



تمرین شماره 5

درس پردازش و تحلیل سیگنال‌های پزشکی

استاد درس : دکتر فاطمی‌زاده

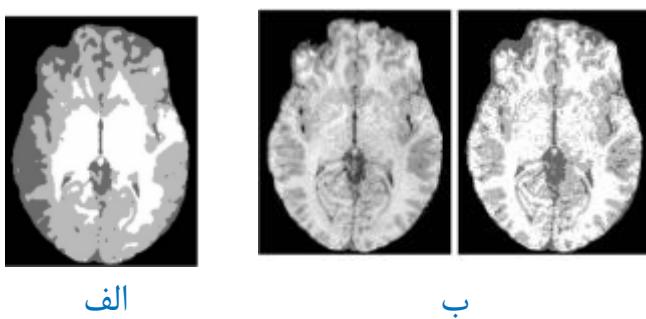
نام : محمد سینا حسن نیا

شماره دانشجویی : 96108515

بخش تئوری

سوال 1

چند کار عمده و اصلی پردازش تصویر MR، مانند segmentation & registration بستگی به کیفیت تصاویر دارد، زیرا از شدت به عنوان ویژگی اصلی استفاده می‌کنند. هر گونه distortion شدت کلاس‌های بافت زیرین در عملکرد الگوریتم منعکس می‌شودیکی از این اعوجاج "bias field" (IIH) نیز می‌باشد. معمولاً به عنوان "gain field" یا "smooth shading" در تصویر درنظر گرفته می‌شود. یکی از اثرات اصلی IIH را می‌توان در شکل الف به عنوان یک segmentation ضعیف مشاهده کرد، در حالی که سگمنتیشن را می‌توان بهبود داد. (شکل ب)



IIH همچنین می‌تواند ناشی از تغییرات ناخواسته local flip variation باشد اما در واقع حرکت‌های این تغییر می‌تواند چندگانه باشد. یک تصویر MR نتیجه ترکیب یک دستگاه تصویربرداری، یک object و یک pulse sequence است که همه این‌ها در واقع می‌توانند منبع این آرتیفیکت باشند. محدودیت‌های فیزیکی دستگاه تصویربرداری می‌توانند منجر به تغییرات زاویه چرخش شود. اولین دلیل ممکن یک میدان ساکن B_0 غیر یکنواخت می‌تواند باشد. با اینکه تغییرات موضعی B_0 تا حدی با shim tuning جبران می‌شود، می‌تواند منجر به تغییر شکل imaging plane شود. همچنین همانطور که در ابتدا نیز گفته شد همگنی سیم پیچ RF، موضعی

که به خواص هندسی و فیزیکی آنها بستگی دارد نیز می تواند تأثیر داشته باشد. **geometrical distortion fields** نیز می تواند به دلیل خواص غیر خطی محتمل که باعث **switching** ها نیز می تواند روی تصویر اثر بگذارد. مشکلات همچنین ممکن است ناشی از امپلی فایر ها و مبدل های دیجیتال آنالوگ باشد. همچنین پارامترهای ثبت (مانند این که اسلایس ها تداخل داشته باشند یا خیر) می توانند در دامنه **IIIH** گذار باشند. همچنین شکل **object** و خواص مغناطیسی آن نیز از جمله دلایل **IIIH** می توانند باشند. همچنین تکان خوردن بیمار و **radio frequency nonuniformity** نیز می تواند ارز دلایل دیگر باشد.

همانطور که می دانیم روش **FCM** عملکرد بدی هنگام **IIIH** دارد :

$$J_{FCM} = \sum_{(i,j)} \sum_{k=1}^C u_k^2(i,j) \|y(i,j) - v_k\|_2^2, \quad s.t. \sum_{k=1}^C u_k(i,j) = 1$$

بنابراین روش **AFCM** پیشنهاد شده است که در واقع اثر **IIIH** نیز در آن لحاظ شده است. تابع هزینه در این روش مطابق زیر تعریف می شود :

- › $m(i,j)$: unknown multiplier field (model the brightness variation), **slow variation!**
- › D_i and D_j are 1st order finite Difference operator for partial derivatives ($\frac{\partial m}{\partial x}, \frac{\partial m}{\partial y}$)
- › D_{ii}, D_{jj} and D_{ij} are 2nd order finite Difference operator for partial derivatives ($\frac{\partial^2 m}{\partial x^2}, \frac{\partial^2 m}{\partial y^2}, \frac{\partial^2 m}{\partial x \partial y}$)

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

همانطور که دیده می شود در رابطه بالا 3 ترم قابل تشخیص است. ترم اول کاری که می کند این هست که یک **multiplier field** در نظر می گیرد که مدل کننده تغییرات روشنایی می باشد.

فاصله در این روش به گونه ای اصلاح شده است که پروتوتایپ‌ها در ناهمگنی میدان مدل شده ضرب شده است . اگر تنها همین ترم را داشتیم این پاسخگوی ما نبود زیرا می شد یک جوابی به دست آورده شود که تابع هدف ما برابر صفر شود ($y(i,j) = m(i,j)v_k$) که این جواب تغییرات شدیدی دارد چون در اینجا $m(i,j)$ رابطه مستقیم با (j,i) پیدا می کند که این مخالف فرض ما یعنی slow variation است. برای تحمیل تغییرات آرام به الگوریتم 2 ترم دیگر را اضافه شده است و بنابراین تابع ما به شکل بالا خواهد شد. در واقع 2 ترم آخر اجازه نمی دهند که تغییرات شدیدی داشته باشد چون اگر تغییرات شدید داشته باشد این 2 ترم بزرگ می شود و جلوی مینیمم شدن را می گیرند. برای اجرای الگوریتم باید :

-1- حدس اولیه برای همه پارامتر ها باید زده شود.

Initial guess for $\{v_k\}_{k=1}^C$ using k-means, FCM, ... and set $m(i,j) = 1$

Membership -2 ها را به صورت زیر محاسبه می کنیم :

$$u_k(i,j) = \frac{\|y(i,j) - m(i,j)v_k\|^{-2}}{\sum_{l=1}^C \|y(i,j) - m(i,j)v_l\|^{-2}}$$

به تعداد پیکسل های تصویر یک محاسبه ساده انجام می شود. در واقع تعداد محاسبات زیاد است ولی هر محاسبه کم است.

-3- حال centroid ها را محاسبه می کنیم:

$$v_k = \frac{\sum_{(i,j)} u_k^2(i,j) m(i,j) y(i,j)}{\sum_{(i,j)} u_k^2(i,j) m^2(i,j)}$$

تعداد محاسبات کم ولی هر کدام از محاسبات سنگین است.

-4- M را محاسبه می کنیم. فرمول کلی به صورت زیر است

$$y(i,j) \sum_{k=1}^C u_k^2(i,j) v_k = m(i,j) \sum_{k=1}^C u_k^2(i,j) v_k^2 + \lambda_1(m(i,j) * H_1(i,j)) + \lambda_2(m(i,j) * H_2(i,j))$$

where $H_1(i,j) = D_i * \tilde{D}_i + D_j * \tilde{D}_j$ and $H_2(i,j) = D_{ii} * \tilde{D}_{ii} + 2(D_{ij} * \tilde{D}_{ij}) + D_{jj} * \tilde{D}_{jj}$, $\tilde{f}(i) = f(-i)$

دستگاه معادله ای که به دست می آید بسیار بزرگ است بنابراین با نوشتن مسئله به فرم زیر خواهیم داشت :

$$y \in \mathbb{R}^{M \times N} \text{ then } f \in \mathbb{R}^{MN}, m \in \mathbb{R}^{MN} \text{ and } A \in \mathbb{R}^{MN \times MN}$$

در واقع این گام خودش یک حلقه درونی دارد و این امکان وجود دارد که آن را با روش *weighted jacobi iteration* حل نمود. بنابراین داریم :

$$m^{(i+1)} = [(1 - \omega)I - \omega D^{-1}(L + U)]m^{(i)} + \omega D^{-1}f, \quad A = D + L + U, \omega^* = \frac{2}{3}$$

5- مراحل 2 و 3 و 4 را تکرار می کنیم .

سوال 2

LOWPASS FILTERING

از آنجایی که H به آرامی در حوزه تصویر تغییر می کند، طیف آن در حوزه فرکانس در فرکانس پایین مرکز خواهد بود. بنابراین H می تواند توسط یک فیلتر L *low pass* جدا شود. بعد از فیلترینگ در *log-domain* خواهیم داشت :

$$L\{y\} \approx \beta.$$

این روش برای تصحیح H شبیه همومورفیک فیلترینگ در پردازش تصویر دیجیتال برای تصحیح ناهمگنی روشنایی است. با توجه به سادگی و کارایی این روش در اجرا، روش های فیلتر پایین گذر به طور گسترده ای مورد استفاده قرار می گیرند و این یکی از مزایای استفاده از آن ها است. همچنین مشخص شده است که این روش ها باید برای جلوگیری از *intensity distortion* و آرتیفکت های مصنوعی دقت به خرج داد. برای تصاویر MR ، با توجه به همپوشانی طیف بین داده های بیمار و H ، اثربخشی اکثر فیلترهای پایین گذر معمولی در حذف H به طور کلی کاملاً محدود است و چندان نمی توان روی آن حساب کرد.

STATISTICAL MODELING

روش های آماری فرض می کنند که IH از یک توزیع مانند توزیع گاوسی پیروی می کند یا IH را به عنوان یک فرآیند تصادفی، مانند مارکوف، مدل کند. یکی از این روش ها روش bayesian است. قاعده بیز اغلب برای تخمین IH map استفاده می شود. (هنگامی که IH توسط یک توزیع مدل شده است). اگر $\beta = (\beta_1, \dots, \beta_n)$ یک بردار زندوم باشد. برای پیش بینی β می توان به صورتی انتخاب کرد که بتواند احتمال مشروط $y|\beta$ را ماکسیمم کند. یعنی :

$$\hat{\beta} = \max_{\beta} p(\beta | y).$$

روشی که استفاده می شود این است که توزیع گوسی را برای مدل کردن log-transformed استفاده کنیم bias field

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

where Γ_i is the tissue class at voxel i with mean value $\mu(\Gamma_i)$

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

$$p(y | \beta) = \prod_i p(y_i | \beta_i)$$
$$= \prod_i \sum_v p(y_i | \Gamma_i, \beta_i) p(\Gamma_i).$$

هنگامی که داده های تصویر توسط IH آلوده نشده اند، متدهای طبقه بندی بافت با استفاده از mixture gaussian model است

از این رو، این روش اساساً تصحیح IH را با یک طبقه بندی کننده گوسی در هم می آمیزد.

مشاهده شد که اثر الگوریتم تصحیح IIIH به طور قابل توجهی تحت تاثیر طبقه‌بندی کننده گوسی قرار دارد.

در تصاویر واقعی، خیلی ممکن هست که هیستوگرام از **guassian mixture** انحراف داشته باشد و این کار را برای استفاده از این مدل کمی سخت می‌کند. برای فائق آمدن بر چنین مشکلی از رابطه زیر استفاده می‌شود :

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

با این تغییر IIIH فقط با توجه به **guassian classes** برآورد می‌شود. استفاده از کامپوننت غیر گوسی طبقه‌بند گوسی را این گونه می‌کند که کمتر توسط **outlier** های تصویر تحت تاثیر قرار گیرد.

Spatial modeling

از آنجایی که IIH به آرامی در حوزه تصویر تغییر می‌کند ، در نتیجه مقادیر IIH در پیکسل‌های کنار هم به هم نزدیک می‌باشد. با توجه به این کورلیشن مکانی می‌تاون از **Markov random field** استفاده کرد. برای این قسمت داریم :

$$p(y) \propto \exp \{ -U(y) \}$$

with the Gibbs energy

$$U(y) = \alpha \sum_{(i,j)} (y_i - y_j)^2 + \beta \sum_i y_i^2,$$

where $\langle i, j \rangle$ sums over every voxel i and its neighbours j .

بر عکس سایر روش ها که از یک گام classification استفاده می کنند این روش میدان III را جستجو می کند تا محتوای فرکانس را به حداقل برساند. این روش ساده می کند مشکل در مشکل log-domain را به عنوان یک مسئله decovolution مشکل در log-domain deconvolution توسط درک اینکه اگر v1 و v2 دو متغیر تصادفی مستقل باشند با توزیع های V1 و V2، به ترتیب، سپس توزیع مجموع آنها کانولوشن V1 و V2 است. برای محدود کردن فضای راه حل، فیلد III به صورت یک توزیع گاوی با واریانس کوچک مدل سازی شده است.

بخش شبیه سازی

سوال 1

(الف) می دانیمتابع هزینه این روش به صورت زیر است :

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

با قید $\sum_{i=1}^c u_{ik} = 1$ که مسئله به صورت زیر می شود:

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

که در آن membership u_{ik} همان u_{ik} می باشد که بیانگر این است که پیکسل k ام با چه احتمالی به خوشه i متعلق است. همچنین class prototype v_i می باشد که همان مرکز خوشه ها می باشد. bias filed نیز همان اثر میدان است که در فرمول بندی آن را از تبدیل لگاریتمی عبور داده ایم. در واقع مدل ضربی را بعد از عبور از تبدیل لگاریتمی به جمع تبدیل کرده ایم :

$$Y_k = X_k G_K \quad \forall k \in \{1, 2, \dots, N\}$$

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

که β_k bias-field نام دارد و G_k gain-field نام دارد. اگر بخواهیم رابطه را بررسی کنیم می دانیم که در واقع ترم دوم همان ترم اول است فقط به ازای هر (j, i) ترم دوم روی یک همسایگی اطرافش انجام می دهد. در واقع این ترم باعث همواری می شود.

حال با استفاده مباحث بهینه سازی و با استفاده از لاغرانژین داریم :

$$F_m = \sum_{i=1}^c \sum_{k=1}^N \left(u_{ik}^p D_{ik} + \frac{\alpha}{N_R} u_{ik}^p \gamma_i \right) + \lambda \left(1 - \sum_{i=1}^c u_{ik} \right)$$

where $D_{ik} = \|y_k - \beta_k - v_i\|^2$ and $\gamma_i = (\sum_{y_r \in \mathcal{N}_k} \|y_r - \beta_r - v_i\|^2)$.

با مشتق گیری از هر کدام از مجهولات داریم :

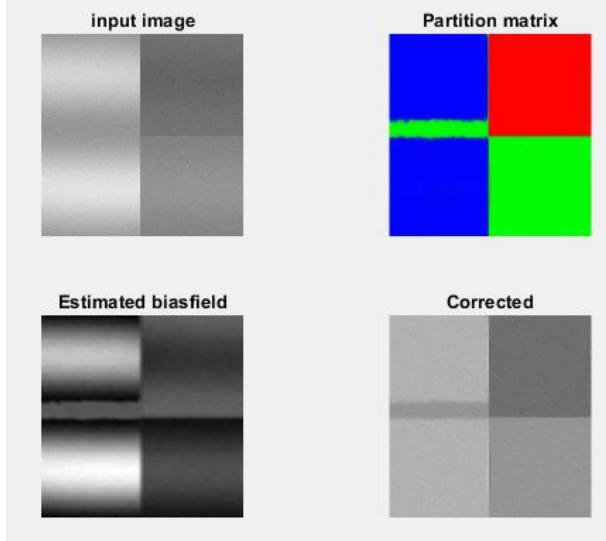
$$u_{ik}^* = \frac{1}{\sum_{j=1}^c \left(\frac{D_{ik} + \frac{\alpha}{N_R} \gamma_i}{D_{jk} + \frac{\alpha}{N_R} \gamma_j} \right)^{1/(p-1)}}$$

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

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

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

ب) برای این قسمت داریم:



سوال 2

الف) روش مرسوم $x(s) = [x(s), y(s)]$ curve ، snake مینیمم کردن رابطه زیر می باشد:

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

که در آن α و β پارامترهای وزن دهی می باشند که rigidity و tension را کنترل می کنند. $(x'(s)$ و $(x''(s)$ مشتق های اول و دوم می باشند. همچنین E_{ext} از تصویر به دست می آید به صورتی که مقادیر کمترش را در ویژگی هایی نظیر مرزها بگیرد. انرژی های مرسوم به صورت زیر تعریف می شود:

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

که $G_\sigma(x, y)$ یک تابع 2 متغیره گوسی می باشد. مقادیر بزرگتر σ باعث می شوند که مرزها تار شوند. E را مینیمم می کند باید در معادل زیر صدق کند :

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

یا

$$\mathbf{F}_{\text{ext}}^1 = -\nabla E_{\text{ext}}, \quad \mathbf{F}_{\text{int}} = \alpha \mathbf{x}''(s) - \beta \mathbf{x}'''(s) \quad \text{و} \quad \mathbf{F}_{\text{int}} + \mathbf{F}_{\text{ext}}^1 = \mathbf{0}$$

که در این رابطه، نیروهای داخلی از کش آمدن و تاشدن خم جلوگیری میکنند و نیروهای خروجی خم را به سمت ویژگی موردنظر یعنی لبه سگمنتها میکشند. میتوان رابطه بالا را تابعی از s و t در نظر گرفت.

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

وقتی معادله زیر حل می شود که که در معادله بالا عبارت $\mathbf{x}_t(s, t)$ صفر شود. این معادله همچنین می تواند با استفاده از الگوریتم گرادیان کاهشی حل شود. معادله بالا را می توان با استفاده از گستته سازی و حل سیستم گستته به صورت تکرار شونده محاسبه نمود.

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

(ب) GVF در واقع میدان برداری $\mathbf{v}(x, y) = (u(x, y), v(x, y))$ که تابع زیر را مینیمم می کند:

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

این فرمول بندی از یک اصل استاندارد پیروی می کند، که زمانی که داده ای وجود ندارد، نتیجه را smooth کند. به طور خاص، می بینیم وقتی که ∇f کوچک است، انرژی تحت تاثیر مشتقات

جزئی میدان برداری، یک میدان ∇f را نتیجه می دهد. در طرف دیگر وقتی ∇f بزرگ است، عبارت دوم بر انتگرال غالب است و با تنظیم $\nabla f = v$ به حداقل می رسد. پارامتر μ نیز یک پارامتری است که trade-off بین ترم اول و دوم معادله بالا را تنظیم می کند و باید بر اساس مقدار نویز موجود در تصویر تعیین شود. f نیز یک edge map است که از انرژی ها معرفی شده در قسمت قبل می توان برای آن استفاده کرد.

با استفاده calculus variations می توان نشان داد که GVF می تواند محاسبه شود با استفاده از حل کردن معادلات اویلر زیر:

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

اگر دقت کنیم می بینیم در رابطه بالا در نواحی هموار که مستقایات جزئی میدان برداری صفر است $\mu \nabla^2 v = 0$ و $\mu \nabla^2 u = 0$ که بیانگر رابطه لاپلاس است و نشان می دهد در نواحی هموار u, v یک درون یابی از نقاط اطراف خود هستند و درنتیجه لزو ما صفر نمی شوند.

این معادلات شهود دیگری درباره فرمول GVF به ما می دهند. در واقع باید به این نکته توجه کرد که در مناطق همگن، جمله دوم هر دو معادله بالا صفر است بنابراین، در این مناطق، u و v هر کدام با معادله لاپلاس تعیین می شود. این منجر به یک نوع از "filling in" اطلاعات گرفته شده از مرزهای region می شود. معادلات بالا می تواند به این صورت فرض شود که u, v تابع زمان هستند و پاسخ steady state به دست آورده شود :

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

باید به این نکته توجه کرد که این معادلات decouple شده اند، و بنابراین می توان آن را به صورت معادلات جزئی اسکالر جداگانه حل کرد معادلات بالا معادلات انتشار تعمیم یافته شناخته می شوند و شناخته می شوند در زمینه های مختلفی مانند هدایت گرما، راکتور بوجود می آیند فیزیک ظاهر شده اند. رابطه جدید به صورت زیر می شود:

$$\mathbf{x}_t(s, t) = \alpha \mathbf{x}''(s, t) - \beta \mathbf{x}'''(s, t) + \mathbf{v}$$

(ج) برای معرفی پارامتر ها داریم :

Parameter s	Description	Parameters	Description
I	تصویر ورودی	P	نقاط اولیه تعیین شده توسط کاربر
Kappa	وزن نیروی خارجی عکس	nPoints	تعداد نقاط خم
Gamma	گام زمانی در آپدیت کردن روابط با استفاده از گرادیان کاهشی	Iterations	تعداد ایتریشن الگوریتم
Sigma1	برای مشتق‌گیری از سیگنال استفاده می‌شود	Wline	میزان جذب شدن به خط‌ها را تعیین می‌کند. اگر منفی باشد به خطوط سیاه و در غیر این صورت به خطوط سفید جذب می‌شود.
Wedge	میزان جذب شدن به لبه‌ها را تعیین می‌کند	wterm	میزان جذب شدن به انتهای خطوط و یا گوش‌ها
Sigma2	برای محاسبه E_{ext}^2 استفاده می‌شود.	Giterations	تعداد ایتریشن برای حل معادلات 11b و 11a
Sigma3	برای محاسبه لاپلاسین‌ها در معادله 11a و 11b	alpha	α در معادله 7 انرژی membrane
beta	در معادله 7 انرژی thin plate	Delta	نیروی بالن

mu	μ در بالا در باره آن توضیح داده شده است. در واقع trade-off بین بردارهای لبه واقعی و بردارهای نویز را تعیین می کند	verbose	عکس های مهم مانند تکامل خم، خم اولیه، E_{ext} و میدان برداری را نشان می دهد. (به صورت دیفالت False است)
----	---	---------	---

حال که به معرفی پارامتر های مربوطه پرداختیم خود تابع را مورد بررسی قرار می دهیم . بعد از makecontureclockwise2d کردن ورودی ها و initialization اولیه از تابع assign استفاده می کنیم که یک کانتور در جهت عقربه های ساعت ایجاد می کند. این کار با محاسبه مساحت داخل کانتور انجام می شود اگر مثبت باشد جهت کانتور را تغییر می دهیم. سپس P را به تابع InterpolateContourPoints2D می دهیم. این تابع در واقع چند نقطه که یک کانتور را توصیف می کنند به یک کانتور smooth با نقاط نمونه برداری یکنواخت resample می کند. حال با استفاده از External energy image تصویر را به یک ExternalForcelImage2D تبدیل می کنیم. منظور از این تابع این است که عملیات ها مشتق گیری را انجام می دهد و در نهایت مطابق زیر عمل می کند که مطابق تئوری مسئله می باشد :

```

Eline = imgaussian(I,Sigma);
Eterm = (Iyy.*Ix.^2 - 2*Ixy.*Ix.*Iy + Ixx.*Iy.^2)./((1+Ix.^2 + Iy.^2).^(3/2));
Eedge = sqrt(Ix.^2 + Iy.^2);
Extern= (Wline*Eline - Wedge*Eedge -Wterm * Eterm);

```

سپس از خروجی این تابع مشتق هایی در جهت X و Y گرفته می شود و با استفاده از آن ها Fext را می سازیم. این کار به صورت زیر انجام می شود:

```

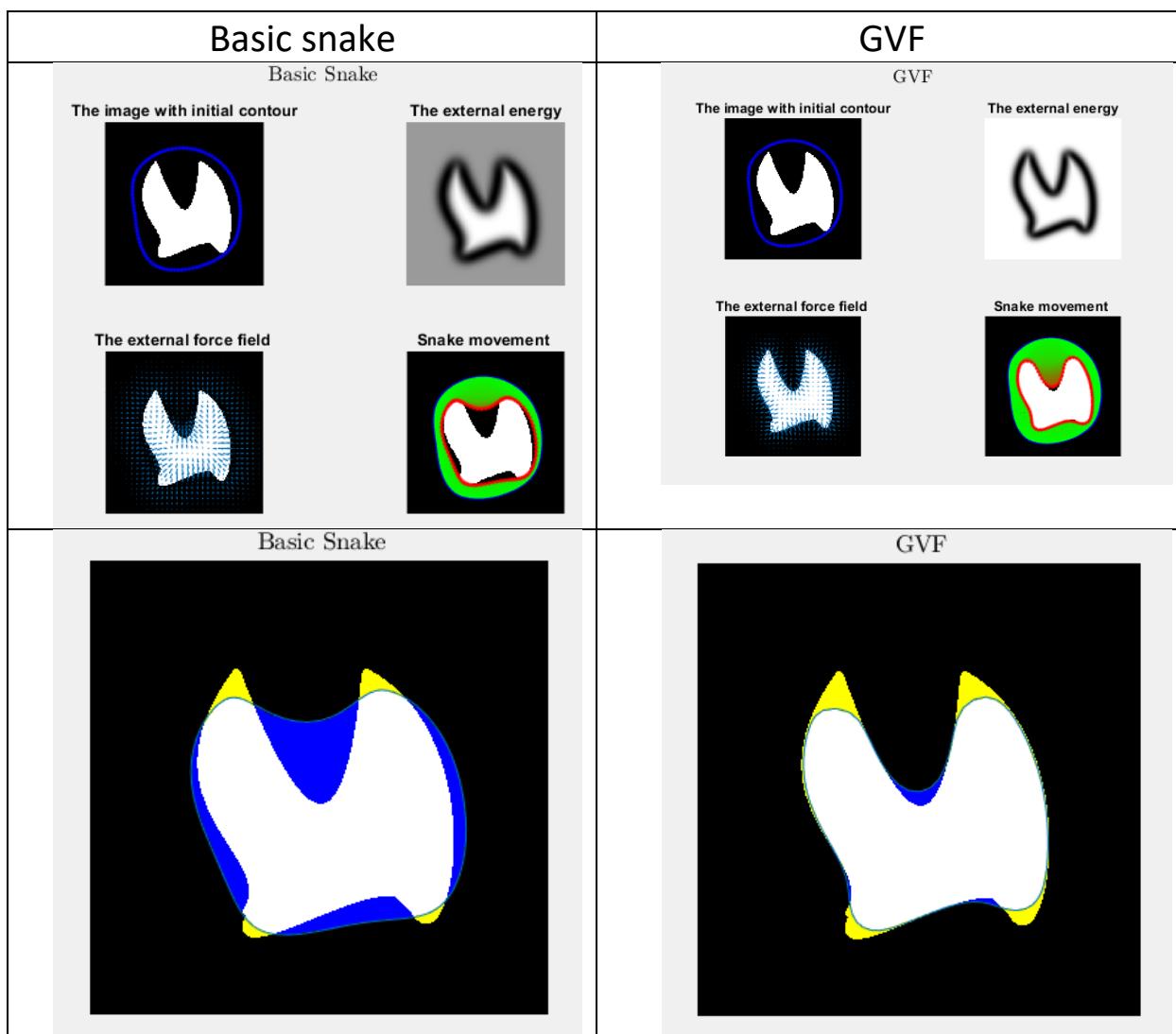
Fext(:,:,1)=-Fx*2*Options.Sigma2^2;
Fext(:,:,2)=-Fy*2*Options.Sigma2^2;

```

حال و با استفاده از تابع GVFOptimizelImageForces2D معادلات 12a و 12b را به SnakeInternalForceMatrix2D صورت iterative محاسبه می کنیم. حال با استفاده از تابع

ماتریس snake smoothness regularization را به دست می آوریم. با استفاده از تابع SnakeMovement2D برای یک ایتریشن contour snake حرکت iteration های مورد نظر را محاسبه می کنیم. در نهایت پس از انجام برای DrawSegmentedArea2D کانتور را به صورت یک خط بسته در یک تصویر رسم می کنیم و داخل کانتور را با True imfill می کنیم.

۵) برای حل این سوال داریم :



همانطور که می بینیم روش **snake basic** نمی تواند در شکل هایی که فرورفتگی دارند خوب عمل کند زیرا نمی تواند در واقع برای دره ها گرادیان مورد نیاز را تامین کند. از با مقایسه GVF با متوجه می شویم که این روش برای شکل هایی که دارای فرورفتگی هستند نسبت به **basic snake** عملکرد بهتری دارد. این مشکل از آن جا رفع شده است که میدان برداری جدیدی در این روش تعریف کرده ایم که مشکل شکل های دارای فرورفتگی را از بین می برد.