~~To compare the two algorithm, we choose, we decide to establish a list of criteria of acceptance to decide which one provide the best performance for our application.~~

~~Among this criteria, we found :~~

~~The velocity: our system should have real time reactions to provide in the fastest time information to the operator~~

~~The robustness: our system should detect 100% of the tools on the workbench, without any error, and work all the time.~~

~~Comfort of the user : the application should be ergonomic, easy to understand and to use, without a lot of click to access to the principal functionality.~~

~~To test , we decide to implements protocols~~

To compare the performances of SIFT and ORB, we decided to establish a list of criteria of acceptance to decide which one provide the best performance for our use case.

First, our application need real time reactions: the user does not have to wait for the information to be displayed. Furthermore, if the environment scanned change (a tool is added for example) the system must adapt himself and add this tool to his parameters.

Second, our application should be robust. The user need to know the right information without any error or missing data. The application need also never stop to function without the user agreement.

Finally, the user experience is very important and need our focus: the application should be ergonomic, easy to understand and to use, without a lot of action to do to access to the principal functionality.

Therefore, we agreed on these criteria of acceptance:

Real time reaction: the system must adapt himself and display the information in 2 seconds

Robustness: the system must found 100% of the tools, display 100% of the tools needed by the user and have 0% of error

Ergonomics: the user should access to the functionality in less than 3 click.