Vernamonte Fabio (3374263), Vignolo Alessia (3365045)

COMPUTER VISION

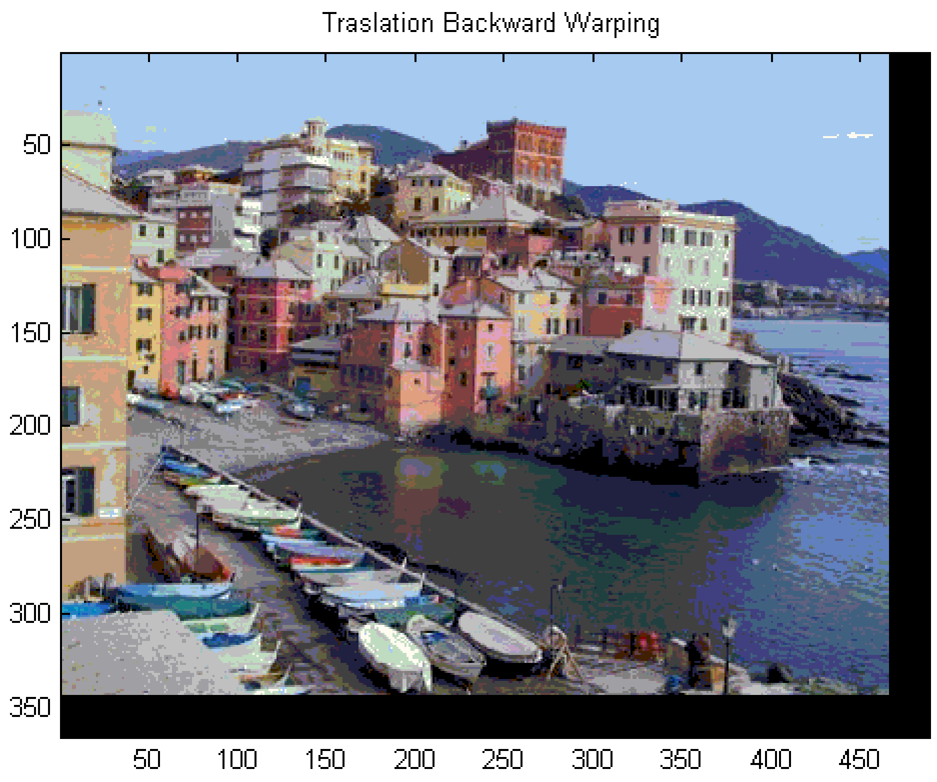
LAB 1 REPORT

Forward warping Vs Backward Warping

The first exercitation of Computer Vision concerns on transformations of images.
Every function we wrote is called from the main file, which loads the images and figure out the transformations.
All the images were inserted into the compressed folder as this report, where we show a few of them in order to make some examples of transformations.
In particular we implemented translation, rotation and swirl to some photos.
There are two ways of implementing these transformations: forward warping and backward warping.
The first one is the easiest; Given a photo, Matlab functions apply the direct transformation like the mathematical way. In fact translation is a shift of coordinates, rotation makes a multiplication between a rotation matrix and the position of a pixel, swirl compute a rotation based on the distance from the centre of the image.
The forward warping has an issue: if we perform a forward transformation with non-integer values, there could not be a correspondence between pixels of the transformed image and pixels of the original image. The not corresponding pixels are replaced with a “0” (which corresponds to black colour).
The backward warping reverses the typical method in order to obtain a best result, even if it uses the same mathematical function to make the transformation.
If we consider the original image as the transformed, we take a little advantage: we know the parameters of the near pixels. When there is not correspondence to the final position for a pixel, it will be replaced by a well-balanced average among the rounding pixels of the original image (which is given).
The main function of backward warping transformations is `backward_warping(A,Xt,Yt)`, which checks if the pixel is compatible with the grid of the final image (it have not to be exist if he will come out of the image during the transformation) and if he needs to be replaced by the bilinear interpolation among the four rounding pixels.
Then, after the algorithm, the backward warping has fixed the transformed image, which is better and without black points (they remain only with the forward warping).
Here is a short analysis of the three transformations and their performance with the backward warping and the forward warping.

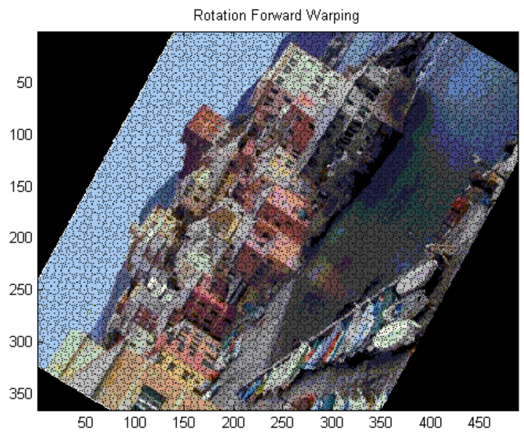
Translation

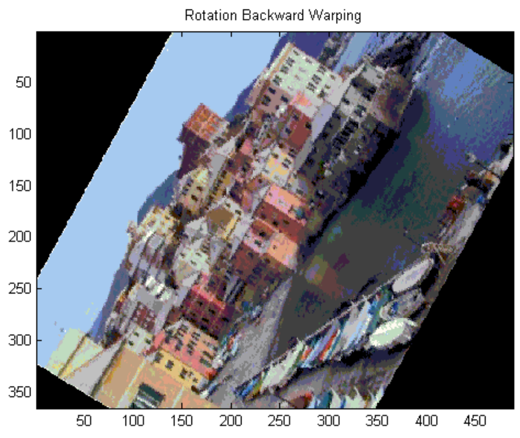
Is like a motion of the image of a given vector. There is no difference between forward and backward warping, as long as the vector of translation has integer components.



Rotation

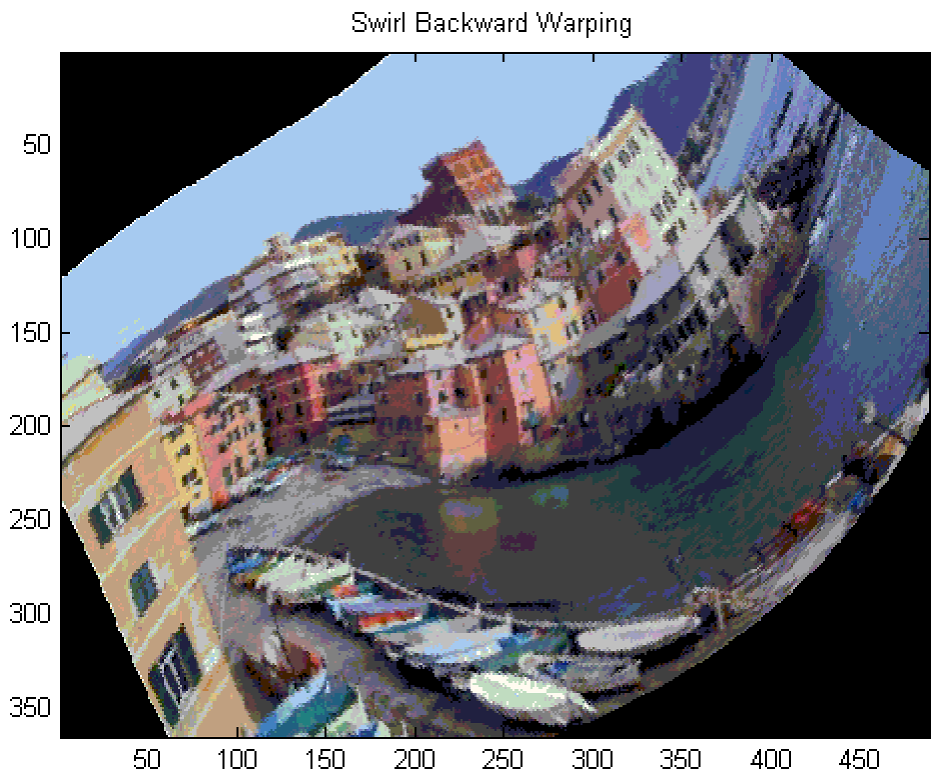
Rotation is the most interesting transformation because shows how the backward warping fix the final image. Given an angle theta, the rotation perform multiplications with trigonometric functions like sin(theta) and cos(theta), so there are often non-integer values for the position of the pixel. Backward warping deletes the black effect of the non-correspondence among pixels of images.





Swirl

Swirl is a rotation where the angle increases instead of being fixed. It is based on the distance from the centre of the image. Here is the Backward warping swirl of "Boccadasse".

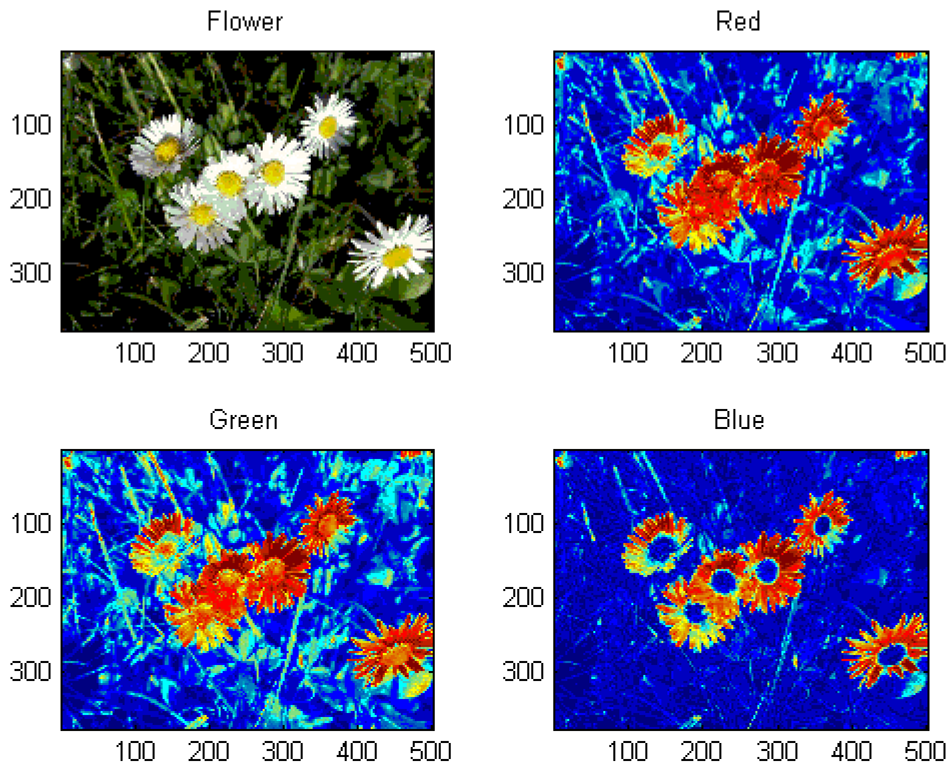


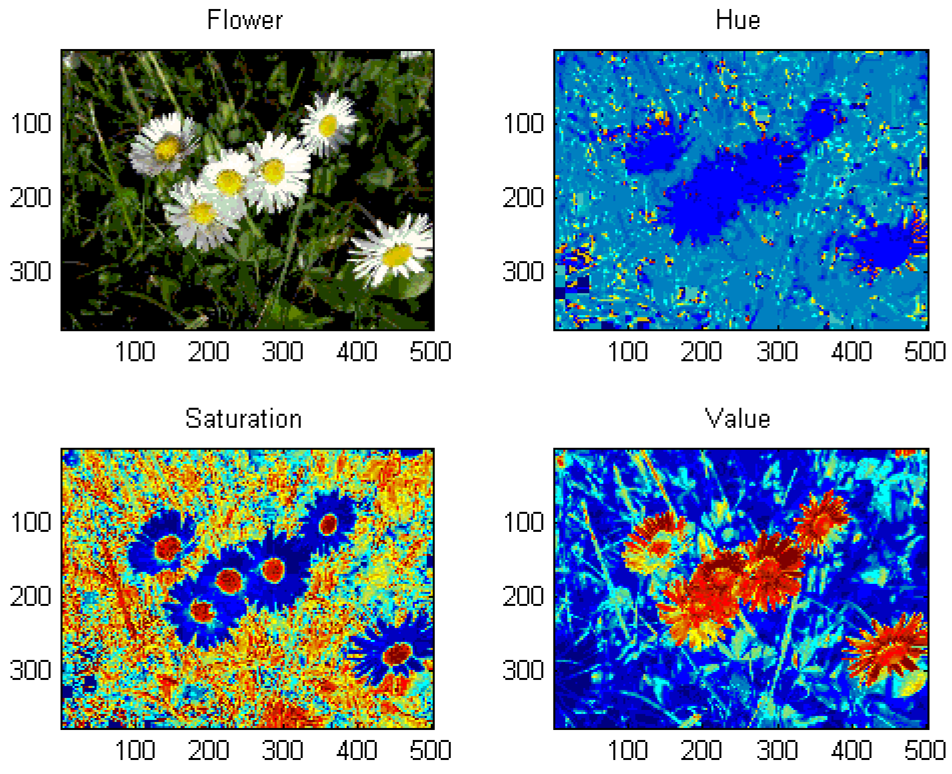
Colour Spaces

A computer representation of an image is given by a group of matrices. Every matrix has its meaning and they are referred to pixels of the image. Usually a pixel has five values: two of these indicate the position into the grid of the image, the other three values give the information about the colour's components.
During the first part of the lab1 transformations modify the position of the pixels, now we consider the colour spaces and how a machine represent them.
The most common system is the RGB= Red,Green and Blue components of the pixel. The sum of these three colours can reach all possible colours we can see on an image.
HSV interpretation looks like a "change of colour's coordinates". It uses three values :

- H=Hue, how the colour of the pixel looks like a primary colour. From the geometrical point of view, Hue is an angle graduation based on the colour of the pixel: it starts form blue to red.
- S=Saturation, it is a numerical value which means the intensity of the colour inside the image.
- V=Value, it can be compared to the brightness of the pixel, even if the value's formula is a little different from the brightness's formula.

With some Matlab functions we can convert a RGB image into a HSV image and vice-versa. We can also set the values as we desire. As the Lab1 requires, we set S value of "Color" to 0.5 (it was 1 before our setting) and the circle becomes more faded. When we set to 0 the S value, the whole image has no saturation, so only gray can be seen.
Here is an example of RGB and HSV vision of the same photo.


Flower
Red
Green
Blue
100
200
300
100 200 300 400 500


Flower
Hue
Saturation
Value
100
200
300
100 200 300 400 500