

Nonuniform Image Patch Exemplars for Low Level Vision

Vincent De Smet¹

¹ESAT-PSI/iMinds, KU Leuven
Belgium

Vinay P. Namboodiri²

²Alcatel-Lucent Bell Labs
Antwerp, Belgium

Luc Van Gool^{1,3}

³BIWI, ETH Zurich
Switzerland

Abstract

In this paper we propose the use of nonuniformly resized image patch exemplars for solving low level vision problems like denoising and super-resolution. While patch-based methods have been shown to be successful for several such applications, these methods have so far assumed uniform sizes for image patches. In this paper we address this restriction. We use an integral image representation for efficient computation of the matching cost for variable-sized patches. We show that nonuniform image patch exemplars are useful in improving classic techniques for non-local means-based denoising and example-based super-resolution. We provide refinement cues to further improve the patch size estimation. This method can be adopted for a large number of related methods and applications due to its simplicity and generality.

1. Introduction

Image patch exemplar-based methods have been quite successful in a variety of applications such as denoising [4, 3], super-resolution [5, 14, 17, 6, 11], and texture synthesis [7]. In image denoising, patch-based methods have been used through the nonlocal means approach [4] by exploiting the non-stationary distribution property of noise. Nonlocal patches are combined by a kernel on their matching distance. On the other hand, in super-resolution the aim is to hallucinate high frequency information that is not present in the low-resolution observation. Patch-based methods have been used in a ‘learning’-based approach [5], through a database of corresponding high-resolution and low-resolution image patches. While in denoising the aim is to remove the high-frequency noise, in super-resolution, the aim is to synthesize high-frequency detail. In this paper, we focus on these two applications as they are characteristic of the problems in low level vision.

While patch-based methods have been quite successful in low level vision problems, it was always assumed that the exemplar patches are of a fixed square size. In general, this assumption is not well-founded. In this paper, we propose

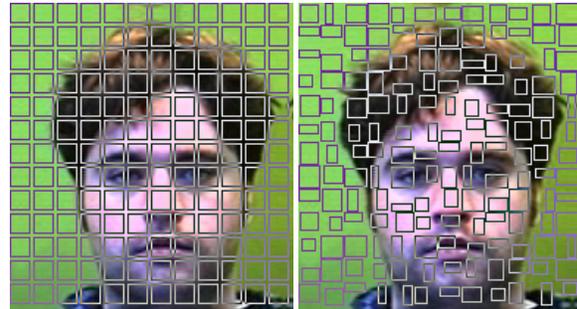


Figure 1. Need for nonuniform image patch exemplars: (a) shows a subset of fixed sized exemplars and (b) shows a subset of varied size exemplars that best match the dataset images.

a more general definition of image patch exemplars that allows for different sizes and aspect ratios for all patches individually. Fig. 1 illustrates the basic concept (for illustrative purposes we do not show all patches here). We determine the patch size and shape based on the matching cost of the patch with the dataset image. An interesting observation is that for each patch in the image, the optimal size that is closest in the database is different. For instance, in the lip region one tends to obtain ‘flattish’ rectangular patches, while in the nose region, matching along the ridge of the nose is more suitable.

The straight-forward approach to obtain nonuniform exemplars is an exhaustive search in which one would match all patches of all sizes and shapes in an image to all patches of all possible sizes and shapes in the dataset. The computational complexity this implies is clearly too high for such strategy to be of any practical use (computational complexity is compared in sec. 3). In order to allow for the variation of patch sizes while still retaining computational efficiency, we adopt a 2-stage approach. First, we perform a nearest neighbor search with a fixed size to identify interesting candidate regions from which to copy an exemplar patch. We then use an efficient technique based on integral images [12] to obtain the variable size exemplar.

As explained further in sec. 3, by restricting the origin of the nonuniformly-sized patches to a reduced set of locations based on local similarity, and by finding the optimal

size in an efficient manner, the computational cost is only marginally increased. The benefit is that the accuracy is improved, as each image patch size is allowed to best adapt to the underlying scene information. In this paper, we apply this method for denoising and super-resolution applications. We demonstrate the benefits on two datasets, the Berkeley image dataset [9] and a license plate dataset that we have collected.

Patch-based techniques have been extended in various ways. For instance, recently the patch-based method for denoising [4] has been extended by Mairal *et al.* [8] where the authors use sparsity and by Brox *et al.* [3] who present an efficient patch retrieval method. The patch-based method for super-resolution [5] has been extended by Wang *et al.* [13], by incorporating mutual co-occurrence information. Glasner *et al.* [6] have shown that the recurrence of patches within the same image across the same and different scales can be used for super-resolution. In recent work by Sun *et al.* [11] the authors proposed a context-constrained hallucination approach for super-resolution. Also recently, Barnes *et al.* [1] proposed an efficient approximate nearest neighbor search using randomized patch matching strategies. This work has been extended by the authors to a generalized patch matching technique [2], where they propagate the matches in a neighborhood to similar neighborhoods.

As we nonuniformly resize each patch, our work is complementary to the above research. Indeed, we can incorporate the nonuniform patch exemplars into these works. To summarize, the main contributions of the paper are to highlight the need for nonuniform image patch exemplars, to propose a method to obtain the exemplars in an efficient way and to demonstrate that the use of nonuniform exemplars improves results in various low level vision tasks.

2. Nonuniform Exemplars

In this section we describe our proposed method to obtain nonuniform exemplars. In sec. 2.1 we create initial exemplars from fixed-sized patches, whose size and aspect ratio we then optimize in sec. 2.2. Sec. 2.3 then provides additional cues to guide the optimization.

2.1. Finding Initial Exemplars

We start from an input image and a database of images from which to draw exemplars. Computing the initial fixed size exemplars involves finding for each input exemplar its nearest neighbors in a space with a number of dimensions depending on the size of the exemplar, which in our experiments is 49 for 7×7 pixel exemplars. We follow the suggestion of Wang *et al.* [13] to find nearest neighbors using an adaptive locality sensitive hashing (ALSH) algorithm, which we observed to give similar results as exact nearest neighbor search methods such as vp-trees [16] while being computationally more efficient. Our proposed approach

depends on the fixed exemplar search only for computing an approximate set of candidate patches from the database. With ALSH we obtain an appropriate candidate set efficiently.

2.2. Efficient Patch Resizing

The initial exemplar search from the previous step is used as a registration of image regions between the input image and the database image containing the candidate patch, such that the candidate patch overlaps with the input patch. This means that for each patch location in the input image we now have specific locations in a set of candidate images where we have an initial matching neighborhood. The process we then use for finding nonuniform patches is illustrated in fig. 2. Let us suppose that in the previous step source patch p_1 in image I_1 (the input image, on top in fig. 2) was matched to candidate patch p_2 in image I_2 (a candidate image from the database, bottom of fig. 2). We then define equally sized image regions I_{r1} and I_{r2} around both patches that serve as the neighborhood we will use for size and shape optimization. These regions are the maximal size to which we allow a patch to grow (in our experiments we use an extension of 3 pixels around p_1 and p_2). Next we compute the squared difference per pixel between I_{r1} and I_{r2} and obtain the matching cost map M_{12} (other metrics can also be incorporated). We then compute the integral image X over M_{12} . For coordinates (i, j) this is given by

$$X(i, j) = \sum_{i' \leq i, j' \leq j} M_{12}(i', j'). \quad (1)$$

The integral image can be calculated in one pass using the equations proposed by Viola and Jones [12]:

$$s(i, j) = s(i, j - 1) + M_{12}(i, j), \quad (2)$$

$$X(i, j) = X(i - 1, j) + s(i, j), \quad (3)$$

with $s(i, j)$ an intermediary variable.

Once we obtain this integral image X of the matching cost map, we compute the rectangular patch minimizing the average per-pixel cost. Note that we avoid trivial solutions of one pixel matches by always enforcing a minimal region of 3×3 pixels around the center of X . For a rectangular patch with coordinates from i_b to i_e (beginning and ending row) and j_b to j_e (beginning and ending column) this cost is given by

$$C = \frac{X(i_e, j_e) - X(i_{b-1}, j_e) - X(i_e, j_{b-1}) + X(i_{b-1}, j_{b-1})}{(i_e - i_{b-1})(j_e - j_{b-1})}. \quad (4)$$

The arguments which minimize C describe the optimal shape for this patch pair. In sec. 2.1 we collected a set of initial candidate patches for each initial input patch. We can

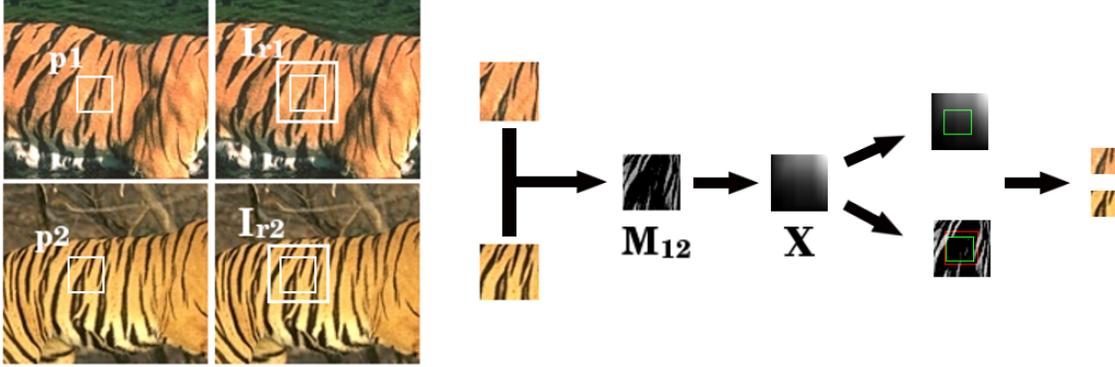


Figure 2. Illustration of image patch resizing steps. The red bounding box represents the fixed patch shape. The green bounding box shows the optimal nonuniform shape found by our algorithm.

now apply the size optimization procedure described here to each of those input-candidate exemplar pairs, resulting in a specific size and aspect ratio for each pair. The candidate with the lowest average per-pixel cost C is then selected to be used in the output image.

2.3. Size Optimization Cues

In the previous section we discussed how to obtain the best matching size and aspect ratio of input-candidate patch exemplar pairs. Let us now consider the overall quality of the matching pairs. When none of the initial exemplars available in the database are close to the input image patch, it may serve the application better to minimize the size of the patch, even though the optimal matching size for this patch pair may be large. This helps to minimize the influence of bad exemplars. Similarly, if we find an exemplar that matches very well, it would be better to maximize the size of the patch.

In this section we explore some optional refinements to the method of the previous section. We propose using the initial patch matching distance as a guideline to decide whether to grow or to shrink a patch. In order to use these cues for growing and shrinking we need certain global thresholds based on the initial patch distances. Depending on what the initial matching distance of a specific patch pair is relative to these thresholds, we either allow only growing, only shrinking or both (i.e. when it lies between both thresholds).

To obtain these thresholds we use a data dependent method. In both super-resolution and denoising, while computing the output Signal-to-Noise Ratio (SNR) for various values of the growing and shrinking thresholds, we observe that there is a correlation between a) intervals for the thresholds over which the growing and shrinking cues performed well and b) the distribution of initial patch matching distances for the image under consideration as can be seen in the patch distance histogram. As an example, let us consider the case where we super-resolve a random example image from the Berkeley dataset [9]. We show the output SNR for

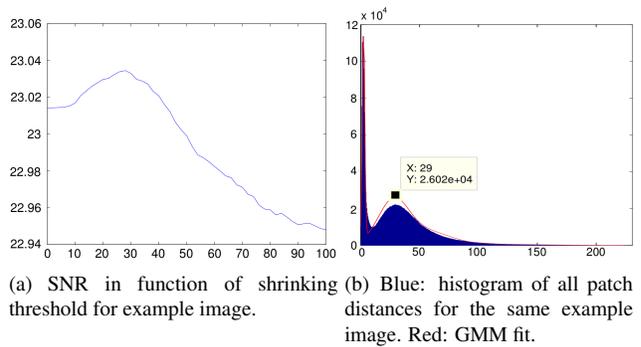


Figure 3. Correlation of SNR and GMM of patch distances for a super-resolution example. Note how the mean of the second Gaussian in fig. 3.(b) correlates with the top SNR in figure 3.(a).

different values of the shrinking threshold in fig. 3.(a). In this figure the ideal shrinking threshold (the patch matching distance below which we will only allow the patch to shrink rather than grow) corresponds to the location of the maximum of the SNR curve. We will call this threshold τ_s , and for this specific example image it lies somewhere near 28. The histogram of all patch distances over this entire example image can be seen in fig. 3.(b). In this figure we observe that the second peak in this distribution corresponds roughly to the same τ_s we found as the location of the maximum SNR in fig. 3.(a), or the ideal value for our shrinking threshold for this image. This is a trend which we observe in most example images, meaning we can find the ideal shrinking threshold by observing the patch distance histogram. The ideal growing threshold can be found in a similar manner. We fit a Gaussian mixture model (GMM) to the distance distribution, from which we obtain for each input image specific growing and shrinking thresholds $\hat{\tau}_g$ and $\hat{\tau}_s$ resp. These thresholds, based on the GMM distribution, are data dependent thresholds and are relevant for the input image. We specify which thresholds we use for super-resolution and denoising respectively in sec. 5.

3. Complexity Analysis

Our method presents an efficient way of creating nonuniform patch exemplars. In this section we give a complexity analysis of our algorithm, and compare it to other possible methods for finding nonuniform exemplars. The computing time needed for the applications of super-resolution or denoising (using a fixed approach or our nonuniform approach) is dominated by the nearest neighbor search. For reference, our implementation of super-resolution on a Berkeley image requires about 650 seconds on a 2.5 GHz Intel Core 2 Quad Q9300 machine, with only about 15 percent of the time spent on size optimization.

3.1. Our method

The main variables we need in order to analyse the complexity of our method are the amount of input patches P_{in} , the amount of database patches P_{db} and the amount of candidates k . Furthermore, we need the fixed patch length S_f (which is 7 in the case of a 7×7 patch size) and the maximal patch length for nonuniform patches S_{nu} (e.g. 7 + 6 when we have an extension of 3 pixels around the fixed size). The amount of calculations needed for our method, A_1 , can be found to be

$$A_1 = P_{in}P_{db}S_f^2N_1 + P_{in}k(S_{nu})^2(N_1 + N_2) + P_{in}k\left(\left(\frac{S_{nu}}{2}\right)^2\right)^2N_3, \quad (5)$$

The first term is the amount of calculations needed to find the initial fixed exemplars if we do not use a structuring or hashing algorithm to speed up the nearest neighbor search. This can be brought down substantially by using e.g. kd-trees, vp-trees or ALSH. N_1 is the amount of calculations needed per pixel to calculate the Euclidean distance between two exemplars, which is 1 subtraction, 1 square operation and 1 addition.

The second term describes the creation of the matching cost map M_{12} (which again comes down to a Euclidean distance and thus also requires N_1 calculations per pixel) and the integral image X (which takes N_2 calculations per pixel). N_2 is found from eqns. 2 and 3 to be 2 additions.

The third term covers the calculation of cost function C , as described in eqn. 4. This is calculated for each possible upper-right corner and lower-left corner pixel, which together define the patch shape. These two corners can lie anywhere in the $\left(\frac{S_{nu}}{2}\right)^2$ pixels of their two respective quadrants around the center, which results in $\left(\left(\frac{S_{nu}}{2}\right)^2\right)^2$ calculations of C . The amount of operations needed to calculate C is N_3 , which according to eqn. 4 is 5 additions/subtractions and 1 division. We have ignored the fact that we enforce a minimum patch size of 3×3 pixels in eqn. 5, for the sake of clarity. If this constraint is included the complexity becomes slightly smaller.

3.2. Other methods

When we try to calculate the cost C without the use of the integral image representation the algorithm becomes more complex. We define one new variable here: the total amount of pixels covered by all the rectangles we consider for our nonuniform patch size. We refer to this variable as Z . For each of these pixels a squared difference is calculated. The total amount of calculations needed is then

$$A_2 = P_{in}P_{db}S_f^2N_1 + P_{in}kZN_1. \quad (6)$$

To find Z we treat the upper-right and lower-left corners identically to each other. Both have coordinates with an origin set to the center, and the coordinates of the upper-right corner (i_1, j_1) increase in the upper and right direction while the coordinates of the lower-left corner (i_2, j_2) increase in the lower and left direction. Z can then be found as

$$Z = \sum_{i_1=1}^{\frac{S_{nu}}{2}} \sum_{j_1=1}^{\frac{S_{nu}}{2}} \sum_{i_2=1}^{\frac{S_{nu}}{2}} \sum_{j_2=1}^{\frac{S_{nu}}{2}} (i_1 + i_2)(j_1 + j_2) \quad (7)$$

$$= \frac{1}{64}(S_{nu}^6 + 4S_{nu}^5 + 4S_{nu}^4). \quad (8)$$

The first term of A_2 is the same as in A_1 . Outside of that term this method is a function of S_{nu}^6 , whereas the method we propose is a function of S_{nu}^4 , proving that our method has a much higher efficiency for calculating nonuniform exemplars.

The last method we consider here is the case where we would optimize patch sizes during the initial nearest neighbor search. The amount of calculations is similar to eqn. 6, except for two things: the size optimization term has to be calculated for all database patches, rather than for k candidates, and structuring the database or using a hashing method to reduce the complexity of the nearest neighbor search is no longer possible. The amount of calculations in this case is

$$A_3 = P_{in}P_{db}ZN_1, \quad (9)$$

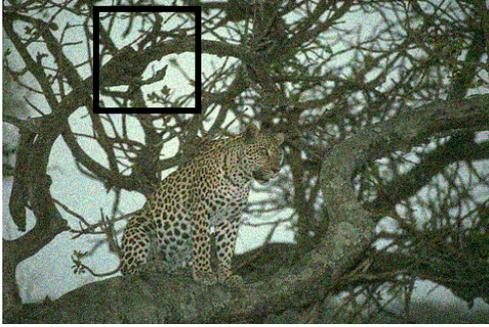
which is too high to be of any practical use.

4. Applications

4.1. Nonlocal means denoising

In the nonlocal means-based denoising algorithm [4], the authors propose a method for denoising that relies on the weighted averaging of the pixels in the noisy input image. The method uses patch-based weights for nonlocal averaging at each pixel. Given a noisy image V , the denoised image $N(i)$ obtained at a pixel location i is

$$N(i) = \sum_{j \in V} w(i, j)V(j). \quad (10)$$



(a) Noisy input image.



(b) Noisy input image. (c) Denoised with fixed size exemplars of size 7×7 . SNR = 17.9158. (d) Denoised with variable size exemplars. SNR = 23.5478. SNR = 23.9990.

Figure 4. Cropped denoising results for an example Berkeley image.

The matching of patches is usually restricted to a region R around the pixel i . In this equation, the weights $w(i, j)$ are obtained based on a weighted patch-based distance match. For two patches at locations i and j respectively, the weight $w(i, j)$ is obtained over a patch of size O as follows:

$$w(i, j) = \frac{1}{Z(i)} \exp \frac{-C_{match}}{2\tau^2}, \quad (11)$$

where the matching cost C_{match} is $\sum_{q \in O} (M(i - q) - M(j - q))^2$ in the original algorithm. In this equation, q iterates over the pixels in the two patches centered at pixel i and pixel j . The pixels j with a similar neighborhood to pixel i have larger weights. $Z(i)$ is a normalization constant such that $\sum_j w(i, j) = 1$. The parameter τ acts as a degree of filtering that controls the decay of the weights.

An approximation of eqn. 10 is to use a set of k nearest patch exemplars. The exponential decay of the filtering parameter has as a consequence that beyond the k nearest neighboring patch candidates the averaging has no significant effect. We obtain k nearest patch candidates (in our case $k = 150$) using fixed- or variable-sized exemplars from a database of all the patches in the noisy input image.

For variable-sized exemplars we use the minimum average cost per pixel C obtained from eqn. 4 as the matching cost C_{match} . The computation of the normalization constant $Z(i)$ again ensures that the weights sum to one.

4.2. Learning-based Super-resolution

We demonstrate the effect of patch exemplars for super-resolution by considering a modified method for learning-based super-resolution [5]. In the original framework, the image is super-resolved by representing the problem in a Markov random field (MRF) framework. The low-resolution input forms the observed variable and a database of low-resolution (LR) and their corresponding high-resolution (HR) image patches is used to super-resolve the image. The data compatibility function for this MRF is obtained by computing k -nearest neighbors and calculating

their distance to the input patch. The compatibility function between neighboring HR patches is obtained by evaluating the overlap between the HR candidates. This MRF is then optimized by minimizing the weighted sum of these two cost functions. This is most commonly done with belief propagation techniques [15] or graph cuts [10].

We use a version of this algorithm which gives improved results to evaluate the effect of the exemplar patch sizes. In our implementation of the baseline method we use fixed patch sizes of 7×7 . To find the k nearest neighbors we use the efficient ALSH method [13]. Instead of using the patch-based compatibility criterion in belief propagation, we use denser patch sampling where each pixel is taken as a center pixel for a patch. The corresponding HR patch for the nearest neighbor is directly used in the output intensity image and the contributions from overlapping patches are averaged. This modification directly considers the effect of the patch hallucination without correction from the belief propagation and therefore enables us to better evaluate the effect of variable size exemplars.

5. Evaluation

We evaluate our method for the two applications described in the previous section: denoising using nonlocal means and patch-based super-resolution. Both are performed on the Berkeley image dataset [9]. We compare the results obtained from using fixed patch sizes and the results from using optimized patch sizes with and without bounds as described in sec. 2.3. We show quantitative results over all images of the dataset and we also show a few visual results. We direct the reader to our website¹, where we show visual results for every image of the dataset. Since both image denoising and super-resolution have interesting applications in the areas of video surveillance and forensic image processing, we also tested the method on a more specific class of images, particularly license plates.

¹<http://homes.esat.kuleuven.be/~vdesmet/nonuniform/>



(a) Noisy input image.



(b) Noisy input image, SNR = 18.9317.



(c) Denoised with fixed size exemplars of size 7×7 , SNR = 22.8931.



(d) Denoised with variable size exemplars, SNR = 23.8168.

Figure 5. Cropped denoising results for example grayscale Berkeley image.

Method	Mean SNR	Std. Dev. SNR	Mean SNR Gain
Noisy image	18.1125	2.0620	0
Fixed patches	22.0625	2.3699	3.9501
Nonuniform: No constraints	22.6176	2.2849	4.5052
Nonuniform: Data dependent	22.6579	2.3414	4.5455

Table 1. Signal-to-noise ratio for nonlocal means performed on Berkeley dataset.

5.1. Denoising Experiments

To create noisy input images we add Gaussian white noise with a standard deviation of 0.06^2 to the Berkeley dataset. We employ the method described at the end of sec. 4.1 to find the optimal patch size for denoising. The database is made up of patches from the full input image. Table 1 shows a comparison of our method to the baseline nonlocal means method [4] for the images of the Berkeley dataset. The baseline method gives an improvement of 3.9501 dB on average over the noisy images. In the rest of the table we compare the proposed two different methods to find optimal patch sizes. The first is the standard patch resizing algorithm described in sec. 2.2, which shows a mean improvement over all images of 0.555 dB (with a standard deviation of 0.3609) over the baseline method.

In order to find good bounds for each image, we use the patch matching distances obtained from the nearest-neighbor search and fit a Gaussian mixture model to this data. From these we obtained the parameter $\hat{\tau}_s$ as the mean of the first Gaussian plus half its standard deviation. The growing threshold did not improve the denoising significantly and so we used $\hat{\tau}_g = 0$. This data dependent method gives a mean improvement of 0.5954 dB over the baseline

²This is assuming the pixel values lie between 0 and 1. When working with values between 0 and 255 this corresponds to a standard deviation of 15.3. Comparable levels of noise are often used in denoising research, e.g. [4, 3].

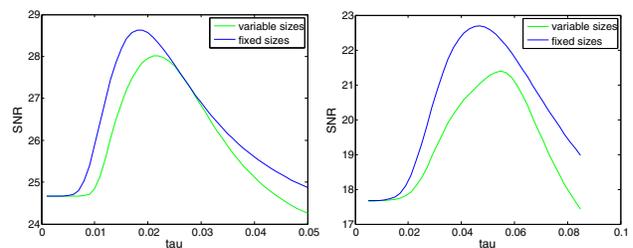


Figure 6. SNR value comparison for different values of τ . Left: Lena, right: license plate.

method. Two example images from the Berkeley dataset and their denoising results are shown in fig. 4 and 5 to demonstrate color and grayscale denoising respectively.

The graph on the left in fig. 6 compares the SNR of the results for the standard ‘Lena’ image, over a range of values for the nonlocal means parameter τ . The blue graph shows results for the nonuniform exemplar method, while the green graph shows results for fixed sizes. The proposed method yields a higher SNR over most of the range. Fig. 7 explores different levels of input noise. The denoising parameter τ is set to 0.5 times the standard deviation of the Gaussian noise. Again the nonuniform exemplar method results in higher output SNR.

5.2. Super-resolution Experiments

We evaluate the results of our method for patch-based super-resolution in a manner similar to the previous section. The magnification factor used for these experiments is 2x. The patch database consists of the low resolution input image itself. In this section we also compare to the recent super-resolution algorithm proposed by Zeyde *et al.* [17] (based on a method by Yang *et al.* [14]), which extends the patch-based method with a sparsity constraint. This method learns an over-complete dictionary from a set of natural images (as provided by the authors) and represents each output patch using a sparse combination of dictionary words. The results of these experiments are summarized in table

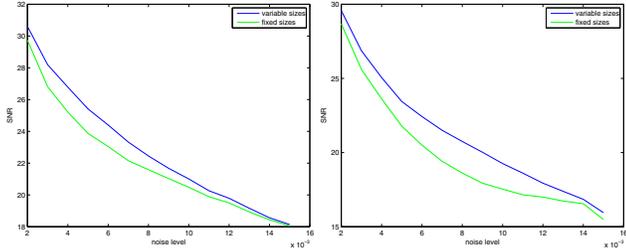


Figure 7. SNR value comparison for different levels of input noise. Left: Lena, right: license plate.

2 and two visual examples are shown in fig. 8. The baseline fixed size super-resolution method [5] achieves an improvement of 0.863 dB over bicubic interpolation. Patch size optimization without resizing thresholds results in a mean improvement of 1.0113 dB over bicubic interpolation, or 0.1483 dB over the fixed size experiments. The standard deviation on this improvement over fixed sizes is 0.0799. Using data-dependent thresholds gives slightly better results, with a mean improvement of 1.0248 dB over bicubic interpolation. As thresholds for super-resolution in these experiments we use the mean of the second Gaussian as $\hat{\tau}_s$ and again zero as $\hat{\tau}_g$. Zeyde’s sparse method performs somewhere between fixed patch sizes and nonuniform patch sizes. It produces a mean SNR improvement over bicubic interpolation of 0.9195 dB.

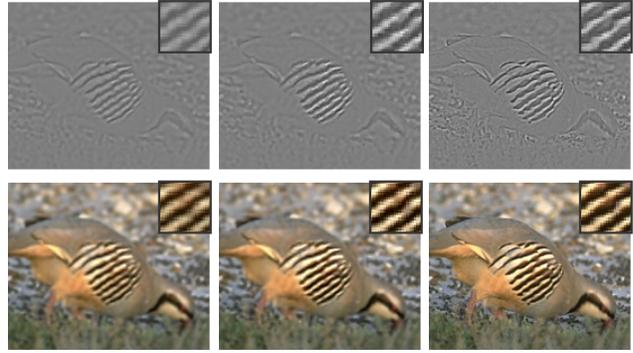
5.3. Evaluation on License Plates

In addition to comprehensive evaluation of the nonuniform exemplar technique on the standard Berkeley Segmentation dataset, we consider the use of nonuniform exemplars for denoising and super-resolving license plate images.

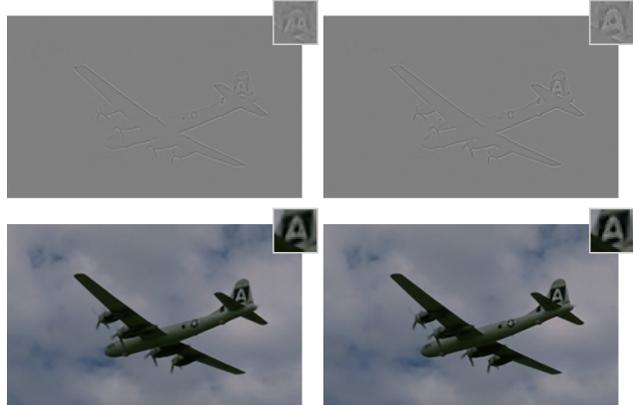
In fig. 10 we show denoising results for a noisy input license plate using fixed patch sizes and nonuniform patch sizes with no constraints. The nonuniform method gives better denoising results overall, and most noticeably in the small homogeneous regions near the letters. The graph on the right of fig. 6 shows an evaluation of the SNR result for denoising this license plate image based on different values of the denoising parameter τ , similar to the graph for Lena on the left. This graph again shows an improve-

Method	Mean SNR	Std. Dev. SNR	Mean SNR Gain
Bicubic	20.9693	3.6845	0
Fixed patches	21.8323	3.9642	0.8630
Nonuniform: No constraints	21.9806	3.9868	1.0113
Nonuniform: Data dependent	21.9941	3.9776	1.0248
Sparse SR [17]	21.8888	3.7842	0.9195

Table 2. Signal-to-noise ratio for super-resolution performed on Berkeley dataset.



(a) Fixed exemplars, (b) Our method, (c) Ground truth. SNR = 26.23. SNR = 26.73.



(d) Fixed exemplars, SNR = 33.88. (e) Our method, SNR = 34.15.

Figure 8. Super-resolution results for two Berkeley example images with close-up. Top row: Hallucinated high-band layer, Bottom row: Super-resolved image.



Figure 9. Some examples of license plates used for testing.



Figure 10. Denoising results for an example of a license plate. Left: noisy input, middle: denoised with fixed patch sizes, right: denoised with variable patch sizes.

ment of nonuniform patch sizes over fixed patch sizes independent of τ . In fig. 7 we again compare different noise levels, showing better performance by nonuniform patches over the entire range.

To test super-resolution for license plate images we collected a database of 450 license plate images similar to the examples shown in fig. 9. We show an example of the result from fixed and variable patch sizes in fig. 11. As can be seen in the result, using variable patch sizes improves the legibility of the license plate (especially the character ‘X’ in this example). Quantitative results on this database, computed in a leave-one-out manner, are shown in the next section.

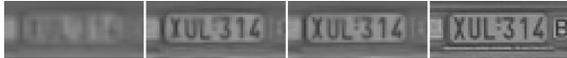


Figure 11. Super-resolution results for a license plate example. Left to right: bicubic interpolation, SR with fixed patch sizes, SR with variable patch sizes, ground truth.

5.4. Comparison of different fixed sizes

Patch sizes of 7×7 are widely used for low level vision tasks like super-resolution and nonlocal means denoising [4, 5]. Other fixed sizes are of course possible, so one might consider whether the improvements presented in this paper could also be achieved by choosing a different fixed size. Table 3 compares a range of different fixed patch sizes with our nonuniform exemplar method. Results are shown for super-resolution and nonlocal means denoising on both the Berkeley dataset and our license plate dataset. Our proposed method shows the best SNR for both applications on both datasets. When considering only fixed sizes, the 7×7 patch gives the best result in all cases except the specific case of nonlocal means denoising on license plates. In that specific case, using 3×3 patches gives better results than using 7×7 inside the license plates, because of a gain in similar patch candidates. However, this turns into a disadvantage around the license plate as the small patches are overfitted to the noise, resulting in weak denoising.

6. Conclusion

In this paper we have explored the use of nonuniform image patch exemplars. We have shown that allowing for variably sized exemplars is feasible and useful. The method has been demonstrated to work on two practical applications, namely nonlocal means denoising and example-based super-resolution. Refinement cues were introduced to guide the patch size optimization. We have evaluated the proposed method on the Berkeley dataset and on a specific class dataset consisting of license plates. From the evaluation of the method one can conclude that the use of variably sized exemplars merits consideration in various patch exemplar-based methods.

It is worthwhile to investigate how to further improve on

Size	Berkeley dataset		License plate dataset	
	SR	NLM	SR	NLM
3×3	21.16	20.55	16.52	24.42
5×5	21.22	21.96	17.75	23.79
7×7	21.83	22.06	17.96	23.20
9×9	20.75	21.76	17.88	22.80
Nonuniform	21.99	22.66	18.11	24.54

Table 3. Comparison of mean signal-to-noise ratio for different fixed sizes and nonuniform exemplars for SR and NLM performed on the Berkeley dataset and our license plate dataset.

the efficiency and accuracy of the proposed method. In the future we intend to explore these techniques for applications like texture-synthesis, structural image editing and others.

Acknowledgements. This study was supported by the Flemish government through the iMinds eHealth project.

References

- [1] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman. PatchMatch: A randomized correspondence algorithm for structural image editing. *SIGGRAPH*, 2009. 2
- [2] C. Barnes, E. Shechtman, D. B. Goldman, and A. Finkelstein. The generalized patchmatch correspondence algorithm. In *ECCV (3)*, pages 29–43, 2010. 2
- [3] T. Brox, O. Kleinschmidt, and D. Cremers. Efficient nonlocal means for denoising of textural patterns. *Transactions on Image Processing*, 17(7):1083–1092, 2008. 1, 2, 6
- [4] A. Buades, B. Coll, and J.-M. Morel. A non-local algorithm for image denoising. In *CVPR (2)*, pages 60–65, 2005. 1, 2, 4, 6, 8
- [5] W. T. Freeman, E. C. Pasztor, and O. T. Carmichael. Learning low-level vision. *International Journal of Computer Vision*, 40(1):25–47, 2000. 1, 2, 5, 7, 8
- [6] D. Glasner, S. Bagon, and M. Irani. Super-resolution from a single image. In *ICCV*, 2009. 1, 2
- [7] V. Kwatra, A. Schodl, I. Essa, G. Turk, and A. Bobick. Graphcut textures: Image and video synthesis using graph cuts. *SIGGRAPH*, 22(3):277–286, July 2003. 1
- [8] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Non-local sparse models for image restoration. In *ICCV*, pages 2272–2279, Oct. 2009. 2
- [9] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *ICCV*, pages 416–423, July 2001. 2, 3, 5
- [10] C. Rother, V. Kolmogorov, V. Lempitsky, and M. Szummer. Optimizing binary mrfs via extended roof duality. In *CVPR*, pages 1–8, 2007. 5
- [11] J. Sun, J. Zhu, and M. F. Tappen. Context-constrained hallucination for image super-resolution. In *CVPR*, pages 231–238, 2010. 1, 2
- [12] P. A. Viola and M. J. Jones. Robust real-time face detection. *IJCV*, 57(2):137–154, 2004. 1, 2
- [13] Q. Wang, X. Tang, and H. Shum. Patch based blind image super resolution. In *ICCV*, pages 709–716, 2005. 2, 5
- [14] J. Yang, J. Wright, T. S. Huang, and Y. Ma. Image super-resolution as sparse representation of raw image patches. In *CVPR*, 2008. 1, 6
- [15] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Generalized belief propagation. In *NIPS*, pages 689–695, 2000. 5
- [16] P. N. Yianilos. Data structures and algorithms for nearest neighbor search in general metric spaces. In *SODA*, pages 311–321, 1993. 2
- [17] R. Zeyde, M. Elad, and M. Protter. On single image scale-up using sparse-representations. In *Curves and Surfaces*, pages 711–730, 2010. 1, 6, 7