A Tutorial on Multi-frame Computational Super-resolution using Statistical Methods

Alexandru Paul Condurache,
Institute for Signal Processing, University of Lübeck,
D-23538 Lübeck, Germany
condurache@isip.uni-luebeck.de

***Abstract*—The term super-resolution is used to describe methods aimed at recovering detail information of an imaged scene that would otherwise be lost during the normal imaging process. Here we discuss only digital cameras that process light reflected by the objects constituting a scene. A digital camera, consisting of optics a digital imaging sensor and signal processing hardware is able to capture a digital image of a scene. For digital cameras, resolution has to do with both the number of imaging elements per unit sensor area – which translates into the number of pixels per unit image area – as well as the information content of the digital image. Intuitively speaking, increasing the information content is directly related to increasing the number of pixels per unit area in the digital image such as to properly render the enlarged information content according to the Shannon sampling theorem. This tutorial is concerned with increasing the level of detail (i.e., information content) in a digital image by means of statistical processing algorithms starting from a set of several (usually) alias-afflicted images of the same scene, acquired from slightly different positions. Such techniques fall in the category of multi-frame computational super-resolution. We start be describing the maximum likelihood (ML) solution to this problem and then show how a maximum a posteriori (MAP) approach can improve upon the ML solution. We will discuss several solution strategies relying on such principles and point to their advantages and disadvantages. We conclude with a short overview of alternative super-resolution approaches.**

I. Introduction

An imaging device captures the information of a scene. The resolution of the imaging device describes the level of scene-detail the device is able to capture. The imaging devices of interest here are cameras (consisting of optics and sensor) that are supposed to capture images of a scene. An ideal imaging device should image a point light source as a point, however, a real imaging device does not usually do so. A real camera images a point light source not as point but rather as a patch even under ideal conditions. The reasons for this are due to both the optics and the sensor. Due to diffraction at the aperture, even perfect optics introduce a small blur. A further blur is introduced by the sensor that integrates light over small surfaces (be they semiconductor-based sensing elements in a digital sensor or a silver halid crystals in an 'analog' film), thus details smaller than this surface will be lost. If we consider this setup from the Fourier perspective, we say that the optics and sensor introduce a low-pass filter such that high-frequency details of the imaged scene are eliminated. Under ideal or near-ideal conditions the lost details have a very high frequency but still they are significant for applications like astronomy. In many real-life applications however, optics and sensor are far from ideal and alias plays a major role as well.

Under the term *super-resolution* (SR) we understand all methods aimed at recovering such lost details. These methods can be divided into two categories: those compensating for the optics-induced blur and those compensating for camera-induced blur. Optical super-resolution includes methods aimed at braking the diffraction limit, like for example by interferometry (where the superposition of several electromagnetic waves is used to extract information about each wave). Here we concentrate on computational super-resolution, which includes methods to improve the resolution of a digital camera by processing the digital images. Computational super-resolution can be further divided into two fields: (i) shift-frame-based and (ii) extrapolation-based. The former is concerned with improving the resolution starting from a set of several low-resolution images acquired at slightly different camera-positions. The latter is concerned with extrapolating the information in a (usually) single low-resolution image such as to recover the lost details of the high-resolution image, i.e., inferring the lost details from the available information in the low-resolution image. Shift-frame or multi-frame methods practically work only in the presence of alias [5]. Extrapolation-based methods should also work starting from an alias-free low-resolution image. However, the presence of alias allows the computation of significantly better results. Clearly, in both cases if alias is present in the low-resolution images, then it will be strongly reduced and ideally eliminated in the high-resolution image. This tutorial covers multi-frame computational super-resolution from a statistical perspective.

In this context it is assumed that the observed data is generated the following way: a scene is imaged in a high-resolution (HR) Nyquist-conform image $\mathbf{x}$, from this image, several other low-resolution (LR) images $\mathcal{Y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_K\}$ are generated. The process by which the LR images are generated includes geometric transforms (usually up to projective transforms, i.e., homographies), different lighting changes λ, blurring by various point spread functions (PSF), decimation by a zoom factor such as to obtain a sub-Nyquist sampling grid and corruption by additive noise. This data-generation model is illustrated in Figure 1. Clearly not the entire original

If this technical report helped you in your research and want to use it, please cite as well "Statistical Pattern Recognition for Biometric Person Identification and Event Detection: Hysteresis and Sparse Classifiers, Dynamic Bayes Networks and the Gaussianity assumption" (see https://www.isip.uni-luebeck.de/mitarbeiter/alexandru-condurache.html), which originated it.

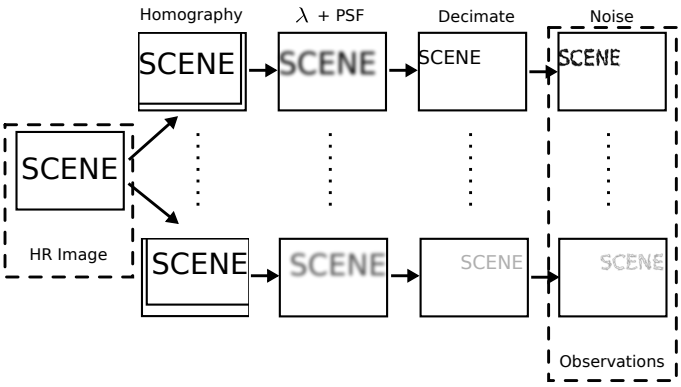

Fig. 1. The relationship between the set of observed low-resolution images and the sought high-resolution image.

HR image can be reconstructed, but only a region. This is the region where all LR images overlap.

Accordingly, we have that each observed image is computed from the HR image as

$$\mathbf{y}^{(k)} = \lambda_1^{(k)}\mathbf{W}^{(k)}\mathbf{x} + \lambda_2^{(k)}\mathbf{1} + \mathbf{s}^{(k)}, \tag{1}$$

with $k = 1, \ldots, K$ and where each $\mathbf{W}^{(k)}$ accounts for warping, blurring and decimation (i.e., subsampling) in the respective LR observation $\mathbf{y}^{(k)}$. $\mathbf{s}^{(k)}$ is additive, uncorrelated Gaussian noise $N(\mathbf{0}, \beta^{-1}\mathbf{I})$.

Next we are going to discuss statistical methods that will allow us to estimate the high-resolution image $\mathbf{x}$ starting from the set of observed low-resolution images $\mathcal{Y}$. Using the above generative model, we begin with the ML approach for the SR problem. The ML solution is strongly afflicted by noise. The MAP solution will then be introduced as a way to increase robustness.

Going from simple to complicated, the initial ML and MAP solutions assume that the warping (accounting for camera motion between the LR images) and the blur (accounting for optic and sensor blurring) have been estimated before entering the stochastic setup, therefore $\mathbf{W}^{(k)}$ can be directly computed from these estimates together with the decimation factor. Assuming further that the lighting parameters (accounting for relative changes in illumination between LR images) have been also estimated beforehand, the stochastic setup returns 'just' the HR image estimate. In this case the parameters of equation (1) are not established simultaneously, i.e., they are established separated, as if they were independent although they are not independent[1]. Thus the next improvement is the simultaneous computation of all involved parameters, i.e, HR image and warping, blurring and lighting.

Warping, blurring and lighting constitute the "nuisance" parameters. Errors in the nuisance parameters afflicts the SR image. Under a set of assumptions, including the assumption of a Gaussian image prior, better results can be achieved with the help of marginalization, either over the nuisance parameters to obtain the SR image or over the SR image to get the nuisance parameters and then the optimal SR image from the above generative model (1). Of course there are alternative paradigms to obtain the SR image, like for example the ℓ_1 based optimization of Farisu et al. [3] or other methods as described in [5]. Simultaneous methods, marginalization and alternative SR paradigms are briefly discussed in the end of this tutorial.

[1]For example, the lightning parameters may depend on the warping (i.e., the assumed position of the LR camera).

II. The ML solution

Assuming the motion and the lighting changes between the observed LR frames as well as the PSF are known, we would like to compute the likelihood $\mathcal{L}(\mathcal{Y}|\mathbf{x}) = \prod_{k=1}^{K} p(\mathbf{y}^{(k)}|\mathbf{x})$. Maximizing this likelihood over the unknown parameter $\mathbf{x}$ returns the sought estimate of the HR image.

A. The nuisance parameters

The warping, or motion model and the blurring or PSF are needed to compute $\mathbf{W}$. The lighting parameters $\boldsymbol{\lambda}$ complete the generating model (1). The additive noncorrelated Gaussian noise term will be discussed later.

In practice, as we have access only to a set of LR images, we assume that one of the LR images contains the imaged scene from the same perspective as the HR image. Equivalently, this LR image is obtained from the HR image only by blurring and downsampling, the warping being the identity matrix, $\lambda_2 = 0$ and $\lambda_1 = 1$, or we can also say that there is neither (camera) motion nor lighting variations between this LR image and the sought HR image. This particular image is called next the "main LR image". It is chosen randomly from the set of LR images and remains then fixed over the entire SR algorithm.

1) Motion estimation: The warping corresponding to a LR image is given by the transform that describes the displacement of the observed scene between this image and the HR image. There are several ways to model this displacement, i.e., we may choose from among various types of transforms (see [6] pp. 22). Considering that we model here camera motion, a projective transform (8 DoF) is enough, furthermore very often in practice even an affine model (6 DoF) suffices.

a) The motion model: In the 2D case an affine model relating a vector $\mathbf{v}$ to a vector $\mathbf{u}$ is given by:

$$\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} + \begin{bmatrix} a_5 \\ a_6 \end{bmatrix}.$$

This may be written in compact form using homogeneous coordinates $\mathbf{v}_h$ and $\mathbf{u}_h$ instead of Cartesian coordinates as:

$$\begin{bmatrix} v_1 \\ v_2 \\ 1 \end{bmatrix} = \begin{pmatrix} a_1 & a_2 & a_5 \\ a_3 & a_4 & a_6 \\ 0 & 0 & 1 \end{pmatrix} \begin{bmatrix} u_1 \\ u_2 \\ 1 \end{bmatrix}.$$

In homogeneous coordinates, a projective transform is given by:

$$\begin{bmatrix} v_1' \\ v_2' \\ v_3' \end{bmatrix} = \begin{pmatrix} a_1 & a_2 & a_5 \\ a_3 & a_4 & a_6 \\ a_7 & a_8 & a_9 \end{pmatrix} \begin{bmatrix} u_1 \\ u_2 \\ 1 \end{bmatrix}, \text{ with } \mathbf{v}_h = \frac{\mathbf{v}'}{v_3}.$$

In Cartesian coordinates, a projective transform would be nonlinear.

To compute the sought motion model for one LR frame from the available set, we have thus to estimate in the affine case a number of six parameters that describe the registration between the respective LR image and the main LR image. Registration is a well-studied problem and there are many algorithmic solutions [6]. We are here interested in parametric registration methods, meaning that the transform is available in close form, in our case being given by the linear relationship

$$\mathbf{v}_h = \mathbf{A}\mathbf{u}_h, \tag{2}$$

with $\mathbf{A}$ the matrix of the our affine transform. Again, this is an even better studied problem with many available solutions including landmark based approaches, mutual information based approaches, etc. For reasons of simplicity we concentrate here on landmark-based approaches.

b) Parameter estimation: For our 2D affine example we need to estimate a number of six parameters. Considering the main LR image as the reference with coordinates $\mathbf{v}$ and the other LR image as the template with coordinates $\mathbf{u}$, we need to know the coordinates of at least three noncollinear scene points in both reference and template, such as to be able to solve over the unknown parameters $\{a_1, \ldots, a_6\}$ an equation system derived from (2).

If the scene points are collinear, the system matrix is rank-deficient and there is no unique solution. Furthermore, in practice more than three such points are usually detected (which ensures noncollinearity) and there is no exact solution. In both cases, a satisfactory solution can be computed, for example with the help of the pseudoinverse or by singular value decomposition. This satisfactory solution minimizes the residuals $r = |\mathbf{A}\mathbf{u}_h - \mathbf{v}_h|$. In the former case it has minimal ℓ_2 norm, and in the latter case it minimizes the square error (i.e., it is the least squares solution, as the residuals are defined with the help of the ℓ_2 norm). If the pseudoinverse is itself close to singular or even singular, the solution that minimizes the residuals can be found in an iterative manner, for example with gradient descent.

c) Landmark detection: A coordinate pair corresponding to the same scene point in both template and reference is called a landmark pair and the scene point itself is called a landmark. Good landmarks should be invariant to the considered motion model, meaning that they should not change their appearance between reference and template. Widely used landmarks in our context are corner points. They are invariant to rotation and translation and largely invariant to small affine distortions.

Corners are points in whose vicinity there are two local orientations. The vicinity of a point exhibits single orientation when the gray levels change in only one direction. By analyzing the local orientation, a corner descriptor can be computed [1] that exhibits the needed invariance properties such that it can be used to detect landmark pairs.

The 2D differential operator in direction ϕ with respect to the horizontal axis is defined as the scalar projection of the gradient vector $\nabla[\cdot] = \left[\frac{\partial}{\partial x}, \frac{\partial}{\partial y}\right]^T$ on the orientation vector $\mathbf{n}(\phi) = [\cos(\phi)\sin(\phi)]^T$. The orientation vector points on a direction perpendicular to that along which the gray levels vary. As the orientation vector has unit norm, the scalar projection equals the scalar product of the two vectors and thus the sought differential operator may be written as:

$$\begin{aligned} \alpha(\phi)[\cdot] &= \text{prj}(\nabla, \mathbf{n}(\phi)) \\ &= \langle \nabla, \mathbf{n}(\phi) \rangle \\ &= \mathbf{n}(\phi)^T \cdot \nabla[\cdot] \\ &= cos(\phi)\frac{\partial}{\partial x} + sin(\phi)\frac{\partial}{\partial y}. \end{aligned} \tag{3}$$

If the image $f(x, y)$ is ideally oriented at (x, y) under an angle θ, then its derivative in along θ should be zero:

$$\alpha(\theta)[f(x, y)] = \mathbf{n}(\theta)^T \cdot \nabla[f] = 0. \tag{4}$$

When this condition is met for θ is met also for $\theta + \pi$ and for $\theta - \pi$, thus by convention $\theta \in (-\frac{\pi}{2}\frac{\pi}{2})$.

Local orientation as described by $\mathbf{n}(\theta)$ is evaluated over a vicinity/neighborhood Ω within which it is assumed to be constant. To compute $\mathbf{n}(\theta)$, we need the orientation angle θ. We find this angle as the argument that minimizes the functional

$$\theta = \min_{\phi} Q(\phi) = \min_{\phi} \left[\int_{\Omega} (\mathbf{n}^T \nabla f)^2 d\Omega \right], \tag{5}$$

where $\mathbf{n} \equiv \mathbf{n}(\theta)$ and with $\mathbf{n}^T\mathbf{n} = 1$. This can be rewritten as

$$Q(\phi) = \mathbf{n}^T \mathbf{T} \mathbf{n}, \tag{6}$$

with $\mathbf{T}$ being a 2×2 tensor computed as:

$$\mathbf{T} = \int_{\Omega} \nabla f (\nabla f)^T d\Omega = \int_{\Omega} \begin{bmatrix} f_x^2 & f_x f_y \\ f_y f_x & f_y^2 \end{bmatrix} d\Omega. \tag{7}$$

Thus, to find our local orientation descriptor $\mathbf{n}$ we have now to minimize the composite criterion

$$L(\mathbf{n}) = \mathbf{n}^T \mathbf{T} \mathbf{n} + \lambda(\mathbf{n}^T \mathbf{n} - 1), \tag{8}$$

including the condition $\mathbf{n}^T\mathbf{n} = 1$ over the Lagrange multiplier λ. This is equivalent to finding $\mathbf{n}$ such that

$$\mathbf{T}\mathbf{n} = \lambda \mathbf{n}, \tag{9}$$

i.e., finding the normalized eigenvector of $\mathbf{T}$ corresponding to the lower eigenvalue λ. As $Q(\phi)$ is a measure of variation of $f(x, y)$ in the direction ϕ, then the minimum quantity of variation in Ω is given by λ:

$$Q(\phi) = \mathbf{n}^T \mathbf{T} \mathbf{n} = \mathbf{n}^T \lambda \mathbf{n} = \lambda. \tag{10}$$

Practically, $\mathbf{T}$ is computed as

$$\begin{bmatrix} S(D_x \cdot D_x) & S(D_x \cdot D_y) \\ S(D_y \cdot D_x) & S(D_y \cdot D_y) \end{bmatrix}, \tag{11}$$

with S a smoothing operator which defines the size of the neighborhood and $D_{x,y}$ the derivative operator on direction x respectively y. The results obtained by derivation are multiplied pixelwise.

$\mathbf{T}$ is also known as the structure tensor and its eigenvalues can be used to analyze the local orientation, its' second eigenvalue being an indication for the presence of a strong second orientation. Together with $\mathbf{n}$ it can be used to compute a corner descriptor.

Better corner descriptors can be computed be explicitly

considering the second orientation as well. In this case there is a θ_1 for the main orientation and a θ_2 for the other orientation in Ω. Two orientations at (x, y) means that

$$\alpha(\theta_1)\alpha(\theta_2)f(x,y) = \mathbf{n}'\mathbf{d}f = 0,$$

with $\mathbf{n}' = (\cos\theta_1 \cos\theta_2, \sin(\theta_1 + \theta_2), \sin\theta_1 \sin\theta_2)^T$ and $\mathbf{d}f = (f_{xx}, f_{xy}, f_{yy})^T$. This is equivalent to analyzing a 3×3 tensor $\mathbf{T}_2$ containing second order derivatives.

2) Lighting estimation: After registration the lighting parameters can be computed from pairs of gray-levels of the same pixel in the reference and the transformed template. For our 2D example we would need at least two pairs to compute $\boldsymbol{\lambda} = (\lambda_1, \lambda_2)^T$

$$\begin{aligned} r_{\mathbf{x}_1} &= \lambda_1 t'_{\mathbf{x}_1} + \lambda_2 \\ r_{\mathbf{x}_2} &= \lambda_1 t'_{\mathbf{x}_2} + \lambda_2 \end{aligned},$$

with $\mathbf{x}_{1,2}$ two position vectors, $r_{\mathbf{x}_i}$ the gray level at pixel i in the reference image and $t'_{\mathbf{x}_i}$ the gray level at pixel i in the transformed template. As discussed before, usually more than two pairs are selected and the solution is found with the help of least squares.

3) Blur estimation: Blur is generated by the optics and by the integration over the active area of each sensor pixel. Both influences are described either as an isotropic 2D Gaussian kernel (covariance matrix $\boldsymbol{\Sigma} = \sigma^2\mathbf{I}$) or as a disc of radius d (equivalent to a moving average sort of filter – but potentially with weights different than $\frac{1}{n}$, with n the number of pixels in the disk).

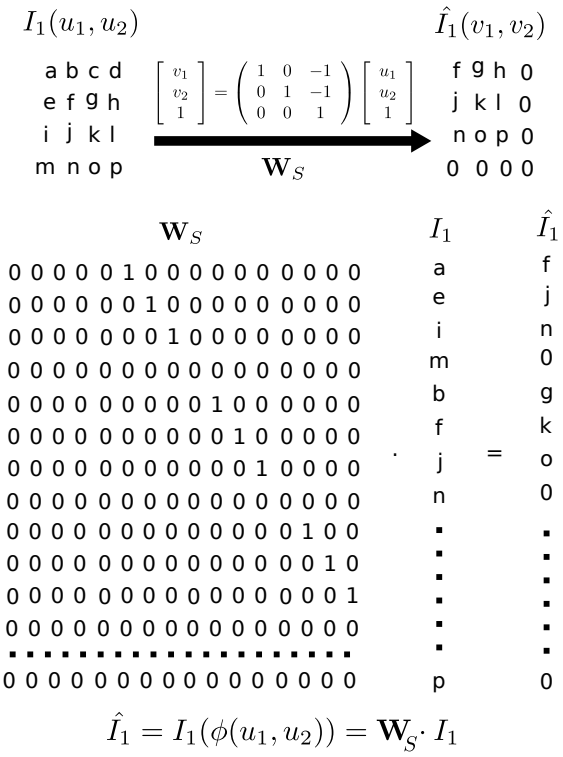

Fig. 2. $\mathbf{W}_S$ corresponding to a shift.

B. How to generate $\mathbf{W}$

The matrix $\mathbf{W}$ in our model (1) encodes the motion model, the blur and the zoom factor (i.e., the downsampling factor relating a LR image to the HR image). All mathematical operations discussed here work on vectors; to obtain a vector from an image, by convention, all columns are stacked one underneath the other starting with the left-most column.

Under these circumstances, a shift in an image with $L = M \times N$ pixels corresponds to a multiplication of the corresponding $L \times 1$ image vector with a $L \times L$ matrix where a few entries are one, the rest being zero. Such an example is shown in Figure 2.

Blurring an image I_1 with a disk-like blur kernel B can be also expressed in this setup. For this purpose, for every position where the blur is applied, an image of the same size as the input image is generated containing the blur kernel at the respective position and otherwise only zeros. To apply the blur, corresponding blur-image vector is multiplied with the image vector. To apply the blur at various positions in the original image, the image vector is multiplied with a matrix whose lines are blur-image vectors. This is illustrated in Figure 3 for a blur kernel B defined as:

$$B = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}.$$

Downsampling an image by keeping only every second line and every second column is equivalent to multiplying I_1 with a $\mathbf{W}_\downarrow$ with $\frac{L}{4}$ lines and L columns with ones at the positions

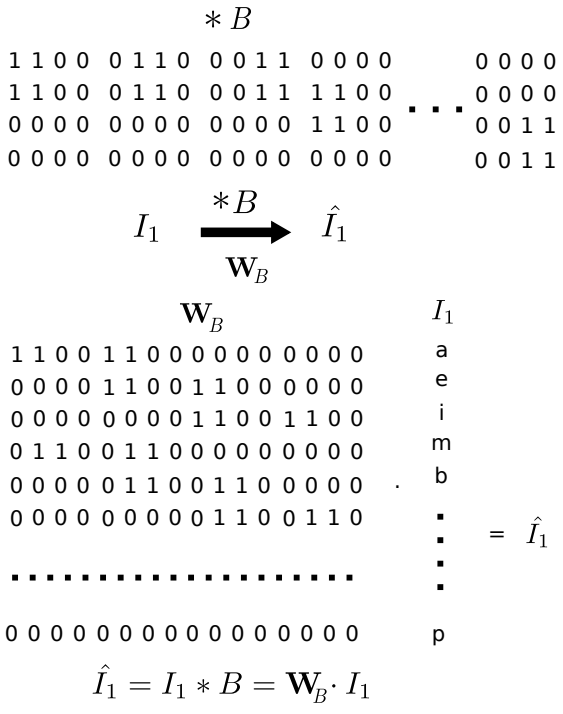

Fig. 3. $\mathbf{W}_B$ corresponding to blurring.

corresponding to the elements that are supposed to be kept and zeros otherwise.

Combining several operations O is equivalent to multiplying the corresponding $\mathbf{W}_O$ matrices, thus the sought matrix is: $\mathbf{W} = \mathbf{W}_S\mathbf{W}_B\mathbf{W}_{\downarrow}$.

C. The HR image

After finding the nuisance parameters we are now ready to estimate the HR image. For this we will model parametrically the distribution of LR images given the sought HR image. Using this distribution we will compute the HR image in a ML approach. The ML approach includes the computation of the likelihood function and the finding of its maximum over the HR image.

1) The statistical model: Establishing a model for the distribution of LR images given the sought HR image will allow us to compute the likelihood function for our problem. A statistical model may be generated in this case either starting from the error term $\mathbf{s}^{(k)}$ in Equation (1) or using the principle of maximum entropy.

a) Error-based model: Starting from the generative model in Equation (1) a statistical model can be derived using the fact that the errors $\mathbf{s}^{(k)}$ in the model are assumed Gaussian distributed:

$$\mathbf{s}^{(k)} = \mathbf{y}^{(k)} - \left(\lambda_1^{(k)}\mathbf{W}^{(k)}\mathbf{x} + \lambda_2^{(k)}\mathbf{1}\right) : N(\mathbf{0}, \beta^{-1}\mathbf{I}).$$

As $\mathbf{W}$ and $\boldsymbol{\lambda}$ are known, it follows directly that

$$\begin{aligned} p(\mathbf{y}^{(k)}|\mathbf{x}) &= \frac{1}{2\pi^{\frac{M}{2}}\left|\boldsymbol{\Sigma}\right|^{-\frac{1}{2}}} e^{-\frac{1}{2}\left[\mathbf{y}^{(k)}-\mathbf{m}^{(k)}(\mathbf{x})\right]^T\boldsymbol{\Sigma}^{-1}\left[\mathbf{y}^{(k)}-\mathbf{m}^{(k)}(\mathbf{x})\right]} \\ &= \left(\frac{\beta}{2\pi}\right)^{\frac{M}{2}} e^{-\frac{\beta}{2}\|\mathbf{y}^{(k)}-\mathbf{m}^{(k)}(\mathbf{x})\|_2^2}, \end{aligned} \tag{12}$$

with $\mathbf{m}^{(k)}(\mathbf{x}) = \lambda_1^{(k)}\mathbf{W}^{(k)}\mathbf{x} + \lambda_2^{(k)}\mathbf{1}$.

b) Maximum entropy-based model: A more interesting derivation of the same model uses the principle of maximum entropy [4]. This principle can be used to construct a density for a random variable from which some information is available. The needed information in this case is not a set of observations, but knowledge over the value of the expectation of some deterministic functions of this random variable. If a set of observations is also available the quality of our density information can be improved.

More formally and taking for convenience the scalar case, we would like to find the distribution function of a random variable x, while knowing that the expectations $E\{f_i(x)\}$ of a number n of functions $f_i(x), i = 1, \ldots, n$ have known values c_i:

$$\int p(x)f_i(x)dx = c_i. \tag{13}$$

This is an ill-posed problem, as we would like to find the true $p(x)$, but there are several functions that verify the constraint (13). In an attempt to regularize this problem, we decide to choose the distribution that has maximal entropy, based on the consideration that such a distribution makes the least assumptions about the available data and is thus the most general.

It can be shown [2], [7] that the maximum-entropy density satisfying (13) is given by

$$p_0(x) = Ae^{\sum_i a_i f_i(x)}, \tag{14}$$

where A and a_i are some constants. These constants can be found as the solution to a system of equations consisting of i equations found by introducing (14) into (13) plus an additional equation given by $\int p(x)dx = 1$ (all integrals are considered over the entire definition domain of x). This system of $i+1$ equations with $i+1$ unknowns can be solved exactly if it is not degenerate.

There is a close relationship between the maximum entropy distribution and the Gaussian distribution. It can be shown that for a zero-mean random variable with unit variance, the maximum entropy density is the Gaussian distribution. Even more, irrespective of the mean, the maximum entropy distribution of a random variable at given variance is the Gaussian distribution (see [4] pp. 112).

In our case, starting from Equation (1) we consider the difference

$$f(\mathbf{y}) = \mathbf{y} - \left(\lambda_1\mathbf{W}\mathbf{x} + \lambda_2\mathbf{1}\right),$$

to be a function of the random variable $\mathbf{y}$, depending on the parameter $\mathbf{x}$ (where we dropped the realization index k to simplify the notation). We know that for a given $\mathbf{x}$, this function has an expectation of zero, and thus $f_1(\mathbf{y}) = \|f(\mathbf{y})\|_2^2$ has also zero expectation. Then, making use of the principle of maximum entropy we may compute:

$$\begin{aligned} p(\mathbf{y}|\mathbf{x}) &= Ae^{a_1 f_1(\mathbf{y})} \\ &= Ae^{a_1\|\mathbf{y}-(\lambda_1\mathbf{W}\mathbf{x}+\lambda_2\mathbf{1})\|_2^2}. \end{aligned}$$

Solving for A and a_1 as described above, we find that $\mathbf{y}$ is Gaussian distributed according to

$$p(\mathbf{y}|\mathbf{x}) = \left(\frac{\beta}{2\pi}\right)^{\frac{M}{2}} e^{-\frac{\beta}{2}\|\mathbf{y}-(\lambda_1\mathbf{W}\mathbf{x}+\lambda_2\mathbf{1})\|_2^2}, \tag{15}$$

which is the same as Equation (12).

2) The likelihood function: Assuming that the set of observed LR images is independently sampled from (15), the likelihood of the observed data is given by

$$p(\mathcal{Y}|\mathbf{x}) = \left(\frac{\beta}{2\pi}\right)^{\frac{KM}{2}} e^{-\frac{\beta}{2}\sum_{k=1}^{K}\left\|\mathbf{y}^{(k)}-\left(\lambda_1^{(k)}\mathbf{W}^{(k)}\mathbf{x}+\lambda_2^{(k)}\mathbf{1}\right)\right\|_2^2},$$

with $\mathcal{Y} = \{\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(K)}\}$. The maximum likelihood HR image is then:

$$\hat{\mathbf{x}}_{ML} = \arg\max_{\mathbf{x}} \left(p(\mathcal{Y}|\mathbf{x})\right).$$

In practice, the sought estimate is found by optimizing the log-likelihood function, which simplifies the matter in so far as we have to deal now with polynomial and not with exponential functions. We have thus to find the maximum of

$$\log\left(p(\mathcal{Y}|\mathbf{x})\right) = \log\left(\gamma\right) - \frac{\beta}{2}\sum_{k=1}^{K}\left\|\mathbf{s}^{(k)}\right\|_2^2,$$

with $\gamma = \left(\frac{\beta}{2\pi}\right)^{\frac{KM}{2}}$ and $\mathbf{s}^{(k)} = \mathbf{y}^{(k)} - \left(\lambda_1^{(k)}\mathbf{W}^{(k)}\mathbf{x} + \lambda_2^{(k)}\mathbf{1}\right)$.

a) Optimization: Finding the maximum of $\log\left(p(\mathcal{Y}|\mathbf{x})\right)$ is equivalent to finding over $\mathbf{x}$ the extreme point (i.e., minimize) of the following objective function

$$\mathcal{L} = \frac{1}{2}\sum_{k=1}^{K}\left\|\mathbf{s}^{(k)}\right\|_2^2,$$

where we ignore terms not contributing to the optimum. Considering that $\mathbf{a}^T\mathbf{a} = \|\mathbf{a}\|_2^2$, the derivative over $\mathbf{x}$ of the objective function is:

$$\frac{\partial\mathcal{L}}{\partial\mathbf{x}} = -\sum_{k=1}^{K}\lambda_1^{(k)}\mathbf{W}^{(k)}\mathbf{s}^{(k)}.$$

By setting this derivative to zero we obtain for the sought estimate:

$$\hat{\mathbf{x}}_{ML}=\left(\sum_{k=1}^{K}{\lambda_1^{(k)}}^2{\mathbf{W}^{(k)}}^T\mathbf{W}^{(k)}\right)^{-1}\left[\sum_{k=1}^{K}\lambda_1^{(k)}{\mathbf{W}^{(k)}}^T\left(\mathbf{y}^{(k)}-\lambda_2^{(k)}\mathbf{1}\right)\right].$$

If the involved matrices are singular, or to large to compute efficiently, an iterative solution can be obtained using the gradient descent method as:

$$\mathbf{x}(i+1) = \mathbf{x}(i) + \sum_{k=1}^{K}\lambda_1^{(k)}\mathbf{W}^{(k)}\mathbf{s}^{(k)}(i). \tag{16}$$

Stochastic gradient descent methods (in which case the sum is eliminated in (16) and the optimization proceeds in epoques, where per epoque all LR image are used once) or better iterative algorithms like the Newton-Gauss algorithm can also be used.

III. The MAP solution

The ML solution has as departing point $p(\mathcal{Y}|\mathbf{x})$, while we are actually interested in the posterior $p(\mathbf{x}|\mathcal{Y})$. By the Bayes rule, this posterior is computed using the likelihood as:

$$p(\mathbf{x}|\mathcal{Y}) = \frac{p(\mathcal{Y}|\mathbf{x})p(\mathbf{x})}{p(\mathcal{Y})}.$$

If the available data is held constant[2], the evidence $p(\mathcal{Y})$ is just a normalization factor and can be ignored in the optimization setup. We can then look for the $\mathbf{x}$ that maximizes this posterior as:

$$\hat{\mathbf{x}}_{MAP} = \arg\max_{\mathbf{x}}\left(p\left(\mathcal{Y}|\mathbf{x}\right)p(\mathbf{x})\right).$$

In this case objective function becomes

$$\mathcal{L} = -\log(p(\mathbf{x})) + \frac{\beta}{2}\sum_{k=1}^{K}\left\|\mathbf{s}^{(k)}\right\|_2^2,$$

and its derivative with respect to $\mathbf{x}$ is:

$$\frac{\partial\mathcal{L}}{\partial\mathbf{x}} = -\frac{\partial\log(p(\mathbf{x}))}{\partial\mathbf{x}} - \sum_{k=1}^{K}\lambda_1^{(k)}\mathbf{W}^{(k)}\mathbf{s}^{(k)}. \tag{17}$$

In order to be able to optimize $\mathcal{L}$ over $\mathbf{x}$, we have to know the image prior $p(\mathbf{x})$.

[2]This means that the components of the set $\mathcal{Y}$ remain the same, no components are exchanged, no new components are added.

A. Image priors

The ML solution has problems in particular when the HR image is sought starting from a small set of noisy LR images. The MAP approach can improve upon the ML solution in this respect, as it offers – by means of the prior – the possibility to introduce prior knowledge into the problem formulation. The prior should be chosen such as to avoid implausible ML solution. An implausible ML solution is in our case an image immersed in noise. As noise has mainly high-frequency components, a prior that encourages smooth images is needed. Conversely, edges are also high-frequency components, but they should be preserved.

The most simple thing to do would be to assume a zero mean Gaussian image prior:

$$p(\mathbf{x}) = 2\pi^{-\frac{L}{2}}\left|\mathbf{Z}\right|^{-\frac{1}{2}}e^{-\frac{1}{2}\mathbf{x}^T\mathbf{Z}^{-1}\mathbf{x}}. \tag{18}$$

Such a prior would encourage smooth, zero-mean images. In this case, the objective function (17)[3] becomes:

$$\frac{\partial\mathcal{L}}{\partial\mathbf{x}} = 2\mathbf{Z}^{-1}\mathbf{x} - \sum_{k=1}^{K}\lambda_1^{(k)}\mathbf{W}^{(k)}\mathbf{s}^{(k)}.$$

By specifying the elements of $\mathbf{Z}$, we could encourage various behaviors. For example a small-norm covariance matrix would strongly penalize gradients, while specifying a covariance matrix with a larger norm, would allow for some gradients.

Defining $\mathbf{Z}^{-1} = \psi\mathbf{D}^T\mathbf{D}$, with $\mathbf{D}$ approximating a gradient filter (e.g., in the horizontal direction)[4], is equivalent to computing in the exponent of (18) the square of the magnitude of the gradient of $\mathbf{x}$ in the same horizontal direction. This would penalize variations in $\mathbf{x}$, but not as strongly as for example $\mathbf{Z} = \mathbf{I}$.

With the elements of $\mathbf{Z}$ defined as

$$Z(i,j) = Ae^{-\frac{\|\mathbf{v}_i-\mathbf{v}_j\|^2}{r^2}},$$

with $\mathbf{v}_i$ the 2D position vector of the pixel i in the image vector, and r a scale factor, we obtain a covariance matrix that does not penalize large variations as strong as $\mathbf{Z}^{-1} = \psi\mathbf{D}^T\mathbf{D}$, and it allows thus edges.

Most often in practice, the Huber prior is used [5]. This is built explicitly to penalize large gradients less than small ones.

IV. Discussion and Conclusions

While there are many methods for computational super-resolution available, we have concentrated here on a simple ML/MAP-based statistical framework. This straightforward approach includes several steps whose completion involves expert knowledge, like for example the choosing of a prior. Should this be an issue in practice, we can move towards a more data-driven approach by learning the prior parameters from some training data.

[3]As both the prior and the likelihood $p(\mathbf{y}^{(k)}|\mathbf{x})$ are Gaussian, a closed-form solution can be also obtained in this case. Due to the size of the involved matrices, the iterative approach is better suited in practice.

[4]$\mathbf{D}$ is not the horizontal derivative filter kernel $[-1, 1]$, but is constructed starting from this kernel in a manner similar to the one used to construct $\mathbf{W}_B$ to approximates a blur in Figure 3.

More involved statistical approaches have also been proposed. The best promise for an optimal mix between a data-driven and an expert-knowledge method is perhaps given by variational methods. In this case the super-resolved image together with the parameters are considered hidden random variables and we attempt to estimate the distribution of the hidden variables given the observed data. Still, the basic statistical framework upon which the variational methods build involves expert knowledge.

REFERENCES

[1] T. Aach, I. Stuke, C. Mota, and E. Barth. Estimation of multiple local orientations in image signals. In *Proceedings of International Conference on Acoustics Speech and Signal Processing (ICASSP)*, pages III 553–556, Montreal, Canada, May 17–21 2004. IEEE.

[2] T. M. Cover and J. A Thomas. *Elements of Information Theory*. J. Willey & Sons Inc., second edition, 2006.

[3] S. Farsiu, M. D. Robinson, M. Elad, and P. Milanfar. Fast and robust multiframe super resolution. *IEEE Transactions on Image Processing*, 13(10):1327 – 1344, 2004.

[4] A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. Wiley and Sons, 2001.

[5] P. Milanfar, editor. *Super-resolution Imaging*. CRC PRess, 2011.

[6] J. Modersitzki. *Numerical methods for image registration*. Oxford university press, 2004.

[7] A. Papoulis. *Probability & statistics.* Prentice-Hall International, Englewood Cliffs, 1990.