**M6 project – 2D Image transformation**

**Introduction (abstract):**

The objective of image capture is to project a real world scene into an image. This projection implies some distortion in the original scene, changing the perception of how we see it. In this work, we explored the three main transformations that an image can suffer and we explained how we could express those transformation with a single matrix. After the transformation analysis, we use two ways to make an inference in a real problem. First of all we compute an affine transformation of the original image and then we perform a metric rectification. In the following sections we will explain in what consist these methods and how to apply them.

**Image transformation:**

Image transformations in the 2D space will be explored in this work, so we will be changing the representation of images assuming they suffer a transformation.

Before analysing the distortion of a real image we must understand what do transformations consist of and what types of these can affect the points of an image.

Any transformation can be written as a matrix, in the case of the 2D space, this is a 3 by 3 matrix, as for example:

*Generic expression of an homography (transformation matrix)*

We must know projective space is expressed in homogeneous coordinates. To pass from Euclidean coordinates (x,y) to homogeneous coordinates, we must add 1 as the third coordinate (x,y,1). Any vector that can be expressed as the linear combination between a scalar factor ‘**s**’ and a vector (x,y,1) is equivalent to that one.

The simplest transformation that we can apply to a point (excluding the identity transformation) is the isometry. This transformation consist in applying a translation and a rotation, or just either of them, to each point on the image. Having all the points in the new position, this transformation keeps the parallelisms in the image, the lengths and the angles. Having an angle ‘alpha’ and a translation vector ‘t’ with two coordinates, the transformation matrix can be expresses as:

In the expression we have a matrix R, that is a 2 by 2 matrix, describing the rotation of the image and a translation vector ‘t’, described by the ‘x’ translation and the ‘y’ translation. The rotation matrix R (non-singular) is computed as:

Another transformation we can apply to an image is that one of a scale factor weighted rotation plus translation. This transformation is called similarity and does not keep the length of the objects in the image. The transformation matrix is very similar to the previous one but includes a scale factor that affects the rotation matrix as we can see:

As this transformation implies two basic transformations: isometry and scaling, similarity matrix can be expresses as a scalar product of two matrix: Hi (isometry matrix) and Hs (the scaling matrix):