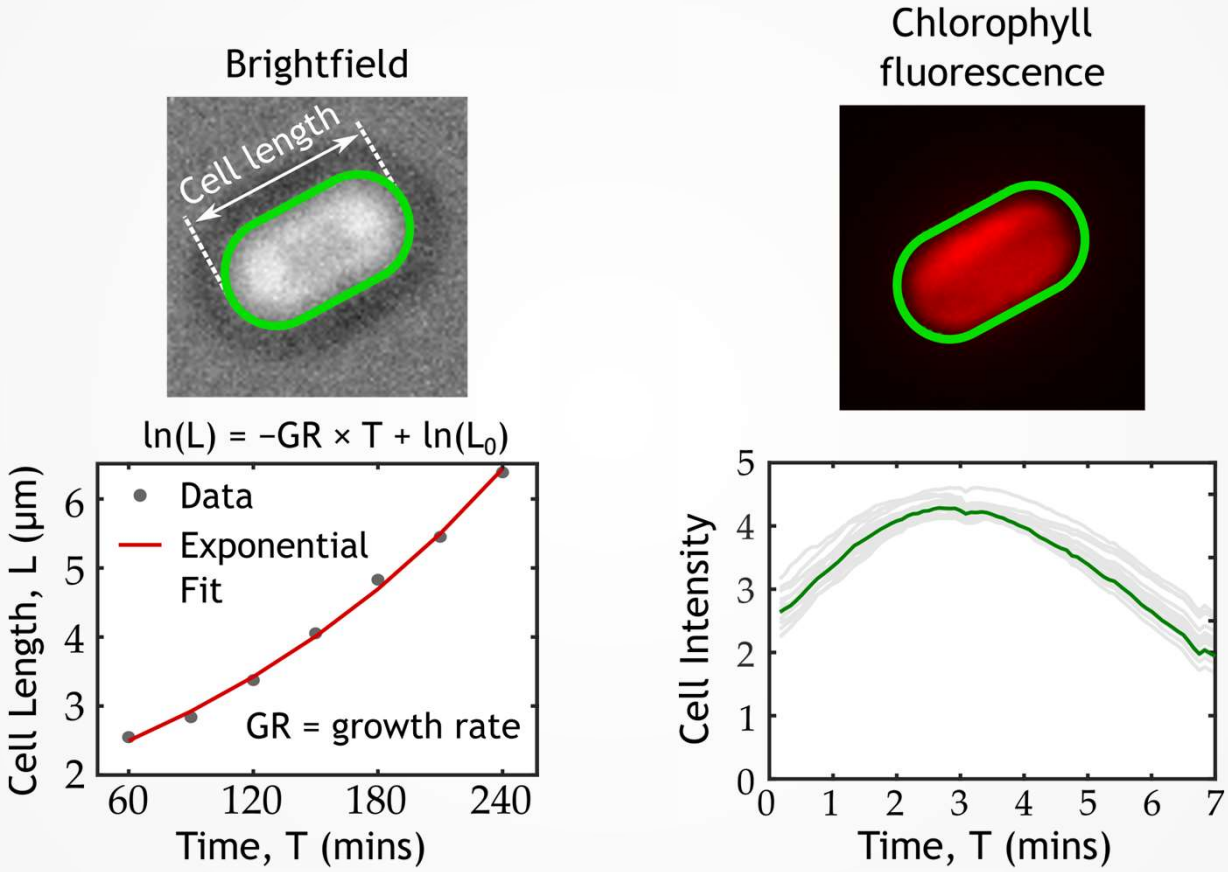
Output

```
S =
  struct with fields:

               ID: 1
         MotherID: NaN
       DaughterID: [46 47]
           Frames: [1 2 3 4]
             Area: [4×1 double]
         Centroid: [4×2 double]
   MajorAxisLength: [4×1 double]
   MinorAxisLength: [4×1 double]
      Orientation: [4×1 double]
     PixelIdxList: {[1230×1 double]  [1335×1 double]  [1461×1 double]  [1589×1 double]}
      TotalIntRed: [4×1 double]
      TotalIntCy5: [4×1 double]
      TotalIntRFP: [4×1 double]
   RegisteredPxInd: {[1230×1 double]  [1335×1 double]  [1461×1 double]  [1589×1 double]}
```

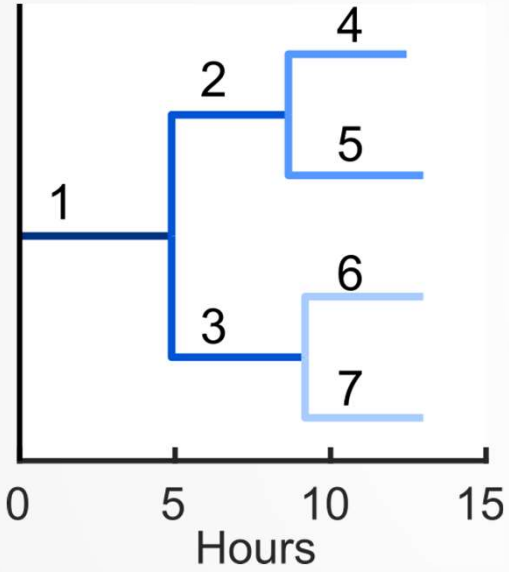
More on data structures next week

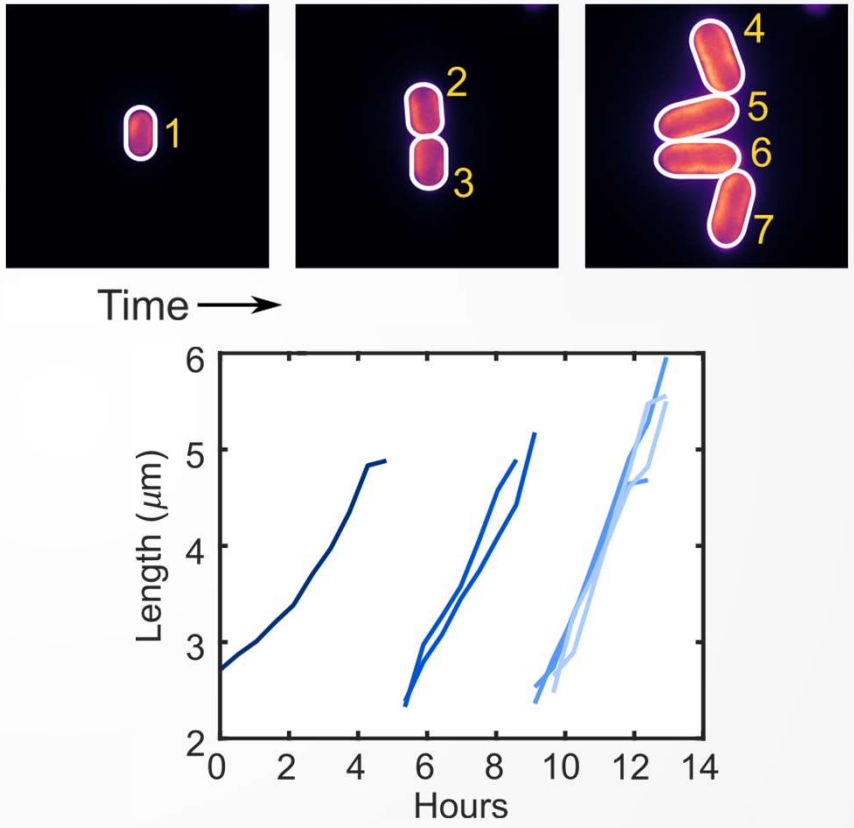
Single-cell measurements



Kristin A. Moore, Jian Wei Tay, Jeffrey C. Cameron *bioRxiv* (2019) doi: 10.1101/661256

Tracking lineage





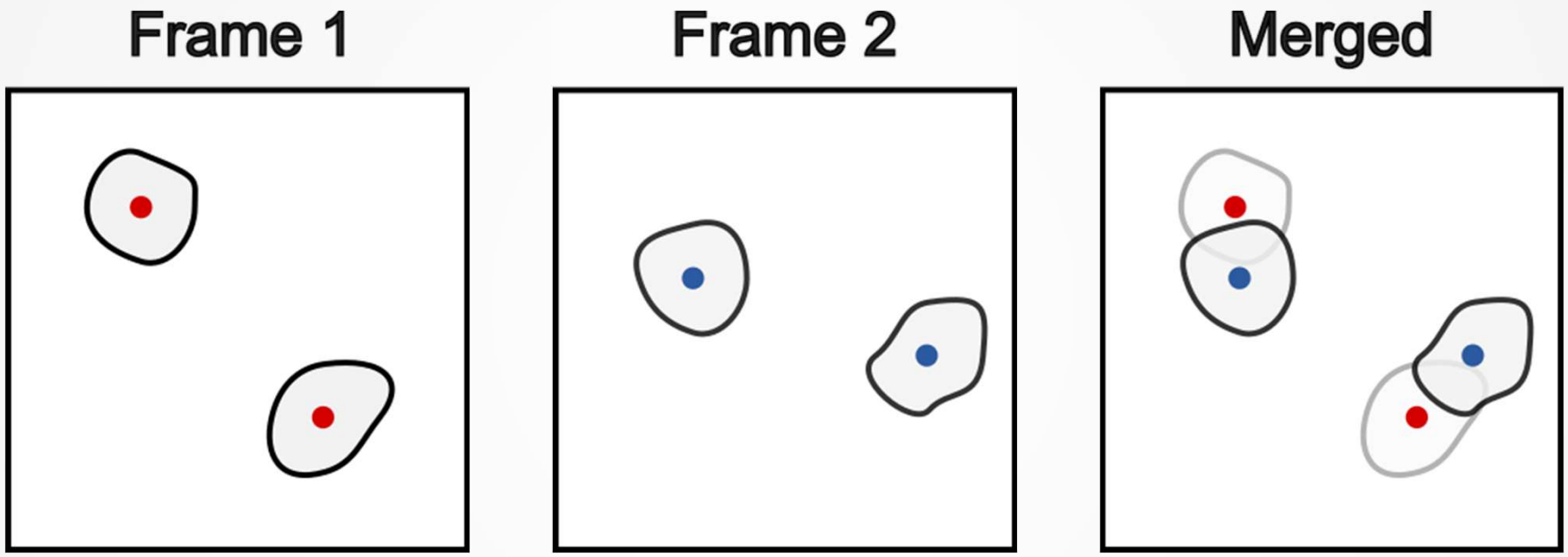
Kristin A. Moore, Jian Wei Tay, Jeffrey C. Cameron *bioRxiv* (2019) doi: 10.1101/661256
Nicholas Hill, Jian Wei Tay, …, Jeffrey C. Cameron *Nature Microbiology* ***6****, aba1269* (2020)

Implementing a tracking algorithm

Terminology

- **Linking** – the process of associating objects in one frame with objects in another
- **Detection** – an unlinked object in current frame (unlinked objects)
- **Track** – a collection of data belonging to a single object

The tracking problem

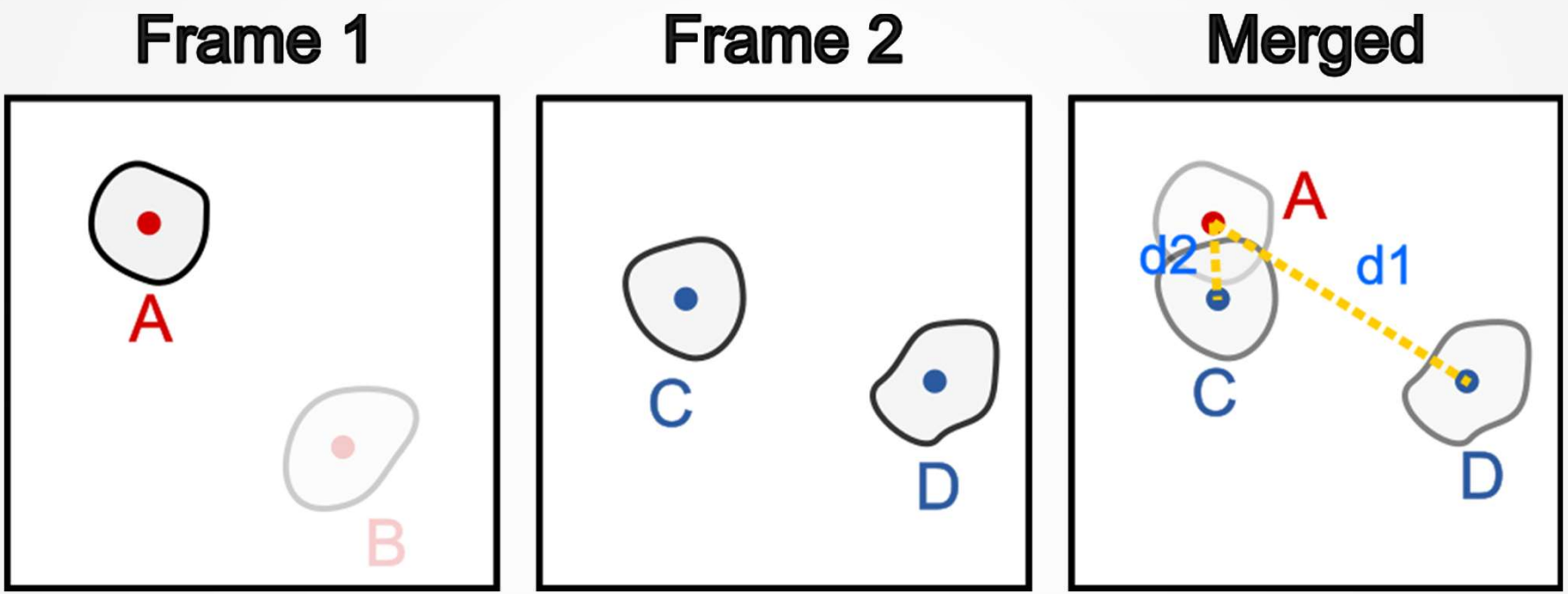


How do we link objects in frame 2 with objects in frame 1?

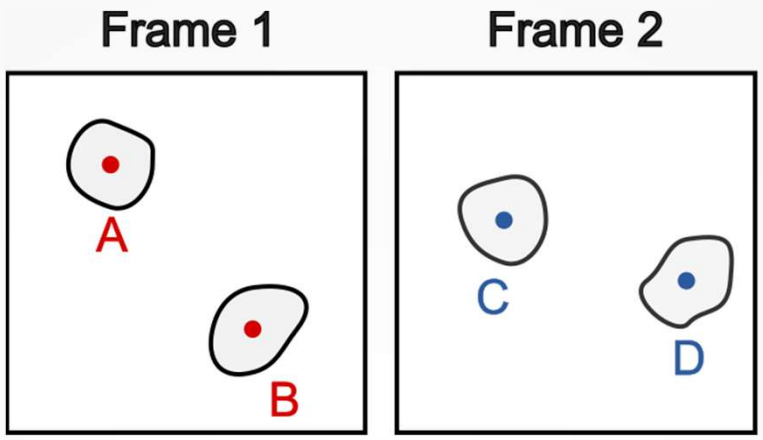
Nearest neighbor algorithm

- 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



Desired output:

$$\texttt{Track1.MeanIntensity} = [I_A, I_C]$$

$$\texttt{Track2.MeanIntensity} = [I_B, I_D]$$

Note: `MeanIntensity` is an example property

Task 1

- Read in the first **two** frames of the image `'twonucl_1.tif'`
- Write a script to segment the nuclei and measure the centroid position

Task 2

- Create a new struct to store data

This struct will hold the "tracks"

Creating struct arrays

- Initializing a struct array

```
track = struct('Centroid', {})
```

Creates an empty (0x0) struct with a field named 'Centroid'

Creating a multi-element struct

- Basic syntax:

  S(index).Fieldname

- Example:

  track(1).Centroid = 1

  track(2).Centroid = 5

Task 2

- Create a new struct to store data

This struct will hold the "tracks"

Task 3

- Modify your code to also measure the area of the objects
- Add this area to the track struct

Adding a field

- Every element in a struct will have the same fields

- Example:

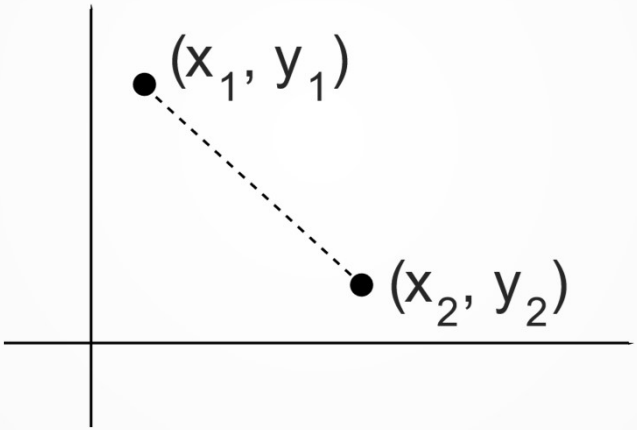
```
track(1).Area
%Adds the field Area to the entire
%structure
track(2)
```

Task 3

- Modify your code to also measure the area of the objects
- Add this area to the track struct

Task 4

- Compute the distance of the objects in frame 1 to the objects in frame 2



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

Task 5

- Find the nearest neighbor

```
[~, index] = min(distance);
```

Returns the index of the smallest value in the vector `distance`

Task 6

- Append the new information to the track struct

Review – Growing a matrix

Which of the following commands will add a row to the 1x2 matrix M? **Select all that apply.**

A. `M(2, :) = [10 15];`

B. `M(:, 2) = [10 15];`

C. `M = [M; [10 15]];`

D. `M(end + 1) = [10 15];`

Review – Growing a matrix

Which of the following commands will add a row to the 1x2 matrix M? **Select all that apply.**

A. `M(2, :) = [10 15];`

B. `M(:, 2) = [10 15];`

C. `M = [M; [10 15]];`

D. `M(end + 1) = [10 15];`

See lecture 4
+ YouTube videos

Task 6

- Append the new information to the data struct

Task 7

- Modify your code so it now repeats the same process for frame 3
- You will need to use a for loop

Assumption

Images are acquired at a high enough frame rate that objects do not move "too much" between frames

How far can each object travel between frames?

- Depends on how far objects can travel in an image before crossing paths with another object
- Average distance between objects in image
- The more crowded an image, the stricter the requirement is (smaller step size is better)

Practice

- Try to write a script from scratch to implement the nearest-neighbor algorithm
- Practice images: `twonucl_2.tif`