Re: Generating a random matrix through a natural image

Source: http://sci.tech−archive.net/Archive/sci.image.processing/2008−08/msg00096.html

- *From*: Martin Brown <|||newspam|||@nezumi.demon.co.uk>
- *Date*: Thu, 21 Aug 2008 08:04:06 +0100

Dev wrote:

> On Aug 21, 3:14 am, aruzinsky <aruzin...@xxxxxxxxxxxxxxxxxxxx> wrote:
>
> > On Aug 20, 2:31 am, Dev <gun...@xxxxxxxxx> wrote:
> >
> > > Thanks for all the replies. I tried the ones sent by Thomas
> > > and
> > > Edward, but came up with problems. For aruzinsky's method,
> > > can you pls
> > > send me the algorithm, so that I can try it out? Meanwhile I
> > > tried a
> > > dewhitening method, though it takes away the correlations, it
> > > doesn't
> > > give the same histogram. Any further help is greatly
> > > appreciated.
> > > Thanks,
> > > Dev− Hide quoted text −
> > > − Show quoted text −
> >
> > I believe illywhacker is correct, you can't get rid of correlations by
> > permuting the pixels. However, I suspect that, for a 256x256 image,
> > the correlations are sufficiently close to zero for most or all
> > practical intents and purposes.
> >
> > You take any sorting algorithm and modify it to operate on two arrays,
> > array1 and array2. Comparisons are made only on the elements of
> > array1, whereas element movements are made on both arrays. Thus,
> > "array2 is sorted with respect to values in array1." Alternatively,
> > you can modify a sorting algorithm to sort an array2 of indices
> > without sorting array1.− Hide quoted text −
> >
> > − Show quoted text −
>
> Just to clarify my intention, I am testing an algorithm fro
> deconvolution. The algorithm works perfectly for a random image, but
> not for a natural image like lena. So I want to test whether there's

> any effect on correlation of pixels to my algorithm. For that I need
> to take away the correlation between pixels in the lena image, but
> keep the same histogram, so that the values are same.

OK. The simplest way is to shuffle the pixels. Doesn't really matter how you do it provided that the random number generator is reasonably good.

But if you are trying to test a convolution/deconvolution algorithm you probably want to start with simple test pieces like a delta function at various places on the source image.

My guess is that you have not handled the edge boundary discontinuity correctly. In a random noise image there will not be one.

Regards,
Martin Brown

.