Autonomous Blimp								
Mapping Generate map data and stores the map data in a file - [(Point3,descriptor)]							Localization Generate Estimated Trajectory - [Pose3]		Robot Communication Send Control Commands to Robot
Final Result Verification						Decide what kind of descriptor to store in the map data.	Basic Pipeline Development	Result Verification	Hack IR controller
Verifications(done)			Close Loop Verification Reconstruct a small square room(4*4*3) with row*col*angle(2*2*8) images (Fail)			Plan: Use Real Data to analysis the magnitude and direction of high dimensional vectors	Plan: Localize by matching in the map (I have started to work on this part and can be done within a week since most functions have been developed before)	Plan:	Plan:

Tasks	Results	Comments	**The result is totally wrong. Therefore, I add gtsam.marginals(graph, sfm_result) and the exception is: *Indeterminant linear system detected while working near variable***
			Possible Problems:
			Bad matches, image distortion, noises in source image
						Find a visualable method to analysis the magnitude and direction of high dimensional vectors	a. Create methods to save and load the mapping result.
							This includes storing the normalized averaged descriptor values in the map.
							b. Create class TrajectoryEstimator based on unittest
							c. Create class VideoStreamer to handle both image set input and video stream input
							d. Create script to execute.
							
							Pipeline:
							Create instances for TrajectoryEstimator and VideoStreamer
							update_trajectory()
							{
							trajectory = [initial pose]
							next_frame()
							{
							get_pose_from_trajectory()
							superpoint_extraction()
							landmark_projection()
							landmark_association()
							pose_estimate()
							plot()
							trajectory.append(new pose)
							}
							}
								Use the same input images used during mapping process and recover the mapping pose results	https://www.alanzucconi.com/2015/08/19/how-to-hack-any-ir-remote-controller/?fbclid=IwAR187aUq2ciNrClwFw-Ca92bjlD0h0S6kcmWqoZUciH5PJZwZ6AkY3SHJR4

Straght line reconstruction Verification	In the result, some aligned points can be reconstructed to straght lines while some can not.	This can be caused by the distortion at image border or the noises in the source image	Test Cases	Executable Plans				Use images at measured poses as input and recover measured poses	
Calibration Matrix Correctness Verification	the reprojection error is small	Calibration matrix is correct. However the undistort image is still distorted at the border. And some of the lines in the image are not very straight.	Test to see if the problem is caused by input image pairs that do not have feature matches.	Reconstruct perpendicular three walls of a small square room(4*4*3) with row*col*angle(1*2*3) images. Not all images pairs have matched features.					
Faces Perpendicularity Verification	the perpendicularity can be recover	Both manually selected features and automatically matched features can be used to recover perpendicular faces.	Test to see if distortion is the problem	Resize source images by cutting the source image border. Use the resized images to test on *Klaus* *Reconstructing with 6 Aligned and well measured Poses that face the same direction*					
Klaus Reconstruction with 6 aligned and well measured Poses (These poses face nearly the same direction and there are lots of landmarks observed by the camera at all 6 poses.)	3D structure can be recover	This prove that the current pipeline can work on: 1. Reconstructing 3D structure that are far from the camera. 2. Reconstructing 3D structure that has lots of good matches	Test with source images that do not have noise	Use phone camera to collect source image. Reconstruct the small square room(4*4*4) again with row*col*angle(2*2*8) images					
Back projection Depth parameter influence		The Back Projection Depth parameter has bigger influence when reconstructing objects that are close to the camera		Store key point color and plot landmark with color, To help verify the correctness of the result.			Plan: Localize by tracking		
Feature Matching method Comparation	4d agri data associate+cv. FindEssentail Matrix(currently used in the pipeline) > 4d agri data associate > Superpoint 2 way nn match+cv. FindEssentail Matrix						Use matched features to estimate the camera motion(row,pitch,yaw,x,y,z,scale).		