3.4 Robustness assessment

After the extraction of the crease, the next step is to iteratively transform one of the images until it becomes properly aligned with the other. The core of the algorithm is the 2–D version of that used for volumes in page 85.

The initial approach was to run the algorithm with the same values as obtained for the volume images. Results were good for all pairs of images for reasonable values of the parameters, but when we started working with *SLO* sequences we realized it was important to state the influence of each variable, regarding:

- **robustness** how many of the results were aligned enough to be good.
- **accuracy** how accurate was the registration for accepted results.
- **time** spend for each registration.

Of course we expected the usual tradeoff between the time and the two other variables, but due to the high volume of frames in the sequences it was important to determine exactly its relationship. We were interested in the following items (default values in brackets, steps as defined in figure 3.5):

- **Levels** ② Number of levels build for the hierarchical piramid. For Levels= 1, the base image is used.
- **Rotations**(6) ③ Rotations performed at the base level, where all translations are sampled using the Fourier transform. Each rotation has a step of 1.5 deg.
- **Ftol** $(10E-3)$ ④ controls the accuracy of the convergence. The convergence factor is defined as:
$$rtol = \frac{C_h - C_l}{(\bar{C_h} + \bar{C_l})/2} \tag{3.1}$$
where C_h and C_l are the highest and lowest values of the seeds stored in the Simplex algorithm, as implemented in [80]. The algorithm exits when:
$$rtol < ftol \tag{3.2}$$
- **Seeds** ④ are the number of seeds keept from one level the next. In theory, a higher number of seeds makes less likely to miss a candidate from the previous level which would have been underestimated. In practice, most seeds would converge to approximately the same transformation. At this point, we implemented an intermediate filter to discard too similar seeds, those which are likely to lead to the same solution.
- **Transf** (**SC**) ③, ④ We have tried three different transformations:
  - **SC** rigid+centered scaling.
  - **NSC** rigid+non centered scaling.
  - **Rigid** only rotation+traslation.

Method	# levels	# seeds (bottom to top)	Comment
1S	1	$\{1\}$	No hierarchical
2M	2	$\{1, 3\}$	2 Levels, multiple seeds
2S	2	$\{1, 1\}$	2 Levels, single seed
3M	3	$\{1, 3, 6\}$	3 Levels, multiple seeds
3S	3	$\{1, 1, 1\}$	3 Levels, single seed
4M	4	$\{1, 3, 6, 12\}$	4 Levels, multiple seeds
4S	4	$\{1, 1, 1, 1\}$	4 Levels, single seed
1S-H	1	1	1S + lower tolerance $ftol = 10E - 6$
1S-NROT	1	1	1S + no trial rotations
3M-NC	3	$\{1, 3, 6\}$	3M + scaling not centered (**NSC**)
3M-RI	3	$\{1, 3, 6\}$	3M + **rigid** transf.

Table 3.4: Benchmark of configurations for the registration algorithm. See text for the explanation of each term.

See appendix A for the formal mathematical description of this transformations.

We have not made all possible combinations of the previous items, with the aim of bounding the time to complete. Table 3.4 presents the complete list of combinations we have tested for three sequences of *SLO* images.

A further run we made was not included in the final figures, for its rate of success was too low to be of any significance other than that it fails. The run was made without the initial search in Fourier for translations, and without the hierarchical structure. The images were set to converge with the Simplex algorithm without initial transformations. Visual inspections showed that the search got trapped most of the times into a neighboor maximum.

We run the tests for three sequences of *SLO* images, all belonging to different patients. Table 3.5 presents the specifications of the images.

Dataset	A	B	C
Number of frames	3190	1510	1820
Frame size	720×400		
Frames per second	25		
Start	700	600	550
End	3190	1250	1820
Step	2	2	2
Empty	49	4	6
Total valid	1196	321	631
Reference frame	2998	1000	1430

Table 3.5: List of control sequences, each belonging to a different patient.

After running the tests for three sequences, we obtained a long set of transformations. Figure 3.10 depicts results for method *1S* . Figures would show similar shapes

for other methods. At the sight of the function profiles, a few remarks can be made.

It is interesting to the ability of the algorithm to recover even high value in the traslations parameters: sometimes the combination of both reduces the usable common region up to 1/4 of the total. The profile of the function is fairly constant through the acquisition, with only a couple of high rate changes caused by the blink of the eyes. Despite this, small oscillations occur between consecutives frames, indicating some sort of error in the registration process. Whether it is caused by our algorithm or intrinsic on the images will be investigated in the next sections.

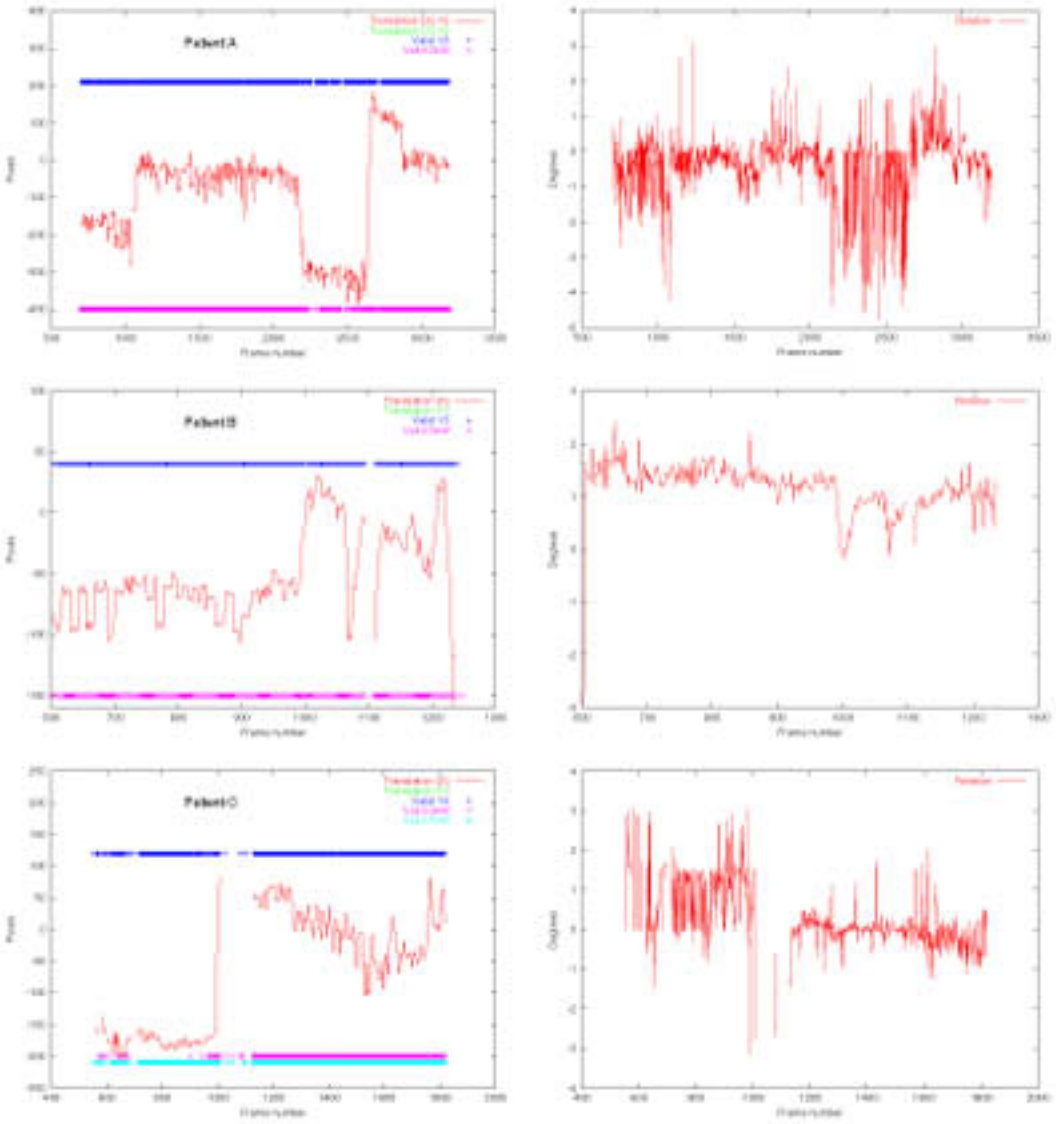

Figure 3.10: Each row corresponds to the results for one patient: traslation values on the left and rotation values on the right. We represent the success of each individual frame registration with straight lines on the left graphic. Lines are discontinuous where the method fails (validation described in next section). Transformation values given here are those of method 1S. We also give the success line for methods G4M and G3M, described later in page 107. The translation between consecutive frames is usually low, but sudden jumps can be very high, up to half the size of the image. The rotational value is low. In both cases, the values present some sort of fluctuation.

3.4.1 Methods to validate the testbench

The first step in the evaluation of results was to assess whether each registration was actually successful or not. That is, to visually compare the reference and the dynamic frames, once the later has been registered. However, the large number of frames made this process extrememly tedious. A more intelligent approach was necessary. First, we computed the histograms of some associated measures of alignement, to see if the two clusters – valid and non valid– could be easily distinguished.

Many measures of alignment exist. Simple linear correlation, the one used at the optimization process, was not suitable because it does not take into account variations on the contents of the image. This is to say, brighter images would have higher correlation, even if unmatched, compared to others with less visible structures.

A related suitable standard measure is the normalized correlation, defined as following. Given two images F and G with mean $\bar{F}, \bar{G}$ and standard deviation $\hat{F}, \hat{G}$,

$$NC(F,G) = \frac{\sum_{\vec{x}\in F}(F(\vec{x}) - \bar{F}) \cdot (G(\vec{x}) - \bar{G})}{\sqrt{\hat{F}\hat{G}}} \tag{3.3}$$

In addition to *NC*, we have defined a function to resemble the semantics of 'number of pixels that actually agree'. However, a measure counting overlapping pixels would not work properly for creases that, although very near, do not match for a few pixels. This case is common, and caused by the different shape they have due to optical distortions. To compensate this effect, we computed a dilation of the creaseness image. Given two registered creasness images F and G, and the threshold value $thrk$ from table 3.3, we define the *Normalized matching* measure as:

After defining the standard threshold operator

$$\mathrm{Thr}(Im, v) = \{Im_{i,j} > v ? 1 : 0\}$$

We threshold the creaseness images

$$F_b = \mathrm{Thr}(F, thrk)$$
$$G_b = \mathrm{Thr}(G, thrk)$$

And dilate the dynamic image

$$G_d = \mathrm{Dilate}(G_n, 3)$$

Now, count the number of non-void pixels

$$V_F = \mathrm{Volume}(F_b)$$
$$V_{FG} = \mathrm{Volume}(F_b \cdot G_d)$$

And finally its quotient

$$\mathrm{NM}(F,G) = \frac{V_{FG}}{V_F}$$

The response of both classification methods is similar, as figure 3.11 shows. Only for this figure, we manually validated each single registration in the sequence, and then computed the histogram of the NC and NM functions for the valid and non-valid frames. For both measures, the two clusters (valid and non-valid) can be separated by means of two threshold values, one for the top in the rejected cluster, and the other for the bottom in the accepted cluster. But since this separation is clearer for NC, we decided to employ solely the NC function at the automatic validation step. Another favorable factor was that NC is much faster to compute than NM.

We checked that this classification method worked for all the sequences, and results were positive and consistent. Taking them into account, we devised a simple semi-automatic classification scheme to validate the correctness of each registration. The algorithm we used was:

Given the normalized correlation NC and two given thresholds NC_B and NC_A (for rejected below and accept above),

$$\begin{array}{rl} \text{if } NC < NC_R & \text{Reject} \\ \text{if } NC > NC_A & \text{Accept} \\ \text{otherwise} & \text{validate manually} \end{array}$$

Once we have validated a sequences for a given method, the existing classification can be for other methods applied to the same sequence. For a given registration with NC values between NC_R and NC_A, which had to be validated manually, it is reasonable to use for comparison the NC_G value of the already classified corresponding pair. Then, for $NC > NC_G$ and NC_G being accepted, the registration in process may be also accepted.

Therefore, in additon to those previous rules, if $NC_R < NC < NC_A$, we have applied:

$$\begin{array}{rl} \text{if } NC > NC_G \text{ And Model Accepted} & \text{Accept} \\ \text{if } NC < NC_G \text{ And Model Rejected} & \text{Reject} \\ \text{if Transformations are similar} & \text{Copy results} \end{array}$$

We have taken, for each patient, the method *1S* as the reference to validate the other methods. This choice makes sense since its search is the most succesful due to the fact it does not employ the sampled levels in the piramid: it uses all the information at the base level to locate the best seeds for translations, while others rely on the hierarchical sample, more prone to be missguided.

Following we present the table 3.6 with the statistical results for the three patients and, in the next section, we make an analysis of the results with the aim of stating which method is considered to be the best.

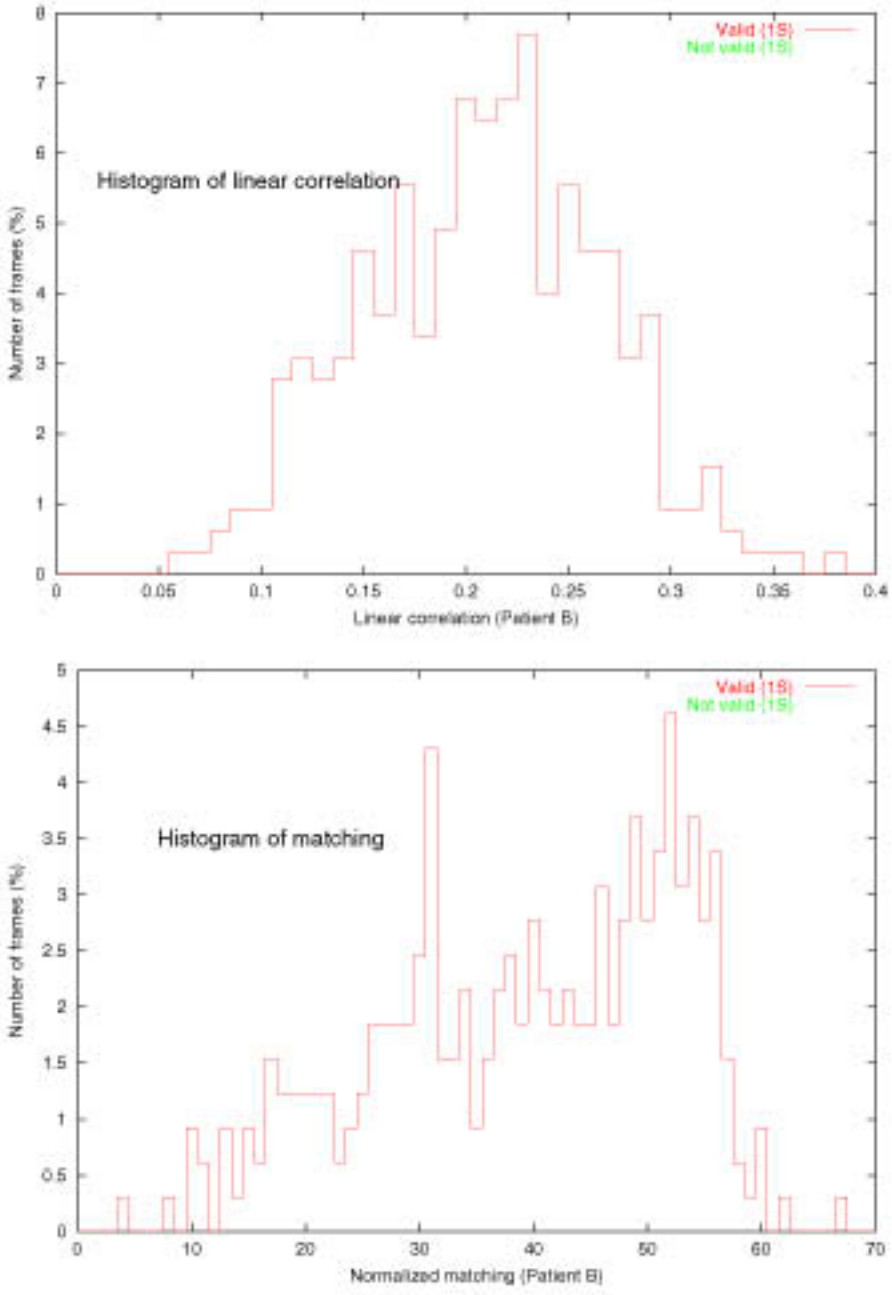

Figure 3.11: To validate automatically the registration of a frame, we have computed two measures: normalized correlation (NC) and number of matchings (NM). The two clusters of valid and not-valid segmentation are easily distinguible.

	Method	Succ. (%)	Fail. (%)	Corr.	NCor.	Time (sec)	Method	Succ. (%)	Fail. (%)	Corr.	NCor.	Time (sec)
Patient A	1S	94.1	1.9	1207	0.21	94.1	1S-H	95.7	0.3	1248	0.22	101.3
							1S-NROT	93.7	2.4	1201	0.20	28.6
	2M	94.5	1.6	1225	0.21	28.7						
	2S	93.9	2.2	1224	0.21	25.9						
	3M	92.5	3.5	1242	0.21	9.2	3M-NC	92.1	3.9	1136	0.19	9.2
	3S	90.9	5.2	1257	0.21	7.7	3M-RI	91.7	4.3	1049	0.18	7.8
	4M	86.8	9.3	1288	0.22	4.9						
	4S	86.1	10.0	1290	0.22	3.3						
Patient B	1S	95.1	3.7	2530	0.21	109.5	1S-H	95.1	3.7	2621	0.22	114.3
							1S-NROT	93.9	4.9	2424	0.20	28.8
	2M	95.1	3.7	2511	0.21	28.1						
	2S	95.1	3.7	2508	0.21	26.2						
	3M	93.3	5.5	2552	0.21	10.3	3M-NC	94.5	4.3	2127	0.17	9.8
	3S	93.9	4.9	2550	0.21	8.2	3M-RI	93.6	5.2	2122	0.17	8.3
	4M	94.5	4.3	2532	0.21	6.1						
	4S	90.8	8.0	2578	0.21	3.8						
Patient C	1S	85.1	14.0	1279	0.17	96.6	1S-H	87.4	11.6	1289	0.17	107.2
							1S-NROT	82.9	16.2	1291	0.17	24.6
	2M	82.7	16.4	1300	0.17	24.1						
	2S	81.8	17.3	1307	0.17	23.1						
	3M	76.1	23.0	1359	0.18	8.1	3M-NC	75.0	24.1	1309	0.17	8.2
	3S	73.4	25.6	1384	0.18	6.6	3M-RI	74.7	24.4	1296	0.17	7.0
	4M	59.3	39.8	1539	0.20	5.0						
	4S	57.9	41.2	1563	0.20	3.2						

Table 3.6: Results of the test bench for the three patients. For a given patient, compare robustness, mean correlation and time in between consecutive levels and in between adjacent levels. A base time of 1.5 sec. to extract the creases must be add to all figures. The time is per registered frame.

3.4.2 Conclusion from the testbench

At a first sight of table 3.6 one can see that robustness depends on the quality of the sequences. For patient A and B, it is above 90%, dropping to $60-90\%$ for patient C. However, we feel that it would not be fair for the method to expect independency on the source data. Indeed, visual examination of the sequence C showed the reference frame not to be contained in the frames where registration failed. This is particularly true for method *1S* , which, as explained in a previous section, would give the top expected rate for each patient. Therefore, the comparison will always be relative to these upper boundary.

It is remarkable that numbers (robustness and time, mainly) follow the same pattern along the methods for all the patients, thus indicating their consistency. The following conclusions have been deduced:

- Because of manual validation for each frame is a user-dependent task, small variations are not statistically significant. For instance, *1S* and *1S-H* should equal in robustness, since the last is simply a version of the first with higher requirements in accuracy. Expected fluctuation is about 2%.

- The correlation mean for some methods is higher than that of the reference method, *1S* . This is because the mean is computed only for the frames accepted as valid. The method *1S* may register frames with less information, i.e. low correlation values, which the other methods would fail to register. Therefore, low correlation frames would lower the mean value for the reference method, while being unused for the others.

- Results do not differ between one and multiple seeds for a given level. This implies that the maximum found at the exhaustive search leads to the final solution, with no false responses, and that the creaseness segmentation together with the correlation is a sound basis for registration for ophthalmologic images.

- If no rotation is tried at the Fourier level, results are equivalent. This is due to the low rotation found for all the sequences. Of course, in images with stronger rotations, results would be worse, but this has not been observed in the sequences analysed until now.

- A more demanding tolerance threshold $ftol$ achieves equivalent robustness, higher correlation means and higher computation times. Whether better correlation is relevant to the final accuracy is not clear, for the characteristics of the images (non-rigid distortions) probably make it useless.

- The transformation model is relevant: if it does not include scaling, correlation mean drops. Robustness, on the contrary, does not, because scaling factors are actually computed as an extra run of the algorithm, which should already have converged to a valid result.

- The sequence to compose the transformation matrix is also relevant. It is better to compute scaling as centered rather than not centered (recall definitions at page 201). Although algebraically any transformation expressed with one

method can be expressed with the other, the structure of the algorithm favours centered scaling. The cause is the same as the previous paragraph: since a rigid transformation is computed first, the optimization would converge more easily with the transformation more similar to the rigid one, which is the centered scaling. Note that at this second optimization step, not only the scaling but all the parameters are computed, and often they change significantly.

- The hierarchic strategy is efficient at reducing the computation time, while keeping results acceptable. For large images (more than 512 rows or columns) the proposed method has a bottleneck at the initial step: each sampled rotation demands the computation of three costly Fourier transforms. As the number of levels increases the size of the images halves and, since the complexity of the transformation to the Fourier domain is $O(n^2 \log n)$, the time required drops. However, the method is less robust because a) false maxima appear, b) true maxima are hidden.

The comparison demonstrates the method *4S* to be the best choice to achieve acceptable success in the shortest time. For sequences with highest rotations or worse contrast, it may be preferable to shorten the number of levels (*3S*), and increase the number of seeds at each level (*3M*).