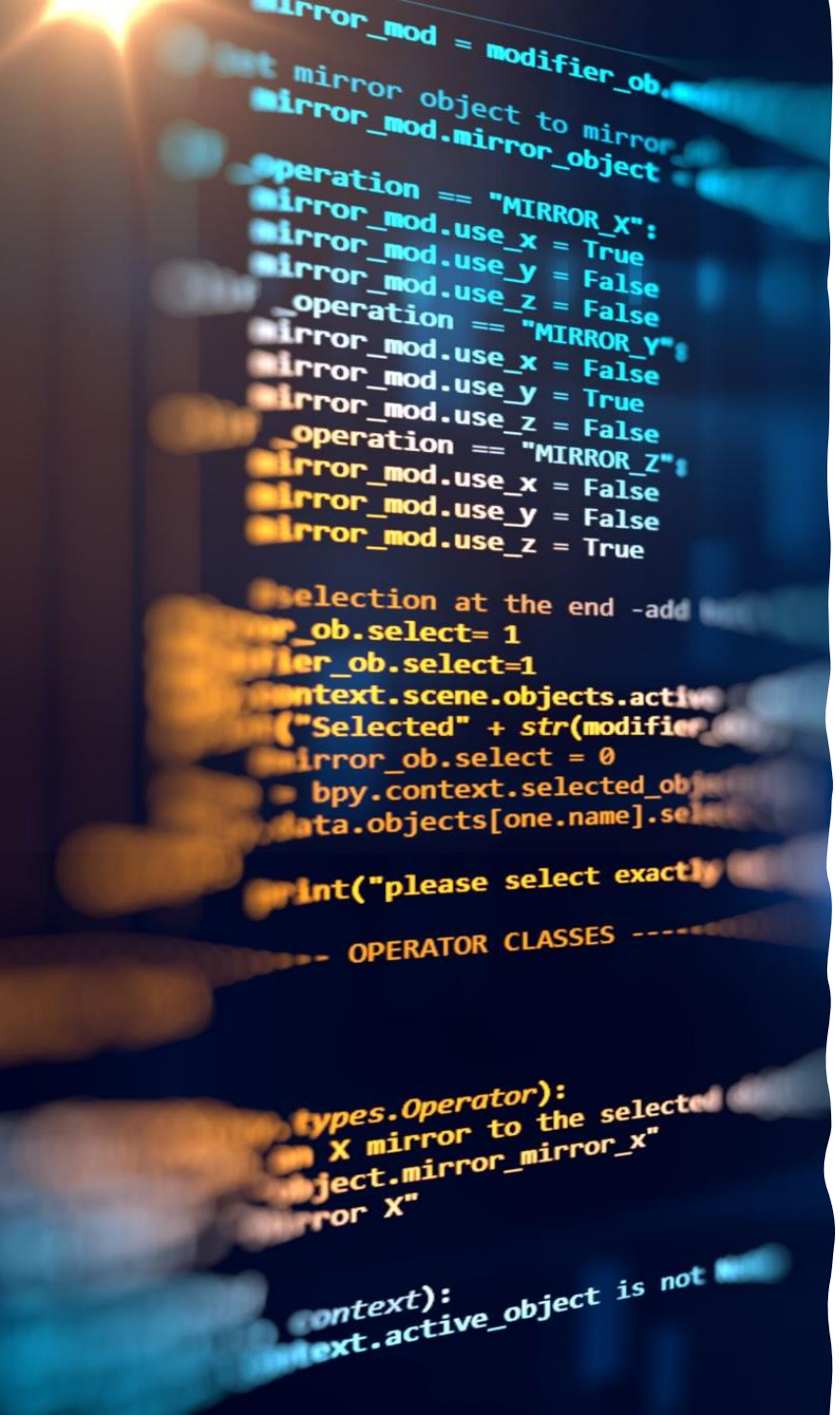


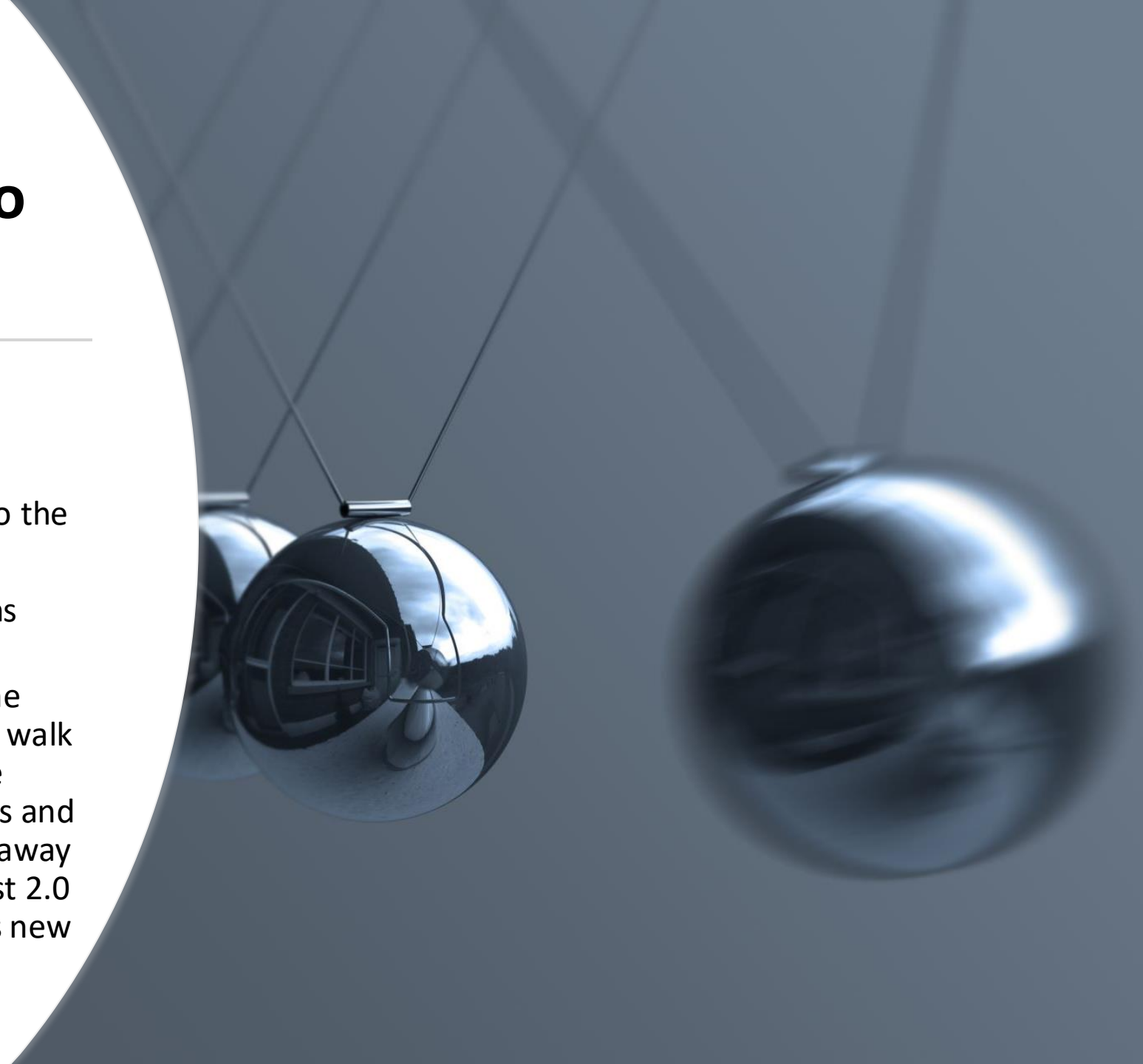
Basics of AR: Anchors, Keypoints & Feature Detection

Basics of AR

- Creating apps that work well with Augmented Reality requires some background knowledge of the image-processing algorithms that work behind the scenes
- One of the most fundamental concepts involves anchors
- These rely on keypoints and their descriptors, detected in the recording of the real-world

Anchor Virtual Objects to the Real World

- As a developer, anchor virtual objects to the world
- This ensures that the hologram stays glued to the physical location where you put it
- Objects stay in place even if the system learns more about the environment over time
    - For example, if an object is placed on the wall 2 meters away from you. Next, you walk closer to the wall. The sensors got more accurate measurements of the distances and found out that the wall was 2.1 meters away from the original position, instead of just 2.0 meters. The engine needs to reflect this new knowledge in the 3D object position



Anchor Virtual Objects to the Real World

- In a traditional single world-coordinate system, every object has fixed x, y, and z coordinates
- Anchors override the position and rotation of the transform component attached to the 3D object
- The perceived real world has priority over the static coordinate system

How are Anchors “Anchored”?

- Anchors are based on a type called “trackables” – detected *feature points* and *planes*
- As planes are grouped feature points that share the same planar surface, it all comes down to feature points

Feature Points in Computer Vision

- What is a feature point?
  - It is a distinctive location in images – for example, corners, blobs , or T-junctions are usually good candidates
- This information is not enough to distinguish one from another or to reliably find feature points again
- Therefore, also the neighborhood is analyzed and saved as a descriptor
- The most important property of a good feature point is **reliability**
- The algorithm must be able to find the same physical interest point under different viewing conditions
- A lot can change while using an AR app:
  - Camera angle/perspective
  - Rotation
  - Scale
  - Lightning
  - Blur from motion or focusing
  - General image noise

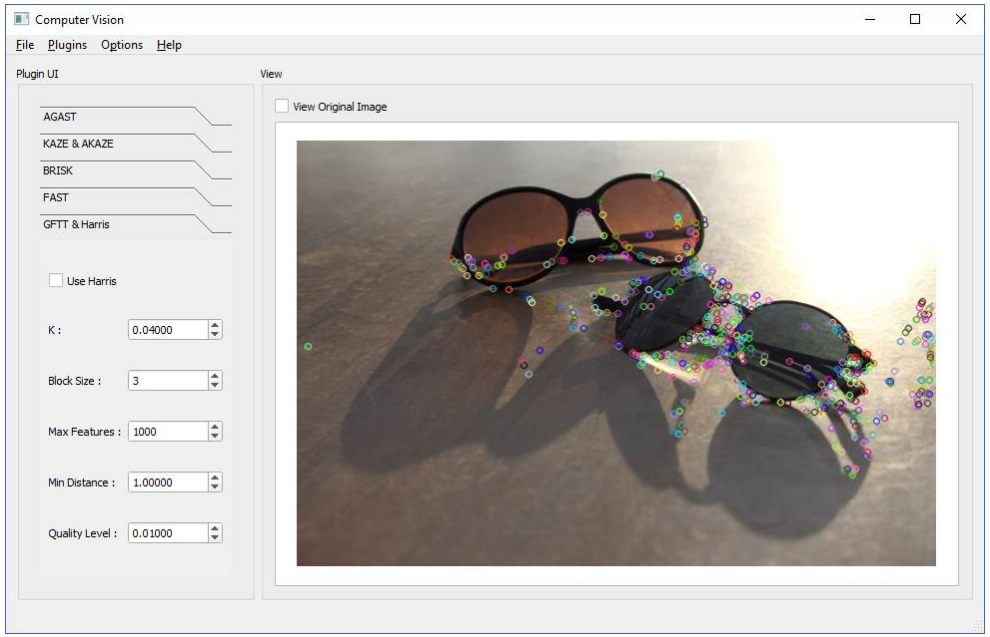
Finding Reliable Feature Points

- How do the AR frameworks find features?
  - The Microsoft HoloLens operates with an extensive amount of sensor data
  - Especially the depth information based on reflected structured infrared light is valuable
  - Mobile AR like Google ARCore and Apple ARKit can only work with a 2D color camera
- Finding distinctive feature points in images has been an active research field for quite some time

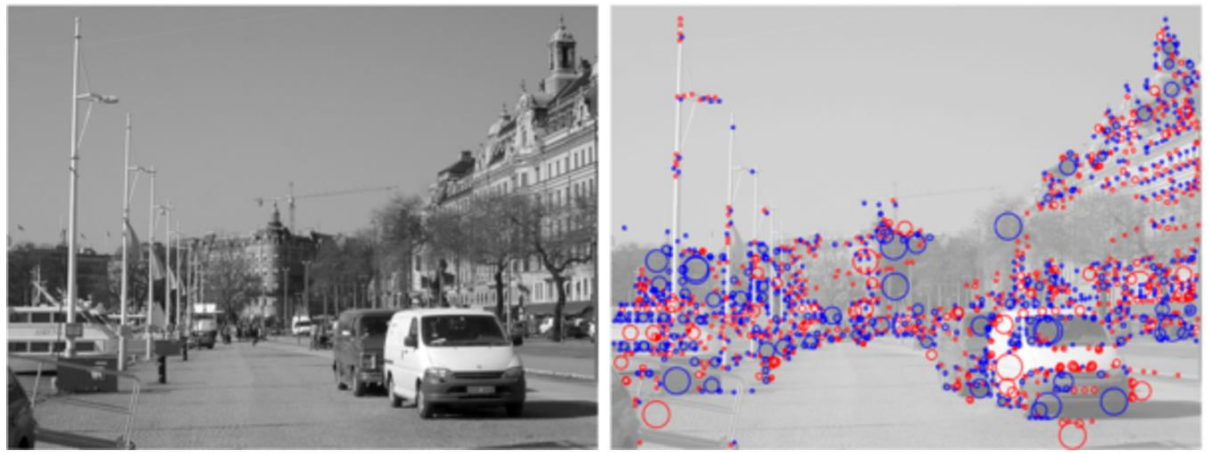
Feature Detectors and Descriptors

- Good Features to Track (GFTT)
  - The detected corners are more uniformly distributed across the image
- Scale Invariant Feature Transform(SIFT)
  - SIFT is invariant to scale, rotation, illumination, and viewpoint change; therefore, it can also correctly locate key points in noisy, cluttered, and occluded environments
- Maximally Stable Extremal Regions (MSER)
  - Is used to detect blobs in images

Good Features to Track (GFTT)



Scale Invariant Feature Transform(SIFT)

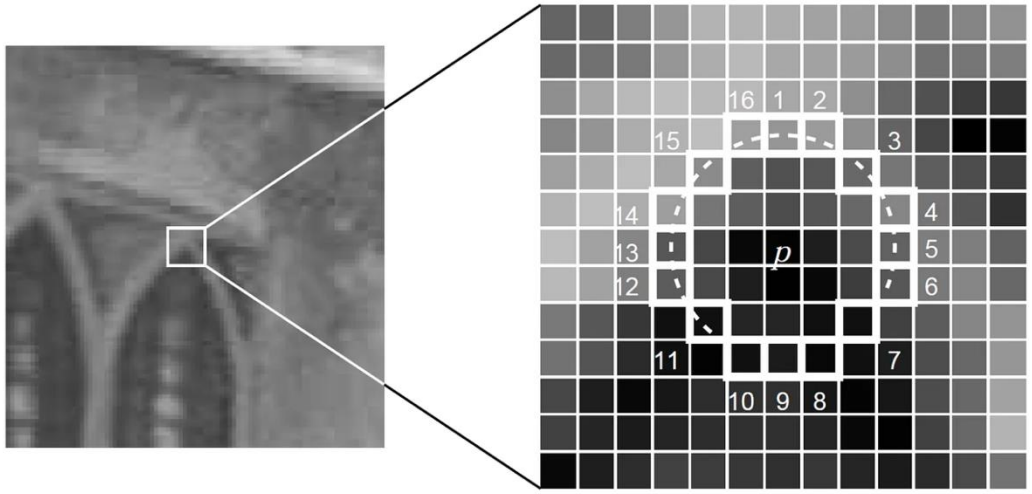


Maximally Stable Extremal Regions (MSER)

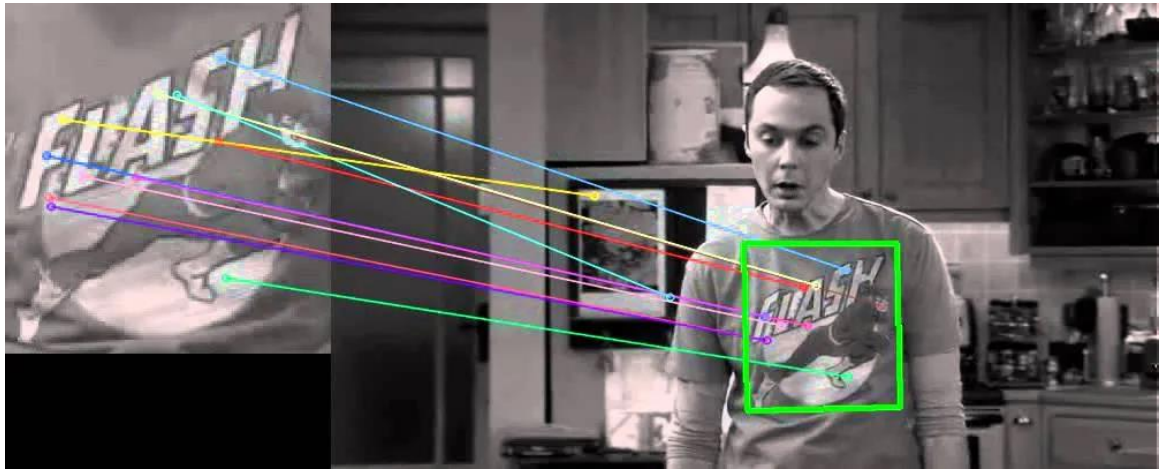
Feature Detectors and Descriptors

- Features from Accelerated Segment Test (FAST)
  - Corner detection is accomplished by first picking a collection of training images then running the FAST algorithm on each image to detect points and then using machine learning to find the optimum detection criterion
- Speeded-Up Robust Features(SURF)
  - A quick and robust approach for feature detection and extraction in the images. The major attraction of the SURF method is its ability to calculate operators fast by applying box filters, enabling its use for real-time applications

Features from Accelerated Segment Test (FAST)

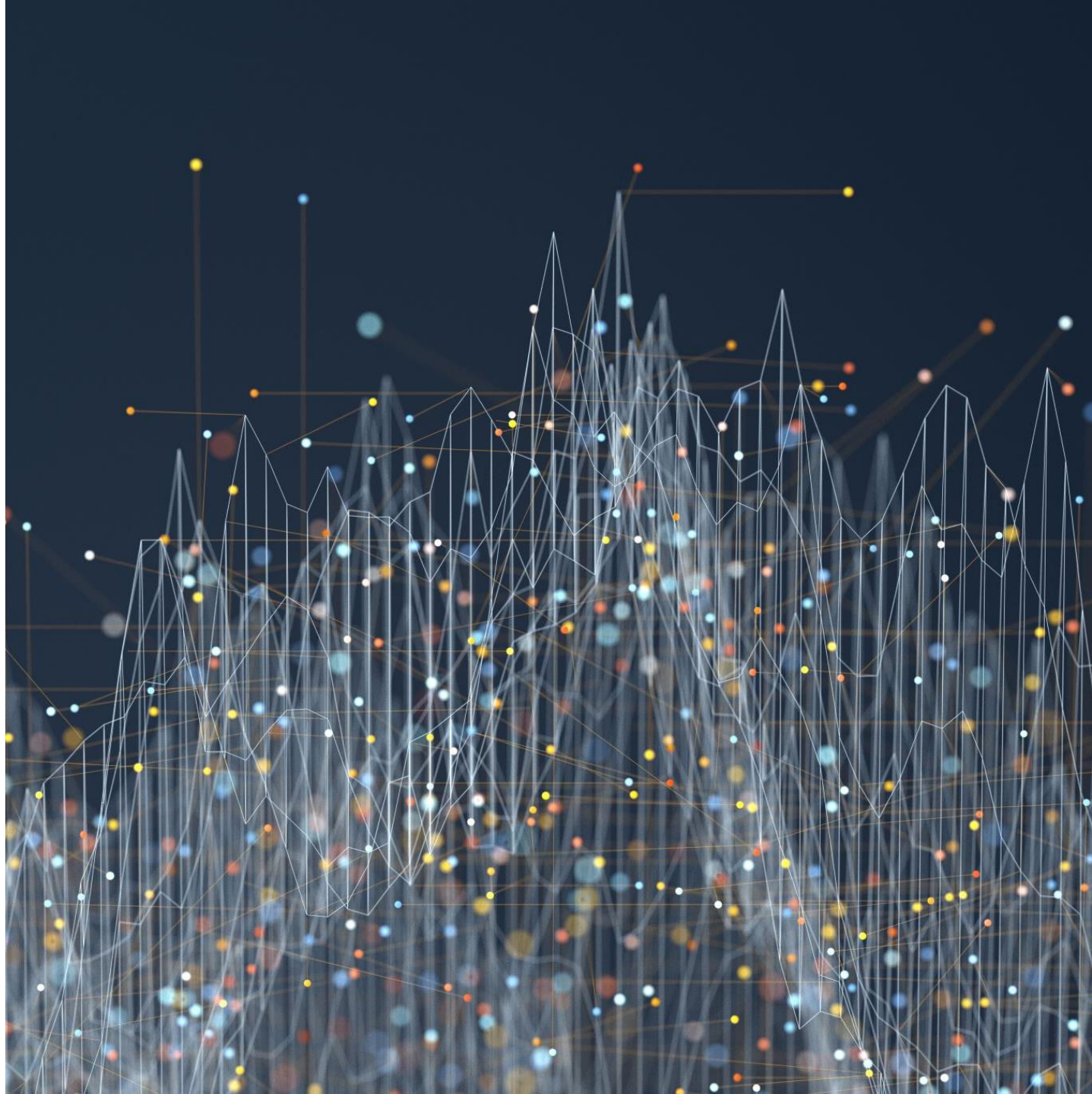


Speeded-Up Robust Features(SURF)

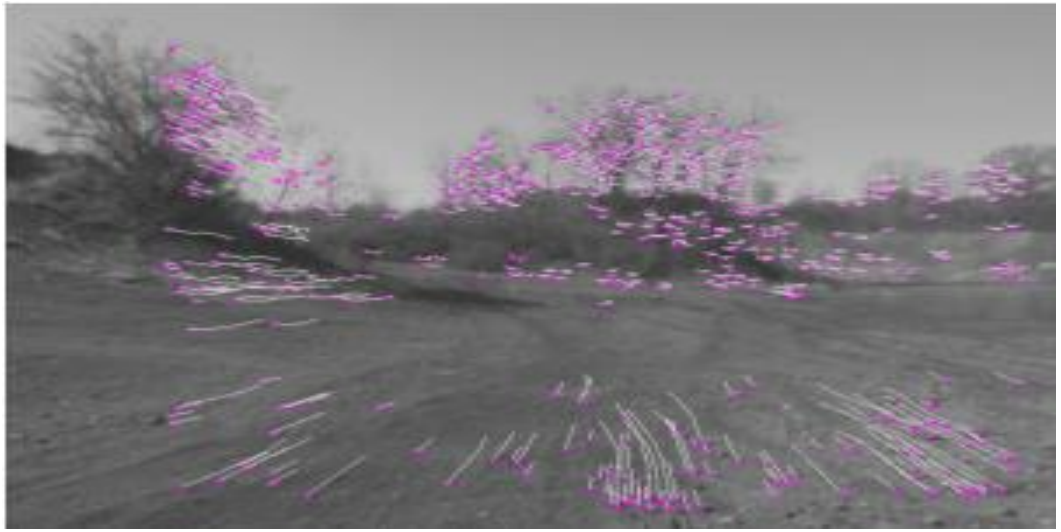


Feature Detectors and Descriptors

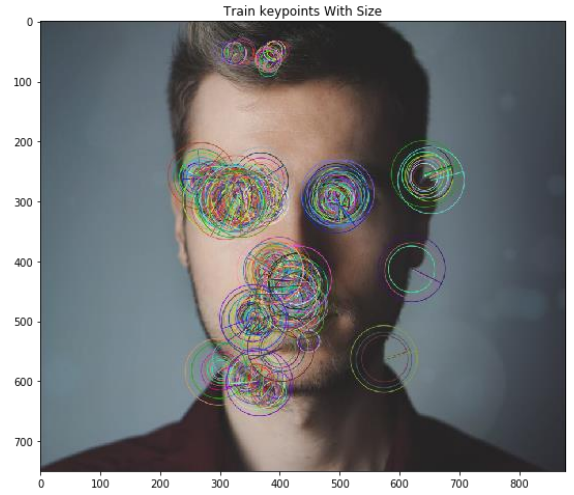
- Star Detector (CENSURE)
  - It is characterized by two considerations: stability (features invariant to viewpoint changes) and accuracy
- Binary Robust Independent Elementary Features (BRIEF)
  - Is based on binary strings
  - First, binary strings are computed from image patches
  - Then the individual bits are generated by comparing the brightness of pairs of points along the same lines

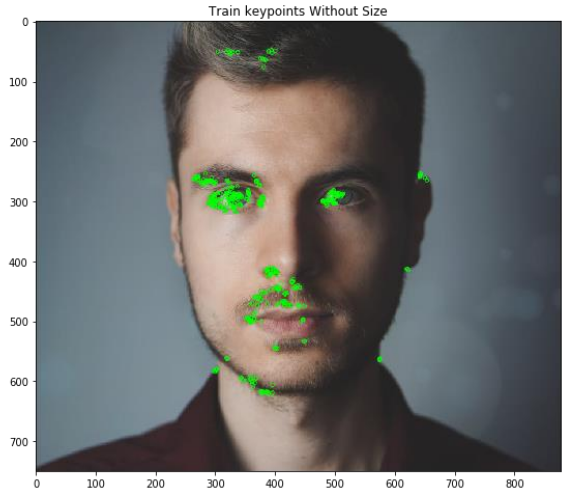


Star Detector (CENSURE)



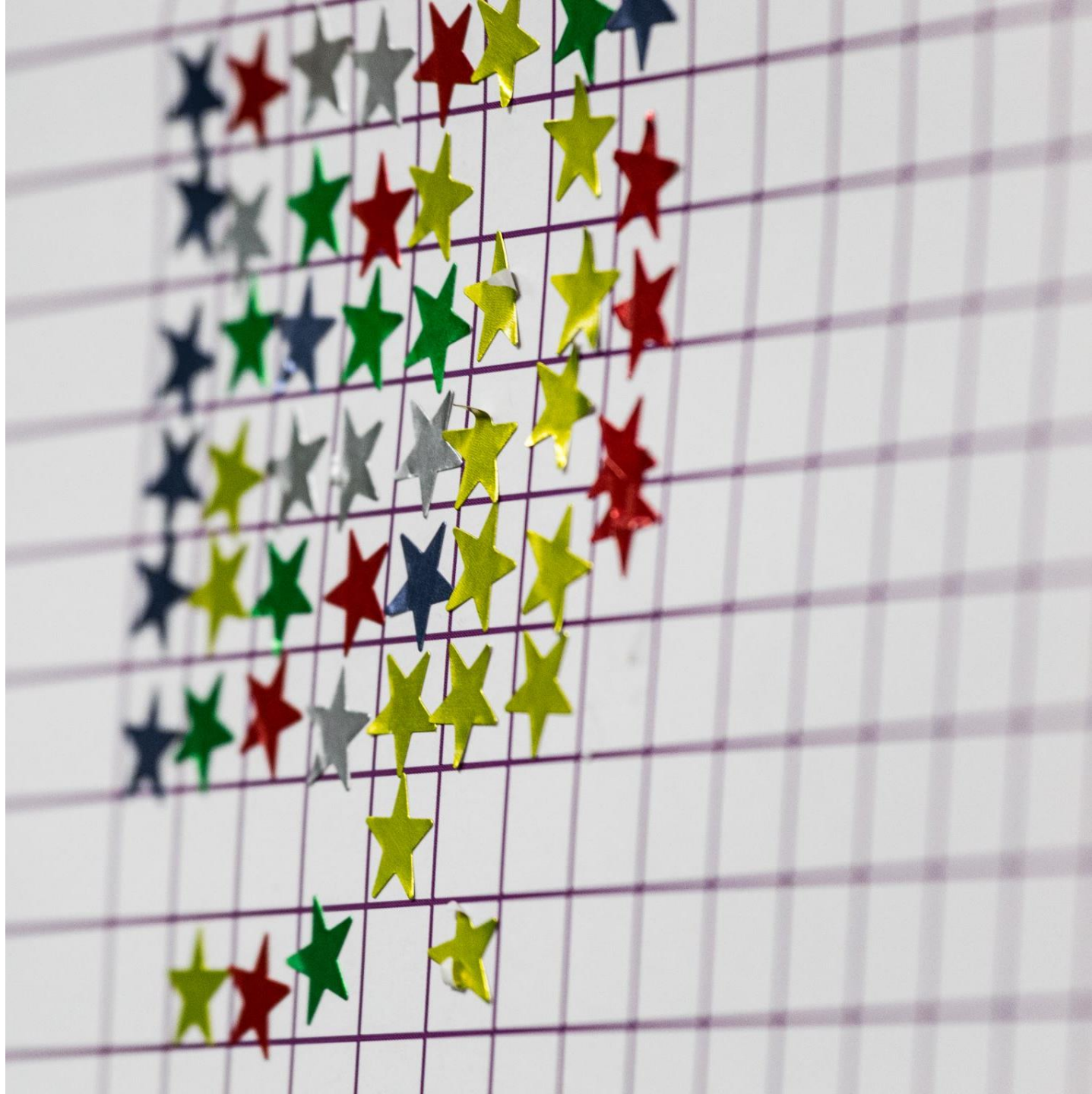
Binary Robust Independent Elementary Features (BRIEF)



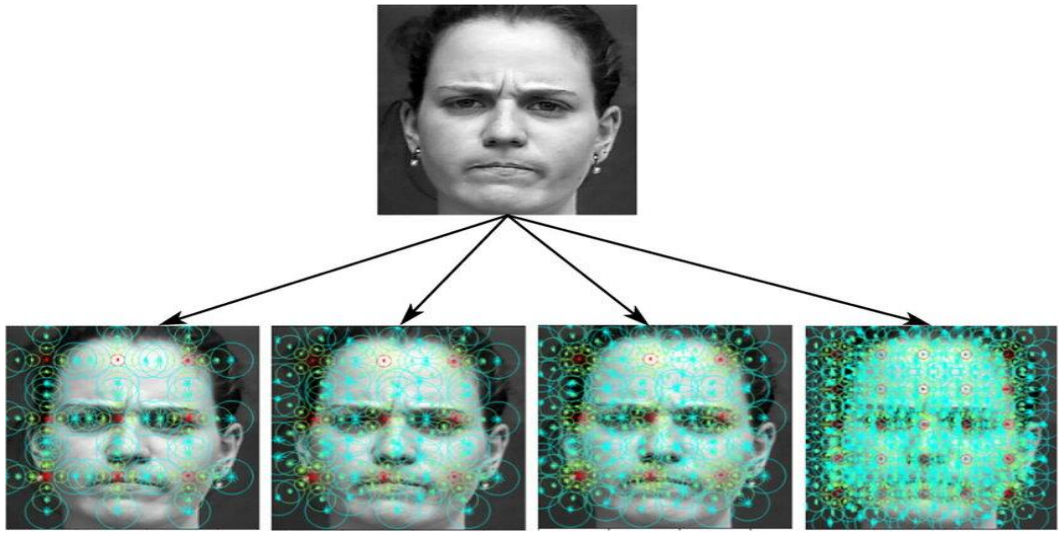


Feature Detectors and Descriptors

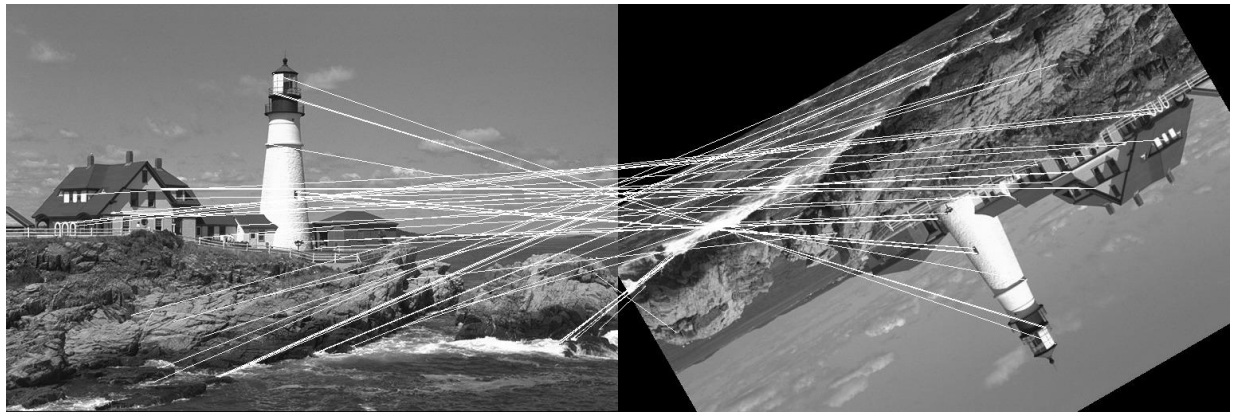
- DAISY
  - DAISY gives very good results in wide-baseline situations in comparison to pixel- and correlation-based approaches
- Binary Robust Invariant Scalable Key-Points (BRISK)
  - In this algorithm, key-points are discovered in the image scale-space pyramid
  - Following that, a circular sampling pattern is created in the vicinity of each detected key-point, which calculates brightness comparisons to form a descriptor string



Daisy

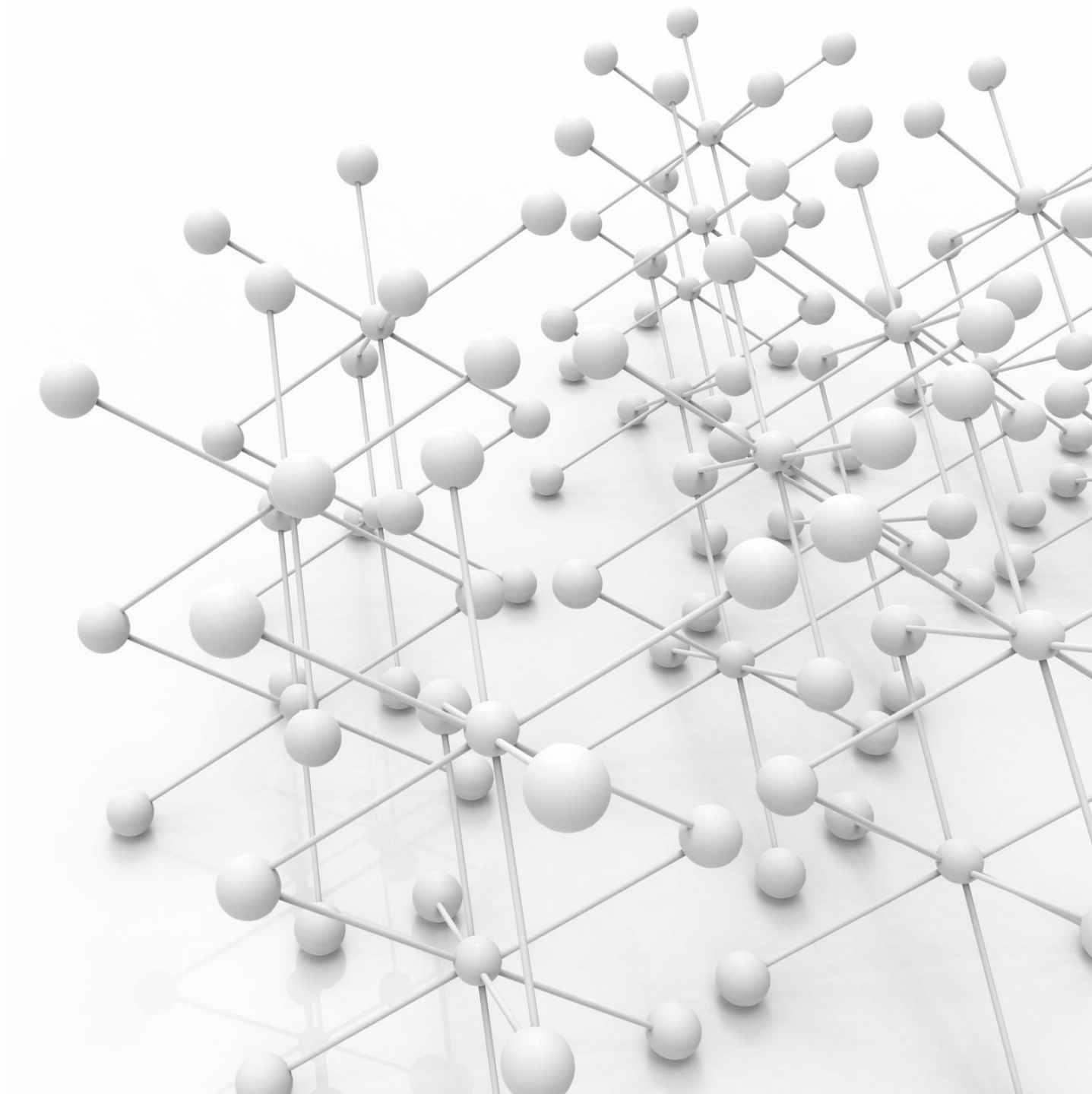


Binary Robust Invariant Scalable Key-Points (BRISK)



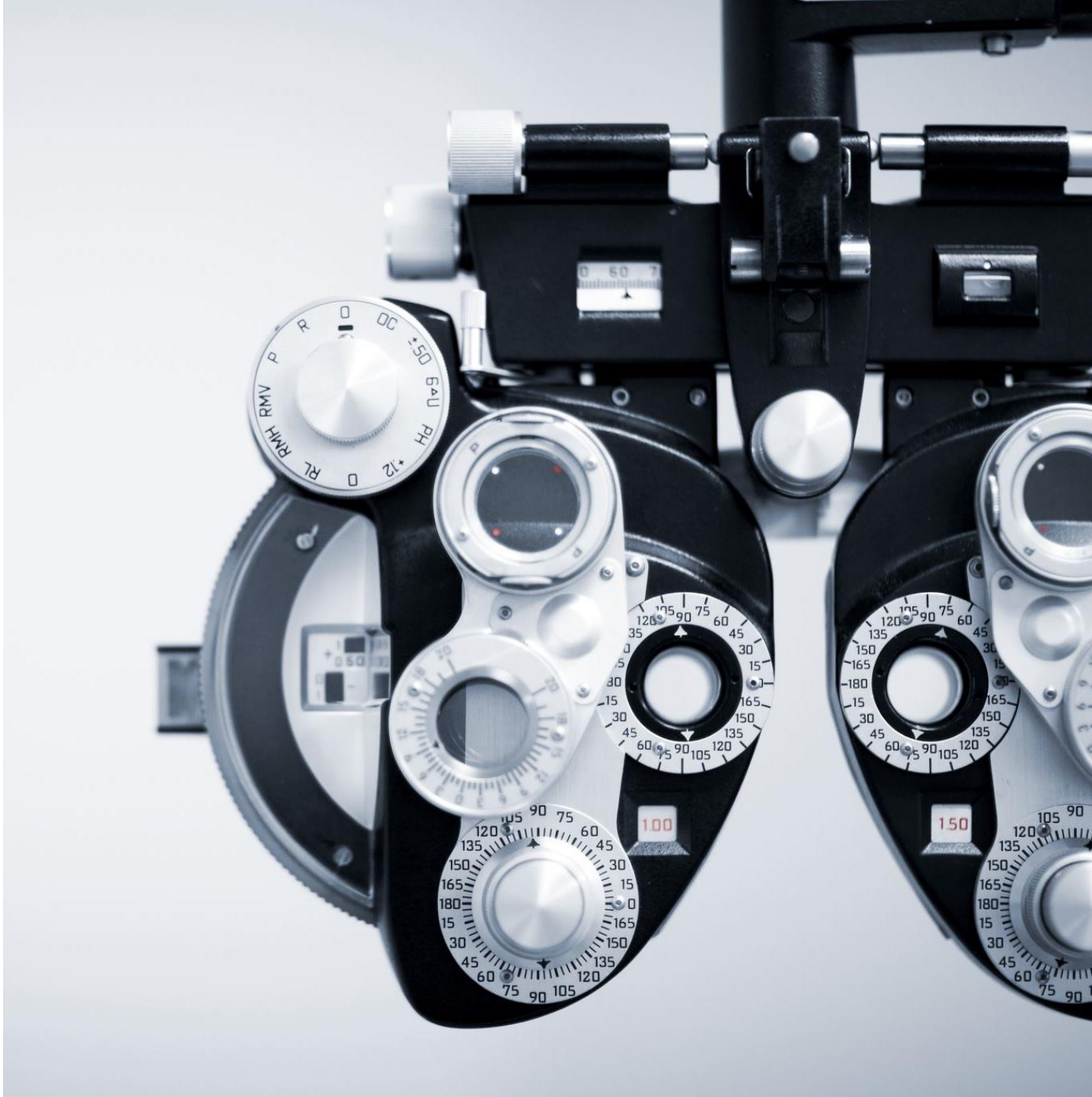
Feature Detectors and Descriptors

- Adaptive and Generic Accelerated Segment Test (AGAST)
  - Uses specialized decision trees to enhance the performance
- ORB
  - ORB is a combination of the FAST detector and the BRIEF descriptor, with several enhancements to improve the performance
  - ORB adds a fast and accurate orientation component to the FAST algorithm
  - It uses an image pyramid to generate multiscale features

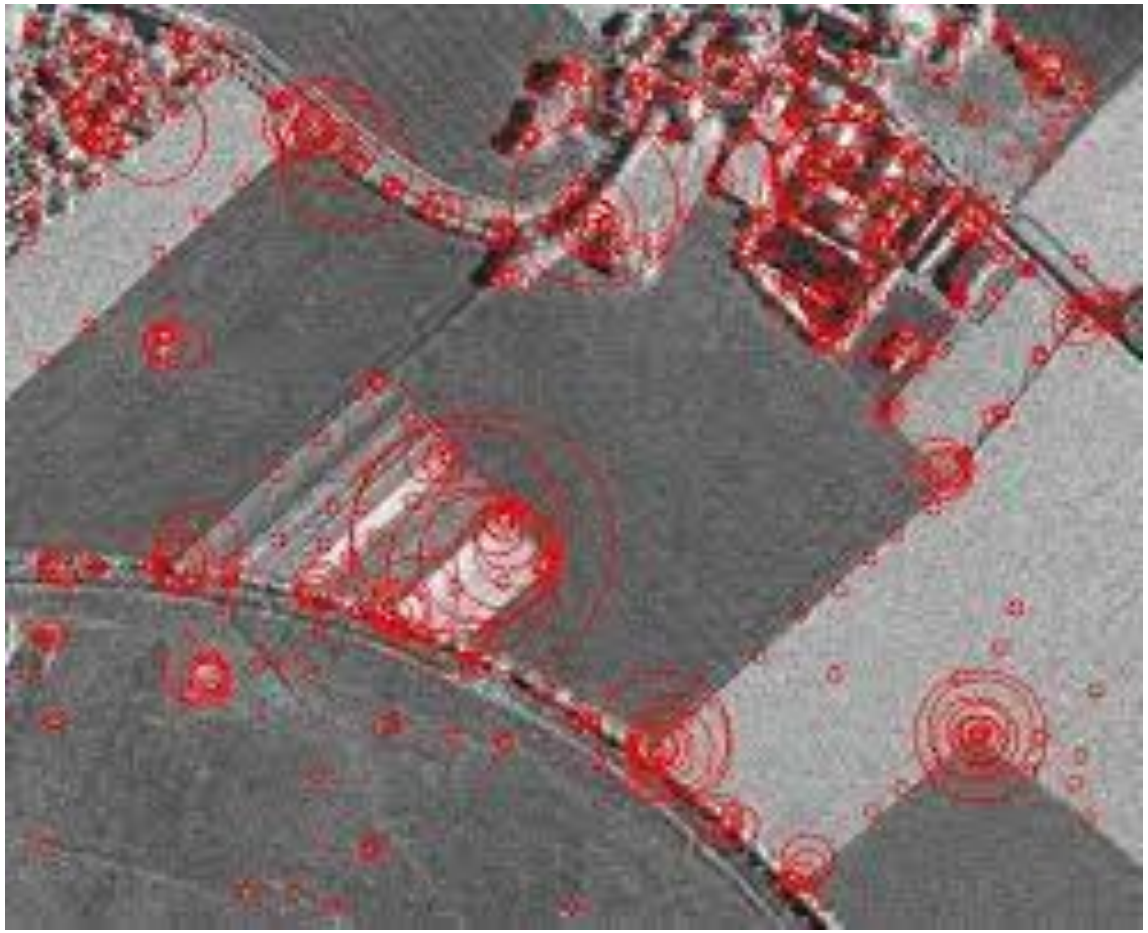


Feature Detectors and Descriptors

- Fast Retina Key-Point (FREAK)
  - It uses a retinal sampling grid with more density of points near the center with the density decreasing exponentially with distance from the center
- Maximal Self-Dissimilarities (MSD)
  - Can detect features across multi-modal image pairs
  - It is based on the intuition that image patches that are very distinct across a reasonably large area of their surroundings are repeatable and distinctive



Maximal Self-Dissimilarities (MSD)

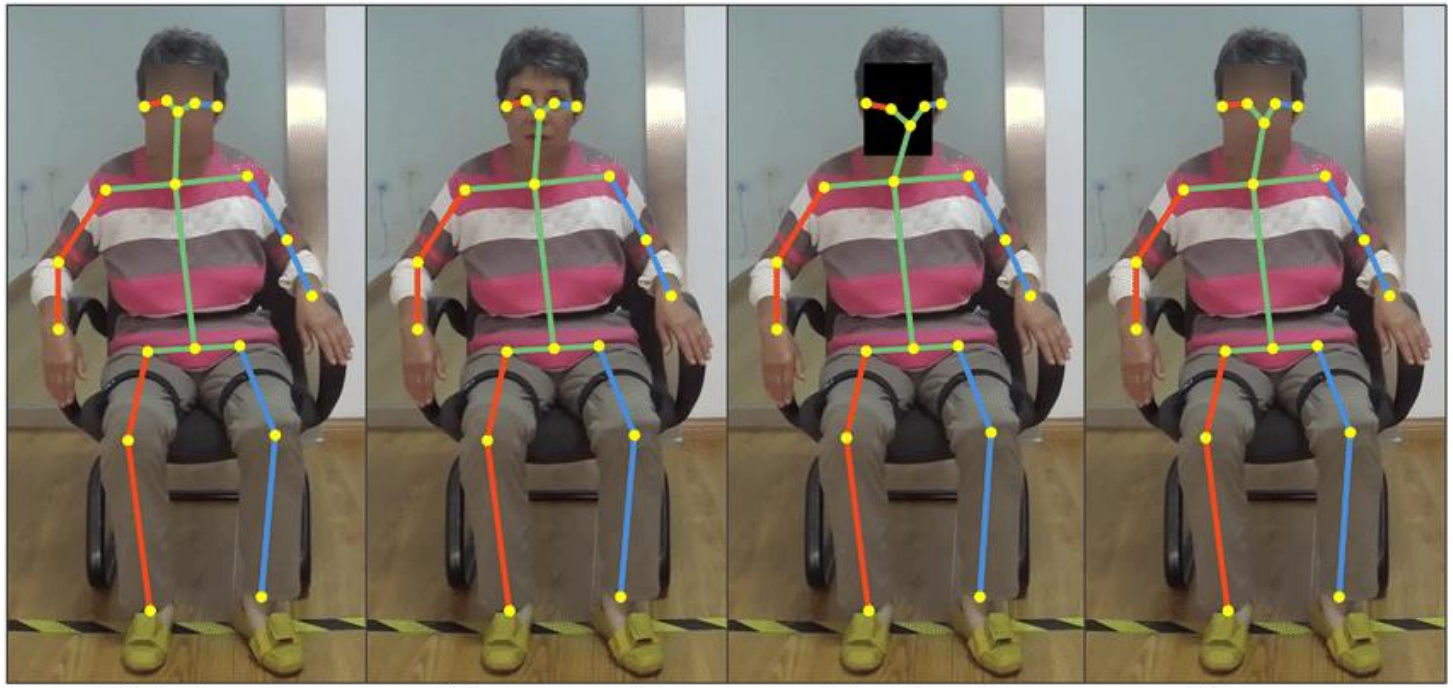


How is a Feature Detected and Stored? With Keypoint

- Feature detection is a multi-step process
- Its components vary depending on the algorithms
- A short description of a typical detection algorithm:

Keypoint Detection

- To make the feature point candidates scale-invariant and less dependent on noise, it is common to blur the image
- Differently scaled variants of the image improve scale independence
- To detect a tree as a whole to achieve reliable tracking; not every single twig



Keypoint Description

- Each of all the detected keypoints needs to be a unique fingerprint
- The algorithm must find the feature again in a different image
- This might have a different perspective, lightning situation, etc. Even under these circumstances, a match has to be possible

- Thank you