Chapter 3
Related Theory

3.1 Image Registration methods for Augmented Reality

As a practical example of Augmented Reality (AR), consider the case of high demand for superior life as human lifespan increases. People will prefer to live a comfortable and healthier life during old age. Wang et al. [Wang et al. 2017] reports a phenomenon, the ageing work force, according to which the old people should be permitted to work as long as they want and are capable of being productive. However, human body, physiologically and mentally starts deteriorating as it gets old. Therefore, to face such complications of old age, Wang et al. proposed few ways to augment human capabilities like hearing, memory, vision etc. to human body using certain electronic devices and AR that visually augments the real-world environment by projecting computer-generated information into it, the concept commonly known as the formation of “bionic people”.

This section gives an overview of image registration methods that have been used or considered effective for image registration process in an AR system. Image registration methods used for developing a markerless AR system extracts natural features from the image scene using two modules: feature detector and feature descriptor. Feature detector detects stable keypoints (points of interest) in the image and feature descriptor selects image characteristics around the extracted keypoint that describe its appearance distinctively. The desirable properties of a feature descriptor includes affine invariance, reliability and low computational complexity for making the execution of tasks like object recognition and tracking, feature matching etc. robust and accurate.

3.1.1 Feature Detectors

Harris-Affine: This detector is designed to result in interest points which are invariant to scale and affine changes. The algorithm is initialized using Harris points extracted at multi-scale of an image (Laplacian-scale selection to make it scale invariant). Harris points provide approximate localization and scale for initialization. An iterative procedure is applied for each point to modify point’s position as well as scale and shape of the point’s neighborhood, converging towards stability of extracted points measured in terms of invariance to affine transformations. The output regions are also normalized using affine shape adaption algorithm to make them affine invariant (Harris-Affine) [Mikolajczyk and Schmid 2002, Mikolajczyk and Schmid 2004]. The second moment matrix used for defining the elliptical region around the point of interest is given as:

$$HaM = Ha\left(X, \sigma_d, \sigma_{gs}\right) = \sigma_d^2 g\left(\sigma_{gs}\right) \times \begin{bmatrix} I_{aa}(X, \sigma_d) & I_{ab}(X, \sigma_d) \\ I_{ab}(X, \sigma_d) & I_{bb}(X, \sigma_d) \end{bmatrix} \tag{3.1}$$

where, local image derivatives are evaluated using Gaussian kernels of scale σ_d. They are then averaged in the neighborhood of the point X by smoothing with a Gaussian window of scale σ_{gs}.

Hessian-Affine: This detector is mostly similar to Harris-Affine Detector i.e. scale-selection is based on the Laplacian, and the point of interest is surrounded by an ellipse which is determined using second moment matrix of the intensity gradient [Baumberg 2000, Lindeberg and Garding 1997]. However, the second moment matrix used by this detector determines blobs and ridges strongly and is given as:

$$HM = H(X, \sigma_d) = \begin{bmatrix} I_{aa}(X, \sigma_d) & I_{ab}(X, \sigma_d) \\ I_{ab}(X, \sigma_d) & I_{bb}(X, \sigma_d) \end{bmatrix} \tag{3.2}$$

[Mikolajczyk and Schmid 2002, Mikolajczyk and Schmid 2004]. But as described in [Lowe 2004], both Harris-Affine and Hessian-Affine detectors are not fully affine invariant.

Maximally Stable Extremal Regions (MSER): MSER algorithm [Matas et al. 2004] follows a four step procedure for extracting stable regions:

1. Component Tree Formation:

   The image is sorted in increasing or decreasing order of their pixel intensity value and then union find algorithm is used to keep a track of list of connected components along with their region size. The complexity of union find implementation is computed to be quasi linear i.e. $O(n\ log\ (logn))$.

2. Extremal Region Extraction:

   A region E is an extremal region if for all r ∈ E, s ∈ ∂E: I(r) > I(s) (maximum intensity region) where: ∂E represents the confines of E where the set of pixels of ∂E are adjacent to atleast one pixel of E and not belonging to E.

3. Arrangement of Extracted Extremal Regions:

   After the extraction of extremal regions from the component tree, two extremal regions R_l and R_{l+1} are connected if and only if $R_l \subset R_{l+1}$. These connected extremal regions are also known as nested extremal regions.

4. Stability Score:

   Let $(R_1, \ldots, R_{l-1}, \mathrm{R_l} \ldots)$ be a sequence of nested extremal regions i.e. $R_l \subset R_{l+1}$, then extremal region R_l is maximally stable if:

   $v_{R_l} = |R_{l+\Delta} - R_{l-\Delta}|/|R_l|$ is a local minima.

   Here $|R_l|$ is the region size

   Suppose, R_l is at gray level g then $R_{l+\Delta}$, $R_{l-\Delta}$ are the extremal regions obtained by moving upwards and downwards in the component tree from region R_l with gray level $g + \Delta$ and $g - \Delta$ respectively.

   So, If $v_{R_l} < v_{R_{l+1}}$, consider $v_{R_{l+1}}$ as unstable

   If $v_{R_l} > v_{R_{l+1}}$, consider v_{R_l} as unstable

   else**,** do nothing.

Scale Invariant Feature Transform (SIFT): Lowe [Lowe 2004] presented a method for performing accurate matching between image pairs representing distinctive views of an object or a scene. Features extracted from images to perform such a task were designed to be invariant to image scale and rotation. The method was proven to result in robust image matching in conditions of affine distortion, viewpoint change and change in illumination. Distinctive property of the extracted features allowed the method to accurately match a single feature against large database of features from many images. The process of extracting SIFT features starts with scale space peak selection to locate potential interest points followed by outlier rejection to identify the key-points. Subsequently, the orientation assignment is done to achieve rotation invariance and finally, image gradient key-point descriptor is used to make SIFT invariant to scale and illumination changes. SIFT is considered to be a fully scale invariant feature detector as it takes into account multiple scale resolutions of the operated image. However, a comparative study performed by Yu and Morel [Yu and Morel 2011] showed that SIFT fails under extreme viewpoint changing conditions.

Affine-SIFT (ASIFT): ASIFT feature detector is an enhancement of SIFT and is considered to be a fully affine invariant feature detector. In this approach, image views obtained by changing two camera axis orientation parameters i.e. latitude and longitude angles are simulated. These image parameters are then clubbed with SIFT evaluated parameters involving simulated scale and normalized rotation and translation parameters. As a result, ASIFT is proven to work well under different affine conditions [Yu and Morel 2011]. The authors based their theory on the fact that camera captured images of a physical object with a smooth boundary in changing positions undergo smooth apparent deformations. Affine transforms of the image plane can very well approximate these local deformations. Extracting affine invariant features for performing tasks like solid object recognition is proposed to be achieved by normalization methods. Performance of affine recognition is evaluated using two parameters: 1) absolute tilt: degree of tilt between the frontal view (f_1) and slanted view (f_2) of the image scene, 2) transition tilt: real time captured images are usually slanted views, transition tilt is used to measure the degree of tilt between two such slanted views $(f_2\ )$ and (f_3). More formally,

- Transition tilt is symmetric, i.e. $Tilt(f_2, f_3) = Tilt(f_3, f_2)$.
- Transition tilt depends upon the absolute tilt and on the longitude angle difference: $Tilt(f_2, f_3) = Tilt(a, a';\ \theta - \theta')$, where a, θ and a', θ' defines the absolute tilt and longitude angle difference between (f_1) (f_2) and (f_1) (f_3) respectively.
- The transition tilt is equal to the absolute tilt: incase if other image is in frontal view.

Speeded Up Robust Features (SURF): SURF feature detector [Bay et al. 2008] holds properties such as high repeatability, distinctiveness and robustness. The detector is based on the Hessian matrix and uses a very basic approximation for determining location and space. Scale space approximation is done using box filters i.e. instead of defining the scale spaces by iteratively applying Gaussian to smooth the image and sub-sampling in order to achieve a higher level of image pyramid, scale space is determined by up-scaling the filter size rather than repeatedly reducing the image size. Use of integral images reduced the computation time and improvised the task of attaining image convolutions.

SURF descriptor, however, describes the distribution of Haar-wavelet responses within the extracted feature neighborhood. It defines a circular region around the extracted feature and uses the analyzed information from within that circular area to fix a reproducible orientation. A square region is then aligned to the selected orientation for extracting the SURF descriptor from it. Each detected SURF keypoint is associated with a 64 dimensional vector descriptor making it faster when compared with a 128 dimensional SIFT descriptor [Bay et al. 2008].

3.1.2 Feature Descriptors

For tasks like image matching, feature tracking etc., correspondences between pair of images need to be established. In such a case, feature descriptor is associated with the detected keypoint or region of interest for uniquely specifying the neighborhood of the detected feature and formulating more accurate correspondences. So, for a detected point of interest represented by a feature vector of location $\vec{x}$, scale σ and orientation θ, feature descriptor defines a neighborhood of $\vec{x}$ at corresponding σ and θ values. This feature descriptor is expected to be invariant to affine transformations to a much extent for performing reliable results under extreme changing conditions of image quality and other deformed conditions.

To analyze the characteristics of the neighborhood of the detected features, many descriptors have been proposed in the literature, image patches [Matas et al. 2004], SIFT [Lowe 2004], SURF [Bay et al.2008] etc. Image patch descriptors, however, do not hold much of the affine invariant properties, yielding results that are not enough satisfactory under varying imaging conditions. SIFT and SURF descriptor on the other hand are proven to perform well under extreme changing imaging conditions and are considered to be efficient and affine invariant to a great extent [Gauglitz et al. 2011].

SIFT descriptor: SIFT descriptor [Lowe 2004] starts with evaluating gradient magnitude gm and orientation r in a region around every detected keypoint. For example: Take d as the detected keypoint. The gm value is weighted using a gradient window in order to avoid sudden changes in descriptor and to give more priority to gradients located near the center of descriptor. A histogram is formulated using this weighted gm for every orientation and r_d specifies highest peak set in the histogram. A local coordinate system is determined for keypoint d using scale σ_d and orientation r_d. Respective to r_d, a new orientation histogram is evaluated from a 16×16 sample array compiled to give a 4×4 descriptor where orientation histogram for each sub-region is associated with 8 orientation bins as shown in Figure 3.1. So, the descriptor finally consists of 4×4×8 histogram values yielding a descriptor length of 128 [Lowe 2004].

SURF descriptor: SURF descriptor [Bay et al. 2008] computes orientation parameter for every extracted keypoint by considering a circular neighborhood region around it. Based on the evaluated orientation and center of interest point, a square region is constructed for descriptor extraction and is regularly split up into 4×4 sub-regions. Each sub-region is intricated with horizontal (dx) and vertical (dy) Haar wavelet responses. These two directional wavelet responses are weighed with Gaussian for pruning geometric deformation errors and are summed up for representing two dimensional feature vector

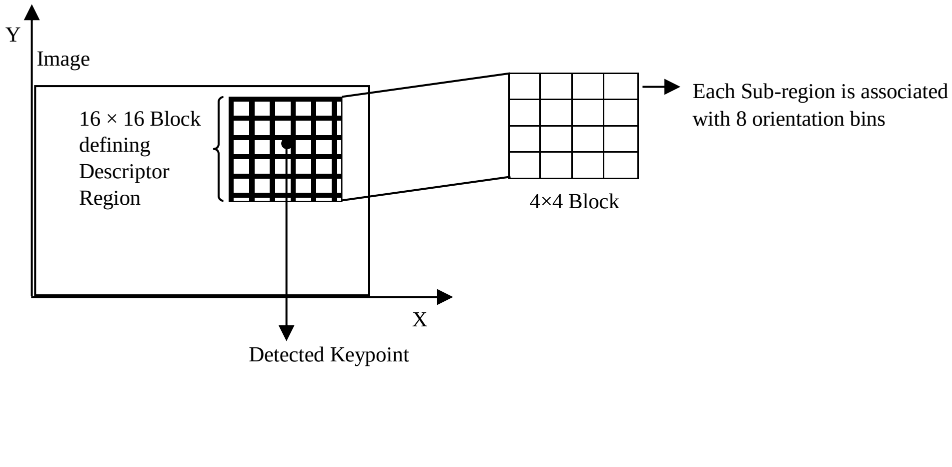

Y
Image
16 × 16 Block defining Descriptor Region
4×4 Block
Each Sub-region is associated with 8 orientation bins
X
Detected Keypoint

Fig. 3.1. SIFT Descriptor

over each sub-region. Therefore for each sub-region, following feature vector is evaluated:

$$[\,\Sigma dx, \Sigma dy, \Sigma|dx|, \Sigma\,|dy|\,] \tag{3.3}$$

where $|dx|$ and $|dy|$ are absolute values for Haar wavelet responses yielding a total SURF descriptor of length 4×4×4 = 64 dimensional vector [Bay et al. 2008].

Improvements done in the work with respect to Image Registration: In present research, focus is laid upon developing an image registration procedure for markerless AR system. In order to do that, an improved method of feature detection using MSER feature detector is designed to meet few limitations of image registration methods like high computational complexity, accurate view alignment and real time performance. Also an attempt is made to design a feature descriptor for fast and robust matching results under varied imaging conditions of viewpoint change, scale change, illumination change etc.

3.2 Metrics for comparison of performance of Image Registration Methods for Augmented Reality

Need of metrics: Metrics are used to benchmark the efficiency of an algorithm, drive improvements and to provide a standard format for comparative study between different approaches. The following subsections provide a brief introduction about the different metrics used for the research work, specifying its respective significance.

Number of Keypoints detected in an image: A keypoint in an image is a point in the image which in general differs from its immediate neighborhood, mathematically possess a well-founded definition, has a well-defined position in image space and the local image area surrounding the keypoint is affluent in terms of useful information content in terms of texture, intensity, color etc. Such keypoints in an image are expected to remain stable under local and global deformations, such as, illumination variations in a scene, scale change, blur change, affine transformations etc. [Mikolajczyk and Schmid 2005]. Therefore, in this research, number of keypoints extracted in an image by a feature detection algorithm is considered as a metric to analyze the performance of the respective algorithm in various changing imaging conditions. Note that, higher number of extracted features in an image does not solely specify the efficiency of the feature detection algorithm. The ability of the algorithm to keep the extracted keypoint count stable, even under extreme changing imaging conditions of an image, also adds up to its adeptness.

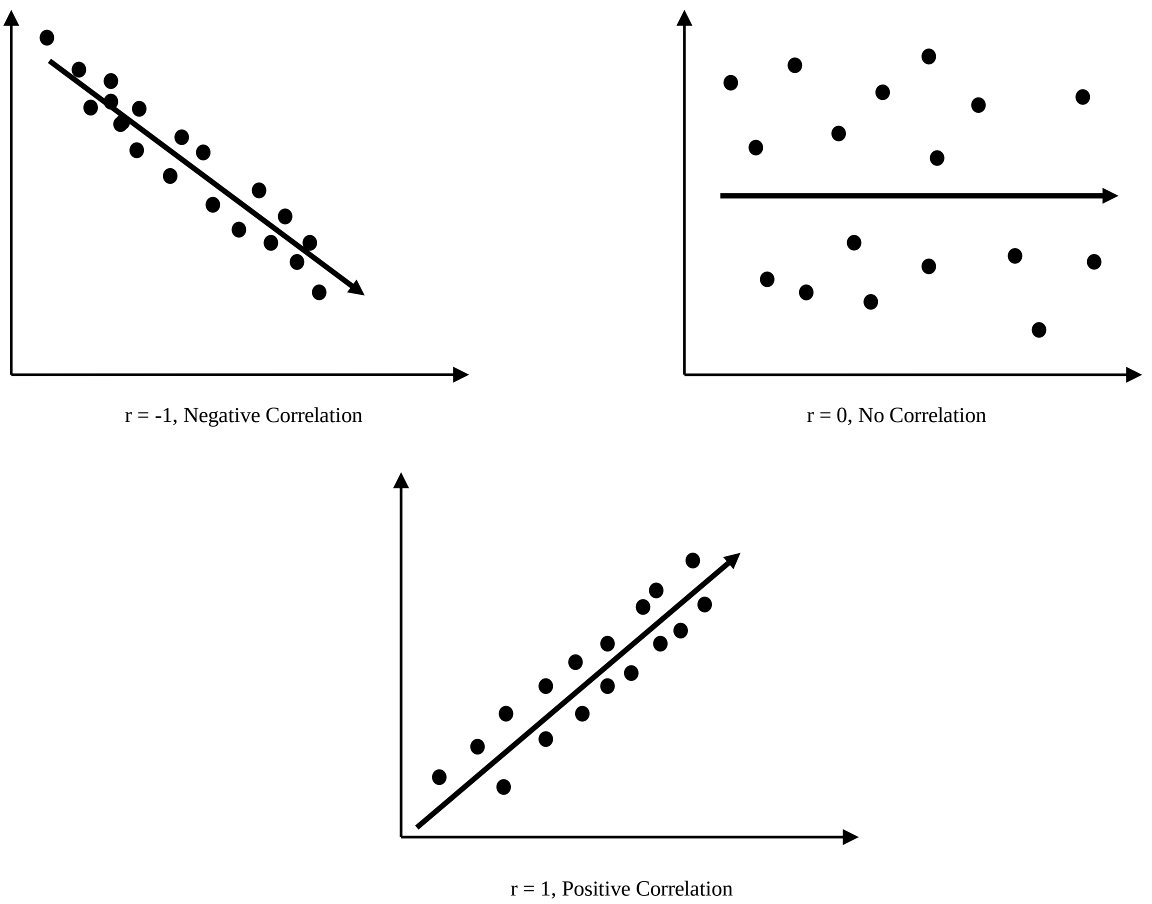
Number of correct correspondences detected between an image pair: Image matching correlates a correspondence for each detected keypoint in the reference image with a detected keypoint in the test image among a pool of detected feature points by finding the one that exhibits most similar properties.

The best candidate match for each keypoint is found by identifying its nearest neighbor in the database of extracted keypoints from the test image. The nearest neighbor is defined as the keypoint with minimum Euclidean distance for the invariant descriptor vector used for defining unique and distinctive properties of keypoint surrounding [Mikolajczyk and Schmid 2005]. However, nearest-neighbor matching results in highly noisy correspondences, and thus using all correspondences would deteriorate the whole performance of high-level computer vision tasks. Therefore, identifying correct number of correspondences between image pairs becomes an essential effort. Also, certain imaging conditions like viewpoint change, added blur or decreased illumination, object occlusion etc. may also lead to few number of correspondences between two images. Therefore, efficiency of an image registration or image matching procedure can be correlated with the amount of correct correspondences computed by it under such changing environments.

Computational Complexity: In the present research, complexity of an algorithm is quantified by the amount of time taken by an algorithm to execute. Time complexity is usually computed by keeping in track the number of fundamental operations performed by an algorithm, assuming that a respective fundamental operation consumes a fixed amount of time to execute [Mikolajczyk and Schmid 2002, Mikolajczyk and Schmid 2005]. Therefore, the amount of time taken and the number of fundamental operations performed by an algorithm differ by at most a constant factor.

Since an algorithm's performance time may vary with different inputs of the same size, one commonly uses the worst-case time complexity of an algorithm, denoted as $T(n)$, which is defined as the maximum amount of time taken on any input of size n. Infrequent, and usually specified explicitly, is the measure of average-case complexity. Time complexities are classified by the nature of the function $T(n)$. For instance, an algorithm with $T(n) = O(n)$ is called a linear time algorithm, and an algorithm with $T(n) = O\left(n \log^k n\right)$ for some positive constant k is called a quasilinear time algorithm [Moreels and Perona 2007].

Pearson Coefficient: Pearson coefficient shows the linear relationship between two sets of data. Pearson coefficient evaluates the statistical correlation between two values, determining the strength between the two. In case of images, Pearson coefficient is calculated between image a and image b, where a and b technically specifies the matrices or vectors of same size. The coefficient value ranges between -1.00 and 1.00 (Figure 3.2). Negative range specifies negative correlation between the two values, i.e. if one value increases, the other decreases. Positive range specifies positively correlation, i.e. both values increase or decrease together.



r = -1, Negative Correlation

r = 0, No Correlation

r = 1, Positive Correlation

Fig. 3.2. Pearson Coefficient

Pearson coefficient is a widely used method in image processing, computer vision, statistical analysis etc. for purposes like pattern recognition, disparity measurement or understanding the degree of similarity between two images. The method is formulated as:

$$r = \frac{\Sigma_i(x_i-x_m)(y_i-y_m)}{\sqrt(\Sigma_i(x_i-x_m)^2)\sqrt(\Sigma_i(y_i-y_m)^2} \qquad (3.4)$$

where x_i is the intensity of i^{th} pixel in image1 and y_i is the intensity of i^{th} pixel in image2. Variable x_m represents the mean intensity of image1 and y_m represents the mean intensity of image2 [Yen and Johnston 1996]. In this research, Pearson Correlation is used to understand the correlation between every pair of image in each image-set used for experimentation.

Precision: The number of correct elements (TP) relative to the total number of retrieved elements $(TP + FP)$ in a collection of data is represented by Precision and is given as:

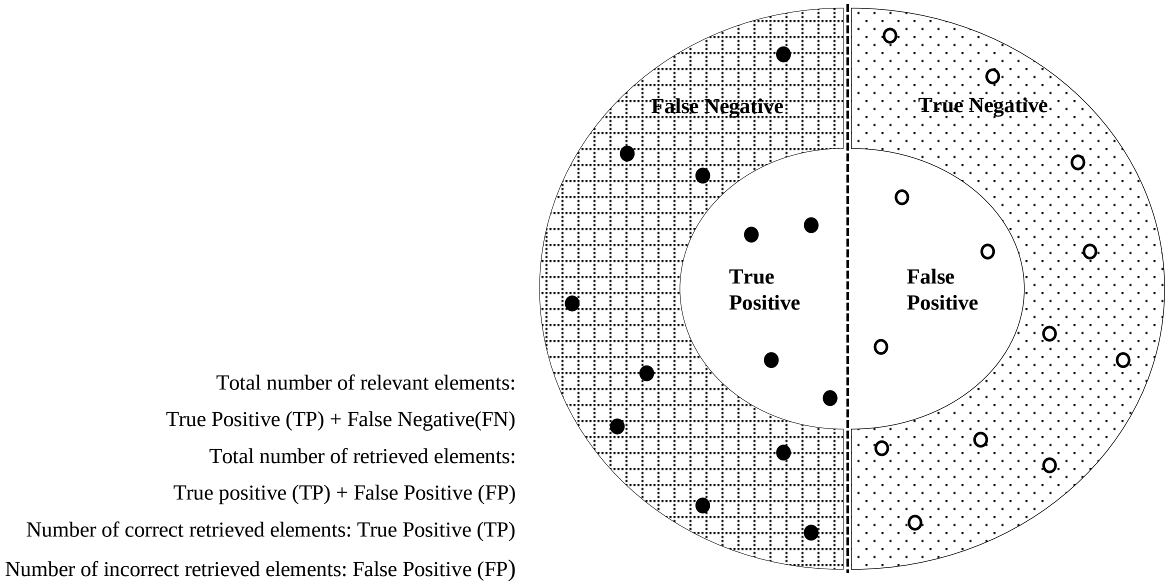

False Negative
True Negative
True Positive
False Positive
Total number of relevant elements:
True Positive (TP) + False Negative(FN)
Total number of retrieved elements:
True positive (TP) + False Positive (FP)
Number of correct retrieved elements: True Positive (TP)
Number of incorrect retrieved elements: False Positive (FP)

Fig. 3.3. Precision Formulation

$$Precision = \frac{TP}{TP+FP} \tag{3.5}$$

where TP corresponds to number of correct retrieved elements and $TP + FP$ corresponds to total number of retrieved elements (Figure 3.3). Therefore, Precision metric attempts to determine the proportion of positive identifications in the retrieved data which are actually correct.

3.3 Variations in Imaging Conditions

Translation: Two-Dimensional (2D) translation transformation in an image is defined as a basic repositioning of an object along a straight-line path from one location to another. Formally, translation of a point in 2D space with coordinate position (x, y) is specified by adding translation distances dx and dy to the original position. Therefore, translated position (x', y') of the point is defined as:

$$x' = x + dx\ ,\ y' = y + dy \tag{3.6}$$

where, added distance (dx, dy) to the original point position (x, y) is usually termed as translation vector. Similarly, in three-dimensional (3D) space, the object is repositioned by considering the 3D translation vector, which determines the relocation of the object in each of the three coordinate directions. Therefore, if $I(x, y, z)$ represents the initial position of a point, then its translated position $I'(x', y', z')$ is given as:

$$x' = x + dx\ ,\ y' = y + dy\ ,\ z' = z + dz \tag{3.7}$$

Translation transformation relocates an object without deformations i.e., every point on the object is translated by the same translation vector. Example, a straight line segment is translated by applying the translation vector to each of the line endpoints. Similarly, for objects constructed by a set of polygon surfaces, each vertex of each surface is translated using the translation vector and the object is represented along the new position. Identical approach is used for translating curved objects, position of a circle or ellipse in 2D space is translated using the center coordinates. However, curves such as splines are translated by determining and translating the coordinate positions defining the object.

Scaling: Scaling transformation leads to the expansion or reduction of the size of an image or a specific part of an image. Basic methods used for subsampling an image involves selection of every alternate row and column pixel values or adopting interpolation between pixel values using any statistics such as mean or average of the local pixel intensity values within a neighborhood (Figure 3.4). Image expansion methods, on the other hand, uses techniques like pixel replication i.e. replacing each original pixel by a group of pixels and interpolation i.e. adding values of the neighboring pixels in the original image (Figure 3.5).

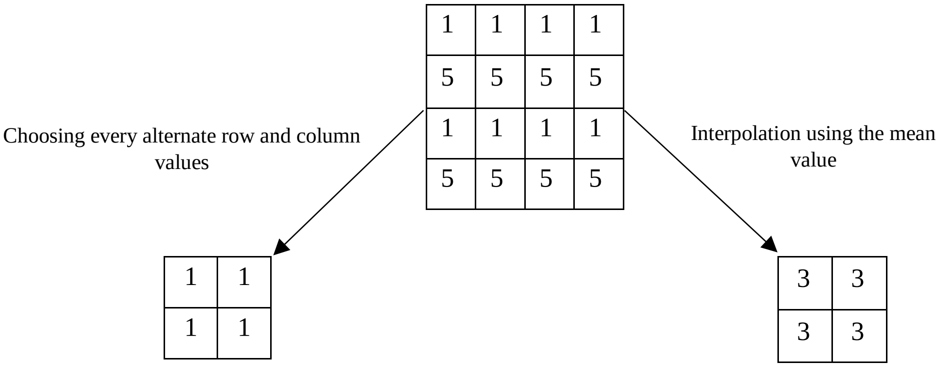
Scaling transformation also allows to modify the size of an object in an image. Such a task is carried out by multiplying the position coordinates of a point (x, y) in 2D space by scaling factors fx and fy to generate the transformed coordinates (x', y').

$$x' = x.fx \,,\; y' = y.fy \tag{3.8}$$

Mathematical significance of equation (3.8) specifies that values for scaling factors (fx, fy) defines the deformations of an object i.e. if $(fx, fy) < 1$, size of the objects reduces; $(fx, fy) > 1$ produces an enlarged object and $(fx, fy) = 1$ maintains the original size of object. Moreover, same value assignment to both the scaling factors fx and fy leads to uniform scaling of the object and distinctive value assignment for fx and fy results in differential scaling. Differential scaling is commonly used concept in design applications, where a few basic shapes are remodeled using scaling and positioning transformations**.** Similar mathematical formulation could be represented for scaling an object in a 3D space i.e. a point in 3D space with position (x, y, z) is transformed to (x', y', z') as:

$$x' = x.fx \,,\; y' = y.fy,\; z' = z.fz \tag{3.9}$$

Rotation: 2D rotation transformation applied to an object in an image scene is determined by repositioning it along a circular path in 2D plane. Degree of rotation depends upon the rotation angle θ and the point (x_p, y_p), with respect to which the object is rotated. Position of a point (x, y) after rotation

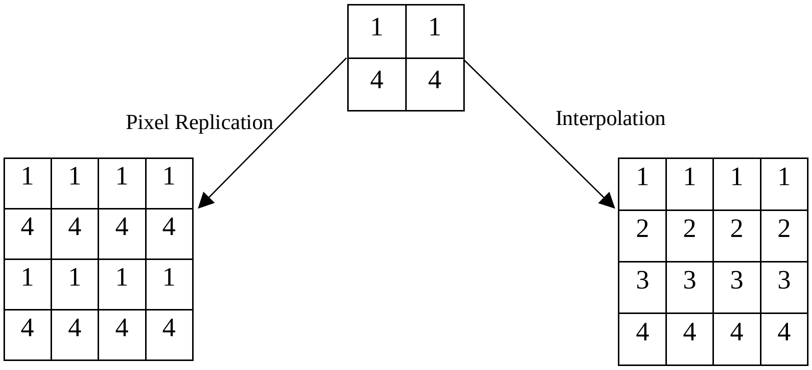

Choosing every alternate row and column values
Interpolation using the mean value

1	1	1	1
5	5	5	5
1	1	1	1
5	5	5	5

1	1
1	1

3	3
3	3

Fig. 3.4. Methods of Subsampling



1	1
4	4

1	1	1	1
4	4	4	4
1	1	1	1
4	4	4	4

1	1	1	1
2	2	2	2
3	3	3	3
4	4	4	4

Fig. 3.5. Methods of Image Expansion

transformation (x', y') is expressed as:

$$x' = x_p + \left(x - x_p\right)\cos\theta - \left(y - y_p\right)\sin\theta,$$

$$y' = y_p + \left(x - x_p\right)\sin\theta - (y - y_p)\cos\theta \qquad (3.10)$$

Positive value of rotation angle θ determines anticlockwise rotations about (x_p, y_p), and negative value of rotation angle θ determines clockwise direction. Every 2D rotation transformation is defined in the xy plane and is evaluated with respect to the point of rotation. However, in a 3D space, these computations are performed with respect to any line parallel to any of the coordinate axis. For example, rotation transformation of a point (x, y, z) in 3D space along x axis is given as:

$$x' = x,\ y' = y\cos\theta - z\sin\theta,\ z' = y\sin\theta + z\cos\theta \qquad (3.11)$$

Along y axis is given as:

$$x' = z\sin\theta - x\cos\theta\,,\ y' = y,\ z' = z\cos\theta - x\sin\theta \tag{3.12}$$

And, along z axis is given as:

$$x' = x\cos\theta - y\sin\theta\,,\ y' = y\cos\theta + x\sin\theta,\ z' = z \tag{3.13}$$

A common property between translation and rotation transformation is that both the transformations are rigid-body transformations that transforms the shape and structure of an object without deformation. For example, polygons are transformed by formulating its each vertex through the specified rotation angle and regenerating the polygon using the new vertices. Similarly, curved surfaces are transformed by repositioning the points defining the curved surface.

Illumination: Illumination is defined as the amount of source light that incidents on a scene. Image Scene illumination characteristics speculates the adaptability and efficiency of various computer vision applications. Current digital format of imaging system classifies the technology based on Charge-Coupled Device (CCD) or Complimentary Metal-Oxide Semiconductor (CMoS). These image sensors could be considered as a 2D array of a large number of solar cells, where each solar cell transforms the light from one small portion of the image into electrons [Kakadiaris et al. 2010]. Such image capturing digital devices work in a manner very similar to that of human visual system, however, the impact is different as human visual system is able to recognize different patterns of color constantly, regardless of the luminance value in the surrounding. Conway & Livingstone [Conway & Livingstone 2006] illustrated a phenomena that could represent the difference between the impression of an image captured by the human visual system and a digital sensor. Conway & Livingstone, in their theory explained that humans with normal visual capabilities are able to remember the color of an object e.g., the color of a leaf on a tree is always green, irrespective of the luminance condition of the surrounding i.e., color of the leaf remains green in the morning, at the noon, or in the dust of sunset. However, for digital image sensors, the captured color of the leaf heavily depends upon the luminance value of the surrounding. Thus, making it difficult to process images captured in different illumination conditions, for performing tasks like object recognition, image matching, visual tracking etc.

Blur: The process of blurring an image involves the reduction of edge content in the image, making the transition from one color to the other very smooth. There are two common types of blur present in an image: 1) near-isotropic blur i.e. out-of-focus blur and 2) directional motion blur. Blur in an image, as described by Liu et al. [Liu et al. 2008] could be analyzed by considering the visual and spectral clues from images. These parameters could be described as following features in an image:

• Local Power Spectrum Slope: Blurred images usually have a steeper amplitude spectrum slope, as high frequency elements tends to vanish due to low-pass filtering of a blurred region

• Gradient Histogram Span: Gradient magnitude dissemination helps in analyzing the presence of blur in an image. Blurred regions in an image usually have lower value for gradient magnitude than that for other regions.

• Maximum Saturation: Blurred regions in an image usually tend to exhibit dull colors. Therefore, maximum saturation value of blurred regions is comparably lower than the unblurred regions.

• Local Autocorrelation Congruency: Formal definition for local autocorrelation function is given as the measure of how well a signal matches to a time-deviated version of itself [Liu et al. 2008]. Therefore, in an image, local autocorrelation can be explained as a measure of how well a local window in a blurred region matches a spatially deviated version of itself, i.e. motion direction between an object and its background interprets the kind of blur present in an image region.

Presence of such deformations in an image leads to unsatisfactory results for various computer vision applications such as visual tracking, aiming to establish accurate correspondences between predefined regions over a sequence of images.

Affine: Given a point in a 2D space with position coordinates as (x, y), then its affine transformed position coordinates (x', y') could be given as:

$$x' = v_{11}x + v_{12}y + u_1, \quad y' = v_{21}x + v_{22}y + u_2 \tag{3.14}$$

Where, parameters $v_{11}, v_{12}, v_{21}, v_{22}$ and u_1, u_2 are constants determined by the transformation type. Equation (3.14) implies that for two different views of on object, known with point coordinates (x, y) and unknown with point coordinates (x', y') and (x', y') could be determined as a linear combination of the corresponding points in the known view [Bebis et al. 1999]. Thus, making it possible to attain desired affine transformed views of the same object by manipulating the parameters of affine transformation.

Transformations such as translation, rotation, scaling, reflection, and shear serves as illustrations of affine transformations. Any general affine transformation can always be signified as a composition of these five transformations. Another affine transformation is the conversion of coordinate depictions from one reference system to another, which can be described as a combination of translation and rotation transformations. Affine transformations have a specific characteristic that all the parallel lines in an image plane are transformed to parallel lines and finite points are mapped to finite points after transformation. Also, as rotation, translation, and reflection transformation are considered to be rigid transformation, therefore, any affine transformation that engages these three transformations retains the angles, lengths and parallel lines in the image plane.

Challenges to Augmented Reality Systems due to varying Imaging Conditions: AR systems, when deal with outdoor or indoor environments, are expected to work well with real-world conditions like illumination, scale, rotation, viewpoint change and other transformations as discussed above. These conditions effects the overall output of the final augmented display in terms of incorrect alignment of virtual objects in the real scene, thereby giving an undesired and implausible display. For example, transformation of an image scene in any form may cause occlusion or blocking of the target feature or object that is being tracked in reference to the reference image, thereby causing augmentation error. Other imaging conditions like low or improper illumination, increased blur due to motion or defocus aberration, also makes it difficult to track and recognize the target in an image frame. Moreover, presence of such transformations in an image scene increases the overall computational overhead of image registration methods and makes it vital for the AR system to process on algorithms that could deal with such extreme conditions.

3.4 Image Quality Metrics

Details of all the metrics used: Digital images go through different forms of distortions during acquisition, processing, compression, storage, transmission, reproduction etc., which sometimes results in degraded visual quality of images. Subjective evaluation of quality of images is a well adopted method in cases when the images are ultimately to be viewed by human beings. However, in a practical scenario, subjective evaluation is generally inconvenient, time-consuming and expensive. Therefore, an objective approach for Image Quality Assessment (IQA) is adapted as a quantitative measure to automatically predict perceived image quality. Such an approach of IQA is usually used to dynamically monitor and adjust image quality, to optimize algorithms and parameter settings of image processing systems, to benchmark image processing systems and algorithms etc.

Objective image quality metrics are categorized based on the availability of a distortion-free (reference) image, with respect to which the distorted (tested) image is compared. Most of the existing approaches falls under the category of full-reference quality assessment i.e. a complete reference image is assumed to be known. However, in many practical applications, the reference image is not available, and a no-reference or “blind” quality assessment approach becomes desirable. There are even cases when the reference image is only partially available, e.g., in the form of a set of extracted features made available as side information to help evaluate the quality of the distorted image. Such a case is referred to as reduced-reference quality assessment.

In present research work, only No-Reference Image Quality Assessment (NR-IQA) and Full-Reference Image Quality Assessment (FR-IQA) is considered. Since the Reduced-Reference Image Quality Assessment (RR-IQA) metrics predicts the perceptual quality of a distorted image with only partial information of the reference image [Wang and Bovik 2011], the present work has not used RR-IQA metrics because of the following reasons: 1) The present work assumes that the images are perceptually acceptable for AR systems. 2) The predictive models used for RR-IQA require much computational time, making it less suitable for an AR system which usually executes in real-time. 3) The work done mainly focuses on understanding the behavior of feature detectors with respect to two parameters: 1) Keypoint detection and 2) Feature matching, and for these two parameters, NR-IQA and FR-IQA metrics met the needed requirements.

3.4.1 No-reference Image Quality Assessment

Availability of a reference image for performing IQA in various practical applications seems a bit improbable, leaving NR-IQA as the only possible method that can be practically embedded into such application systems. As NR-IQA involves quality evaluation of an image based on only test image, in such a scenario, quality quantification becomes a bit difficult.

In present research, Spatial and Spectral entropies based IQA (SSEQ), Naturalness Image Quality Evaluator (NIQE), Blind/Reference-less Image Spatial Quality Evaluator (BRISQUE) and BLind Image Integrity Notator using Discrete Cosine Transform (DCT) Statistics-II (BLIINDS-II) Index, NR_IQA metrics are used for quality assessment of images.

SSEQ: SSEQ is designed as a general-purpose quality assessment method based on training and learning. It is a two-stage framework where the distortion classification is followed by quality assessment. Support Vector Machine (SVM) is used for training the image distortion and quality evaluation engine. It incorporates local spatial and spectral entropy features of distorted images for understanding the kind of distortion present in a particular image [Liu et al. 2014]. Liu et al. [Liu et al. 2014] illustrates that image entropy signifies the useful content within an image and when evaluated over multi scales of an image, image entropy provides the statistical deterioration of scale space. Type and amount of distortion present in an image usually affects the local entropy of an image. The authors of [Liu et al. 2014] based the formulation of SSEQ method on a hypothesis that the statistical properties contained by local entropy of undistorted images, are generally the result of the dependency between the neighboring pixels. Presence of any kind of distortion tends to hinder the dependency and thereby resulting in change of local entropy.

SSEQ make use of entropies evaluated from local image blocks corresponding to spatial scale responses and DCT coefficients. Spatial entropy, is therefore defined as a function of the probability distribution of the local pixel values, while spectral entropy is a function of the probability distribution of the local DCT coefficient values. Spatial entropy (E_L) is defined as:

$$E_L = -\sum_i f(i) \log_2 f(i) \tag{3.15}$$

where, i represents the number of pixel values in an image block, $f(i)$ corresponds to empirical probability density. However, Spectral entropy directly relates to the type and degree of distortion present in an image. Local spectral entropy (E_D) is evaluated by first producing a spectral probability map $M_{(x,y)}$ defined by normalizing the DCT coefficients in the local image block. Therefore,

$$M_{x,y} = \frac{T(x,y)^2}{\sum_x \sum_y T(x,y)^2} \tag{3.16}$$

where, $T(x, y)$ represents the $x \times y$ DCT coefficient matrix. If a $i \times j$ image block is considered, then $1 \leq x \leq i$ and $1 \leq y \leq j$ and hence E_D is given as:

$$E_D = -\sum_x \sum_y M_{(x,y)} \log_2 M_{(x,y)} \tag{3.17}$$

Figure 3.6 describes the working framework of SSEQ, where the pipeline starts with down sampling processing of the input distorted image by a factor of two along both dimensions, enabling image analysis

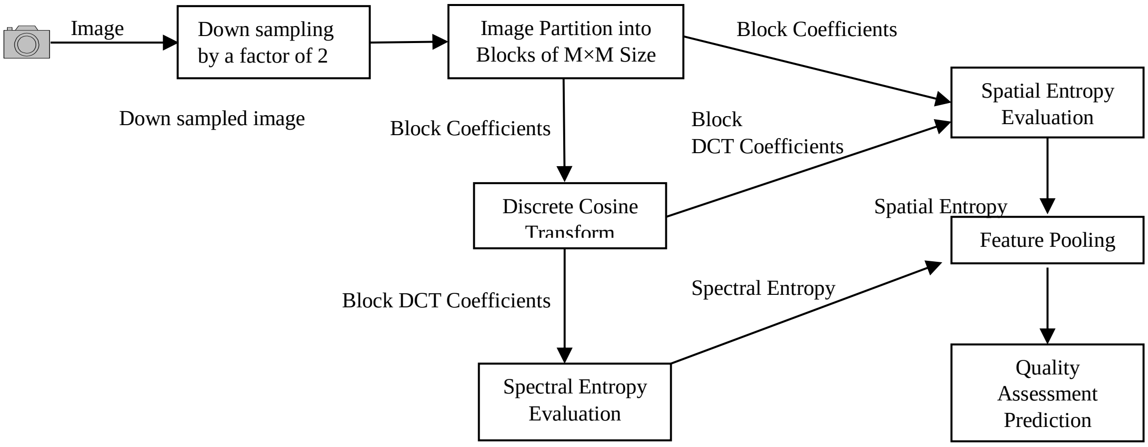


Image
Down sampling by a factor of 2
Image Partition into Blocks of M×M Size
Block Coefficients
Spatial Entropy Evaluation
Down sampled image
Block Coefficients
Block DCT Coefficients
Discrete Cosine Transform
Spatial Entropy
Feature Pooling
Block DCT Coefficients
Spectral Entropy
Spectral Entropy Evaluation
Quality Assessment Prediction

Fig. 3.6. SSEQ Framework

at a multiscale level. The second step divides the image into M×M blocks and each block is referred as an image patch. Now, spatial and spectral entropy computation within each image patch in done. The next step now scrolls down to feature pooling, where the two feature sets, spatial and spectral entropies represented as $S_{pa} = (pa_1, pa_2, \dots, pa_n)$ and $S_{pe} = (pe_1, pe_2, \dots, pe_n)$ respectively, are sorted in ascending order, where n represents the number of blocks per scale.

Percentile Pooling [Moorthy and Bovik 2009] is adopted for extracting $SC_{pa} = \left(pa_{\lfloor 0.2n \rfloor}, pa_{\lfloor 0.2n \rfloor+1}, \dots, pa_{\lfloor 0.8n \rfloor}\right)$ and $SC_{pe} = (pe_{\lfloor 0.2n \rfloor}, pe_{\lfloor 0.2n \rfloor+1}, \dots, pe_{\lfloor 0.8n \rfloor})$, i.e. 60% of the central elements from set S_{pa} and S_{pe} respectively. Final features (fs) from each scale are evaluated as:

$$fs = \left(mean\ (SC_{pa}), skew\ (S_{pa}), mean\ (SC_{pe}), skew\ (S_{pe})\right) \tag{3.18}$$

NIQE: Given the impracticality of obtaining collections of distorted images with defined human scores, Opinion-unaware models that do not require training on databases of human judgments of distorted images forms a more fundamental basis for determining image quality. As such, NIQE is designed as an Opinion-Unaware and Distortion-Unaware, NR-IQA model that constructs a set of 'quality aware' features using Natural Scene Statistics (NSS) model and then uses Multivariate Gaussian (MGV) model fitting for expressing the quality of the test image [Mittal et al. 2013].

Local image patches capturing essential low order statistics of natural images are used for extracting perceptually relevant spatial domain NSS features. The image is first processed for local mean removal and divisive normalization and the resulting image $\hat{P}(x, y)$ is given as:

$$\hat{P}(x, y) = \frac{P(x,y) - \mu(x,y)}{\sigma(x,y) + 1} \tag{3.19}$$

where $x \in \{1,2,3 \dots, I\}, y \in \{1,2,3, \dots, K\}$ represents spatial indices for $I \times K$ dimensional image. The mean $(\mu(x, y))$ and contrast $(\sigma(x, y))$ values for the image are calculated as:

$$\mu(x, y) = \sum_{m=-M}^{M} \sum_{n=-N}^{N} g_{m,n} P(x + m, y + N) \tag{3.20}$$

$$\sigma(x, y) = \sqrt{\sum_{m=-M}^{M} \sum_{n=-N}^{N} g_{m,n} [P(x + m, y + N) - \mu(x, y)]^2} \tag{3.21}$$

where $g = \{g_{m,n} | m = -M, \dots, M, n = -N, \dots, N\}$ representing a 2D circularly symmetric Gaussian weighting function. After obtaining the image coefficients using equation (3.19), the image is partitioned into T×T patches. Among these collection of patches, only those patches that are richest in information and are less likely to have been subjected to a limiting distortion are used for evaluating NSS features form the computed coefficient of each patch. The quality of the given test image is then expressed as the distance between MVG fit of the NSS features extracted from the test image, and a MVG model of the quality aware features extracted from the corpus of natural images [Mittal et al. 2013].

BRISQUE: BRISQUE design is based on the principle that natural images possess certain regular statistical properties that are determinable in the presence of distortions. For example, it is observed that the normalized luminance coefficients of the natural image closely follows Gaussian-like distribution while the same is not true for artificial images. BRISQUE is built as an Opinion-Aware and Distortion-Aware, NR-IQA model and is considered as statistical model of locally normalized luminance coefficients in the spatial domain. The NSS features used in BRISQUE are similar to those used by NIQE. However, NIQE only uses the NSS features from a corpus of natural images while BRISQUE is trained on features obtained from both natural and distorted images and also on human perception of the quality of these images [Mittal et al. 2011, Mittal et al. 2012].

The pre-processing strategy depicted by equation (3.19) is used and $\hat{P}(x,y)$, the transformed luminance is determined as Mean Subtracted Contrast Normalized (MSCN) coefficients. MSCN coefficients have characteristic statistical properties that are disturbed and are determinable by the presence of distortion in the image and measure of such changes is used to predict the type of distortion affecting an image as well as its perceptual quality. For final quality evaluation, a mapping is learned from feature space to quality scores using a regression module. BRISQUE framework uses SVM regressor (SVR) [Schölkopf et al. 2000].

BLIINDS-II: BLIINDS-II is a single-stage framework that relies on a statistical model of local DCT coefficients. It extracts a set of features using a NSS model of local DCT coefficients and then uses a simple Bayesian approach to predict quality scores. The NSS model parameters are used to design features suitable for perceptual image quality score prediction [Saad et al. 2012].

The framework of BLIINDS-II can be expressed as a four step approach as represented in Figure 3.7. The initial step involves partitioning of the test/distorted image into $N \times N$ local image patches and then computing DCT coefficient for each patch. The coefficient extraction is performed locally in the spatial domain in accordance with the Human Visual System (HVS) property of local spatial visual processing. Next step focuses on computing directional information from image patch by dividing DCT block into three oriented sub-regions and then estimating a generalized Gaussian fit for each sub-region. Third step deals with generalized Gaussian model based feature extraction and finally in the fourth step a Bayesian

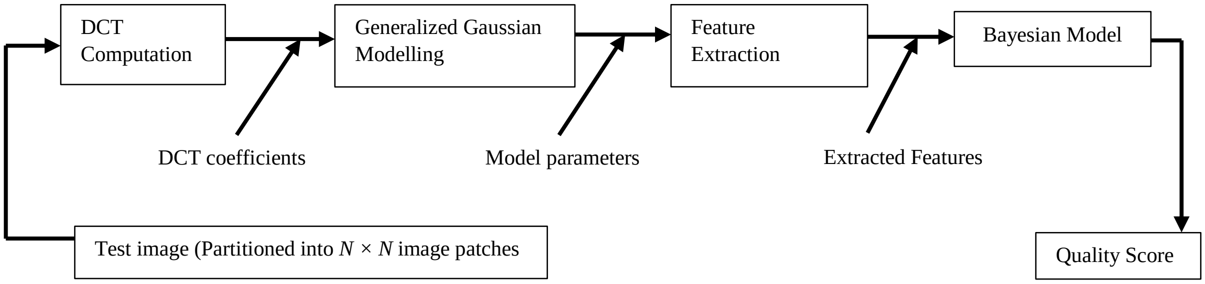


Fig. 3.7. BLIIND-II Framework

model is used for predicting the quality score.

3.4.2 Full-reference Image Quality Assessment

FR-IQA methods involve a reference image that is considered to be of an acceptable quality of an image, and hence, the quality quantification of the deformed image is done with respect to this reference image. Present research involves four FR-IQA metrics, Structure SIMilarity Index (SSIM) and Multi-Scale Structure SIMilarity Index (MS-SSIM), Mean Square Error (MSE) and Normalized Cross-Correlation (NK) for quality assessment of images.

For mathematical formulation of these two methods, we consider, $a = \{a_i | i = 1,2, \dots, z\}$ as the reference image and $b = \{b_i | i = 1,2, \dots, z\}$ as the test image, where z is the number of pixels in each image.

SSIM: SSIM method make use of structural information from a scene to match human visual system perspectives. This method is based on an assumption that a good quality image demonstrates strong dependencies among pixels and recognized change in structural content of an image could be related to image degradation. Also, SSIM method indicates that the structural information of an image could only be gathered if it is quantified independently from the local luminance and contrast factors of the image. Therefore three terms are considered for SSIM evaluation: 1) Luminance: average pixel intensity, 2) contrast: variance between two images and, 3) structure: cross-correlation between two images [Wang et al. 2004]. Mathematically, SSIM index can be defined as:

$$SSIM = [L(a,b)] \, . [C(a,b)]. [S(a,b)] \tag{3.22}$$

where, $[L(a,b)]$, $[C(a,b)]$ and $[S(a,b)]$ are the respective luminance, contrast and structure comparison measures given as:

$$L(a,b) = \frac{2\mu_a\mu_b+x_1}{\mu_a^2+\mu_b^2+x_1}, \tag{3.23}$$

$$C(a,b) = \frac{2\sigma_a\sigma_{yb}+x_2}{\sigma_a^2+\sigma_b^2+x_2}, \tag{3.24}$$

$$S(a,b) = \frac{\sigma_{ab}+x_3}{\sigma_a\sigma_b+x_3} \tag{3.25}$$

x_1, x_2, x_3 are constants given by: $x_1 = (y_1 m)^2$, $x_2 = (y_2 m)^2$ and $x_3 = \frac{y_2}{2}$ respectively, where, variable m defines the pixel value range, $y_1 \ll 1$, $y_2 \ll 1$ are two scalar constants, μ_a and μ_b determines mean of image 'a' and image 'b' respectively. Similarly σ_a and σ_b determines variance of image 'a' and image 'b' respectively.

MS-SSIM: MS-SSIM holds the ability to overcome certain drawbacks of SSIM algorithm i.e. MS-SSIM incorporates image details at different resolutions and viewing conditions of an image. As a result, a more accurate subjective evaluation between the referenced and the distorted image could be achieved [Wang et al. 2003].

The images are indexed between scales from 1 to k. The contrast $C(a,b)$ and structure $S(a,b)$ comparison at the r^{th} scale is defined as $C_r(a,b)$ and $S_r(a,b)$ respectively. Luminance comparison is only done at scale k and is denoted as $L_K(a,b)$. Therefore, overall MS-SSIM is given as:

$$MSSSIM = L_k(a,b) \prod_{r=1}^{K} [C_r(a,b)][S_{\mathrm{r}}(a,\mathrm{b})] \tag{3.26}$$

MSE: MSE [Avcibas et al. 2002] gives a quantitative score specifying the similarity or distortion score between the two images. MSE is defined as:

$$MSE(a,b) = \frac{1}{N}\sum_{i=1}^{N}(a-b)^2 \tag{3.27}$$

where N determines image size.

NK: Correlation relates to similarity attained between two input signals x and y. NK is mathematically formulated as [Eskicioglu and Fisher 1995]:

$$NK(a,b) = \left(\sum_{j=1}^{M}\sum_{k=1}^{N} a_{j,k}.\, b_{j,k}\right) / \left(\sum_{j=1}^{M}\sum_{k=1}^{N} a_{j,k}^2\right) \tag{3.28}$$

where $M\,x\,N$ corresponds to image representation, where M = number of rows and N = number of columns.

3.5 Image Quality change due to Compression

Image quality is also affected by image compression **[**Barten 2012, Moorthy and Bovik 2011]. Image compression is defined as the reduction of required bandwidth of electronically displayed images for easy transmission and storage [Barten 2012]. It is believed that human visual system does not require all bits of luminance and chrominance information that are present in the undistorted image. Therefore, it seems acceptable to reduce the number of bits per pixel. However, a too large reduction may lead to a visible loss of image quality. Image compression techniques are generally categorized to lossless and lossy formulations, i.e., whether the compressed or reconstructed image could produce the replica of the original image without information loss. Compression is achieved by eliminating one or more of the following redundancy [Sheikh et al. 2005, Sazzad et al. 2008, Barten 2012].

- Coding Redundancy: Codes assigned to pixel values of an image can be used to navigate their contribution to the efficiency of a particular task. Therefore, code that does not take part to attain full advantage of the probabilities of events is treated as a redundant value and hence, respective pixel is removed.
- Inter-pixel Redundancy: High correlation between pixels of an image make it easier to predict any given pixel from its neighboring pixel.
- Perceptual Redundancy: Some regions in an image exhibits high degree of similarity e.g., smooth region of a natural image. In such a case, minimal variation in neighboring pixels values is not evident, making the information redundant.

Lossless image compression: This technique allows the generation of perfectly recovered original image from the compressed image. It uses decomposition methods to minimize the redundancy and do not add distortions like noise to the signal, and therefore considered as a noiseless approach [Moorthy and Bovik 2011].

Lossy image compression: Lossy compression, such as JPEG compression, tends to compress an image with some loss of information. In such a case, decompression of a compressed image does provide the replica of the original image. JPEG compression uses DCT to change the pixels in the original image into frequency domain coefficients. A study conducted by Wang et al. [Wang et al. 2002], proposed a no-reference perceptual assessment of JPEG compressed images based on initial subjective evaluation. Features extracted from the compressed image were designed to describe the distortions introduced by JPEG compression. Their objective quality assessment gave satisfactory agreement to the Mean-Opinion

Score (MOS) of the subjective evaluation, thereby proving the adverse effect of compression on image quality.

In present research work, variation in quality of images due to JPEG compression is considered.

Challenges to AR systems due to Poor Image Quality: AR with its emerging wide applications holds great innovations and practicality for mobile users in the near future. However, mobile devices processing capabilities have not yet reached a considerable level to provide suitable environment for executing object recognition, tracking, or rendering methods. Moreover, dealing with low quality images also becomes difficult in such a scenario as AR system then suffers from incorrect alignment of virtual information in the real scene.

3.6 Multi-Linear Regression

Multi-Linear Regression (MLR) is a form of linear predictive regression analysis that determines the relationship between one continuous dependent variable C and two or more independent variables I. Therefore, using these notations, mathematical expression for an MLR model, given p observations is given as:

$$C_i = \beta_0 + I_{i1}\beta_1 + I_{i2}\beta_2 + I_{i3}\beta_3 + \cdots + I_{in}\beta_n + \varepsilon_i \text{ , for } i = 1,2,3,\ldots,p \tag{3.29}$$

where n defines the number of independent variables $I_1, I_2, I_3, \ldots, I_n$ through the parameters $\beta_1, \beta_2, \beta_3, \ldots, \beta_n$. The mean response for a particular observation is given as $\mu_C = \beta_0 + I_1\beta_1 + I_2\beta_2 + I_3\beta_3 + \cdots + I_n\beta_n$ and the observed values for C vary about their mean μ_C and are assumed to have the same standard deviation σ_C. Variable ε_i in equation (3.29) represents the residual error which in turn determines the deviations of the observed values of C from their μ_C.

To attain a most successful MLR model for a defined type of data, regression residuals ε_i must be normally distributed with a mean equal to zero and standard deviation equal to σ_C and a linear relationship is assumed between C (the dependent variable) and I (the independent variables). Moreover, absence of multicollinearity, i.e., disassociation between the independent variables is assumed in the MLR model. MLR analysis follows a task of fitting a single straight line through a scatter plot defined between the dependent variable, also referred as the output variable and the independent variables in a multi-dimensional space of data points.

MLR analysis is usually used for two major tasks: 1) for identify the strength of the effect that the independent variables have on a dependent variable. 2) to determine in advance how changes in independent variables effects the overall change in the dependent variable.

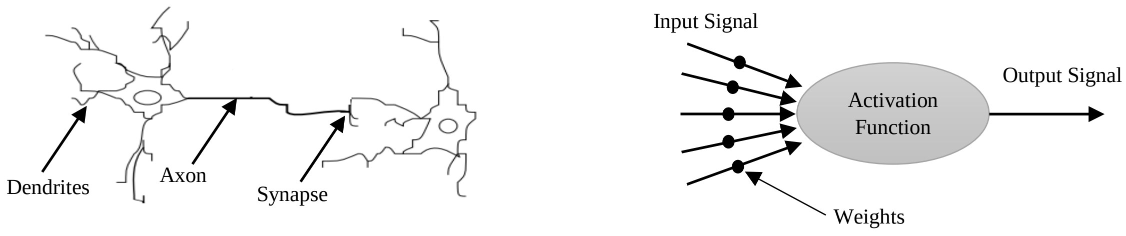
In present research work, MLR is used for designing an improved NR-IQA metric detailed in Chapter 5.

3.7 Introduction to Neural Network

Network approach is considered as one of the most efficient way of solving complex problems by decomposing it into simpler elements or by accumulating simple elements to produce a complex and efficient system to perform a certain task [Bar Yam 1997]. A network mainly comprises of a set of nodes known as computational units, and connections between nodes that determines the information flow between nodes in a unidirectional or a bidirectional manner. Global behavior of the network is a result of the interactions of nodes though the various connections and this global behavior is depicts a more powerful functionality of the network than any of its individual elements.

An Artificial Neural Network (ANN), commonly called Neural Network (NN) is based on the concept of an artificial neuron that is defined as a computational model inspired by the natural neurons. Natural neurons receive the input signal through synapses located on the dendrites or membrane of the neuron (Figure 3.8(a)). The neuron gets activated and emits an output signal through axon when the input signal received exceeds a certain threshold that in turn depicts a strong desired property of the signal. This emitted output signal is then sent to another synapse to activate other neurons. In ANN, the complexity of natural neurons is highly abstracted when modelling artificial neurons. Artificial Neurons consists of input signals that are multiplied by the weights of the respective signal and a mathematical function is computed which determines the activation of the neuron and finally evaluates the output signal (Figure 3.8(b)).

In present research work, ANN is used for designing an efficient No-Reference Video Quality Assessment (NR-VQA) metric detailed in Chapter 5.



(a) Natural Neuron **(b) Artificial Neuron**

Fig. 3.8. Neural Network

3.8 Conic Equations in General Form for an ellipse

Given a general conic equation of the form:

$$C(x,y) = Gx^2 + Hxy + Iy^2 + Jx + Ky + L = 0 \tag{3.30}$$

represents an ellipse if:

$$H^2 - 4GI < 0 \text{ or } 4GI - H^2 < 0 \tag{3.31}$$

Therefore, based on equation (3.30), various parameters of an ellipse are given as:

Coefficient Normalization factor (C_N):

$$C_N = \frac{64\left(L\left(4GI-H^2\right)-GK^2+HJK-IJ^2\right)}{4GI-H^2} \tag{3.32}$$

Distance between center of ellipse and either of the focus point (DS) : For a given ellipse, foci are two points that lie in the interior of ellipse such that the sum of the distance to each focus from a point on the ellipse is constant.

$$DS = 0.25\sqrt{|C_N|\sqrt{H^2 + (G-I)^2}} \tag{3.33}$$

Semi-Major Axis Length (MJ):

$$MJ = 0.125\sqrt{2|C_N|\sqrt{H^2 + (G-I)^2} - 2C_N(G+I)} \tag{3.34}$$

Semi-Minor Axis Length (MN):

$$MN = \sqrt{MJ^2 - DS^2} \tag{3.35}$$

Latus Rectum (L) : Latus Rectum for an ellipse is defined as a line perpendicular to ellipse major axis (the axis on which foci and vertex lie) and parallel to directix. Directix of an ellipse is a line drawn at a distance equal to focal length in the opposite direction of focus of a conic section.

$$L = 2\left(\frac{MJ^2}{MN^2}\right) \tag{3.36}$$

Eccentricity (E_C):

$$E_c = DS/MJ \tag{3.37}$$

Distance between center and closest Directix point (DD):

$$DD = (MJ^2)/DS \tag{3.38}$$

Center of ellipse (x_c, y_c):

$$x_c = (HK - 2IJ)/(4GI - H^2)$$

$$y_c = (HJ - 2GK)/(4GI - H^2) \tag{3.39}$$

The angle between x axis and ellipse major axis (θ_{xmj}) is given as:

If $(C_N G - C_N I = 0)$, $(C_N G - C_N I = 0)$ and $C_N H = 0$ then $\theta_{xmj} = 0$

If $(C_N G - C_N I = 0)$, $(C_N G - C_N I = 0)$ and $C_N H > 0$ then $\theta_{xmj} = 0.25\pi$

If $(C_N G - C_N I = 0)$, $(C_N G - C_N I = 0)$ and $C_N H < 0$ then $\theta_{xmj} = 0.75\pi$

If $(C_N G - C_N I > 0)$, $(C_N G - C_N I > 0)$ and $C_N H \geq 0$ then $\theta_{xmj} = 0.5 \tan(\frac{b}{a-c})$

If $(C_N G - C_N I > 0)$, $(C_N G - C_N I > 0)$ and $C_N H < 0$ then $\theta_{xmj} = 0.5 \tan(\frac{b}{a-c}) + \pi$

If $(C_N G - C_N I < 0)$, $(C_N G - C_N I < 0)$ then $\theta_{xmj} = 0.5 \tan(\frac{b}{a-c}) + 0.5\pi$ (3.40)

Two Focal Points $F_1(x_1, y_1)$ and $F_2(x_2, y_2)$ are given as:

$x_1 = x_c - DS \cos\theta_{xmj}$

$y_1 = y_c - DS \sin\theta_{xmj}$

$x_2 = x_c + DS \cos\theta_{xmj}$

$y_2 = y_c + DS \sin\theta_{xmj}$ (3.41)

In present research work, General Conic Equations are used for performing elliptical sampling detailed in Chapter 7.

3.9 Summary

In this chapter, conventional image registration methods are reviewed and various image quality assessment metrics used for performance comparison of image registration methods are also discussed. The chapter also details the theoretical explanation of different imaging conditions and image quality that brings challenges in designing an AR system. Image quality metrics along with their significance and use are also discussed in detail. In the next chapter, effect of image quality and varying imaging condition on image registration methods is studied and analyzed.