

به نام خدا



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

دانشکده مهندسی کامپیوتر

سیستم‌های چندرسانه‌ای نیمسال دوم ۱۴۰۱-۱۴۰۲

تمرین عملی dithering

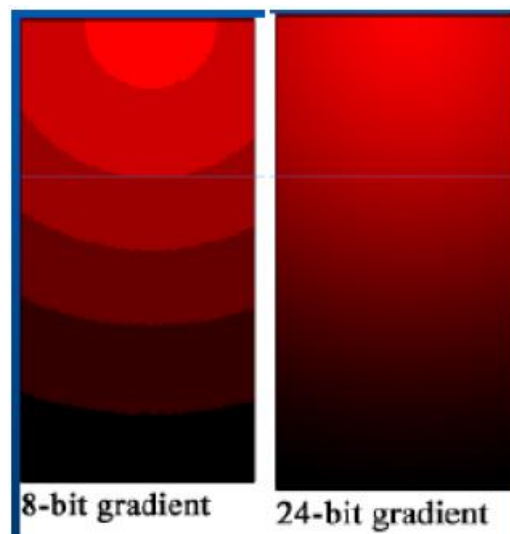
مهلت تحویل:

۲۲ اردیبهشت ۱۴۰۲ ساعت ۵۹:۲۳

بخش اول – توضیحات:

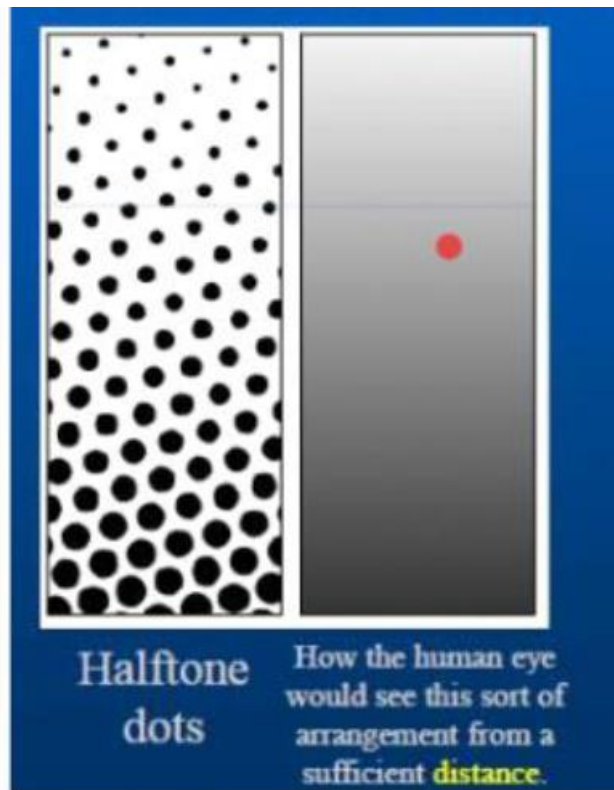
سلام بچه‌ها. ایشالا که ایام به کامتون باشه (☺). همونطور که می‌دونید (یا شایدم نمیدونید الان می‌خواید آشنا شید!)، ما به طور کلی برای نمایش پیکسل‌های تصاویر از ۲۴ بیت یا ۳ بایت استفاده می‌کنیم که ۸ بیت برای هر کدوم از سه رنگ قرمز، آبی و سبز استفاده میشه. بدیهیه که مقدار فضای storage مورد نیاز برای ذخیره اطلاعات یک پیکسل برابر 256^3 بیت (تقریباً ۱۶ میلیون) میشه که مقدار زیادیه! سپس بحث گسسته‌سازی^۱ رو مطرح کردن که یکی از کارهایی که انجام شد این بود که بجای ۲۴ بیت کلا از ۸ بیت برای نشون دادن RGB استفاده کردند. ۳ بیت برای قرمز، ۳ بیت برای سبز و ۲ بیت باقی برای رنگ آبی.

اما این کم کردن تعداد بیت مشکلاتی را هم بوجود آورد. یکی از اونها، تشکیل شدن مرزهایی بین رنگهای مختلف در عکس جدید شد. به این مورد، اثر مرزی^۲ می‌گن و نمونش رو در عکس زیر مشاهده می‌کنید:



اینجا بود که یکی از راهکارا مطرح شد: دیترینگ که با استفاده از تکنیک halftoning میاد و مثلاً در یک عکس grayscale با اینکه یک بیت برای نمایش سیاه یا سفیده جوری نمایش میده که طیف خاکستری انگار وجود داره که با تغییر سایز اون نقطه‌ها مثلاً حالت خاکستری به ما نشون داده میشه. مثالش را در زیر می‌بینیم:

^۱ quantization
^۲ Banding effect



خب حالا که فهمیدیم halftone چیه حدودا، چطوری استفادش کنیم؟ ما یک ماتریس دیترینگ $n \times n$ داریم که معمولا 2×2 یا 4×4 است که خب داخل این ماتریس بر این اساس که n چنده، اعداد از 0 تا $n^2 - 1$ چیده میشن (نحوه چیدنش رو خودتون تحقیق کنید!) که نهایتا میشه $n^2 + 1$ رنگ درست کرد. همونطور که در اسلایدها هم می بینین، یک الگوریتم ساده برای دیترینگ وجود داره که اونو روی تصویر سیاه سفید می لغزونیم و یک black-white thresholding خواهیم داشت که مقدار روشنایی پیکسل عکس از مقدار متناظر آن در ماتریس دیترینگ ما بزرگتر باشه به سفید و در غیر این صورت به سیاه map میشه. (این الگوریتم صفحه ۶۶ کتاب مرجع موجوده)

مضاف بر روش بالا که بهش ordered-dithering میگن، روش بهتری هست که به الگوریتم Floyd-Steinberg Dithering معروفه. این الگوریتم بر اساس پراکندگی خطا عمل میکنه. شبه کدش رو تو تصویر زیر می بینید: (پس از تحقیق روی این الگوریتم به سوالات تشریحی ۳ و ۴ پاسخ

بدید)

دیترینگ بر روی تصاویر رنگی

Floyd–Steinberg Algorithm

- ◆ **Distribute the quantization residual to neighboring pixels that have not yet been processed.**
- ◆ **Pseudocode:**

```
for each y from top to bottom
  for each x from left to right
    oldpixel := pixel[x][y]
    newpixel := find_closest_palette_color(oldpixel)
    pixel[x][y] := newpixel
    quant_error := oldpixel - newpixel
    pixel[x+1][y] := pixel[x+1][y] + 7/16 * quant_error
    pixel[x-1][y+1] := pixel[x-1][y+1] + 3/16 * quant_error
    pixel[x][y+1] := pixel[x][y+1] + 5/16 * quant_error
    pixel[x+1][y+1] := pixel[x+1][y+1] + 1/16 * quant_error
```

1	0	0	0
16	0	0	7
	3	5	1

Distribution matrix

بخش دوم – پیاده‌سازی:

خب حالا، یسری پیاده‌سازی می‌خوایم ازتون برامون انجام بدید:

۲-۱: الگوریتم ordered dither را روی عکس با نام s.jpg که بارگذاری شده است پیاده‌سازی

کنید. یعنی ابتدا تصویر را grayscale.png و بعد با نام grayscale.png ذخیره‌اش کنید. در نهایت

دیترینگ رو روش انجام بدید و حاصل را با نام o_dithered.png ذخیره کنید. (سایز ماتریس

دیترینگ رو ۴*۴ در نظر بگیرید)

۲-۲: الگوریتم Floyd–Steinberg را روی عکس دوم که بارگذاری شده است پیاده‌سازی کنید و

نتیجه را با نام FS-dithered.png ذخیره کنید. (عکس دوم با نام

1665__girl_with_a_pearl_earring_sm.jpg ذخیره شده است).

بخش سوم – سوالات تشریحی:

۱. افزایش یا کاهش اندازه ماتریس دیتترینگ توی ordered dithering چه تاثیراتی

میذاره؟- سایز این ماتریس میتونه 3×3 باشه؟ اگه آره چرا و اگه نه چرا.

۲. اگر m سایز ماتریس باشد، یک تابع یا شبه کد بنویسید که ماتریس دیتترینگ را بسازد و

خروجی دهد.

۳. در ماتریس توزیع در الگوریتم Floyd-Steinberg دلیل صفر بودن ۵ درایه آن چیست؟

۴. درباره کاری که تابع `find_closest_palette_color` باید بکنه توضیح بدید.

بخش چهارم - نکات:

- پروژه تک نفره و زبان برنامه نویسی پایتونه.
- در صورت مشاهده تقلب، نمره طرفین صفر منظور میشه.
- سعی کنید زودتر شروع کنید تمرینا رو حل کنید که سر ددلاینای نخواید بزنید 😊
- برای این پروژه، استفاده از کتابخانه برای grayscale کردن مجاز نیست و خودتون دستی باید بزنید اما برای باز کردن عکس و عوض کردن RGB و گرفتن مقادیر اون، طبیعتا باید از یک کتابخانه‌ای استفاده کنید.
- حداقل یک خط برای توابع یا قسمتهای مهم کدتون کامنت بذارید.
- برای موارد مشکوک احتمالا تدریس‌یارا ازتون تحویل بگیرن پس آمادگی داشته باشید:))
- نتایج و کدها رو با نام `MM-dithering-StudentNumber.zip` تو سایت درس قرار بدید.

با تشکر

تیم تدریس‌یاری