Assignment NO.1 Solutions

Digital Image Processing | Fall 1400 | Dr.Mohammadi

Teacher Assistant : Ramin Kamali

Student name : **Amin Fathi**

Student id : **400722102**

Problem 1.a

زوم اپتیکال به معنای حرکت فیزیکی لنز دوربین است که با تغییر فاصله کانونی، نزدیکی ظاهری سوژه تصویر را تغییر می دهد. برای بزرگنمایی، لنز از حسگر تصویر دور می‌شود و صحنه بزرگ‌نمایی می‌شود در حالی که زوم دیجیتال بدون استفاده از اپتیک‌های لنز است و در درون دوربین انجام می‌شود. برای این کار دوربین عکس را می‌بُرد تا مرکز عکس باقی بماند. سپس مرکز عکس که باقی مانده است با اضافه کردن پیکسل به آن بزرگ می‌شود تا کل قاب را پر کند. در حین این روند کیفیت عکس ممکن است به‌شدت کاهش یابد.

ویژگی ها	Optical Zoom	Digital Zoom
رزولوشن	زوم اپتیکال به لنز اجازه می‌دهد تا منشورهای درونی را تغییر دهد، که نحوه انتقال نور را تغییر می‌دهد ، در واقع تصویر را قبل از گرفتن یک عکس فوری بزرگ می کند، بنابراین شما همچنان می توانید وضوح بالایی را از راه دور دریافت کنید. می توان از طریق تصاویری در دوردست با وضوح k۴ ثبت کرد.	زوم دیجیتال در واقع حسگرهای دوربین را از طریق اپتیک به تصویر نزدیک نمی‌کند، اما در عوض، فقط روی تصویر زوم می‌کند تا هر سوژه را بزرگ تر کند که این اقدام باعث کاهش وضوح تصویر می شود. چنانچه از این روش برای ثبت تصویری در دوردست با استفاده از دوربینی با رزولوشن p1080 استفاده شود ، تصویر نهایی پس از زوم کیفیتی در حد p480 (یا حتی کمتر) خواهد داشت
کاربرد	لنز های دروبین اپتیکال برای ثبت تصاویری با زوم بیشتر از ۲ بیسار حجیم هستند و نیاز به تجهزات اضافی دارند	زوم دیجیتال به هیچ آموزش و تجهیزاتی درباره لنز ها نیاز ندارد .
وضوح	زوم اپتیکال وضوح تصویر را تغییر نمی دهد	زوم دیجیتال وضوح عکس را تغییر می دهد .

References:

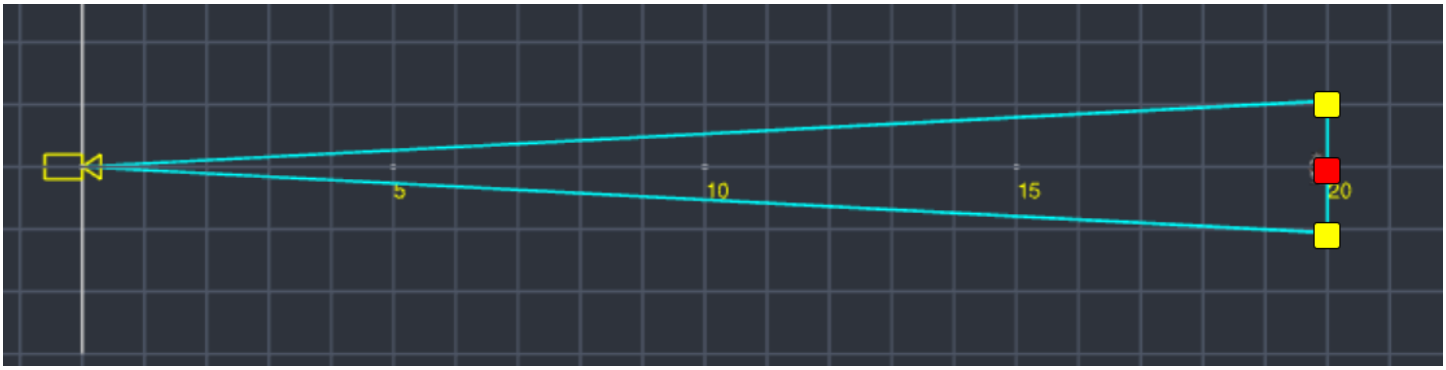
What’s the Difference Between Optical Zoom and Digital Zoom? - 2021 - MasterClass

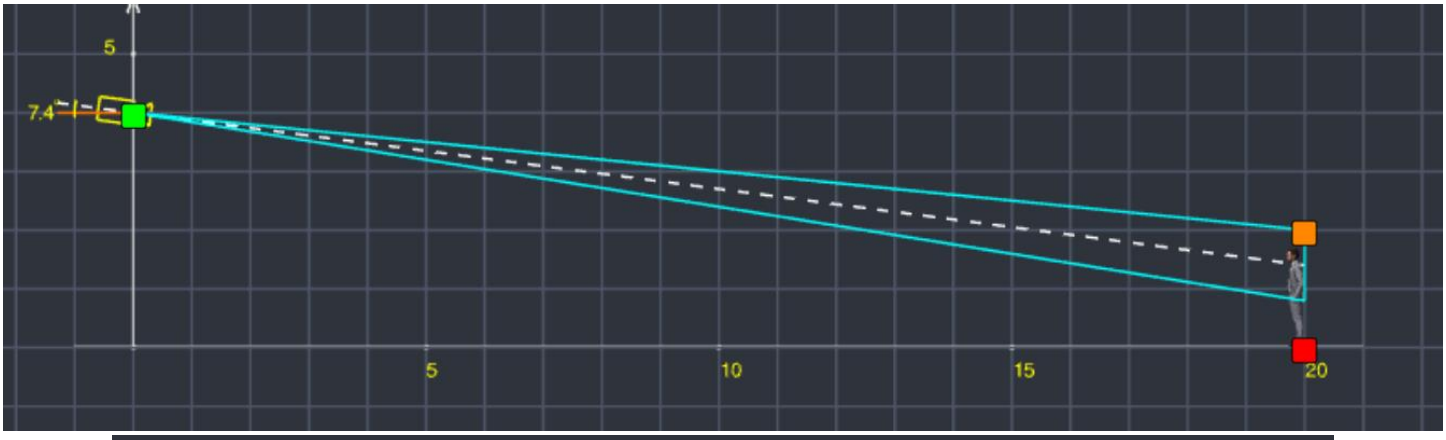
Optical Zoom Vs Digital Zoom: What's The Difference? (pocketphotography.com)

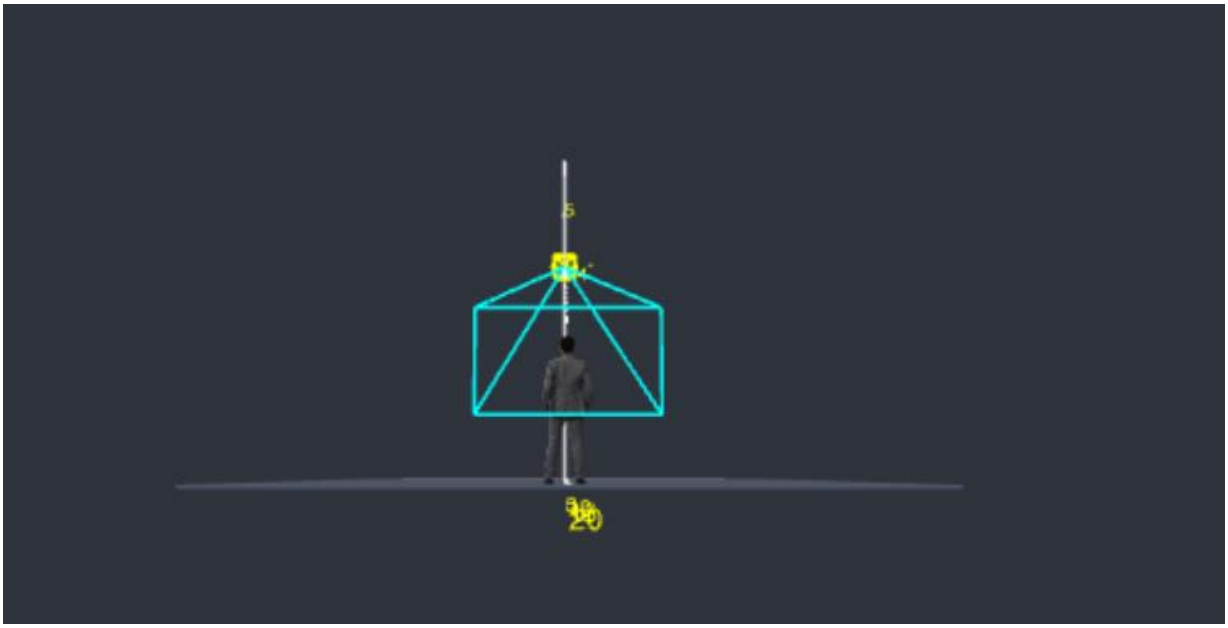
زوم اپتیکال | ⋆ تفاوت زوم دیجیتال و اپتیکال دوربین دیجیتال-مجله نورنگار (noornegar.com)

Problem 1.b

مقدار FOV Width حدود ۲٫۲ متر و زاویه افقی برابر حدود ۶/۳ درجه می باشد







Problem 2.a

```python
def calc_hist(image):
  '''
  calculate image histogram
  don't use libraries
  input(s):
    image (ndarray): input image
  output(s):
    hist (ndarray): computed input image histogram
  '''
  hist = np.zeros(256,dtype=int)
  ############
  # Your code
  # Start
  (n,m) = (image.shape)
  k = 0
  for k in range(0, 256):
    count1 = 0

        # loops to traverse each pixel in
        # the image
    for i in range(n):
      for j in range(m):
        if image[i, j]== k:
          count1 = count1+1
    hist[k] = count1

  # End
  return hist
```

در این تمرین ، هیستوگرام شکل را مصاحبه میکنیم . ابتدا یک آرایه ۲۵۶ عضوری با مقدار اولیه ۰ به نام hist تشکیل می دهیم . سپس به تعداد تکرار مقدار k در تصویر ، مقدار k ام آرایه hist را یک واحد افزایش میدهیم و در نهایت آرایه hist را به عنوان خروجی برمیگردانیم.

مشاهده میشود که هیستوگرام به دست آمده از توابع opencv و هیستوگرام به دست آمده در روش دستی یکسان هستند.

Problem 2.b

```python
def stretch_hist(image):
  '''
  don't use libraries
  input(s):
    image (ndarray): input image
  output(s):
    output_image (ndarray): enhanced image with histogram stretching
  '''
  # output_image = image.copy()
  ############
  # Your code
  # Start
  constant = (255-0)/(image.max()-image.min())
  image2 = image - image.min()
  output_image = image2* constant

  # End
  return output_image
```

در این بخش با توجه به فرمول تدریس شده و مربوط به stretch کردن هیستوگرام ، این اقدام را انجام میدهیم

Problem 2.c

```
def modified_stretch_hist(image):
  ############
  # Your code
  # Start
  i , j = image.shape
  count1 = 0
  count2 = 0
  hist =calc_hist(image)
  for a in range( 10 , 256):
   if hist[a] != 0:
     count1 = a
     break
  for b in range( 245 , 10 , -1):
    if hist[b] != 0:
      count2 = b
      break

  for k2 in range(0 ,i):
    for k3 in range(0 , j):
        if image[k2][k3] <= 10 :
          image[k2][k3] = count1
        elif image[k2][k3] >= 245 :
          image[k2][k3] = count2
  output_image = stretch_hist(image)

  ############
  return output_image
```

در این سوال ابتدا داده های پرت هیستگورام را حذف می کنیم و سپس همانند قسمت قبلی stretch میکنیم ، برای حذف قسمت داده های پرت ابتدا مقدار اولین k ای که داده غیرپرتِ غیرِ صفر دارد را به متغیر count۱ پاس میدهیم ، در واقعا از k شماره ۱۰ به بعد به جست و جوی اولین مقدار غیر صفر میگردیم . همین کار را برای انتهای تصویر و مقدار count2 هم انجام میدهیم . سپس مقدار پیکسل هایی که در فاصله ۱۰ k اول یا آخر بودند را به ترتیب به count1 و count2 نگاشت می کنیم .

Problem 2.d

در این قسمت ابتدا تابع تبدیل را تعریف کرده و به دست می آوریم

```python
def find_T(image):
  i1 , i2  = image.shape
  hist = calc_hist(image)
  hist2 = np.zeros(256,dtype=int)
  hist3 = np.zeros(256,dtype=int)
  hist2[0] = hist[0]
  for k in range(1 , 256):
      hist2[k] = hist[k] + hist2[k-1]

  a = int(hist2[255])
  for k in range(0 , 256):
      hist3[k] = round(hist2[k] * (255/a))
  return hist3
```

برای اینکار ابتدا مجموع آرایه های صفرم تا k-1 ام هیستوگرام را در خانه k ام آرایه ۲۵۶ عضوی دیگری (hist2) ذخیره می کینم ، سپس مقدار خانه k ام hist2 را در مقدار l-1 یا همان ۲۵۵ ضرب و بر مقدار آخرین خانه آرایه hist 2 تقسیم میکنیم ، خروجی همان تابع تبدیل T است.

```python
def equalize_hist(image):
  '''
  don't use libraries
  input(s):
    image (ndarray): input image
  output(s):
    output_image (ndarray): enhanced image with histogram equalization
  '''

  # pixelMap = image.load()

  ############
  # Your code
  # Start
  i1 , i2  = image.shape
  T = find_T(image)




  for j1 in range(0 , i1 ):
    for  j2 in range(0 ,i2 ):
      for k in range(0 ,256):
        if image[j1,j2] == k:
          image[j1,j2]= T[k]

  output_image = image
  # End
  return output_image
```

برای اکوالایز کردن هم ، مقدار تابع T را فراخوانده و مقادیر پیکسل های تصویر را به روی آن نگاشت میکنیم .

Problem 3

```python
def hist_matching(src_image,ref_image):
  '''
  don't use libraries
  input(s):
    src_image (ndarray): source image
    ref_image (ndarray): reference image
  output(s):
    output_image (ndarray): transformation of source image so that its histogram matches histogram of refrence image
  '''
  i1 , i2 , i3 = src_image.shape
  j1 , j2 , j3 = ref_image.shape
  ############
  # Your code
  # Start
  # Split
  redsrc = src_image[:, :, 0]
  greensrc = src_image[:, :, 1]
  bluesrc = src_image[:, :, 2]
  redref = ref_image[:, :, 0]
  greenref = ref_image[:, :, 1]
  blueref = ref_image[:, :, 2]
  # determine histograms
  hist_redsrc = find_T(redsrc)
  hist_greensrc = find_T(greensrc)
  hist_bluesrc = find_T(
  hist_redref = find_T(redref)
  hist_greenref = find_T(greenref)
  hist_blueref = find_T(blueref)
```

Loading...

برای حل این سوال ابتدا تصاویر را به کانال های رنگی BGR تبدیل می کنیم و سپس برای هر کدام از آن ها تابع تبدیل T را به دست می آوریم ،

```python
    # for red

    for k in range(0 ,256):
      a = hist_redsrc[k]
      for k2 in range(0 ,256):
        if hist_redref[k2] == a :
          for n1 in range(0 , i1):
              for n2 in range(0 , i2):
                if redsrc[n1,n2] == k:
                    redsrc[n1,n2] = k2
```

سپس برای هر کدام از کانال های تصویر src ابتدا به ازای k های مختلف ،مقدار تابع تبدیل را پیدا کرده و سپس پیدا میکنیم که متناسب با آن مقدار (a) چه k2 ای در تصوری ref نگاشت میشود و مقدار تصویر را معادل سازی میکنیم

```
(-0.5, 1199.5, 799.5, -0.5)
```