دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

تمرین سری شش

درس بینایی ماشین

فرهاد دلیرانی

۹۶۱۳۱۱۲۵

dalirani@aut.ac.ir

dalirani.1373@gmail.com

فهرست

ابزارهای استفاده شده

زبان برنامه نویسی: پایتون 3.6

محیط توسعه: PyCharm

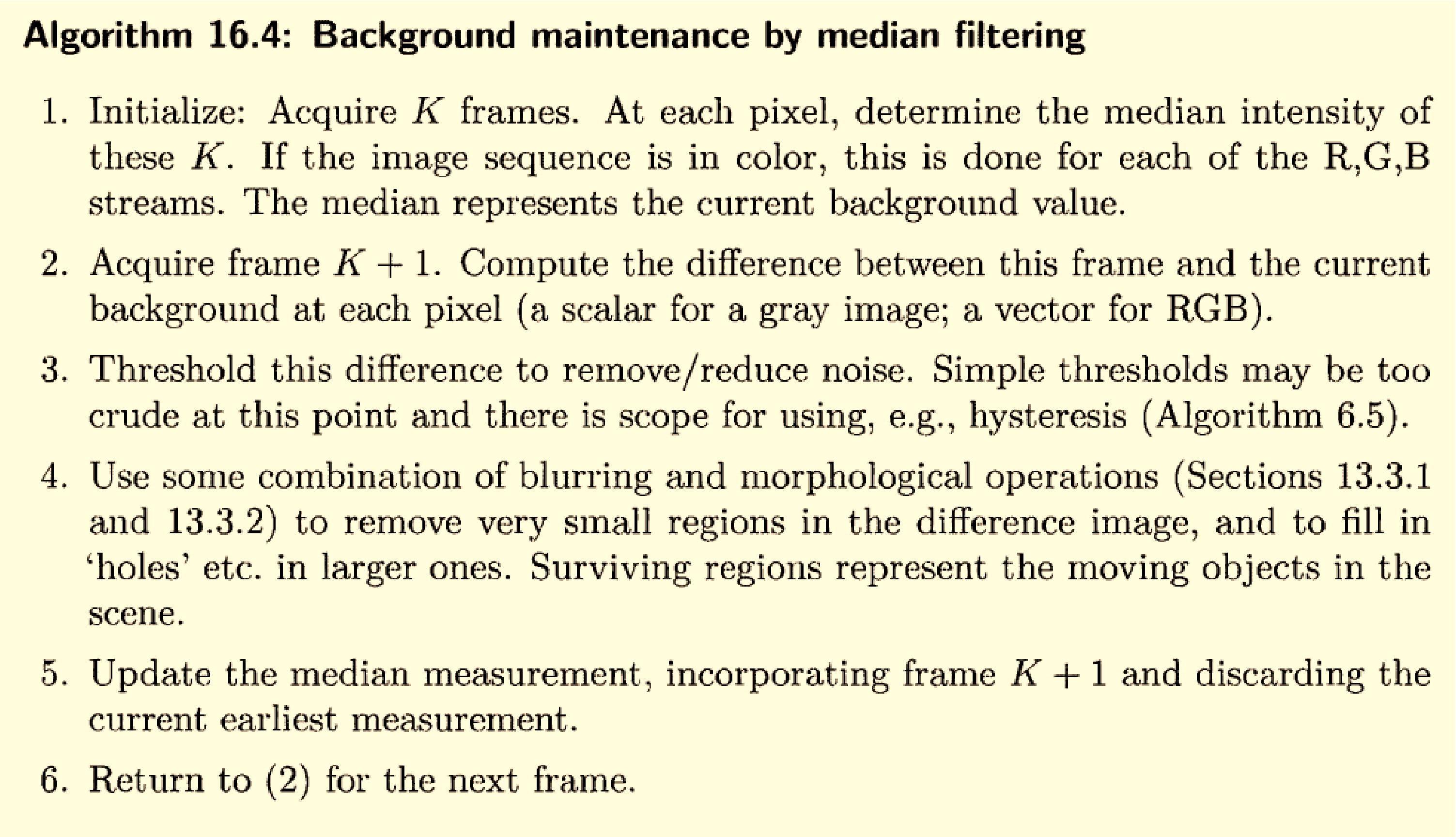
سیستم عامل: Windows 10

تمرین ۱

کدهای این سوال در فایل‌های read_a_image_series.py, mixture_of_gaussians.py و median_background_subtraction.py قرار دارد.

خروجی‌های برنامه در mog_output و median_output_v1 قرار دارد.

روش Background Subtraction با استفاده از روش Median در کتاب Sonka به صورت زیر آورده شده است:

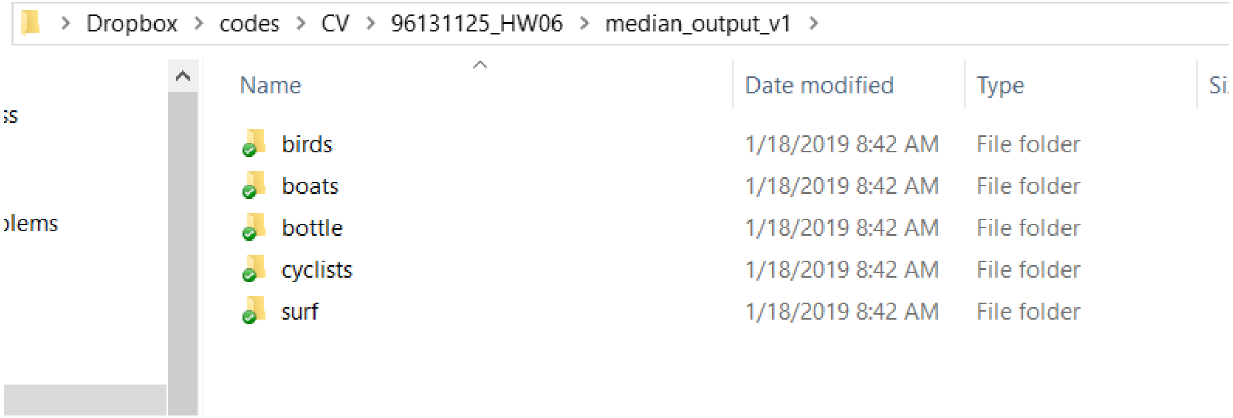


Algorithm 16.4: Background maintenance by median filtering

1. Initialize: Acquire K frames. At each pixel, determine the median intensity of these K. If the image sequence is in color, this is done for each of the R,G,B streams. The median represents the current background value.
2. Acquire frame $K + 1$. Compute the difference between this frame and the current background at each pixel (a scalar for a gray image; a vector for RGB).
3. Threshold this difference to remove/reduce noise. Simple thresholds may be too crude at this point and there is scope for using, e.g., hysteresis (Algorithm 6.5).
4. Use some combination of blurring and morphological operations (Sections 13.3.1 and 13.3.2) to remove very small regions in the difference image, and to fill in 'holes' etc. in larger ones. Surviving regions represent the moving objects in the scene.
5. Update the median measurement, incorporating frame $K + 1$ and discarding the current earliest measurement.
6. Return to (2) for the next frame.

طبق این روش با آمدن فریم K+1، ابتدا از K فریم قبلی میانه می‌گیریم، میانه k فریم قبلی معیار مناسب و رباستی به عنوان پس زمینه رفرنس است.سپس فریم K+1 و میانه را از هم کم می‌کنیم، سپس از یک Threshold استفاده می‌کنیم تا نقاط را به دو دسته تقسیم کنیم، دسته یک نقاطی است که اختلاف در آن‌ها کم بوده است و نشان دهنده‌ی پس‌زمینه است و نقاطی که در آن‌ها اختلاف زیاد بوده است که نشان دهنده‌ی نقاط غیر پس‌زمینه است.

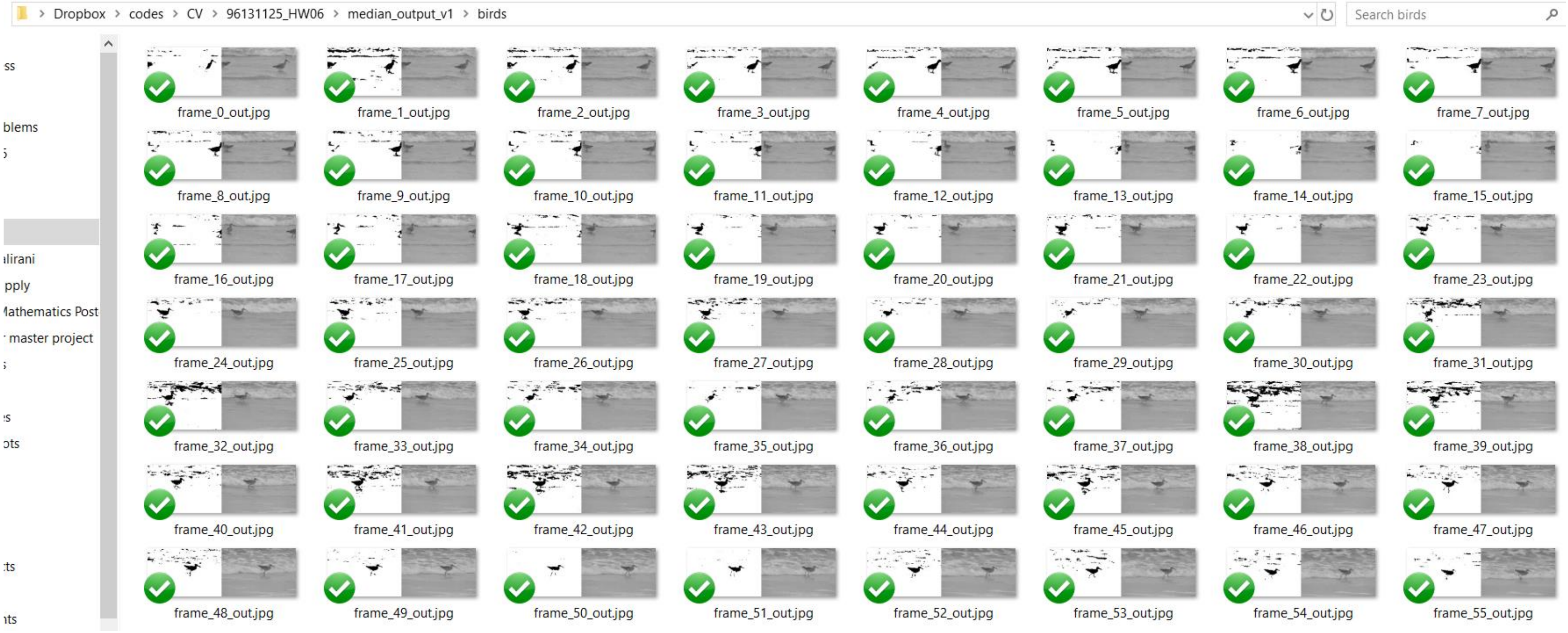
در پیاده‌سازی نیز مانند آنچه در بالا شرح داده شد عمل شده است. ابتدا میانه k فریم محاسبه می‌شود. سپس تفریق فریم k+1 و میانه محاسبه می‌شود. سپس تصویر با استفاده از حد میانه ۱۰۰ باینری می‌شود. در آزمایش‌هایی که انجام شد مقادیر مختلف k و حد آستانه مورد آزمایش قرار گرفت که با توجه به آزمایش‌هایی که انجام شد مقدار 15 برای k و مقدار 100 برای حد آستانه انتخاب شد. خروجی‌های این کد در پوشه median_output_v1 قرار دارد.



Dropbox › codes › CV › 96131125_HW06 › median_output_v1 ›

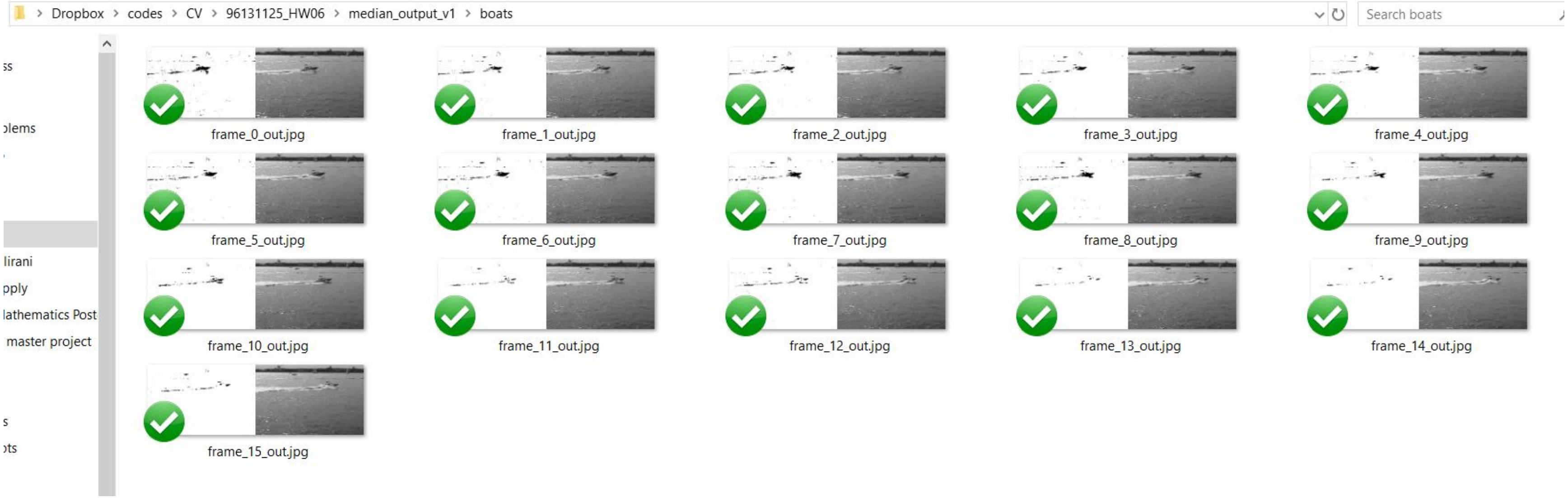
Name	Date modified	Type
birds	1/18/2019 8:42 AM	File folder
boats	1/18/2019 8:42 AM	File folder
bottle	1/18/2019 8:42 AM	File folder
cyclists	1/18/2019 8:42 AM	File folder
surf	1/18/2019 8:42 AM	File folder

در هر پوشه داخلی، خروجی برای هر سری قرار داده می‌شود. در زیر تصویری از محتوای پوشه‌ها آورده شده است، برای دیدن خروجی‌های به پوشه‌ها مراجعه شود.



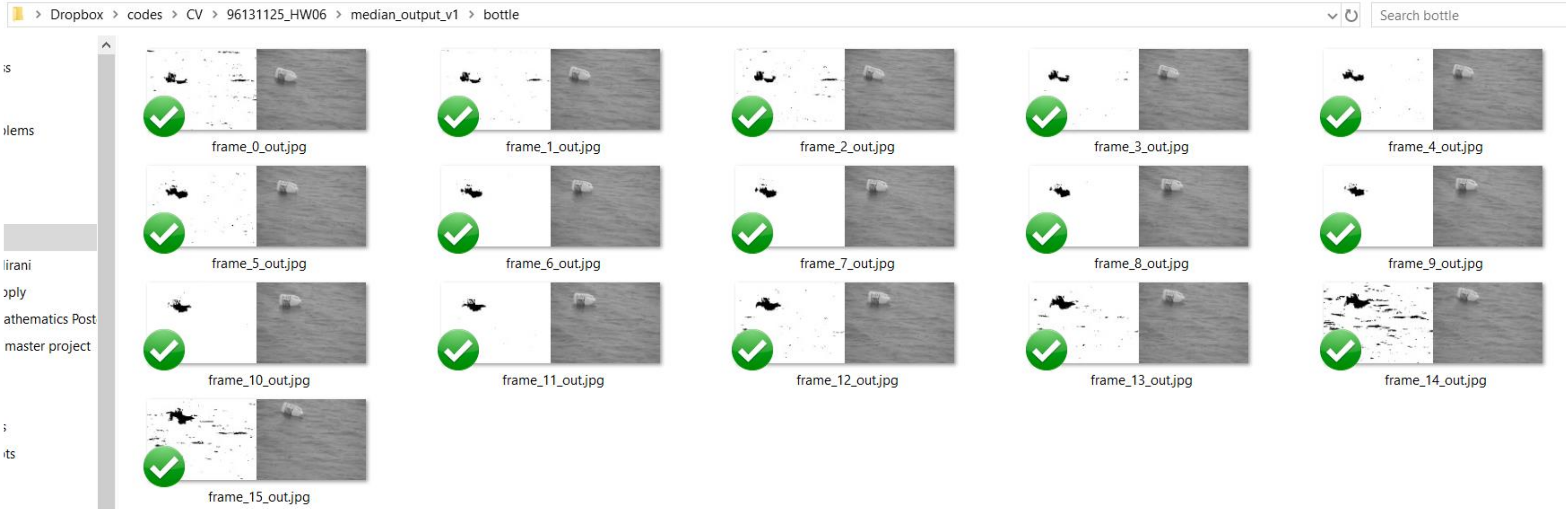
Dropbox > codes > CV > 96131125_HW06 > median_output_v1 > birds

frame_0_out.jpg frame_1_out.jpg frame_2_out.jpg frame_3_out.jpg frame_4_out.jpg frame_5_out.jpg frame_6_out.jpg frame_7_out.jpg
frame_8_out.jpg frame_9_out.jpg frame_10_out.jpg frame_11_out.jpg frame_12_out.jpg frame_13_out.jpg frame_14_out.jpg frame_15_out.jpg
frame_16_out.jpg frame_17_out.jpg frame_18_out.jpg frame_19_out.jpg frame_20_out.jpg frame_21_out.jpg frame_22_out.jpg frame_23_out.jpg
frame_24_out.jpg frame_25_out.jpg frame_26_out.jpg frame_27_out.jpg frame_28_out.jpg frame_29_out.jpg frame_30_out.jpg frame_31_out.jpg
frame_32_out.jpg frame_33_out.jpg frame_34_out.jpg frame_35_out.jpg frame_36_out.jpg frame_37_out.jpg frame_38_out.jpg frame_39_out.jpg
frame_40_out.jpg frame_41_out.jpg frame_42_out.jpg frame_43_out.jpg frame_44_out.jpg frame_45_out.jpg frame_46_out.jpg frame_47_out.jpg
frame_48_out.jpg frame_49_out.jpg frame_50_out.jpg frame_51_out.jpg frame_52_out.jpg frame_53_out.jpg frame_54_out.jpg frame_55_out.jpg



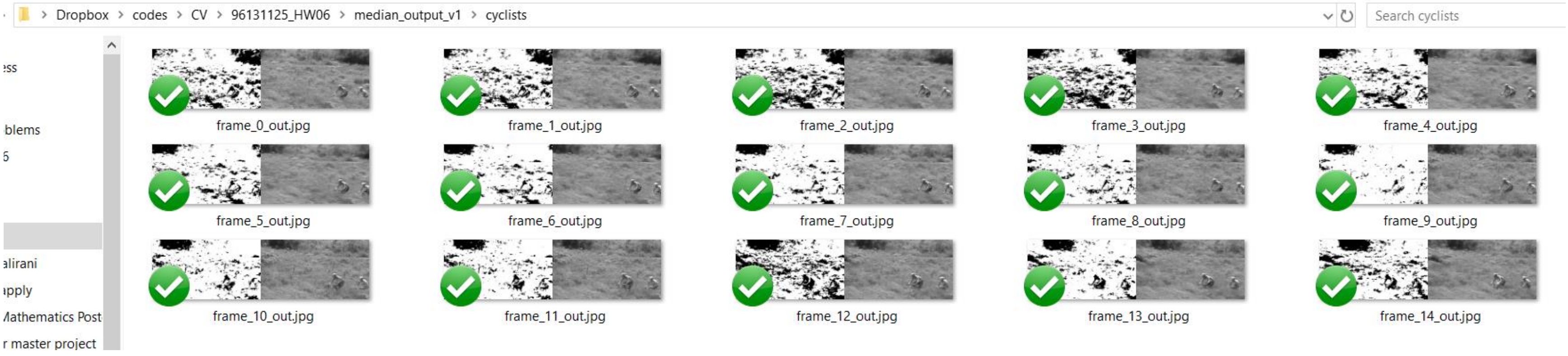
Dropbox > codes > CV > 96131125_HW06 > median_output_v1 > boats

frame_0_out.jpg frame_1_out.jpg frame_2_out.jpg frame_3_out.jpg frame_4_out.jpg
frame_5_out.jpg frame_6_out.jpg frame_7_out.jpg frame_8_out.jpg frame_9_out.jpg
frame_10_out.jpg frame_11_out.jpg frame_12_out.jpg frame_13_out.jpg frame_14_out.jpg
frame_15_out.jpg



Dropbox > codes > CV > 96131125_HW06 > median_output_v1 > bottle

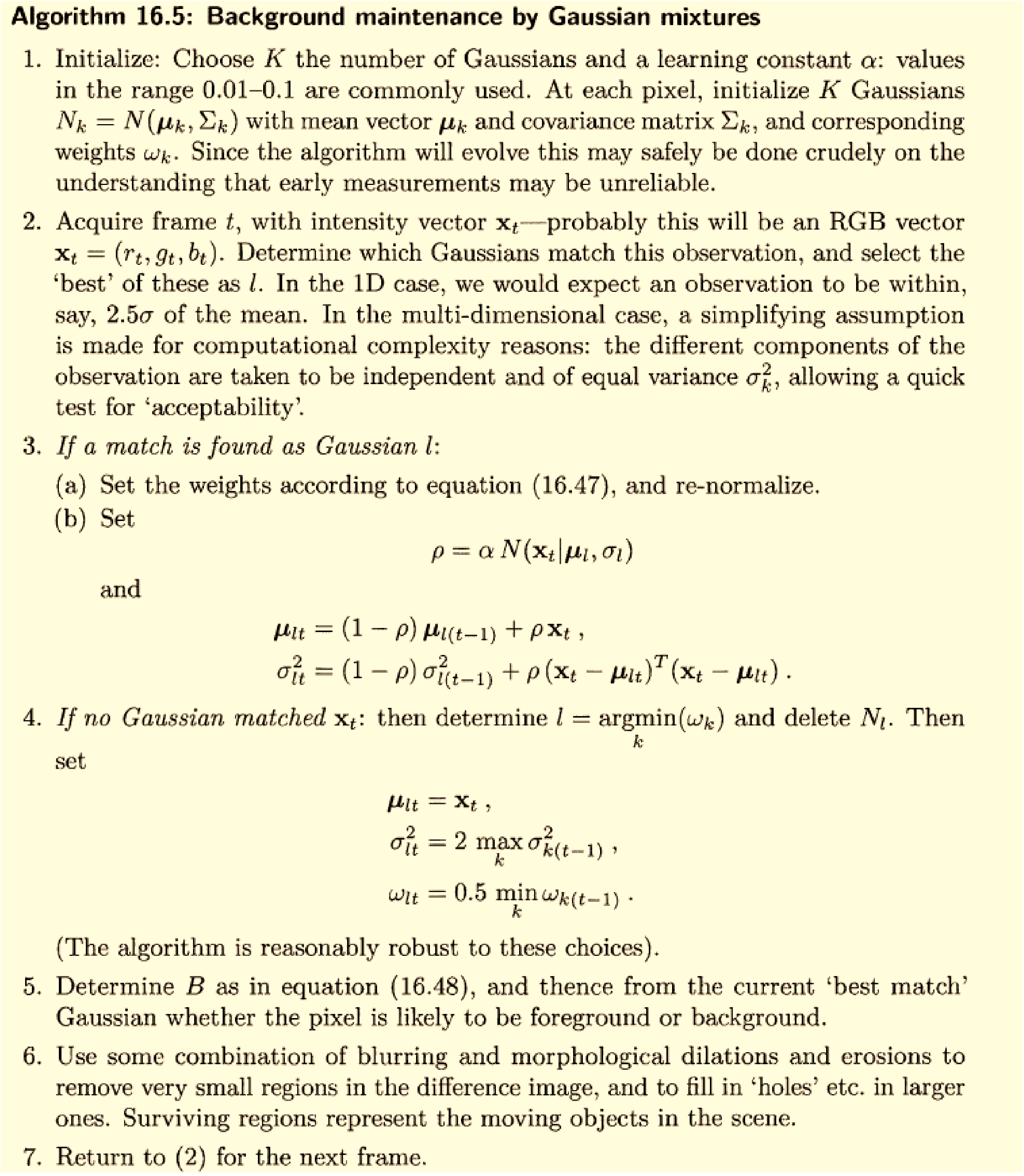
frame_0_out.jpg frame_1_out.jpg frame_2_out.jpg frame_3_out.jpg frame_4_out.jpg
frame_5_out.jpg frame_6_out.jpg frame_7_out.jpg frame_8_out.jpg frame_9_out.jpg
frame_10_out.jpg frame_11_out.jpg frame_12_out.jpg frame_13_out.jpg frame_14_out.jpg
frame_15_out.jpg



Dropbox > codes > CV > 96131125_HW06 > median_output_v1 > cyclists

frame_0_out.jpg frame_1_out.jpg frame_2_out.jpg frame_3_out.jpg frame_4_out.jpg
frame_5_out.jpg frame_6_out.jpg frame_7_out.jpg frame_8_out.jpg frame_9_out.jpg
frame_10_out.jpg frame_11_out.jpg frame_12_out.jpg frame_13_out.jpg frame_14_out.jpg

روش Background Subtraction با استفاده از روش Gaussian Mixture Model در کتاب Sonka به صورت زیر آورده شده است:



Algorithm 16.5: Background maintenance by Gaussian mixtures

1. Initialize: Choose K the number of Gaussians and a learning constant α: values in the range 0.01–0.1 are commonly used. At each pixel, initialize K Gaussians $N_k = N(\boldsymbol{\mu}_k, \Sigma_k)$ with mean vector $\boldsymbol{\mu}_k$ and covariance matrix Σ_k, and corresponding weights ω_k. Since the algorithm will evolve this may safely be done crudely on the understanding that early measurements may be unreliable.
2. Acquire frame t, with intensity vector $\mathbf{x}_t$—probably this will be an RGB vector $\mathbf{x}_t = (r_t, g_t, b_t)$. Determine which Gaussians match this observation, and select the 'best' of these as l. In the 1D case, we would expect an observation to be within, say, 2.5σ of the mean. In the multi-dimensional case, a simplifying assumption is made for computational complexity reasons: the different components of the observation are taken to be independent and of equal variance σ_k^2, allowing a quick test for 'acceptability'.
3. *If a match is found as Gaussian* l:
   (a) Set the weights according to equation (16.47), and re-normalize.
   (b) Set
   $$\rho = \alpha\, N(\mathbf{x}_t | \boldsymbol{\mu}_l, \sigma_l)$$
   and
   $$\begin{aligned} \boldsymbol{\mu}_{lt} &= (1-\rho)\, \boldsymbol{\mu}_{l(t-1)} + \rho\, \mathbf{x}_t\,, \\ \sigma_{lt}^2 &= (1-\rho)\, \sigma_{l(t-1)}^2 + \rho\, (\mathbf{x}_t - \boldsymbol{\mu}_{lt})^T (\mathbf{x}_t - \boldsymbol{\mu}_{lt})\,. \end{aligned}$$
4. *If no Gaussian matched* $\mathbf{x}_t$: then determine $l = \operatorname*{argmin}_k(\omega_k)$ and delete N_l. Then set
   $$\begin{aligned} \boldsymbol{\mu}_{lt} &= \mathbf{x}_t\,, \\ \sigma_{lt}^2 &= 2 \max_k \sigma_{k(t-1)}^2\,, \\ \omega_{lt} &= 0.5 \min_k \omega_{k(t-1)}\,. \end{aligned}$$
   (The algorithm is reasonably robust to these choices).
5. Determine B as in equation (16.48), and thence from the current 'best match' Gaussian whether the pixel is likely to be foreground or background.
6. Use some combination of blurring and morphological dilations and erosions to remove very small regions in the difference image, and to fill in 'holes' etc. in larger ones. Surviving regions represent the moving objects in the scene.
7. Return to (2) for the next frame.

طبق این روش در قسمت یک الگوریتم، برای هر پیکسل k توزیع گاوسی با میانگین و ماتریس کواریانس تصادفی ایجاد می‌شود و به هر کدام وزنی تصادفی نسبت داده می‌شود. در قسمت دو این الگوریتم یک فریم می‌آید، هر پیکسل این فریم با k توزیع‌های گاوسی نظیرش مقایسه می‌شود و نزدیک‌ترین توزیع گاوسی به مقدار پیکسل انتخاب می‌شود. البته ممکن است مقدار پیکسل از k توزیع گاوسی بسیار دور باشد، در آن صورت می‌گوییم توزیع گوسی پیدا نشد. قدم سوم این الگوریتم برای زمانی است که بهترین توزیع گوسی (l) پیدا شده باشد. در این مرحله وزن بهترین توزیع l و سایر توزیع‌ها به صورت زیر به‌روزرسانی می‌شوند.

$$\begin{aligned} \omega_{kt} &= (1-\alpha)\, \omega_{k(t-1)} && \text{for} \quad k \neq l\,, \\ &= \omega_{k(t-1)} && \text{for} \quad k = l\,, \end{aligned}$$

سپس میانگین و کواریانس بهترین توزیع (l) مانند فرمول‌های بالا به روزرسانی می‌شوند. به طوری که احتمال پیکسل جدید در بهترین توزیع بیشتر شود.

قدم چهارم این الگوریتم برای زمانی است که بهترین توزیع متناسب با مقدار پیکسل جدید پیدا نشود. در این مرحله دورترین توزیع به مقدار پیکسل پیدا می‌شود و سپس آن توزیع گاوسی حذف می‌شود و توزیع جدیدی افزوده می‌شود. میانگین این توزیع برابر با مقدار پیکسل است. وزنش برابر با نصف کمترین وزن سایر توزیع‌های آن پیکسل است و انحراف از معیارش دو برابر بیشتر انحراف معیار سایر توزیع‌های آن پیکسل است.

در قدم پنج مقدار B باید پیدا شود. در این قدم ابتدا توزیع‌های گاوسی یک پیکسل بر اساس معیار زیر مرتب می‌شوند:

$$\omega_{kt}/\sigma_{kt}$$

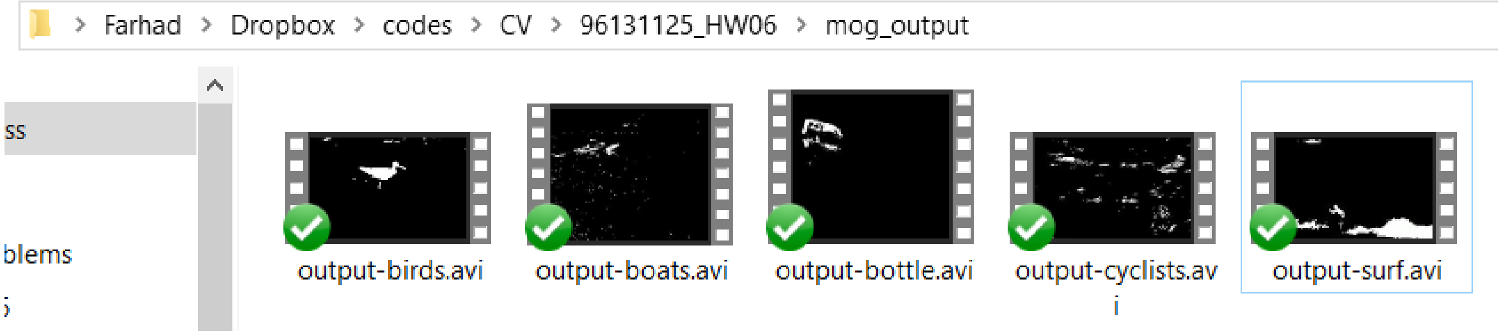
سپس مقدار B اینگونه انتخاب می‌شود:

$$B = \operatorname*{argmin}_{b} \left(\sum_{k=1}^{b} \omega_{kt} > T \right)$$

B توزیع اول بیشتر وزن را دارند که نشان دهندهی تغییرات کم است. این توزیع‌ها نشان دهنده پس زمینه هستند. اکنون توزیع برنده‌ی هر پیکسل را می‌دانیم و با توجه به B می‌توانیم تشخیص دهیم پیکسل پس‌زمینه یا غیرپس‌زمینه است.

دوباره این مراحل برای فریم‌های بعدی انجام می‌شود.

خروجی‌های این الگوریتم در پوشه‌ی mog_output قرار دارد در این پوشه ۵ ویدیو قرار دارد که خروجی متناظر با هر دسته از دنباله‌ها است.



به‌طور کلی روش Mixture Of Gaussian بهتر از روش Median برای Background subtraction عمل کرده است. تفاوت‌هایی که مشاهده می‌شود به صورت زیر است:

- روش Mixture Of Gaussian خطای کمتری در تشخیص پس‌زمینه دارد.
- روش Mixture Of Gaussian نیاز به حد آستانه برای باینری کردن تصویر ندارد، در حالی که این موضوع در روش Median وجود دارد و در صورت عدم انتخاب حد آستانه مناسب برای یک دنباله تصویر، نتایج خوبی حاصل نمی‌شود. هر دنباله نیاز به حد آستانه‌ی بخصوصی دارد تا عملکرد حداکثر شود.
- در روش Mixture Of Gaussian شاهد تغییرات شدیدی نیستیم در حالی که در روش median در برخی فریم‌ها تغییرات شدیدی نسبت به فریم قبل مشاهده می‌کنیم.
- در بعضی از دنباله تصویرها پس زمینه با اینکه خودش دچار تغییراتی می‌شود و تغییرات کم است مانند دنباله بطری در دریا و یا دنباله موج سواری، در این دنباله‌ها روش Median عملکرد خوبی داشته است و حتی می‌توان گفت عملکرد بسیار نزدیکی به Mixture of Gaussian داشته است. ولی در بعضی از دنباله‌ها تغییرات پس‌زمینه شدید است مانند دنباله‌ی دوچرخه سواری، در این دنباله Median نسبت به MOG بسیار ضعیف عمل کرده است. در این مورد MOG به خوبی پس زمینه را تشخیص می‌دهد.

نتیجه می‌گیریم در صورتی که تغییرات پس‌زمینه شدید نباشد و محیط ساده‌ای داشته باشیم می‌توان از روش‌های ساده‌ای مانند Median استفاده کرد و نتایج خوبی گرفت و در زمان‌هایی که پس زمینه پیچیده است و دچار تغییرات زیادی می‌شود و محیط پیچیده‌ای داریم باید از روش‌های پیشرفته‌تری نسبت به Median مانند MOG استفاده کنیم.