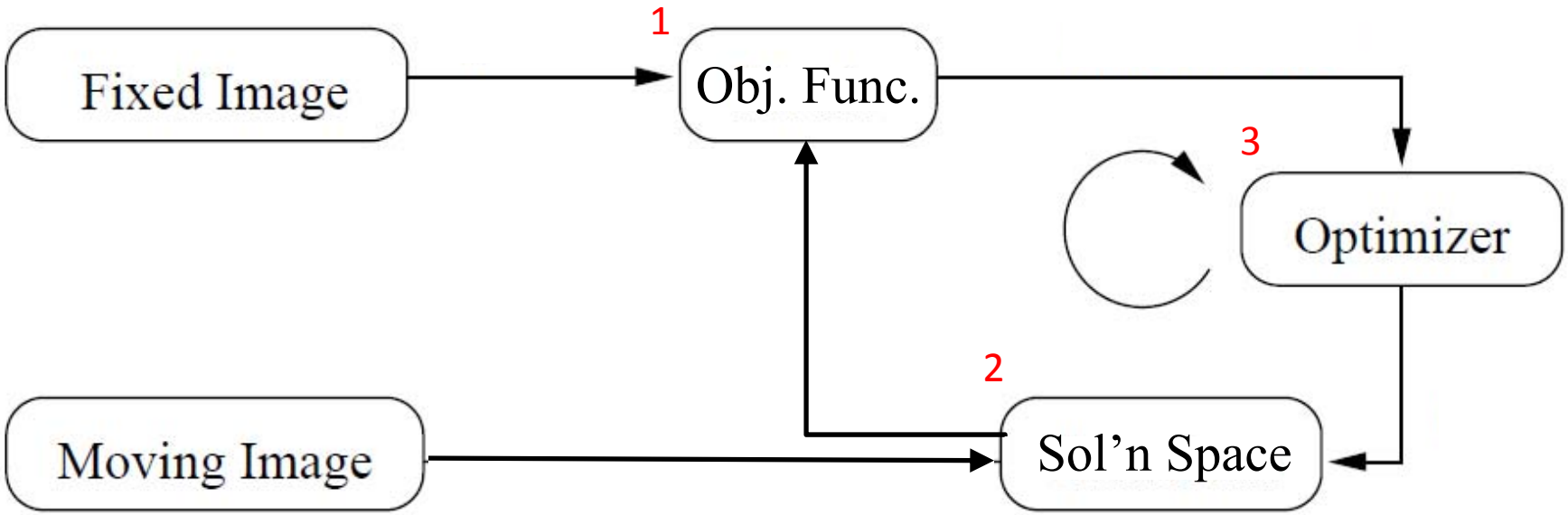
Algorithm specification

Recall that the description of an algorithm (e.g. for image registration) can have three components:

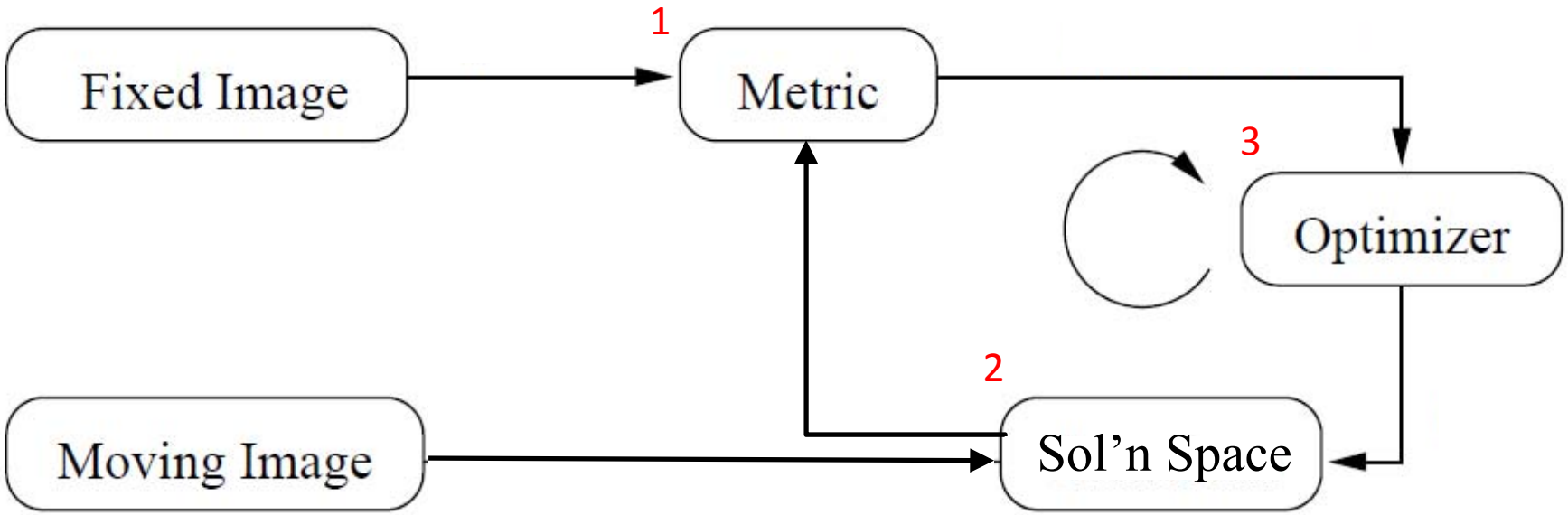
1. The *objective function*.
2. The *solution space*.
3. The *optimizer*.

An image registration framework



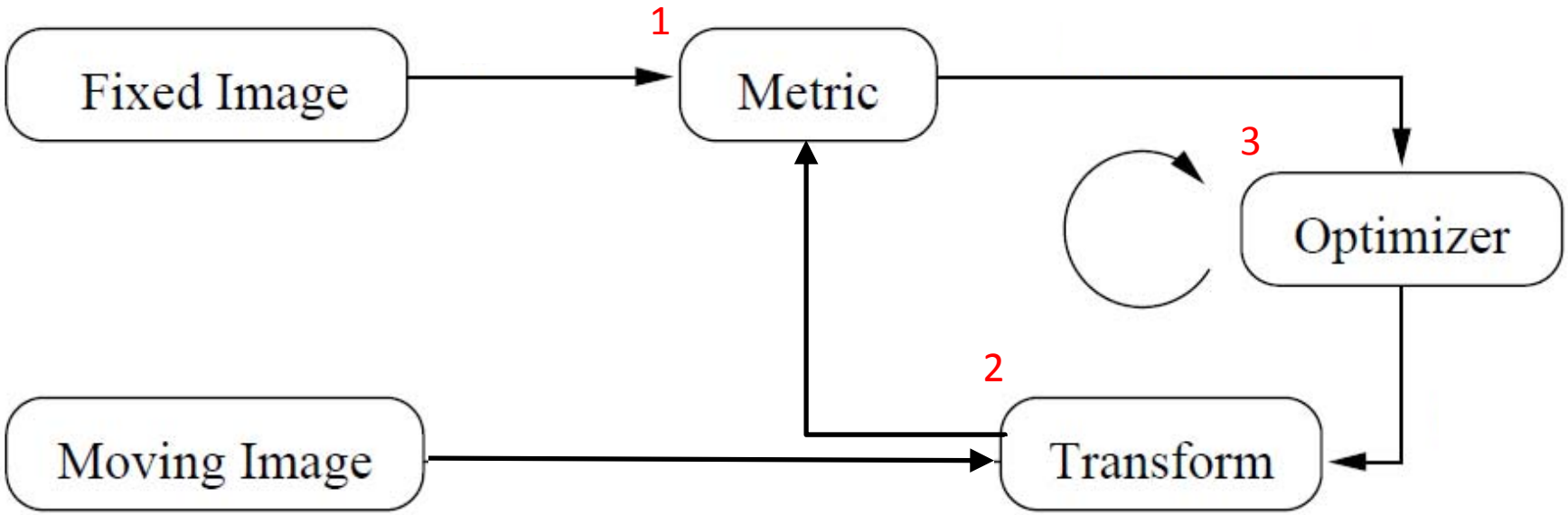
For image registration, we could arrange these three components in a loop, like this.

An image registration framework



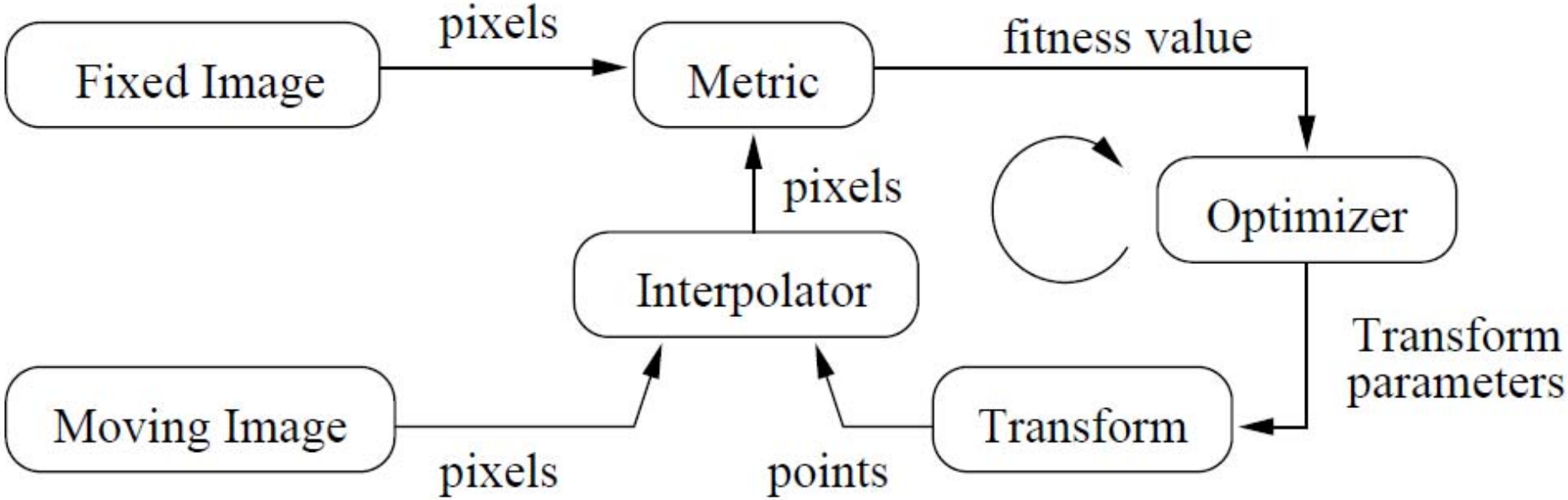
In registration, the objective function can be the image similarity metric….

An image registration framework



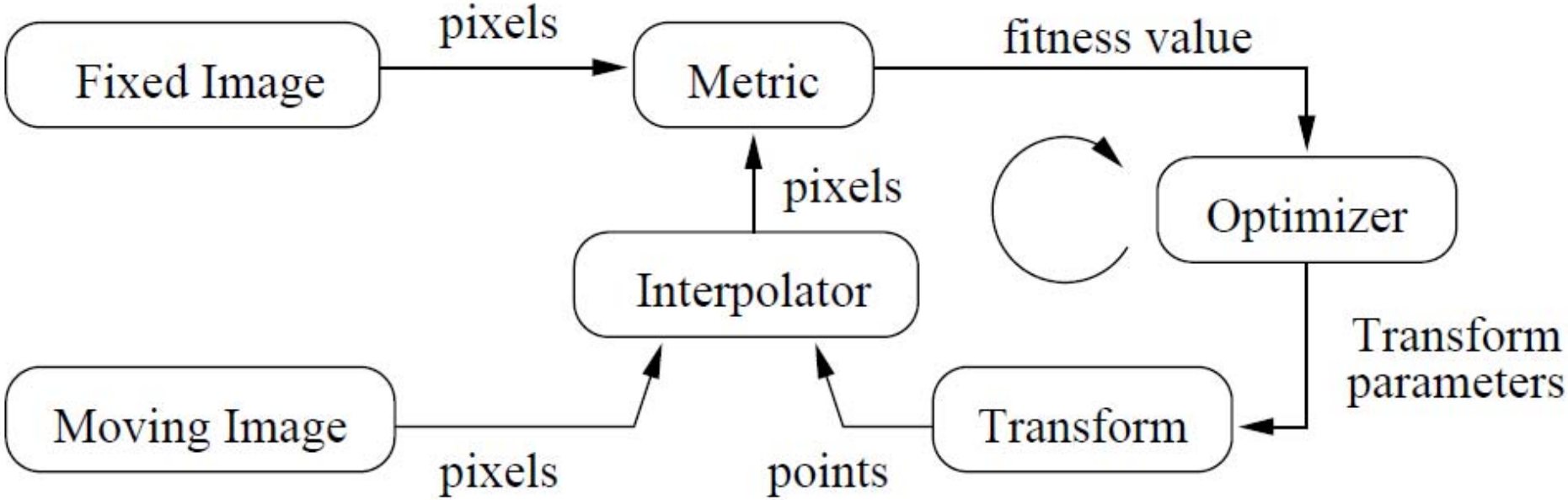
…and the solution space can be the space of possible spatial transformations of the moving image.

An image registration framework



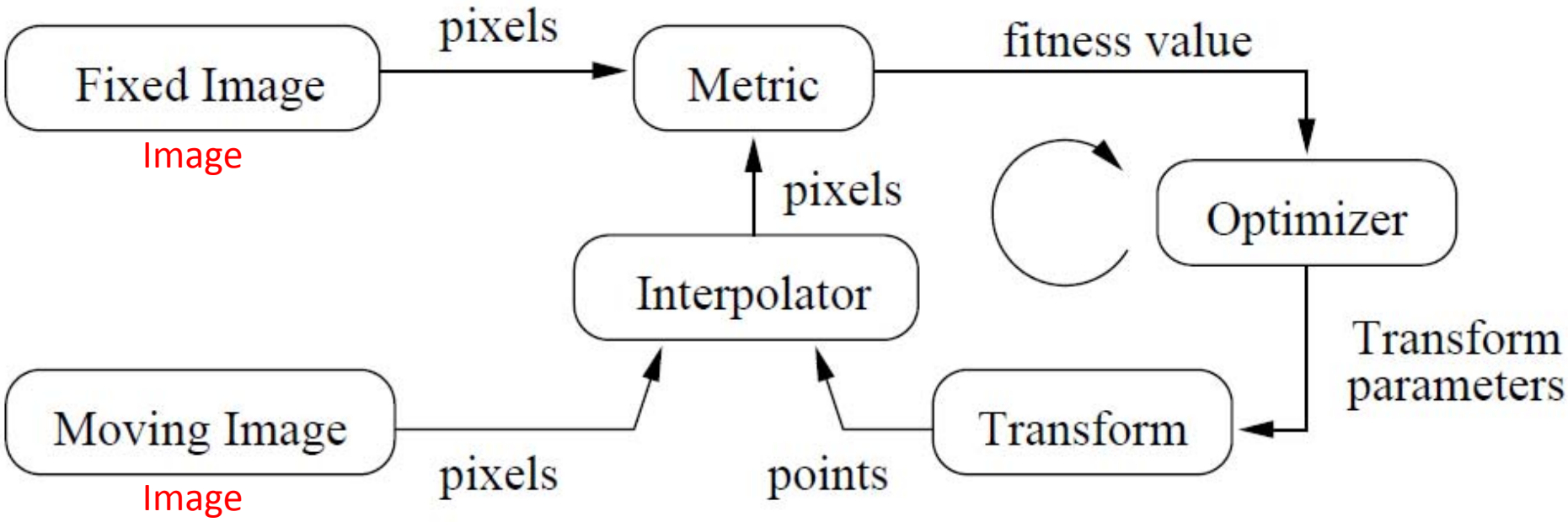
In a practical implementation, each transformation of the moving image requires a resampling step, and a specific type of interpolator needs to be used (e.g. nearest neighbour, linear, cubic, etc.)

ITK registration framework



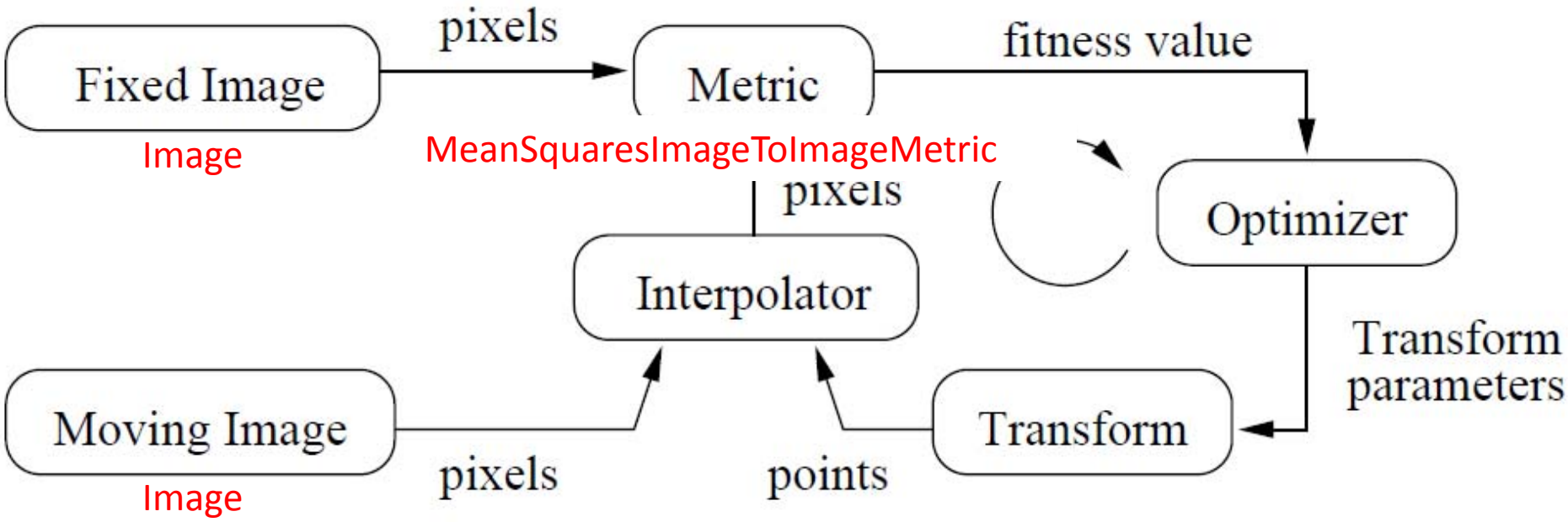
What you are seeing here is the “ITK registration loop” verbatim from the ITK software guide. We can follow this diagram like a recipe to make an ITK-based image registration tool.

ITK registration framework



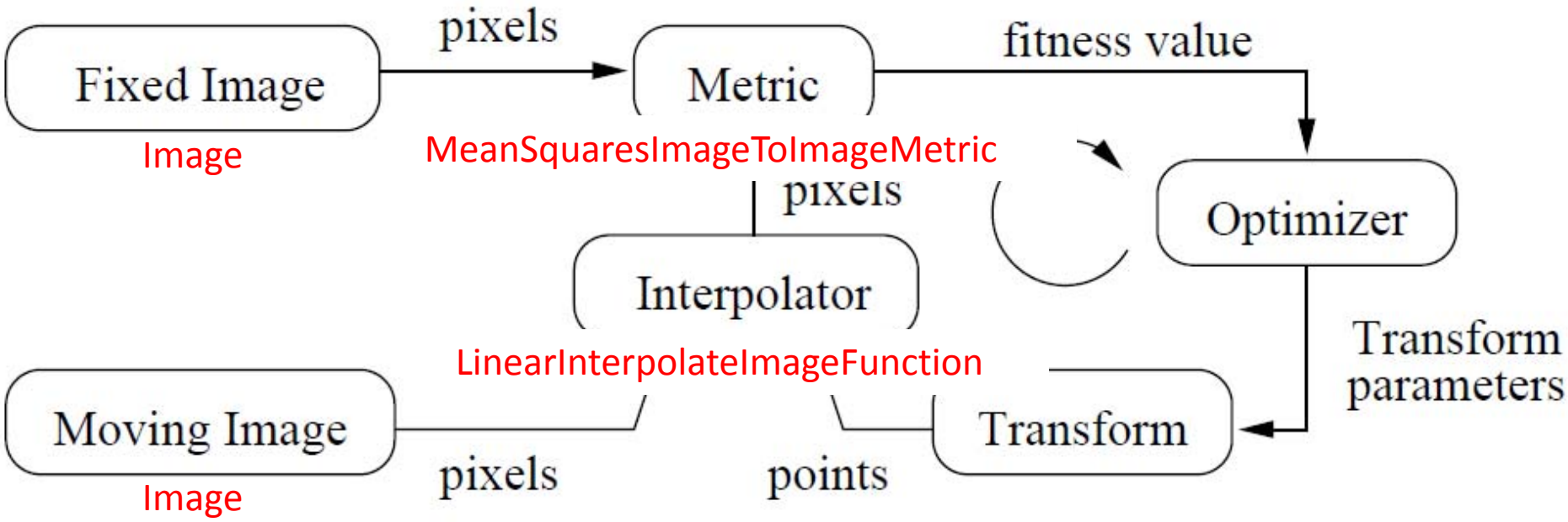
Each of the elements of this framework is implemented using a specific ITK class. The two images are represented by itk::Image.

ITK registration framework



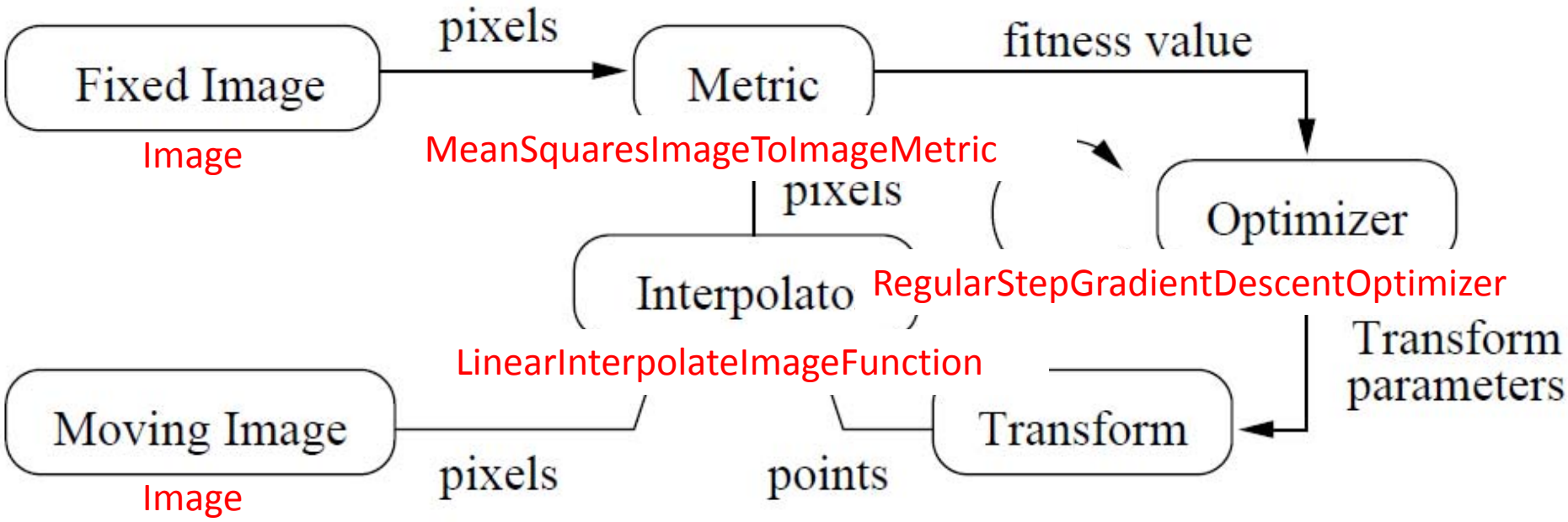
There are several image similarity metrics in ITK. We'll start with ITK's implementation of MSE, which is in itk::MeanSquaresImageToImageMetric.

ITK registration framework



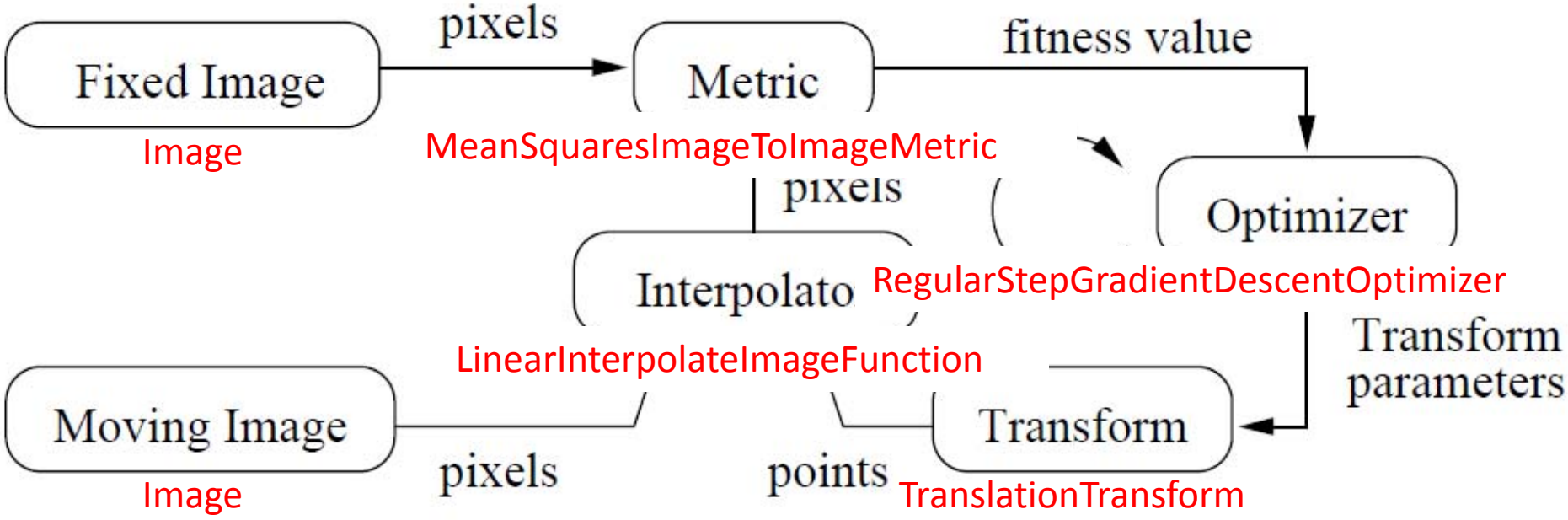
We will resample the transformed moving image using linear interpolation, implemented in itk::LinearInterpolateImageFunction.

ITK registration framework



We will use gradient descent optimization, implemented in itk::RegularStepGradientDescentOptimizer.

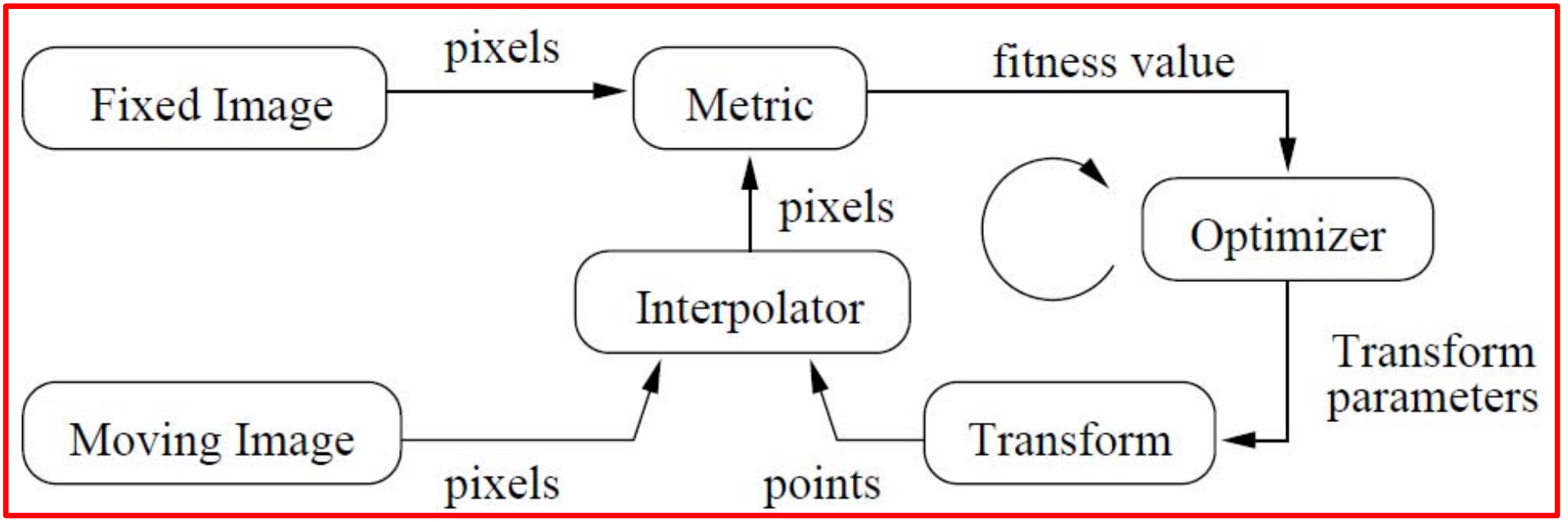
ITK registration framework



And lastly, for our first example, we'll implement X- and Y-translation only, which is implemented in itk::TranslationTransform.

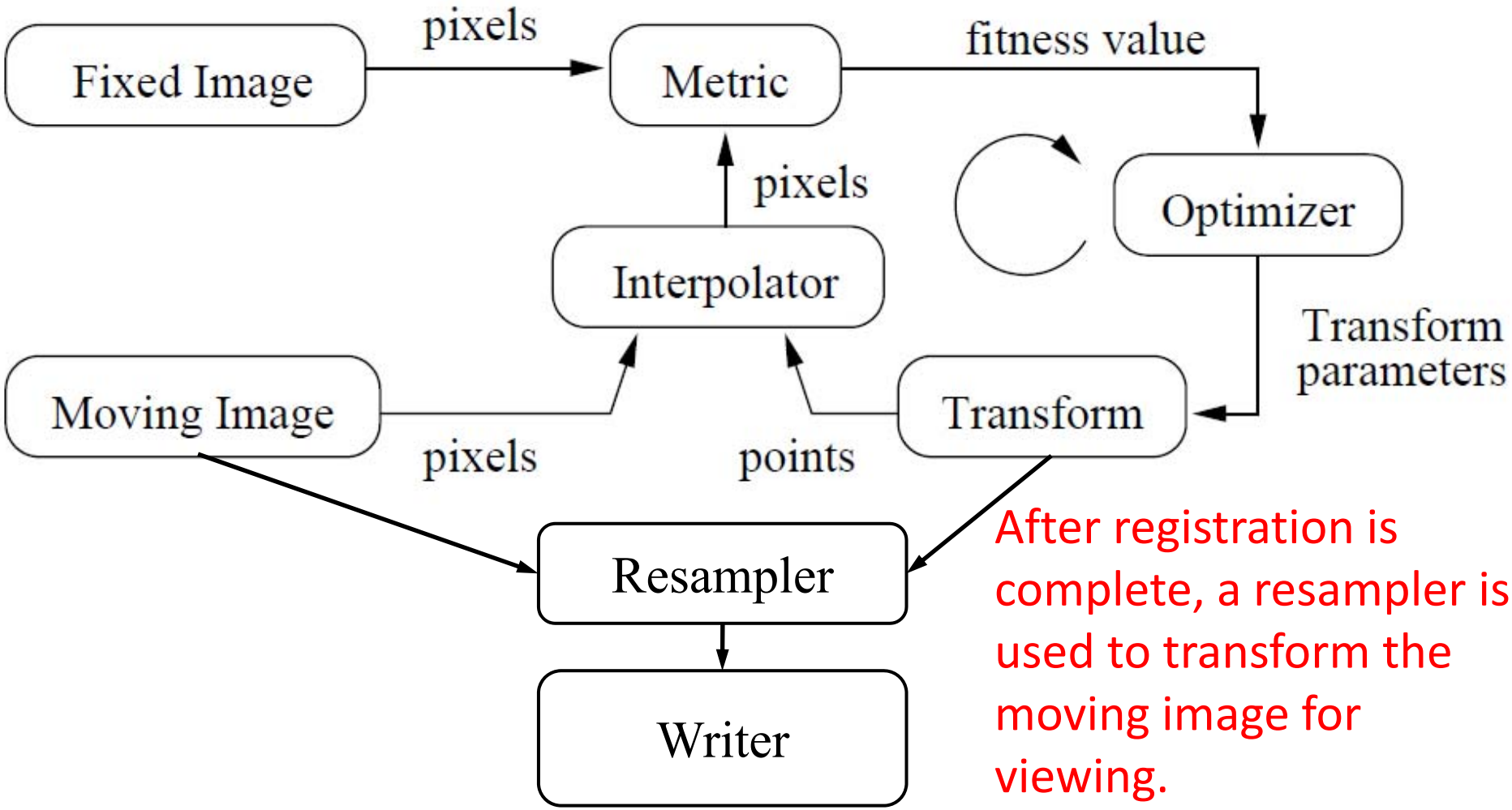
ITK registration framework

ImageRegistrationMethod



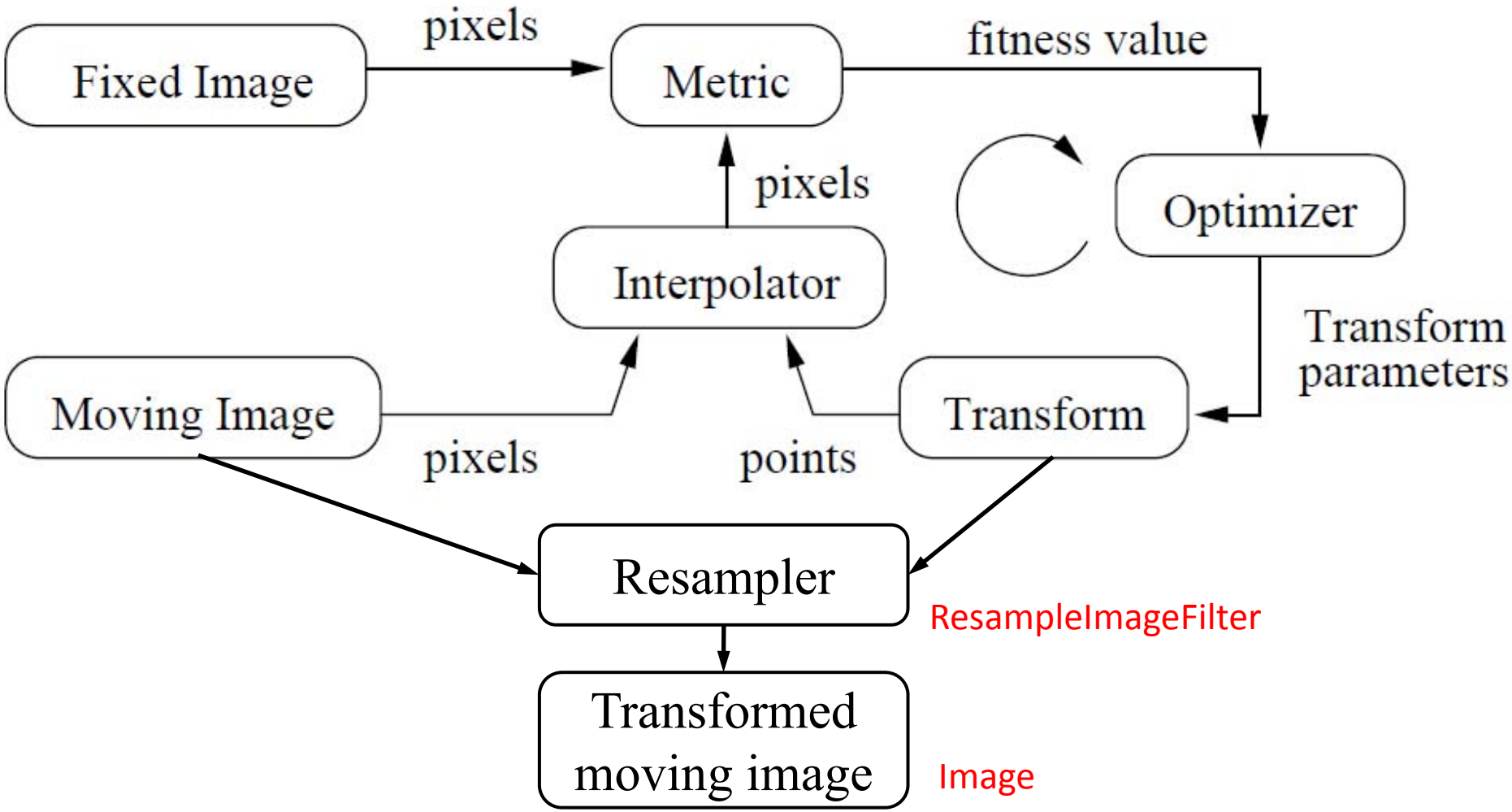
In ITK, all of these “ingredients” are plugged together on a “framework” class called itk::ImageRegistrationMethod.

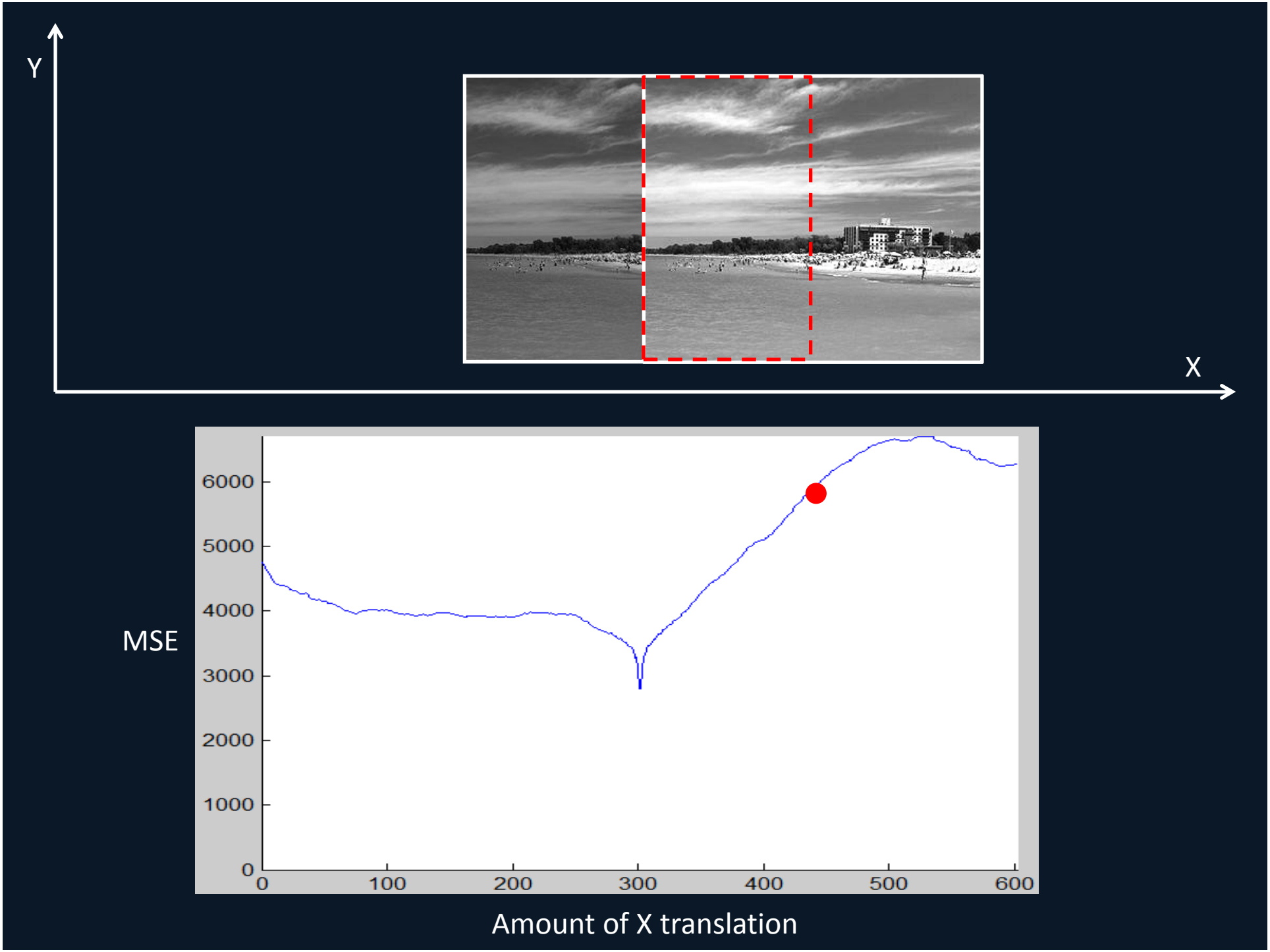
ITK registration framework



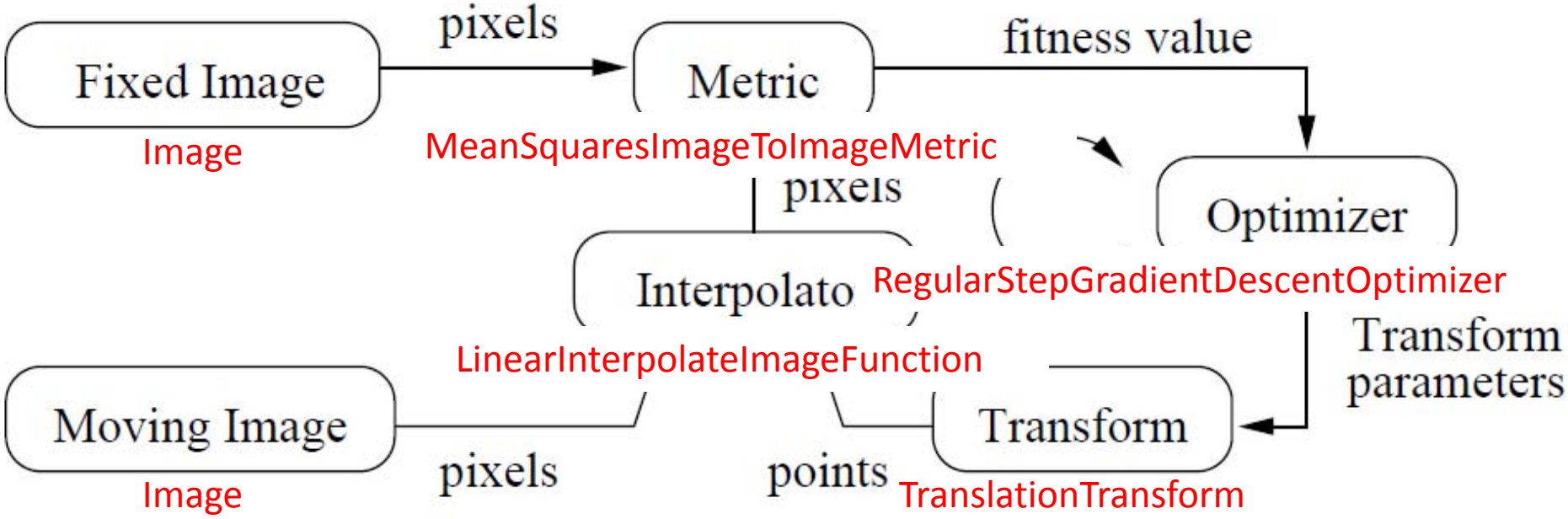
After registration is complete, a resampler is used to transform the moving image for viewing.

ITK registration framework



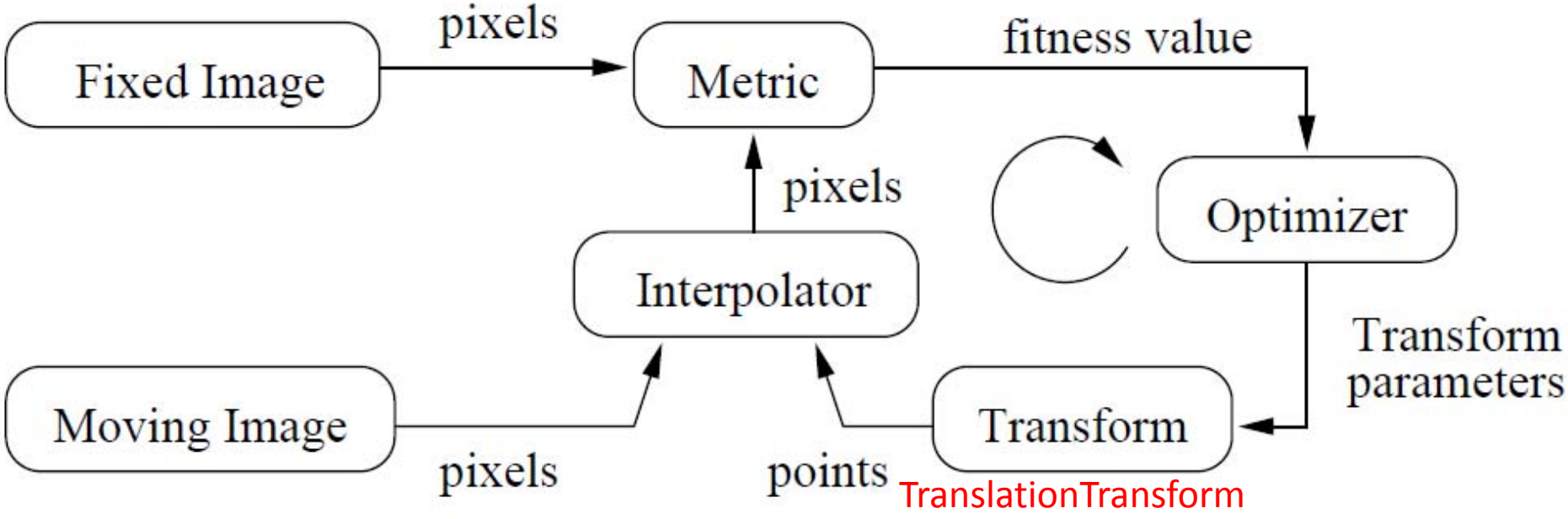

Y
X
MSE
6000
5000
4000
3000
2000
1000
0
0
100
200
300
400
500
600
Amount of X translation

ITK registration framework: 2D translation registration



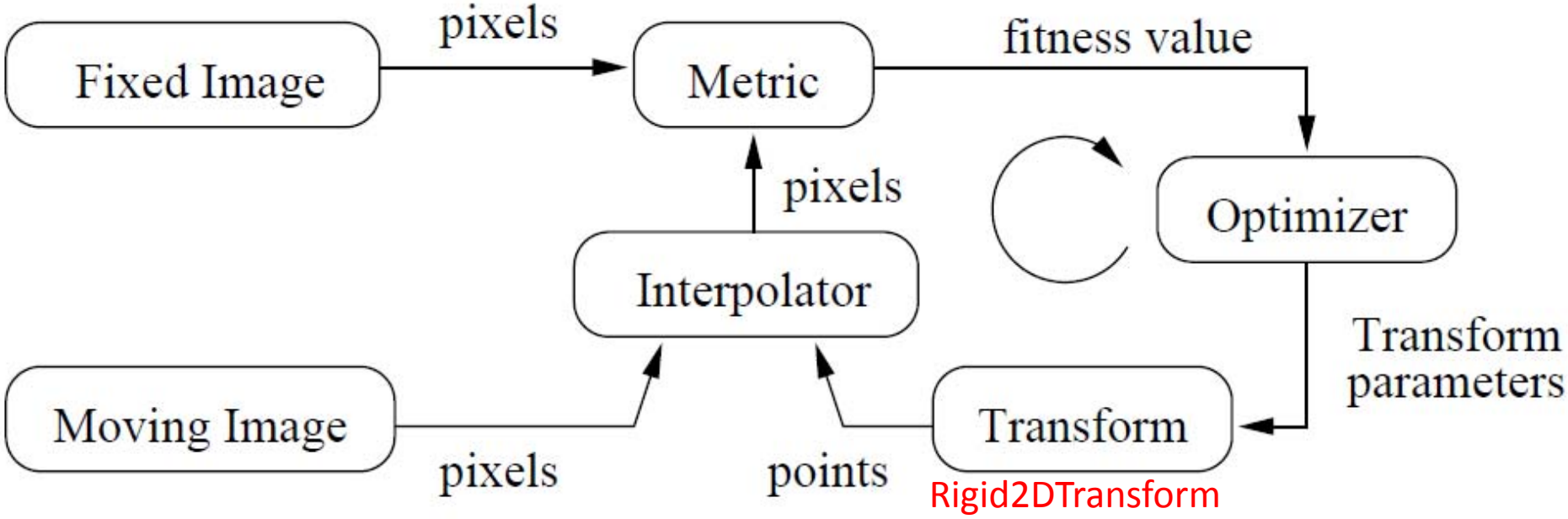
Which part to change to enable translation + rotation as well?

ITK registration framework: 2D translation registration



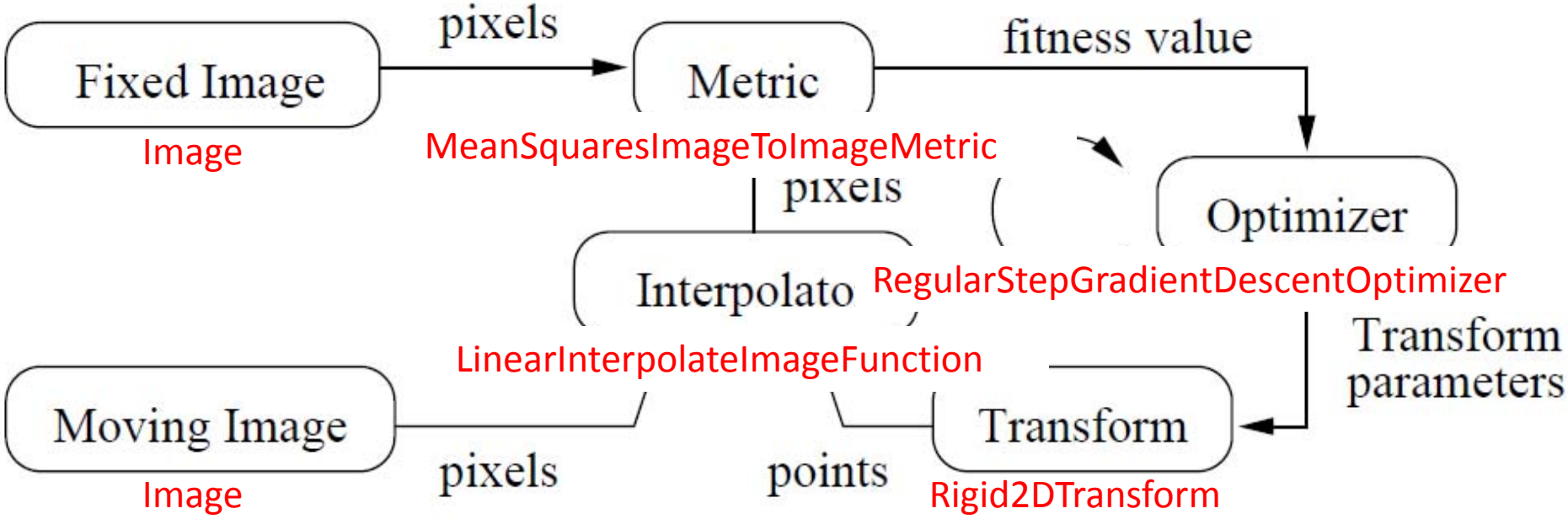
Which part to change to enable translation + rotation as well?

ITK registration framework: 2D rigid registration



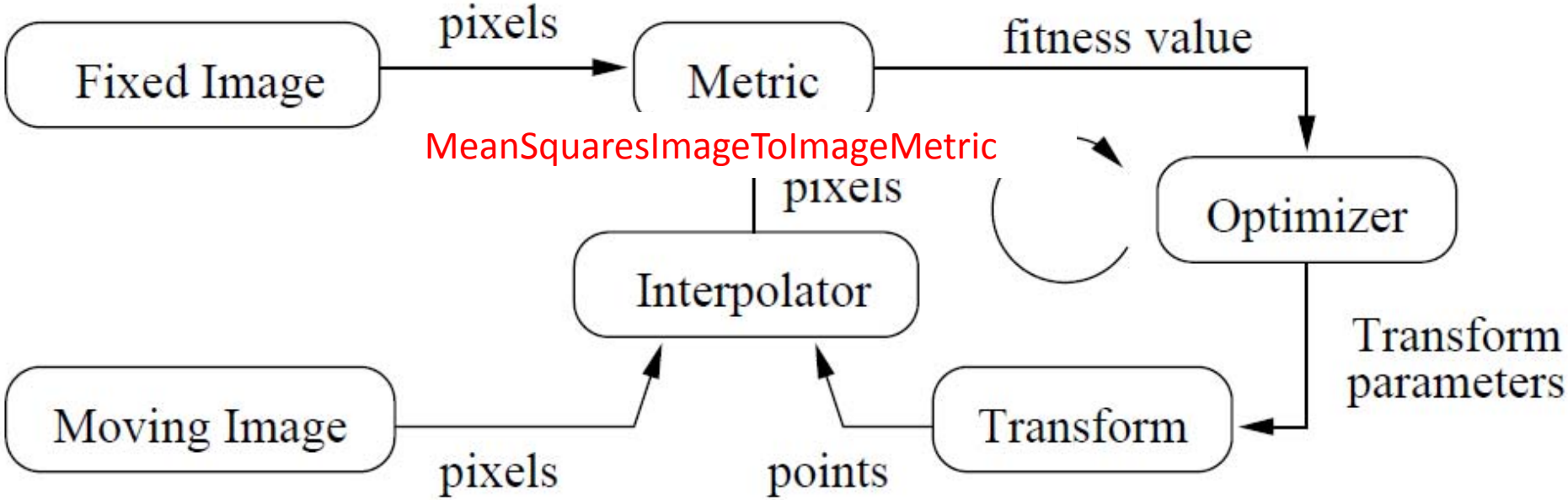
We can modify only the transform class, and leave everything else as-is.

ITK registration framework: 2D translation registration



Which part to change to enable multi-modal registration using mutual information?

ITK registration framework: 2D translation registration



Which part to change to enable multi-modal registration using mutual information?