Available online at www.elixirpublishers.com (Elixir International Journal)

Electrical Engineering

Elixir Elec. Engg. 41 (2011) 5746-5749

ISSN: 2229-712X

A non-neighbourhood super-resolution using phantom hashing

Kathiravan S[1] and J. Kanakaraj[2]

[1]Department of Electronics and Communication Engineering, Kalaignar Karunanidhi Institute of Technology, Kannampalayam, Coimbatore 641402, India

[2]Department of Electrical and Electronics Engineering, PSG College of Technology, Peelamedu, Coimbatore 641004, India.

ARTICLE INFO

Article history:

Received: 17 September 2011;

Received in revised form: 16 November 2011;

Accepted: 28 November 2011;

Keywords

Super-resolution,

Phantom hashing,

Non-neighbourhood.

ABSTRACT

Super-resolution is the process of creating an high resolution image from a low resolution input sequence. To overcome the difficulties of good image registration, several methods have been proposed exploiting the non-neighbourhood intuition, i.e. any data-point can contribute to the final result if it is relevant. These algorithms however limit in practice the search region for relevant points in order to lower the corresponding computational cost. Furthermore, they define the non-neighbourhood relations in the high resolution space, where the true images are unknown. In this work, we introduce the use of phantom hashing to efficiently compute fully non-neighbourhood neighbours. We also restate the super-resolution functional using fixed weights in the low resolution space, allowing us to use resolution schemes that avoid many artifacts.

Introduction

Preamble

Super-resolution (SR) is the task of creating an high resolution image from a sequence of several lower resolution frames. It has many purposes, such as producing high quality stills from a video sequence or up-scaling a movie to an higher resolution. The main intuition to SR methods is to exploit sub-pixel motions between the frames of the input sequence to infer the missing data on the target high resolution grid. While initial work involved the fusion of several image spectra, this approach was then replaced by an inverse problem formulation, which is less sensitive to noise and does naturally handle arbitrary motion. In this context, SR can be stated as an inverse problem, where the Low Resolution (LR) image sequence $(I_{LR}^{k})_{k=0}^{n-1}$ is obtained by applying a blur *B*, a downsampling *D* and geometric warping W_k to the unknown High Resolution (HR) image I_{HR} :

$$I_{HR} = \underset{I_{HR}}{\operatorname{argmin}} \sum_{k=0}^{n-1} \rho\left(I_{LR}^{k}, W_k D B I_{HR}\right) + 2\lambda\kappa\left(I_{HR}\right), \quad (1)$$

The function $\rho(\cdot,\cdot)$ measures the error between the *k*-th frame in the sequence and the solution, while $\kappa(\cdot)$ penalizes solutions that are too far away from the chosen prior. Standard choices are the L_2 norm for the error term $\rho(\cdot)$, and the Tikhonov regularization : $\kappa(I_{HR}) = \|I_{HR}\|^2$, for the prior. Eq.(1) can then be solved using various frameworks : gradient descent, Iterated Back-Projection (IBP) [1], maximum likelihood estimation [2] and various other techniques [3]. Although they lead to well known resolution schemes, these choices have inherent drawbacks. The L_2 norm is not robust to misalignments between the images. Also, the Tikhonov regularization is known to produce over-smoothed results by penalizing abrupt changes in the gradient of the solution, actually blurring the boundaries of the objects. Hence, further works [4] [5] considered using the more robust L_1 norm and the Total Variation (TV) regularization to produce sharper outputs. However, these methods still rely on the accuracy of the motion estimation step.

Non-neighbourhood super-resolution

Related work

To bypass the limitations of motion estimation methods, recent works take advantage of example-based regularization, successfully introduced by [6] for the SR of a single image. This algorithm however required the training of a Markov random field on a huge database, limiting its practical interest. Since a movie contains many redundancies, a subsequent work [7] followed the same intuition as the Non-neighbourhood Means (NL-Means) image and movie denoising algorithm [8]. The intuition at the heart of NL-Means is to exploit self-similarities anywhere in the image. Pixels are described by a patch, which is simply all the values comprised in a small square neighborhood. If two patches are very similar, then the two pixels are very likely to represent the same phenomenon and should be exploited together, even when they are spatially far from each other. Hence, the algorithm in [7] uses non-neighbourhood averaging when fusing together the LR frames interpolated to the final resolution. However, the authors limit the search of self similarities to a small learning window around each pixel to limit the computational overhead. In [9], the authors adapt this approach to the IBP algorithm, again considering non-neighbourhoodity as a post-processing constraint on the up-scaled images, and within the limits of a search window. In this work, we propose to enforce the non-neighbourhood constraints on the low resolution images (Sec. 2.2). This has two main advantages. First, the weights are fixed by the input LR sequence, which allows to compute the gradient of the non-neighbourhood error term. Second, since there are less LR pixels, this makes fewer weights to compute. Furthermore, we propose to use an efficient image hashing algorithm called Phantom Hashing [10] to sort all the input LR patches in a single table, hence leveraging fully non-neighbourhood SR at a moderate computational cost.

Tele: 0091-7708914377, 0091-422-2572177

E-mail addresses: kathiravan.sss@gmail.com, jkr@eee.psgtech.ac.in

Variational TV non-neighbourhood super-resolution

We consider that the immutable, and hence reliable, non-neighbourhood information is contained in the LR frames. Consequently, we do not proceed with a standard SR process followed by an NL-Means like enhancement step, but compute instead a non-neighbourhood error on the LR frames. Hence, our SR functional to be minimized is:

$$I_{HR} = \underset{I_{HR}}{\operatorname{argmin}} \sum_{i=0}^{k} \|I_{LR}^{k} - W_{k}^{NL} DBI_{HR}\|_2^2 + 2\lambda TV(I_{HR}). \quad (2)$$

Note that W_k^{NL} is not a geometric warping anymore, but a weighting matrix. Writing x_i^k the patch extracted at the location *i* in the image *k*, W_k^{NL} is defined by:

$$W_k^{NL}(i,j) = \frac{1}{K_i^k} \exp\left(-\frac{\|x_i^* - x_j^k\|^2}{h^2}\right), \quad (3)$$

where the superscript * is the index of the reference LR frame to be upscaled. The constant K_i^k ensures that each line of W_k^{NL} sums to 1. The parameter h is a similarity parameter: for small values of h, a candidate patch x_j needs to be very similar to the reference patch x_i^* to have a significant contribution. Following [4, 7], we solve Eq. (2) in two steps iteratively:

1. We first look for an HR blurred estimate $Z_{HR} = BI_{HR}$ using non-neighbourhood back-projection;
2. Then, we compute I_{HR} by TV deblurring of Z_{HR}.

The TV deblurring subproblem is solved using the Monotonous FISTA (MFISTA) algorithm described in [11]. Our specific form of the SR objective function in Eq. (2) calls for several comments:

- Since the weight matrices W_k^{NL} are fixed throughout the minimization process, it is possible to define the gradient of the non-neighbourhood error. Hence, we can use a procedure similar to the FISTA algorithm [11] to iteratively solve the two steps non-neighbourhood error minimization- TV minimization. This minimization process has the advantages of faster convergence and a tighter control of the solution;
- The normalization of each line of the matrix W_k^{NL} leads to a straightforward interpretation of non-neighbourhood SR as a standard SR process with a probabilistic motion estimation instead of the usual univocal motion model : each line W_k^{NL} (*i*, .) is indeed the motion probability density of the pixel *i* with respect to the *k*-th image;
- Finally, remark that it is possible to separate the constant from the similarity score in Eq. (3). The matrix W_k^{NL} can then be written as the product of a diagonal matrix (storing the constants 1/ K_i^k by a similarity matrix. This will be useful for the Phantom Hashing (Sec. 2.3).

Patch phantom hashing

The bottleneck of non-neighbourhood algorithms is the huge number of computations needed to compute the matrices W_k^{NL}. In the context of denoising, it is possible to pre-compute some values (integral images, pre-selection criterion) to reduce the final cost. These techniques however can hardly be used in our case, since we need to find the nearest neighbors to a given patch, and not infer a final value. This is a problem of nearest neighbors in high dimensions: typical patch sizes are 25 or 49 (5-by-5 or 7-by-7 patches). A naive approach like Principal Component Analysis (PCA) followed by sorting in a *kd*-tree does not help, because the performance of *kd*-trees quickly decreases when the dimension of the space is greater than 5, which is too small to correctly describe a patch. We considered instead hashing based techniques. Hash tables are highly efficient: the access to the bucket containing a given element is a simple array lookup, hence done in constant time. The main problem then becomes to design a sorting efficient and computationally fast code to distribute the patches over the table. Recently, Locality Sensitive hashing (LSH) [12] has become very popular for high dimensional data. However, it relies on the choice of several random projections. These projections need to be carefully chosen to ensure that they cover the whole space spanned by the data. This makes LSH unsuitable to sort image patches, since patches in a limited number of locations: many of the random projections become useless. On the other hand, Phantom Hashing (SH), introduced in [10], was specifically designed to produce compact codes grouping images or parts of images by similarity. Let us consider first scalar points uniformly distributed in the interval [*a*, *b*], and a similarity matrix *C* defined by:

$$C(i,j) = \exp\left(-\frac{(x_i - x_j)^2}{h^2}\right), \quad (4)$$

which is our patch similarity measure (excluding the normalization constant, as explained in the previous section). Under these hypotheses, one has a closed form solution for the eigenfunctions and eigenvalues of *C* [13]:

$$\Phi_k(\mathbf{x}) = \sin\left(\frac{\pi}{2} + \frac{k\pi}{b-a}x\right),$$

$$\lambda_k = 1 - \exp\left(\frac{h^2}{2}\left|\frac{k\pi}{b-a}\right|^2\right).$$

Then, retaining the *k* the smallest eigenvalues, SH simply consists in thresholding the *k* corresponding eigenfunctions to form binary codes by concatenation. In the case of vector data (such as image patches) of dimension *d*, one can see that

$$\exp\left(\frac{(\mathbf{x}_i - \mathbf{x}_j)^2}{h^2}\right) = \prod_{l=0}^{d-1} \exp\left(-\frac{(\mathbf{x}_i^*(l) - \mathbf{x}_j^k(l))^2}{h^2}\right)$$

Hence, one can simply apply the scalar result on each component and the take the product. While the assumption of uniform distribution seems to be restrictive, SH is surprisingly robust to its violations [10]. Furthermore, the PCA basis provides a coordinate system in which the data follows roughly this distribution along each axis. Finally, this gives us a simple algorithm to compute binary codes associated to a set of patches:

1. Find the principal components of the patches from the LR frames. They form the reference (or learning) data;
2. Compute the single-dimension eigenfunctions (Eq. (5));
3. Retain the *k* eigenfunctions corresponding to the smallest scalar eigenvalues given by Eq. (6).

The concatenation of the binary thresholded eigenfunctions gives the code associated to any patch x, which in turn corresponds to a bucket index in the hash table. At runtime, one simply needs to project the current patch onto the PCA basis (a matrix-vector multiplication), then threshold *k* obtained by Eq. (5) values which are also fast to compute. This returns all the patches that have a Hamming distance of 0 with the query patch, which is usually enough. If one wants to find more neighbors, then simple bit offsets produce the codes of the buckets with Hamming distance 1. Fig. 1 shows the outputs of several queries (with Hamming distance 0) using codes of 8 and 16 bits. Note how the nearest neighbours concentrate along geometrically and radiometrically similar image features. Codes of length 16 bits seem to be a good compromise between finding very similar patches while not being too specific. The computations are very fast: each patch is read only twice during the training, then the lookup procedure simply projects the query patch on the PCA basis and reads an index in a table.

Non-neighbourhood super-resolution algorithm

Putting Sec. 2.2 and 2.3 together, we obtain the SR algorithm:

Initialization:

1. Form all the patches from the LR sequence, and compute their PCA.
2. Compute the desired number of eigenfunctions of the similarity matrix.
3. Sort all the input patches in a table, using the retained eigen functions to obtain binary codes.

Super-resolution:

1. Select the frame to upsample and form an initial estimate by interpolation.
2. Apply the degradation model (blur and downsampling) to the current solution.
3. For each point in the downsampled solution, compute the non-neighbourhood error using the corresponding patch in the selected frame and its non-neighbourhood neighbors retrieved by SH.
4. Back-project the non-neighbourhood error and deblur the updated solution with MFISTA.

Note that, since the hash table contains patches from the whole input sequence, our algorithm is fully non-neighbourhood: we do not limit the search of relevant patches to a space-time search window. Furthermore, unlike bilateral regularization, there is no additional attenuation factor due to the space or time distance between the patch to update and its neighbours.

Experiments

Implementation details

The algorithm described in Sec. 2.4 is implemented on a standard laptop using C++. Although a Matlab implementation of Phantom Hashing is available online1, we did re-implement it using the OpenCV library. In the nearest neighbour search, we did not use all the patches returned by a query, but only those with a similarity score above 0.9.

The parameter *h* is fixed depending on the noise level of the input sequence. For clean movies without noise, we used a small value of 0.08, and values between 0.5 and 1 for noisy inputs. In all our experiments, we have chosen patches of size 5-by-5 pixels.

The initialization of the algorithm is very fast: computing the PCA of a whole LR sequence is the longest part and takes only a few seconds on a laptop for the patch size considered. Then, we used a naive implementation of the algorithm described in Sec. 2.4. Note however that the operations for each pixel are independent: this parallelism can be exploited to design faster multithreaded implementations.

Fig. 1. Query examples by Patch Phantom Hashing.

The green square is the query. Red squares are the patches of the same bucket (Hamming distance 0) in the hash table, whose similarity score is greater than 0.5. The length of the codes is 8 bits in the top row, 16 bits in the bottom row

Super-resolution results

We have tested the proposed algorithm on several sequences and reference data available on the internet23. Fig. 2 shows the interest of our functional: having fixed weights on the LR input avoids the apparition of the high frequency bright artifacts around the mouth. On the other hand, since we have numerous non-neighbourhood neighbors from the whole sequence, this is counter-balanced by a more pronounced visual blur with respect to [7].

Fig. 2. Comparison with related work

Left : original image. Middle: result of Generalized Non-neighbourhood Means [7] for a zoom factor of 3 along each dimension. Right : our result. Our minimization scheme avoided the artifacts around the mouth. While we used a fully non-neighbourhood approach, there are very few artifacts coming for learning the defects of other frames in the sequence.

Fig. 3 illustrates the application of our algorithm to other noise free sequences. Finally, since accurate and near real time optical flow algorithms are becoming available [5], one may wonder why we should still perseverate with non-neighbourhood SR. If there is only one reason, it should be to deal with noisy sequences. In the presence of noise, the precision of optical flow diminishes quickly. Non-neighbourhood algorithms, on the other hand, are able to integrate information all along the sequence without explicitly needing any motion estimation, and are designed to naturally deal with independent random noise. This is demonstrated in Fig. 4, using either wider patches or relaxed similarity conditions.

Fig. 3. Results in the noise free case

In both experiments, we used a zoom factor of 3 along each dimension and patches of size 5-by-5 pixels.

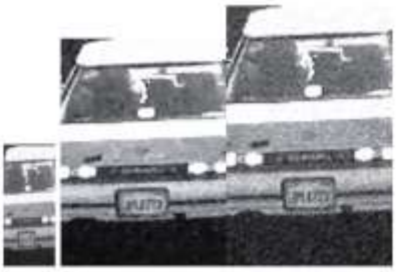

Fig. 4. Noisy example

The input sequence was corrupted with a Gaussian noise of variance 0.05 (the images were rescaled between 0 and 1). In the first case (middle image), we took wider patches (of size 7-by-7 pixels) to make the patch comparison process more robust, which also removed some details with the noise. In the second experiment (right image), we considered a lower threshold to declare a reference patch meaningful, preserving more detail.

Conclusion and future work

In this work, we have demonstrated the interest of phantom hashing for non-neighbourhood methods, and applied it to the super-resolution problem. Phantom hashing allows the development of fully non-neighbourhood algorithms; hence exploiting all the information of the input sequence, while maintaining a low computational complexity. We also presented a new non local SR functional, where the non-neighbourhood weights are fixed. Hence, we can then compute the gradient of the functional and insert it inside algorithms where the optimization path is controlled, preventing from the apparition of spurious artifacts. Future work will include two main directions: first, the adaptation of the proposed algorithm to the single image SR task, when no input sequence is available.

Second, we also plan to apply phantom hashing to previously proposed non-neighbourhood algorithms, starting with inpainting [14], to study the effects of full non-neighbourhoodity on these specific problems: as can be seen from our results, there is clearly a trade-off to be found between the amount of non-neighbourhoodity and the over-regularization of the result.

References

[1] M. Irani and S. Peleg, "Improving resolution by image registration,"CVGIP: Graphical models and image processing, vol.53, no. 3, pp. 231–239, 1991.

[2] D. Capel and A. Zisserman, "Computer vision applied to super-resolution," IEEE Signal Processing Magazine, vol.20, no. 3, pp. 75– 86, 2003.

[3] S.C. Park, M.K. Park, and M.G. Kang, "super-resolution image reconstruction: a technical overview," IEEE Signal Processing Magazine, vol. 20, no. 3, pp. 21– 36, 2003.

[4] S. Farsiu, M.D. Robinson, M. Elad, and P. Milanfar, "Fast and robust multiframe super resolution," IEEE Transactions on Image Processing, vol. 13, no. 10, pp. 1327 – 1344, 2004.

[5] D. Mitzel, T. Pock, T. Schoenemann, and D. Cremers, "Video super resolution using duality based TV-L1 optical flow," Pattern Recognition, pp. 432–441, 2010.

[6] W.T. Freeman, T.R. Jones, and E.C. Pasztor, "Example-based super-resolution," Computer Graphics and Applications, vol. 22, no. 2, pp. 56–65, 2002.

[7] M. Protter, M. Elad, H. Takeda, and P. Milanfar, "Generalizing the nonlocal-means to super-resolution reconstruction," IEEE Transactions on Image Processing, vol. 18, no. 1, pp. 36 – 51, 2009.

[8] A. Buades, B. Coll, and J.-M. Morel, "Nonlocal image and movie denoising," International journal of computer vision, Jan 2008.

[9] W. Dong, L. Zhang, G. Shi, and X. Wu;, "Nonlocal backprojection for adaptive image enlargement," 16th IEEE International Conference on Image Processing (ICIP), pp. 349 – 352, 2009.

[10] Y. Weiss, A. Torralba, and R. Fergus, "Phantom hashing," Advances in neural information processing systems, vol. 21, pp. 1753–1760, 2009.

[11] A. Beck and M. Teboulle, "Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems," IEEE Transactions on Image Processing, vol. 18, no. 11, pp. 2419 – 2434, 2009.

[12] A. Andoni and P. Indyk, "Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions," 47th Annual IEEE Symposium Foundations of Computer Science, 2006. FOCS '06., pp. 459–468, 2006.

[13] B. Nadler, S. Lafon, R.R. Coifman, and I.G. Kevrekidis, "Diffusion maps, phantom clustering and reaction coordinates of dynamical systems," arXiv, 2005.

[14] E. D'Angelo and P. Vandergheynst, "Towards unifying diffusion and exemplar-based inpainting," in 17th IEEE International Conference on Image Processing (ICIP), 2010.