UNIVERSITÀ
DEGLI STUDI
FIRENZE

Final-term programming assignments

Parallel Programming

What to do

- It’s possible to implement the assignments in pairs

- 6 credits course: implement a programming assignment using one of the approaches presented (parallelization/ vectorization). Implement a sequential and a parallel version.

- 9 credits course (or 6 credits with pair): implement a programming assignment using one of the approaches presented. Implement a sequential and a parallel version.

- Write a tech report with performance analysis (speedup) and prepare a presentation

- Keep everything on a public Github/Bitbucket/Gitlab

Image reader

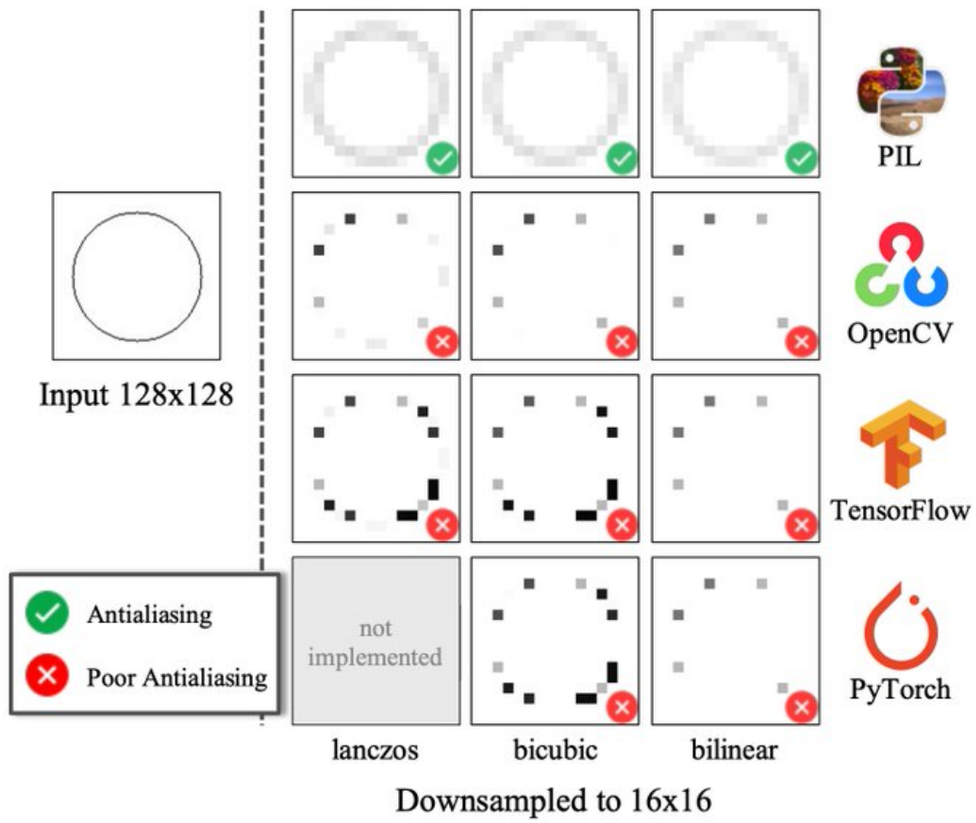
- Multithread JPEG image reader
  - read X images from a directory
  - Keep the uncompressed bitmaps (or the compressed stream) in an appropriate data structure, that manages access to the images
  - OpenMP does not allow to implement a non-blocking version since it requires asynchronous multi-threading (ie.. CUDA allows it).

Image resizer

- Implement a fast image scaling that is robust to image-scaling attacks (see https://scaling-attacks.net)

- Scaling is a basic functionality in computer vision systems based on Convolutional Neural Networks

- Can use, OpenMP, vectorization and CUDA.



Kernel image processing

- Image processing
- CUDA is a very good candidate
- See https://en.wikipedia.org/wiki/Kernel_(image_processing)

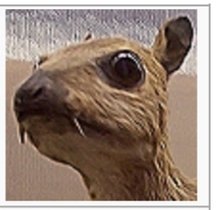
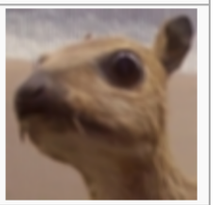
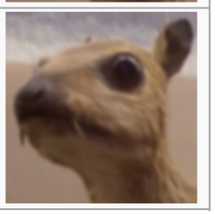
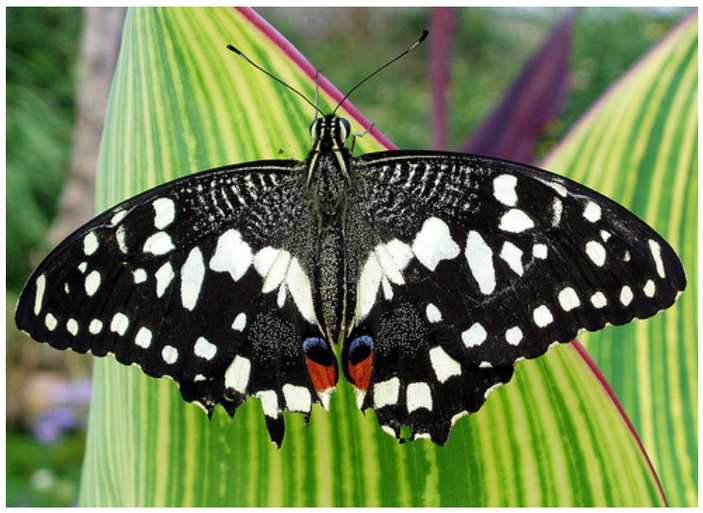
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
Gaussian blur (approximation)	$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	

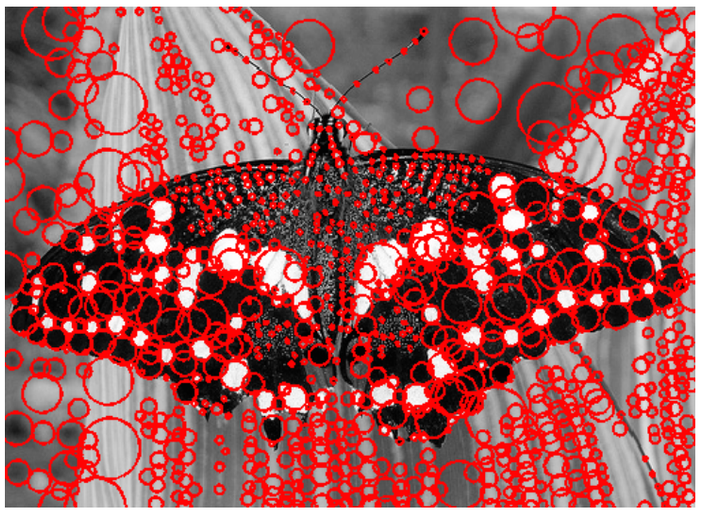
Image composition

- Implement a parallel alpha composition of images
- Useful to create image augmentation to train Convolutional Neural Networks, e.g. for detection (see Blend augmentation in ImgAug library)
- Can parallelize applying the same object to different images, or in different positions in the same image

Scale space blob detection

- Detect image blobs using a scale space approach to recognize them at different scales
  - It's possible to use Lowe's algorithm (proposed in the SIFT descriptor), based on differences of Gaussians (DoG)

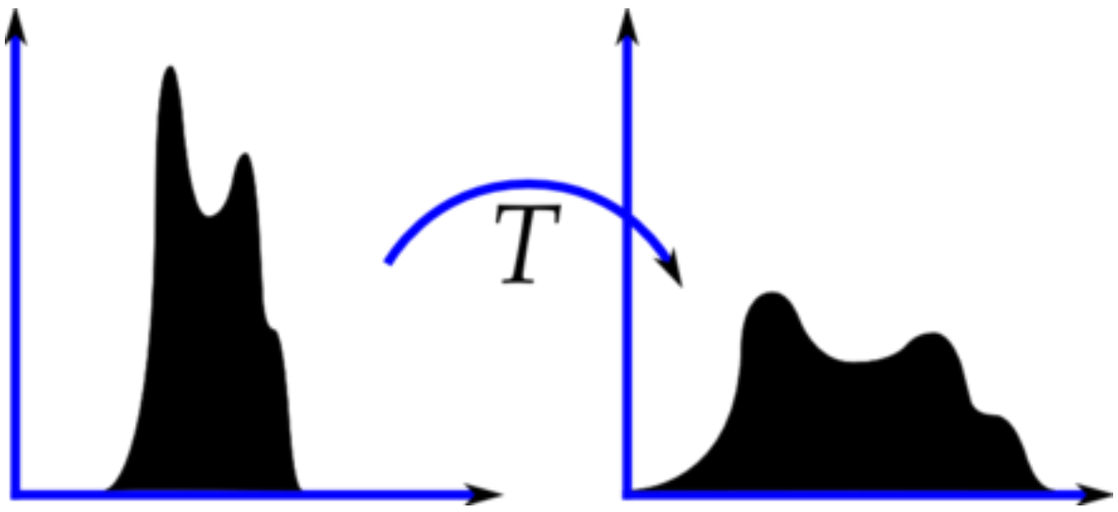




Histogram equalization

- Histogram equalization is a digital image processing method by which you can calibrate the contrast using the image histogram.

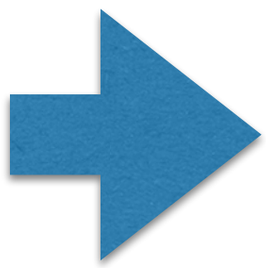
- This method usually increases the overall contrast of many images, especially when the usable image data is represented by very close contrast values. Through this adaptation, the intensities can be better distributed on the histogram.



Integral image

- An integral image is obtained accumulating the values of the pixels along x and y axis.
- Eg. The value of the pixel at (2,1) coordinates is computed from the sum of the pixels at (0,0), (0,1), (1,2) coordinates …
- It' often used to compute visual features with a reduced computational cost (eg. Mean values of parts of the image, used in Viola&Jones detector or also in more modern CNN-based features based on NetVLAD).

5	4	3	8	3
3	9	1	2	6
9	6	0	5	7
7	3	6	5	9
1	2	2	8	3



5	9	12	20	23
8	21	25	35	44
17	36	40	55	71
24	46	56	76	101
25	49	61	89	117

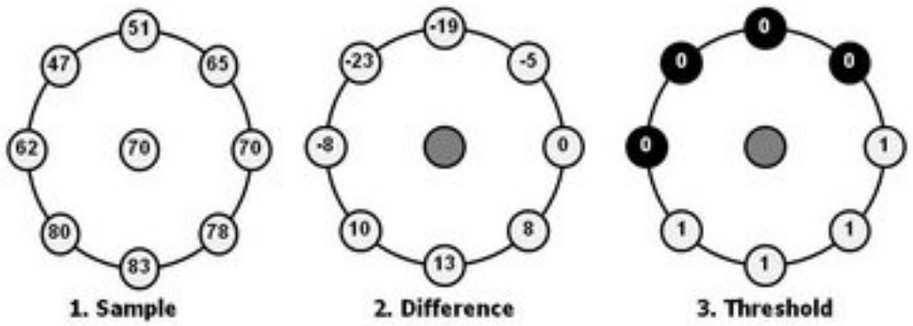
LBP Descriptor

- Local Binary Pattern (LBP) is a simple and powerful texture descriptor. It assigns to each pixel a binary value which is obtained by comparing the value of the surrounding pixels and binarizing the difference with respect to the pixel itself.

- Info: http://www.scholarpedia.org/article/Local_Binary_Patterns

The value of the LBP code of a pixel (x_c, y_c) is given by:

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c)2^p \qquad s(x) = \begin{cases} 1, if\ x \geq 0; \\ 0, otherwise. \end{cases}$$



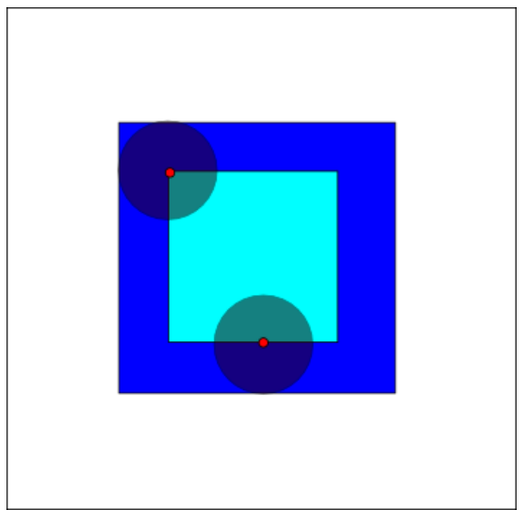
1. Sample 2. Difference 3. Threshold

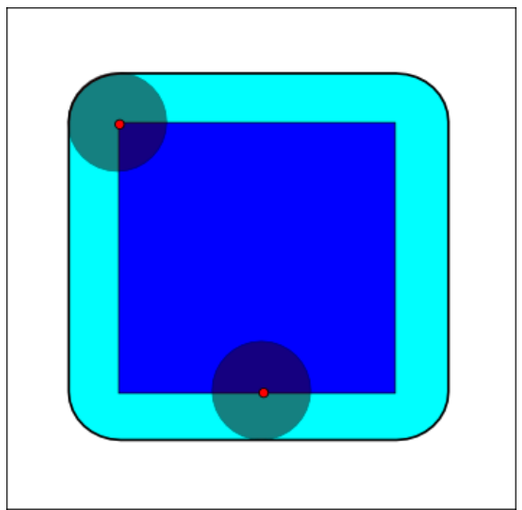
1*1 + 1*2 + 1*4 + 1*8 + 0*16 + 0*32 + 0*64 + 0*128 = 15

4. Multiply by powers of two and sum

Morphological image processing

- Image processing
  - CUDA is a good candidate
  - See https://en.wikipedia.org/wiki/Mathematical_morphology





Bigrams / trigrams

- Compute histograms of bigrams/trigrams on texts (eg. Wikipedia / Gutenberg project documents)
  - Consider bigrams/trigrams of characters and words
  - If needed re-use more and more times the collected data to create a large enough corpus

Password decryption

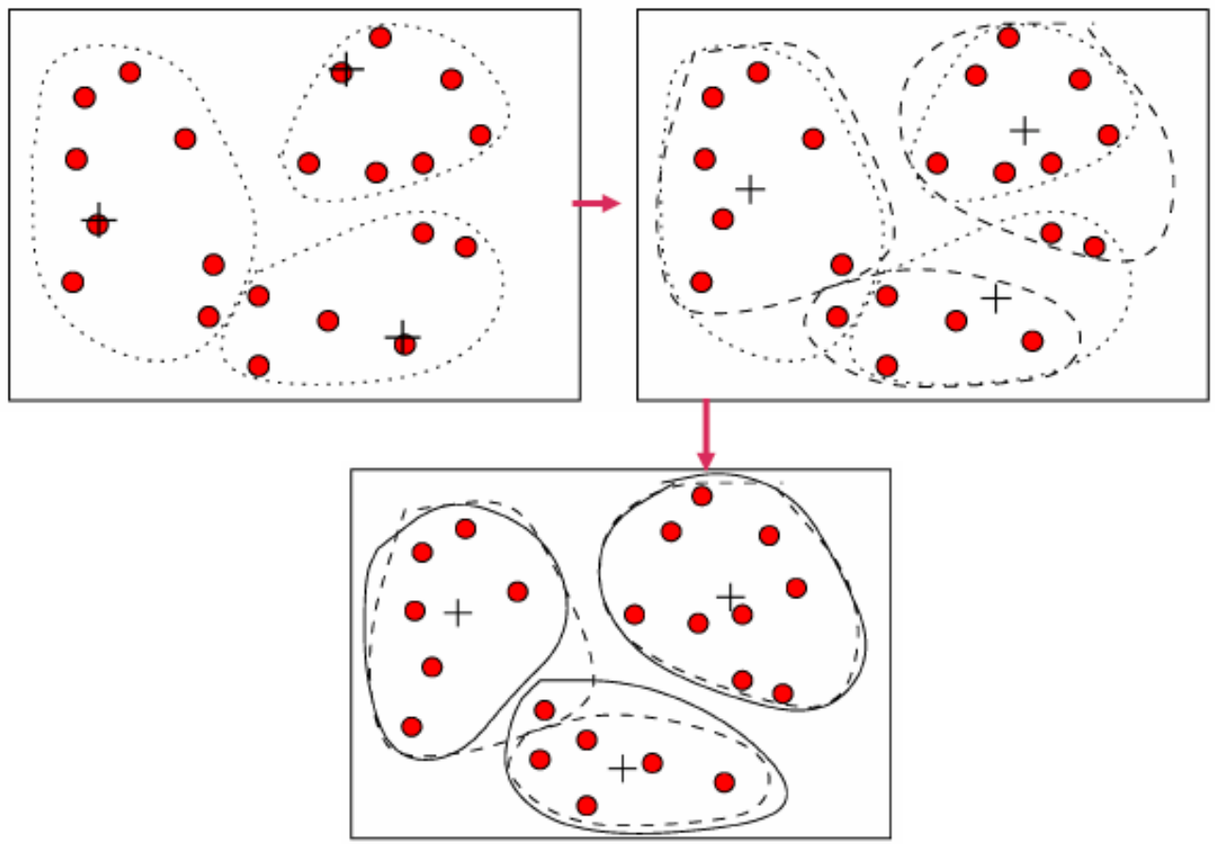
- Decrypt passwords encrypted using crypt (DES)
  - Consider 8 characters lengths in the set [a-zA-Z0-9./]
  - CUDA does not provide the crypt C-library function. Must use a special implementation (check Moodle)
  - It’s a (slowed down) search problem.
  - Can choose to attack a specific password in a list (in different positions), or decrypt a full list of passwords (in this case reduce the search space)
  - Can consider only dates or common 8-chars passwords

Pattern recognition

- Search a given time series within a larger and longer set of time series
  - Get time series from existing machine learning datasets
- Use SAD as the metric to evaluate the match

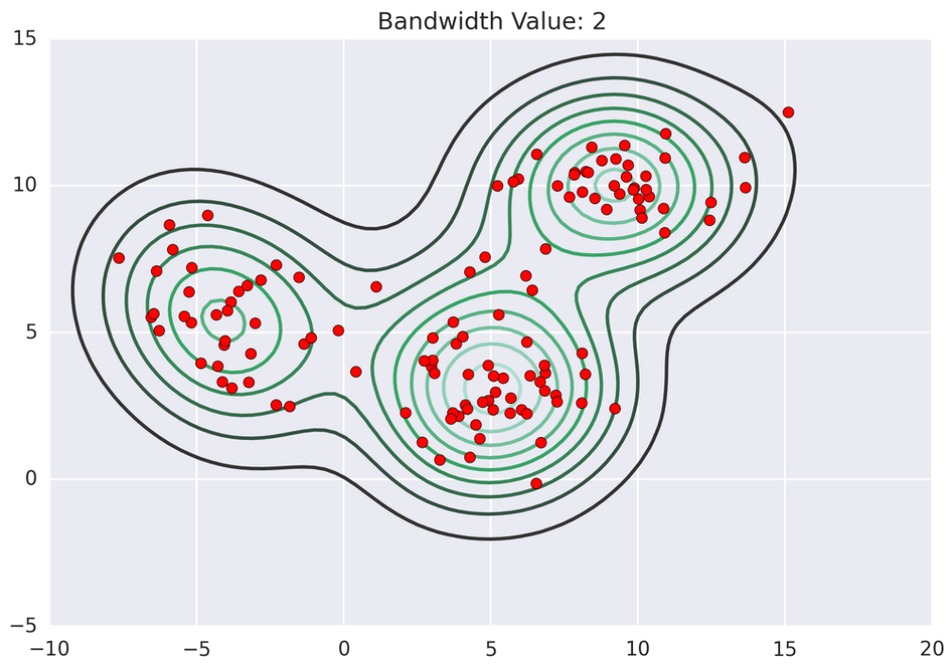
k-means

- Implement a parallel version of k-means clustering
  - There’s no need to implement K-means++ centroid assignments
  - Feel free to chose to operate on 2D or larger dimensionality vectors
- **Esempio**



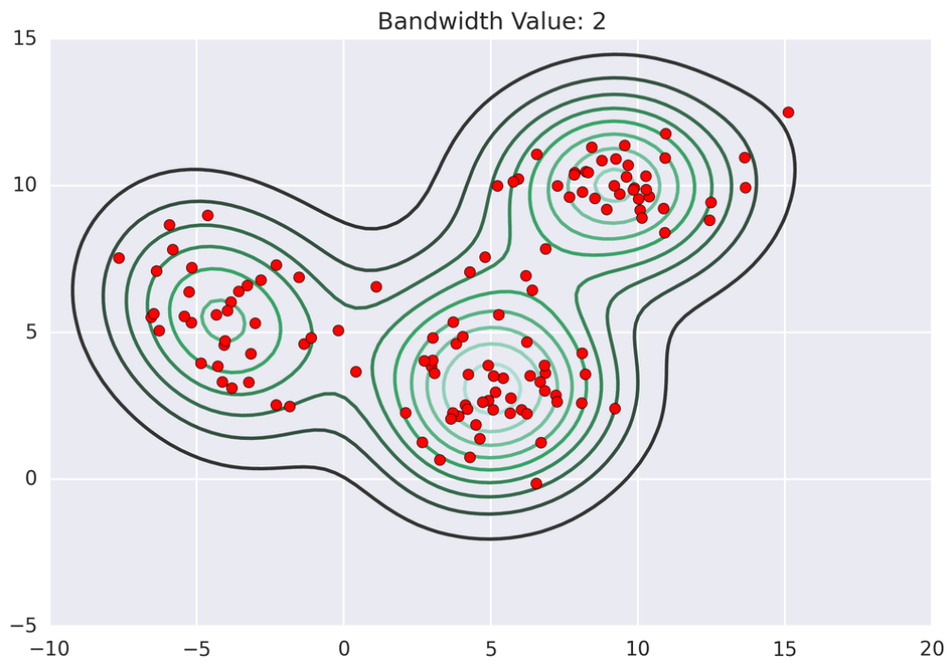
Mean shift clustering

- Implement a parallel version of mean-shift clustering. This clustering doesn't require to specify the number of desired clusters (it uses KDE, so needs a bandwidth parameter) but its complexity is $O(N^2)$
- Can be used to segment images



Mean shift clustering

- Implement a parallel version of mean-shift clustering. This clustering doesn't require to specify the number of desired clusters (it uses KDE, so needs a bandwidth parameter) but its complexity is $O(N^2)$
- Can be used to segment images

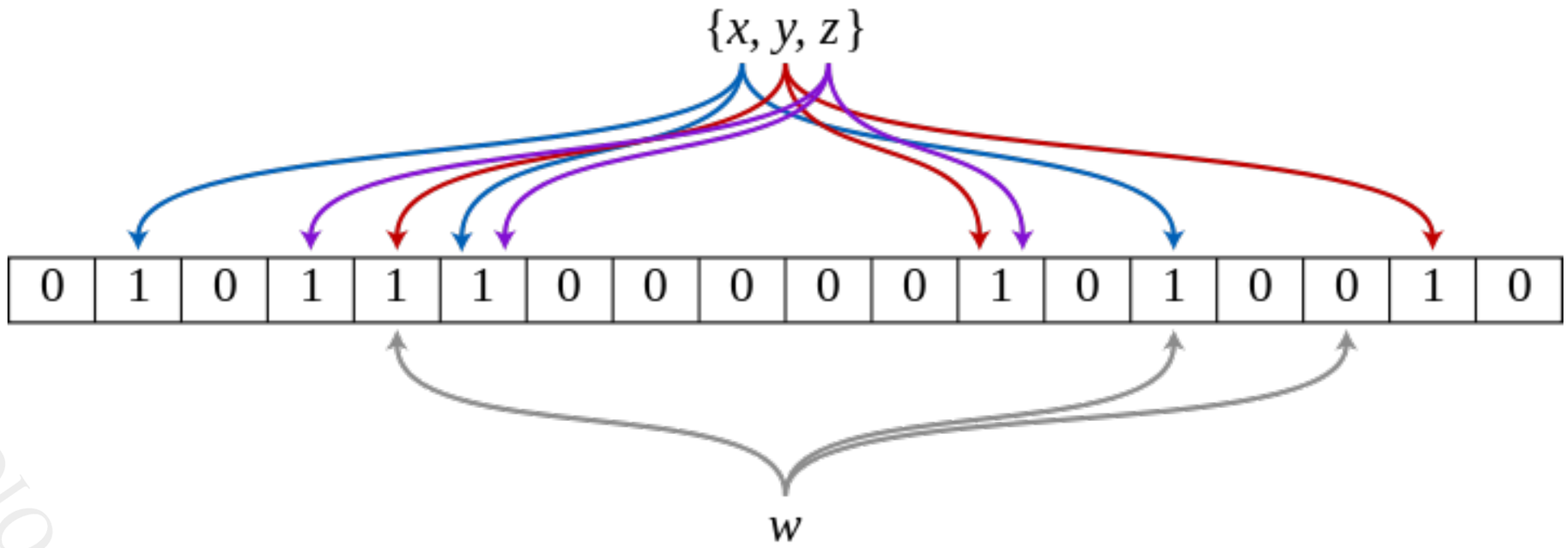


ANN retrieval

- Consider high dimensionality vectors (eg. 128)
- Given a specified query find the *k nearest neighbors* in a database of vectors
- Can use LSH for approximate indices, eg. using mlpack o LSHkit
- Can get sample datasets from http://corpus-texmex.irisa.fr

Bloom filter

- The Bloom filter is a probabilistic and memory efficient structure that is used to test whether an element is part of a set or not.
- If the filter says no, it is sure that the element is not part of it ("definitely not in set"), in the case of a yes this may not actually be part of it ("possibly in set")



String search

- Implement a parallel edit distance (Levenshtein) or Bitap (approximate search, used in agrep)

- See https://en.wikipedia.org/wiki/Levenshtein_distance and https://en.wikipedia.org/wiki/Bitap_algorithm

Video object detection

- Use Python to parallelize object detection in multiple video sources
- Use Ultralytics for YOLO object detection
- Parameterize the video inputs (e.g. videos or cameras)
- Allow to use different networks, select the classes of the objects

Video sequences alignment

- Use Python/OpenMP to align a short video sequence within a larger video sequence, i.e. the short video sequence may be an altered version of the longer video sequence
  - Use SSIM quality metric or ANN features (e.g. ResNet18) to evaluate the similarity of two frames
- Access video frames using OpenCV or ffmpegcv or PyAV to read the video frames

Image stitching

- Implement image stitching to create panoramic images.

- Use OpenCV (Python or C++ API), parallelize the process (feature detection/matching and following homography estimation, warping and blending)

- Use OpenMP or Python