Laboratory 5: Panoramic Images

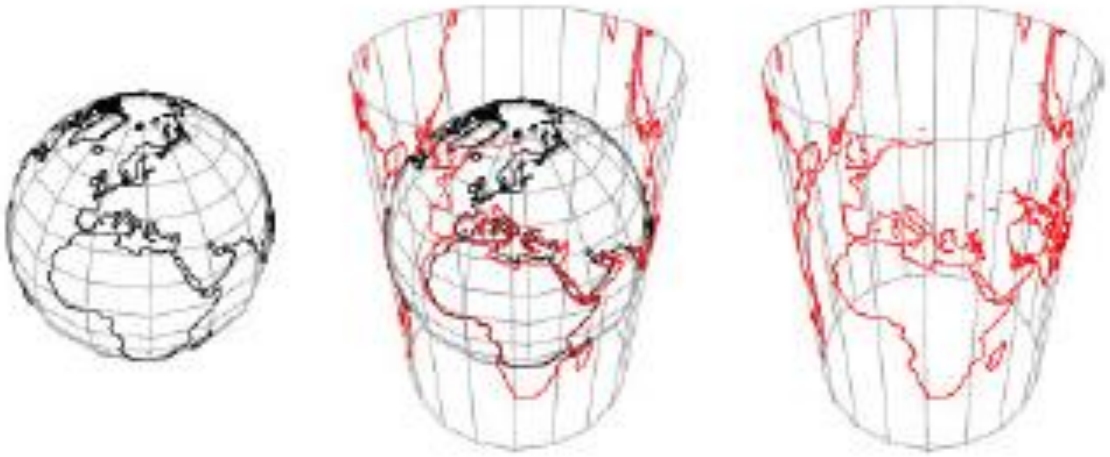
Computer Vision 2018

P.Zanuttigh

Panoramic Images

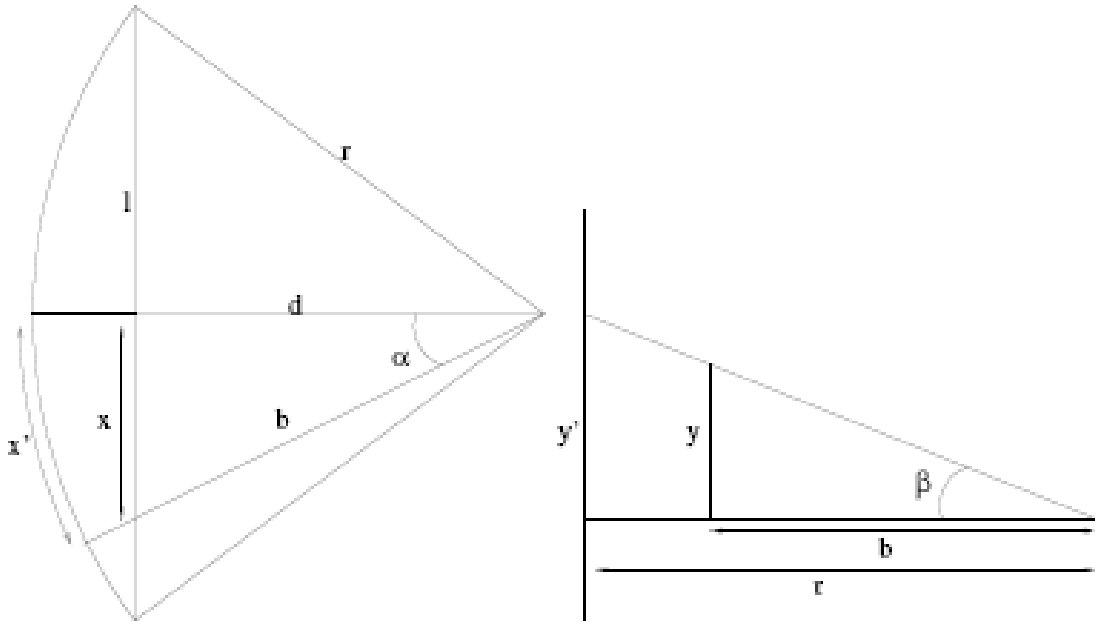
- Pictures covering a 360° field of view
- The images are built from a set of pictures taken from a single viewpoint

Cylindrical Projection (1)



- The photos are projected on a cylinder
- After the cylindrical mapping the transformation between the various pictures is simply *a translation*
- See the file '*cylindrical_projection.pdf*' for the theory and equations

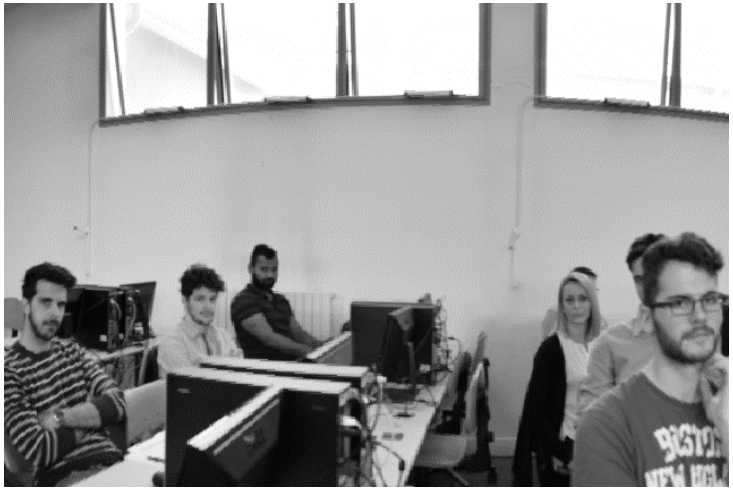
Cylindrical Projection (2)



$$x = d\ tan(\alpha) = d\ tan(\frac{x'}{r})$$

$$y = y' \frac{d}{r} \frac{1}{cos(\frac{x'}{r})}$$

Example (Projection)



Algorithm to be Developed

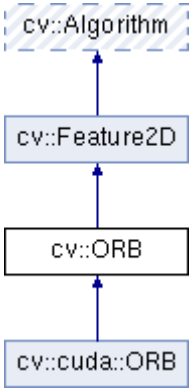
1. Project the images on the cylinder
2. Extract ORB or SIFT descriptors
3. Find the matching features
4. Estimate the translation between couples of adjacent images starting from the SIFT matches using a robust estimator (RANSAC)
5. Build the panoramic image
6. Visualize it !

OpenCV Feature 2D

virtual void cv::Feature2D::detectAndCompute	(	InputArray	image,	Input image
		InputArray	mask,	Compute KP only in regions where mask !=0
		std::vector< KeyPoint > &	keypoints,	Output keypoints (location, orientation, scale)
		OutputArray	descriptors,	KP descriptors
		bool	useProvidedKeypoints = false	
	)			

- Base class for feature extractor and descriptors
- detect (feature extraction), compute (feature description) and detectAndCompute (both stages) methods
- Constructor depends on the employed subclass

ORB in OpenCV



static Ptr<ORB> cv::ORB::create	(	int	nfeatures = 500,	Max # of features to extract
		float	scaleFactor = 1.2f,	Scale step between different pyramid levels
		int	nlevels = 8,	# of pyramid levels (multi-scale)
		int	edgeThreshold = 31,	Avoid computing features close to edges
		int	firstLevel = 0,	Set to 0
		int	WTA_K = 2,	2 for comparison between couples of points as in the theory
		int	scoreType = ORB::HARRIS_SCORE,	Rank extracted corners with Harris criteria
		int	patchSize = 31,	Size of patch for feature computation
		int	fastThreshold = 20	Threshold in FAST algorithm
	)			

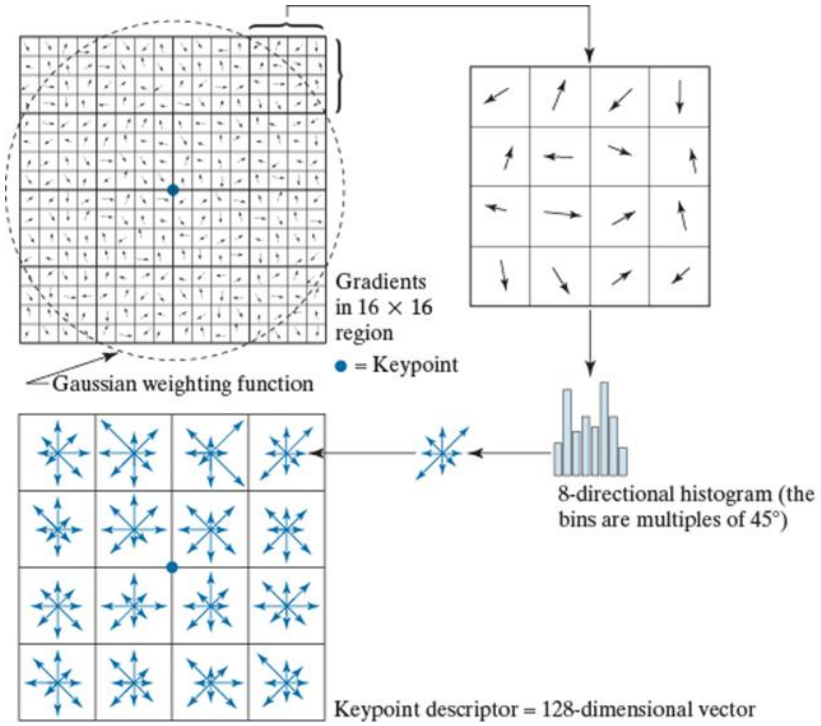
Scale Invariant Feature Transform (Lowe 2004)

4 Steps :

1. Scale-space extrema detection
2. Keypoint localization
3. Orientation assignment
4. Keypoint descriptor

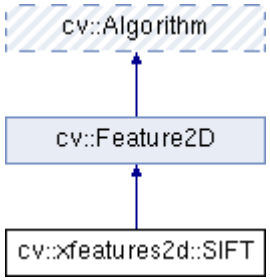
3 and 4 will be detailed on Monday

Descriptor Computation: Synthesis



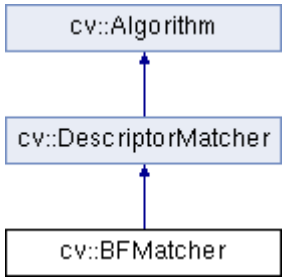
Details in the next lecture, for the moment need to know that:
the descriptor is a 128-dimensional vector computed on the basis of the gradient modules and orientations in a 16x16 window surrounding the keypoint

SIFT in OpenCV



static Ptr<SIFT> cv::xfeatures2d::SIFT ::create	(int	nfeatures = 0,	# of feature points to extract
	int	nOctaveLayers = 3,	# of ocatves in each layer
	double	contrastThreshold = 0.04,	Threshold on $D(\hat{x})$
	double	edgeThreshold = 10,	Threshold on eigenvalue ratio
	double	sigma = 1.6	Smoothing of the first octave
	)		

Feature Matching




```
static Ptr<BFMatcher>
cv::BFMatcher::create      (     int       normType = NORM_L2,
                                 bool      crossCheck = false
                           )
```

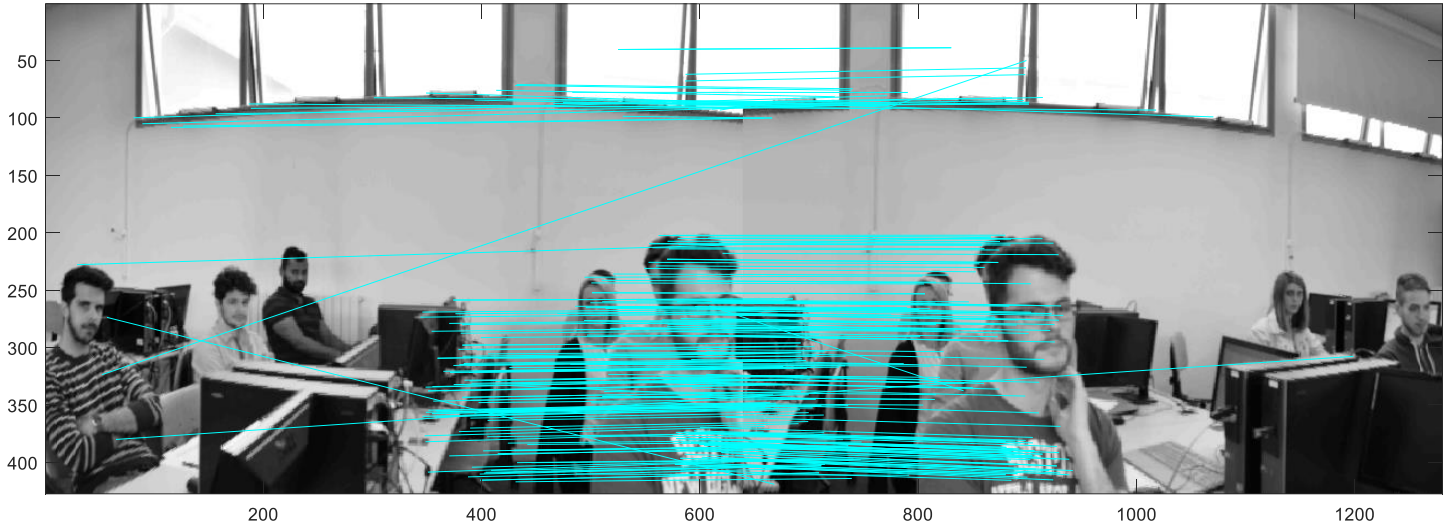
- Brute-Force matching
- Type of distance function
  - Use NORM_L2 for SIFT and NORM_HAMMING for ORB
- Cross check: if A match B then B should match A
- *(optional)* A threshold on the matching distance can be used to discard weak matches
  - Ratio w.r.t. min_distance

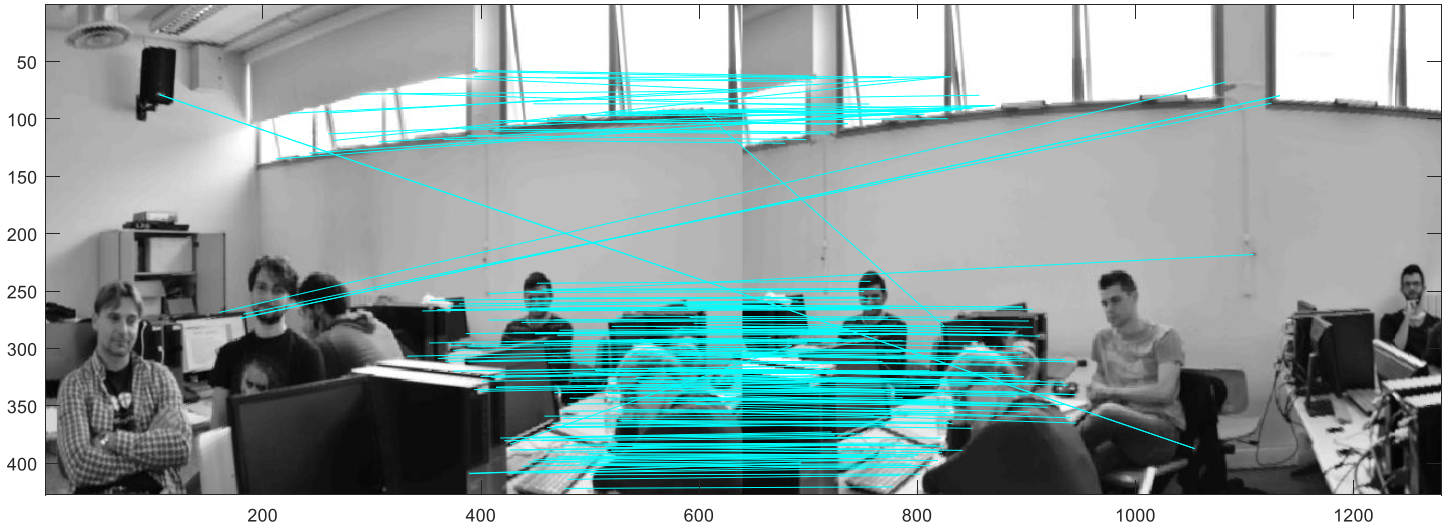
RANSAC (Simplified)

1. Select a random correspondence and get the corresponding shift $(\Delta x, \Delta y)$
2. Count how many correspondences are consistent with the selected one. The criteria can be a threshold on the difference from the estimated shift: e.g., $(|\Delta x_n - \Delta x| + |\Delta y_n - \Delta y|) < 5$
3. Iterate *k* times (ex. *k*=50) and keep the correspondence with the largest compatible set
4. Compute the average $(\overline{\Delta x}, \overline{\Delta y})$ using only the compatible points

Implement the algorithm manually or use cv::findHomography() with method=RANSAC

Example (Matching)





Examples (Panoramic Image)

Matlab+Lowe's toolbox

c++ - OpenCV - SIFT

c++ - OpenCV - ORB