Superresolution software manual

S. Villena, M. Vega, D. Babacan, J. Mateos
R. Molina and A. K. Katsaggelos

Version 1.0

Contact information:
Rafael Molina
Departamento de Ciencias de la Computación e I. A.
E.T.S. de Ingeniería Informática y de Telecomunicación
Universidad de Granada
18071 Granada (Spain)
e-mail: rms@decsai.ugr.es

This manual describes the use of the software application implementing the SR algorithms developed in [1, 2, 3, 4] as well as other superresolution methods developed by the authors [5]. The application has been developed in MATLAB® (ver. 7.9/R2009b) including the graphic user interface (see figure 1).

The developed software:

- Allows the execution of SR methods in real and simulate modes.

- Evaluates the following SR methods:

    - Bicubic interpolation.
    - The TV prior based method proposed in [5].
    - The method described in the algorithm 3.1 of [1] which utilizes a SAR prior model.
    - The algorithm proposed in [2], with the $\ell1$-norm of the horizontal and vertical gradients as the image prior model.
    - The method proposed in [3] based on a SAR and $\ell1$-norm model combination with registration parameter estimation.
    - The method described in the work [4] which combines SAR and TV prior models with the estimation of the registration parameters.

If you use this software to evaluate any of the described methods, please cite the corresponding papers. The papers can be downloaded for research purposed from `http://decsai.ugr.es/vip`.

- Saves the results (HR images and registration parameters).

- Allows the user to stop the execution at any time, with the possibility of saving all the estimated parameters as well as the estimated HR image.

We assume here that a digital camera captures a sequence of L low resolution images of size N pixels. In the spatial domain, the process to capture the k-th image, $\mathbf{y}_k$, can be expressed using matrix-vector notation, as

$$\mathbf{y}_k = \mathbf{A}\mathbf{H}_k\mathbf{C}(\mathbf{s}_k)\mathbf{x} + \mathbf{n}_k, \qquad (1)$$

where $\mathbf{x}$ represents a digitalization of the real scene with a resolution of PN pixels, with $P > 1$ representing the magnification factor, $\mathbf{C}(\mathbf{s}_k)$ is a $PN \times PN$ matrix representing the motion of the k-th image in the sequence with respect to the reference image, with $\mathbf{s}_k = (\theta_k, c_k, d_k)^T$ the motion parameters (rotation angle and horizontal and vertical displacement), $\mathbf{H}_k$ is a $PN \times PN$ matrix representing the blur, $\mathbf{A}$ is the $N \times PN$ integration and subsampling matrix and $\mathbf{n}_k$ represents the noise. This model is used in the software to simulate the low resolution observed images.

The simulated mode allows to evaluate the quality of the resulting HR images and the error in the estimated parameters for each method (see the experiment sections in chapters 4–6 of [1]). In contrast the real mode execution allows to run the methods with real datasets. This mode has been utilized in the real experiments in chapter 6 of [1].

Depending on the selected mode, and before running the selected method, the user has to provide the information required by the method. Additionally, if the simulated mode is selected, the user has to introduce the necessary information to generate the observed low resolution image set according to Eq. (1), that is, the original HR image, the number of observation to generate, the blur, the additive Gaussian noise variance and the subsampling factor. In real mode, the user has to provide the LR image set, the assumed, or previously estimated, blur and the desired upsampling factor.

Once the initial information has been filled in, the user has to choose the SR method and run it. Note that, when selecting any of the proposed methods that use model combination, it is also necessary to introduce the combination parameter (λ) in the range $[0, 1]$.

The user can intuitively navigate through the GUI and complete the initial information needed by the SR method. Also, the user can access the reconstructed HR image and registration parameters on each iteration, in case they were estimated.

The access and workflow through the developed GUI is detailed next.

1   Installation

The application does not need to be installed. To access the application just uncompress the provided package. The application was developed in MATLAB® version 7.9.0.529 (R2009b) and it has been tested to work only in this version.

2   User interaction with the application

The main steps to run the application are:

1. Start the application.

2. Select the (**Real** or **Simulated**).

3. Fill in the initial information.

4. Select the SR method to execute.

5. Execution monitoring and saving options.

Let us describe each step in details.

2.1   Running the application

To run the MATLAB® application, change your current directory to the application folder and run the program ***SuperResolutionv1.m***. In a few seconds the GUI depicted in figure 1 will appear.

The GUI is divided in the following interactive areas:

1. **Mode/Reset**: Allows the user to select between *simulated* or *real* execution mode and to reset the program.

2. **Low Resolution Image**: this area shows the utilized simulated or real observations depending on the current execution mode.

3. **High Resolution Image (Simulated Mode)**: Used only in simulated mode. It is utilized to load and previsualize the HR image from which the LR observations will be generated.

4. **LR Generator**: (Only active in simulated mode). This panel allows to introduce the required information to generate LR observations from the image loaded in the previous area.

5. **Super Resolution**: Allows to select the SR method to be applied to the LR observations.

6. **High Resolution Image**: This area displays the resulting HR image on each iteration. The reconstruction process can so be monitored.
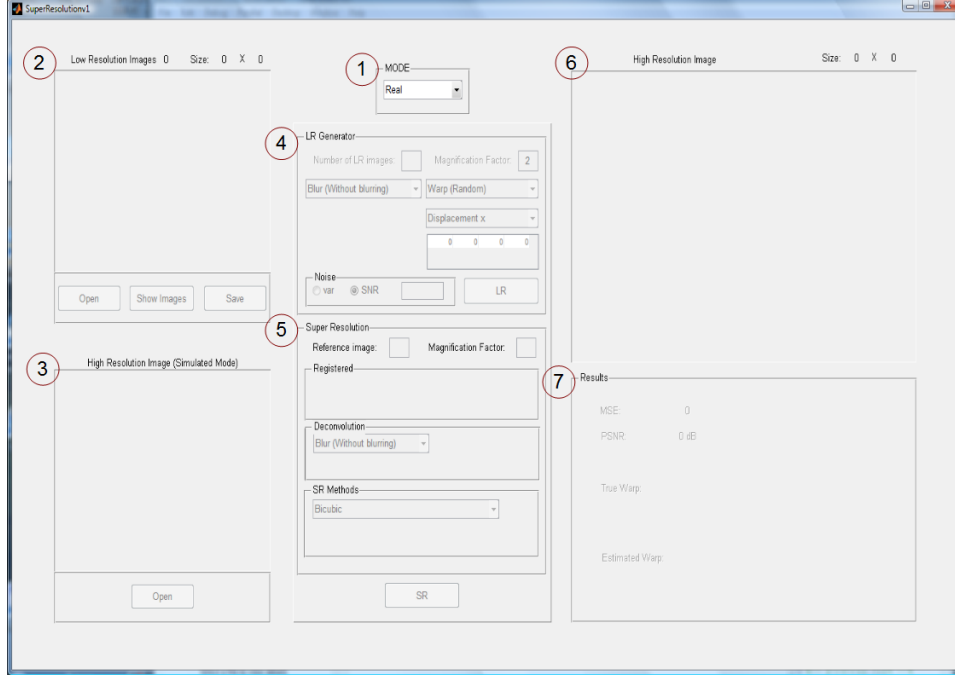
Figure 1: GUI appearance. Numbers 1 to 7 inside the red circles indicate the interface interaction areas.

7. **Results**: In simulated mode, this area shows error measures of the HR reconstruction and the value of the real and estimated parameters (if applicable). In real mode, it shows the estimated registration parameters.

The only GUI element active at the beginning is the area **Mode**. The user must then select one of the two available execution modes: real or simulated.

2.2 Execution Mode

The execution mode can be selected by clicking on the **MODE/RESET** pull-down menu (see figure 2). Note that the **Real** mode is chosen by default.

Depending on the selected mode, the GUI activates the fields where to provide the information needed to run the selected SR method.

Each execution mode is described in a separate section since the selected mode determines the application workflow.

Selecting the execution mode at any time involves resetting the application to the selected mode. All the loaded images and obtained results will be lost and all the parameters will be reset to their default values.
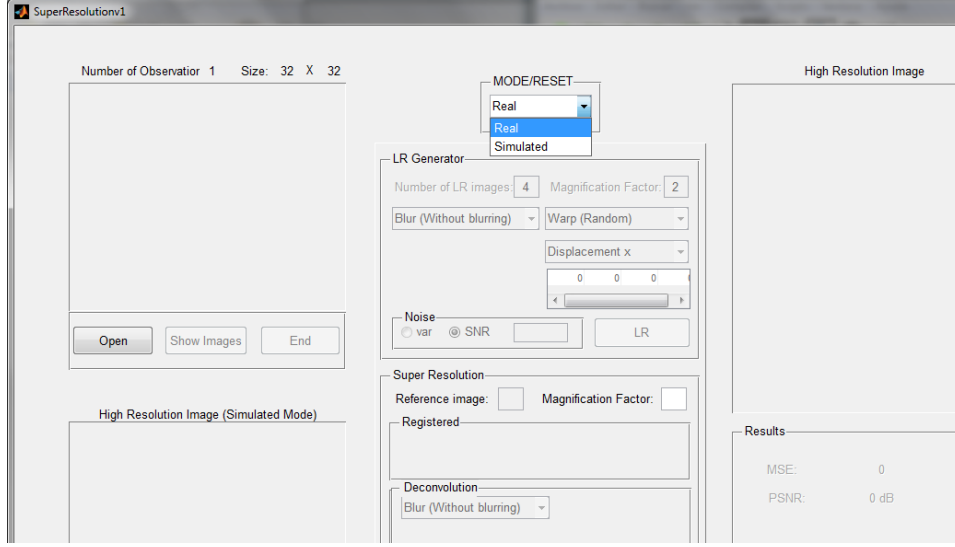
Figure 2: GUI appearance when the area **MODE/RESET** is active.

2.2.1 Simulated Mode

When the simulated mode is selected, the **Open** button, in the area **High Resolution Image (Simulated Mode)**, is activated (see figure 3).

When the **Open** button is pressed, the file manager is launched, this allows to select the HR image to be used in the experiments, see figure 4. For instance, if we select the image *lenag80.png*, the image is displayed and the **LR Generator** panel becomes available (see figure 5). This panel shows all the required information needed to generate the LR images using Eq. (1).

All fields have been initialized to default values that can be modified at any time. The blurring function is defined by the selected kernel, as shown in figure 6. Note that, for each type of blur, the values of the blur matrix coefficients are requested by the application. They are initialized with a default value in order to facilitate the user input.

When the **Customized** option for the blur is selected, a new window is displayed in order to introduce the blur coefficients, see figure 7. Additionally one can select either the **Save** option to save the current kernel or **Load** to load a new one.

The registration parameters can be defined in three forms: **Warp (Random)**, **Customized** and **Example**, depending on whether you want to generate them randomly, introduce the parameter manually (rotation angle and vertical and horizontal displacements) for each observation or using the parameters in the experiments described in chapters 4–6 of [1], respectively (see figure 8).

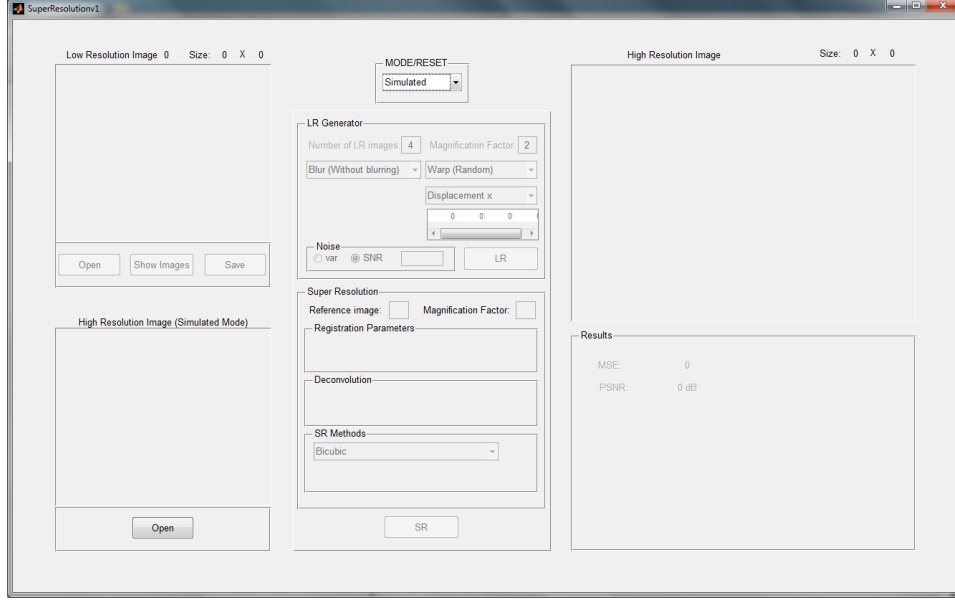Finally, the user must set the zero-mean Gaussian noise level, which

Figure 3: GUI appearance when the area **Mode** is activated to **Simulated**. Note that the button **Open** is now active.

can be defined in two forms: introducing the noise variance or by setting the signal-to-noise ratio (SNR), both in exclusive buttons. Depending on the selected option, the numeric field allows to introduce either the noise variance or the SNR value. The SNR mode is active by default with a value of **30** dB.

The **LR** button is available once all the required information to generate the observations has been provided (see figure 9). Notice how the areas **Low Resolution Image** and **Super Resolution** have been activated. Now the **Low Resolution Image** area allows to save the generated observations. The observations are saved by typing a generic name in the file manager window. For instance, if the user generated 5 observations and set the generic name *lenag.png*, then the application creates the image files *lenag_1.png, lenag_2.png, lenag_3.png, lenag_4.png* y *lenag_5.png*.

Now we can apply the SR methods implemented in the software to the generated observations. Notice that **Reference image**, **Registration parameters** and **SR Methods** are the only areas enabled in simulated mode. The subarea **Registration parameters** allows the user to choose between using the true registration parameters or estimate them. True registration parameters are used by default.

The next step consists of selecting the desired SR method to execute as shown in figure 10.

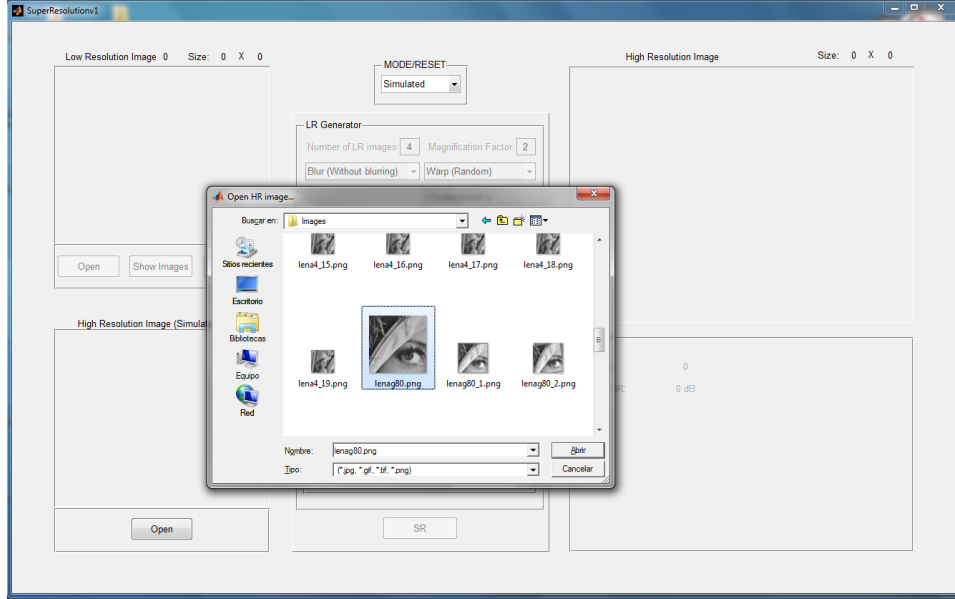At this point, there are 5 active areas: **Low Resolution Image**, **Mode**,

Figure 4: GUI appearance after clicking the **Open** button.

High Resolution Image (Simulated Mode), **LR Generator** and **Super Resolution**. That means that you can modify the fields of any of these areas, and generate new observations, with the possibility of saving the previous ones.

The selected SR method is executed by pressing the **SR** button. The areas **High Resolution Image** and **Results** are now enabled, as displayed in figure 11. On each iteration the HR reconstruction and its quality metrics are displayed. The execution can be stopped at any time by pressing the **Cancel** button. When the method stops, due to convergence or cancellation, the application shows the true registration parameters and, if applicable, the estimated ones (see figure 12).

Additionally the application allows the user to display the best reconstruction in the PSNR sense, by pressing the **Best Results** button, that could not coincide with the solution obtained in the last iteration, as seen in figure 13. If the button is pressed again, the HR reconstruction in the last iteration is shown. Both reconstructions can be saved pressing the **Save** button.

In order to request the file name to save the HR reconstruction, the file manager is launched. Let us point out that the application saves the HR reconstruction in the last iteration with the given name, the best reconstruction in the PSNR sense with the same name but ending in *Best* and it also saves a file with the same name but ending in -Dat.mat containing the estimated parameters in a data structure named *outopt* (see figure 14).
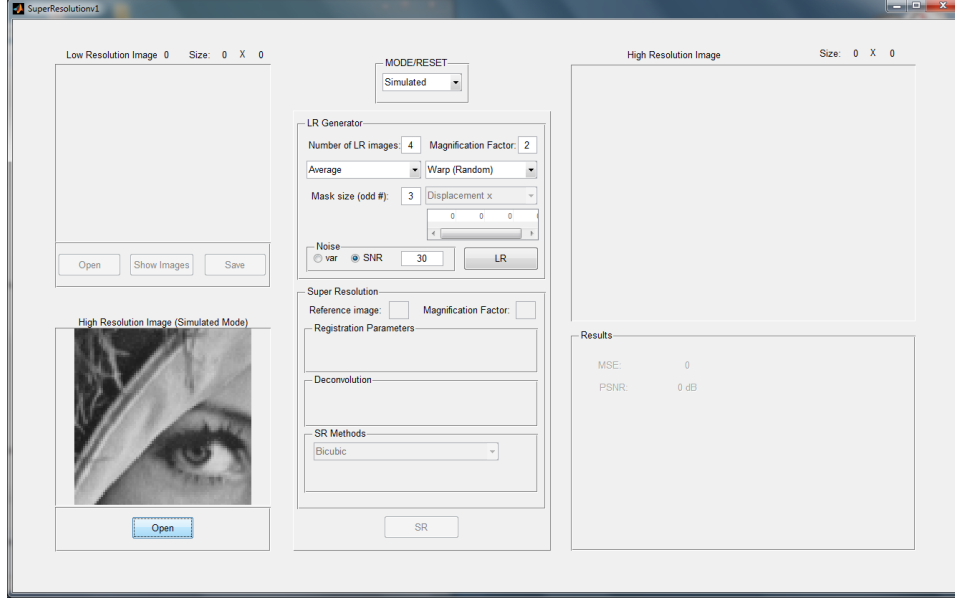
7

Figure 5: GUI appearance after the initial HR image is loaded.

When any SR method using prior model combination is selected, the combination coefficient, λ, is shown as depicted in figure 15. The combination coefficient, in the range 0 to 1, is displayed on a new field. The default value 0, means that the algorithm is exclusively applying the SAR prior model.

2.2.2 Real Mode

The **Number of Observations** panel and **Open** button are activated on Real mode (see figure 16).

When pressing the **Open** button, the file manager is launched requesting for the first observed image file name. The rest of options (**Show Images** and **End**) are available once the first observation is introduced. Adding new observations is as simple as repeating the procedure until the last observation is opened and the **End** button is pressed. The user can also load all the observed images at once by selecting a .mat file containing a single variable of size (rows × columns × n. observations) that stores all the images.

The **show images** button displays the set of images currently loaded.

When the **End** button is pressed, the **Super Resolution** panel is activated, allowing the user to set the magnification factor (set to 2 in each direction by default), and the blurring kernel (without blurring, by default) in the **Deconvolution** subpanel.

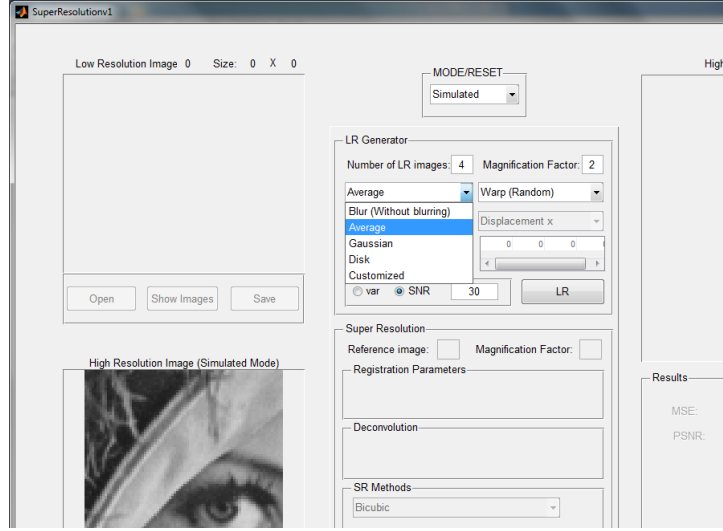As for the simulated mode, the user selects the SR method and presses

Figure 6: Blur kernel selection.

the **SR** button to run it. The appearance of the panels **High Resolution Image** and **Results** are shown in figure 17. Once the method has finished, the registration parameters are shown and the results can be saved by pressing the **Save** button, see figure 18.

3 Files output during the execution

The application displays the execution status in the GUI but it also saves *log* files in the directory **tempSR** which contains the information of all the parameters estimated in each iteration. An example of the content of this file, for the real mode execution depicted in figure 18, is shown in figure 19.

Acknowledgments

Note

Please report bugs to Rafael Molina (rms@decsai.ugr.es).
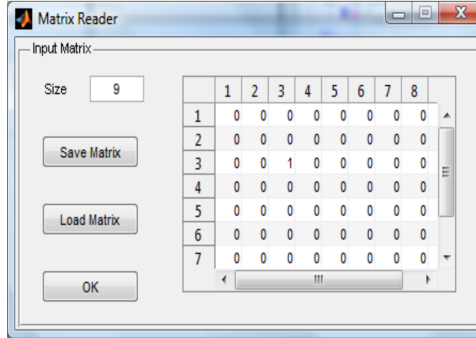
Figure 7: This matrix allows the user to define the blur kernel in the **Customized** mode.

License

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

References

[1] S. Villena. *Superresolución y Reconstrucción Bayesiana de Imágenes a Partir de Imágenes de Baja Resolución Rotadas y Desplazadas. Combinación de Modelos.* PhD thesis, Universidad de Granada, December 2011.

[2] S. Villena, M. Vega, R. Molina, and A. K. Katsaggelos. Bayesian super-resolution image reconstruction using an l1 prior. In *6th International Symposium on Image and Signal Processing and Analysis (ISPA 2009) Best paper award, Image Processing and Analysis Track*, pages 152–157, 2009.

[3] S. Villena, M. Vega, D. Babacan, R. Molina, and A. Katsaggelos. Bayesian combination of sparse and non sparse priors in image superresolution. *Submitted to Digital Signal Processing*, 2012.
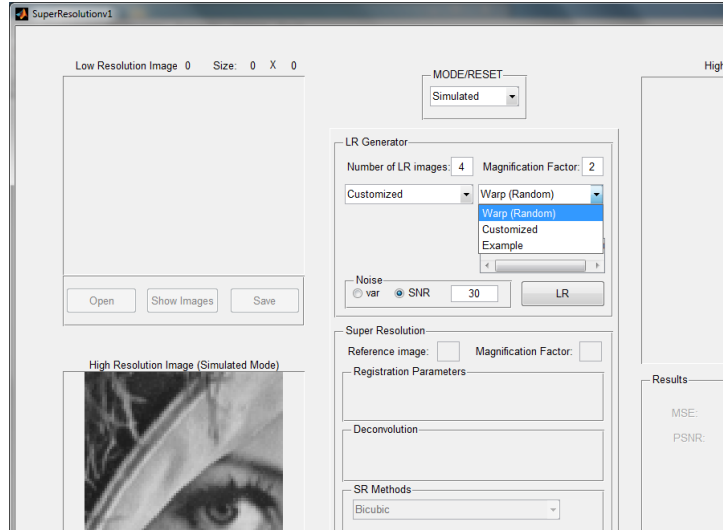
Figure 8: Setting the warp

[4] S. Villena, M. Vega, D. Babacan, R. Molina, and A. Katsaggelos. Using the Kullback-Leibler divergence to combine image priors in super-resolution image reconstruction. In *IEEE International Conference on Image Processing*, pages 809–812. Hong-Kong (China), September 2010.

[5] S. D. Babacan, R. Molina, and A.K. Katsaggelos. Variational Bayesian super resolution. *IEEE Transactions on Image Processing*, 20(4):984–999, 2011.
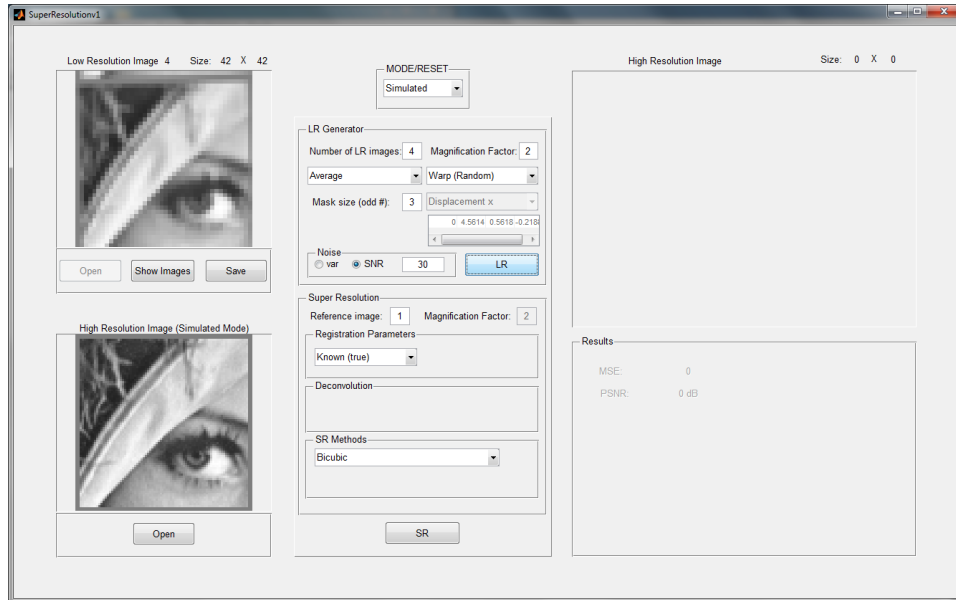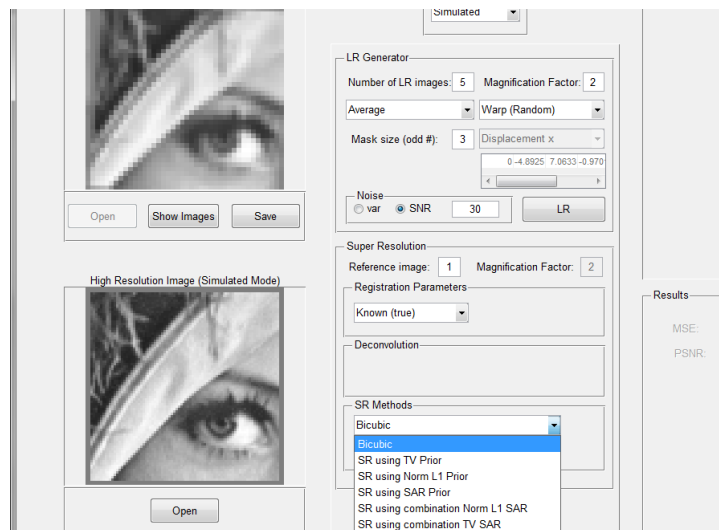
Figure 9: Observations generation

Figure 10: Available SR methods.
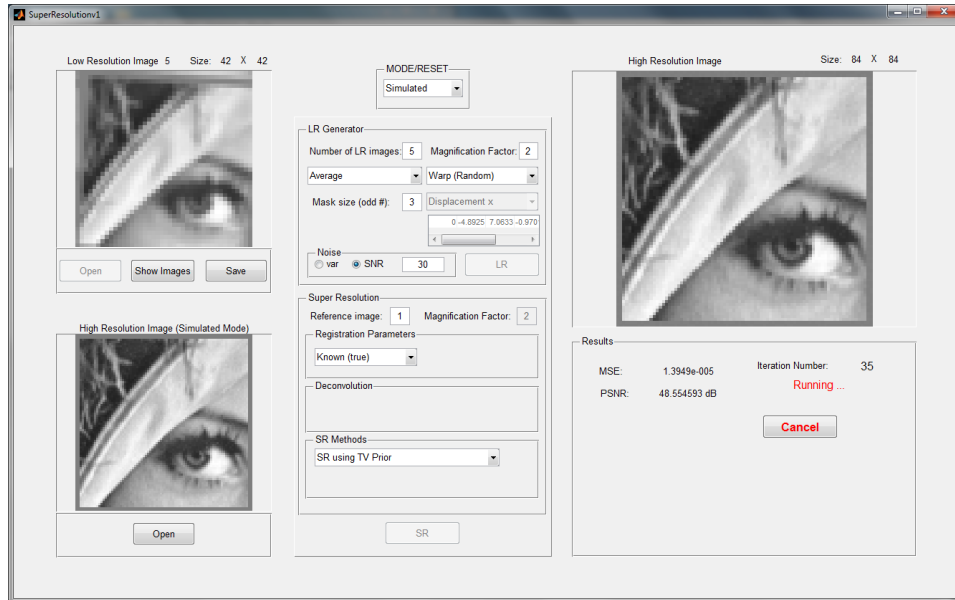
Figure 11: Running the SR method.
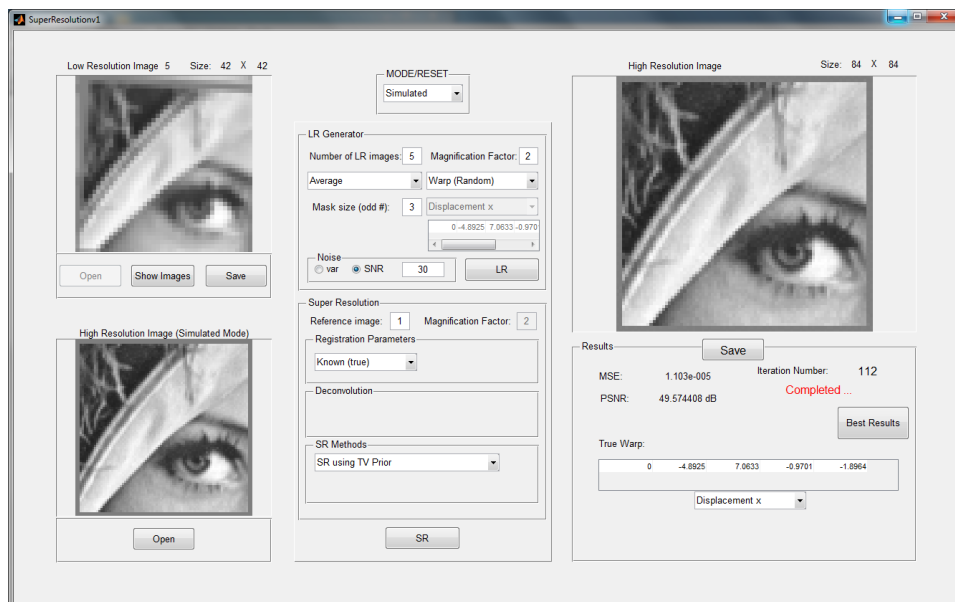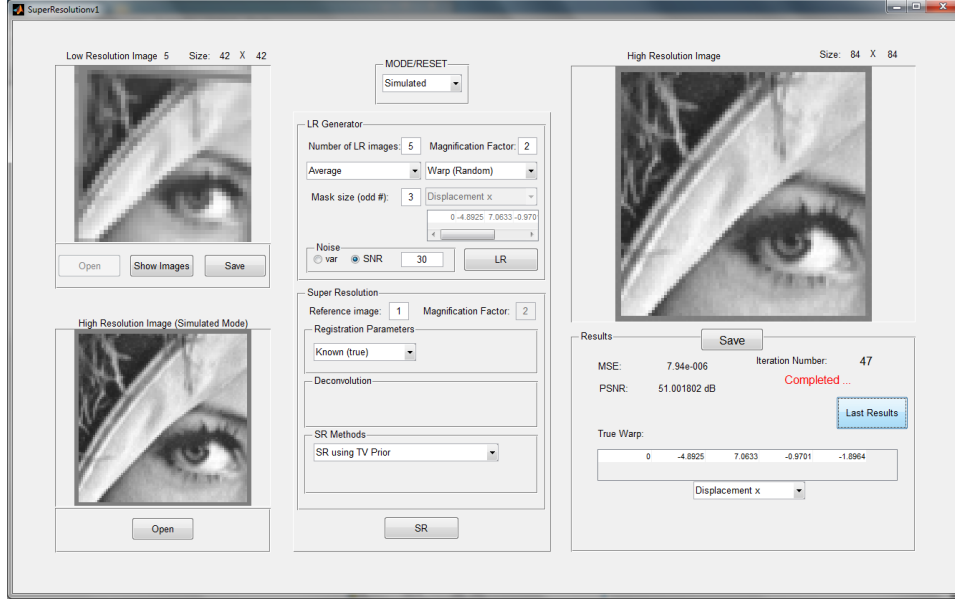

Figure 12: Execution of the SR method finalized.

Figure 13: Display the best HR reconstruction.

Figure 14: Stored parameter in the structure *outopt* when saving the estimated HR image using the method in [2].
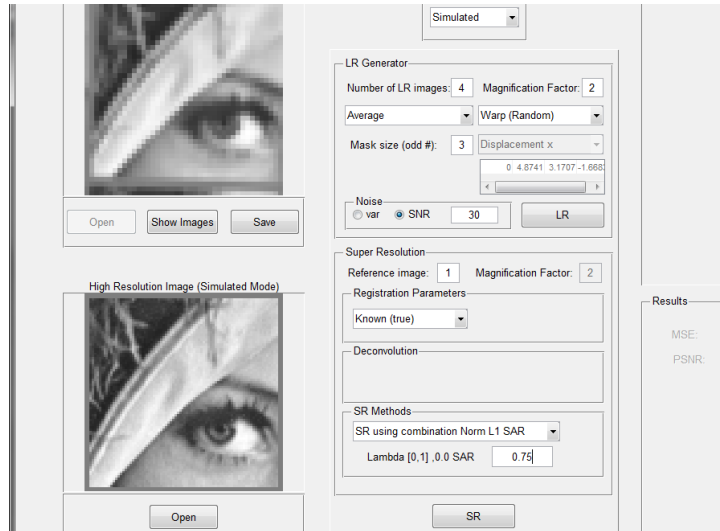
Figure 15: Coefficient value for the prior model combination when method in [3] or [4] is selected.
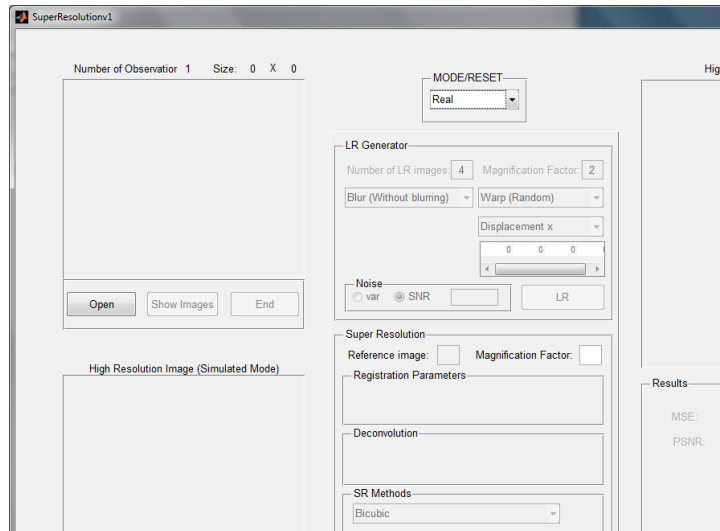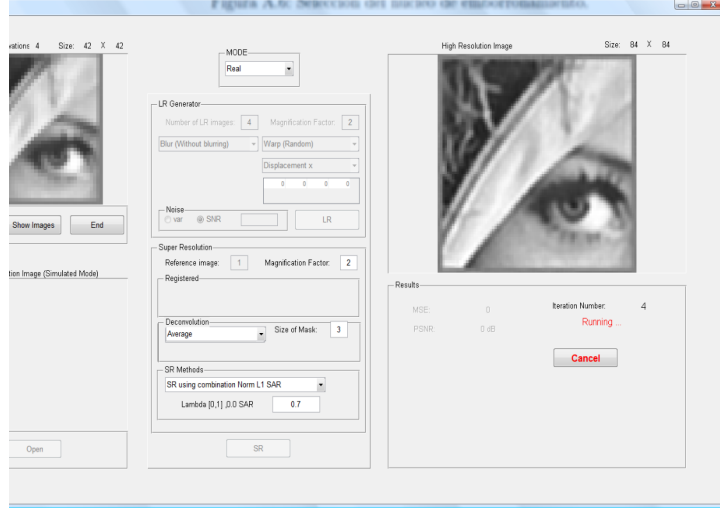

Figure 16: **Real Mode**.

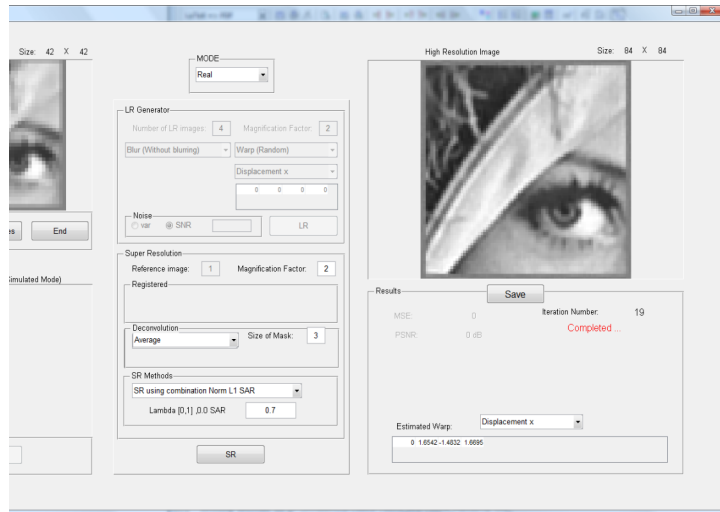Figure 17: Execution of the method in [3] in **Real** mode with $\lambda = 0.7$.

Figure 18: End of execution of method in [3] in **Real** Mode with $\lambda = 0.7$ at iteration 19.

```
Running SR with optimal distributions
-------------------------------------------------------------------

Calculate initial average image
betak =
957.763 1168.29 1152.24 1138.75
alpha_h = 9.32744, alpha_v = 9.47627

-------------------------------------------------------------------
Iteration 1
Finished calculating Sigma registration terms
Running PCG to estimate the image
PCG returned the solution at 88 iterations, flag = 1
Estimating the registration

**********************************************
Image 2
Initial registration    sx = 1.65418, sy = -0.843637, theta = 9.43549
Previous registration  sx = 1.65418, sy = -0.843637, theta = 9.43549
Estimated registration sx = 1.65418, sy = -0.843637, theta = 9.43549

**********************************************
Image 3
Initial registration    sx = -1.48317, sy = 0.214256, theta = 9.14108
Previous registration  sx = -1.48317, sy = 0.214256, theta = 9.14108
Estimated registration sx = -1.48317, sy = 0.214256, theta = 9.14108

**********************************************
Image 4
Initial registration    sx = 1.66946, sy = 1.83849, theta = -0.264915
Previous registration  sx = 1.66946, sy = 1.83849, theta = -0.264915
Estimated registration sx = 1.66946, sy = 1.83849, theta = -0.264915
Estimating the hyperparameters
avg DxSigma = 0.00115548, avg DySigma = 0.00115548
k = 1, e = 0.522118, xOx = 0, traceBkSigma = 0.113152, traceOSigma = 0
k = 2, e = 0.410507, xOx = 0, traceBkSigma = 0.0871786, traceOSigma = 0
k = 3, e = 0.401887, xOx = 0, traceBkSigma = 0.0880384, traceOSigma = 0
k = 4, e = 0.391035, xOx = 0, traceBkSigma = 0.0886754, traceOSigma = 0
betak =
2776.77 3544.41 3600.55 3677.22
alpha_h = 4.7704, alpha_v = 4.84304,xconv = 0.0388807

-------------------------------------------------------------------
Iteration 2
Finished calculating Sigma registration terms
Running PCG to estimate the image
PCG returned the solution at 81 iterations, flag = 3
Estimating the registration

**********************************************
Image 2
Initial registration    sx = 1.65418, sy = -0.843637, theta = 9.43549
Previous registration  sx = 1.65418, sy = -0.843637, theta = 9.43549
Estimated registration sx = 1.65418, sy = -0.843637, theta = 9.43549

**********************************************
Image 3
Initial registration    sx = -1.48317, sy = 0.214256, theta = 9.14108
Previous registration  sx = -1.48317, sy = 0.214256, theta = 9.14108
Estimated registration sx = -1.48317, sy = 0.214256, theta = 9.14108

**********************************************
Image 4
Initial registration    sx = 1.66946, sy = 1.83849, theta = -0.264915
Previous registration  sx = 1.66946, sy = 1.83849, theta = -0.264915
Estimated registration sx = 1.66946, sy = 1.83849, theta = -0.264915
Estimating the hyperparameters
avg DxSigma = 0.00198266, avg DySigma = 0.00198266
k = 1, e = 0.0576815, xOx = 0, traceBkSigma = 0.189113, traceOSigma = 0
k = 2, e = 0.0497142, xOx = 0, traceBkSigma = 0.145384, traceOSigma = 0
k = 3, e = 0.0431147, xOx = 0, traceBkSigma = 0.145948, traceOSigma = 0
k = 4, e = 0.0365582, xOx = 0, traceBkSigma = 0.146958, traceOSigma = 0
betak =
7147.66 9041.6 9330.24 9612.23
alpha_h = 3.69758, alpha_v = 3.86702,xconv = 0.0361037

-------------------------------------------------------------------
```

Figure 19: Content of the log file after running the method in [3] in real mode with $\lambda = 0.7$ in iteration 2.