Graphs in Computer Vision

EN 500.111 Lecture 9

Presentation Check-In

- ➔ Presentations are next week. **Please send your slides by midnight the night before next class to mkirsche @ jhu.edu!**
- ➔ Is there anything you need from me as you work on your presentations?
- ➔ Feel free to email me if you have any questions!

Images are everywhere

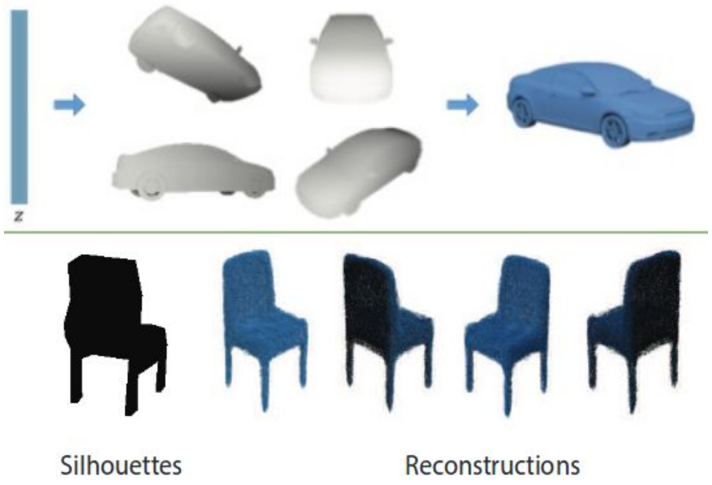
- ➔ Everything you see can be thought of as a very large image that updates about 30-60 times per second
- ➔ With the growth of smartphone cameras, digital images are becoming an extremely popular way to record some of these moments

> In 2014, according to Mary Meeker's annual Internet Trends report, people uploaded an average of **1.8 billion** digital images every single day. That's **657 billion** photos per year. Another way to think about it: Every two minutes, humans take more photos than ever existed in total 150 years ago. Nov 2, 2015
>
> www.theatlantic.com › technology › archive › 2015/11
>
> How Many Photographs of You Are Out There In the World ...

About Featured Snippets Feedback

What is computer vision?

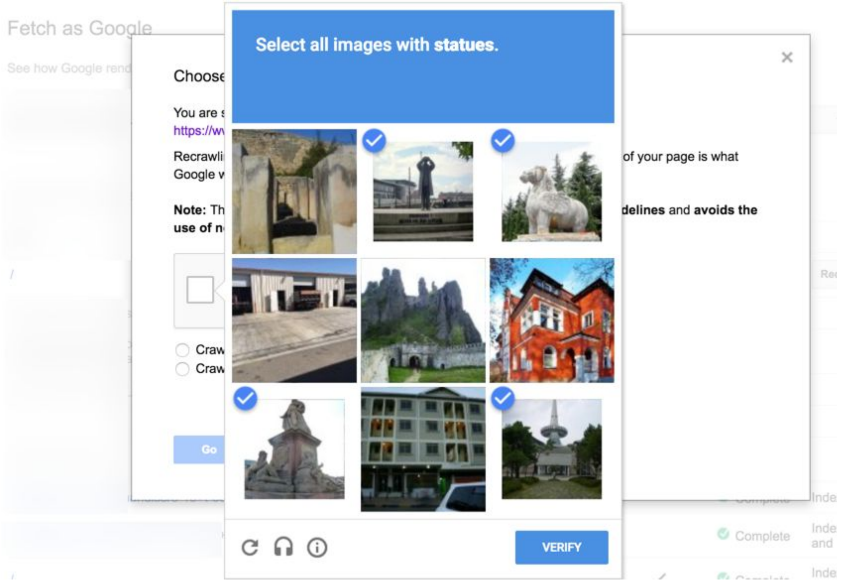
➔ Taking images and inferring high-level meaning from them



(Wikipeda)

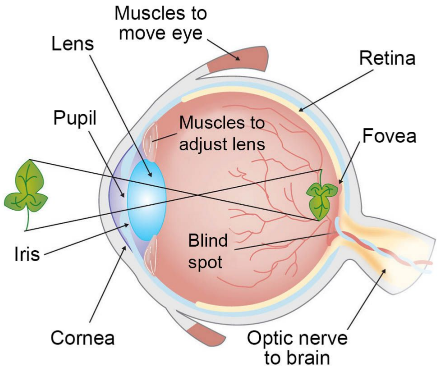
Humans are pretty good at it already

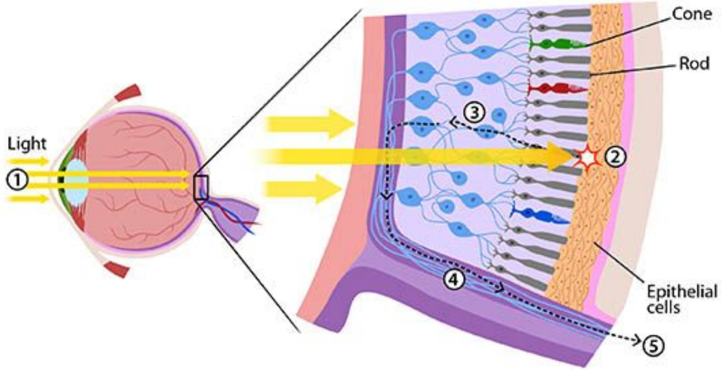
➔ We're being constantly bombarded with tons of visual input, and can still pick out specific objects and recognize things we've seen before



How do we do it?

- ➔ Photoreceptors in the back of the eye get activated by different types/colors of light and transmit that information back to the brain
- ➔ How the brain classifies and recognizes objects is still largely unknown
- ➔ Experiments have been done to highlight different portions of the brain which are active during different tasks



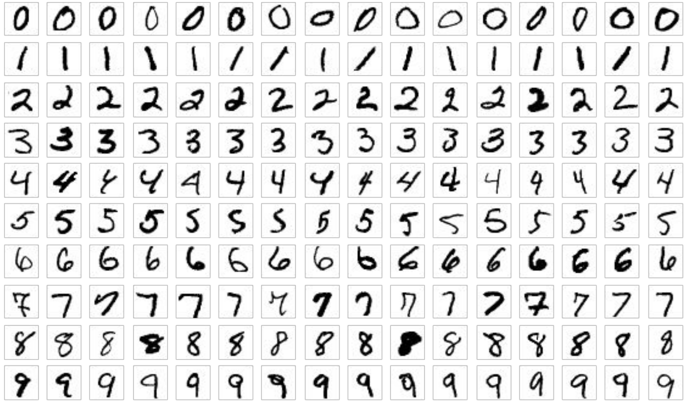


(https://askabiologist.asu.edu/rods-and-cones)

Computer Vision Applications

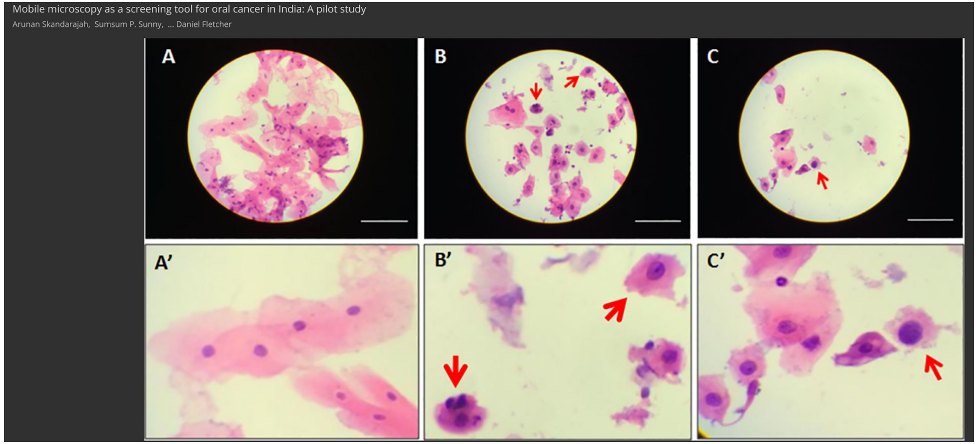
- Tagging images with content (e.g., Google Images database)
- Finding images similar to an existing one (search-by-image applications)
- Deciphering shapes or characters (image to text conversion)
- And many others!

What applications can you think of where a computer has to process an image in some way?



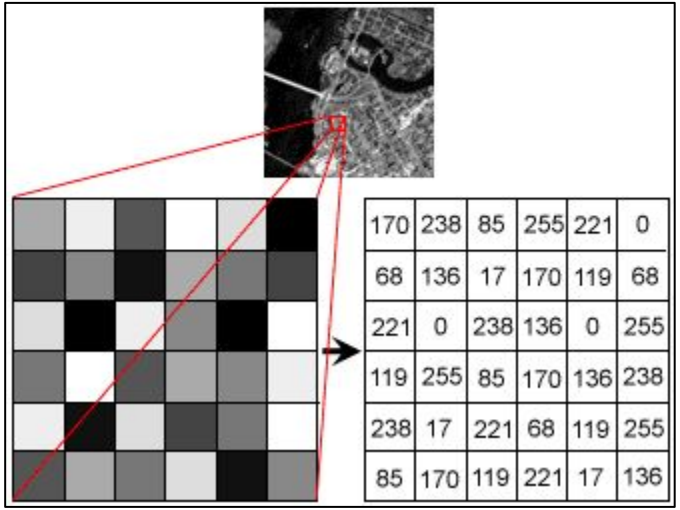
Computer Vision for Disease Diagnosis

- ➔ One application is to analyze microscopy images to screen for different diseases or abnormalities
- ➔ Software exists for detecting different types of cancer (see right), diabetes, and more



Computer Image Representation

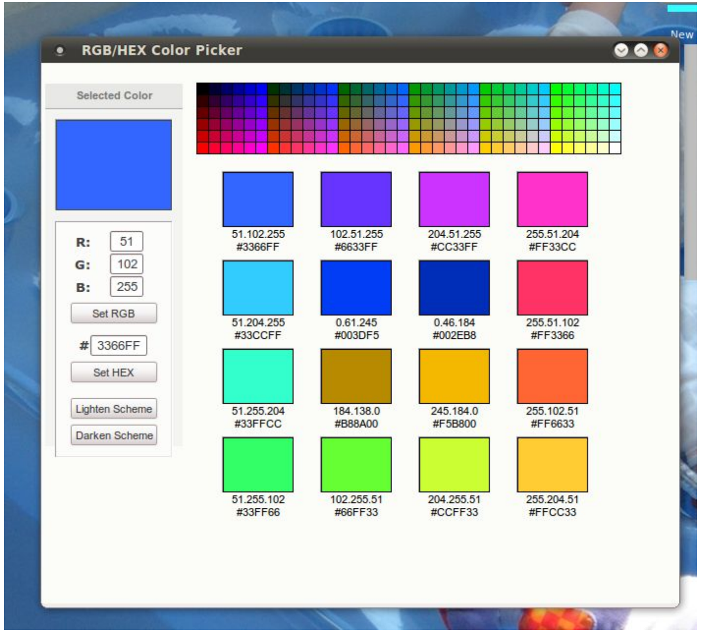
- ➔ Every image is represented as a rectangle of pixels, each of which has its own intensity value
- ➔ For example, the image to the right is 334 by 251 pixels (about 83,000 pixels total)
- ➔ Intensity values range from 0 to 255



https://ai.stanford.edu/~syyeung/cvweb/tutorial1.html

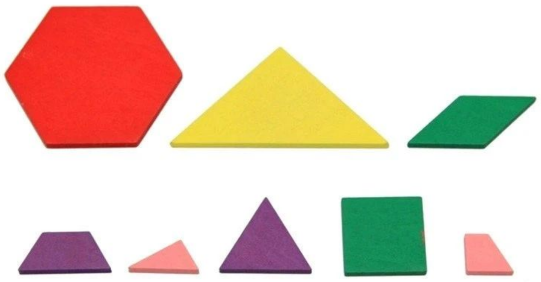
Color images

➔ The same as grayscale images, but instead of a single intensity value each pixel has 3 values - the intensities of red, green, and blue

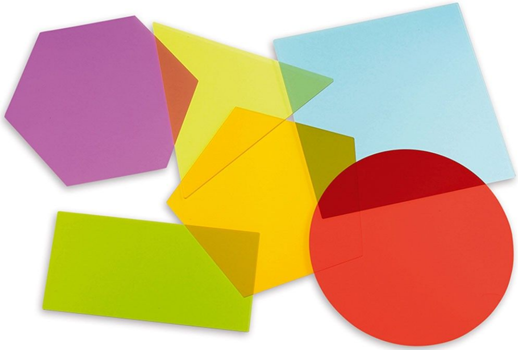


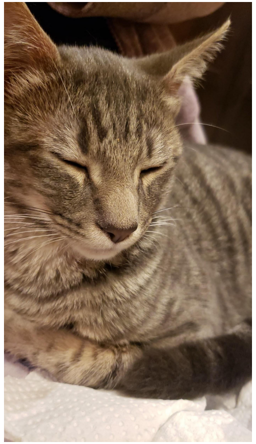
What can computers do easily?

- ➔ Looking for pixels/regions of a particular color (they would be great at "I Spy")
- ➔ Finding areas where everything is the same color
- ➔ Distinguishing shapes when there is a very clear background



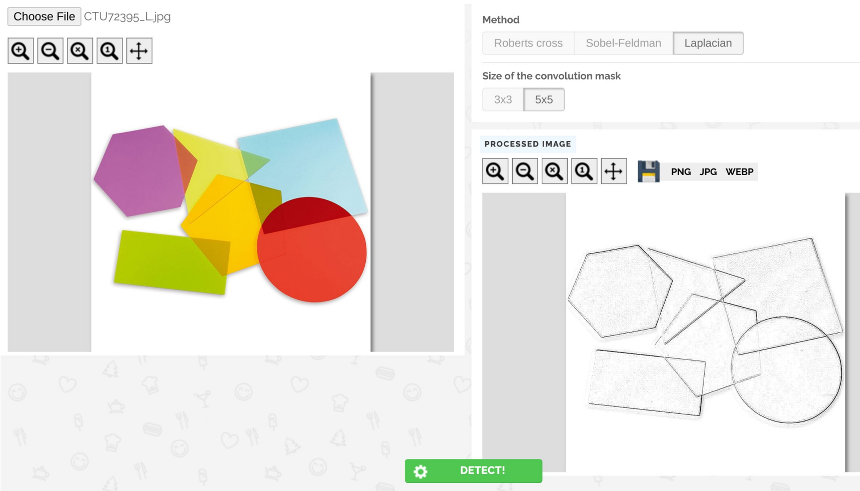
Unfortunately real images are messier

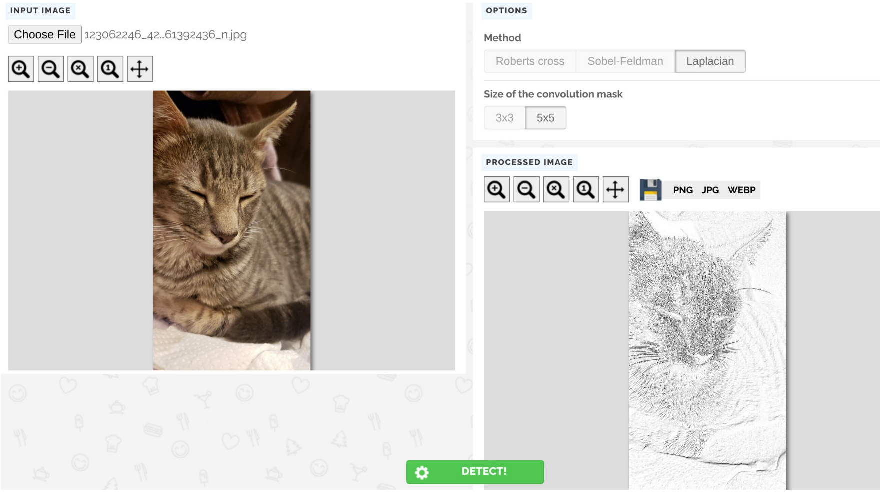




ikea.com

Edge Detection



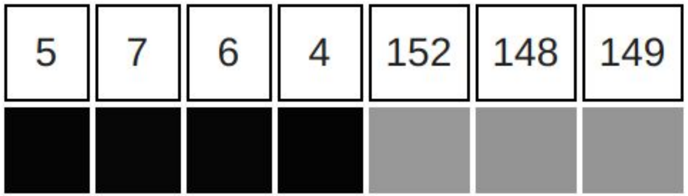


https://pinetools.com/image-edge-detection

How does edge detection work?

- ➔ Generally based on sudden changes in intensity

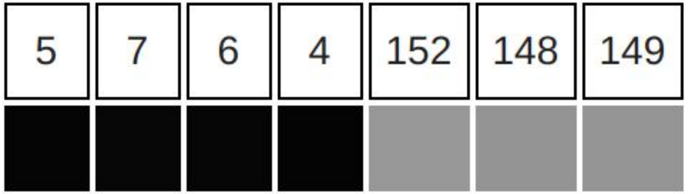
Between which pixels would you draw an edge in this “one dimensional image”?



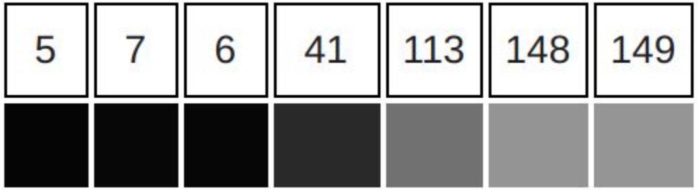
How does edge detection work?

- ➔ Generally based on sudden changes in intensity

Between which pixels would you draw an edge in this “one dimensional image”?



What about this one?



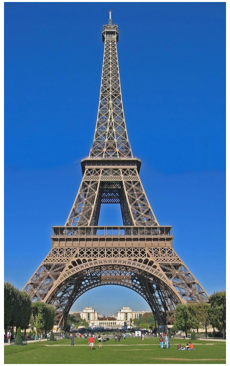
Canny Edge Detection

1. **Apply smoothing to the image -** move each pixel's intensity and color a little closer to that of its surrounding pixels
2. **Find intensity gradients -** pixels which are very different from its neighbors in particular directions
3. **Non-maximum suppression -** filter out pixels with gradients which are similar to other pixels near them, even if they are very high
4. **Thresholding and edge tracing -** identify contiguous segments of high gradients corresponding to edge

(Wikipedia)

Instance Recognition

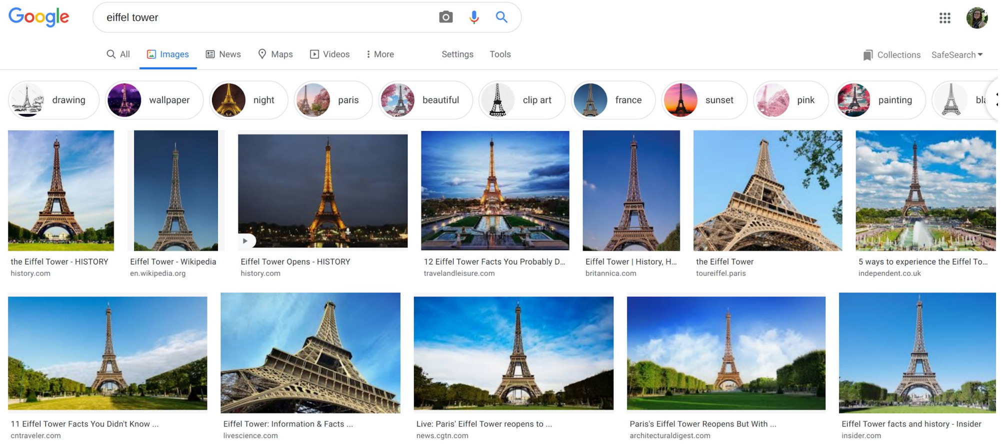
Eiffel Tower



Given an image of a particular object, we want to find other images which contain that object (and possibly other things).

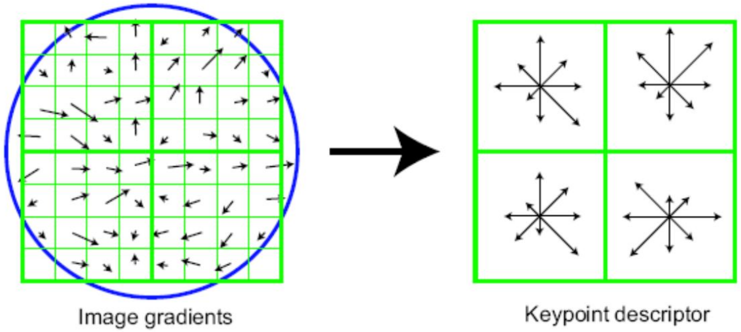
Difficulties:

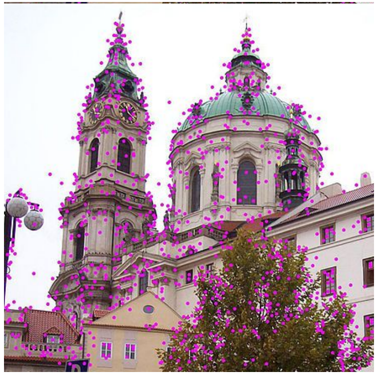
- ➔ Differences in background
- ➔ Rotation
- ➔ Scaling
- ➔ Lighting



Feature Detection

➔ SIFT (scale invariant feature transform) takes an image and marks keypoints which describe the image in a way which is robust to rotation and scaling
➔ Each pixel is represented as a histogram of gradients in the neighborhood around it
➔ Pixels with extreme values compared to surrounding ones are maintained

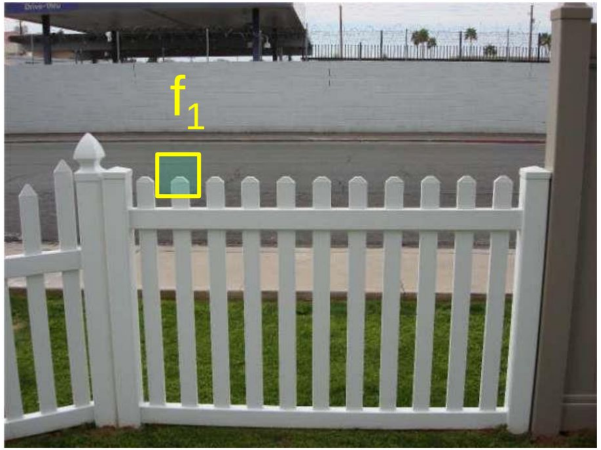




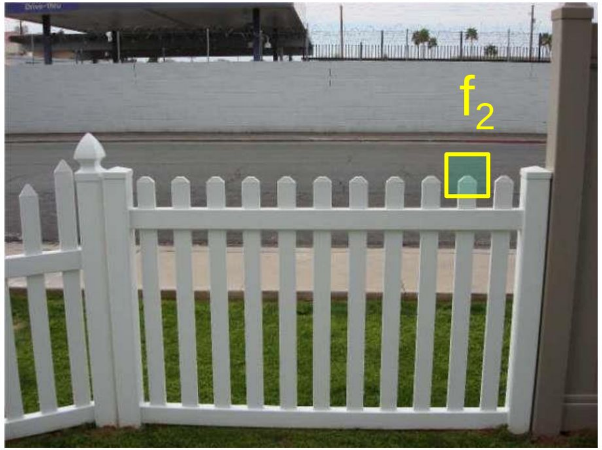
Comparing Features Across Images

Sometimes we want to know if a point in one image is similar to a point in another image

Here, F_1 and F_2 have very similar context in their images, so we expect their SIFT features to be highly similar



I_1



I_2

Comparing Features Across Images

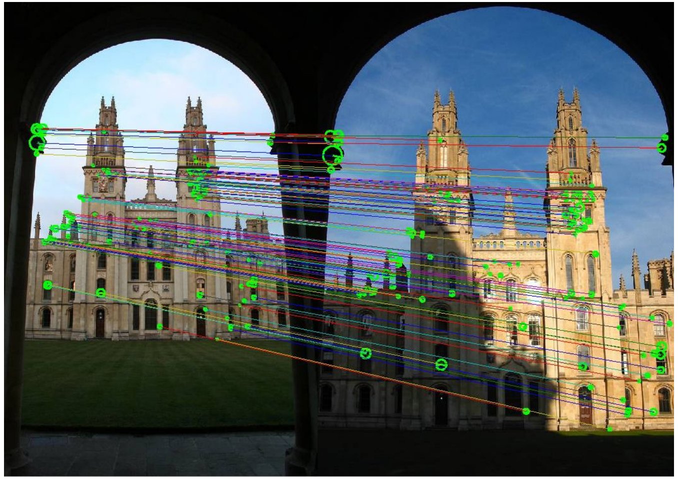
➔ To avoid matching up regions which have many possible matches, we can look at the second best match and see how much better the best match is

I_1 I_2

Goal: Match up features that are best matches

- ➔ Now we have a method to compare individual features between images, but what we really want is a mapping from features in one image to features in another
- ➔ There may be cases where two features in image one both have the same “best match” in image two

How would you find the best way to match up the features?

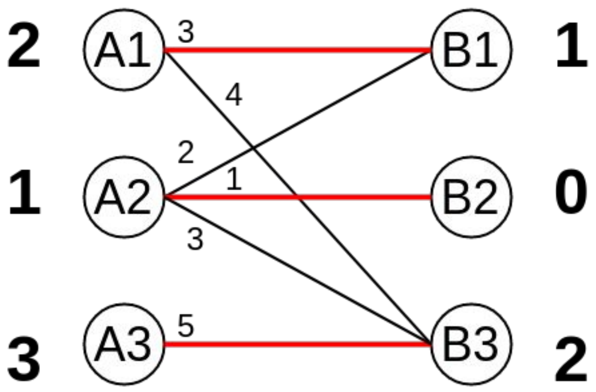


Maximal Matching Problem

➔ A bipartite graph is a graph in which the vertices can be partitioned into two groups where all edges go from one group to the other and are not within a group

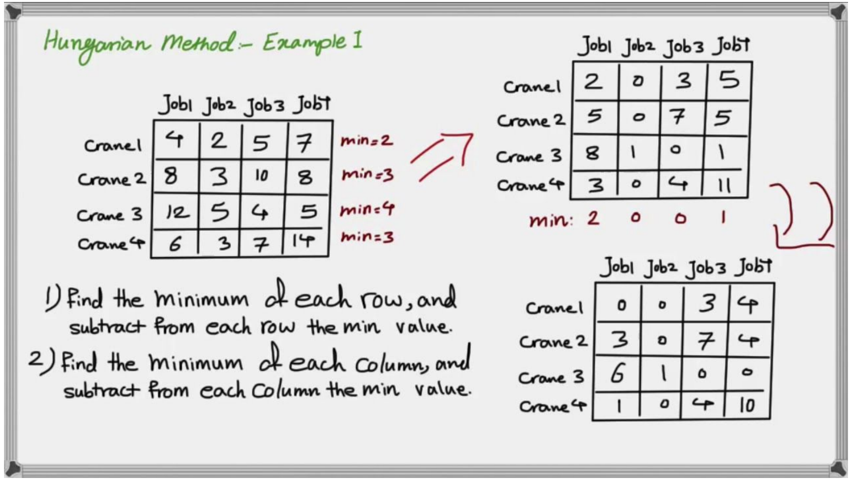
➔ The maximal weighted matching problem is where you are given a weighted bipartite graphs and want to match nodes in one group to nodes in the other group that they have edges to

➔ In doing so, we want to maximize the sum of the weights of the edges we choose



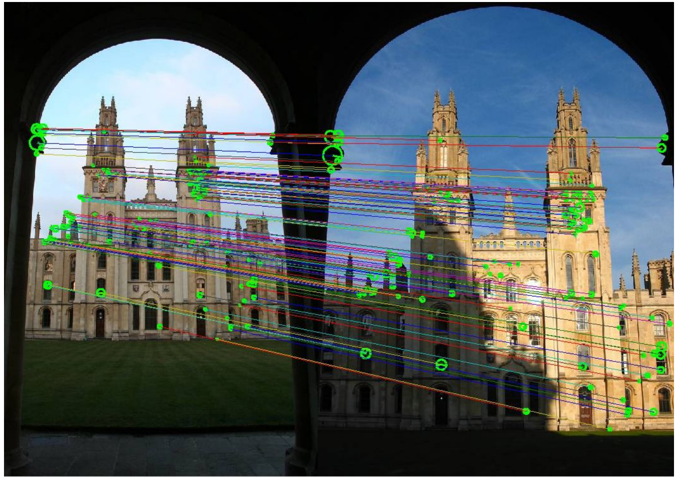
Hungarian Algorithm

- ➔ Solves the maximum weighted matching problem n $O(n^3)$, where n is the number of nodes on each side of the graph
- ➔ It works by forming a table of edge weights and iteratively eliminating the smallest weights in each row and column
- ➔ Eventually the only edges left will be the ones that are part of the maximal matching



Back to Feature Mapping

- ➔ We can set up a bipartite graph where the two sides are features in each image
- ➔ Edges are weighted according to how similar the SIFT features are
- ➔ The Hungarian algorithm gives us a way to match the features and maximize feature similarity

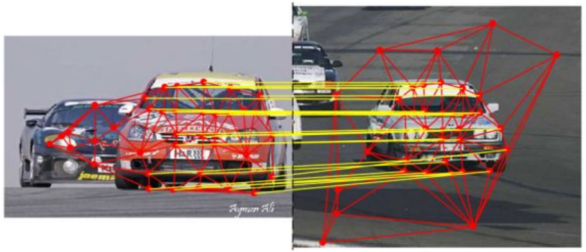


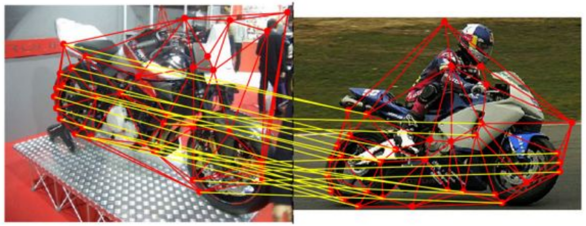
Face Recognition

- ➔ One application of the instance recognition problem - used particularly for security
- ➔ Use a mixture of graph-based techniques and neural networks to identify instances of a specific face
- ➔ These algorithms can recognize faces much better than humans can (and store large databases of known faces)

Incorporating geometric/spatial information

- Usually we expect the difference two images of the same thing to be geometrically consistent - the features should make up a similar shape even if that shape gets turned or squished or moved around
- Algorithms such as RANSAC have been developed to leverage this and instead of finding a matching directly looks for a spatial transformation which matches up the points

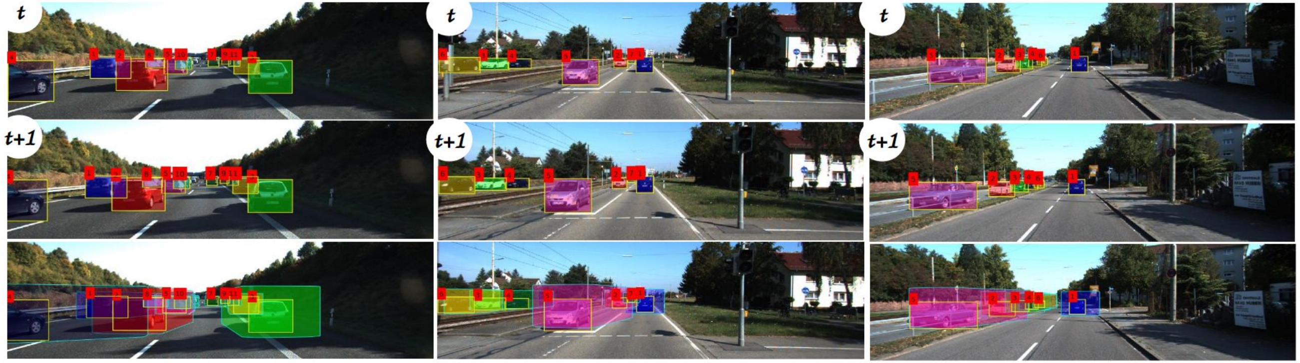




Other Graph Applications

- ➔ Image segmentation (right)
- ➔ Object tracking (below)
- ➔ Machine learning for image classification
- ➔ And more!

(https://paperswithcode.com/task/semantic-segmentation)



(https://paperswithcode.com/task/multi-object-tracking)

Conclusions

- ➔ Computer vision is the field of developing algorithms which discern meaning from images
- ➔ In some areas computer vision lags behind what the human brain can do, while in others the improved precision and consistency of computers makes them better than even trained professionals
- ➔ Graph algorithms are one of many ways of comparing or segmenting images