Home Assignment 2 – Blind Super Resolution

Jamal Tannous	208912337
Mohamed Hosh	208340307

1 Theoretical questions

Q1 Solution: **Both the image and the PSF are in the continuous domain that's why we get that the captured images are a product of a convolution:**

$$\boldsymbol{l[n] = \int f(x)p_L(n-x)dx}$$

According to what is given about h[n] we get that:

$$\boldsymbol{h[n] = \int f(x)p_H\left(\frac{n}{\alpha} - x\right)dx}$$

****

Q2 Solution: **Previously it was introduced that** $l[\mathrm{n}] =\downarrow_\alpha (h * k)[\mathrm{n}]$ **, that is why we get:**

$$\boldsymbol{l[n] = \sum_m h[m]\, k[\alpha n - m]}$$

While shifting k we recover the correct samples related to $\boldsymbol{l[n]}$

****

Q3 Solution:

We know that k is a PSF so it is a low pass filter, when moving to the $\frac{1}{\alpha^2}\mathbb{Z}$ **lattice we get that the kernel corresponds to k[n/** $\boldsymbol{\alpha}$**] and so from Q2 we get that**

$$l[n] = \sum_m h\left(\frac{1}{\alpha}m\right) k\left(n - \frac{1}{\alpha}m\right)$$

In the previous question we got that $\boldsymbol{l[n] = \sum_m h[m]\, k[\alpha n - m]}$ **, we will massage this equation and arrive to the desired equation.**

After plugging in the integral we get that:

$$l[n] = \sum_m \left[\int f(x)p_H\left(\frac{1}{\alpha}m - x\right)dx\right] k\left(n - \frac{1}{\alpha}m\right)$$

$$\int f(x)\, p_L(n-x)dx = \sum_m \left[\int f(x) p_H\left(\frac{1}{\alpha}m - x\right)dx\right] k\left(n-\frac{1}{\alpha}m\right)$$

$$(f * p_L)(n) = \sum_{m\in\frac{1}{\alpha^2}\mathbb{Z}} \left(\int f(x)p_H(m-x)dx \cdot k(n-m)\right)$$

$$(f * p_L)(n) = \sum_{m\in\frac{1}{\alpha^2}\mathbb{Z}} (f * p_H)(m)\cdot k(n-m)$$

In the end we get:

$$\boldsymbol{p_L(x) = \sum_m k[m] p_H\left(x-\frac{m}{\alpha}\right)}$$

and so, we get that:

$$\boldsymbol{p_L[n] = \sum_m k[m] p_H\left(n-\frac{m}{\alpha}\right)}$$

****

Q4 Solution:

In the continuous case we substitute the sum with an integral and so we get the expression:

$$\boldsymbol{p_L(x) = \int k_c(t) p_H\left(x-\frac{t}{\alpha}\right)dt}$$

And if we remember that convolution turns to multiplication in the frequency domain then we get that:

$$\boldsymbol{K_c(\xi) = \frac{P_L(\xi)}{P_H(\xi)} = \frac{P_L(\xi)}{P_L(\xi/\alpha)}}$$

So we can look at k_c as a deconvolution with p_H and a convolution with p_L.

****

Q5 Solution:

In the case of p_L=sinc:

If we assume that $k_c(x)$=sinc(x) then $K_c(\xi) = rect(\xi)$ and we get that (Proof in the end)

$$\boldsymbol{K_c(\xi) = rect(\xi) = \frac{rect(\xi)}{rect(\xi/\alpha)}}$$

And in this case the assumption is correct.

In the case of p_L=Isotropic Gaussian (With width σ): (Proof in the end)

Here we get that the division by $P_L(\xi/\alpha)$ amplifies high frequencies and that's why in this case we get that there is a division of two gaussian of different widths and so we get that the optimal $k_c(x)$ is an isotropic Gaussian with width $\sigma\sqrt{1 - 1 / \alpha^2}$, which is narrower than the given PSF.

So the assumption is not correct in this case.

In real life: Typically, we do not get a very straight line of transition around the cutoff frequency so normally the PSF looks more like a Gaussian rather than a sinc.

That is why typically we get a $k_c(x)$ which looks more like a Gaussian.

****

Q6 Solution:

We want to get the optimal k, in other words we want to get the k that gives as the highest sum of probabilities, so we get that the MLE problem is described as follows:

$$\hat{k} = \underset{k}{argmax} \prod_{i=1}^{N} p(q_i|k)$$

$$\hat{k} = \underset{k}{argmax} \prod_{i=1}^{N} \int_{p_{ji}} p(q_i|p_{ji}, k)\, p(p_{ji}) dp_{ji}$$

We know that $\{p_i\}$ are i.i.d

It is given that the noise has an isotropic distribution with variance σ_N^2 so we get:

$$\hat{k} = \underset{k}{argmax} \prod_{i=1}^{N} \int_{p_j} \exp\left(-\frac{||q_i - \downarrow\alpha\,(\mathrm{pj} * \mathrm{k})||^2}{2\sigma_N^2}\right) p(p_j) dp_j$$

We can substitute the integral with the expectation:

$$\hat{k} = \underset{k}{argmax} \prod_{i=1}^{N} E\left(-\frac{||q_i - \downarrow\alpha\,(\mathrm{p} * \mathrm{k})||^2}{2\sigma_N^2}\right)$$

While using the empirical expectation we get:

$$\hat{k} = \underset{k}{argmax} \prod_{i=1}^{N} \frac{1}{N} \sum_{j=1}^{N} (-\frac{||q_i - \downarrow \alpha\,(\text{pj} * \text{k})||^2}{2\sigma_N^2})$$

And so we get that:

$$\boxed{\hat{k} = \underset{k}{argmin} \; -\frac{1}{N} \sum_{i=1}^{N} \log\,[\sum_{j=1}^{N} \left(-\frac{||q_i - \downarrow \alpha\,(\text{pj} * \text{k})||^2}{2\sigma_N^2}\right)]}$$

****

Q7 Solution:

We get that the MAP estimation of k is given by:

$$\hat{k} = \underset{k}{argmax}\, p(k) \prod_{i=1}^{N} p(q_i|k)$$

We know that $\{p_i\}$ are i.i.d

It is given that the noise has an isotropic distribution with variance σ_N^2 so we get:

$$\hat{k} = \underset{k}{argmax}\, D(k) \prod_{i=1}^{N} \int_{p_j} \exp\,(-\frac{||q_i - \downarrow \alpha\,(\text{pj} * \text{k})||^2}{2\sigma_N^2})\, p(p_j) dp_j$$

Now we substitute the integral with the expectation:

$$\hat{k} = \underset{k}{argmax}\, D(k) \prod_{i=1}^{N} E(-\frac{||q_i - \downarrow \alpha\,(\text{p} * \text{k})||^2}{2\sigma_N^2})$$

While using the empirical expectation we get:

$$\hat{k} = \underset{k}{argmax}\; D(k) \prod_{i=1}^{N} \frac{1}{N} \sum_{j=1}^{N} (-\frac{||q_i - \downarrow \alpha\,(\text{pj} * \text{k})||^2}{2\sigma_N^2})$$

$$\hat{k} = \underset{k}{argmin} - log(D(k)) - \frac{1}{N}\sum_{i=1}^{N} \log \left[\sum_{j=1}^{N} \left(- \frac{||q_i - \downarrow \alpha\, (\mathrm{pj} * \mathrm{k})||^2}{2\sigma_N^2} \right) \right]$$

And here after plugging the distribution of D(k) we get that:

$$\hat{k} = \underset{k}{argmin} - \log \left(\frac{1}{\sigma_D\sqrt{2\pi}} \exp\left(-\frac{||k||^2}{2\sigma_D^2}\right)\right) - \frac{1}{N}\sum_{i=1}^{N} \log \left[\sum_{j=1}^{N} \left(- \frac{||q_i - \downarrow \alpha\, (\mathrm{pj} * \mathrm{k})||^2}{2\sigma_N^2} \right) \right]$$

Denote the constants which are multiplied by k to be C

After getting rid of the constants, we get that:

$$\hat{k} = \underset{k}{argmin} \frac{||Ck||^2}{2} - \frac{1}{N}\sum_{i=1}^{N} \log \left[\sum_{j=1}^{N} \left(- \frac{||q_i - \downarrow \alpha\, (\mathrm{pj} * \mathrm{k})||^2}{2\sigma_N^2} \right) \right]$$

****

Q8 Solution:

The objective function in this case is:

$$f(k) = \frac{||Ck||^2}{2} - \frac{1}{N}\sum_{i=1}^{N} \log \left[\sum_{j=1}^{N} \left(- \frac{||q_i - \downarrow \alpha\, (\mathrm{pj} * \mathrm{k})||^2}{2\sigma_N^2} \right) \right]$$

If we derivate it, we will get:

$$\nabla \mathrm{f(k)} = ||Ck|| - \frac{1}{N}\sum_{i=1}^{N} \sum_{j=1}^{N} \frac{\left(- \frac{2||q_i - \downarrow \alpha\, (\mathrm{pj} * \mathrm{k})||}{2\sigma_N^2} \right)}{\left(- \frac{||q_i - \downarrow \alpha\, (\mathrm{pj} * \mathrm{k})||^2}{2\sigma_N^2} \right)}$$

$$\nabla \mathrm{f(k)} = ||Ck|| - \frac{1}{N}\sum_{i=1}^{N} \sum_{j=1}^{N} \frac{2}{||q_i - \downarrow \alpha\, (\mathrm{pj} * \mathrm{k})||}$$

Suggested algorithm to estimate k:

Here we use the MAP_k algorithm presented in the paper: [Nonparametric Blind Super-Resolution by Tomer Michaeli and Michal Irani]

We will explain the algorithm in the following lines:

Algorithm setup

As the input to the algorithm, we have the low-resolution patches $\{q_i\}$, we also have the high-resolution patches $\{p_i\}$.

We choose an arbitrary constant σ.

Note: **for each high-res patch pj we can denote P_j to be the equivalent to convoluting p_j and then subsampling it by α**

We initialize $\hat{k}$ to be the delta function and set a constant T to be the number of iterations.

Create a Laplacian matrix C which has the dimensions of a patch, so the patch would be smoother toward the edges.

for t=1……T:

Main Loop

Step 1

$p_j^{\alpha} = \downarrow \alpha \left(pj * \hat{k}\right) = P_j\hat{k}$ **# Here we down sample the high-res patches**

Step 2

Find the NNs of the low-resolution patch according to our previous explanation and for the patches in the

For each neighbor of the patch calculate its weight this way:

$$w_{ij} = \frac{exp[||q_i - p_j^{\alpha}||^2/\sigma^2]}{\sum_j exp[||q_i - p_j^{\alpha}||^2/\sigma^2]}$$

Step 3

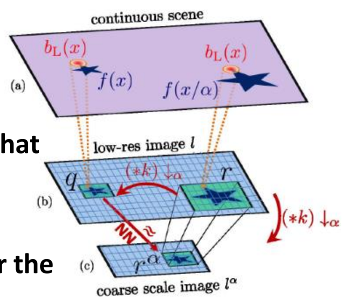
Update $\hat{k}$:

$$\hat{k} = \left(\frac{1}{\sigma^2}\sum_{ij} w_{ij}P_j^T P_j + C^T C\right)^{-1} \sum_{ij} w_{ij}P_j^T q_i$$

The algorithm will yield an estimated a kernel $\hat{k}$ which according to the used paper gives some of the best results regarding the proximity to the real kernel.

Note, finding the NNs of a low-res patch:

Here we use the figure in the presented paper:



We use the kernel $\hat{k}$ to down sample the patches that correspond to the big star in low-res image l (patch r in the image) and then we find the NNs for the q patch in the coarse scale image l^a, and for those NNs we conclude that the large patches from l (their sources) are candidates to be the "parents" of q.

Another idea for iterative algorithm (but it is not the one we used):Gradient Descent

We set a step size η and start from a certain $k^{(0)}$=delta and then get the following algorithm:

$$k^{(t+1)} = k^{(t)} - \eta \nabla f(k)$$

****

Q9 Solution:

We know that $p_H(x) = \alpha^2 p_L(\alpha x)$ and in other words $p_L(x) = \frac{1}{\alpha^2} p_H\left(\frac{x}{\alpha}\right)$

And so, we get that:

$$z[n] = \int f\left(\frac{x}{\alpha}\right) p_L(n-x)\, dx$$

$$z[n] = \frac{1}{\alpha} \int f\left(\frac{x}{\alpha}\right) p_H\left(\frac{n-x}{\alpha}\right) dx$$

****

Q10 Solution:

From the previous question we got that $z[n] = \frac{1}{\alpha^2} \int f\left(\frac{x}{\alpha}\right) p_H\left(\frac{n-x}{\alpha}\right) dx$

If substitute $\frac{x}{\alpha}$ with t then we get that dt=$\frac{dx}{\alpha}$ and so we get that

$$z[n] = \int f(t) p_H\left(\frac{n}{\alpha} - t\right) dx$$

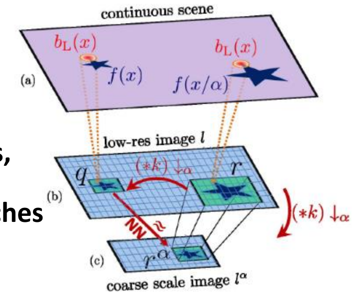
And here we reach the same result from question 2 where we got that

$$l[n] = \downarrow \alpha\, (\mathbf{z}[\mathbf{n}] * \mathbf{k})$$

■

****

Q11 Solution:



Here we also used the presented paper and especially this figure:

Here we can see that if there are recurrences of patches across scales, then the NNs of the low-resolution patch from f(x) would be the patches corresponding to the similar patch in f(x/a).

So, if we denote the low-res patches to be q[n] and the high-res patches to be p[n] then we get that:

$$q[n] = \int f(x) p_L(n-x)dx$$

$$p[n] = \int f\left(\frac{x}{\alpha}\right) p_L(n-x)dx$$

And so, we would use the algorithm presented in question 8 with the low-res patches q[n] and the high-res patches p[n], and so after the subsampling of patches p[n] we would get something similar to that of image (c) in the figure and we would be able to extract the NNs of each low-res patch and perform the algorithm as normal.

****

Q12 Solution:

We want to use a wiener filter to recover the image, but first we want to prepare some data:

We have the low-res image l and the true kernel k.

First, we want to up-sample the image l by a factor α and then store the up-sampling result in a matrix named p.

Then we want to pass p through a Wiener Filter with the kernel k and a constant c, and the output of the Wiener Filter would be the high-resolution image h.

So, in short:

Let us denote the Fourier transforms to be L, K, P and H which correspond to the spatial signals with lower case letters.

p[n]=$\uparrow \alpha\,(l)$[n]

H[ω]=$L[\omega] \cdot \frac{\overline{H(\omega)}}{|H(\omega)|^2+c}$

$h[n] = |F^{-1}\{\text{H}[\omega]\}|$

Important Note:

Another way to recover the image is a normal Minimum problem:

We know k and l, and so we predict that the optimal h image would be:

$$\hat{h} = \underset{\mathbf{h}}{argmin} \left|\left| l - \downarrow \alpha \left(\mathrm{h} * \mathrm{k} \right) \right|\right|^2$$

2 Practical questions

A brief explanation discussing the results:

We got some very interesting results:

It is clear to see that the recovering the high-resolution images using the estimated kernel yield much better results than those from using the true kernels.

We notice from the results that the PSF is the wrong blurring kernel k to use in SR algorithms.

A much better way is to use the feature of recurrences between patches and then try to estimate the blur kernel k using those recurrences, there are multiple ways to estimate the blur kernel k, we chose to implement the algorithm of MAP_k from the paper Nonparametric Blind Super-Resolution by Tomer Michaeli and Michal Irani

Proof for Q5 the p_L=sinc case:

We work at the Fourier domain so the sinc will become rect function:

$P_L(w)$=rect(w)
$P_L(w/\alpha)$=rect(w/α)

$$K(w) = \frac{\mathrm{rect}(w)}{\mathrm{rect}(w/\alpha)} = \frac{\begin{cases} 0\,, & \left|w > \frac{1}{2}\right| \\ \frac{1}{2}\,, & \left|w = \frac{1}{2}\right| \\ 1\,, & \left|w < \frac{1}{2}\right| \end{cases}}{\begin{cases} 0\,, & \left|w > \frac{a}{2}\right| \\ \frac{1}{2}\,, & \left|w = \frac{a}{2}\right| \\ 1\,, & \left|w < \frac{a}{2}\right| \end{cases}}$$

If we look at the values of w, the values of the rect in the numerator have a smaller domain than that of the rect in the denominator and so the numerator reaches zero before the denominator and the division is valid.

If we check those values we get the function which is yielded is also a rect(w).

■

Proof for Q5 the p_L=Isotropic Gaussian (With width σ):

Using the Fourier transform "tricks" we get that:

$P_L(w)=\frac{1}{(2\pi)^{\frac{d-1}{2}}} e^{-2\sigma^2\pi^2||w||^2}$

And so, we get that

$$K(w)= \frac{P_L(w)}{P_L\left(\frac{w}{\alpha}\right)} = e^{-2\sigma^2\pi^2||w||^2\left(1-\frac{1}{a^2}\right)}$$

■

And so, we reached the end of the homework…

We hope that you have enjoyed reading it 😊