

Image Patching for Image Denoising: A Review

Ashutosh Garg^{1*} and Amit Shrivastava²

¹M.Tech. Student, Department of Electronic and Communication Engineering, VNS Faculty of Engineering, Bhopal (Madhya Pradesh), INDIA

²Assistant Professor, Department of Electronic and Communication Engineering, VNS Faculty of Engineering, Bhopal (Madhya Pradesh), INDIA

* Correspondence: E-mail: ashutoshgarg92@gmail.com

(Received 28 July, 2017; Accepted 04 Sept, 2018; Published 12 Sept, 2018)

ABSTRACT: Usually we have a large number of devices by which we can take pictures, from a digital camera to a cell phone. Problem is that usually we take the picture we don't see some errors that can be present in the image until we observe it in a computer or printed image. Now the real problem arises because it is almost impossible to recapture the same moment in a new photo. Hence to find a solution for this problem comes the need to implement a method to correct some of the defects that appear. In this paper we compare different methods, problems and their proposed solutions algorithms to patch an original image, using different texture image techniques. Traditional match matching algorithms treat each pixel/patch as an independent sample and build a hierarchical data structure, such as the kd-tree, to accelerate nearest patch finding. However, most of these approaches can only find approximate nearest patch and do not explore the sequential overlap between patches. Hence, they are neither accurate in quality nor optimal in speed.

Keywords: Image Patching; patch Detection contrast enhancement; texture synthesis algorithm and Image Denoising.

INTRODUCTION: Liu et. al., 2016¹ have used hybrid regularizes-based adaptive anisotropic diffusion method for image denoising. They used the method to eliminate the stair casing effect for total variation filter and synchronously avoid the edges blurring for fourth order PDE filter. They use non-linear method which have a good balance between noise removal and edge preserving. They proposed an adaptive diffusion model for image denoising which is composed of a hybrid regularization. Experiment show that proposed model can preserve important structure such as edge and corner.

Taeg Sang Cho. et al., 2011² have given, the image patching demonstrated that the patch transform can be used in several image editing operations. The patch transform provides an alternative to an extensive user intervention to generate natural looking edited images. It has to specify two inputs to reconstruct an image: the bounding box that contains the object of interest and the desired location of the patches in the bounding box. The algorithm is robust to changes in the size of the bounding box. That found it the best to fix as small a region as possible if the user wants to fully explore space of natural looking images. However, if the user wants to generate a natural-looking image with a small number of iterations, it's better to fix a larger region in the image. The algorithm is quite robust to changes in the rela-

tive location of bounding boxes, but the user should roughly place the bounding boxes in such a way that a natural looking image can be anticipated.

After this has modified the patch statistics of the original image, or has constrained some patch positions, It want to perform an inverse patch transform, piecing together the patches to form a plausible image. To accomplish this, we define a probability for all possible combination of patches. In a good placement of patches, (1) adjacent patches should all plausibly fit next to each other, (2) each patch should not be used more than once (in solving the patch placements, we relax this constraint to each patch seldom being used more than once), and (3) the user's constraints on patch positions should be maintained. Each of these requirements can be enforced by terms in a Markov Random Field (MRF) probability.

Jose M. Celaya-Padilla et. al., 2012³ have demonstrated the texturing patching process includes a variety of issues such as access, sampling, and filtering, texture patching is the process to translate one texture to an image, this is commonly used to extract objects from an scene, and fill full the empty area with some texture that blend with the rest of the image, this can be found widely on animated movies, videogames and digital content. The mapped image, usually rec-

tangular, is called a texture map or texture, and is used to generate a new texture, of N by M size, which fill full the image, in order to do that, we developed and strategy to overcome this problem, first we find the area that we want to fill, in order to do that, we search for the specific area of interest, in this case, we arbitrary edited an image to draw a square figure, with a non common color, such as Cyan, or Magenta, this area is showed in (figure 1), once the area is detected, the dimensions of the local area are obtained and used to calculate the number of different mask's to be extracted, this is done by dividing the area's high by the mask high, once we know the number of mask's we extract a maskj of the adjacent area, this mask, will work as seed, to generate a texture area, after the new texture is generated.

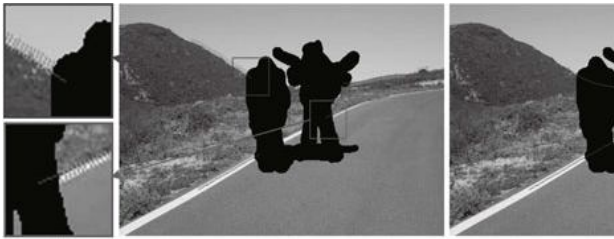


Figure 1: Structure completion for patch Detection.

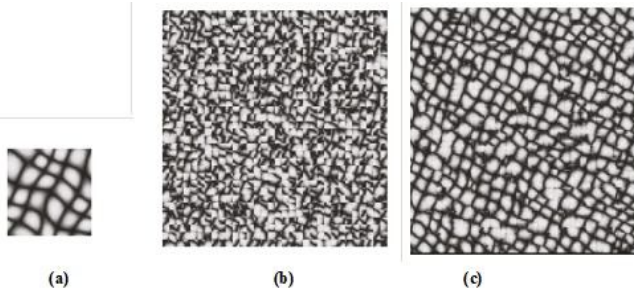


Figure 2: Image generated by the texture synthesis algorithms using 5 pixels block (a) Sample texture; (b) Random square blocks; (c) proposed.³



Figure 3: Test images (a) Original Image; (b) Edited input image.

David Harbater, 1999⁴ has evolved method the rigid approach to patching is often regarded as more intuitive than the formal approach, its foundations are less well-established. But constructions involving the formal approach have tended to be technically more cumbersome. The purpose of the current paper is to

build on previous formal patching results to create a framework in which such constructions are facilitated.

In the process we prove a result asserting that singular curves over a field that can be thickened to curves with prescribed behavior in a formal neighborhood of the singular locus, and similarly for covers of curves. Afterward, we obtain applications to fundamental groups of curves over large fields.

In, Idan Ram, 2013⁵ proposed work took a different approach, and proposed a denoising scheme which consists of reordering the noisy image pixels to what should be a 1D regular signal, and applying it linear smoothing filters. Then applied several permutations to the image, each was obtained by calculating distances between the noisy image patches, and ordering them such that they were chained in the shortest possible path, essentially solving the traveling salesman problem. They note that similar permutations were employed in Generalized Tree-Based Wavelet Transform and Redundant Wavelets on Graphs and High Dimensional Data Clouds to construct image-adaptive wavelet transforms, which were used for image denoising. First, start by calculating K permutation matrices P_k from the image patches x_j . Then apply these matrices to z and obtain $z_k^p = P_k z$. We wish to modify the NL-means algorithm [1] so it will make use of these signals. The NL-means algorithm estimates each pixel $\hat{y}[n]$ as a weighted average of pixels in Z , which reside in a square neighborhood S_n^{NL} surrounding $z[n]$. The weights are determined by the distances between the patch surrounding the estimated pixel and the patches surrounding the pixels in

$$\hat{y}[n] = \frac{1}{D_n} \sum_{m \in S_n^{NL}} z[m] w_{n,m}$$

Where, the weights $w_{n,m}$ and D_n satisfy.

$$w_{n,m} = \exp\left(-\frac{\|X_n - X_m\|^2}{g\gamma}\right), D_n = \sum_{m \in S_n^{NL}} w_{n,m}.$$

Let P_k denote a vector containing the permutation of the pixel indices applied by the matrix P_k .

It next use the permutations to construct for each pixel $z[n]$ a neighborhood S_n . Let j_n^k be the index of the pixel $y[n]$ in y_k^p , i.e. $y_k^p[j_n^k] = y[n]$. As y_k^p should be smooth, $y_k^p[j_n^k]$ should be close to its Q neighboring pixels, and therefore it chooses to estimate it from their noisy versions. Thus, first define the neighborhood $S_{k,n}$ of $z[n]$ as the set of indices.

$$S_{k,n} = \left\{ P_k \left[j_n^k - \frac{Q}{2} \right], \dots, P_k \left[j_n^k + \frac{Q}{2} \right] \right\}$$

Then define the total neighborhood of $z[n]$ as

$$S_n = \bigcup_{k=1}^K S_{k,n}$$

image denoising algorithm by replacing the neighborhood S_n^{NL} with S_n in equation (4) and (5). As can be seen, unlike the algorithm in¹³ we do make use of the distances between the patches, and we do not require a preliminary filter-learning stage. The algorithm¹³ proposed in Image processing using smooth ordering of its patches had two main limitations: 1) the distances between the patches were not employed in the denoising scheme, although they carry additional information regarding patch similarity; 2) the smoothing filters required a Patch classification the image differently than areas with edges or texture. It first divide the patches into two sets: S_s - which contains smooth patches, and S_e - which contains patches with edges or texture. Let $\text{std}(xi)$ denote the standard deviation of the patch xi and let C be a scalar design parameter. Next divide the image z into two signals: z_s - which contains the pixels corresponding to the smooth patches, and z_e - which contains the pixels corresponding to the patches with edges and texture. Then apply the denoising scheme described above to the signals z_s and z_e , with the sets of patches S_s and S_e , and two sets of parameters Q_s, γ_s and Q_e, γ_e for the NL-means, respectively. Then recover the two signals \hat{y}_s and \hat{y}_e from z_s and z_e , respectively, and obtain the final estimate \hat{y} by returning the pixels in each signal to their original place in the image canvas. separate training set to be learned from.

The Inverse Patch Transform: After the user has modified the patch statistics of the original image, or has constrained some patch positions, we want to perform an “inverse patch transform”, piecing together the patches to form a plausible image. To accomplish this, we define a probability for all possible combination of patches.

In a good placement of patches,

- (1) adjacent patches should all plausibly fit next to each other,
- (2) each patch should not be used more than once (in solving the patch placements, we relax this constraint to each patch seldom being used more than once), and
- (3) the user’s constraints on patch positions should be maintained.

Each of these requirements can be enforced by terms in a Markov Random Field (MRF) probability.

Nearest Patch Matching: Nearest patch search algorithms can be roughly classified into two categories: the exact nearest patch matching and approximate nearest patch matching. PCA Trees, K-means, are often used to achieve exact nearest patch matching. Currently, there are several methods, such as Kd-Tree¹⁰, ANN, TSVQ and Vantage Point Trees, that can perform both exact and approximate nearest patch matching. All these methods apply hierarchical tree structure to accelerate searching.

Some other methods such as local propagation method, k-coherence technique and randomized correspondence algorithm only perform approximate nearest patch matching. These approximate methods find the approximate nearest patch in local regions based on local coherence assumption. The performance of the nearest patch matching method usually depends on several factors: including image size, patch size, the number of nearest patches, search range, and the number of input images. The kd-tree based matching is one of the most widely-used algorithms for finding the nearest patch. Although it is easy to create and efficient for range query, the kd-tree only works well for low-dimensional data. Using kd-tree, the number of searched nodes increases exponentially with the space dimension.

The dimensionality of the search space: the number of splits tends to grow approximately exponentially with the dimensionality. However, in the application presented here, this dimensionality is always fixed at four. The distribution of the patches in the images: the number of splits tends to decrease strongly if good matches are present. The number of matching labels between R and S: fewer matches allow to reduce the match lists and to find the solution with fewer splits. The accuracy constraints imposed: if a more precise solution is needed, the number of splits increases. The Noises contained in an image.

Image models enable us to systematically develop algorithms for accomplishing a particular image-related task.

Depending on the specific task (e.g., recognition vs. compression), we often face two classes of image models-descriptive and generative often associated with the tasks of image analysis and synthesis respectively. Descriptive models focus on the extraction of distinctive features from a given image such that they can facilitate the task of classifying the image into one of several classes.¹⁴ Whether a plausible image can be reconstructed from those features is irrelevant to descriptive models.

By contrast, generative models do preserve the “information” in an image and their synthesis capability lends them more desirable for the task of compression and restoration than that of classification and recognition. However, recent trend in sparsity-based recognition has challenged the above common perception about generative models - they could achieve highly competent performance in recognition tasks even though the capability of synthesis is not necessary.

The influential paradigm is writer’s own observation that the design principle has historically followed two competing yet complementary paths: the construction of a signal-independent dictionary (or basis functions) and the learning of from training data (and therefore signal-dependent). Both lines of attacks have turned out fruitful and it has often been found that two approaches end up with dictionaries of similar characteristics and algorithms of comparable performance.

The patch transform represents an image as bag of overlapping patches sampled on a regular grid. This representation allows users to manipulate images in the patch domain, which then seeds the inverse patch transform to synthesize a modified image. Possible modifications in the patch domain include the spatial locations of patches, the size of the output image, or the pool of patches from which an image is reconstructed. When no modifications are made, the inverse patch transform reduces to solving a jigsaw puzzle. The inverse patch transform is posed as a patch assignment problem on a Markov random field (MRF), where each patch should be used only once, and neighboring patches should fit to form a plausible image. It find an approximate solution to the MRF using loopy belief propagation, introducing an approximation that encourages the solution to use each patch only once.

The image reconstruction algorithm scales well with the total number of patches through the use of a label pruning method that finds loops of patches that are likely to fit together. In addition, structural misalignment artifacts are suppressed through a patch jittering scheme that spatially shifts the assigned patches by a sub-patch size.

Robust Patch regression: It is well-known ℓ_1 minimization is more robust to outliers than ℓ_2 minimization. A simple argument is that the unsquared residuals $\|P - P_j\|$ are better guarded against the aberrant data points compared to the squared residuals $\|P - P_j\|^2$. The former tends to better suppress the large residuals that may result from outliers. This basic principle of robust statistics can be traced back to the

works of von Neumann, Tukey¹⁷, and Huber¹⁸, and lies at the heart of several recent work on the design of robust estimators e.g., see¹⁵, and the references therein. A natural question is what happens if we replace the regression ℓ_1 . In general, one could consider the following class of problems:

$$\hat{P}_i = \arg \min P \sum_{j \in S(i)} w_{ij} \|P - P_j\|^p$$

The intuitive idea here is that, by taking smaller values of P , we can better suppress the residuals $\|P - P_j\|^2$ induced by the outliers. This should make the regression even more robust to outliers, compared to what we get with $P = 1$. We note that a flip side of setting $P < 1$ is that (6) will no longer be convex (this is essentially because t is convex if and only if $P \geq 1$), and it is in general difficult to find the global minimizer of a non-convex functional. However, we do have a good chance of finding the global optimum if we can initialize the solver close to the global optimum. The purpose of this note is to numerically demonstrate that, for all sufficiently large σ , the \hat{u} obtained by solving (6) (and letting \hat{u}_i to be the center pixel in \hat{P}_i) results in a more robust approximation of f as $p \rightarrow 0$, than what is obtained using NLM. Henceforth, we will refer to (6) as Non-Local Patch Regression, where p is generally allowed to take values in the range $(0, 2)$.

The accuracy of estimating the parameters is dependent on the strength of the noise corrupting the image. Noise affects different parameter estimation steps differently. The moment estimation steps are dependent on the ability of the clustering step to classify structurally similar patches. Although the LARK features are quite robust, errors in clustering due to noise cannot be fully avoided. This is demonstrated in Fig. 4 where differences in clustering the noisy and noise-free images are apparent.

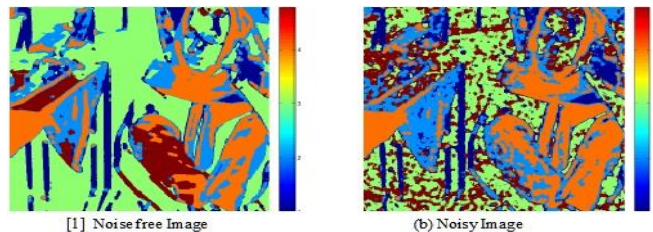


Figure 4: Clustering of Barbara image into 5 clusters based on geometric structure of patches. Clustering is performed with features calculated from (a) clean image, and (b) noisy image of noise standard deviation.

Note how the kernel features can capture structural information and thereby properly cluster majority of patches even in the presence of noise.

$$W_{ij} = \frac{1}{\sigma^2} \exp\left\{\frac{\|Y_i - Y_j\|^2}{h^2}\right\}$$

Although outliers do influence the moment estimates, the process that is most sensitive to noise is the weight calculation of above. Identifying photometrically similar patches becomes challenging in the presence of strong noise, which in turn influences the similarity measure calculation of Eq. above.

To alleviate such detrimental effects of noise, pre-filter the image once before the parameters of the proposed framework are learned. Note that such pre-filtering is quite typical of competing approaches, and is necessary only for strong noise. For the pre-processing step, apply algorithm once on the input noisy image with a reduced noise variance estimate to ensure that finer details are not lost. The necessary filter parameters are then learned from the resultant noise-suppressed image.

This maximization will be a complex task for most functional forms of Q. In many applications, such fits of parameters are carried out iteratively and heuristically, which involves the risk that the results found are only locally optimal solutions. Other methods include randomized approaches like e.g. random sample consensus¹⁸ employ a branch-and-bound technique⁹ to perform the maximization. This algorithm guarantees to find the globally optimal parameter set by recursively subdividing the parameter space and processing the resulting parameter hyper-rectangles in the order given by an upper bound on the total quality.

Moreover, with small modifications, the algorithm allows us to efficiently determine the k best matches, not only the best match. (a), (c) show the region of the search space that is considered and (b),(d) show possible matchings of a model to points in the image for transformations with parameters contained in the region. (Note that these are not computed explicitly in the algorithm, but an upper bound of the quality for all possible matches is determined instead.) After splitting the region (c), (d), fewer transformations are possible and the upper bound for the quality of a match is recomputed accordingly. This process is repeated for each of the subregions. Figure 5 shows an illustration of a subdivision of the transformation space and Figure 6 shows the subdivisions occurring during an actual run of the algorithm.

The number of times the initial region is split before a solution is reported. The interactions between the following variables influence the dimensionality of the search space: the number of splits tends to grow approximately exponentially with the dimensionality.

However, in the application presented here, this dimensionality is always fixed at four. The distribution of the patches in the images: the number of splits tends to decrease strongly if good matches are present. The number of matching labels between R and S: fewer matches allow to reduce the matchlists and to find the solution with fewer splits. The accuracy constraints imposed: if a more precise solution is needed, the number of splits increases.

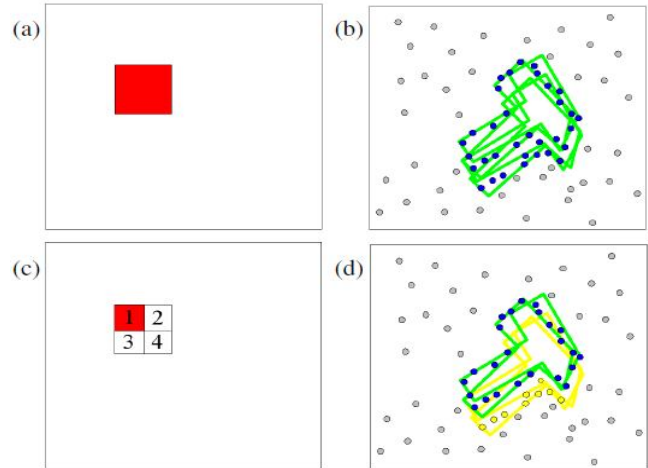


Figure 5: Illustration of the subdivision step within the RAST algorithm.

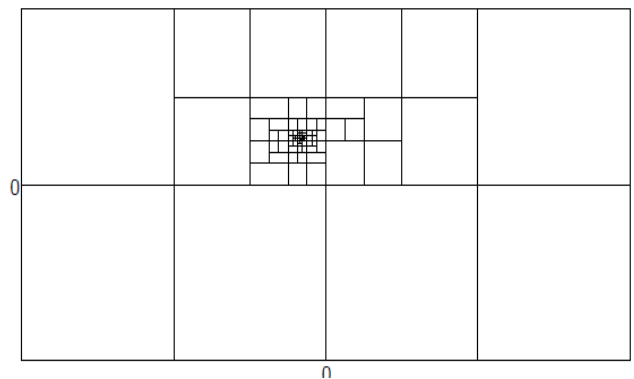


Figure 6: Illustration of the explored space during an actual run of the RAST algorithm.

Paolo Piro et. al., 2009¹¹ propose a new descriptor based on Sparse Multiscale Patches. In short, they integrate using probability distributions the local information brought by the SMP. The key aspects of these descriptors are the following:

- A multiscale representation of the images;
- Inter/intrascale patches that describe locally the structure of the image at a given scale;
- A sparse repartition: most of the energy is concentrated in a few patches.

Note that the occurrence in different parts of an image of similar patches of spatially neighboring pixels has been exploited in image processing.¹⁴⁻¹⁶ Here the

concept is used for multiscale coefficients as proposed in.¹⁷ The visual content of images is represented by patches of multi-resolution coefficients. The extracted feature vectors are viewed as samples from an unknown multidimensional distribution. The multiscale transform of an image being sparse, a reduced number of patches yields a good characterization of the distribution. They estimate the similarity between images by a pseudo-distance (or measure) between these multidimensional probability density functions. Also proposed to use the Kullback-Leibler (KL) divergence as a similarity measure that quantifies the closeness between two probability density functions. Such measure has already shown good performances in the context of image retrieval.¹³ It has already been used for the simple case of parametrized marginal distributions of wavelet coefficients,¹⁸ assuming independence of the coefficients.

In contrast, it defines multidimensional feature vectors (patches) that capture interscale and intrascale dependencies among subband coefficients. These are better adapted to the description of local image structures and texture. In addition, for color images, we take into account the dependencies among the three color channels; hence patches of coefficients are also interchannel. This approach implies to estimate distributions in a high-dimensional statistical space, where fixed size kernel options to estimate distributions or divergences fail. Alternatively the proposed to estimate the KL divergence directly from the samples by using the k-th nearest neighbor (kNN) approach, i.e. adapting to the local sample density. An RBM consists of a layer of binary stochastic “visible” units connected to a layer of binary, stochastic “hidden” units via symmetrically weighted connections. A joint configuration, (v, h) of the visible and hidden units has an energy.

Patch based Hierarchical PCA (PHPC): This approach aims to provide an intermediate solution between local and global PCAs, which is less time consuming than the PLPCA and is more adapted to local regions than the PGPCA. The idea is to create hybrid bases that contain elements characterizing global features of the image along with elements characterizing localized features.

This strategy allows us to combine the advantages both of the PGPCA, to accurately estimate axes which explain most of the variance, and of the PLPCA to model the behavior of rare patches. To extract the first principal axes of and to include them in all the local bases. The remaining axes will be computed from the residual patches, i.e., the patches being the projector onto the subspace spanned by the axes. In practice small and chosen between one and

five. The fact that the remaining axes are not kept is justified by the observation that, as mentioned above, they are irrelevant to model the underlying signal since they look like randomly drawn vectors from the orthogonal.

CONCLUSION: The main limitation identified is that the control over the patch location is inherently limited by the size of the patch, which can lead to visible artifacts. If patches are too small, the patch assignment algorithm breaks down due to exponential growth in the state dimensionality. A simple extension to address this issue is to represent the image with overlapping patches, and generate the output image by quilting these patches. It could define the compatibility using the seam energy. Since seams can take arbitrary shapes, fewer artifacts is expected. Another limitation recognized is the large amount of computation. To enable an interactive image editing using the patch transform, both the number of iterations and the amount of computation per iteration should be reduced. The overlapping patch transform framework may help in this regard as well since larger patches (i.e. less patches per image) can be used without degrading the output image quality.

REFERENCES:

1. Kui Liu, Jieqing and Liefu (2016) Hybrid equalizers-based adaptive anisotropic diffusion for image denoising, *Springerplus*, 5, 404.
2. Taeg Sang Cho, Moshe Butman, Shai Avidan, William T. Freeman (2011) The patch transform and its applications to image editing, Massachusetts Institute of Technology Bar-Ilan University Adobe Systems Inc.
3. Jose M. Celaya-Padilla, Carlos E. Galvan T.b, J. Ruben Delgado C.C., Issac Galvan-Tejadad, Ernesto Ivan Sandoval (2012) Multi-seed texture synthesis to fast image patching, *International Meeting of Electrical Engineering Research ENIINVIE 2012 (Science Direct proceeding)*, 35, 210-216.
4. David Harbater (1998) Patching and Thickening Problems, *Journal of Algebra*, 212, 272-304 1999 Article ID jabr.1998.7574, available online at <http://www.idealibrary.com> on IDEAL.
5. Idan Ram, Michael Elad, Israel Cohen (2013) *Image Denoising Using Nl-Means Via Smooth Patch Ordering*, Japan Technion Society Research Fund Hillman Foundation for Global Security - Collaboration Technion and University Northeastern.
6. Henry Adams, Gunnar Carlson (2013) *On the Non-Linear Statistics of Range Image Patches*,

Research supported in part by DARPA HR 0011-05-1-0007 and NSF DMS 0354543.

7. J. Weickert (1998) Anisotropic Diffusion in Image Processing, Ser. ECMI, Teubner, Stuttgart, Germany.
8. J. Weickert (1999) Coherence-enhancing diffusion filtering, *International Journal on Computer*, 31, 111–127.
9. Chourmouziou Tsotsios, Maria Petrou (2013) On the choice of the parameters for anisotropic diffusion in image processing, *Pattern Recognition*, 46.
10. Jose M. Celaya-Padilla, Carlos E. Galvan T., J. Ruben Delgado C., Issac Galvan- Tejada Ernesto Ivan Sandoval (2012) Multi-seed texture synthesis to fast image patching, *International Meeting of Electrical Engineering Research ENIINVIE 2012, Procedia Engineering*, 35, 210 – 216.
11. Paolo Piro, Sandrine Anthoine, Eric Debreuve, and Michel Barlaud (2009) Sparse Multiscale Patches for Image Processing, *Springer Verlag LNCS*, 5416, 284-304.
12. Puzicha, J., Rubner, Y., Tomasi, C., Buhmann, J. M. (1999) Empirical evaluation of dissimilarity measures for color and texture, *ICCV*, 1165–1172.
13. Buades, A., Coll, B., Morel, J. M. (2005) A review of image denoising algorithms, with a new one, *Multiscale Modeling and Simulation*, 4, 490–530
14. Awate, S. P., Whitaker, R. T. (2006) Unsupervised, information-theoretic, adaptive image filtering for image restoration, *IEEE Trans. on PAMI*, 28 364–376.
15. Angelino, C. V., Debreuve, E., Barlaud, M. (2008) Image restoration using a knn-variant of the mean-shift, *ICIP*, San Diego, USA.
16. Portilla, J., Strela, V., Wainwright, M., Simoncelli, E.P. (2003) Image denoising using a scale mixture of Gaussians in the wavelet domain, *TIP (Transaction on image processing)*, 12, 1338–1351.
17. Do, M., Vetterli, M. (2002) Wavelet based texture retrieval using generalized Gaussian density and Kullback-Leibler distance, *TIP (Transaction on image processing)*, 11, 146–158
18. Wang, Z., Wu, G., Sheikh, H. R., Simoncelli, E. P., Yang, E. H., Bovik, A. C. (2006) Qualityaware images, *TIP (Transaction image processing)*, 15, 1680–1689.
19. paolo piro, Sandrine anthoine, Eric debreuve, and Michel barlaud (2009) Sparse multiscale patches for image processing, *springer verlag LNCS*, 54(16), 284-304.
20. puzicha, J., Rubner, y., Tomasi, C., Buhman, J.M (1999) Emprical evaluation of dissimilarity measures for color and texture, *International journal of computer application, graphics and vision*, 1165-1172.