

Algorithme	Principe	Fonction Opencv	Performance
Detection d'obstacle	- cet algorithme est basé sur la detection de contours		dans cette partie on traite les performances des algorithmes
	Les différentes algorithmes de detection de contours		Computing Time The number of feature points The number of matching points (left/right) False match rate
	<p>Principe = les coins sont associées au maximums de la fonction</p> <p>2. Spatial derivative calculation</p> <p>3. Structure tensor setup</p> <p>4. Harris response calculation</p> <p>5. Non-maximum suppression</p> <p>Harris Corner Detector</p> <p>https://drive.google.com/open?id=1Bzz2PqGdNqnrhQIUjxnXLD</p>	<p>cv2.cornerHarris()</p> <p>img - Input image, it should be grayscale and float32 type.</p> <p>blockSize - It is the size of neighbourhood considered for corner detection</p> <p>ksize - Aperture parameter of Sobel derivative used.</p> <p>k - Harris detector free parameter in the equation.</p> <p>Dans une approche d'améliorer la précision on utilise la commande cv2.cornerSubPix()</p>	<p>Matching is used to describe things which are of the same colour or design dans les conditions statique</p> <p>10.556951 (indoor image)</p> <p>8.844036 (outdoor image)</p> <p>Il calcule le nombre de point d'intérêt dans une image ! (on utilise deux images pour une caméra stéréo) dans les condition statiques</p> <p>800/800 (indoor image)</p> <p>413/413(indoor image)</p> <p>305/305 (outdoor image)</p> <p>2.2%(9/413)(indoor image)</p> <p>2.3%(7/305) (outdoor image)</p>
Shi-Tomasi Corner Detector	<p>utilise la même matrice que la méthode Harris mais en calculant à la fin le minimum des valeurs propres de cette dernière min (a1,a2)</p> <p>http://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_feature2d/py_shi_tomasi/py_shi_tomasi.html#shi-tomasi</p>		
Not free to use	<p>SIFT (Scale-Invariant Feature Transform)</p> <p>-Transformer une image en ensemble de vecteurs de caractéristiques qui sont invariants par transformations géométriques usuelles (homothétie, rotation) et de manière moins fiables aux transformations affines et à l'illumination</p> <p>https://drive.google.com/open?id=1mcW-mYiUrF96XiB-4FH-yA</p>	<p>sift.detect() : finds the keypoint in the images Each keypoint is a special structure which has</p>	<p>54.328028(indoor image)</p> <p>90.981467 (outside image)</p> <p>2501/2685(indoor image)</p> <p>5529/5729 (outside image)</p> <p>710/710(indoor image)</p> <p>1156/1156</p> <p>0.99%(7/710)(indoor image)</p> <p>0.78%(9/1156)</p>
	<p>SURF (Speeded-Up Robust Features)</p> <p>- Considérer comme un SIFT plus rapide/les descripteur SURF sont obtenues avec un filtre gaussien centré au niveau des points d'intérêts SURF algorithm greatly improves the computational efficiency by the use of box filter and integral image.</p>	<p>surf.detect() : finds the keypoint in the image !</p> <p>surf.compute() : compute the keypoints using a gaussian window the use of those functions is not free (so we have to implement)</p>	<p>18.257477</p> <p>21.924250</p> <p>792/936</p> <p>1341/1125</p> <p>201/201</p> <p>150/150</p> <p>1.0%(2/201)</p> <p>2.0%(3/150)</p>
	<p>FAST (Features From Accelerated Segment Test)</p> <p>L'algorithme fonctionne en 2 étapes :</p> <p>1- test de segment basé sur les luminosités relatives est appliqué</p> <p>2-permet d'affiner et de limiter les résultats par la méthode suppression</p> <p>https://drive.google.com/open?id=17zuD7eZxxSYyOmV5alCyE</p>		