تمرین شماره ۵ (بخش اول)

درس تحلیل و پردازش تصاویر پزشکی

امید شرفی

400201518

دکتر فاطمی زاده

June 30, 2022

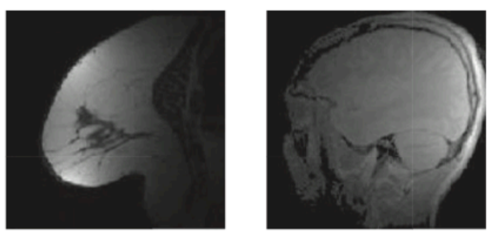
بخش تئوری

سوال ۱

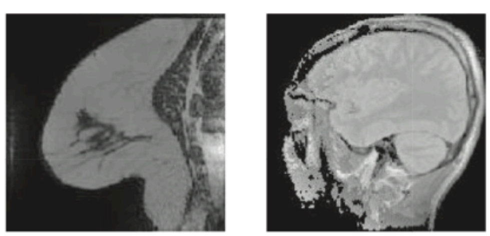
تصاویر حاصل از روش MR (تصویربرداری رزونانس مغناطیسی) کاربردهای مختلفی در تشخیص‌های بالینی و انواع آنالیزهای تصاویر دارند. یکی از معظلاتی که در تصاویر MR با آن سر و کار داریم ناهمگنی شدت میدان (IIH) در تصاویر حاصله می‌باشد.

این عدم یکنواختی شدت میدان در تصویر به صورت تغییرات آهسته و غیرتشریحی در یک بافت یکسان در نقاط مختلف تصویر مشاهده می‌گردد. علت اصلی این عدم یکنواختی به خاطر ناهنگنی میدان استاتیک یا عدم یکنواختی فرکانس رادیویی دستگاه بوده یا حرکت بیمار می‌باشد.

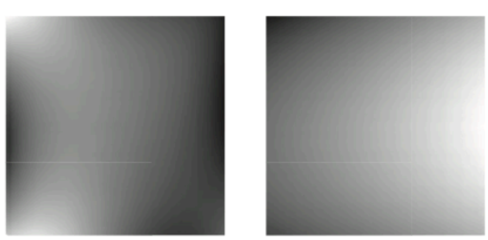
به طور مثال در شکل زیر دو تصویر آورده شده که در بخش a تصاویر شامل آرتیفکت IIH بوده و با نقشه تخمینی که در بخش c آروده شده‌اند، پردازش شده و تصویر تصحیح شده‌ی b به دست آمده است.



(a)



(b)



(c)

آرتیفکت ناهمگنی را معمولا با مدل زیر شبیه سازی می‌کنند.

$$\mathbf{x} = \boldsymbol{\alpha}\mathbf{x}' + \boldsymbol{\xi}$$

در این مدل فرض می‌شود که تصویر اصلی توسط یک ضریب آلفا که نماینده آرتیفکت IIH است به همراه ξ که نماینده نویز ما است تصویر دریافت شده توسط دستگاه را تشکیل می‌دهند.

در نتیجه حضور این آرتیفکت در تصویر باعث کاهش شدید دقت نتایج سگمنتیشن و رجیستریشن تصویرمان می‌شود.

الگوریتم AFCM به این صورت عمل می‌کند که تابع انرژی این روش به صورت زیر تعریف می‌شود که نسبت به تابع انرژی FCM معمول شامل دو ترم اضافه‌تر با ضرایب رگولاریزیشن مربوطه می‌باشد.

$$J_{AFCM} = \sum_{(i,j)} \sum_{k=1}^{C} u_k^2(i,j) \| y(i,j) - m(i,j) v_k \|_2^2 + \cdots$$
$$\lambda_1 \sum_{(i,j)} \left((m(i,j) * D_i)^2 + \left(m(i,j) * D_j\right)^2 \right) + \cdots$$
$$\lambda_2 \sum_{(i,j)} \left((m(i,j) ** D_{ii})^2 + 2\left(m(i,j) ** D_{ij}\right)^2 + \left(m(i,j) ** D_{jj}\right)^2 \right)$$

در این تابع ترم استاندارد اول مشخص کننده مجموع حاصل ضرب نرم دوم اختلاف مقدار پیکسل مربوطه و اصلاح شده پروتوتایپ خوشه ی K ام توسط میدان ضرب شونده‌ی m بوده که این نقشه ضرایب مدل کننده‌ی تغییرات شدت روشنایی می‌باشد.

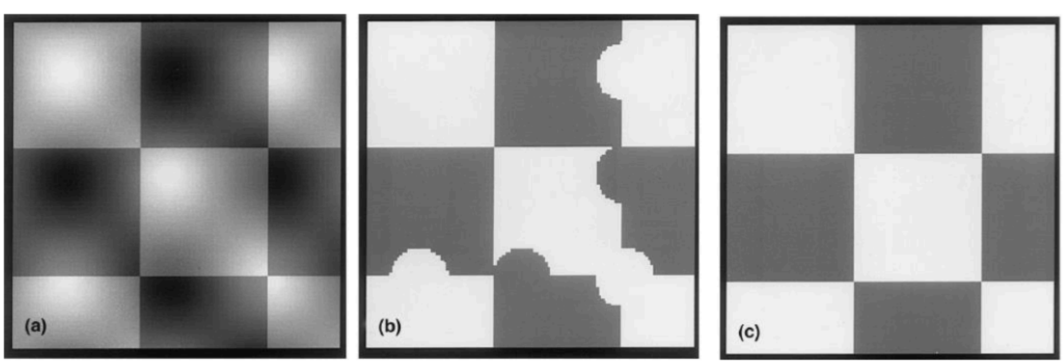
ترم دوم تابع بالا مجموع توان دوم مشتق نقشه ضرایب در دو جهت افقی و عمودی بوده و ترم سوم نیز مشتق دوم این میدان ضرایب می‌باشد. در نتیجه دو ترم آخر اجازه نمی‌دهند میدان تغییرات شدیدی بکند و ساختار هموار در میدان مدل کننده شدت روشنایی وجود داشته باشد.

در نتیجه با استفاده از الگوریتم تکرارشونده‌ی زیر می‌توان ترم‌های مساله از جمله میدان شدت روشنایی را به ترتیب به صورت عددی محاسبه کرد.

› Algorithm steps:

1. Initial guess for $\{v_k\}_{k=1}^{C}$ using k-means, FCM, ... and set $m(i,j) = 1$
2. Compute membership functions:
$$u_k(i,j) = \frac{\|y(i,j) - m(i,j)v_k\|^{-2}}{\sum_{l=1}^{C}\|y(i,j) - m(i,j)v_l\|^{-2}}$$
3. Compute centroids:
$$v_k = \frac{\sum_{(i,j)} u_k^2(i,j)m(i,j)y(i,j)}{\sum_{(i,j)} u_k^2(i,j)m^2(i,j)}$$
4. Compute multiplier field:
$$y(i,j)\sum_{k=1}^{C} u_k^2(i,j)v_k = m(i,j)\sum_{k=1}^{C} u_k^2(i,j)v_k^2 + \lambda_1\big(m(i,j) ** H_1(i,j)\big) + \lambda_2\big(m(i,j) ** H_2(i,j)\big)$$
where $H_1(i,j) = D_i * \breve{D}_i + D_j * \breve{D}_j$ and $H_2(i,j) = D_{ii} * \breve{D}_{ii} + +2\big(D_{ij} * \breve{D}_{ij}\big) + D_{jj} * \breve{D}_{jj}, \breve{f}(i) = f(-i)$
5. Step over 2-3-4 until convergence

در نتیجه دقیقا با همین ترم ضرب شونده ای که این روش حل در نظر گرفته است ناهمگنی میدان که در تصاویر MR نیز رخ می‌دهد مدیریت شده و پردازش نهایی تصویر خروجی با صحت درست حاصل میگردد. به طور مثال در شکل زیر که تصویر شامل ناهمگنی میدان می‌شود، تصویر حاصل از کیفیت درست برخوردار می‌باشد.



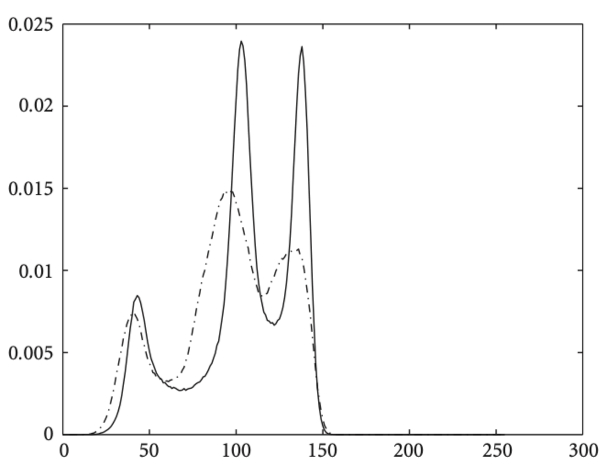
(a) original (b) FCM (c) AFCM

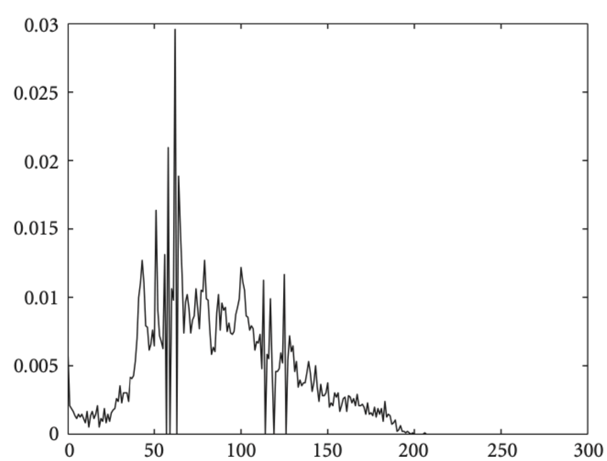
سوال ۲

همانطور که در سوال اول توضیح داده شد، مدل آرتیفکت IIH به فرمت یک مپ شدت میدان ضرب شونده که ناهمگنی میدان را مدل می‌کند در سیگنال اصلی بوده و هدف مدلهای بازسازی کننده نیز تخمین همین نقشه ضریب شدت روشنایی ضرب شده در سیگنال می‌باشد که بتوانند با استفاده از آن به حذف آرتیفکت ناهمگنی و بازسازی مجدد تصویر بپردازند.

در مقاله‌ی مذکور روش‌های گوناگونی برای حذف آرتیفکت ناهمگنی ارائه گردیده است.

۱. دسته‌ای از روش‌ها تحت عنوان Surface Fitting عموما بر مبنای این موضوع هستن که در یک الگوریتم تکرار شونده، همزمان مساله‌ی خوشه‌بندی تصوی حل شده و از طرف دیگر با استفاده از نتیجه حاصل دسته‌ها، ضریب اصلاح کننده‌ی ناهمگنی میدان تخمین زده شده و مقادیر آپدیت می‌گردد. لزوما اما همیشه از خواص محلی دسته‌ها استفاده نمی‌شود و می‌توان از خواص عمومی کل تصویر نیز استفاده کرد. بدین منظور در روش‌های مبتنی بر Entropy minimization بدین صورت عمل می‌کنند که ما می‌دانیم با اضافه شدن آرتیفکت ناهمگنی به تصویر ساختار هیستوگرام تصویر صافتر می‌شود.





به طور مثال در تصویر بالا، نمودار خط سمت چپ هیستوگرام تصویر بدون آرتیفکت را نشان می‌دهد، نمودار نقطه‌چین هیستوگرام با افزودن ۴۰ درصد آرتیفکت IIH به تصویر اصلی مغز را نشان داده و نمودار سمت راست هیتوگرام تصویر با افزودن شدید آرتیفکت ناهمگنی را نشان می‌دهد. همانطور که مشاهده می‌شود، صاف شدن هیستوگرام با افزودن آرتیفکت IIH باعث افزایش آنتروپی تصویر شده و در نتیجه در روش های مبتنی بر Entropy minimization به گونه‌ای دنبال پارامتر ضرایب اصلاح کننده تصویر می‌گردیم که باعث کاهش آنتروپی تصویر شود. طبیعتا صرف حداقل کردن آنتروپی تصویر ما رو به پاسخ بدیهی تمام صفر می‌برد و در روش‌های مختلف انواع قیود مشروط کننده دیگر نیز به مساله بهینه سازی اضافه می‌گردد.

۲. دسته‌ی دیگری از روش‌ها مبتنی بر STATISTICAL MODELING بوده که در این دسته ما برای توزیع آماری مپ IIH یک پیش فرض کرده و متناسب با این توزیع فرض شده مساله بهینه سازی را به صورت شرطی حل می‌کنیم. از روش‌های این این دسته می‌توان به Bayesian framework اشاره نمود. در این روش با در نظر گرفتن وکتور رندوم β المان های وکتور با حداکثر کردن احتمال رخ داد نسبت به داده‌ی موجود مساله یعنی پارامتر y که به خاطر مدلی که برای آرتیفکت IIH در نظر گرفتیم، همان لگاریتم مشاهده‌ی x می‌باشد محاسبه می‌گردد. حال مساله‌ی MAP به دست آمده با استفاده از قانون بیز به فرم زیر فرمبندی می‌شود :

$$\hat{\beta} = \max_{\beta} p(\mathbf{y} \mid \beta) p(\beta)$$

در نتیجه مساله‌ی ما تخمین دو ترم بالا بوده که این تخمین توسط مدل توزیع گوسی به صورت زیر محاسبه می‌گردد.

$$p(\beta) = G_{\psi_\beta}(\beta),$$
$$p(y_i \mid \Gamma_i, \beta_i) = G_{\psi_{\Gamma_i}}(y_i - \mu(\Gamma_i) - \beta_i),$$

در رابطه‌ی بالا Γi نماینده‌ی کلاس مربوط به پیکسل i-ام بوده و در نتیجه μ(Γi) میانگین ناحیه‌ی کلاس مربوطه بوده و می‌تواند در تخمین دقیق تر به ما کمک کند. همچنین توزیع آماری G به صورت زیر بوده که Ψ_x ماتریس کواریانس می‌باشد.

$$G_{\psi_{\mathbf{x}}}(\mathbf{x}) = (2\pi)^{-n/2} \, |\psi_{\mathbf{x}}|^{-1/2} \exp\left(-\frac{1}{2}\mathbf{x}^T \psi_{\mathbf{x}}^{-1}\mathbf{x}\right)$$

در نتیجه با فرض استقلال پیکسل‌ها می‌توانیم ترم اول احتمال شرطی رابطه‌ی محاسبه‌ی $\hat{\beta}$ را با بسط رابطه‌ی احتمال شرطی یک پیکسل بر روی کلاس‌های مختلف به صورت زیر محاسبه کنیم.

$$\begin{aligned} p(\mathbf{y} \mid \beta) &= \prod_i p(y_i \mid \beta_i) \\ &= \prod_i \sum_{\Gamma_i} p(y_i \mid \Gamma_i, \beta_i) \, p(\Gamma_i) \end{aligned}$$

در نتیجه با استفاده از این مدل دسته بندی تصویر با استفاده از مدل‌های گوسی اختصاص داده شده انجام شده و همچنین آرتیفکت IIH نیز در دل این تخمین گوسی اصلاح می‌گردد. در مطالعات دیگر در مورد تاثیر آرتیفکت IIH بر روی این الگوریتم، آورده شده است که ممکن است توزیع تصویر اصلی لزوما از ترکیب تعدادی توزیع گوسی مانند آنچه که در الگوریتم فرض شده است، پیروی نکند. در نتیجه در مدل اصلاح شده‌ی زیر یک کلاس Γother ما توزیع غیر گوسی فرض شده و نویز IIH مشاهده مساله‌ی اصلی در دل توزیع های گوسی حل شده و توزیع غیرگوسی فرض شده به دسته‌بند های گوسی کمک می‌کند تا کمتر تحت تاثیر داده‌های پرت تصویر قرار بگیرند.

$$p(y_i \mid \beta_i) = \sum_{\Gamma_j} p(y_i \mid \Gamma_j) \, p(\Gamma_j) + \lambda p(\Gamma_{\text{other}})$$

بخش شبیه سازی

سوال ۱

الف)

تابع هزینه استاندارد روش FCM به صورت زیر می‌باشد.

$$J = \sum_{i=1}^{c}\sum_{k=1}^{N} u_{ik}^{p}\|x_k - v_i\|^2$$

که بر روی N واکسل و c دسته ای که می‌خواهیم تصویر را به آنها دسته بندی کنیم، تقسیم می‌شود. در این تابع x_k مقدار لگاریتم تصویر صحیح بوده که عملا جلوتر با $y_k - \beta_k$ که به ترتیب لگاریتم تصویر ورودی و بایاس فیلد پیکسل k-ام می‌باشند تخمین زده می‌شود. v_i نیز مرکز دسته‌ی i-ام بوده و u_{ik} نیز میزان اختصاص پیکسل i-ام به دسته‌ی k-ام را نشان می‌دهد و در نتیجه باید از مجموعه‌ی زیر پیروی کنند.

$$\mathcal{U}\left\{u_{ik} \in [0,\ 1] \middle| \sum_{i=1}^{c} u_{ik} = 1\ \forall\ k \text{ and } 0 < \sum_{k=1}^{N} u_{ik} < N\ \forall\ i\right\}$$

حال در روش BCFCM علاوه بر ترم روش استاندارد، همین ترم را با فرض هموار بودن تصویر در یک همسایگی پیکسل مربوطه محاسبه و میانگین گرفته و با ضریب آلفا در تابع هزینه روش اعمال می‌کند. در نتیجه همسایه‌های یک پیکسل مانند یک ترم هموارساز در لیبل زدن و تخمین بایاس فیلد به صورت تکه‌ای همنواخت کمک می‌کنند.

$$J_m = \sum_{i=1}^{c}\sum_{k=1}^{N} u_{ik}^{p}\|y_k - \beta_k - v_i\|^2 + \frac{\alpha}{N_R}\sum_{i=1}^{c}\sum_{k=1}^{N} u_{ik}^{p}\left(\sum_{y_r\in\mathcal{N}_k}\|y_r - \beta_r - v_i\|^2\right)$$

پارامترهای Membership، Class Prototypes و Bias Field دقیقا سه مجموعه ای هستند که به عنوان متغیر در مساله‌ی بهینه سازی تابع هزینه‌ی بالا وجود دارند.

$$\min_{U,\{v_i\}_{i=1}^{c},\{\beta_k\}_{k=1}^{N}} J_m, \qquad \text{subject to } U \in \mathcal{U}.$$

پارامتر Membership در حقیقت همان ماتریس U ما است که u_{ik} میزان اختصاص پیکسل i-ام به دسته‌ی k-ام را نشان می‌دهد و همانطور که در بالا توضیح داده شد باید از قیود مجموعه $\mathcal{U}$ پیروی کند. با حل روابط فرم نهایی آپدیت پارامتر u_{ik} هر مرحله از فرمول زیر پیروی می‌کند.

$$u_{ik}^{*} = \frac{1}{\sum_{j=1}^{c}\left(\frac{D_{ik}+\frac{\alpha}{N_R}\gamma_i}{D_{jk}+\frac{\alpha}{N_R}\gamma_j}\right)^{1/(p-1)}}$$

پارامتر Bias Field پارامتری است که تخمین آرتیفکت IIH را برای ما می‌زند و شدت روشنایی تصویر را برای هر پیکسل اصلاح می‌کند. فرض مدل ما برای داده‌ی دریافت شده در پیکسل k-ام به صورت زیر بوده که یک ساختار ضربی بوده و در نتیجه با لگاریتم گرفتن از طرفین ترم Bias Field برای پیکسل k-ام به دست می‌آید. ما در مساله‌ی بهینه سازی لگاریتم X_k را احتیاج داشتیم که به جای آن از تفاوت داده‌ی مشاهده شده و بایاس فیلد استفاده کردیم و بایاس فیلد را به عنوان یکی از متغیرهای مساله بهینه سازی در نظر گرفتیم.

$$Y_k = X_k G_k \qquad \forall\, k \in \{1,\, 2,\, \ldots,\, N\}$$

مدل داده‌ی مساله

$$y_k = x_k + \beta_k \qquad \forall\, k \in \{1,\, 2,\, \ldots,\, N\}$$

لگاریتم رابطه و ظهور ترم Bias Field

با مشتق گیری از تابع هزینه نسبت به پارامتر β_k و صفر قرار دادن آن داریم:

$$\left[\sum_{i=1}^{c} \frac{\partial}{\partial \beta_k} \sum_{k=1}^{N} u_{ik}^{p} (y_k - \beta_k - v_i)^2\right]_{\beta_k = \beta_k^*} = 0.$$

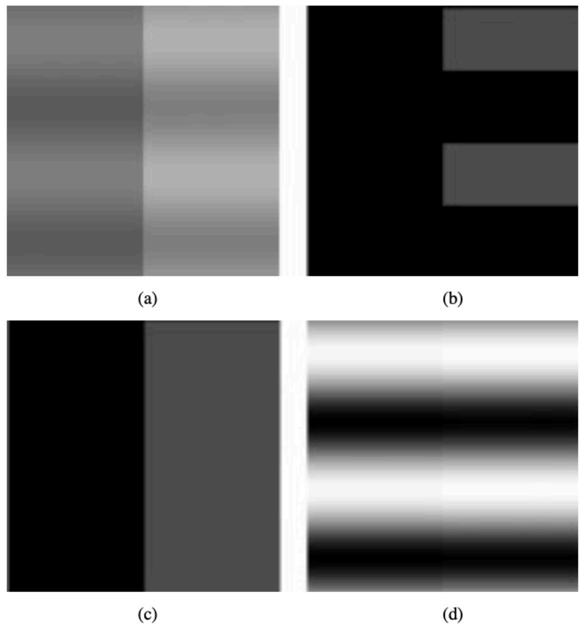
و در نتیجه رابطه‌ی آپدیت بایاس فیلد در هر تکرار به صورت زیر در می‌آید.

$$\beta_k^* = y_k - \frac{\sum_{i=1}^{c} u_{ik}^{p} v_i}{\sum_{i=1}^{c} u_{ik}^{p}}$$

ترم Class Prototypes نیز همان v_i های ما هستند که پس از لیبل زدن پیکسل‌ها و با استفاده از ماتریس U نماینده مرکز سگمنت i-ام ما می‌باشد که در هر تکرار با استفاده از رابطه‌ی زیر آپدیت می‌شوند.

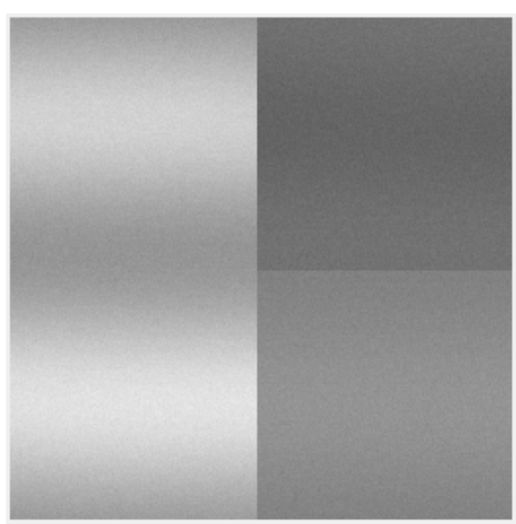
$$v_i^* = \frac{\sum_{k=1}^{N} u_{ik}^{p} \left((y_k - \beta_k) + \frac{\alpha}{N_R} \sum_{y_r \in \mathcal{N}_k} (y_r - \beta_r) \right)}{(1+\alpha) \sum_{k=1}^{N} u_{ik}^{p}}$$

ب)

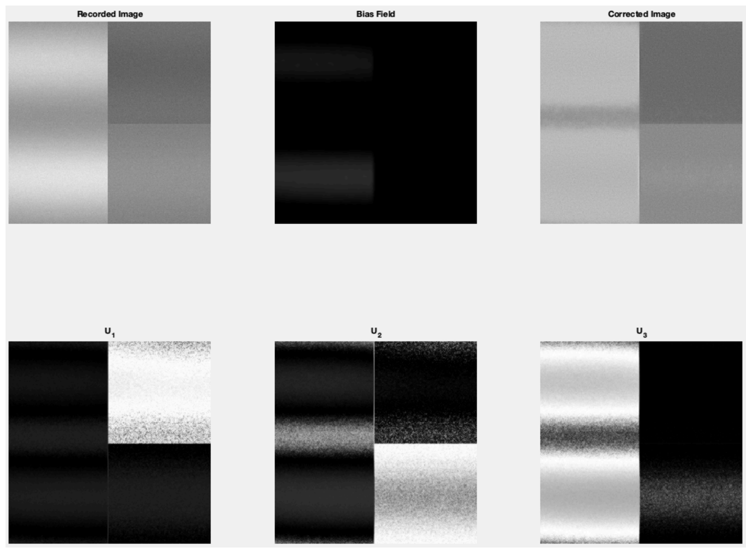


نمونه تصویر موجود در مقاله‌ی مربوطه در شکل بالا آمده است که با توجه به شکل ورودی در بخش a بایاس فیلد مربوطه در شکل d نمایش شده و ساختار موجی ناهمگن بخش ها گرفته شده و به دو بخش اصلی چپ و راست در شکل c به صورت همگن تقسیم شده است.

با توجه به نتیجه‌ی گزارش شده در شکل بالا، تصویر test_biasfield_noise موجود در شکل زیر باید به سه بخش همگن شده پس از اصلاح بایاس فیلد تقسیم شود.



در نتیجه خروجی روش، بایاس فیلد به دست آمده و نقشه‌ی ماتریس های تعلق U برای هر کدام از دسته‌ها به صورت زیر به دست می‌آیند.



سوال ۲

الف)

فرم‌بندی روش سنتی snake به صورت یک خم x(s) بوده که از مینیمم کردن تابع انرژی زیر به دست می‌آید.

$$E = \int_0^1 \frac{1}{2}(\alpha|\mathbf{x}'(s)|^2 + \beta|\mathbf{x}''(s)|^2) + E_{\text{ext}}(\mathbf{x}(s))ds$$

پارامترهای آلفا و بتا کنترل کننده تنش و سختی خم بوده و ترم انرژی خارجی که در نواحی ویژگی مورد نظر ما از تصویر (به طور مثال لبه ها در مساله‌ی سگمنتیشن) حداقل مقدار خود را می‌گیرد.

در نتیجه تابع E_{ext} را می‌تواند به صورت گرادیان تصویر اصلی تعریف کرد یا برای رفع نویز اولیه، ابتدا یک کرنل گوسی را در تصویر کانوال کرده و گرادیان حاصل را به عنوان تابع انرژی خارجی قرار دهیم. البته باید توجه داشته باشیم که با توجه به آن که تابع انرژی نماینده تابع هزینه است پس در لبه‌ها باید به صورت علامت منفی ظاهر شود.

$$\begin{aligned} E^1_{\text{ext}}(x,y) &= -|\nabla I(x,y)|^2 \\ E^2_{\text{ext}}(x,y) &= -|\nabla(G_\sigma(x,y) * I(x,y))|^2 \end{aligned}$$

و مشخصا اگر تصویر به صورت باینری مشکی بر روی سفید باید می‌توان تابع‌های بالا را به حالات زیر ساده کرد.

$$\begin{aligned} E^3_{\text{ext}}(x,y) &= I(x,y) \\ E^4_{\text{ext}}(x,y) &= G_\sigma(x,y) * I(x,y) \end{aligned}$$

با توجه به تعریف تابع انرژی، خم پاسخ مساله باید در رابطه‌ی اویلر زیر صدق کند :

$$\alpha \mathbf{x}''(s) - \beta \mathbf{x}''''(s) - \nabla E_{\text{ext}} = \mathbf{0}$$

این رابطه را می‌توان به دو ترم F_{int} و F_{ext} تقسیم کرد که ترم F_{int} شامل دو جمله‌ی اول رابطه بالا بوده و تنش و سختی خم را کنترل کرده و ترم F_{ext} خم را به سمت مرز موردنظر هل می‌دهد.

برای حل معادله اویلر بالا به صورت تکرار شونده به سادگی خم را وابسته به پارامتر زمان کرده و مقدار معادله‌ی بالا را برابر با مشتق نسبت به زمان خم گذاشته که در نتیجه بسته به علامت معادله خم به سمت صفر کردن آن پیش برود.

$$\mathbf{x}_t(s,t) = \alpha \mathbf{x}''(s,t) - \beta \mathbf{x}''''(s,t) - \nabla E_{\text{ext}}$$

ب)

تابع انرژی روش GVF به صورت زیر است:

$$\mathcal{E} = \iint \mu({u_x}^2 + {u_y}^2 + {v_x}^2 + {v_y}^2) + |\nabla f|^2 |\mathbf{v} - \nabla f|^2 dxdy$$

این تابع انرژی یک عملگرد دوگانه دارد. تابع f برابر با edge map تصویر ما بوده و در اطراف لبه‌های تصویر مقادیر بزرگی دارد. با توجه به تابع انرژی خارجی تعریف شده در بخش قبل می‌توان تابع f را به صورت زیر نوشت. گرادیان این تابع در مکان هایی که تصویر یکنواخت است صفر خواهد بود.

$$f(x, y) = -E^i_{\text{ext}}(x, y)$$

حال در حالتی که گرادیان f کوچک باشد یعنی در نواحی هموار باشیم، ترم اول تابع انرژی که مشتق‌های جزئی میدان برداری می‌باشد در انتگرال انرژی مقدارخواهد داشت و میدان به سمت همنواخت بودن خواهد رفت. پارامتر μ نیز صرفا یک ترم تنظیم کننده همواری میدان بوده که متناسب با سطح نویز در صورت بالا بودن نویز مقادیر بزرگتری تنظیم می‌شود.

در عین حال زمانی که گرادیان بزرگ باشد ترم دوم ترم مقداردار انتگرال خواهد بود و مقدار میدان برداری v به سمت گرادیان f که همان نماینده‌ی بزرگی لبه‌ی تصویر هست می‌رود تا ترم ضرب شونده در گرادیان f را به سمت صفر ببرد.

در ادامه با استفاده از calculus of variations می‌توان نشان داد که ترم‌های u و v میدان برداری پاسخ مساله‌ی بالا در معادلات اویلر زیر صدق می‌کنند:

$$
\begin{aligned}
\mu\nabla^2 u - (u - f_x)({f_x}^2 + {f_y}^2) &= 0 \\
\mu\nabla^2 v - (v - f_y)({f_x}^2 + {f_y}^2) &= 0
\end{aligned}
$$

این معادلات به ما نشان می‌دهند که در نواحی هموار که گرادیان f صفر است بخش دوم دو معادله‌ی بالا صفر بوده و در نتیجه دو ترم u و v در معدلات لاپلاس صدق می‌کنند.

همچنین برای حل معادلات به صورت تکرار شونده نیز مشابه روشی که در بخش قبل به کار گرفتیم مقادیر ترم های u و v را نسبت به زمان پارامتری کرده و مشتق نسبت به زمان آنها را دقیقا برابر با دو معادله‌ی بالا قرار داده تا در صورتی که مقادیر معادلات به سمت صفر رود و پاسخ صدق کند، تغییرات نیز صفر شده و به حالت پایدار پاسخ رسیده باشیم.

$$
\begin{aligned}
u_t(x,y,t) &= \mu\nabla^2 u(x,y,t) - (u(x,y,t) - f_x(x,y)) \\
&\quad \cdot(f_x(x,y)^2 + f_y(x,y)^2) \qquad \text{(12a)} \\
v_t(x,y,t) &= \mu\nabla^2 v(x,y,t) - (v(x,y,t) - f_y(x,y)) \\
&\quad \cdot(f_x(x,y)^2 + f_y(x,y)^2) \qquad \text{(12b)}
\end{aligned}
$$

ج)

تابع Snake2D به عنوان ورودی تصویر ورودی I که به بازه‌ی صفر تا یک اسکیل شده و یک آرایه‌ی 2 * N که مختصات x و y مربوط به N نقطه‌ی کانتور اولیه می‌باشد. همچنین یک استراکت Options نیز به عنوان ورودی دریافت می‌کند که تنظیمات عمومی شامل موارد زیر می‌شود.

```
options (general),
 Option.Verbose : If true show important images, default false
 Options.nPoints : Number of contour points, default 100
 Options.Gamma : Time step, default 1
 Options.Iterations : Number of iterations, default 100
```

که در آن تعداد تکرار، استپ زمانی الگوریتم و تعداد نقاط کانتوری که الگوریتم ما در نظر می‌گیرد را مشخص می‌شود.

```
% options (Image Edge Energy / Image force))
%  Options.Sigma1 : Sigma used to calculate image derivatives, default 10
%  Options.Wline : Attraction to lines, if negative to black lines otherwise white
%                  lines , default 0.04
%  Options.Wedge : Attraction to edges, default 2.0
%  Options.Wterm : Attraction to terminations of lines (end points) and
%                  corners, default 0.01
%  Options.Sigma2 : Sigma used to calculate the gradient of the edge energy
%                   image (which gives the image force), default 20
```

همچنین مجموعه پارامترهای روابط از جمله سیگماهای، پارامترهای Wline و Wedge که تنظیم مناسب آنها توسط کاربر متناسب به نوع تصویر و کانتور حدودی اولیه‌ای که انتخاب می‌شود بسیار مهم بوده و مشخص کننده میزان حرکت و نحوه‌ی حرکت کانتور به صورت شکستهایی در راس‌ها یا به سمت خطوط تصویر می‌باشد. Wterm نیز میزان حرکت کانتور به سمت انتهای خطوط و نقاط پایانی شکل را مشخص می‌کند.

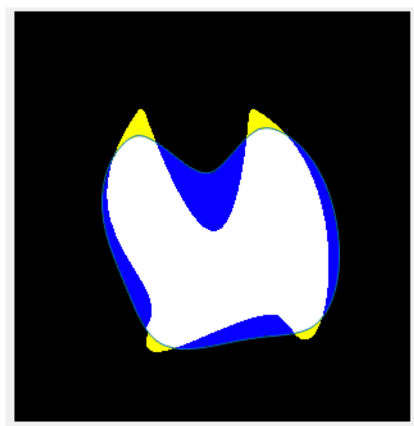
تابع Snake2D در ادامه ابتدا با استفاده از تابع InterpolateContourPoints2D و مجموعه نقاط کانتور اولیه‌ی P کانتور دو بعدی مرحله اول الگوریتم را می‌سازد. همچنین با استفاده از سه ترم W ای که در عنوان پارامترهای ورودی مشخص کردیم و سیگما۱ به عنوان ورودی‌های تابع ExternalForceImage2D مقدار تابع E_{ext} را مشخص کرده و با استفاده از توابع ImageDerivatives2D و ورودی‌های E_{ext} و سیگما۲ مقادیر مشبق در جهات افقی و عمودی به دست می‌آیند. در ادامه با استفاده از F_{ext} GVFOptimizeImageForces2D که Gradient vector flow را محاسبه و آپتیمایز می‌کنیم.

در ادامه اگر GVF روشن باشد و Options.Verbose برابر با یک باشد میدان نیروی انرژی محاسبه شده از GVF ترسیم شده و حرکت snake در subplot چهارم در طول هر تکرار الگوریتم ترسیم می‌گردد. در هر تکرار الگوریتم کانتور snake با استفاده از تابع

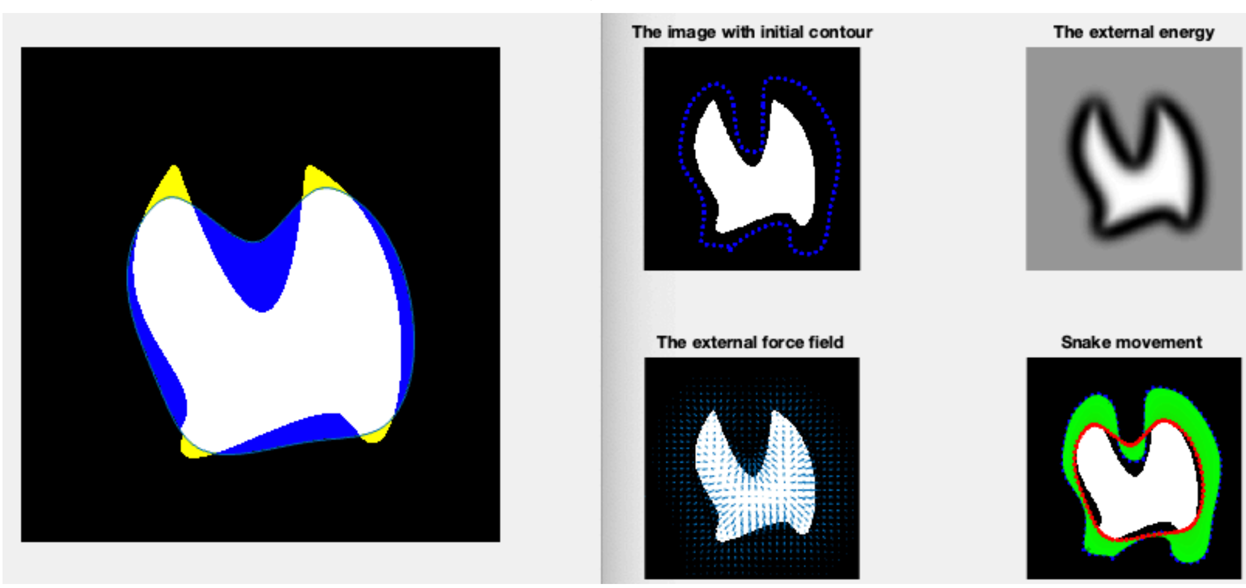
SnakeMoveIteration2D که نیروی خارجی و پارامترهای گاما و کاپا و دلتا را می‌گیرد محاسبه و آپدیت می‌شود.

د)

بسته با انتخاب کانتور اولیه و تنظیم پارامترها مختلف پاسخ‌های گوناگونی برای هر دو روش حاصل می‌شود. همچنین تعداد تکرار الگوریتم نیز در پاسخ موثر می‌باشد. در بعضی از تنظیمات پارمترها پاسخ کند بوده و نیاز به تعداد تکرار بالا به منظور همگرایی snake به دور مرز شکل مربوطه می‌شود.

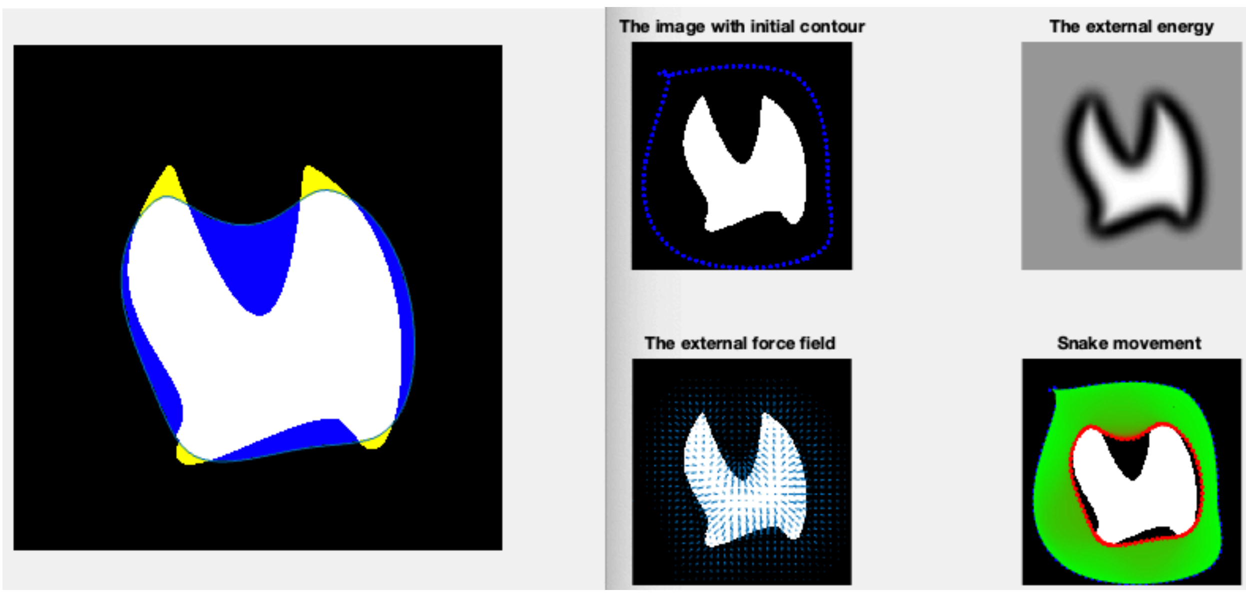


مدل Snake با انتخاب نقطه در داخل
حفره و تنظیمات پیش فرض



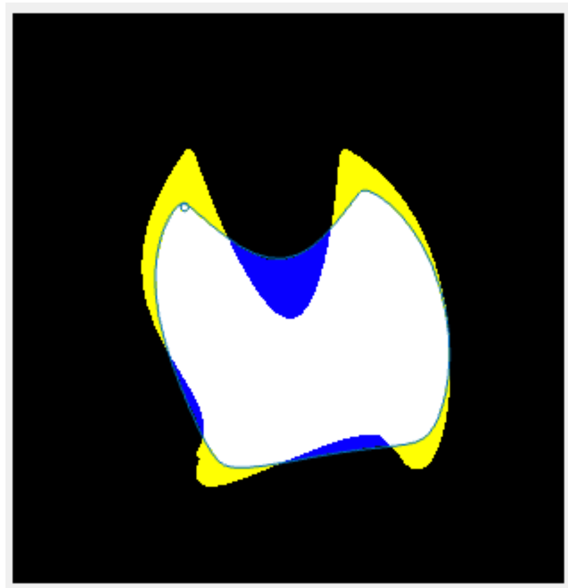
پاسخ GVF با انتخاب نزدیک نقطه با مرزها و تنظیمات پیش فرض

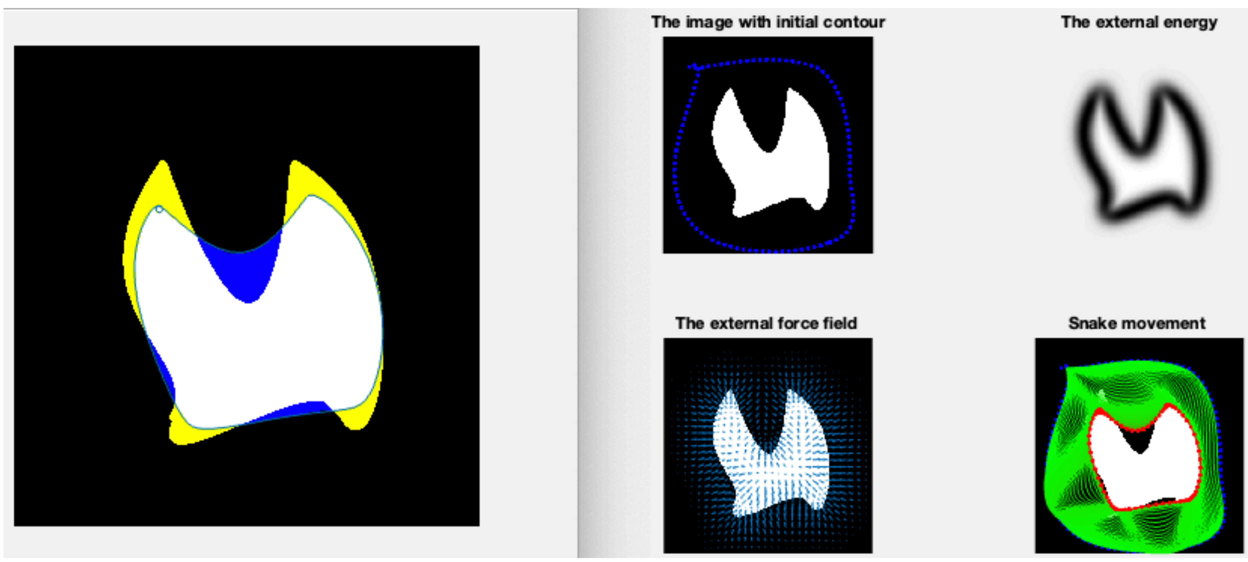
همانطور که مشاهده می‌شود در حالتی که در کانتور اولیه نقطه داخل حفره‌ی بالا قرار بگیرد پاسخ با همان تنظیمات اولیه قابل قبول بوده و البته GVF اندکی عمکرد بهتر دارد. در مجموع با توجه به ساختار حفره و حفظ شکلی که در تنظیمات Snake با استفاده از ضرایب آلفا و بتا وجود دارد، اگه Snake بخواهد وارد حفره شود ناخودآگاه لبه های تیز شکل اصلی در طرفین حفره خورده می‌شود.



همچنین در شرایطی که کانتور اولیه را صرفا یه شش ضلعی محدب دور از شکل انتخاب کنیم نیز پاسخ به سمت مرزهای شکل همگرا شده که البته همچنان به پاسخ مناسبی می‌رسد.

همچنین با تغییر پارامترهای مساله، پاسخ دو روش به شرح زیر به دست می‌آیند که همچنان روش GVF اندکی بهتر عمل می‌کند.






```
Options.nPoints = 100;
Options.Iterations = 500;

Options.Verbose = 0;
Options.Alpha = 0.1;
Options.Beta = 3;
Options.Delta = 0.4;
Options.Kappa = 4;

Options.Wline = 0.01;
Options.Wedge = 4.0;
Options.Wterm = 3;
[O,J] = Snake2D(I,[yi2 xi2], Options);
```