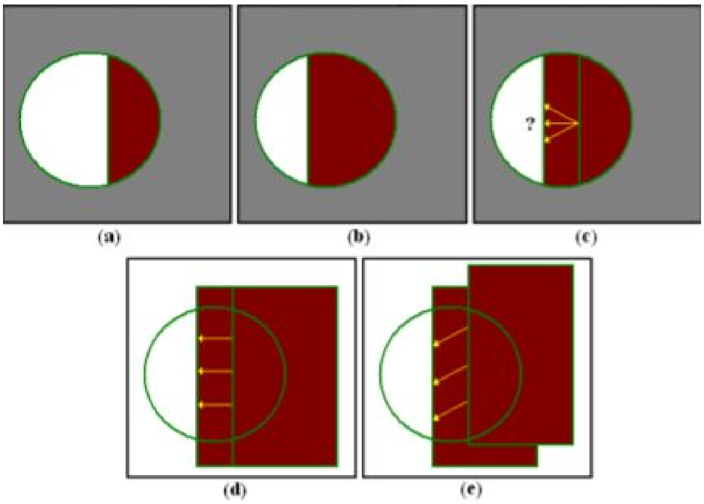
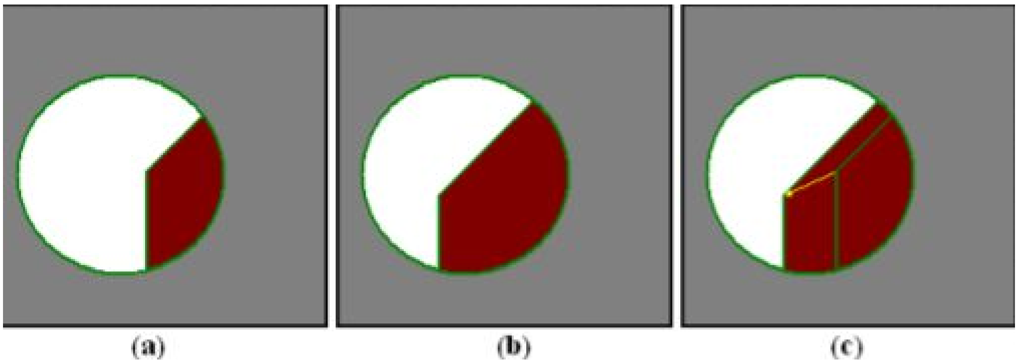
Features

- **Features**
    - Used to combat the ambiguity of edges
        - 
        - **"Aperture Problem"**
    - **Corners/Features are less ambiguous**
        - 
        - A corner is the intersection of two edges
        - **An interest point is any feature which can be robustly detected**
        - **Benefits**
            - Less points to consider than edges
            - Easier to establish the correspondences
        - Can have 'Spurious' (illegitimate) features
    - **General approach to corner detection**
        - **Determine the cornerness values**
            - For each pixel, based on local neighbours
            - Produce a cornerness map
        - **Non-maxima suppression**
            - Multiple responses - compare to local neighbours
            - Suppress all but local max to prevent multiple responses from one corner
        - **Thresholding**
            - Threshold cornerness map - get significant corners

- **Moravec Corner Detection**
  - Looks at the local variation around a given pixel
    - Compares a region centred on the pixel to 8 regions shifted slightly, and uses the sum of the squared difference
    - $$V_{u,v}(i,j) = \sum_{\forall a,b \in Window} \left(f(i+u+a, j+v+b) - f(i+a, j+b)\right)^2$$
      - Where (u,v) are the minor shifts about the point
    - **Cornerness value is taken to be the minimum value of $V_{u,v}(i,j)$**
  - **Flaws**
    - **Anisotropic response**
      - Responds differently to diagonal lines than it does to horizontal/vertical
      - Can be reduced with smoothing
    - **Noisy reponse**
      - Very sensitive to noise
      - Can be reduced using a larger region to consider, or by smoothing the image beforehand
- **Harris/Plessey Corner Detection**
  - **Cornerness value determined using partial derivatives, gaussian weighting, and the eigenvalues of a matrix representation of the equation**
  - Consider the intensity variation (sum of squared differences) of an image patch W for a small shift (Δi,Δj)
    - $$SSD_W(\Delta i, \Delta j) = \sum_{(i,j)\in W} \left(f(i,j) - f(i-\Delta i, j-\Delta j)\right)^2$$
    - **We can approximate the second term as..**
    - $$f(i-\Delta i, j-\Delta j) \approx f(i,j) + \begin{bmatrix} \frac{\delta f(i,j)}{\delta i} & \frac{\delta f(i,j)}{\delta j} \end{bmatrix} \begin{bmatrix} \Delta i \\ \Delta j \end{bmatrix}$$
    - **Then we can sub back in, and rewrite the equation as...**
    - $$SSD_W(\Delta i, \Delta j) = \sum_{(i,j)\in W} \left(f(i,j) - f(i,j) - \begin{bmatrix} \frac{\delta f(i,j)}{\delta i} & \frac{\delta f(i,j)}{\delta j} \end{bmatrix} \begin{bmatrix} \Delta i \\ \Delta j \end{bmatrix}\right)^2$$
      $$SSD_W(\Delta i, \Delta j) = \sum_{(i,j)\in W} \left(\begin{bmatrix} \frac{\delta f(i,j)}{\delta i} & \frac{\delta f(i,j)}{\delta j} \end{bmatrix} \begin{bmatrix} \Delta i \\ \Delta j \end{bmatrix}\right)^2$$
      $$SSD_W(\Delta i, \Delta j) = \sum_{(i,j)\in W} \left(\begin{bmatrix} \Delta i & \Delta j \end{bmatrix} \left(\begin{bmatrix} \frac{\delta f(i,j)}{\delta i} \\ \frac{\delta f(i,j)}{\delta j} \end{bmatrix} \begin{bmatrix} \frac{\delta f(i,j)}{\delta i} & \frac{\delta f(i,j)}{\delta j} \end{bmatrix}\right) \begin{bmatrix} \Delta i \\ \Delta j \end{bmatrix}\right)$$
      $$SSD_W(\Delta i, \Delta j) = \begin{bmatrix} \Delta i & \Delta j \end{bmatrix} \begin{bmatrix} \sum_{(i,j)\in W} \left(\frac{\delta f(i,j)}{\delta i}\right)^2 & \sum_{(i,j)\in W} \frac{\delta f(i,j)}{\delta i}\frac{\delta f(i,j)}{\delta j} \\ \sum_{(i,j)\in W} \frac{\delta f(i,j)}{\delta i}\frac{\delta f(i,j)}{\delta j} & \sum_{(i,j)\in W} \left(\frac{\delta f(i,j)}{\delta j}\right)^2 \end{bmatrix} \begin{bmatrix} \Delta i \\ \Delta j \end{bmatrix}$$

- **Using this matrix we compute the eigenvalues**
    - **Both high => Corner**
    - **One high => edge**
    - **None high => constant region**
- This is formalised by the cornerness metric proposed by harris
    - $$M = \begin{bmatrix} \sum_{(i,j)\in W} \left(\frac{\delta f(i,j)}{\delta i}\right)^2 & \sum_{(i,j)\in W} \frac{\delta f(i,j)}{\delta i}\frac{\delta f(i,j)}{\delta j} \\ \sum_{(i,j)\in W} \frac{\delta f(i,j)}{\delta i}\frac{\delta f(i,j)}{\delta j} & \sum_{(i,j)\in W} \left(\frac{\delta f(i,j)}{\delta j}\right)^2 \end{bmatrix} = \begin{bmatrix} A & B \\ B & C \end{bmatrix}$$
      $$\det(M) = \lambda_1 \lambda_2 = AC + B^2$$
      $$\mathrm{trace}(M) = \lambda_1 + \lambda_2 = A + C$$
      $$C(i,j) = \det(M) - k(trace(M))^2$$
        - **Where k is a constant to be in the range of 0.04 to 0.06**
- Uses Gaussian weighting over the image patch W, with the centre holding the most weight
- **Pros**
    - Very repeatable, better detection rate
- **Cons**
    - Computationally expensive
    - Sensitive to noise, and slightly anisotropic

- **FAST Corner Detection**
    - “Features from Accelerated Segment Test”
    - Considers a circle of points, if it finds an arc of 9+ of continuous points that are all brighter or darker than the nucleus
        - **Using a threshold T, where T specifies the minimum difference between the points in the arc and the centre point**
    - **No cornerness metric**
        - To do non-maxima suppression and ensure a single response per corner, the maximum threshold to still classify the point as a corner is taken
    - **Pros: very fast**
- ~~**Scale Invariant Feature Transform (SIFT)**~~
    - ~~Designed to provide:~~
        - ~~Repeatable, Robust features~~
        - ~~Used for tracking, stitching etc~~
    - ~~**Invariant to rotation, scaling, and partially to illumination and viewpoint changes**~~
    - ~~**Process**~~
        - ~~**Scale space extrema detection**~~
            - ~~Considers the image at multiple scales simultaneously~~
            - ~~Detects the extrema (maxima/minima) within the scaled images~~
            - ~~**Uses gaussians**~~

        - $L_n(i,j,k,\sigma) = G(i,j,k^n\sigma) * f(i,j)$ *where* $n = 0, 1, 2, 3, ...$
    - ~~Potential keypoint locations are then found by considering the **difference of gaussian (DoG) across various scale spaces**~~
        - ~~Multiple DoGs are considered~~
        - ~~Points that are greater or less than their neighbours in both the current scale and the neighbouring scales are considered~~
- ~~**Accurate keypoint location**~~
    - ~~To locate keypoints more accurately the data is modelled onto a 3D quadratic~~
        - ~~Interpolated maximum/minimum can then be found~~
    - ~~Located keypoints weren't robust enough, so additional tests were used~~
        - ~~**First test:** considers local contrast using the curvature of the quadratic around the point~~
            - ~~Low curvature = low contrast - discard the keypoint~~
        - ~~**Second test:** aims to discard keypoints that are poorly localised (i.e. on an edge)~~
            - $\mathbf{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}$ $\quad \mathrm{Tr}(\mathbf{H}) = D_{xx} + D_{yy} = \alpha + \beta$, $\quad \mathrm{Det}(\mathbf{H}) = D_{xx}D_{yy} - (D_{xy})^2 = \alpha\beta$ $\quad \alpha = r\beta$

              $$\frac{\mathrm{Tr}(\mathbf{H})^2}{\mathrm{Det}(\mathbf{H})} = \frac{(\alpha+\beta)^2}{\alpha\beta} = \frac{(r\beta+\beta)^2}{r\beta^2} = \frac{(r+1)^2}{r}$$
            - $$\frac{\mathrm{Tr}(\mathbf{H})^2}{\mathrm{Det}(\mathbf{H})} < \frac{(r+1)^2}{r}$$
            - ~~**We use differences of neighbouring sample points to estimate the derivatives**~~
                - ~~We also avoid computing eigenvalues as we only want the ratio(Tr(H) and Det(H))~~
            - ~~$Tr(H)^2/Det(H) < (r+1)^2/r$ is comparing the ratio of the curvature to a threshold (r)~~
- ~~**Keypoint orientation assignment**~~
    - ~~So we now have scale invariant features~~
    - ~~Now we need them to be rotation invariant…~~
    - ~~**For scale invariance, the keypoint scale is used to select the smoothed image with the closest scale**~~
    - ~~**For orientation invariance we describe the keypoint wrt the principal orientation**~~
        - ~~Create an orientation histogram (36 bins)~~
            - $$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2}$$
              $$\theta(x,y) = \tan^{-1}((L(x,y+1) - L(x,y-1))/(L(x+1,y) - L(x-1,y)))$$

        - **~~Weight by gradient magnitude and using a gaussian function based on the distance to the keypoint~~**
        - **~~Sample points around the keypoint~~**
        - ~~Highest peaks define the keypoint orientation~~
            - ~~Peaks within 80% are used to create new keypoints with different orientations~~
        - ~~Stable results~~
- **~~Keypoint descriptors~~**
    -