INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH & DEVELOPMENT

ISSN 2278 – 0211 (Online)

Image Processing
Using Patch Based Processing and Filtering Techniques

M. Lakshmi Chaithra
Department of Computer Science Engineering
Saveetha School of Engineering, Saveetha University, Thandalam, Chennai, India

Abstract:
We propose a picture process scheme supported rearrangement of its patches. For a given corrupted image, we tend to extract all patches with overlaps, visit these as coordinates in high dimensional house and organize them such they're enchained Within the "shortest potential path," basically finding the commercial traveler drawback. The obtained ordering applied to the corrupted image implies a permutation of the image pixels to what ought to be a daily signal. This permits USA to get smart recovery of the clean image by applying comparatively easy one-dimensional smoothing operations. Before stepping into additional steps the enhancement of clamant image is critical, for removing clamant image that is corrupted by the salt and pepper noise by exploitation mean filter. Presently concentrating on the removing the noise, increasing the bar chart of a picture and purpose mapping based mostly upon the color, abstraction location and motion. Demonstrating the effectiveness and also the machine potency of each depth estimation and image de-noising are mentioned and their drawbacks and advantages are given.

1. Introduction

IN RECENT years, image processing using local patches has become very popular and was shown to be highly effective for representative work. The core idea behind these and many other contributions is the same: given the image to be processed, extract all possible patches with overlaps; these patches are typically very small com- pared to the original image size (a typical patch size would be 8×8 pixels). The processing itself proceeds by operating on these patches and exploiting interrelations between them. The manipulated patches (or sometimes only their center pixels) are then put back into the image canvas to form the resulting image. There are various ways in which the relations between patches can be taken into account: weighted averaging of pixels with similar surrounding patches, as the NL-Means algorithm does , clustering the patches into disjoint sets and treating each set differently, , seeking a representative dictionary for the patches .This paper highlights the removal of noise or distortion gift in each grey scale image and color image and conjointly to extend the bar chart of a picture that is littered with the salt and pepper noise, before moving to depth recovery half, the depth based mostly rendering method are exhausted the long run. The removal of noise is completed by exploitation some filtering technique like mean, median, guassian, minimum and most filtering[37]. Here we have a tendency to solely think about the median filter so as to get the correct image with the image quality.

2. Methods of smooth patching:

2.1. Basic Scheme

Let Y be an image of size N1 × N2 where N1N2 = N, and let Z be a corrupted version of Y, which may be noisy or contain missing pixels. Also, let z and y be the column stacked versions of Z and Y, respectively. Then we assume that the corrupted image satisfies z = My+v (1) where the N × N matrix M denotes a linear operator which corrupts the data, and v denotes an additive white Gaussian noise independent of y with zero mean and variance σ2. In this work the matrix M is restricted to represent a point- wise operator, covering applications such as denoising and inpainting. The reason for this restriction is the fact that we will be permuting the pixels in the image, and thus spatial operations become far more complex to handle. Our goal is to reconstruct y from z, and for this end we employ a permutation matrix P of size N × N. We assume that when P is applied to the target signal y, it produces a smooth signal yp = Py. We will explain how such a matrix may be obtained using the image patches in Section II-B. We start by applying P to z and obtain zp = Pz. Next, we take advantage of our prior knowledge that yp should be smooth, *2.2. Building the Permutation Matrix P*

We wish to design a matrix P which produces a smooth signal when it is applied to the target image y. When the image Y is known, the optimal solution would be to reorder it as a vector, and then apply a simple sort operation on the obtained vector. However, we are interested in the case where we only have the corrupted image Z (noisy, containing missing pixels, etc.), and any permutation is defined by simply reordering the corrupted pixels into a regular signal does not necessarily smooth y. Therefore, as the pixels in the corrupted image are not helpful to us, we settle for a suboptimal ordering operation, using patches from the corrupted image. Let y_i and z_i denote the ith samples in the vectors y and z, respectively. We denote by x_i the column stacked version of the $\sqrt{n}\times\sqrt{n}$ patch around the location of z_i in Z. We assume that under a distance measure2 $w(x_i,x_j)$, proximity between the two patches x_i and x_j suggests proximity between the uncorrupted versions of their center pixels y_i and y_j. Thus, we shall try to reorder the points x_i so that they form a smooth path, hoping that the corresponding reordered 1D signal y_p will also become smooth. The "smoothness" of the reordered signal y_p can be measured using its total-variation measure

2.3. Image Processing Using Smooth Ordering of Its Patches 2767

Two normalized histograms of the spatial distances between adjacent patches, shown in Figure 2. Figure 2(c) shows that when the search neighborhood is not restricted, only about 5% of neighboring patches in the path are also immediate spatial neighbors, and that far away patches are often assigned as neighbors in the reordering process. The histogram in Figure 2(d), obtained with restricted search neighborhood, is limited to show only distances which are smaller or equal to 43, the maximal possible distance within the search window. It can be seen that despite the restriction to a smaller search neighborhood, only about 11% of neighboring patches in the path are also immediate spatial neighbors, and patches all over the search neighborhood are assigned as neighbors in the reordering process. In order to facilitate the cycle-spinning method mentioned above, we simply run the proposed ordering solver K times, and the randomness (both in the initialization and in assigning the neighbors) leads to different permutation results. We next describe how the quality of the produced images may be further improved using a subimage averaging scheme, which can be seen as another variation of "cycle spinning".

2.4. Subimage Averaging

Let $N_p=(N_1-\sqrt{n}+1)(N_2-\sqrt{n}+1)$ denote the number of overlapped patches in the image Z, and letX be an $n\times N_p$ matrix, containing column stacked versions of these patches. We extract these patches column by column, starting from the top left-most patch. When we calculated P as described in the previous section, we assumed that each patch is associated only with its middle pixel. Therefore P was designed to reorder the signal composed of the middle points in the patches, which reside in the middle row of X. However, we can alternatively choose to associate all the patches with a pixel located in a different position, e.g., the top left pixel in each patch. This means that the matrix P can be used to reorder any one of the signals located in the rows of X. These signals are the column stacked versions of all the n subimages of size $(N_1-\sqrt{n}+1)\times(N_2-\sqrt{n}+1)$ contained in the image Z. We denote these subimages by $\tilde{Z}_j$, $j = 1,2,...,n$.

2.5. Connection to Bm3d and Clustering Based Method

The above processing scheme can be described a little differently. We start by calculating the permutation matrix P from the image patches x_i. We then gather the patches by arranging them as the columns of a matrix X_p in the order defined by P. This matrix contains in its rows the reordered subimages $\tilde{z}^p_j$, therefore we next apply the operator H to its rows, and shuffle the columns of the resulting matrix according to the permutation defined by P^{-1}. We obtain a matrix $\bar{X}$, which contains in its rows the reconstructed subimages $\hat{\tilde{y}}_j$, and in its columns reconstructed versions $\bar{x}_i$ of the image patches x_i.

3. Filtering Operations

3.1. 2 Image De-Noising

Image de-noising is additionally referred to as filter, it's the method of removing the assorted forms of noise or unwanted data gift in a picture primarily based upon their properties whereas keeping the main points of the image preserved. Image filtering isn't solely accustomed improve the image quality however additionally used as a preprocessing stage in several applications as well as image encryption, pattern recognition. General purpose image filter lack the flexibleness and adaptableness of un-modeled noise sorts. Pictures area unit typically corrupted by random variations in intensity, illumination, or have poor distinction and can't be used directly. The term Filtering is outlined as rework component intensity values to reveal sure image characteristics

- Enhancement: improves distinction
- Smoothing: take away noises
- Template matching: detects better-known patterns.

Noise reduction involves 2 steps one is noise detection and another one is noise replacement. The noise detection is that the opening move, location of noise is known. The noise replacement is that the second step, within which the detected reedy pixels area unit replaced by the calculable values.

3.2. Noise Removal Technique

- Minimum Filtering
- Maximum Filtering
- Mean Filtering
- Median Filtering
- Guassian Filtering
- Rank Order Filtering
- In Minimum filtering technique, this component is replaced by the minimum component worth of its neighboring pixels.
- In most filtering technique, this component is replaced by the most component worth of its neighboring pixels.
- In Mean filtering technique, current component is replaced by the {arithmetic mean, first moment, expectation, expected worth, mean, mean value} value of its neighboring pixels.
- In Median filtering technique, current picture {element, component, constituent, element} is replaced by the middle element of its neighboring pixels.
- In ordering filter, current component is replaced by the user outline order its neighboring pixels. For Example: (order=20)
- In Guassian filtering technique, the reedy component is replaced by the resulted worth of multiplication of kernel matrix and designated region from the image. it's accustomed take away the noise and blur from a picture

4. Noise Replacement

There are unit several ways used for reducing noise. However the normal median filter and mean filter area unit accustomed scale back the salt and pepper noise and Guassian noise severally. The noise replacement is that the method of exchange the calculable values within the place of detected reedy pixels. During this approach a picture with the salt associated pepper noise is taken as an input and it's removed by mistreatment the median filter so as to realize the great quality of the image compared to all or any the filters.

4.1. Salt and Pepper Noise Removal

It is additionally referred to as impulse noise or spike noise. This noise will be caused by sharp and sudden disturbance within the image signal. It look is arbitrarily scattered as white and black dot (or both) pixels on the image. a picture containing salt and pepper noise can have dark pixels in bright regions and bright pixels in dark regions. This kind of noise is caused by the dead pixels, analog to digital device error and bit error whereas transmission. If the intensity worth of the component is a smaller amount than or adequate zero then there's pepper noise and if the intensity worth of component is bigger than or adequate 255 then there's salt noise. There are solely 2 risk values exists that's, a and b. and also the likelihood of every is a smaller amount than 0.2

Intensity value of pixel at position(x, y) = ≤

$$\begin{cases} 0, \text{ pepper noise} \\ \geq 255, \text{ salt noise} \end{cases}$$

4.2. Median Filter

Median filter is that the easy and powerful filter. It's used for reducing the number of intensity variation between one component and also the alternative component. During this filter, we tend to replace component worth with the average. The median is calculated by initial sorting all the component worth's into ascending order so replaces the component being calculated with the center component value.

L (u,v)→mid{|(u+i,v+j)|(i,j)ЄR}

- Step 1: Put the pixel values of the surrounding (of noisy pixel) pixels in a single dimensional array
- Step 2: Sort this array in ascending order
- Step 3: The noisy pixel value is replaced by middle element of the sorted array.

Syntax to remove the salt and pepper noise using median filter:

- Read the image from the file system to matrix I
- Createsa new figure to show the image.
- Show the loaded imageasafigure1.
- Apply median filter using the function medfilt2.
- Show image after applying the filter asafigure2.
- Write the new image to the file system.

Figure 5.3(a): Original Image
Figure 5.3(b): Image with Salt and Pepper Noise
Figure 5.3(c): Filtered Image by Using Median Fitter

5. Conclusion

We have proposed a new image processing scheme which is based on smooth 1D ordering of the pixels in the given image.We have shown that using a carefully designed permutation matrices and simple and intuitive 1D operations such as linear filtering and interpolation, the proposed scheme can be used for image denoising and inpainting, where it achieves high quality results.Therefore, we tend to specialize in the image de-noising, the planned ways area unit utilized in order to scale back the unwanted data or distortion that is termed as noise which will be caused by the external force whereas a picture is being transmitted, where as transmittal a picture knowledge over Associate in Nursing unsecure channel, a noise also can be other by effort. This paper highlights the noise removal ways for the grey scale image likewise because the color image and conjointly to extend the bar graph of a picture that is corrupted by the salt and pepper noise. For removing the salt and pepper noise, noise filtering techniques area unit used like minimum, maximum, mean, median and guassian. However the median filter produces the proper image compared to all or any alternative filtering techniques.

6. References

1. A. Buades, B. Coll, and J. M. Morel, “A review of image denoising algorithms, with a new one,” Multiscale Model. Simul., vol. 4, no. 2, pp. 490–530, 2006.
2. P. Chatterjee and P. Milanfar, “Clustering-based denoising with locallylearned dictionaries,” IEEE Trans. Image Process., vol. 18, no. 7, pp. 1438–1451, Jul. 2009.
3. G. Yu, G. Sapiro, and S. Mallat, “Image modeling and enhancement via structured sparse model selection,” in Proc. 17th IEEE Int. Conf. Image Process., Sep. 2010, pp. 1641–1644.
4. G. Yu, G. Sapiro, and S. Mallat, “Solving inverse problems with piecewise linear estimators: From Gaussian mixture models to structured sparsity,” IEEE Trans. Image Process., vol. 21, no. 5, pp. 2481–2499, May 2012.
5. W. Dong, X. Li, L. Zhang, and G. Shi, “Sparsity-based image denoising via dictionary learning and structural clustering,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2011, pp. 457–464.
6. W. Dong, L. Zhang, G. Shi, and X. Wu, “Image deblurring and superresolution by adaptive sparse domain selection and adaptive regularization,” IEEE Trans. Image Process., vol. 20, no. 7, pp. 1838–1857, Jul. 2011.
7. D. Zoran and Y. Weiss, “From learning models of natural image patches to whole image restoration,” in Proc. IEEE Int. Conf. Comput. Vis., Nov. 2011, pp. 479–486.
8. M. Elad and M. Aharon, “Image denoising via sparse and redundant representations over learned dictionaries,” IEEE Trans. Image Process., vol. 15, no. 12, pp. 3736–3745, Dec. 2006.
9. J. Mairal, M. Elad, and G. Sapiro, “Sparse representation for color image restoration,” IEEE Trans. Image Process., vol. 17, no. 1, pp. 53–69, Jan. 2008.
10. J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, “Non-local sparse models for image restoration,” in Proc. IEEE 12th Int. Conf. Comput. Vis., Sep.–Oct. 2009, pp. 2272–2279.
11. R. Zeyde, M. Elad, and M. Protter, “On single image scale-up using sparse-representations,” in Proc. 7th Int. Conf. Curves Surf., 2012, pp. 711–730.
12. K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, “Image denoising by sparse 3-D transform-domain collaborative filtering,” IEEE Trans. Image Process., vol. 16, no. 8, pp. 2080–2095, Aug. 2007.
13. X. Li, “Patch-based image interpolation: Algorithms and applications,” in Proc. Int. Workshop Local Non-Local Approx. Image Process., 2008, pp. 1–6.

14. I. Ram, M. Elad, and I. Cohen, “Generalized tree-based wavelet transform,” IEEE Trans. Signal Process., vol. 59, no. 9, pp. 4199–4209, Sep. 2011.
15. I. Ram, M. Elad, and I. Cohen, “Redundant wavelets on graphs and high dimensional data clouds,” IEEE Signal Processing Letters, vol. 19, no. 5, pp. 291–294, May 2012. 2774 IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 22, NO. 7, JULY 2013
16. G. Plonka, “The easy path wavelet transform: A new adaptive wavelet transform for sparse representation of two-dimensional data,” Multiscale Model. Simul., vol. 7, no. 3, pp. 1474–1496, 2009.
17. D. Heinen and G. Plonka, “Wavelet shrinkage on paths for denoising of scattered data,” Results Math., vol. 62, nos. 3–4, pp. 337–354, 2012.
18. T. H. Cormen, Introduction to Algorithms. Cambridge, MA, USA: MIT Press, 2001.
19. M. Elad, Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing. New York, NY, USA: Springer-Verlag, 2010.
20. R. R. Coifman and D. L. Donoho, “Translation-invariant de-noising,” Wavelets and Statistics. New York, NY, USA: Springer-Verlag, 1995, pp. 125–150.
21. T. Yang, Finite Element Structural Analysis, vol. 2. Englewood Cliffs, NJ, USA: Prentice-Hall, 1986.
22. D. Watson, Contouring: A Guide to the Analysis and Display of Spatial Data (With Programs on Diskette). New York, NY, USA: Pergamon, 1992