

## روشی نو برای انطباق پایدار اثرانگشت، با استفاده از پردازنده‌های گرافیکی، توسط کودا

فائزه میرزائی<sup>۱</sup>، محسن بیگلری<sup>۲</sup> و حسین ابراهیم‌پور کومله<sup>۳</sup>

<sup>۱</sup> دانشگاه کاشان، fmirzaei@grad.kashanu.ac.ir

<sup>۲</sup> دانشگاه شاهرود، biglari@shahroodut.ac.ir

<sup>۳</sup> دانشگاه کاشان، ebrahimpour@kashanu.ac.ir

چکیده - در دنیای مدرن امروز، جایی برای رمزهای عبور باقی نمانده و امنیت در سیستم‌های کنونی حرف اول را می‌زند. بیومتریک‌هایی چون چهره و صوت، تا حد زیادی قابل جعل هستند و اثرانگشت از این جنبه امنیت بسیار بالاتری دارد. در این مقاله، روشی جدید برای انطباق اثرانگشت ارائه شده است. روش پیشنهادی با اصلاحات گوناگون، به دقتی برابر و بعضاً بیشتر از بهترین روش‌های موجود دست یافته است؛ در این روش، از ارتباط سه‌گانه (مثلثی) مینوشیاها استفاده می‌شود. نتایج به دست آمده نشان دهنده دقت بالا و پایداری سیستم در برابر جابجایی و چرخش تصاویر می‌باشد. بخش‌های زمانبر سیستم پیشنهادی بسته به حجم انتقال داده و پردازش، توسط CPU یا GPU موازی‌سازی می‌شوند. استفاده از GPU توسط سکوی نوظهور کودا انجام گرفته است. نتایج به دست آمده، نشان دهنده دقت و سرعت مناسب سیستم پیشنهادی در کاربردهای برخط است؛ به شکلی که سرعت سیستم پس از موازی‌سازی، تقریباً نه برابر گشته و دقت آن بر روی پایگاه داده FVC2002 بالای ۹۸٪ می‌باشد.

کلید واژه‌ها- اثرانگشت، انطباق اثرانگشت، مینوشیا، انطباق مثلثی، پردازنده‌های گرافیکی، موازی‌سازی، کودا.

گلتنون<sup>۵</sup> برای نخستین بار، ویژگی‌های سطح دوم اثرانگشت را معرفی نمود. به این ویژگی‌ها مینوشیا<sup>۶</sup> گفته می‌شود. استخراج ویژگی‌های اثرانگشت به شدت وابسته به کیفیت تصاویر است؛ بنابراین هر چه سطح ویژگی افزایش می‌یابد، نیاز به کیفیت بالاتر نیز بیش‌تر خواهد شد. هر اثرانگشت شامل خطوط تیره و روشنی است که با الگوی خاصی جریان یافته‌اند. به خطوط تیره و برآمده، اصطلاحاً لبه<sup>۷</sup> و به خطوط روشن و تورفته، شیار<sup>۸</sup> گفته می‌شود. لبه‌ها و شیارها حاوی ویژگی‌های سطح دوم و سوم اثرانگشت می‌باشند. نقطه‌ی پایانی<sup>۹</sup> و انشعاب<sup>۱۰</sup>، دو مینوشیای شناخته شده هستند که در واقع نقطه‌ی پایانی لبه‌ها و انشعاب‌ها

### ۱. مقدمه

امروزه سیستم‌های شناسایی اثرانگشت<sup>۱</sup> جایگاه ویژه‌ای در علم بیومتریک<sup>۲</sup> دارند. با وجود سیستم‌های نوظهوری چون شناسایی عنبیه<sup>۳</sup> و شبکیه<sup>۴</sup>، شناسایی اثرانگشت هم‌چنان در راس اغلب سیستم‌های کاربردی مرتبط، به چشم می‌خورد. این امر به دلیل خواص انحصاری موجود در انگشت انسان می‌باشد. اثبات شده است که اثرانگشت انسان کاملاً یکتاست؛ تا حدی که دوقلوهای همسان نیز اثرانگشت متفاوت دارند. ویژگی‌های دیگری هم‌چون مانایی، سادگی در دریافت، جامعیت و امنیت، از دیگر دلایل اعتماد به این بیومتریک منحصر بفرد است.

<sup>۵</sup> Galton

<sup>۶</sup> Minutiae

<sup>۷</sup> Ridge

<sup>۸</sup> Valley

<sup>۹</sup> End Point

<sup>۱۰</sup> Branch

<sup>۱</sup> Fingerprint

<sup>۲</sup> Biometric

<sup>۳</sup> Iris

<sup>۴</sup> Retina

می‌باشند.

انطباق ویژگی، یک مرحله‌ی اساسی در سیستم‌های شناسایی اثرانگشت بشمار می‌رود. این فاز از سیستم شناسایی با روش‌های مختلفی قابل پیاده‌سازی است. مینوشیا را می‌توان از معروفترین ویژگی‌های مورد استفاده در این مرحله دانست. هر مینوشیا شامل ویژگی‌هایی هم‌چون مختصات، جهت و نوع می‌باشد؛ از این ویژگی‌ها در انطباق دو به دوی مینوشیاها استفاده می‌شود. راه‌حل‌های مختلفی در زمینه‌ی انطباق مینوشیا ارائه گردیده است [۱-۵]. برای مثال در [۳]، فرض شده است که جابجایی در تصاویر وجود ندارد و جهت مینوشیا نیز از قبل استخراج شده است که ضعف مهمی به شمار می‌آید؛ زیرا جابجایی و استخراج جهت از مراحل اصلی انطباق محسوب می‌شوند. مرجع [۴] از ترکیب ویژگی‌های مختلف از جمله مینوشیا استفاده کرده است. در این سیستم، ۱۷ ویژگی به صورت ترکیبی به کار رفته است. با وجود دقت مناسب، این روش سرعت پایینی دارد و در سیستم‌های برخط قابل استفاده نمی‌باشد. مرجع [۵] از روش مبتنی بر وابستگی<sup>۱</sup> بهره برده است؛ این روش نسبت به چرخش، جابجایی و حتی تغییرات کنتراست پایدار نیست. در حال حاضر، انطباق اثرانگشت در تصاویر کم‌کیفیت از مسائل باز محسوب می‌شود.

در کل، انطباق مینوشیا به شکل ساده و با توجه به مختصات و جهت آن‌ها، کارایی مناسبی ندارد. حتی اگر تصاویر هم‌جهت شوند، مشکل جابجایی حل نخواهد شد. قطعه‌بندی<sup>۲</sup> نیز در حالتی که مقیاس تصاویر متفاوت باشد مناسب نیست. مقیاس متفاوت با فشار متفاوت انگشت بر سنسور ورودی ایجاد می‌شود. استفاده از روابط بین مینوشیاها ایده‌ای است که مطالعات زیادی بر روی آن انجام شده است [۶-۸]. به‌عنوان مثال، در [۶] با توجه به مکان هسته‌ی تصویر که از ویژگی‌های سطح اول اثرانگشت می‌باشد، مجموعه‌ی مینوشیاها مرتبط باهم تعیین شده و فقط اعضای این مجموعه با یکدیگر مقایسه می‌شوند. در این حالت، اگر هسته به اشتباه تشخیص داده شود، مجموعه‌ی مینوشیاها نیز مناسب نخواهند بود. ارتباط مینوشیاها همسایه در اندیس‌گذاری و کلاس‌بندی و به تازگی در انطباق اثرانگشت نیز مورد مطالعه قرار گرفته است [۹-۱۲]. در این سیستم‌ها از جهت مینوشیا، جهت رئوس مثلث ایجاد شده و ویژگی‌های دیگر

مشابه، استفاده می‌شود.

در سیستم پیشنهادی، از روش مبتنی بر مینوشیا بهره گرفته شده است. این سیستم، نیاز به هم‌جهت کردن تصاویر و استخراج جهت مینوشیا ندارد. از این رو، در عین کارایی بالا، از سرعت زیادی برخوردار است. خاصیت مثلثی استفاده شده در این روش، موجب پایداری سیستم در برابر چرخش و جابجایی تصویر می‌شود. سرعت بسیار بالای روش مثلثی توسعه یافته، پس از موازی‌سازی، کاملاً برای کاربردهای برخط ایده‌آل است.

در بخش دوم مقاله، مرحله‌ی انطباق اثرانگشت توضیح داده شده است. سپس در بخش سوم به سیستم پیشنهادی پرداخته شده است. نتیجه‌ی آزمایشات سیستم ترتیبی در بخش چهارم ارائه گردیده است؛ و در بخش پنجم، موازی‌سازی بخش‌های مختلف سیستم تشریح شده است. بخش ششم، شامل نتایج آزمایشات مربوط به موازی‌سازی می‌باشد. در پایان نیز نتیجه‌گیری صورت گرفته است.

## ۲. سیستم شناسایی اثرانگشت

در یک سیستم شناسایی اثرانگشت، تصویر ورودی باید پس از دریافت<sup>۴</sup> و پیش‌پردازش<sup>۵</sup>‌های مناسب، وارد مرحله‌ی مقایسه و انطباق شود. از آنجا که بسیاری از تصاویر، از کیفیت مطلوبی برخوردار نمی‌باشند، در ابتدا باید توسط پیش‌پردازش‌های خاص، بهبود<sup>۶</sup> یابند. پس از این مرحله، ویژگی‌های مورد نیاز مرحله‌ی انطباق استخراج می‌گردد.

### ۲.۱. بهبود

کیفیت تصاویر اثرانگشت بسته به سنسور ورودی و روش اسکن تصاویر متفاوت است. از طرفی شرایط پوستی خاص مانند خشکی، چربی و زخم نیز منجر به کاهش کیفیت تصاویر می‌شوند. با توجه به روش‌های معمول ثبت اثرانگشت در سازمان‌های مرتبط، اکثر اثرانگشت‌ها کیفیت پایینی دارند. مخصوصاً در صحنه‌های ارتکاب جرم، احتمال اینکه تصویر استخراج شده از اثرانگشت مظنون کیفیت بالایی داشته باشد، بسیار کم است. به منظور مقابله با مشکلات ذکر شده، ابتدا تصویر ورودی پیش‌پردازش شده و پس از بهبود کیفیت، وارد مراحل بعد می‌شود. یکی از راه‌های کاهش نویز، اعمال فیلترهای

<sup>۴</sup> Acquisition

<sup>۵</sup> PreProcessing

<sup>۶</sup> Enhancement

<sup>۱</sup>Correlation Based

<sup>۲</sup> Segmentation

<sup>۳</sup> Core

### ۲.۳. انطباق

انطباق مینوشیا شناخته شده‌ترین و پر استفاده‌ترین روش انطباق دو اثرانگشت به‌شمار می‌رود؛ این امر به دلیل اثبات یکتایی و مانایی مینوشیا به‌عنوان یک ویژگی کارا است. یک تعریف از مسئله‌ی انطباق مینوشیا در ادامه ارائه شده است. اگر  $I$  و  $T$  به ترتیب تصاویر الگو و ورودی باشند. اغلب الگوریتم‌های انطباق مینوشیا، هر مینوشیا را به‌صورت سه جمله‌ای  $m = \{x, y, \theta\}$  می‌شناسند که  $x$  و  $y$  مختصات و  $\theta$  زاویه‌ی مینوشیاست. مینوشیای  $m'_j$  از  $I$  با مینوشیای  $m_i$  از  $T$  منطبق شناخته می‌شود اگر فاصله‌ی فضایی ( $sd$ ) بین آن‌ها از آستانه‌ی  $r_0$  و فاصله‌ی جهتی ( $dd$ ) آن‌ها از آستانه‌ی  $\theta_0$  کمتر باشد (معادله‌ی ۲).

$$sd(m'_j, m_i) = \sqrt{(x'_j - x_i)^2 + (y'_j - y_i)^2} \leq r_0 \quad \text{and} \quad (2)$$

$$dd(m'_j, m_i) = \min(|\theta'_j - \theta_i|, 360 - |\theta'_j - \theta_i|) \leq \theta_0$$

کاربرد سیستم‌های شناسایی اثرانگشت را می‌توان به دو دسته‌ی تصدیق<sup>۵</sup> و تعیین هویت<sup>۶</sup> تقسیم نمود. در یک سیستم تصدیق هویت، تصویر ورودی حامل اطلاعاتی درباره‌ی تصویر معادل خود در پایگاه داده است. در این حالت، مقایسه تنها بین یک جفت اثرانگشت انجام می‌شود. خروجی سیستم تصدیق، قبول یا رد تصویر ورودی است. سیستم‌های کنترل دسترسی نمونه‌ای از این سیستم‌ها می‌باشند. در سیستم تعیین هویت، تصویر ورودی هیچ اطلاعات مازادی ندارد. به‌عنوان مثال، اثرانگشت‌های موجود در صحنه‌ی جرم از این نمونه هستند. در این حالت، تصویر ورودی با تک‌تک تصاویر موجود در پایگاه داده مقایسه می‌شود. خروجی این سیستم، اثرانگشتی با بیش‌ترین شباهت به تصویر ورودی است. تفاوت این دو نوع سیستم در دفعات تکرار عمل مقایسه و خروجی آن‌هاست و مرحله‌ی انطباق در هر دو به یک شکل پیاده‌سازی می‌شود [۱۴].

مرحله‌ی انطباق را می‌توان اساس سیستم‌های شناسایی اثرانگشت برشمرد. این مرحله را می‌توان با استفاده از انواع مختلفی از ویژگی‌ها پیاده‌سازی کرد. با توجه به نوع ویژگی‌های مورد استفاده، سه دسته‌ی کلی برای روش‌های انطباق تعریف می‌شود [۱۴].

- روش‌های مبتنی بر وابستگی

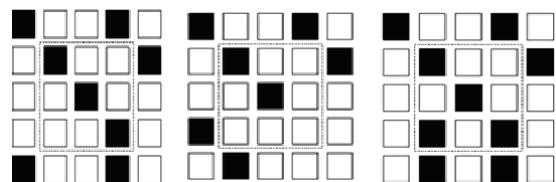
مختلف بر تصویر ورودی است. اغلب این فیلترها در حوزه فرکانسی به تصویر اعمال شده و تصویر پس از بهبود کیفیت، به حوزه فضایی برگردانده می‌شود. به‌عنوان مثال می‌توان از فیلتر گابور، گابور لگاریتمی، WFT و فیلتر مرکب<sup>۱</sup> نام برد. در این بین، فیلترهای گابور و گابور لگاریتمی از سایرین رایج‌تر بوده و می‌توانند به صورت بانکی از فیلترها بر تصویر اعمال شوند.

### ۲.۲. استخراج ویژگی

به‌منظور استخراج مینوشیا، ابتدا تصاویر به معادل دودویی<sup>۲</sup> و باریک<sup>۳</sup> خود تبدیل می‌شوند. در مرحله باریک‌سازی، ضخامت خطوط تا حد یک پیکسل کاهش می‌یابد [۱۳]. در حقیقت، نقطه‌ای از تصویر به عنوان مینوشیا شناخته می‌شود که تعداد همسایه‌های آن یک یا سه باشد. معیار تعداد همسایه‌ها که برای نقطه‌ی  $p$  با  $Cn(p)$  نمایش داده می‌شود، به صورت معادله‌ی (۱) قابل تعریف است. تعداد همسایه‌ها، نصف جمع تفاوت بین جفت همسایه‌های یک نقطه در بین هشت همسایه‌ی آن است. شکل ۱، همسایگی نقاط درون لبه، نقاط پایانی و انشعاب را نشان می‌دهد. برای هر نقطه‌ی  $p$ ، اگر  $Cn(p) = 2$  باشد، نقطه‌ی  $p$  بر روی یک لبه قرار دارد. در صورتی که  $Cn(p) = 1$  باشد، نقطه‌ی  $p$  یک نقطه‌ی پایانی است؛ و در حالتی که  $Cn(p) = 3$  باشد، نقطه‌ی  $p$  یک انشعاب است. در اغلب سیستم‌های اثرانگشت یک مرحله‌ی پس‌پردازشی برای حذف مینوشیاهای اشتباه صورت می‌گیرد. زیرا اغلب در مناطق نویزدار اثرانگشت و یا به علت پیش‌پردازش‌های صورت گرفته در مراحل قبل، تعدادی مینوشیا، به اشتباه استخراج می‌شوند.

$$Cn(p) = \frac{1}{2} \sum_{i=1,8} |val(p_{i \bmod 8}) - val(p_{i-1})| \quad (1)$$

$$val(p) \in \{0,1\}$$



شکل ۱: تشخیص مینوشیاها با استفاده از تعداد همسایگی نقاط

<sup>۴</sup> Threshold

<sup>۵</sup> Verification

<sup>۶</sup> Identification

<sup>۱</sup> Composite

<sup>۲</sup> Binary

<sup>۳</sup> Thin

دودویی شدن، باریک سازی و بهبود یافتن، وارد مرحله‌ی استخراج ویژگی می‌شود. در مرحله‌ی قطعه‌بندی، تصویر از پس زمینه‌ی خود جدا می‌شود. پس از مرحله‌ی دودویی کردن، تصویر با اعمال قالب‌های خاص، باریک شده و ضخامت لبه‌ها به یک پیکسل کاهش داده می‌شود. در روش پیشنهادی از قالب [۱۶] استفاده شده است.

### ۳.۱. بهبود

برای انتخاب روش پیش‌پردازش مناسب در سیستم پیشنهادی، بهترین روش‌های موجود مورد بررسی قرار گرفته و آزمایش شدند. سپس روش مناسب برای استفاده در سیستم انتخاب گردید. روش استفاده شده، در [۱۷] معرفی شده است. این روش شامل مراحل زیر می‌باشد:

- نرمال‌سازی تصویر
- محاسبه جهت محلی تصویر در بلوک‌های با اندازه‌ی مشخص
- محاسبه فرکانس محلی تصویر در بلوک‌های با اندازه‌ی مشخص
- ساخت بانکی از فیلترهای گابور با جهت و فرکانس محلی
- بهبود نقاط قابل تعمیر تصویر با فیلتر گابور و حذف نقاط غیر قابل تعمیر

### ۳.۲. استخراج ویژگی

مینوشیای استخراج شده از تصویر دودویی و باریک شده در سیستم پیشنهادی در شکل ۳ نشان داده شده است. در این شکل، نقاط پایانی با دایره و انشعاب‌ها با علامت مثبت نشان داده شده‌اند. به دلیل نویز موجود در تصاویر، اشتباهات مرحله‌ی باریک سازی و یا مرحله‌ی دودویی کردن و مشکلاتی از این قبیل، تعدادی از مینوشیاهای استخراج شده، واقعی نبوده و به اشتباه شناخته شده‌اند. بنابراین، این ویژگی‌ها باید فیلتر شوند. عمل فیلترسازی با مقایسه‌ی فاصله‌ی مینوشیاهای از یکدیگر انجام می‌شود. در این فیلترها، اگر دو مینوشیا از حد مشخصی به یکدیگر نزدیکتر باشند، حذف خواهند شد. با وجود حذف مینوشیای زائد از این طریق، گاه تعدادی مینوشیای اشتباه در حاشیه‌ی تصویر باقی خواهند ماند. بنابراین، سیستم در حواشی تصویر جستجو کرده و مینوشیاهایی که طول یا عرض مشابه دارند را حذف می‌کند. شکل ۴ نتیجه‌ی حذف مینوشیای اشتباه

- روش‌های مبتنی بر مینوشیا
  - روش‌های مبتنی بر عوامل غیر مینوشیا
- در روش اول، مقایسه تنها بر مبنای چگالی<sup>۱</sup> نقاط تصاویر انجام می‌شود. دو تصویر مورد نظر، در جهات مختلف از یکدیگر تفریق شده و هرچه تصویر حاصل تیره‌تر باشد، به معنی انطباق بیشتر دو تصویر مربوطه است.

همان‌طور که انتظار می‌رود این روش کارایی بالایی ندارد. در روش‌های مبتنی بر عوامل غیر مینوشیا از ویژگی‌های سطح اول یا سوم تصویر استفاده می‌شود. استخراج ویژگی‌های سطح سوم، مستلزم کیفیت و رزولوشن بالای تصاویر می‌باشد؛ زیرا در این روش حفره‌های کوچک موجود در لبه‌ها مورد پردازش قرار می‌گیرند. روش مبتنی بر مینوشیا (ویژگی‌های سطح دوم) رایج‌ترین روش به شمار می‌رود که در سیستم پیشنهادی نیز، روشی جدید مبتنی بر آن ارائه شده است.

هر مینوشیا شامل دو ویژگی مختصات و جهت می‌باشد. هدف از انطباق مبتنی بر مینوشیا، یافتن تصویری است که مجموعه مینوشیای آن بیش‌ترین شباهت را به مجموعه مینوشیای تصویر ورودی داشته باشد. در روش تصدیق، در صورتی که مینوشیاهای منطبق از تعداد خاصی تجاوز کنند، دو تصویر منطبق شناخته می‌شوند. مقایسه‌ی دو به دوی مختصات و زوایای مینوشیا راه حل مناسبی نیست، زیرا تصاویر موجود در پایگاه داده اغلب چرخیده و یا جابجا شده‌اند. شکل ۲ نمونه‌ای از چرخش و جابجایی تصاویر مشابه پایگاه داده‌ی FVC2002 را نشان می‌دهد [۱۵].



شکل ۲: چرخش و جابجایی در اثرانگشت‌های مشابه [۱۵]

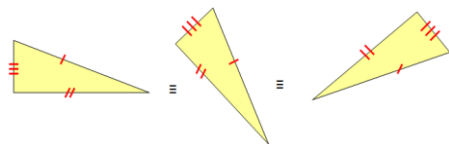
### ۳. سیستم پیشنهادی

استخراج ویژگی از تصاویر اولیه به دلیل وجود کنتراست متفاوت، نویز و عوامل مخرب دیگر کار دشواری است؛ بدین منظور ابتدا تصویر وارد مرحله‌ی پیش‌پردازش شده و پس از قطعه‌بندی،

<sup>۱</sup> Density

پایه استوار است که: "نقاط انشعاب در تصویر اولیه همان نقاط پایانی در تصویر معکوس<sup>۳</sup> هستند". این خاصیت، خاصیت دوگانگی<sup>۴</sup> نام دارد [۱۴].

با همه‌ی موانعی که گفته شد، در تصاویر کم‌کیفیت، تعیین دقیق جهت در یک نقطه امر دشوار است و ممکن است دقیق و معتبر نباشد. بنابراین روشی که بتواند از جهت مینوشیا استفاده نکند و در عین حال، دقیق باشد، با ارزش است. سیستم پیشنهادی، از رابطه‌ی بین مینوشیاهای تصویر استفاده می‌کند. ارتباط مثلی بین هر سه مینوشیا، ایده‌ی اولیه در این روش است. در این حالت، تمام مثلث‌های ممکن بین مینوشیاها در نظر گرفته شده، زوایا، اضلاع و محیط آن‌ها محاسبه می‌شود. سپس با بررسی وجود رابطه‌ی مشخصی بین مثلث‌های استخراج شده از دو تصویر، انطباق و عدم انطباق آن‌ها تعیین می‌گردد. برای مقایسه دو مثلث متشابه، از سه قانون معروف هندسه (ضضض، ضضز، زضز) بسته به شرایط اثر انگشت‌ها استفاده می‌شود. در این روش، نیازی به هم‌جهت کردن تصاویر نیست؛ زیرا همان‌طور که انتظار می‌رود، دو مثلث معادل، با چرخش به هر جهتی، معادل باقی می‌مانند. شکل ۵ نمونه‌ای از پایداری تشابه مثلثات را نشان می‌دهد.

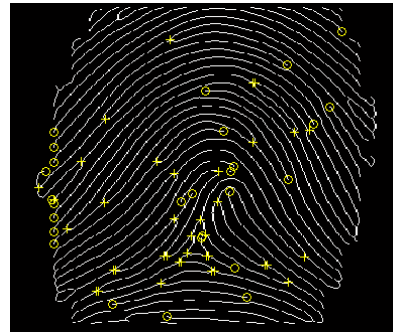


شکل ۵: مقاومت تشابه دو مثلث معادل با چرخش در حالت سه ضلع

پردازش مجموعه تصاویری که مقیاس آن‌ها برابر است، با قانون سه ضلع انجام می‌شود. در حالتی که مقیاس تصاویر برابر نباشد، می‌توان از روش سه زاویه استفاده کرد. ایده‌ی دیگر، استفاده از مقیاس اضلاع است؛ اضلاع مثلث دوم باید با مقیاس مشخصی بزرگتر یا کوچکتر از اضلاع معادل در مثلث اول باشند (در روش سه ضلع).

در روش پیشنهادی، فقط مختصات مینوشیاها محاسبه شده و نیازی به استفاده از جهت آن‌ها نمی‌باشد. این امر به سرعت سیستم کمک شایانی می‌کند. هر مینوشیا را می‌توان با  $m_i = \{x_i, y_i, t_i\}$  نشان داد. در این مجموعه،  $x_i$  و  $y_i$  مختصات مینوشیا و  $t_i$  نوع مینوشیا است. دو نوع مینوشیای

و مینوشیای حاشیه را نشان می‌دهد. روشی دیگر برای این منظور، تشخیص محدوده مورد استفاده (ROI) از اثر انگشت و حذف مینوشیاهای موجود در سایر مناطق است. ولی به دلیل خطای موجود در تشخیص این محدوده، روش پیشنهادی، کارایی بهتری دارد.



شکل ۳: استخراج مینوشیا از تصویر دودویی و باریک شده



شکل ۴: فیلترسازی مینوشیا و حذف نقاط زائد و حاشیه

### ۳.۳. انطباق

در روش استفاده از مختصات و جهت مینوشیا به صورت معمول، اگر تصویر اندکی چرخیده و یا جابه‌جا شده باشد، کار دشوار می‌شود؛ برای حل این مشکل، در بسیاری از سیستم‌ها، ابتدا دو تصویر هم‌جهت می‌شوند؛ مرحله‌ی هم‌جهت کردن فرایندی زمان‌بر است و در عین حال همیشه دقیق نیست. در اغلب سیستم‌های موجود، از ویژگی جهت مینوشیا نیز استفاده می‌شود. این ویژگی برای نقطه‌ی پایانی با استفاده از روش‌هایی همچون گرادیان<sup>۲</sup> قابل محاسبه است ولی تعیین جهتی مناسب برای انشعاب کار دشواری است. انشعاب از اتصال سه خط مجزا به وجود می‌آید و جهت انشعاب را نمی‌توان به راحتی به خط خاصی اختصاص داد. در برخی سیستم‌ها از سه جهت و در برخی دیگر یک جهت استفاده می‌شود. ایده‌ی اصلی روش دوم بر این

<sup>۳</sup> Inverse

<sup>۴</sup> Duality

<sup>۱</sup> Region Of Interest

<sup>۲</sup> Gradient

می‌گردند. در مرحله‌ی بعد، زوج مثلث‌های متشابه در دو تصویر تشخیص داده شده و از بین این مجموعه، مثلث‌های منطبق تعیین می‌شوند. بعد از تشخیص مثلث‌های متشابه مجموعه‌ای خواهیم داشت که در هر خانه‌ی آن، یک زوج مثلث متشابه قرار دارد. سپس به صورت تکراری، فاصله‌ی دو مثلث موجود در سطر جاری از مجموعه، با سایر مثلث‌ها محاسبه می‌شود. فاصله مثلث‌ها نیز با خطوط متصل کننده‌ی رئوس آن‌ها تعیین می‌گردد. در نهایت، بیشترین تعداد مثلث‌هایی که فاصله آن‌ها از یکدیگر با احتساب مقدار خطای مشخص  $\sigma$ ، برابر بوده است، به عنوان مقدار مشابهت دو اثرانگشت در نظر گرفته می‌شود.

فرض می‌کنیم مجموعه‌ی  $K$  شامل زوج مثلث‌های متشابه از دو تصویر باشد. در این صورت، بین رئوس هر زوج مثلث از هر اثرانگشت، خطوطی رسم می‌شود. این خطوط در اثرانگشت دیگر نیز بین مثلث‌های معادل با اثرانگشت اول رسم می‌شوند. اگر این خطوط معادل باشند، این دو زوج مثلث، متشابه و منطبق شناخته می‌شوند. نمونه‌ای از این خطوط در شکل ۷ در یک تصویر واقعی نشان داده شده‌اند. رئوس معادل با اشکال هندسی مشابه نشان داده شده‌اند. نهایتاً، تعداد مثلث‌های منطبق بین دو اثرانگشت شمارش می‌شوند. این تعداد، درصد مشابهت دو تصویر ورودی را مشخص می‌سازد. شبه‌کد شکل ۸، روش محاسبه‌ی تعداد بیشینه‌ی مثلث‌های منطبق را نشان داده است.

سخت‌گیری این روش ابتدا در متشابه بودن دو مثلث و سپس در معادل بودن خطوط متناظر آن‌هاست. در این صورت، اگر دو مثلث معادل باشند ولی در تصاویر منطبق نباشند؛ احتمال اینکه مثلث منطبق دیگری در همسایگی آن وجود داشته باشد که خطوط متصل کننده‌ی آن‌ها معادل باشند، بسیار کم است. شکل ۹، حالتی که دو زوج مثلث متشابه، واقعاً منطبق هستند را نشان می‌دهد. اثرانگشت‌های موجود در این تصاویر نسبت بهم ۱۸۰ درجه چرخیده‌اند.

#### ۴. موازی‌سازی سیستم پیشنهادی

##### ۴.۱. سکوی کودا

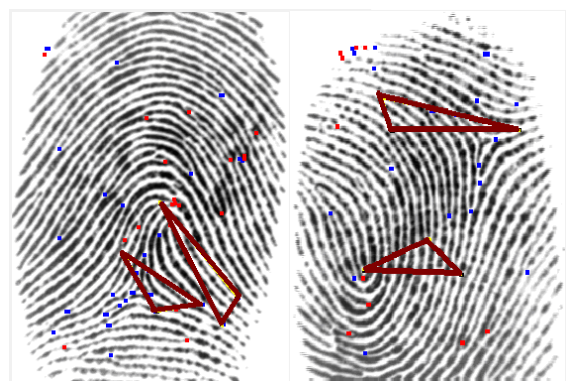
شرکت Nvidia در سال ۲۰۰۶، سکوی کودا<sup>۱</sup> را برای انجام محاسبات موازی حجیم و با کارایی بالا بر روی GPUهای<sup>۲</sup>

نقطه‌ی پایانی و انشعاب وجود دارد. هر مثلث نیز با مجموعه‌ی  $Triang_j = \{A, B, C, AB, AC, BC, AMB\}$  مشخص می‌شود. در این مجموعه،  $A, B, C$  رئوس مثلث،  $AB, AC, BC$  اضلاع مثلث و  $AMB$  محیط آن است. همان‌طور که انتظار می‌رود، با افزایش تعداد مینوشیاهای، تعداد مثلثات بسیار زیاد خواهد شد. روشن است که این مسئله باعث افزایش زمان محاسبات می‌گردد. برای اجتناب از این موضوع، مثلث‌های بسیار کوچک و بسیار بزرگ در نظر گرفته نمی‌شوند. مثلث‌های بسیار کوچک از مینوشیاهای بسیار نزدیک به هم تشکیل می‌شوند که به دو دلیل معیار مناسبی نیستند:

(۱) به احتمال زیاد، این نقاط بسیار نزدیک به اشتباه تشخیص داده شده‌اند.

(۲) مثلث‌های بسیار کوچک با بسیاری از مثلث‌های کوچک دیگر مشابه شناخته می‌شوند (با احتساب مقداری خطا).

مثلث‌های بسیار بزرگ نیز از نقاط بسیار دور از هم تشکیل شده‌اند که استفاده از آن‌ها ضروری نیست؛ زیرا به صورت تجربی دیده شده است که مثلث‌های ایجاد شده از هر نقطه و نقاط موجود در محدوده‌ی منطقی آن، تاثیرگذارترند. اگر به مثلث‌های معادل دو تصویر توجه شود، تعدادی مثلث متشابه و غیرمنطبق دیده می‌شود. این مثلث‌ها از لحاظ هندسی متشابه‌اند ولی در دو تصویر اثرانگشت منطبق نیستند. این مثلث‌ها نباید در شمار مثلث‌های منطبق قرار گیرند. نمونه‌ای از این مثلث‌ها در شکل ۶ نشان داده شده است.



شکل ۶: مثلث‌های متشابه (هندسی) و نامنطبق

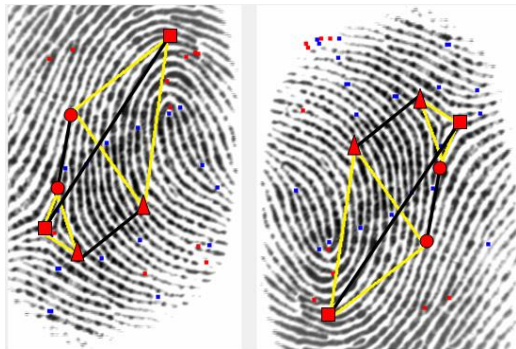
در سیستم پیشنهادی، برای تشخیص مثلث‌های متشابه و نامنطبق، به صورت زیر عمل می‌شود. پس از استخراج مثلث‌های حاصل از مینوشیای هر دو تصویر ورودی و الگو، طبق قوانین حذف مثلث‌های زائد اشاره شده، ابتدا تعدادی از آن‌ها حذف

<sup>۱</sup> CUDA Platform

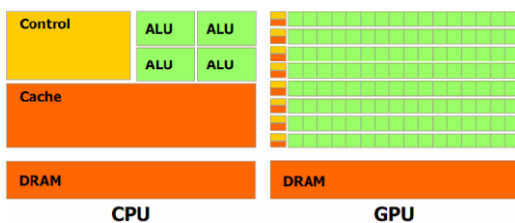
<sup>۲</sup> Graphic Processing Units



شکل ۸: محاسبه امتیاز شباهت دو تصویر



شکل ۹: نمونه‌ای از مثلث‌های متشابه و منطبق در یک تصویر واقعی



شکل ۱۰: تفاوت ساختار داخلی CPU و GPU

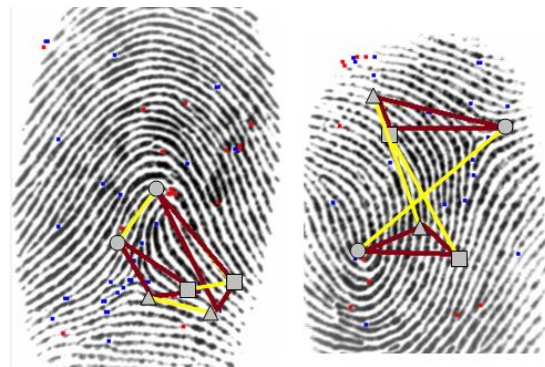
#### ۴.۲. موازی‌سازی

در این بخش، موازی‌سازی سیستم پیشنهادی توسط CPU و GPU مورد بررسی قرار می‌گیرد. تعیین نوع موازی‌سازی، به نسبت عملیات پردازشی و عملیات انتقال داده وابسته است؛ بنابراین با توجه به تحلیل بخش‌های زمانبر سیستم، یکی از دو روش موازی‌سازی استفاده می‌شود. مشخصات سخت‌افزاری CPU و GPU مورد استفاده برای موازی‌سازی، در ادامه آورده شده است:

GPU  
 Model: 'GeForce GT 230'  
 Cores Count: 48  
 Shared Memory: 1024 kb  
 CPU  
 OS: Windows 7  
 Processor: Intel(R) Core(TM) i7 Q720 @ 1.60GHz  
 RAM: 4 GB

همان‌طور که در بخش‌های پیش اشاره شد، در مرحله‌ی انطباق سیستم پیشنهادی، ابتدا مثلث‌های متشابه استخراج شده و سپس از بین جفت مثلث‌های متشابه، مثلث‌های منطبق تعیین می‌گردند. در مرحله‌ی اول، مثلث‌ها از لحاظ هندسی و با قانون سه ضلع مقایسه شده و در مرحله‌ی دوم، خطوط متصل‌کننده‌ی مثلث‌ها با یکدیگر مقایسه می‌شوند. لازم به ذکر است که

ساخت خود که در کارت‌های گرافیکی این شرکت قرار داشتند، ارائه کرد [۱۸]. GPU به دلیل واحدهای محاسباتی بیش‌تری که نسبت به CPU دارد، در انجام محاسبات ریاضی بسیار سریع‌تر است. شکل ۱۰، ساختار داخلی CPU و GPU را نشان می‌دهد. همراه با کودا یک محیط نرم‌افزاری ارائه شد که به توسعه‌دهندگان امکان نوشتن برنامه به زبان C و اجرای آن بر روی GPU را می‌داد. زبان استفاده شده برای نوشتن برنامه‌های کودا، نسخه‌ی ساده شده‌ی از C به همراه مجموعه‌ای از ضمایم شرکت Nvidia با تعدادی محدودیت است؛ البته با ارائه‌ی نسخه‌های جدیدتر، این محدودیت‌ها نیز کاهش پیدا کردند. در این بین زبان‌های دیگری مانند متلب<sup>۱</sup> نیز می‌توانند با واسطه برای نوشتن برنامه‌های کودا مورد استفاده قرار گیرند. کدهای مربوط به برنامه‌های کودا باید در فایل‌هایی با پسوند ".cu" نوشته شوند. این کدها برای اجرا باید ابتدا توسط کامپایلر nvcc به کد PTX که نوعی اسمبلی است، تبدیل شوند. بخش‌هایی از کد که به زبان C هستند، توسط کامپایلر C و بخش‌های مرتبط با کودا توسط nvcc کامپایل می‌شوند. در نهایت خروجی این دو کامپایلر با یکدیگر ترکیب شده و فایل اجرایی نهایی را خواهند ساخت [۱۸].



شکل ۷: نمونه‌ای از مثلث‌های متشابه و نامنطبق در یک تصویر واقعی

```

S = { Set of congruent triangles }
MaxScore=0
For each pair p_i in S
    Score=0
    For each pair p_j in S
        if Match( p_i, p_j )
            Score=Score+1
    MaxScore = Max(Score, MaxScore)
Result= MaxScore
    
```

<sup>۱</sup> Matlab

#### ۴.۲.۲. موازی‌سازی بخش انطباق

در بخش ایجاد مثلث، مجموعه مینوشیاهای هر تصویر دریافت شده و مجموعه مثلث‌های ممکن برای نقاط پایانی و انشعاب بازگردانده می‌شود. در این مسئله، اگر تعداد مینوشیا  $N$  باشد، تعداد مثلث‌های ممکن حاصل از این مینوشیاهای  $N_T$  خواهد بود. معادله‌ی (۳) محاسبه مقدار  $N_T$  را بر اساس  $N$  نشان می‌دهد. برای ایجاد این تعداد مثلث، پیچیدگی زمانی  $O(n^3)$  مورد نیاز است. شبه کد ایجاد مثلث‌ها، در شکل ۱۱ نشان داده شده است.

$$N_T = \binom{N}{3} = \frac{N!}{(N-3)!3!} \in O(n^3) \quad (3)$$

```

Triangles= [];
Points= [Set of points];
N= Len(Points);
for i=1 to N
  for j=i+1 to N
    for k=j+1 to N
      Triangles.add(Create_Triangle(
        Points[i],Points[j],Points[k]));
    End
  end
end
end
    
```

شکل ۱۱: شبه کد ایجاد مثلث‌ها از مجموعه‌ی مینوشیاهای

در سیستم پیشنهادی، وظیفه‌ی انجام حلقه‌ی سوم را به هر نخ سپرده و به تعداد تکرار دو حلقه‌ی اول، نخ تعریف می‌کنیم. هر تکرار از دو حلقه‌ی اول، هر نخ باید تعدادی مثلث ایجاد کند. برای استفاده از کودا، باید ورودی و خروجی‌ها را به قالب یک‌بعدی تبدیل کنیم. در نتیجه، خروجی درجه‌ی سوم حاصل از کد بالا که در واقع تعداد مثلث‌های ایجاد شده است، باید در یک آرایه‌ی یک بعدی قرار داده شود. در نتیجه‌ی یک نگاشت سه‌بعدی به یک‌بعدی خواهیم داشت.

در صورتی که کمی بیشتر به کد بالا دقت کنیم، مشاهده می‌کنیم که در پیمایش اول از دو حلقه بیرونی،  $N-2$  مثلث ایجاد می‌شود. در پیمایش دوم،  $N-3$  و به همین ترتیب تا پیمایش آخر. بنابراین در آرایه یک‌بعدی خروجی، ابتدا  $N-2$  مثلث قرار داده می‌شود؛ سپس  $N-3$  و به همین ترتیب الی آخر. نکته‌ی مهم و دشوار کننده‌ی کار این است که هر نخ باید به صورت مستقیم به اندیس خانه‌ی مربوط به خود دسترسی داشته باشد و از طرفی، هیچ دو نخ‌ی به یک خانه‌ی یکسان دسترسی نداشته باشند. به ازای هر مقدار معتبر از  $i$  و  $j$  تعدادی مثلث خواهیم داشت. تعداد مثلث مربوط به هر یک از مقادیر این دو متغیر در شکل ۱۲، برای پنج نقطه نشان داده شده است. هدف، نگاشت

تغییرات حد آستانه انتخابی بر سرعت سیستم تاثیر معکوس دارد. با افزایش مقادیر حد آستانه، از سرعت کاسته و بر دقت افزوده خواهد شد. برای مثال، در حالتی که  $\sigma_1 = 1$  و  $\sigma_2 = 1$  باشد، زمان مورد نیاز مرحله‌ی انطباق ۵۰۰ میلی ثانیه و در صورتی که  $\sigma_1 = 2$  و  $\sigma_2 = 1$  قرار داده شود، زمان این مرحله تا ۲ ثانیه افزایش می‌یابد. در این جا  $\sigma_1$  به معنی حد آستانه برابری دو ضلع از دو مثلث و  $\sigma_2$  به معنی حد آستانه برابری خطوط متصل کننده‌ی مثلث‌هاست. به منظور تشخیص مدت زمان نسبی توابع درونی هر قطعه از برنامه در نرم‌افزار متلب، از ابزار پروفایلر استفاده شده است. با استفاده از این ابزار، زیربخش‌های زمان‌بر از هر بخش مشخص گردیده‌اند.

عمده‌ی زمان مصرفی در بخش پیش‌پردازش، توسط مرحله‌ی بهبود صرف می‌شود. در نتیجه، این زیربخش در GPU و توسط کودا موازی شده است. پس از آن، مرحله‌ی انطباق مورد بررسی قرار گرفت. این مرحله شامل سه زیربخش می‌باشد. در بخش اول و سوم، عملیات پردازشی بیش‌تر از انتقال داده است؛ از این رو، موازی‌سازی این دو بخش توسط GPU صورت گرفته است. بخش دوم، به دلیل کثرت عملیات انتقال داده، توسط CPU موازی‌سازی شده است؛ پیاده‌سازی این بخش بر روی GPU به صرفه نخواهد بود، زیرا زمان موردنیاز برای انتقال داده، افزایش سرعت حاصل از بخش پردازشی را بی‌اثر خواهد ساخت. در ادامه، هر یک از بخش‌های موازی شده، به همراه ساختار مورد استفاده برای موازی‌سازی مختصرا توضیح داده شده‌اند.

#### ۴.۲.۱. موازی‌سازی بخش بهبود

در بخش بهبود، اعمال فیلتر به تصویر زمان‌برترین زیربخش و کاملا مستعد برای موازی‌سازی است. در این بخش، فیلتر موردنظر بر روی هر یک از پیکسل‌های تصویر اصلی قرار داده شده و نتیجه‌ی به‌دست آمده در خانه متناظر در تصویر نهایی قرار می‌گیرد. موازی‌سازی، با سپردن هر پیکسل به یک نخ آغاز می‌شود. در صورتی که اندازه‌ی فیلتر را  $s \times s$  و اندازه‌ی تصویر را  $n \times n$  فرض کنیم، پیچیدگی زمانی در حالت ترتیبی برابر با  $O(s^2 n^2)$  خواهد بود. با سپردن اعمال فیلتر به پیکسل  $i, j$  به یک نخ، پیچیدگی به  $O(s^2)$  کاهش می‌یابد. از آنجا که اندازه  $s$  مقدار کوچکی است؛ در نتیجه پیچیدگی زمانی  $O(1)$  را خواهیم داشت.



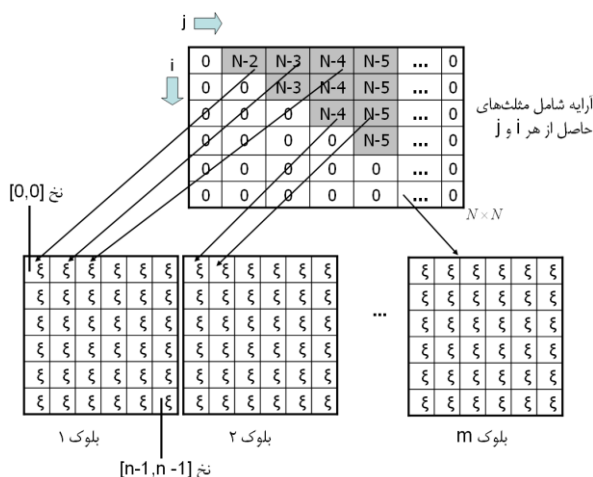
$$\sum_{i=1}^{curr\_row} \sum_{j=2}^i (N-j) \quad (7)$$

برای قرار دادن یک مثلث در هر خانه، باید تمامی اطلاعات موردنیاز برای یک مثلث را در نظر گرفت. در ساده‌ترین حالت، مختصات سه راس مثلث و طول سه ضلع آن مورد نیاز است؛ بنابراین به نه مقدار نیاز داریم (مختصات دو بعدی است). در نتیجه مقدار بدست آمده توسط محاسبات بالا، نهایتاً در عدد نه ضرب می‌شود. فرمول (۸)، اندیس‌های آغازین را برای نخ  $i, j$  در آرایه‌ی یک‌بعدی محاسبه می‌کند (جمع‌بندی توضیحات قبل):

$$index = \frac{(N-2) \times (N-1)}{2} \times i - \frac{(N-j-1) \times (N-j)}{2} - \sum_{i=1}^{curr\_row} \sum_{j=2}^i (N-j) \quad (8)$$

$$index = index * 9$$

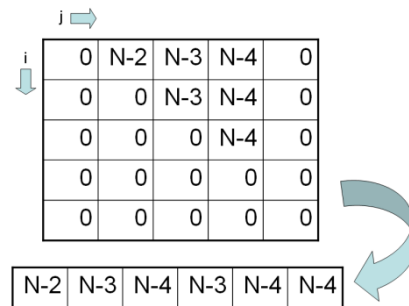
با استفاده از فرمول بالا، پیچیدگی اجرایی برنامه از  $O(n^3)$  به  $O(2n) \in O(n)$  کاهش می‌یابد که افزایش سرعت بسیار بالایی در بر خواهد داشت. هر نخ با صرف هزینه  $O(n)$  اندیس مربوط به خود را محاسبه می‌کند. سپس یک حلقه‌ی  $n$  تایی برای محاسبه‌ی داخلی‌ترین حلقه اجرا می‌کند که در مجموع پیچیدگی زمانی  $O(2n)$  را در بر دارد. شکل ۱۳، طریقه انتساب هر بخش از کد اولیه را به نخ‌های کودا نشان می‌دهد. هر نخ،  $k$  مثلث را تولید می‌کند که این تعداد بسته به  $i$  و  $j$  مربوط به نخ موردنظر، متغیر است.



شکل ۱۳: نگاشت کد به نخ‌های هر بلوک

بخش تشخیص مثلث‌های متشابه، همه‌ی مثلث‌های دو تصویر را پیمایش کرده و مثلث‌های متشابه را استخراج می‌کند (شبه‌کد

این تعداد به یک آرایه یک بعدی است به شکلی که بتوان با داشتن مقدار  $i, j$  و  $k$  به یک خانه از آرایه یک‌بعدی مراجعه کرد.



شکل ۱۲: تعداد مثلث‌های ایجاد شده با توجه به مقدار متغیرهای  $i$  و  $j$

با داشتن سه مقدار  $i, j$  و  $k$ ، برای یافتن یک اندیس از آرایه یک‌بعدی، ابتدا باید تعداد مثلث‌هایی که از ابتدا تا سطر  $i$  و ستون  $j-1$  تولید شده‌اند را بدست آوریم. همان‌طور که در شکل ۱۲ مشخص است، تعداد کل مثلث‌های سطر اول برابر است با مقدار نشان داده شده در فرمول (۴).

$$\frac{(N-2) \times (N-1)}{2} \quad (4)$$

فرض می‌کنیم همه‌ی سطرها به همین تعداد مثلث داشته باشند، سپس تعداد مثلث‌هایی اضافی را از تعداد کل کم می‌کنیم. تعداد کل مثلث‌ها تا سطر  $i$  ام برابر خواهد بود با (فرمول (۵)):

$$\frac{(N-2) \times (N-1)}{2} \times i \quad (5)$$

در صورتی که دو تعداد زیر را از تعداد کل نمایش داده شده در بالا کم کنیم، تعداد مثلث‌های تولید شده تا قبل از سطر  $i$  و ستون  $j$  را خواهیم داشت:

۱. تعداد مثلث‌های بعد از ستون  $j$  در سطر  $i$  ام
  ۲. تعداد مثلث‌های اضافی در هر سطر (سطر ۱ تا  $i$ )
- تعداد مثلث‌های بعد از ستون  $j$  در سطر  $i$  ام برابر است با (فرمول (۶)):

$$\frac{(N-j-1) \times (N-j)}{2} \quad (6)$$

تعداد مثلث‌های اضافی از سطر ۱ تا سطر جاری توسط فرمول (۷) محاسبه می‌شود.  $curr\_row$  اندیس سطر جاری است.

عدد بوده است، افزایش تعداد نخ‌ها تا حدود این مقدار، منطقی است. دیده می‌شود که تعداد ۱۰ نخ نیز باعث کاهش زمان اجرا گردیده است؛ دو نخ اضافی، موجب گردیده‌اند تا هیچ نخ‌ی در حین پروسه‌ی اجرا بیکار نماند. در صورتی که افزایش تعداد نخ‌ها به ۱۶ عدد، زمان اجرا را افزایش داده است. از این رویداد، می‌توان نتیجه گرفت که افزایش بی‌رویه تعداد نخ‌ها، موجب افزایش سرعت نمی‌شود؛ در چنین حالتی، بخش قابل توجهی از زمان پردازنده، صرف زمان‌بندی نخ‌ها شده و در نتیجه سرعت را کاهش می‌دهد.

جدول ۱: رابطه تعداد نخ‌های CPU و زمان اجرا

تعداد نخ‌ها	زمان اجرا (ms)
۱	۳۱۰
۲	۲۱۲
۴	۱۵۹
۸	۱۱۵
۱۰	۱۰۳
۱۶	۱۲۹

بخش پایانی مرحله‌ی انطباق، تشخیص مثلث‌های منطبق است. در این بخش، به‌ترتیب هر یک از زوج مثلث‌های متشابه موجود در مجموعه، به عنوان مرجع در نظر گرفته می‌شوند؛ سپس فاصله تمام زوج‌های دیگر با آن‌ها مقایسه می‌شود. بنابراین، پیچیدگی زمانی این مرحله نیز  $O(n^2)$  است. در حالت موازی، مقایسه‌ی هر زوج مرجع با سایر عناصر، به یک نخ سپرده شده است. به این صورت پیچیدگی زمانی به  $O(n)$  کاهش می‌یابد. هر نخ، تعداد مثلث‌های منطبق با هر مرجع را در بخش مربوط به خود از آرایه‌ی یک بعدی سراسری قرار می‌دهد. امتیاز شباهت دو تصویر با ماکزیمم‌گیری از این مقادیر به دست می‌آید. این عمل نیز به زمان  $O(n)$  نیاز دارد. در نتیجه پیچیدگی زمانی کل، به  $O(2n) \in O(n)$  کاهش می‌یابد.

## ۵. نتایج آزمایشات

روش پیشنهادی بر روی پایگاه داده‌ی FVC2002، بخش DB1-A آزموده شده است [۱۵]. در این پایگاه، ۱۰۰ گروه و در هر گروه، هشت تصویر اثرانگشت وجود دارد (۸۰۰ تصویر). در ابتدا به منظور سنجش مقاومت سیستم پیشنهادی در برابر چرخش و جابه‌جایی تصاویر، یک آزمون با تصاویر ساختگی صورت گرفته است. دو تصویر از پایگاه داده‌ی FVC2002

شکل ۱۴). این عمل با دو حلقه‌ی تودرتو انجام شده و پیچیدگی اجرایی آن در حالت ترتیبی  $O(n^2)$  است.

```

Triangles1= [Set of triangles of input image];
Triangles2= [Set of triangles of template image];
CongruentTris= [];
N1= Len(Triangles1);
N2= Len(Triangles2);

for i=1 to N1
  for j=1 to N2
    if (Triangles[i].Cong(Triangles[j]))
      CongruentTris.add([Triangles[i], Triangles[j]]);
    end
  end
end
    
```

شکل ۱۴: شبه کد استخراج مثلث‌های متشابه از مجموعه‌ی مثلث‌های دو تصویر ورودی و الگو به صورت ترتیبی

این بخش به دلیل حجم انتقال داده‌ی بالا، توسط CPU موازی‌سازی می‌شود. در روش پیشنهادی، مقایسه‌ی هر  $k$  مثلث از تصویر اول با  $n$  مثلث از تصویر دوم، به یک نخ سپرده شده است. مقدار  $k$  بستگی به تعداد هسته‌های CPU موردنظر دارد (شبه کد شکل ۱۵). در این صورت پیچیدگی اجرایی به  $O(kn)$  کاهش می‌یابد.

```

k= N1/ThreadsCount;
for t=0 to ThreadsCount-1 in Parallel
  for i= (t*k) to ((t+1)*k)
    for j=1 to N2
      if (Triangles[i].Cong(Triangles[j]))
        CongruentTris.add([Triangles[i], Triangles[j]]);
      end
    end
  end
end
    
```

شکل ۱۵: شبه کد استخراج مثلث‌های متشابه از مجموعه‌ی مثلث‌های دو تصویر ورودی و الگو به صورت موازی توسط CPU

تعداد نخ‌های CPU مورد استفاده در موازی‌سازی این بخش، بر افزایش سرعت حاصل موثر است. تعداد نخ‌ها باید رابطه‌ی منطقی با تعداد هسته‌های پردازنده داشته باشند.

در سیستم پیشنهادی، برای انتخاب تعداد بهینه‌ی نخ‌ها، شبه کد شکل ۱۵ با تعداد نخ‌های متفاوت اجرا شده و نتیجه در جدول ۱ گزارش شده است. در این جدول مشاهده می‌شود که با افزایش تعداد نخ‌ها، زمان اجرا کاهش یافته است. از آنجا که تعداد هسته‌های پردازنده در سیستم اجراکننده‌ی کد، برابر با هشت

برای آزمایش سیستم بر روی تصاویر ساختگی، تصویر اول از هر مجموعه با بقیه تصاویر مجموعه خود و تصاویر مجموعه دیگر، مورد مقایسه قرار گرفت. جداول ۲ و ۳، نتایج آزمایش را نشان می‌دهند. در جدول ۲، تنها از مثلث‌های متشابه و در جدول ۳، از مثلث‌های متشابه و منطبق برای انطباق استفاده شده است. در جدول ۲، مشاهده می‌شود که تعداد زیادی مثلث متشابه در اثر انگشت‌های غیر هم‌گروه وجود دارد. در صورتی که با در نظر گرفتن مثلث‌های منطبق، این تعداد به حد بسیار کم و معقولی کاهش می‌یابد.

در یک سیستم تعیین هویت، تصویر ورودی با تک‌تک داده‌های آموزشی مقایسه می‌شود. برای امکان‌سنجی استفاده از روش پیشنهادی در چنین سیستمی، آزمایش دیگری بر روی پایگاه داده *FVC2002* صورت گرفت. در این آزمایش، تصویر آخر از هر گروه، به‌عنوان داده‌ی آزمایشی (Test) تعیین گردید و تعداد مختلفی از تصاویر هر گروه به‌عنوان داده‌ی آموزشی (Train) بررسی شد. در هیچ یک از حالات آزمون، تصویر آزمایشی وارد داده‌های آموزشی نشده است.

جدول ۴، نتیجه‌ی آزمایشات انجام شده را نشان می‌دهد. در هر سطر، دقت با در نظر گرفتن تعداد خاصی از داده‌های آموزشی ( $i$ ) و در هر ستون با مقدار متفاوتی از  $k$  محاسبه شده است. در سطر اول از جدول ( $i=1$ )، تنها یک تصویر از هر گروه، به‌عنوان داده‌ی آموزشی در نظر گرفته شده است. بنابراین، در این حالت ۱۰۰ داده‌ی آموزشی و ۱۰۰ داده‌ی آزمایشی وجود خواهد داشت. به ازای هر تصویر ورودی، ۱۰۰ عمل مقایسه (انطباق) صورت می‌گیرد. برای  $k=1$ ، تصویری که دارای بالاترین امتیاز است به‌عنوان تصویر منطبق در نظر گرفته می‌شود. به‌صورت کلی به تعداد  $k$  بالاترین امتیاز در نظر گرفته می‌شوند. برای نمونه در حالت  $k=3$ ، اگر گروه واقعی تصویر در یکی از سه تصویر با بالاترین امتیاز قرار داشته باشد، یک واحد به تعداد تصمیمات درست اضافه خواهد شد.

جدول ۴: نتیجه‌ی کلاس‌بندی با استفاده از روش پیشنهادی

$i \downarrow$	$k=1$	$k=2$	$k=3$
1	90%	92%	93%
2	94%	94%	96%
3	97%	98%	98%
4	99%	100%	100%
5	100%	100%	100%
6	100%	100%	100%
7	100%	100%	100%

انتخاب گردیده و از هر کدام از آن‌ها، پس از اعمال تغییرات زیر، هفت تصویر جدید ایجاد شده است. شکل ۱۶، نمونه‌ای از این تصاویر را نشان می‌دهد.

- بریدن قسمتی از سمت چپ تصویر (شیفت به چپ)
- بریدن قسمتی از سمت راست تصویر (شیفت به راست)
- بریدن قسمتی از پایین تصویر (شیفت به پایین)
- بریدن قسمتی از بالای تصویر (شیفت به بالا)
- چرخش تصویر به سمت چپ
- چرخش تصویر به سمت راست
- چرخش تصویر به اندازه ۱۸۰ درجه



شکل ۱۶: تصاویر سفارشی جهت آزمون مقاومت روش پیشنهادی نسبت به چرخش و جابه‌جایی

جدول ۲: نتیجه‌ی مقایسه تصاویر با در نظر گرفتن مثلث‌های متشابه

الف: تصویر اول

$1_1$	1_2	1_3	1_4	1_5	1_6	1_7	1_8
	10616	12573	3309	18523	3253	1678	2502
	2-2	2-3	2-4	2-5	2-6	2-7	2-8
	211	285	435	269	287	437	421

ب: تصویر دوم

$2_1$	2_2	2_3	2_4	2_5	2_6	2_7	2_8
	3453	2353	699	1941	1241	434	623
	1-2	1-3	1-4	1-5	1-6	1-7	1-8
	127	145	134	172	117	78	207

جدول ۳: نتیجه‌ی مقایسه تصاویر با در نظر گرفتن مثلث‌های متشابه و

منطبق

الف: تصویر اول

$1_1$	1_2	1_3	1_4	1_5	1_6	1_7	1_8
	9525	11507	2054	17123	2097	1027	949
	2_2	2_3	2_4	2_5	2_6	2_7	2_8
	2	7	11	7	6	11	9

ب: تصویر دوم

$2_1$	2_2	2_3	2_4	2_5	2_6	2_7	2_8
	3126	2119	446	1752	928	238	139
	1_2	1_3	1_4	1_5	1_6	1_7	1_8
	2	2	3	2	2	1	9



شکل ۱۷: تفاوت ظاهری تصاویر هم‌گروه. ستون اول: تصاویر آموزشی  
ستون دوم: تصاویر آزمایشی

با افزایش  $k$  در هر سطر، دقت افزایش می‌یابد. هنگامی که تعداد تصاویر آموزشی به چهار عدد می‌رسد، دقت به ۱۰۰٪ افزایش می‌یابد. در سطر و ستون اول ( $k = i = 1$ )، علت پایین بودن نسبی دقت، تفاوت ظاهری زیاد بین تصویر آموزشی و تصویر آزمایشی است. گاه در تصاویر یک پایگاه داده، تشخيص شباهت و تفاوت ظاهری، حتی با چشم انسان نیز مقدور نیست. شکل ۱۷، نمونه‌ای از این تصاویر را نشان می‌دهد. مجموعه مینوشیای استخراج شده از این دو زوج تصویر بسیار متفاوت است.

دقت به‌دست آمده از جدول ۳، نشان دهنده‌ی پایداری روش پیشنهادی است. کیفیت و نوع اعوجاج تصاویر پایگاه داده‌ی  $FVC$  به شکلی است که دقت نسبتاً بالا بر روی داده‌های آن، برابر با دقت بسیار بالا در سیستم‌های کاربردی بشمار می‌رود. برای امکان‌سنجی استفاده از روش پیشنهادی در یک سیستم تصدیق هویت، آزمایشی مطابق با پروتکل پایگاه داده‌ی  $FVC$  صورت گرفت. از آنجا که تصاویر پایگاه داده‌ی  $FVC2002$  دارای مقیاس مشابه‌ای می‌باشند، محاسبه مثلث‌های متشابه با قانون سه‌ضلع انجام شده است. در مرحله‌ی انطباق مثلث‌ها، دو حد آستانه تعریف می‌شود. یکی برای اضلاع مثلث‌های متشابه و دیگری برای خطوط متصل کننده‌ی آن‌ها که هر دو به صورت تجربی بدست آمده‌اند و در این آزمایش برابر با ۱ پیکسل قرار داده شده‌اند. در جدول ۵، پارامترهای  $FMR100$ ،  $FMR1000$  و  $EER$  محاسبه شده برای سیستم پیشنهادی نمایش داده شده است. بر اساس پروتکل‌های پایگاه‌های داده‌ی  $FVC$ ، دو پارامتر  $FMR100$  و  $FMR1000$  برابرند با مقدار  $FNMR$  هنگامی که  $FMR = 1/100$  و  $FMR = 1/1000$  باشد. در این جدول، روش پیشنهادی با چند روش جدید دیگر مورد مقایسه قرار گرفته است و همان‌طور که مشاهده می‌شود، روش پیشنهادی بهتر از سایر روش‌های مقایسه شده عمل کرده است و اختلاف قابل ملاحظه‌ای با اغلب آنها دارد.

جدول ۵: نتایج آزمایشات سیستم تصدیق هویت

$EER$	$FMR1000$	$FMR100$	سیستم‌های انطباق اثرانگشت
2.81	6.75	4.18	سیستم [۹]
1.32	2.77	1.53	سیستم [۶]
3.5	9.2	5.3	سیستم [۲]
1.06	--	--	سیستم [۱۹]
2.5	1.7	1.5	سیستم [۴]
0.9	1.14	0.8	سیستم پیشنهادی

جدول ۶: زمان اجرای هر یک از بخش‌های سیستم پیشنهادی

بخش	زیربخش	زمان هر زیربخش (ms)	زمان کل بخش (ms)
پیش‌پردازش	قطعه‌بندی	49	2270
	بهبود	2200	
	باریک‌سازی	21	
انطباق	ایجاد مثلث	86	566
	تشخیص تشابه	310	
	تشخیص انطباق	170	

جدول ۷: مقایسه زمان ترتیبی و موازی بخش‌های مختلف سیستم پیشنهادی

بخش	زمان ترتیبی (ms)	زمان موازی (ms)	افزایش سرعت هر بخش	افزایش سرعت کل
پیش‌پردازش (بهبود)	2200	85	25.88	8.95
انطباق	566	112	5.05	

زمان صرف شده توسط روش پیشنهادی برای مقایسه دو تصویر اثرانگشت با ابعاد  $374 \times 388$  به صورت میانگین برابر با 460ms می‌باشد. با مقایسه این زمان با میانگین ۸ روش اول از بهترین روش‌های گزارش شده در سایت پایگاه داده‌ی  $FVC2002$  که برابر با 1.21s است، به‌سادگی می‌توان نتیجه گرفت که روش پیشنهادی در دسته‌ی روش‌های پر سرعت قرار می‌گیرد.

- [2] X. Li and F. Yu, "A new fingerprint matching algorithm based on minutiae," in *Communication Technology and Applications*, 2009, pp. 869-873.
- [3] A. R. Patil and M. A. Zaveri, "A Novel Approach for Fingerprint Matching Using Minutiae," 2010, pp. 317-322.
- [4] J. Feng, "Combining minutiae descriptors for fingerprint matching," *Pattern Recognition*, vol. 41, pp. 342-352, 2008.
- [5] A. Lindoso, L. Entrena, J. Liu-Jimenez, and E. San Millan, "Correlation-based fingerprint matching with orientation field alignment," *Advances in Biometrics*, pp. 713-721, 2007.
- [6] H. Wei, M. Guo, and Z. Ou, "Fingerprint Verification Based on Multistage Minutiae Matching," in *Pattern Recognition*, 2006, pp. 1058-1061.
- [7] P. Shi, J. Tian, Q. Su, and X. Yang, "A novel fingerprint matching algorithm based on minutiae and global statistical features," 2007, pp. 1-6.
- [8] R. Prabhakar and C. V. K. Rao, "A parallel algorithm for fingerprint matching," in *TENCON '89*, 1989, pp. 373-376.
- [9] X. Jiang and W. Y. Yau, "Fingerprint minutiae matching based on the local and global structures," in *Pattern Recognition*, 2000, pp. 1038-1041 vol. 2.
- [10] Z. M. Kovacs-Vajna, "A fingerprint verification system based on triangular matching and dynamic time warping," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, pp. 1266-1276, 2000.
- [11] G. Parziale and A. Niel, "A fingerprint matching using minutiae triangulation," *Biometric Authentication*, pp. 1-50, 2004.
- [12] W. Xu, X. Chen, and J. Feng, "A robust fingerprint matching approach: Growing and fusing of local structures," *Advances in Biometrics*, pp. 134-143, 2007.
- [13] G. Aguilar, G. Sanchez, K. Toscano, M. Nakano-Miyatake, and H. Perez-Meana, "Automatic Fingerprint Recognition System Using Fast Fourier Transform and Gabor Filters," *Cientifica*, pp. 9-16, 2008.
- [14] D. Maltoni, D. Maio, A. K. Jain, and S. Prabhakar, *Handbook of fingerprint recognition*: Springer-Verlag New York Inc, 2009.
- [15] D. Maio, D. Maltoni, R. Cappelli, J. L. Wayman, and A. K. Jain, "FVC2002: Second fingerprint verification competition," in *Pattern Recognition*, 2002, pp. 811-814 vol. 3.
- [16] T. M. Bernard and A. Manzanera, "Improved low complexity fully parallel thinning algorithm," in *Image Analysis and Processing, 1999. Proceedings. International Conference on*, 1999, pp. 215-220.
- [17] L. Hong, Y. Wan, and A. Jain, "Fingerprint image enhancement: algorithm and performance evaluation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 20, pp. 777-789, 1998.
- [18] D. Kirk, "NVIDIA CUDA software and GPU parallel computing architecture," in *International Symposium on Memory Management*, 2007, pp. 103-104.
- [19] C. Wu, "Advanced feature extraction algorithms for automatic fingerprint recognition systems," Citeseer, 2007.

نتایج به دست آمده از ابزار پروفایلر، در جدول ۶ نشان داده شده است. بخش‌های بهبود، تشکیل مثلث‌ها و تشخیص مثلث‌های متشابه و منطبق، به صورت موازی پیاده‌سازی شده‌اند. نتایج موازی‌سازی این بخش‌ها در جدول ۷ نمایش داده شده است. افزایش سرعت کل به دست آمده، تقریباً نه برابر حالت ترتیبی است.

### نتیجه‌گیری

در این مقاله، پس از بررسی روش‌های مختلف انطباق، مخصوصاً روش‌های مبتنی بر مینوشیا، روش مثلثی معرفی شد. این روش، از مثلث‌های تشکیل شده از اتصال مینوشیاها برای انطباق بهره می‌برد. در سیستم پیشنهادی، نسخه‌ی جدیدی از انطباق مثلثی ارائه شد که بدون نیاز به استخراج جهت مینوشیاها و هم‌جهت کردن تصاویر، انطباق دو اثرانگشت را امکان‌پذیر می‌ساخت. برای اثبات کارایی روش پیشنهادی در جنبه‌های مختلف، چند آزمایش بر روی پایگاه داده‌ی شناخته‌شده FVC2002 (بخش DB1-A) صورت گرفت. نتایج حاصل شده همگی نشان دهنده‌ی سرعت و دقت بالای روش پیشنهادی می‌باشند. روش پیشنهادی مورد استفاده در بخش انطباق به نسبت دیگر روش‌های موجود دارای مزایای زیر می‌باشد:

- عدم نیاز به جهت مینوشیا و در نتیجه بالا بردن سرعت و دقت
- عدم نیاز به هم‌جهت کردن دو تصویر و در نتیجه بالا بردن سرعت و دقت
- اجتناب از مقایسه تعداد بسیار زیاد مثلث‌ها
- اصلاح روش مثلثی ساده و رفع خطای مثلث‌های متشابه و نامنطبق

پس از کسب اطمینان از دقت مطلوب سیستم پیشنهادی، بخش‌های زمانبر سیستم تعیین شده و سپس با استفاده از CPU و GPU موازی‌سازی شدند. پس از موازی‌سازی، زمان اجرای سیستم تا نه برابر کاهش یافت. این امر، روش پیشنهادی را برای استفاده در کاربردهای برخط کاملاً ایده‌آل می‌سازد.

### مراجع

- [1] J. Chen, F. Chan, and Y. S. Moon, "Fingerprint Matching with Minutiae Quality Score," *Advances in Biometrics*, vol. 4642, pp. 663-672, 2007.