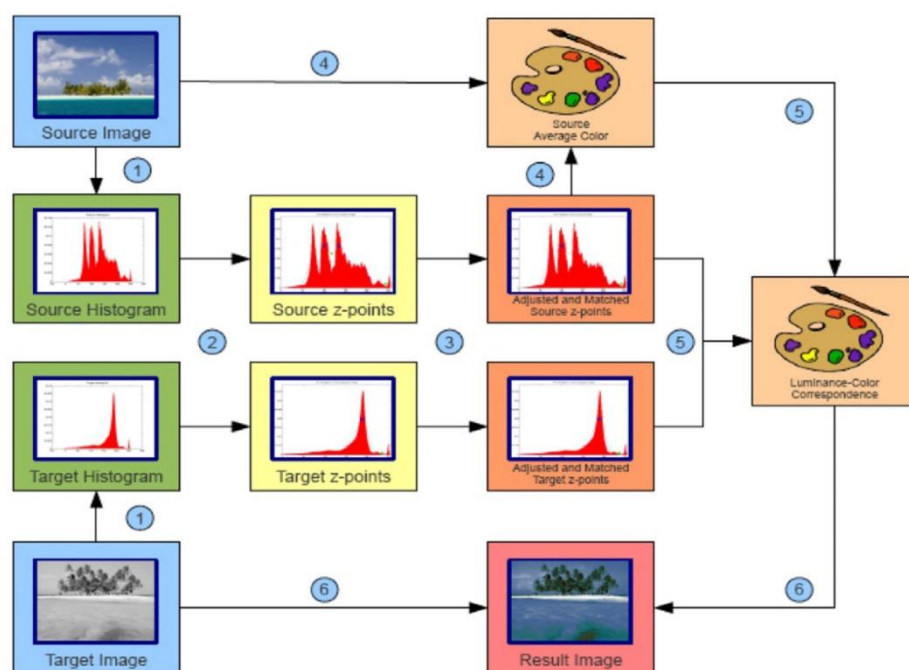


#### ۱-۴- بررسی اجمالی رگرسیون هیستوگرام بر اساس متد رنگ آمیزی اتوماتیک

در این الگوریتم، برای اولین بار خطی به صورت محلی وزن رگرسیون در نمودار هیستوگرام روشنایی هر دو منبع و تصویر هدف انجام داده شد. با دامنه‌ها در نتیجه‌ی رگرسیون، ما می‌توانیم نقاط صفر (به عنوان مثال، حداکثر و حداقل محلی) از هیستوگرام تقریبی را شناسایی کنیم و سپس این نقاط را طوری تنظیم کنیم که بین منبع و مقصد همسان شوند. سپس، با رنگ متوسط محاسبه شده از تصویر منبع، می‌توانیم تطابق روشنایی-رنگ را برای تصویر موردنظر بدست آوریم و وزن‌ها توسط نقاط صفر تنظیم شده تعیین می‌شوند. نتیجه‌ی رنگ آمیزی توسط نقشه برداری از این تطابق درخشندگی-رنگ با تصویر مورد نظر به طور مستقیم به دست می‌آید. شکل ۱-۴ چارچوب الگوریتم را به ما می‌دهد.



شکل ۱-۴- چارچوب الگوریتم

#### ۲-۴- رگرسیون وزن دار شده‌ی محلی بر هیستوگرام‌ها

هیستوگرام یک تصویر نشان دهنده‌ی تعداد پیکسل‌ها در فواصل روشنایی مختلف می‌باشد. اگر ما می‌توانستیم نقاط همسان را در دو هیستوگرام بین تصویر سیاه و سفید و تصویر منبع پیدا کنیم، به دست آوردن یک راه حل برای مطابقت رنگ‌های تصویر سیاه و سفید به رنگ‌های تصویر منبع امکان پذیر است. برای تحقق مداوم این تطابق، بایستی هیستوگرام را با یک منحنی توصیف کنیم. در اینجا، ما رگرسیون وزن

دار شده‌ی محلی را به هیستوگرام‌های تصویر منبع و تصویر سیاه و سفید برای ایجاد منحنی‌های مناسب بکار می‌بریم (شکل ۴-۲ را ببینید). در زیر، ما این روش را مورد بحث قرار می‌دهیم.

رگرسیون وزن دار شده‌ی محلی رگرسیون را حول یک نقطه با استفاده از داده‌ی آموزش که برای آن نقطه محلی هستند انجام می‌دهد. این یک رویکرد غیر پارامتری است که صراحتاً داده‌های آموزش را حفظ می‌کند، و می‌توان آن را هر بار که باید یک پیش بینی صورت گیرد، مورد استفاده قرار داد. این یک مشکل به حداقل رسانی است که می‌تواند بصورت زیر بیان شود:

$$\theta = \arg \min \sum_{j=1}^m w_j (\theta^T X_j - Y_j) \quad (1-4)$$

بطوری که:

$$\theta = \begin{pmatrix} \theta_0 \\ \theta_1 \end{pmatrix}, X_j = \begin{pmatrix} X_{j0} \\ X_{j1} \end{pmatrix} \quad (2-4)$$

در اینجا،  $(Y, X)$  و  $\theta$  به ترتیب داده‌های آموزش و داده‌های مناسب را نشان می‌دهند،  $(X_j, Y_j)$  نمونه‌ی  $j$ ام داده‌های آموزشی را نشان می‌دهد و  $m$  اندازه‌ی داده آموزش است. بردار  $\theta$  نتایج رگرسیون برای یک  $j$  خاص است،  $\theta_0$  تقاطع  $y$  در  $j$ ، و  $\theta_1$  شیب در  $j$  می‌باشد.  $W_j$  یک وزن دار شده‌ی غیر منفی است. بطور مستقیم، اگر  $W_j$  برای یک مقدار خاص از  $j$  بزرگ باشد، ما هنگام انتخاب  $\theta$  سعی می‌کنیم  $(\theta^T X_j - Y_j)$  را کوچک کنیم. از طرف دیگر، اگر  $W_j$  کوچک باشد،  $(\theta^T X_j - Y_j)$  در روند تناسب رد خواهد شد.  $W_j$  بصورت زیر محاسبه می‌شود:

$$W_j = \exp\left(-\frac{\|X_j - X\|^2}{2\tau^2}\right) \quad (3-4)$$

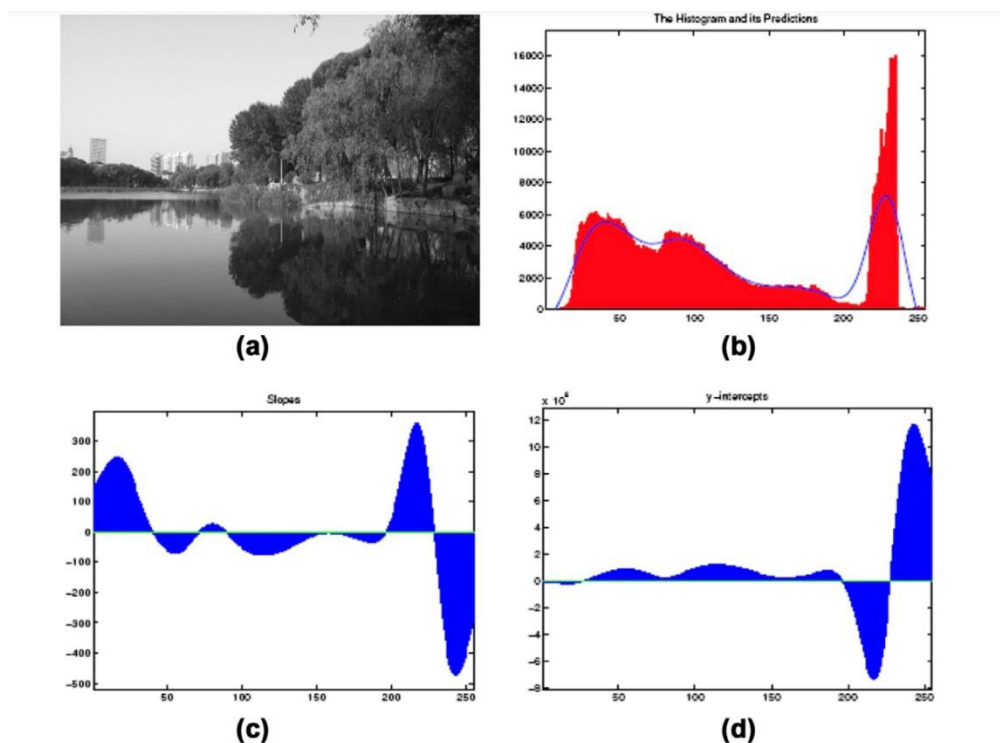
بطوری که  $\tau$  در آزمایشات ما ثابت است. مقدار آن ۱۰ می‌باشد. در حالت رنگ آمیزی ما، هیستوگرام یک تصویر را بصورت  $Y_j$  مشخص می‌کنیم و

$$X_j = \begin{pmatrix} 1 \\ j \end{pmatrix} \quad (4-4)$$

فرض کنید که روشنایی در یک فاصله‌ی  $[0, 255]$  نمایش داده شود،  $Y_j$  تعداد پیکسل‌هایی را نشان می‌دهد که روشنایی آنها  $j$  می‌باشد. همانطور که در فرمول ۴-۱ نشان داده شده است، رگرسیون وزن دار شده‌ی محلی برای اجرای رگرسیون محلی برای هر داده‌ی  $X$  با یک سنجش وزن دار شده‌ی آثار سایر داده‌ها بر پروسه‌ی تناسب آن است. با تعریف دو ماتریس دیگر به شکل زیر، نتیجه‌ی رگرسیون وزن دار شده‌ی محلی می‌تواند بصورت  $\theta = \bar{X}^T \bar{Y} (\bar{X}^T \bar{X})^{-1}$  بدست آید.

$$\bar{X} = \begin{bmatrix} X'_1 \\ X'_2 \\ \vdots \\ X'_m \end{bmatrix}^T, \bar{Y} = \begin{bmatrix} Y'_1 \\ Y'_2 \\ \vdots \\ Y'_m \end{bmatrix} \quad (4-5)$$

در اینجا،  $X'_j = \sqrt{W_j} X_j$  و  $Y'_j = \sqrt{W_j} Y_j$ ، نتایج رگرسیون وزن دار شده‌ی محلی را نشان می‌دهد. شکل 2a یک تصویر سیاه و سفید است. شکل 2b هیستوگرام تصویر سیاه و سفید و منحنی تناسب را نشان می‌دهد که به رنگ آبی کشیده شده است. شکل 2c و 2d مقادیر  $\theta_0$  و  $\theta_1$  را نشان می‌دهند که به ترتیب شیب‌ها و نقاط تقاطع با  $y$  منحنی تناسب را نمایش می‌دهند. با استفاده از این منحنی تناسب، ما می‌توانیم نقاط حداکثر هیستوگرام را بدست آوریم که برای قطعه بندی استفاده خواهند شد. در روش ما، نقاط حداکثر محلی از طریق محاسبه‌ی نقاطی که شیب صفر دارند، هنگامی که که شیب‌ها مانند اولین مشتقات عملکرد هیستوگرام رفتار می‌کنند، بدست می‌آیند.



شکل ۴-۲- نتایج عمل LWR بر هیستوگرام تصویر: (a) تصویر سیاه و سفید، (b) هیستوگرام و منحنی تناسب، (c) شیب‌ها و (d) نقاط تقاطع  $y$

با فرض این که محتویات تصویر منبع و تصویر سیاه و سفید مشابه هستند، ما می‌توانیم این تشابه را بصورت تشابه هیستوگرام‌های آنها تعبیر کنیم. در واقع، تشابه هیستوگرام‌ها می‌تواند بوسیله‌ی نقاط صفر آن‌ها، بطور اختصاصی‌تر تعداد حداکثرها و تغییرات حداقل - حداکثر آن بیان شود. ما تعداد بی‌نهایت‌ها و تغییرات حداقل - حداکثر دو هیستوگرام را یکسان با یکدیگر تنظیم می‌کنیم، بنابراین می‌توانیم رنگ‌های مشابه هر دو تصویر را مطابقت دهیم.

#### ۴-۳-۱- ایجاد نقاط صفر

نقاط صفر توسط تکرار تمام مقادیر روشنایی و تعیین اینکه آیا هیچ یک از آن نقاط، مینیمم یا ماکسیمم محلی است تولید می‌شوند. اگر یک نقطه، یک حد نهایی محلی باشد، اولین مشتق آن بایستی ۰ باشد. برای حالت گسسته‌ی نتایج رگرسیون، بایستی شیب‌هایی که بسیار به ۰ نزدیک هستند را بیابیم تا حد نهایی محلی امکان پذیر را تعیین کنیم. ما جفت نقاطی را که ارتباط مقایسه‌ای قراردادی با ۰ دارند را تعیین می‌کنیم و نزدیک‌ترین حد نهایی محلی را با انتخاب یک نقطه‌ی نزدیک‌تر به صفر در یک جفت منتخب پیدا می‌کنیم. اگر جفت منتخب حد نهایی محلی در حال بالا رفتن باشد، پس یک حداقل محلی است. در غیر اینصورت، یک حداکثر محلی می‌باشد.

از پروسه‌ی بالا دو توالی بدست می‌آوریم. یکی از آنها توالی نقاط صفر  $Z$  و دیگری توالی تناوبی حداکثر  $T$  می‌باشد. مقادیر  $T$  تناوب حداقل و حداکثر  $Z$  را بیان می‌کنند. اگر  $T(i) = 1$ ،  $Z(i)$  یک حداکثر محلی می‌باشد. اگر  $T(i) = -1$ ،  $Z(i)$  یک حداقل محلی است. قطعه بندی هیستوگرام می‌تواند مستقیماً توسط قطعه سازی هیستوگرام با این نقاط حداکثر محلی انجام شوند. ما این قطعه بندی را بصورت  $S$  نمایش می‌دهیم. بعدها ازین قطعه بندی‌ها برای رنگ آمیزی استفاده می‌کنیم. این پروسه‌ی تولید نقاط صفر به نام الگوریتم تولید نقاط صفر خوانده می‌شود.

هنگام کاربرد الگوریتم تولید نقاط صفر بالا به هیستوگرام‌های هردوی تصاویر خاکستری و منبع، می‌توانیم قطعات هیستوگرام را برای آنها بدست آوریم. ما به ترتیب  $Z_S$  و  $T_S$  را بعنوان توالی‌های نتیجه‌ی تصویر منبع علامت گذاری می‌کنیم. بطور مشابهی،  $S_S$  و  $S_T$  را برای قطعه بندی هیستوگرام برای تصویر منبع و تصویر سیاه و سفید علامت گذاری می‌کنیم.

اگر  $T_S = T_T$ ، محتویات تصویر منبع و تصویر سیاه و سفید مشابه هستند. این موضوع همچنین بدین صورت تعبیر می‌شود که شباهت هیستوگرام‌ها برخی شباهت‌ها را بین تصاویر نشان می‌دهد. در این شرایط، قطعات  $S_S$  و  $S_T$  بطور طبیعی و مرتب با یکدیگر مطابقت می‌یابند. در مقابل، اگر  $T_S \neq T_T$ ، ما بایستی توالی‌های  $T$  و  $Z$  را تنظیم کنیم تا برای رنگ آمیزی مطابقت یابند. ایده‌ی اولیه‌ی ما، حذف نقاط مجموعه با

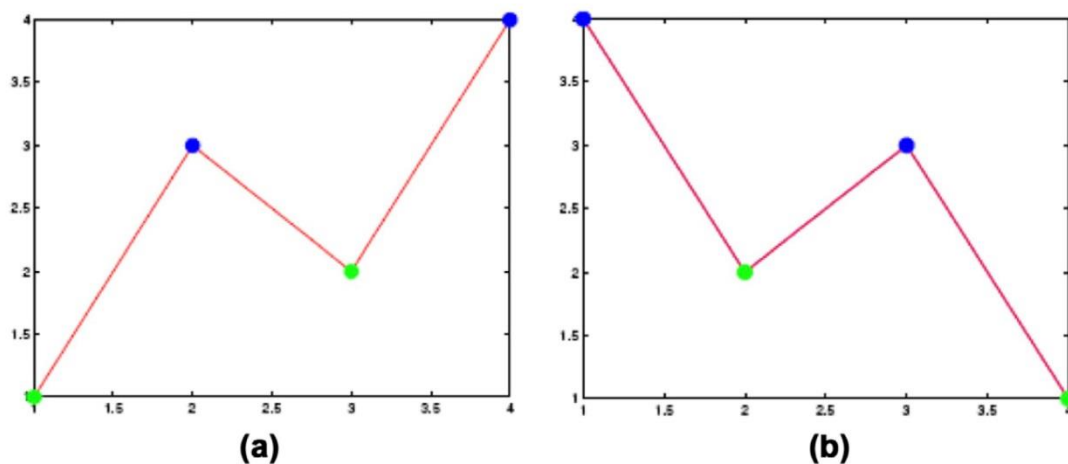
تعداد بزرگتری از نقاط صفر می باشد. همچنین با ثبت نقاط حذف شده، می توانیم شباهت دو تصویر را نشان دهیم. در ادامه، تنظیم مجموعه های نقاط صفر را بررسی می کنیم.

#### ۴-۳-۲- تنظیم نقاط صفر:

با توجه به توضیحات بالا، می دانیم که اساس توالی  $T$ ، تناوب ۱ و ۱- است. با شروع اولین عنصر در توالی  $T$ ، ۱ و ۱- به نوبت ظاهر می شوند. بنابراین به منظور تنظیم توالی  $T$ ، بایستی این خاصیت تناوب را حفظ کنیم. ما بایستی ابتدا بخش جلویی و عقبی مجموعه ها را تنظیم کنیم. این در صورتی ضروری است که تعداد تفاوت اندازه های توالی ها ثابت نیست. این الگوریتم، این کار را انجام می دهد. فرض کنید که  $A$  یک توالی یا یک مجموعه است،  $\delta(A)$  سایز  $A$  را نشان می دهد.

هدف تنظیم جلو-عقبی، یکسان سازی تفاوت اندازه ی دو توالی نقطه-صفر است، بنابراین بعداً می توانیم جفت نقاط را در توالی بزرگتر بدون تغییر ویژگی تناوبی آن تعیین کنیم. طرح الگوریتم کلی به شکل بالا نشان داده شده است. بطور اختصاصی، می تواند بصورت زیر خلاصه شود. اول، الگوریتم تطبیق جلو-عقب تست می کند که آیا اختلاف اندازه ی دو توالی غیر عادی است یا نه. اگر غیرعادی است، حذف نقطه ی جلویی یا عقبی مورد نیاز است. سپس، الگوریتم توالی بزرگتر را انتخاب می کند. بعلاوه، الگوریتم جلو و عقب توالی ها را مقایسه می کند. اگر هریک از آنها مشابه نبود، الگوریتم آن را در توالی بزرگتر حذف می کند. نتیجه ی این الگوریتم، تفاوت دو توالی را مساوی می کند. در نتیجه، ما می توانیم بعداً دو نقطه را بصورت همزمان از توالی بزرگتر بدون تغییر ویژگی تناوبی آن حذف کنیم.

پس از تنظیم جلو-عقب، ما نقاط صفر را در بخش جلویی و عقب توالی مطابقت می دهیم. اگرچه توالی ها هنوز منطبق نیستند، اختلاف بین اندازه های آنها مساوی است. ما می توانیم برای تطابق توالی ها، نقاط صفر را بطور مساوی از دسته ی بزرگتر حذف کنیم. ما یک الگوریتم تکرار شونده ی جدید را به اسم الگوریتم تطبیق جفتی برای درک آن پیشنهاد می کنیم که دو نقطه ی مجاور را در هر تکرار حذف می کند. حذف نقاط نباید بصورت اتفاقی انجام شود. روش انتخاب این که کدام جفت مجاور نقاط بایستی از هر تکرار حذف شوند، از طریق اثر آن بر تغییر توالی ها بدست می آید. ما جفت نقاطی را که حذف می کنیم که این گرایش را تغییر می دهند که توالی های نقاط-صفر  $Z$  تعداد کمتری از عناصر را دارند. شکل ۳-۴ جفت نقاط درون یک گرایش افزایشی یا کاهشی در  $Z$  را نشان می دهد. شکل های  $4-3a$  و  $4-3b$  به ترتیب نقاط صفر را در یک ترتیب افزایشی و کاهشی نشان می دهند. نقاط آبی و سبز به ترتیب نقاط حداکثر و حداقل محلی را در توالی نشان می دهند. همانطور که در شکل ۳-۴ دیده می شود، در هر دو مورد  $4-3a$  و  $4-3b$ ، جفت نقاط داخلی کاندید حذف هستند. اگر آنها را حذف کنیم، گرایش کلی توالی به میزان زیادی تغییر نمی کند.



شکل ۳-۴- جفت نقاط در یک روند: (a) نقاط صفر در روند افزایشی و (b) نقاط صفر در روند کاهشی

ما این نقاط را با مقایسه‌ی هر جفت نقطه با نقاط مجاور آن پیدا می‌کنیم. فرض کنید که داریم،  $Z_i, Z_{i+1} \in Z$  و  $g(Z_i) > g(Z_{i+1})$ . اگر  $g(Z_i) < g(Z_{i+2})$  و  $g(Z_{i-1}) < g(Z_{i+1})$ ، پس جفت نقاط  $(Z_i, Z_{i+1})$  دارای روند افزایشی است. بطور مشابهی، می‌توانیم روش مشابهی را برای یافتن جفت نقاطی که روند کاهشی دارند بکار ببریم. پس از یافتن تمام جفت نقاط، تفاوت‌ها را مقایسه می‌کنیم. جفت نقاط با کمترین تفاوت حذف خواهند شد. بدین معنی که برای یک جفت نقطه‌ی مجاور،  $Z_i, Z_{i+1} \in Z$ ، اگر  $|g(Z_i) - g(Z_{i+1})|$  کوچکترین باشد، این جفت نقاط در این تکرار الگوریتم حذف خواهند شد. حالتی دیگری نیز هست که هیچ جفت نقطه‌ی وجود ندارد. در این حالت، تنها جفت نقطه‌ی که کمترین اختلاف را دارند حذف می‌کنیم.

در پروسه‌ی تنظیم جفتی، ابتدا مقادیر حداقل و حداکثر روشنایی را به بخش جلو و عقب توالی بزرگتر اضافه خواهیم کرد، بنابراین پروسه‌ی پس از آن از نظم خارج نخواهد شد. قابل ذکر است که ما این دو نقطه را بعنوان کاندیدای حذف در نظر نمی‌گیریم. سپس، الگوریتم تکرار می‌شود تا زمانی که اختلاف بین اندازه‌ی توالی‌ها ۰ شود. در هر تکرار، اطمینان حاصل خواهیم کرد که یک جفت نقطه‌ی حذف شده وجود خواهد داشت. در هر تکرار، اگر جفت نقطه‌ی وجود داشته باشند که دارای روند باشند، ابتدا این جفت نقاط با کمترین اختلاف در مقادیر هیستوگرام را حذف می‌کنیم تا با نقاط صفر مطابقت یابند. اگر هیچ جفت دارای روندی وجود نداشته باشد، ما تنها جفت نقاط با کمترین تفاوت در مقادیر هیستوگرام را حذف می‌کنیم. روشی که با آن اطمینان حاصل می‌کنیم که آیا یک جفت از نقاط دارای نظام می‌باشد، انجام مقایسه‌ی آنها با یک جفت نقطه‌ی اضافی است که مجاور آنها هستند.

در مراحل قبلی، اختلاف اندازه‌های مجموعه‌های توالی‌ها را متعادل کردیم. بنابراین می‌توانیم دو نقطه را در هر تکرار حذف کنیم. الگوریتم تنظیم بخش جلویی و تنظیم جفتی با یکدیگر تحت عنوان الگوریتم تنظیم نقاط صفر نامیده می‌شوند. با این الگوریتم، ما می‌توانیم دو هیستوگرام را طوری تنظیم کنیم تا با یکدیگر تطابق یابند.

۴-۴- رنگی سازی وزن دار شده:

پس از اینکه ما هیستوگرام‌های منطبق تصویر منبع و تصویر سیاه و سفید را بدست آوردیم، تصویر سیاه و سفید را از طریق اقتباس رنگ از بخش‌های مرتبط در تصویر منبع بدست می‌آوریم. ما  $C$  را بعنوان رنگ،  $C(u, v)$  را بعنوان رنگ تصویر در مختصات  $(u, v)$ ، و  $U \times V$  را بعنوان سایز تصویر تعریف می‌کنیم. روشنایی رنگ  $C$  بصورت تابع  $L(C)$  نشان داده می‌شود.

الگوریتم رنگی سازی ما شامل دو مرحله است. اول، ما رنگ متوسط  $C(j)$  هر روشنایی  $j$  از تصویر منبع را محاسبه می‌کنیم. سپس، رنگی سازی را برای هر جفت قطعه‌ی هیستوگرام تطبیق یافته انجام می‌دهیم. مرحله‌ی اول با استفاده از یک عملیات میانگین قابل انجام است. ما برای راحتی  $C_S(j)$  را بعنوان رنگ متوسط محاسبه شده برای روشنایی  $j$  از تصویر منبع معنی می‌کنیم. برای مرحله‌ی دوم، ما از تابع وزن در رگرسیون وزن دار شده‌ی محلی اقتباس می‌کنیم تا یک الگوریتم وزن دار شده‌ی خاص ایجاد کنیم. برای هر نقطه با روشنایی  $j_T$  در تصویر سیاه و سفید، آن را با استفاده از میانگین وزن دار شده‌ی رنگ متوسط مشابه  $C_S(j_S)$  رنگ آمیزی خواهیم کرد. تابع وزن بصورت زیر نشان داده می‌شود:

$$w(x, \mu, \tau) = \exp\left(-\frac{\|x - \mu\|^2}{2\tau^2}\right), \quad (6-4)$$

$$\mu = (j - n_T) \frac{\delta(S_S)}{\delta(S_T)} + n_S, \quad \tau = \frac{\delta(S_S)}{\delta(S_T)} \quad (7-4)$$

بطوری که  $\mu$  و  $\tau$  به ترتیب مقدار مرکزی و پهنای باند هستند.  $n_T$  و  $n_S$  مقادیر حداقل  $S_T$  و  $S_S$  هستند که قطعه بندی روشنایی را نشان می‌دهند که با استفاده از رگرسیون در هیستوگرام بدست آوردیم. از فرمول بالا درمی‌یابیم که این رنگ متوسط وزن دار شده، یک پروسه‌ی ساده‌ی شبه گوسی برای تشابه رنگ-روشنایی است، اما بافت تصویر را تغییر نخواهد داد. تشابه رنگ-روشنایی وزن دار شده تنها یک میانگین وزن دار شده‌ی رنگ‌هاست که نشان می‌دهد دو مقدار روشنایی نباید دارای تغییر رنگ ناگهانی در نتیجه باشند. همچنین، امکان حفظ روشنایی تصویر اصلی با استفاده از یک فضای رنگی روشنایی مجزا وجود دارد (و این اطمینان را می‌دهد که بافت تحت تاثیر قرار نمی‌گیرد) مانند  $YCbCr$ ، که تنها با استفاده از کانال-

های رنگی محاسبه و رنگ آمیزی می‌کند، بطوری که  $Cb$  و  $Cr$  و نه  $Y$  را نگه می‌دارد. بطور خلاصه، برای دو قطعه بندی منطبق  $S_T$  و  $S_S$ ، فرمول رنگ آمیزی وزن دار شده می‌تواند بصورت زیر بیان شود:

$$C_T(j) = \frac{\sum_{i=1}^{255} w\left(\left(i, (j-n_T)\frac{\delta(S_S)}{\delta(S_T)} + n_S, \frac{\delta(S_S)}{\delta(S_T)}\right)\right) \cdot C_S(i)}{\sum_{i=1}^{255} w\left(\left(i, (j-n_T)\frac{\delta(S_S)}{\delta(S_T)} + n_S, \frac{\delta(S_S)}{\delta(S_T)}\right)\right)} \quad (۸-۴)$$

بطوری که  $C_T(j)$  رنگ میانگین تصویر سیاه و سفید است. در این الگوریتم، مقدار مرکزی و پهنای باند به ترتیب  $\mu$  و  $\tau$  هستند. این محاسبات می‌توانند تغییر شکل مقادیر را از  $S_T$  به  $S_S$  بصورت خطی انجام دهند. پهنای باند همچنین سادگی هیستوگرام‌ها را تضمین می‌کند. برای هر پیکسل  $(u, v)$  با روشنایی  $j$  در تصویر سیاه و سفید، داریم:  $C_T(u, v) = C_T(j)$

۴-۵- اجرا و نتایج:

پیاده سازی الگوریتم بر روی ۵ نمونه عکس آزمایش شده است.

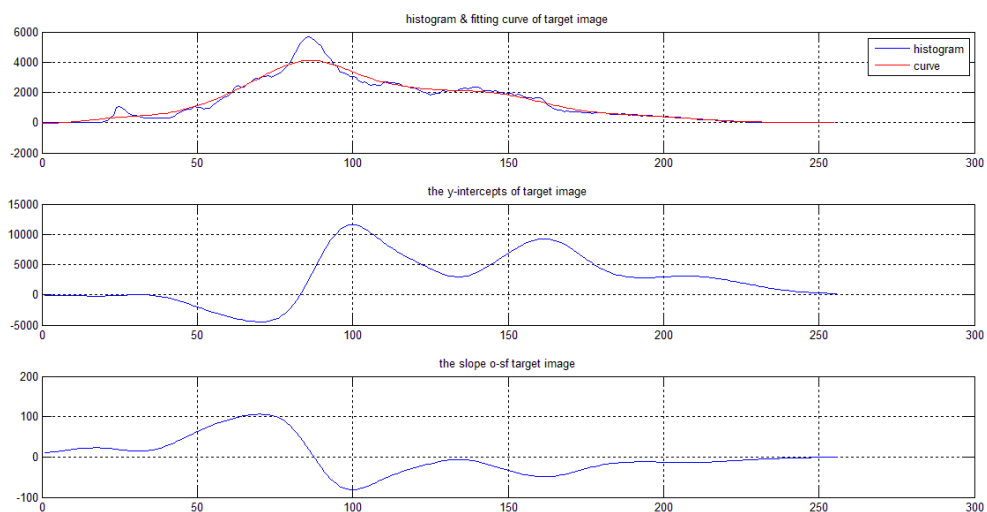
این آزمایشات بر روی سیستم های شخصی با سیستم عامل ۸ و ۶۴ بیتی و پردازنده 2.5GHz, Core i5 و رم ۴ گیگ انجام شده است.



نمونه اول:



شکل ۴-۴- تصویر هدف (تصویر خاکستری)



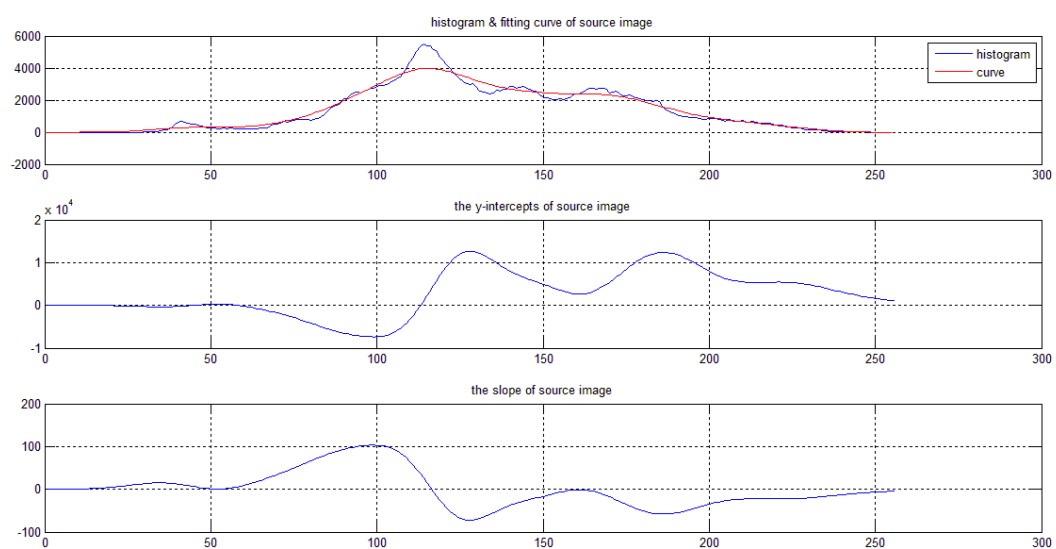
شکل ۴-۵- نمودار های هیستوگرام و منحنی تخمین زده شده تصویر هدف

نمودار  $\theta_0$  تصویر هدف

نمودار  $\theta_1$  تصویر هدف



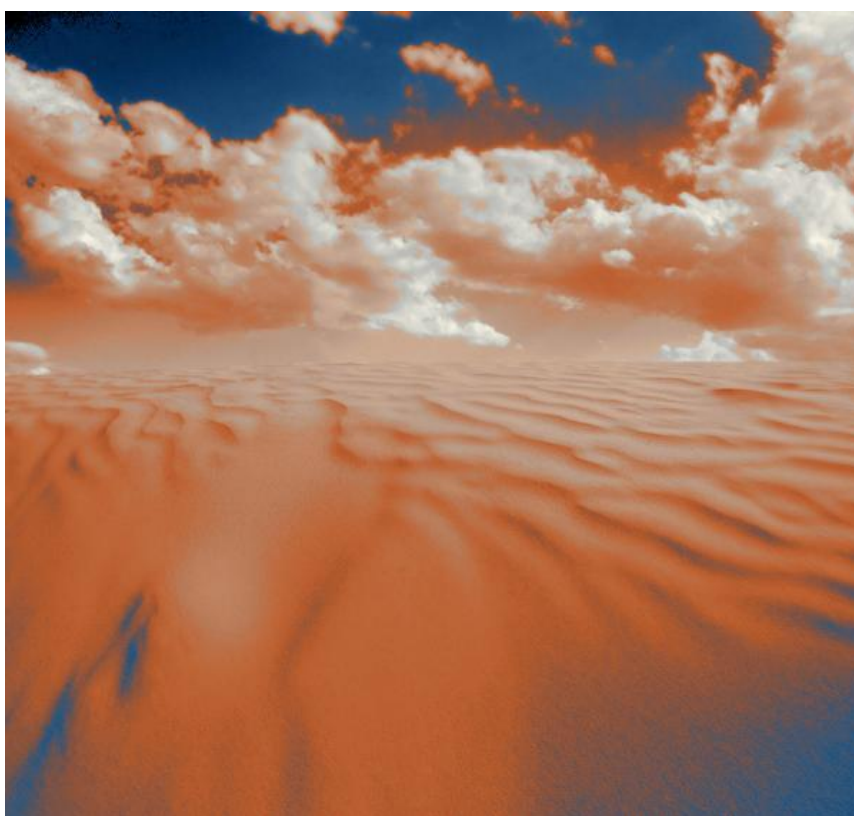
شکل ۴-۶- تصویر مرجع (تصویر رنگی)



شکل ۴-۷- نمودار های هیستوگرام و منحنی تخمین زده شده تصویر مرجع

نمودار  $\theta_0$  تصویر مرجع

نمودار  $\theta_1$  تصویر مرجع

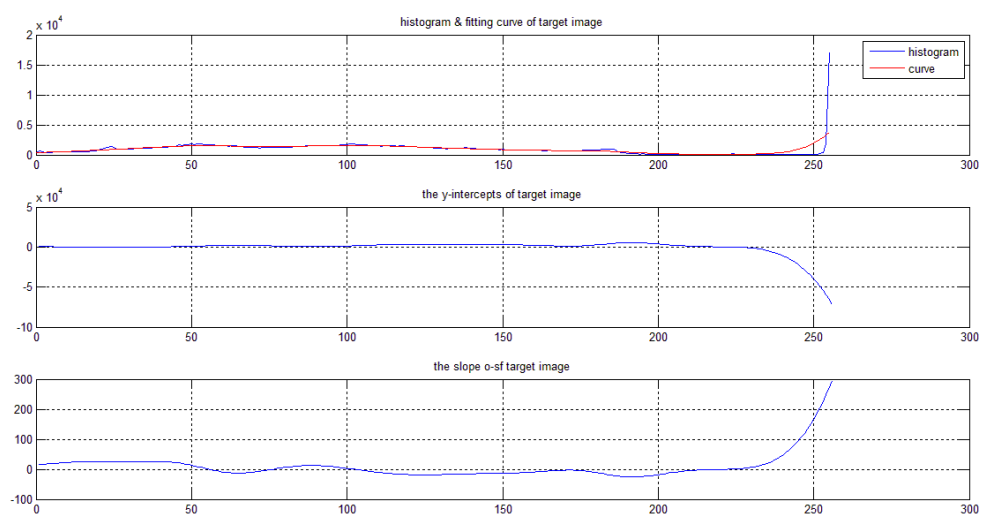


شکل ۴-۸- تصویر رنگدهی شده توسط الگوریتم

نمونه دوم:



شکل ۴-۹- تصویر هدف (تصویر خاکستری)



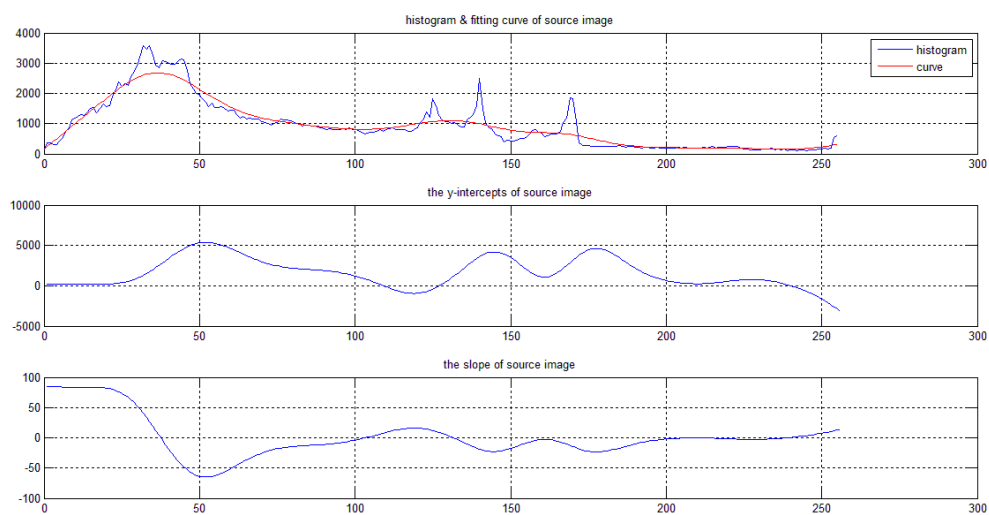
شکل ۴-۱۰- نمودار های هیستوگرام و منحنی تخمین زده شده تصویر هدف

نمودار  $\theta_0$  تصویر هدف

نمودار  $\theta_1$  تصویر هدف



شکل ۴-۱۱- تصویر مرجع (تصویر رنگی)



شکل ۴-۱۲- نمودار های هیستوگرام و منحنی تخمین زده شده تصویر مرجع

نمودار  $\theta_0$  تصویر مرجع

نمودار  $\theta_1$  تصویر مرجع



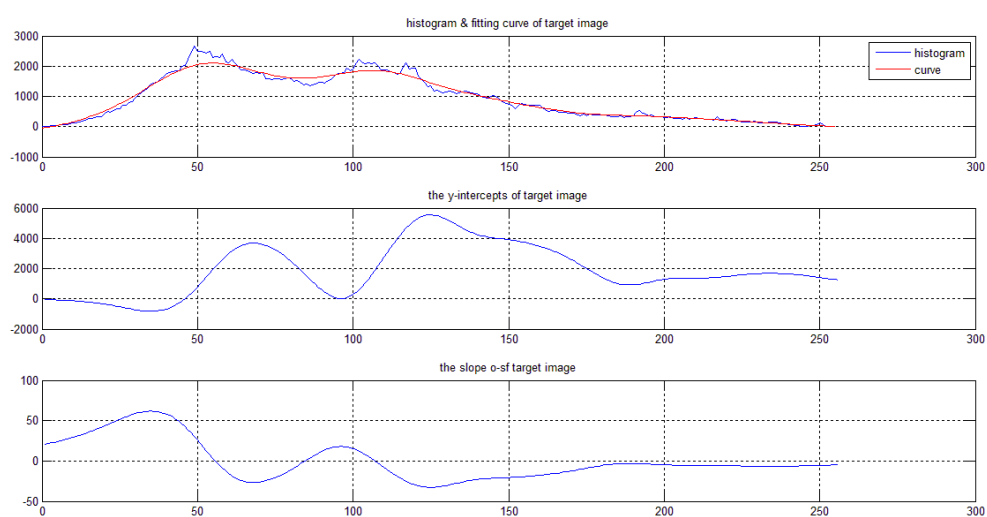
شکل ۴-۱۳- تصویر رنگدهی شده توسط الگوریتم



نمونه سوم:



شکل ۴-۱۴- تصویر هدف (تصویر خاکستری)



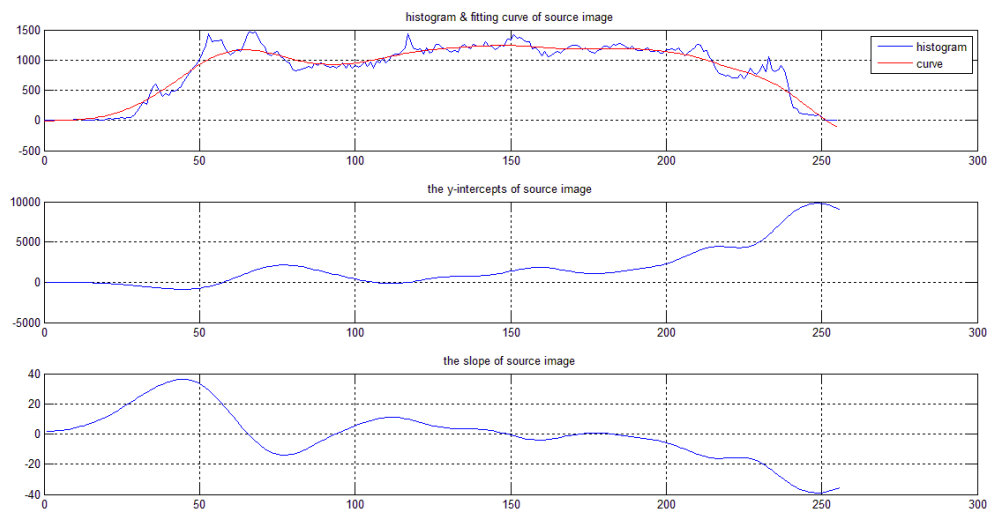
شکل ۴-۱۵- نمودار های هیستوگرام و منحنی تخمین زده شده تصویر هدف

نمودار  $\theta_0$  تصویر هدف

نمودار  $\theta_1$  تصویر هدف



شکل ۴-۱۶- تصویر مرجع (تصویر رنگی)



شکل ۴-۱۷- نمودار های هیستوگرام و منحنی تخمین زده شده تصویر مرجع

نمودار  $\theta_0$  تصویر مرجع

نمودار  $\theta_1$  تصویر مرجع



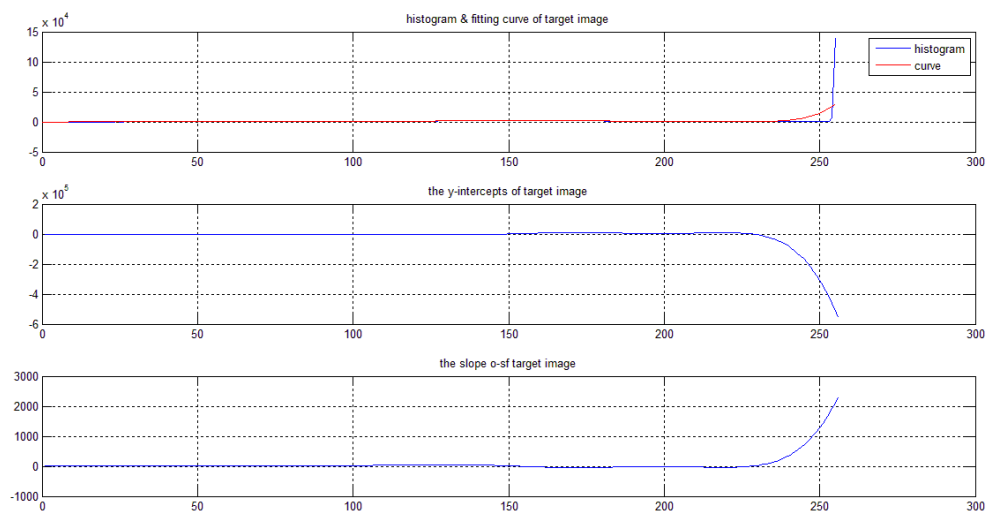


شکل ۴-۱۸- تصویر رنگدهی شده توسط الگوریتم

نمونه چهارم:



شکل ۴-۱۹- تصویر هدف (تصویر خاکستری)



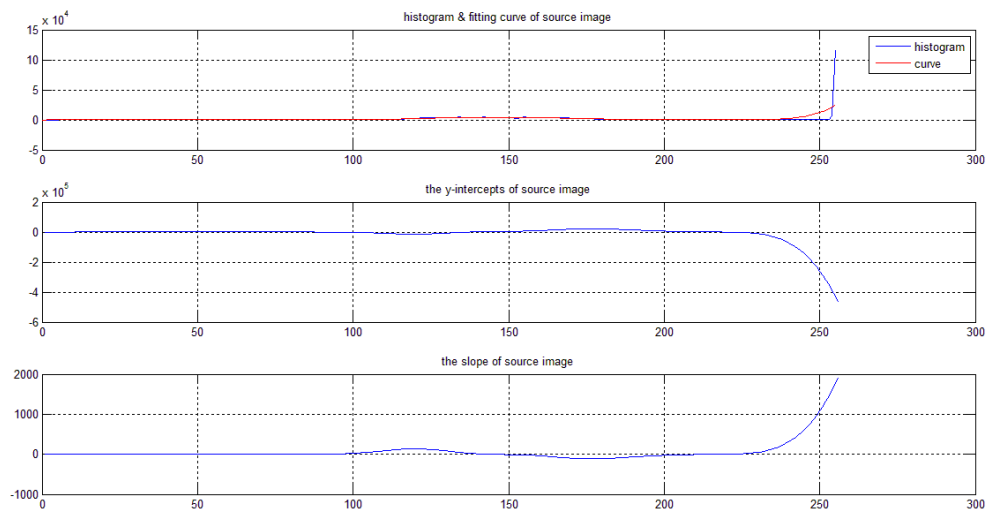
شکل ۴-۲۰- نمودار های هیستوگرام و منحنی تخمین زده شده تصویر هدف

نمودار  $\theta_0$  تصویر هدف

نمودار  $\theta_1$  تصویر هدف



شکل ۴-۲۱- تصویر مرجع (تصویر رنگی)



شکل ۴-۲۲- نمودار های هیستوگرام و منحنی تخمین زده شده تصویر مرجع

نمودار  $\theta_0$  تصویر مرجع

نمودار  $\theta_1$  تصویر مرجع

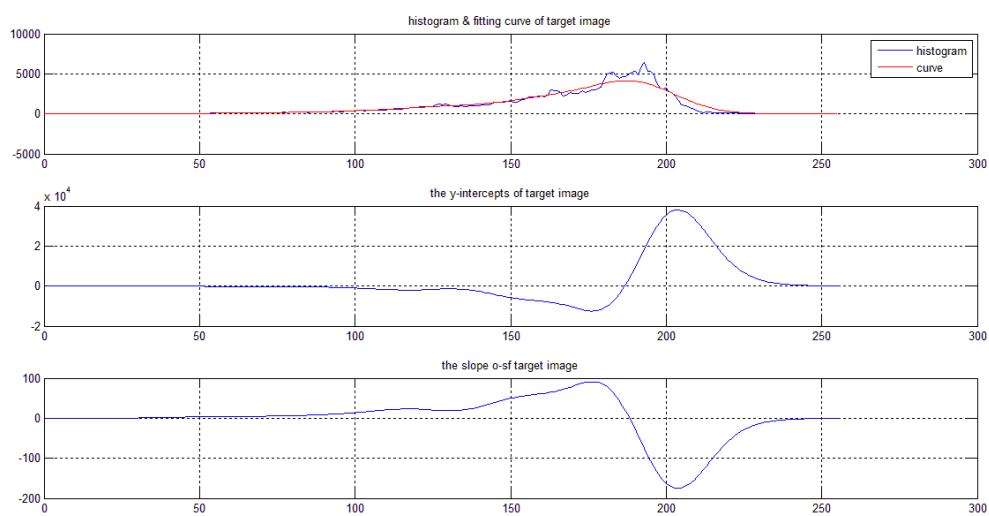


شکل ۴-۲۳- تصویر رنگدهی شده توسط الگوریتم

نمونه پنجم:



شکل ۴-۲۴- تصویر هدف (تصویر خاکستری)



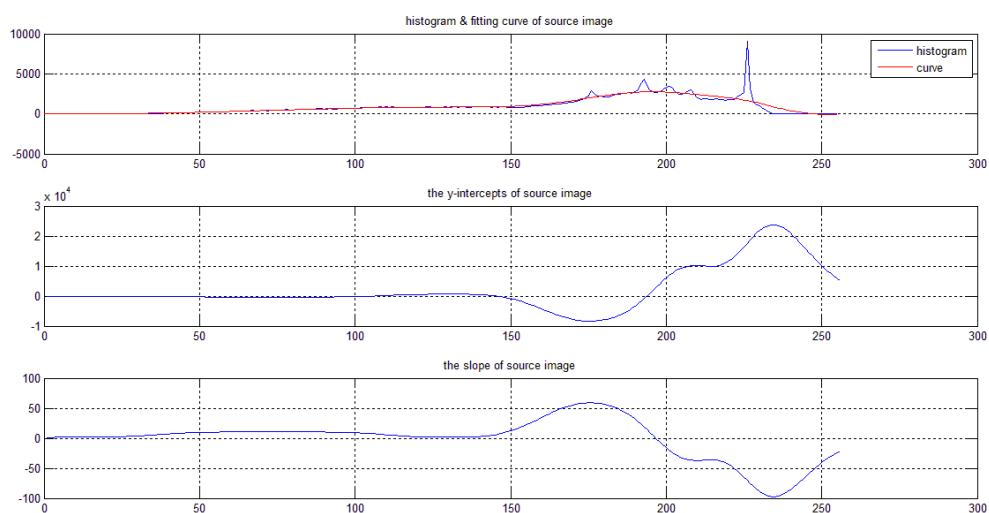
شکل ۴-۲۵- نمودار های هیستوگرام و منحنی تخمین زده شده تصویر هدف

نمودار  $\theta_0$  تصویر هدف

نمودار  $\theta_1$  تصویر هدف



شکل ۴-۲۶- تصویر مرجع (تصویر رنگی)



شکل ۴-۲۷- نمودار های هیستوگرام و منحنی تخمین زده شده تصویر مرجع

نمودار  $\theta_0$  تصویر مرجع

نمودار  $\theta_1$  تصویر مرجع





شکل ۴-۲۸- تصویر رنگدهی شده توسط الگوریتم

عکس	رزولیشن تصویر مرجع	رزولیشن تصویر هدف	زمان اجرا	میانگین خطای رنگ پیکسل ها	اطمینان رنگدهی
نمونه اول	۶۰۰*۵۶۷	۶۰۰*۵۶۷	۳,۲۰۷	۸,۶۰	۱
نمونه دوم	۶۰۰*۴۰۰	۶۰۰*۴۰۲	۲,۴۴۷	۱۰,۹۵۰	۱
نمونه سوم	۶۰۰*۳۶۸	۶۰۰*۴۰۰	۲,۳۸۰	۱۳,۵۶۹۲	۰,۵۱۳۴۲
نمونه چهارم	۶۰۷*۶۰۰	۶۰۰*۵۷۵	۳,۲۰۴	۷,۲۴۷۴	۰,۳۶۷۸۸
نمونه پنجم	۶۰۰*۴۰۰	۶۰۰*۳۹۹	۲,۳۷۲	۱۱,۸۰۳۱	۱