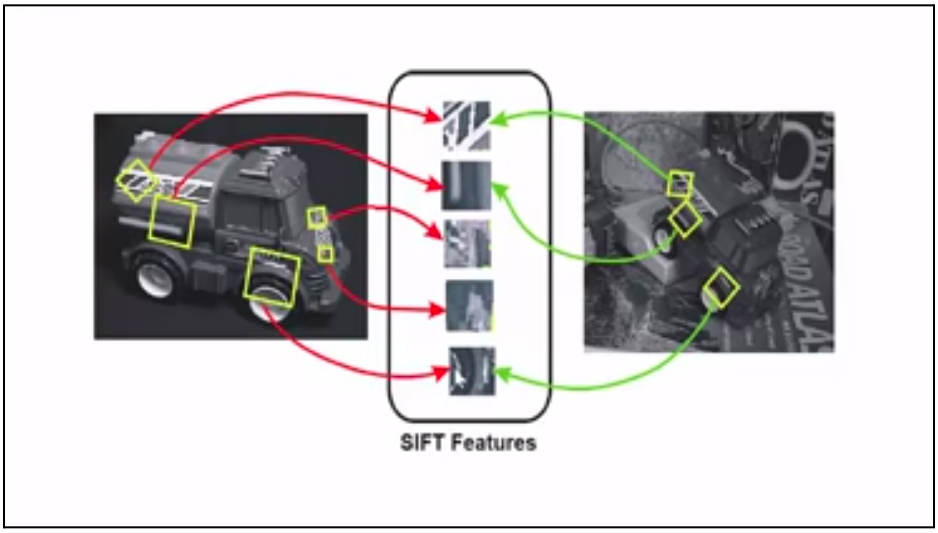
# ***CSEE 4840 Embedded Systems Project Proposal***

# ***qSIFT***

| Name | UNI |
|---|---|
| Madhav Bhat | mb4989 |
| Khushi Gupta | kg3023 |
| Prathamesh Sahasrabudhe | ps3320 |
| Jeffrey Wolberg | jnw2138 |
| Daniel Seligson | ds3824 |

Background:

Scale Invariant Feature Transform (SIFT) is an algorithm that is widely used in image processing. It detects and describes image features in scale and rotation invariant encoding, and therefore can be used to register images taken from different distances and angles. Below is an example of SIFT features from two images of the same object, taken at different scales and angles.



Algorithm Explanation:

There are four main steps of the algorithm, the first one being the most computationally expensive.

1. Scale-space peak selection: Finding the potential location for features.
2. Keypoint Localization: Accurately locating the feature keypoints.
3. Orientation Assignment: Assigning orientation to keypoints.
4. Keypoint descriptor: Describing the keypoints as a high dimensional vector.

Implementation:

*Hardware* – We will implement convolution accelerators for use in the first step of the SIFT algorithm, which involves taking a difference of Gaussian blurs of the image in order to extract the points of interest. We will also be displaying our output on a VGA monitor, so that interface will be done in the FPGA, as well as the user interface.

*Software* – Steps 2 through 4 of the algorithm will be implemented in software, and we will communicate with the FPGA to retrieve the results computed in step 1.

*User Interface* – We plan to have a list of preselected images loaded on the SD card from which the user can select to run the algorithm on. We will display the image before running, and then also after the algorithm with circles around the selected keypoints.

Key Milestones

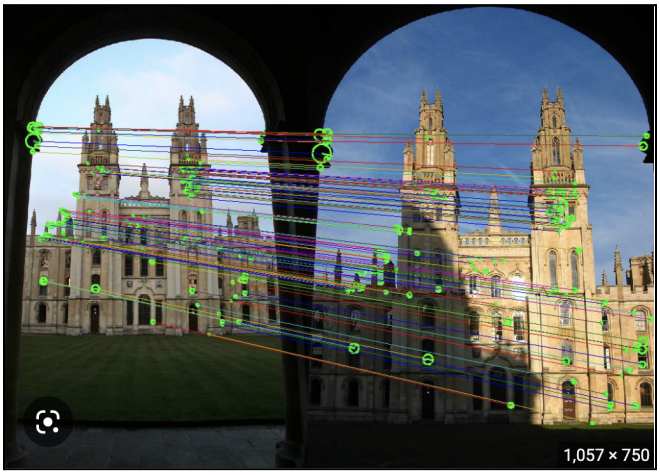
1. Literature review, understanding the SIFT algorithm, writing its pseudocode, and implementing the first step of the algorithm in software.
2. Implement steps 2, 3, and 4 in software and map out a detailed implementation of step 1 in hardware. Design protocol to pass data back and forth between HW and SW.
3. Complete the hardware implementation, resolve issues, and integrate the software and hardware.

Potential Issues:

1. Images are memory intensive, and therefore we will not be able to run convolutions on the entire image at once. Therefore, we plan on breaking up the image into strips and running convolutions on them. We will have to implement a protocol to load and retrieve the data to the FPGA and ensure proper timing.

If we have time (to do ambitious things):

1. We plan to use the FPGA as a server that can take in images dynamically from a client (Eg: laptop) and then pass it through SIFT.
2. We can also compare images taken from different angles of the same object, and match the SIFT keypoints from one image to the other.




References:

1. https://medium.com/data-breach/introduction-to-sift-scale-invariant-feature-transform-65d7f3a72d40
2. https://www.sciencedirect.com/science/article/pii/S0141933115001921?via%3Dihub
3. https://www.cs.ubc.ca/~lowe/papers/iccv99.pdf
4. FPGA Based Parallel Hardware Architecture for SIFT Algorithm
5. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7784039&tag=1