

Exploring Image Processing and Image Restoration Techniques

Batyrkhan Sultanovich Omarov¹, Aigerim Bakatkaliyevna Altayeva² and Young Im Cho²

¹Universiti Tenaga Nasional, Kuala Lumpur, Malaysia,

²Department of Computer Engineering, Gachon University, Seongnam, Korea



Abstract

Because of the development of computers and high-technology applications, all devices that we use have become more intelligent. In recent years, security and surveillance systems have become more complicated as well. Before new technologies included video surveillance systems, security cameras were used only for recording events as they occurred, and a human had to analyze the recorded data. Nowadays, computers are used for video analytics, and video surveillance systems have become more autonomous and automated. The types of security cameras have also changed, and the market offers different kinds of cameras with integrated software. Even though there is a variety of hardware, their capabilities leave a lot to be desired. Therefore, this drawback is trying to compensate by dint of computer program solutions. Image processing is a very important part of video surveillance and security systems. Capturing an image exactly as it appears in the real world is difficult if not impossible. There is always noise to deal with. This is caused by the graininess of the emulsion, low resolution of the camera sensors, motion blur caused by movements and drag, focus problems, depth-of-field issues, or the imperfect nature of the camera lens. This paper reviews image processing, pattern recognition, and image digitization techniques, which will be useful in security services, to analyze bio-images, for image restoration, and for object classification.

Keywords: Image processing, Image restoration, Image enhancement, Super-resolution,, Object classification

Received: Aug. 18 2015
Revised : Sep. 23, 2015
Accepted: Sep. 24, 2015

Correspondence to: Young Im Cho
(yicho@gachon.ac.kr)
©The Korean Institute of Intelligent Systems

©This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. Introduction

Nowadays, video surveillance systems are used for traffic control. Many organizations, buildings, and houses have security cameras, but the resolution and quality of these cameras are insufficient because the cameras record and store all captured video on hard drives, which are expensive. In order to store recorded video at high quality, there is a need to have a huge number of volumes, which are not affordable by many. Accordingly, these systems create difficulties in video analysis by producing low-quality outputs. Captured videos and photos frequently appear blurred, distorted, or fuzzy. Digital image processing techniques are used to overcome this problem.

2. Digital Image Representation



Figure 1. Example of image in grayscale.

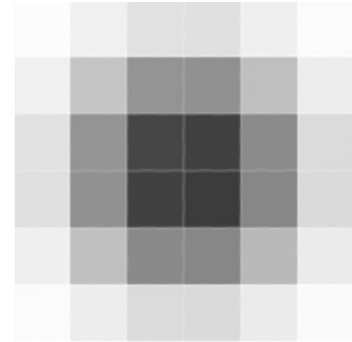
It is important to discuss image representation in computers in order to understand image-processing algorithms, which are used in the project. First, let us talk about images in *grayscale* (*panchromatic, gray-level, black-and-white*) (Figure 1) [1, 2].

This is a way of wrapping colors of an image into its brightness values. The representation of color images will be discussed later, but their constructions are accomplished in the same manner with little difference.

Grayscale images are constructed by a two-dimensional light intensity function $f(x, y)$, where x and y are coordinates of a point and the value of the function at (x, y) gives the light intensity of that pixel. This representation of an image is said to be in the *spatial domain*, which means that the image is described as a matrix of pixels with light intensity or brightness values (Figure 2) [2].

The matrix is the representation of the image of Figure 2 in the spatial domain. Each value of the matrix at (x, y) describes the light intensity. In a panchromatic image, white color takes the maximum value, and black color takes zero. In this situation, a question may arise: What number can be taken as a maximum value? In order to answer this question, it is important to define the gray level. The gray level for the image in Figure 1 is 8, and the values for color are defined as 2^m . Accordingly, each pixel can take values between 0 and 255, and white color is represented as 255. There may be different gray levels for different images. If the gray level is high, more details can be seen in the image [1, 2].

There is also a *frequency-domain* representation. For color pictures, this function's result at point (x, y) is a vector containing brightness values for spectrum colors or channels. There



251	240	225	225	240	251
240	190	148	148	190	240
225	148	65	65	148	225
225	148	65	65	148	225
240	190	148	148	190	240
251	240	225	225	240	251

Figure 2. Gray-level image representation in the spatial domain.

are different types for representing the color scheme:

- RGB: image contains three channels, and color at a pixel is represented by a set of three colors (red, green, and blue);
- CMYK: image has four channels, and the color of a pixel is a vector containing the colors cyan, magenta, yellow, and black;
- HSV (hue-saturation-value): stores color information in three channels, just like RGB, but one channel is for a brightness value and the other two store color information.

Alpha channel: a transparency value that can be included as an extra value for the color vector.

Images in this project are processed as RGB images.

A *digital image* is an image $f(x, y)$ containing spatial coordinates and brightness values. It is a two-dimensional array or collection of vectors for each color.

As mentioned above, the digitized brightness value is called a gray level or grayscale. Each element in the array is called a pixel. One image may contain one to several dozen pixels.

The mathematical representation of a digital image looks like this [3]:

$$f(x, y) = \begin{bmatrix} f(1, 1) & \cdots & f(1, N) \\ \vdots & \ddots & \vdots \\ f(M, 1) & \cdots & f(M, N) \end{bmatrix} \quad (1)$$

Usually, most image processing algorithms refer to gray images. There are various reasons:

- All operations applied to gray images can be extended to color images by applying that method for each channel of an image;
- A lot of information can be extracted from a gray image, and colors might be not necessary;
- Handling one channel takes less CPU time;
- For many years, images were black and white. Therefore, most algorithms have been created for this type of image.

There are many other reasons for using gray scaled images in image processing, but the list above shows the most important ones.

2.1 Image Processing

Along with the digitization of images, image processing has been developed to solve problems. Many image processing tools, techniques, and algorithms have been created. Not many people completely understand what image processing is and why we need it. Why do we need image processing?

There are three major problems that paved the way for the development of image processing [3]:

- Picture digitization and creating coding standards to make transmission and storing easier;
- Image restoration and making enhancements for further interpretations, for example, refining pictures of surfaces of other planets;
- Image segmentation and preparation for machine vision.

Nowadays, image processing concerns operations mainly on digital images. This is a branch of science called computer vision[3-5]. Computer vision is an important key for robotics and machine learning.

2.2 Image Restoration

In image processing, there might be a need for image enhancements to overcome physical limits such as camera resolution and degradation. Image enhancement is the process of improving an image by making the image look better [3]. Actually, we do not know how the image should finally look like, but it is possible to say whether the image is improved by considering whether more detail can be seen or unwanted noise has

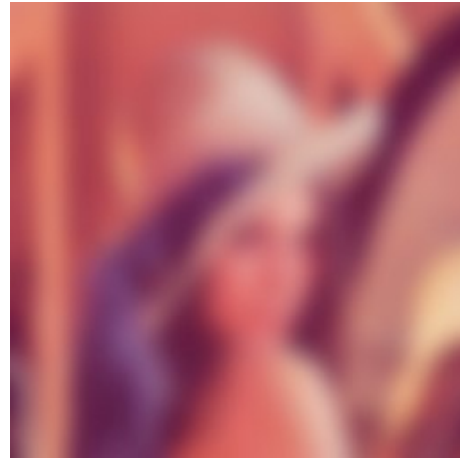


Figure 3. Blurred image.

been removed. Therefore, an image is enhanced when we have removed additive noise, increased the contrast, and decreased blurring.

In addition, when we scale up an image, it loses its quality and some details are unrecognizable. In order to avoid this, there is a need for super-resolution techniques, which allow restoring of the image after its transformations.

This project includes two types of image enhancements: restoring a blurred image and super-resolution.

2.2.1 Blurred image restoration

Restoring distorted images is one of the important problems in image processing. A specific case is blurring owing to improper focus, which is familiar to everyone. In addition to this type of degradation, there is also noise, incorrect exposition, and so on, but all of them can be restored using any image editing software.

The image below is an example of a blurred image (Figure 3).

It is believed by many that image blurring is an irreversible operation and that information is permanently lost and cannot be restored. Because each pixel is converted into a spot, everything is mixed, and with a large radius of blur we get uniform color through the entire image. However, this is inaccurate. All information is simply distributed according to some law, and can be uniquely recovered with some reservations. The only exception is the width of the image in the blur radius; full recovery is not possible for this characteristic.

2.3 Distortion Process Model

Let us examine a more formal and scientific description of image degradation and restoration processes. Only gray scaled images will be reviewed, considering that color images can be handled in the same way for each color channel [1, 2, 4]. First we define the following notations:

$f(x, y)$: Initial clear image

$h(x, y)$: Degradation function

$n(x, y)$: The additive noise term

$g(x, y)$: Degraded image

The degraded image is produced by the following formula:

$$g(x, y) = h(x, y) * f(x, y) + n(x, y) \quad (2)$$

As seen from the formula, image degradation is achieved by the convolution of the initial image with a degradation function.

The objective of restoration is to obtain an estimate $F(x, y)$ of the original image using some restoration filters. We want the estimate function to be as close as possible to the initial function. This would be so if we knew more about the degradation function and additive noise values. This is clear for functions for initial and degraded images, but it is important to provide more of an explanation for $h(x, y)$.

In the degradation process, each pixel turns into a spot, and into a section in the case of a simple blur. Alternatively, each pixel of the degraded image is obtained by pixels from some neighboring regions. All of these pixels construct the degraded image. The law of distribution for pixels is called the degradation function. The degradation function is also known as a point spread function (PSF) or kernel [6]. Usually the size of kernel function is less than the size of the initial image. In the example where we have degraded and restored a one-dimensional image, the size of the kernel was equal to two, i.e., each pixel was obtained from two.

Let us demonstrate using an example of a one-dimensional image:

The initial image is $[x_1, x_2, x_3, x_4, \dots]$.

After applying a blur, each pixel sums with the neighboring left pixel: $x'_i = x_i + x_{i-1}$.

Accordingly, we get a blurred image: $[x_1 + x_0, x_2 + x_1, x_3 + x_2, x_4 + x_3, \dots]$.

Next, we try to restore the image. In order to do this, we have to subtract from second pixel the value of the first pixel, from the third one the result of the previous subtraction, and so on.

The restored image: $[x_1 + x_c, x_2 - x_c, x_3 + x_c, x_4 - x_c, \dots]$.

Here we have added an unknown constant value x_c for each pixel with an alternating sign. It is possible to choose a constant visually; one can assume that it is approximately equal to the value of x_1 . However, everything changes when we add noise (which always exists in real images). In this scheme, at each step there will be an accumulated error value of the overall noise component, which can result in a completely unacceptable result. However, as we have seen, recovery is quite possible, even in such a primitive way.

2.4 Noise Model

Noise in digital images occurs because of image digitization or transmission. Digital camera sensors can be affected by environmental conditions, or the problem might be in the performance of the sensor itself. For instance, while taking a picture with a digital camera, the temperature and light levels can affect the resulting image. Transmission of an image through wireless networks can also corrupt the image by atmospheric disturbance. In most cases, the noise that appears is Gaussian (also called *normal*) [1], which is additive, does not correlate with the image, and is independent from the pixels.

3. Super-resolution

“Super-resolution” (SR) is a term used to describe a part of image processing methods that is designed to enhance the image and fetch a high-resolution image from one or multiple low-resolution (LR) images. Image processing programmers or researchers frequently ask what super-resolution is. Most of the time it is explained by using an example from movies about CSI or the FBI where an agent looks at a LR image and pushes the magic “enhance” button, and then the LR image becomes clear and readable. This method from TV shows is an example of SR.

The images may appear at insufficient resolutions for several reasons. They may be taken from a great distance, for example, an area photo taken by a satellite. The images may have been recorded without knowing what objects are important. These kinds of problems often appear in security camera recordings, which are positioned to record a crowd of people or a large area. Sometimes it appears that the images are taken with LR imaging sensors of cell phone cameras.

There are two approaches to achieve SR images: single-image SR and multiple-image super-resolution. As their names denote, the single-image SR method needs only one LR image to get the SR image, and multiple-image SR needs several LR

images of the same scene with slight shift to achieve the desired result. In this project, multiple-image SR is used.

In order to understand SR techniques, it is important to know about the formation of LR images. Typically, a set of LR images is recorded sequentially by a single camera. Usually there is a little motion between any pair of frames from the observing LR images, and they are blurred while passing through the system of a camera. Finally, the set of initial images is sampled at a relatively low spatial frequency. Additive noise makes the images even worse.

In this model of LR image formation, the LR image is taken from a high-resolution image by applying three linear operators and adding noise. For example, let us take a high-resolution image of a scene as Z . This is the desired result that SR algorithms try to estimate. According to our model, each LR image y_i of the high-resolution image can be described in the following way:

$$y_i = D_i F_i H_i Z + \eta_i \tag{3}$$

In the formula for the LR image, D_i represents the down sampling of the i -th image, F_i represents the transformation of the i -th image relative to the first image from the observation set, H_i represents the effect of blur on the i -th image, and η_i is the additive noise term. The main objective of the project is to estimate such a Z from the set of observed LR images using conditions described above.

3.1 Image Transformations

The process of obtaining a high-resolution image from the LR image or a set of LR images requires a basic technique from image processing for image transformations. The variable D_i from the formula of the LR image model (3) is an example of image transformation. In this section, the main important image transformation algorithms will be discussed.

The most useful transformation in this project is *interpolation* (*upsampling*). In order to interpolate an image to a higher resolution, we must select an interpolation kernel in order to convolve the image [4]:

$$g(i, j) = \sum_{k,l} f(k, l) h(i - rk, j - rl) \tag{4}$$

This formula (4) is related to the discrete convolution formula, but indexes l and k in the function $h()$ are replaced by rl and rk , respectively, where r is the upsampling rate. This process can be clearly seen in the figure below, where the process is shown from the superposition perspective. Each sample is weighted

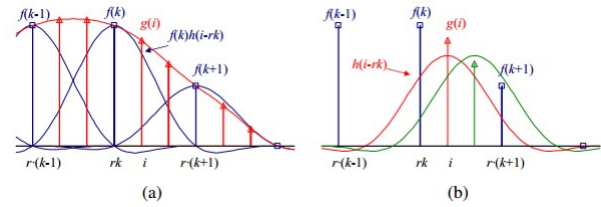


Figure 4. Signal interpolation $g(i) = \int_k f(k) h(i - rk)$: (a) weighted summation of input values and (b) *polyphase filter* interpolation.

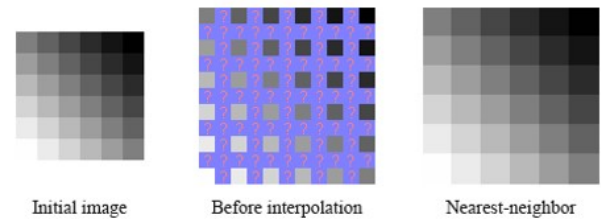


Figure 5. Example of interpolation using the nearest-neighbor method [7].

by interpolation kernels, one centered at each input sample k (Figure 4(a)). Alternatively, the mental model is shown on the right (Figure 4(b)), where the kernel is centered at the output pixel value i [4].

These two forms are equivalent, and the second form is also called a *polyphase filter* form because the kernel values can be stored as separate kernels that can be applied depending on the *phase* of i relative to the upsampled one.

There are different types of kernels, which are used depending on the application and computational time.

Nearest-neighbor interpolation: The simplest type of interpolation, which determines the gray level value from the closest pixel to the specified input coordinates and assigns that value to the output coordinates. It is important to mention that the nearest-neighbor method actually does not interpolate values but just copies values of neighboring pixels.

Figure 5 [7] shows how the nearest-neighbor method works. This type of interpolation copies the nearest pixel values by just enlarging the pixels.

Bilinear interpolation: A type of interpolation where new pixels are constructed by estimation from known pixels. Let us first review linear interpolation, which is illustrated in Figure 6 [7], where a point is estimated using known values.

The estimation value is more precise when more values are known. In the image on the left there are two known points, and

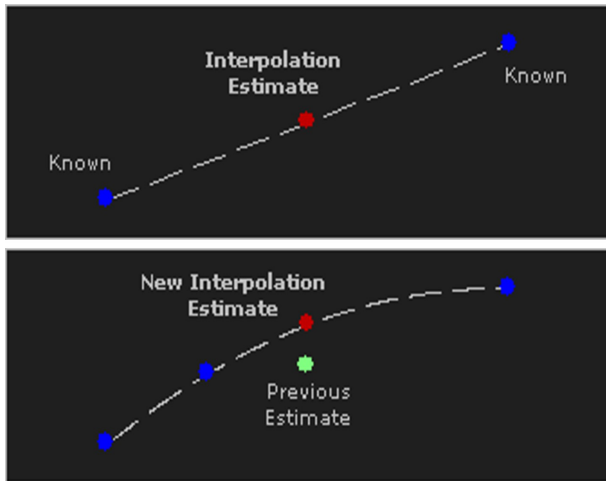


Figure 6. Linear interpolation scheme [7].

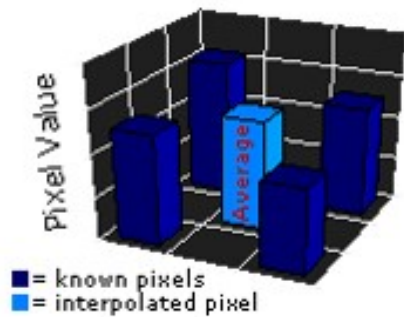


Figure 7. Bilinear interpolation [7].

the estimation point is located in the center. It can be clearly seen in the right picture how estimation becomes more accurate when a new known point is added. Bilinear interpolation uses the same principle, but the interpolating process occurs in three dimensions. This can be seen in Figure 7 [7].

Bicubic interpolation: This type of interpolation is very similar to bilinear interpolation, but it interpolates a new pixel value using the closest 4×4 pixels. This can be seen in Figure 8 [7].

Lanczos interpolation: This type of interpolation is similar to the previously described interpolation methods, but it considers the closest 8×8 pixels.

The types of interpolations were listed from less to more complex in consuming processing time and implementation. The results of interpolations become more applicable to many areas.

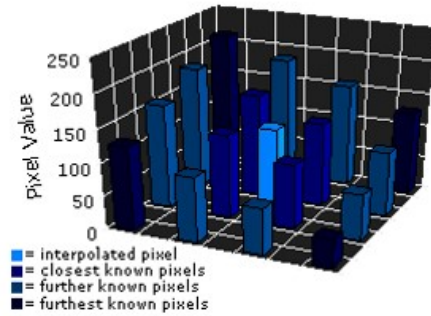


Figure 8. Bicubic interpolation [7].

3.2 Multiple-Image Super-resolution

The main difference between multiple-image SR and the described interpolation methods in the previous section is that the final result of the former is constructed by observing LR images. For the latter new pixel values are just averaged from neighboring areas. In order to obtain a high-resolution image from multiple LR images, it is important to know more information about observed LR images. For example, if we know the motion estimates of a set of LR images, then we know where to plot every pixel of each LR observation image on the high-resolution [8].

4. Conclusions

Image processing and pattern recognition techniques are helpful in analyzing images, pattern classification, image restoration, and recognition. Since a great number of techniques have been proposed, the choice of appropriate technique for a specific task is important. This paper can be used as the first step in working with image processing. Thus, for a specific task, we may need to develop a new technique. This will be made possible by the collaboration of researchers in image processing and pattern recognition. In future, we will improve image processing and restoration algorithms using image digitization and mathematical algorithms.

Conflict of Interest

No potential conflict of interest relevant to this article was reported.

Acknowledgements

This paper is supported by Gachon University in 2015.

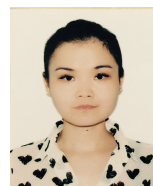
References

- [1] M.Petrou and C.Petrou, Image Processing: The Applications. London: Springer-Verlag London Limited., pp.3-10, 2008.,
- [2] R.Szeliski, Computer Vision: Algorithms and Applications. London: Springer-Verlag London Limited., pp.3-10, 2008.
- [3] G.Bradoski and A. Kaehler, Learning OpenCV. Sebastopol, CA: O'Reilly, p.1, 2008.
- [4] G.Bradoski and A. Kaehler, Learning OpenCV. Sebastopol, CA: O'Reilly, p.10, 2008.
- [5] M. Petrou and P. Bosdogianni, Image processing, 2nd ed. Chichester
- [6] : Wiley, p.293, 2010.
- [7] R.Gonzalez and R.Woods, Digital Image Processing, 2nd ed. New Jersey: Prentice Hall, p.221, 2002.
- [8] J.Parker, Algorithms for image processing and computer vision, 2nd ed. New York: Wiley Computer Pub., p.252, 2010.,
- [9] M.Petrou and C. Petrou, Image Processing: The Fundamentals, 2nd ed. JohnWiley&Sons, Ltd., p.13, 2010.
- [10] R.Gonzalez and R.Woods, Digital Image Processing, 2nd ed. New Jersey: Prentice Hall, p.222, 2002.
- [11] J.Parker, Algorithms for image processing and computer vision, 2nd ed. New York: Wiley Computer Pub., p.253, 2010.
- [12] R.Gonzalez and R.Woods, Digital Image Processing, 2nd ed. New Jersey: Prentice Hall, pp.262-263, 2002.
- [13] R.Gonzalez, R.Woods and S. Eddins, Digital Image processing using MATLAB. Gatesmark Publishing, p.187, 2009.
- [14] R. Gonzalez, R. Woods and S. Eddins, Digital Image processing using MATLAB. Gatesmark Publishing, p.199, 2009.
- [15] R.Szeliski, Computer Vision: Algorithms and Applications, 1st ed. Washington: Springer, pp.145-146, 2015.
- [16] Cambridgeincolour.com, 'Understanding Digital Image Interpolation', 2015.
- [17] J.Simpkins and R.Stevenson, 'An Introduction to Super-Resolution Imaging', M.S., in Electrical Engineering, University of Notre Dame, 2012.
- [18] J.Parker, Algorithms for image processing and computer vision, 2nd ed. New York: Wiley Computer Pub., p.286, 2010.
- [19] Docs.opencv.org, 'Cascade Classification — OpenCV 2.4.11.0 documentation'
- [20] Aram Danielyan, Vladimir Katkovnik, and Karen Egiazarian. Bm3d frames and variational image deblurring. Image Processing, IEEE Transactions on, vol.21, no.4, pp.1715–1728, 2012.
- [21] Kang M. G. Park S. C., Park M. K. Super-resolution image reconstruction: a technical overview. IEEE Signal Processing Magazine, no.20, pp.21–36, 2013.



Batyrkhan Sultanovich Omarov received his B.S. and M.Sc. from Kazakh National University named after Al-Farabi, Almaty Kazakhstan, in 2008 and 2010, respectively. He is currently a Ph.D. student at The Universiti Tenaga Nasional, Malaysia Kuala Lumpur. His research interests are image processing, computer vision, artificial intelligence, and big data.

E-mail: batyahan@gmail.com



Aigerim Bakatkaliyevna Altayeva received her B.S. and M.Sc. from the Department of Computer Science, IITU, Kazakhstan, in 2012 and 2014, respectively. Now she is a Ph.D. student at Gachon University. Her areas of interest include artificial intelligence, smart cities, big data, and information retrieval.

Tel: +82-31-750-5800

Fax: +82-31-750-5662

E-mail: aikosha1703@gmail.com



Young Im Cho received her B.S., M.Sc., and Ph.D. from the Department of Computer Science, Korea University, Korea, in 1988, 1990, and 1994, respectively. She is a professor at Gachon University. Her areas of interest include artificial intelligence, big data, and

information retrieval.

Tel: +82-31-750-5800

Fax: +82-31-750-5662

E-mail: yicho@gachon.ac.kr