المجلة العراقية للعلوم الإحصائية (20) 2011
عدد خاص بوقائع المؤتمر العلمي الرابع لكلية علوم الحاسوب والرياضيات
ص ص [691-713]

التمييز باستخدام الترميز التسلسلي

رهام جاسم عيسى*   سندس خليل إبراهيم **

الملخص:

عملية تمييز الأشكال في المعالجة تجعل بيانات الشكل مفيدة للحاسوب وانها تؤدي دورا مهما في مجال تمييز الأنماط والرؤيا الحاسوبية, ان التقاط الصورة هي عملية ليست دقيقة من حيث البعد عن الجسم, لذا فان التقاط صور لكائنات من داخل المركبة مثلا سيؤدي إلى تغير في بعد الكائن عن المركبة المتحركة لكل صورة تلتقط ومن ثم تغير في حجم صورة الكائن داخل الصورة,. وبما ان تقنيات الترميز التسلسلي Chain Code تعد التقنية الأكثر استخداما في تمثيل الأشكال. لذا فقد تم في هذا البحث تمييز الأشكال داخل الصورة باستخدام الترميز التسلسلي, إذ تم عرض أهم المشاكل التي تعاني منها تطبيقات تمييز الأنماط باستخدام الترميز التسلسلي وطرائق حلها, وقد تم تطبيق الترميز التسلسلي على صور الأشكال الهندسية المنتظمة و الأشكال غير المنتظمة, وتم أيضا حل مشكلة تغير الحجم بطريقة مقترحة, وقد أظهرت الطريقة كفاءة عالية في تمييز الأشكال المنتظمة وغير المنتظمة ذات الأحجام المختلفة وأظهرت أيضا كفاءة عالية بالمقارنة مع طريقة Rehman لمعالجة تغير الحجم باستخدام الترميز التسلسلي .

* طالبة ماجستير/كلية علوم الحاسوب والرياضيات/جامعة الموصل
** مدرس/كلية علوم الحاسوب والرياضيات/جامعة الموصل

تاريخ التسلم :2011/10/1 ــــــــــ تاريخ القبول : 21 /12 /2011

Recognition using Chain Code

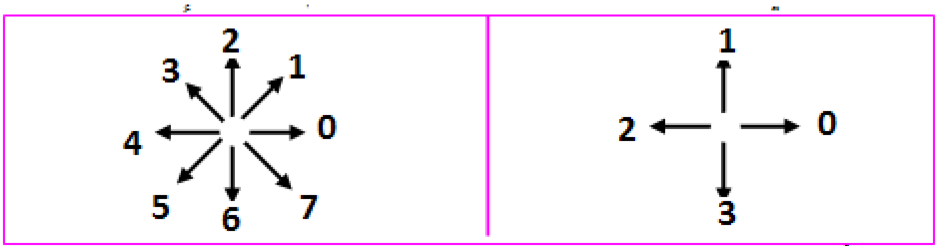
Abstract

The process of recognizing the processed patterns makes the pattern data useful for the computer. It plays a significant role in the field of pattern recognition and computer vision. So, the image capture is an inaccurate process in terms of the distance from the object. Therefore, the image capturing for object from the inside of a moved vehicle, for example, will lead to a change in the distance between the object and the vehicle for each captured image and finally a change in the size of the object image size inside the image. Viewing that the techniques of chain coding are considered the most used in presenting. Therefore, this study has recognized the patterns inside the image by using the chain coding .In the paper, the most important problems that the application of recognizing patterns by using the chain coding and the methods of solving them have been presented .The chain coding has been applied on the images of the regulated and the unregulated geometrical patterns .The problem of changing size has been solved by a suggested method and this method has proved to be of high efficiency in recognizing the patterns of various sizes .Also, the method has proved a higher efficiency compared with Rehman to process the size changing in which the chain coding is used.

1. المقدمة

عملية تمثيل الأشكال المعالجة تجعل بيانات الشكل مفيدة للحاسوب وأنها تلعب دورا مهما في مجال تمييز الأنماط والرؤيا الحاسوبية (Shahab,2009), والتي تستخدم في كثير من التطبيقات وفي عدة مجالات منها الطب, والتصنيع, واستكشاف الفضاء, وغيرها. وتصنف خصائص الشكل إلى خصائص الحدود وخصائص المنطقة.إذ يمكن استخلاص خصائص الحدود من حدود الشكل مثل الحواف والزاوية, في حين تستخلص خصائص المنطقة من المنطقة التي يشغلها الشكل مثل المساحة (Sleit), وتعد تقنيات الترميز التسلسلي الأكثر استخداما في تمثيل الأشكال كما تعد صيغة الإدخال القياسية للعديد من

خوارزميات تحليل الأشكال وتمييز الأنماط, اذ تمتاز بسهولة التعامل معاه كما انها لا تتطلب سعة خزنيه كبيرة (Rehman,2010), ويمكن استخدام الترميز التسلسلي في ترميز نقاط الشاشة التي تمثل الحدود عن طريق خزن تتابع متجه وحدة الطول ومجموعة محددة من الاتجاهات(Sleit), اذ يمكن استخدامها في تمثيل الحدود الخارجية للأشكال بالاعتماد على أربعة مقاطع متصلة 4_Connectivity of the (Segments) أو ثمانية مقاطع متصلة (8_connectivity of the segments) يرمز كل اتجاه برقم معين يعبر عن موقع ذلك المقطع في مخطط السلسلة (الخطيب,2011) انظر الى الشكل(1). وقد تم تقديم أول ترميز تسلسلي من قبل(Freeman) في سنة 1961 وسمي Freeman Chain Code(FCC), والذي يعد أساسا لبقية أنواع الترميز التسلسلي التي ظهرت فيما بعد (Junding,2009). إن هدف البحث هو دراسة طريقة الترميز التسلسلي بوصفها طرائق تمييز الأنماط وبيان أهم مشاكله وإيجاد الحلول لها.



(ب)8 - اتصالات بين المقاطع (أ) 4- اتصالات بين المقاطع

الشكل (1) مخطط ترميز السلسلة

يستخدم الترميز التسلسلي في عدة تطبيقات ومنها استخدامه في تمييز الأنماط (Pattern Recognition) ومطابقة الأشكال (Shape Matching) وكما يستخدم في تحديد الحركة(Motion Analysis) وفي تحليل الاتصال (connectivity Analysis) وتحليل الخط المستقيم (Straight Line Determination) وفي تصنيف الهدف (Classification Target) (Bose,2000). وقد تم تقديم العديد من البحوث في هذا المجال, وعلى الرغم من ذلك تبقى هذه الأساليب في تحسين مستمر. ففي العام(2008) قدم الباحث Wanitchai et al. خوارزمية لكشف وتمييز العلامات المرورية

التحذيرية, إذ تم كشف العلامة المرورية باستخدام الفضاء اللوني(HSV) عن طريق تقنيات المنطق المضبب (Fuzzy Logic), أما عملية التمييز فقد تمت باستخدام الترميز التسلسلي للعلامة المرورية واستخدام خوارزمية السلسلة الثانوية المشتركة الطولى (Longest Common Subsequence Algorithm) لمقارنته مع قاعدة بيانات تحوي ترميزاً تسلسلياً لعدة علامات مرورية , وتم تطبيق هذه الخوارزمية على أكثر من 4000 صورة تحتوي علامة مرورية وكانت النتائج جيدة جدا(Wanitchai,2008).وفي العام (2009) طور الباحث Rehman خوارزمية لحل مشكلة تغير الحجم في الترميز التسلسلي لتمثيل حروف الـ(Nastaliq) , تم فيها تكوين مقاطع باستخدام نقاط الزاوية ومن ثم إيجاد الترميز التسلسلي بالاعتماد على المقاطع وليس على نقاط الشكل. وهذه الطريقة حققت نجاحا في تمثيل حدود أي شكل(Rehman,2009). وفي السنة نفسها تقدم الباحث Junding بطريقة مقارنة بين خمس طرائق للترميز التسلسلي وهي (Chain Code Histogram (CCH , وكذلك Minimize Statistical Direction Code (MSDC) , والتي لا تأخذ الخصائص الموزعة بنظر الاعتبار , فضلا عن إلى ثلاث طرائق مبتكرة في الوصف وهي : Chain Code Distribution Vector(CCDV) , و Chain Code Relativity Entropy(CCRE), وطريقة Chain Code Spatial Distribution entropy(CCSDE), إذ تم تطبيق هذه المقارنة على مجموعة كبيرة من الأشكال في قاعدة البيانات وقد دلت النتائج على كفاءة هذه الطريقة(Junding,2009). كما قدم الباحث .Raviraj et al في السنة نفسها أيضا طريقة لتمثيل الأشكال عن طريق دمج الترميز التسلسلي وطرائق وصف الخط , وهذه الطريقة تحقق تمثيلا جيدا للشكل من خلال استخدام طريقة دمج الخطوط بدلا من استخدام طريقة الاحداثيات القطبية هذه الطريقة بإمكانها تجاوز مشكلة تغير الحجم من خلال تصميم خوارزمية مطابقة متكيفة(Raviraj,2009). وفي العام (2010) تقدم الباحث .Lawal et al بطريقة لتمييز الأرقام العربية (الهندية)

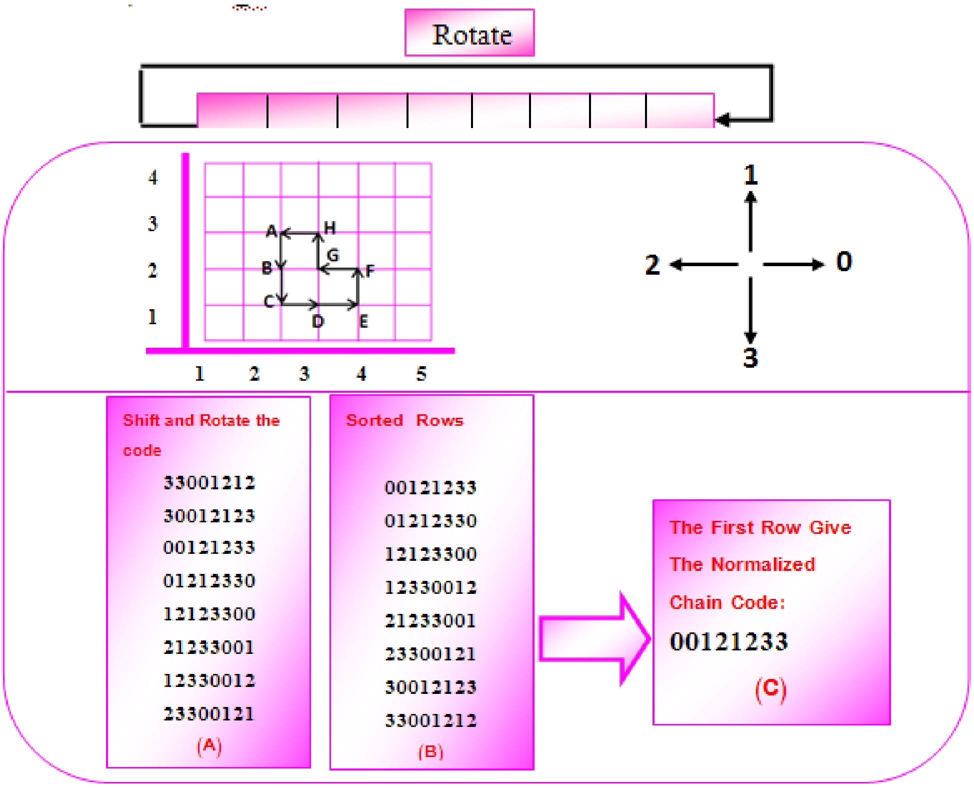
المكتوبة بخط اليد باستخدام الترميز التسلسلي (FCC) كما اعتمد في عملية التصنيف على(abductive network) لقاعدة بيانات تضم 21120 نموذجاً للكتابة بخط اليد للأرقام من (0-9) لـ 44 كاتبا. وقد كان معدل التمييز بهذه الطريقة 99% كحد أعلى(Lawal,2010). وفي العام (2011) قدم الباحث John et al. طريقة استخدم فيها الترميز التسلسلي لتمييز الحروف المنفصلة المكتوبة بخط اليد, إذ تم استخلاص خصائص الصورة من خلال استخدام الترميز التسلسلي و (Image Centroid) (John,2011). وفي السنة نفسها اقترح الباحث An et al. خوارزمية لاسترجاع معلومات حدود الشكل باستخدام التردد بالاعتماد على إعادة ترتيب الترميز التسلسلي ومن ثم تطبيع السلسلة , ومن ثم اعتماد أعلى تردد للسلسلة . وكانت هذه الطريقة فعالة في حل مشكلة التدوير ومشكلة تغير الحجم. كما إنها حققت سرعة في عملية استرجاع الشكل(An,2011).

3. مشاكل الترميز التسلسلي :

هناك بعض المشاكل التي تواجه الترميز التسلسلي وهي :

1. اختيار نقطة البداية (Start Point)
2. الدوران (Rotation)
3. تغير الحجم (Scaling)

إذ يجب أن تثبت نقطة البداية و زاوية الدوران والحجم من اجل تمثيل حدود الشكل.وقد تم تجاوز مشكلة نقطة البداية عن طريق عملية التطبيع (Normalization) لسلسلة الرموز, إذ يفترض أنها سلسلة دائرية من عدد من الاتجاهات, ويتم اختيار نقطة البداية بحيث تمثل اصغر عدد صحيح(Rehman,2010). وتتم هذه العملية عن طريق تزحيف الصف (السلسلة) باتجاه اليسار(Shift Left) مع التدوير (Rotate) انظر إلى الشكل(2), وتستمر العملية لكل رموز السلسلة حتى النهاية انظر الشكل (3-A), تليها عملية ترتيب السلاسل الناتجة ترتيبا تصاعديا من اليسار انظر إلى الشكل (3-B), إن أول صف يمثل اصغر متجه وهو يمثل السلسلة الصحيحة انظر إلى الشكل (3-C).



الشكل (3) حل مشكلة نقطة البداية

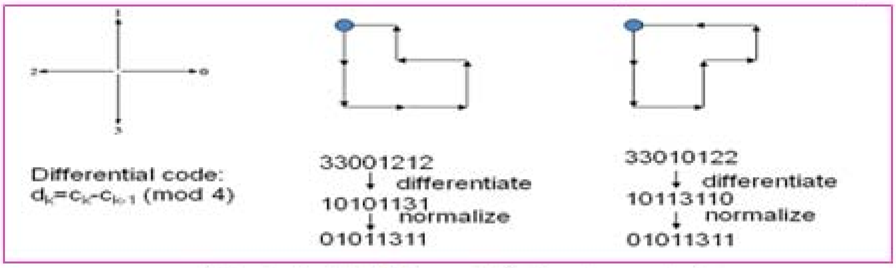
(A)عملية التزحيف (B) السلاسل بعد الترتيب (C) الترميز التسلسلي للشكل.

[فكرة المثال مستنبطة من المصدر(Shahab,2009)]

أما مشكلة التدوير فيمكن حلها باستخدام الـ (Differential chain code). و يتم حسابه عن طريق عملية طرح كل عنصر في السلسلة من العنصر الذي يسبقه ثم

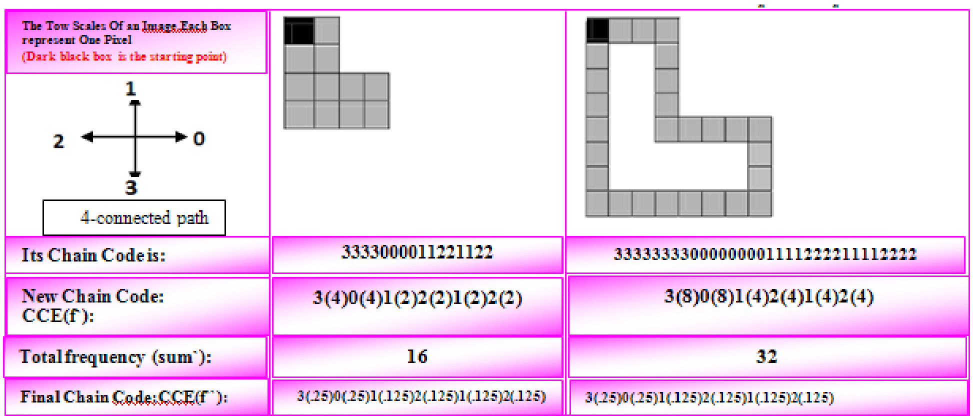
اخذ ناتج الطرح مقسوما على (N), إذ تمثل (N) المقاطع الثمانية أو المقاطع الأربعة المتصلة (Boundary Description and Representation,2007).

وكما في المثال الموضح في الشكل(4) .



الشكل (4) حل مشكلة التدوير (shahab,2009)

أما مشكلة تغير الحجم فلا يمكن أن تطبع كليا وذلك بسبب أن الشبكة محدودة, لذا فإنها تؤدي إلى اختلاف الرموز, هذه المسألة لم تحل بشكل كلي(Sleit), وقد قدم الباحث Rehman حلا لهذه المشكلة من خلال إيجاد الترميز التسلسلي للشكل, ثم إيجاد عدد تكرار كل عنصر في السلسلة, تتبعها عملية جمع التكرارات الناتجة ثم قسمة عدد تكرار كل عنصر على مجموع التكرارات للحصول على الترميز التسلسلي النهائي.كما في الشكل (5) (Binary Image Analysis,2011) .



الشكل (5) مثال عن الترميز التسلسلي ذي المقاطع الأربعة المتصلة (حل مشكلة تغير الحجم)
[فكرة المثال مستنبطة من المصدر (Rehman,2010)]

5 . خوارزمية الترميز التسلسلي المطبقة في هذا البحث :

تتناول هذه الفقرة الخوارزمية المتبعة لإيجاد الترميز التسلسلي للشكل عن طريق تحويل الصورة إلى متجه, انظر إلى المخطط الانسيابي كما موضح في الشكل (6), وذلك عن طريق عدة خطوات يمكن إدراجها بالاتي:

1. قراءة الصورة.

2. عمل تنحيف للشكل.
3. تحديد نقطة البداية ثم إيجاد نقاط حدود الشكل باتجاه عقارب الساعة وبشكل متتابع على شرط عدم تكرار أية نقطة.
4. حذف النقاط التي لا تقع على الشبكة.
5. حذف النقاط المتقاربة التي لا تؤثر في الشكل.
6. إيجاد الترميز التسلسلي.
7. خزن الترميز التسلسلي للشكل في ملف.

الشكل (6) المخطط الانسيابي العام لطريقة الترميز التسلسلي

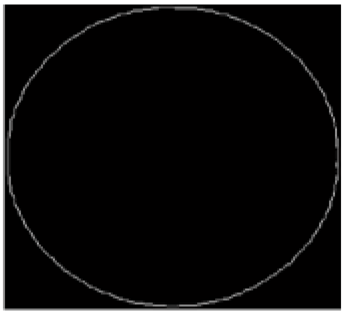
5. الجانب التطبيقي: مراحل إيجاد ألترميز التسلسلي

<u>1.5 قراءة الصورة :</u>

تم في هذه المرحلة قراءة الصورة ثم تنحيفها للحصول على هيكل الشكل ثم إيجاد احداثيات النقاط التي تكون قيمتها مساوية لـ(1) في الصورة الأصلية, كما موضح في الشكل (7), كما في الايعاز الأتي:

[xx,yy]=find(a);

إذ يمثل (xx) و(yy) احداثيات كل النقاط في الصورة .



الشكل (7) الصورة الأصلية

إيجاد نقطة البداية للشكل ق البحث عن اقل قيمة في
المتجه xx واقل قيمة فم قطة التي تمثل اصغر قيمة
لكل من x وy, كما في ا

```
x=min(xx);
y=min(yy);
```

3.5 إيجاد محيط الشكل:

يتم في هذه المرحلة تثبيت بقية النقاط التي تمثل محيط الشكل وخزنها وكما يأتي:
في البداية يتم خزن النقطة التي تلي نقطة البداية (x,y) وذلك بالبحث عن النقطة باتجاه عقرب الساعة والتي تمثل احد المواقع (x-1,y-1) او(x-1,y) او(x-1,y+1) او(x,y-1) او(x,y+1) وكما في الشكل (8):

x−1,y−1	x−1,y	x−1,y+1
x,y−1	x,y	x,y+1
X+1,y−1	x−1,y	x−1,y+1

الشكل (8) النقاط التي يجب البحث فيها

فإذا كان مجموع هذه النقاط اكبر من القيمة (0) فان احدى هذه النقاط ستكون قيمتها(1) أي أنها احدى نقاط الشكل والتي يجب أن تخزن, انظر إلى المقطع البرمجي الأتي :

```
k=1;
xxp(k)=x;
yyp(k)=y;
k=k+1;
if (a(x-1,y-1)+a(x-1,y)+a(x-1,y+1)+a(x,y-1)+a(x,y+1) >0)
```

```
      if (a(x-1,y-1)==1)
       [xxp,yyp,k,x,y]= fill_x_y (xxp,yyp,x-1,y-1,k);
      elseif (a(x-1,y)==1),
         [xxp,yyp,k,x,y]= fill_x_y (xxp,yyp,x-1,y,k);
        elseif (a(x-1,y+1)==1)
           [xxp,yyp,k,x,y]= fill_x_y (xxp,yyp,x-1,y+1,k);
          elseif (a(x,y-1)==1)
            [xxp,yyp,k,x,y]= fill_x_y (xxp,yyp,x,y-1,k);
            elseif (a(x,y+1)==1)
              [xxp,yyp,k,x,y]= fill_x_y (xxp,yyp,x-1,y+1,k);
             end
end
```

وتكرر العملية السابقة لإيجاد بقية نقاط محيط الشكل ولكن سيتم البحث عن المجاورات الثمانية لإيجاد النقطة التالية وكما في المقطع البرمجي :

```
flag=1;
ii=0;
while   ((flag==1)&&(ii<=LL))
    flag=0;
    if (( a(x-1,y-1)+a(x-1,y)+a(x-1,y+1)+...
         a(x,y-1)+a(x,y+1)+...
         a(x+1,y-1)+a(x+1,y)+a(x+1,y+1))>1 )
        flag=1;
        ii=ii+1;
        if     (a(x-1,y-1)==1)&&      (find_pixel_in_xxp_yyp(xxp,yyp,x-1,y-
1,k)==0)
          [xxp,yyp,k,x,y]= fill_x_y (xxp,yyp,x-1,y-1,k);
         elseif           (a(x-1,y)==1)&&(find_pixel_in_xxp_yyp(xxp,yyp,x-
1,y,k)==0)
          [xxp,yyp,k,x,y]= fill_x_y (xxp,yyp,x-1,y,k);
         elseif        (a(x-1,y+1)==1)&&(find_pixel_in_xxp_yyp(xxp,yyp,x-
1,y+1,k)==0)
          [xxp,yyp,k,x,y]= fill_x_y (xxp,yyp,x-1,y+1,k);

         elseif          (a(x,y-1)==1)&&(find_pixel_in_xxp_yyp(xxp,yyp,x,y-
1,k)==0)
          [xxp,yyp,k,x,y]= fill_x_y (xxp,yyp,x,y-1,k);
         elseif
(a(x,y+1)==1)&&(find_pixel_in_xxp_yyp(xxp,yyp,x,y+1,k)==0)
          [xxp,yyp,k,x,y]= fill_x_y (xxp,yyp,x,y+1,k);
```

```
        elseif  (a(x+1,y-1)==1)&&(find_pixel_in_xxp_yyp(xxp,yyp,x+1,y-1,k)==0)
         [xxp,yyp,k,x,y]= fill_x_y (xxp,yyp,x+1,y-1,k);
        elseif (a(x+1,y)==1)&&(find_pixel_in_xxp_yyp(xxp,yyp,x+1,y,k)==0)
         [xxp,yyp,k,x,y]= fill_x_y (xxp,yyp,x+1,y,k);
        elseif (a(x+1,y+1)==1)&&(find_pixel_in_xxp_yyp(xxp,yyp,x+1,y+1,k)==0)
         [xxp,yyp,k,x,y]= fill_x_y (xxp,yyp,x+1,y+1,k);
      end
   end
end
```

حيث تمثل (xpp) و(ypp) إحداثيات نقاط محيط الشكل, و (fill-x-y) تمثل دالة تقوم بخزن النقطة التي تم إيجادها والانتقال إلى موقع جديد انظر المقطع البرمجي :

```
function[xxp1,yyp1,k,x,y]= fill_x_y (xxp1,yyp1,x1,y1,k1)
   xxp1(k1)=x1;
   yyp1(k1)=y1;
   k=k1+1;
   x=x1;
   y=y1;
end
```

بينما (find_pixel_in_xxp_yyp) فهي تمثل الدالة التي تبحث عن النقطة , حيث تعيد الدالة قيمة ret فأن وجدت النقطة فأن قيمة ret ستكون مساوية للواحد اذا لم تجد النقطة فان قيمة ret التي ترجعها الدالة ستكون صفراً, و كما في المقطع البرمجي الأتي :

```
function ret=find_pixel_in_xxp_yyp(xxp,yyp,x,y,k)
i=2;
ret=0;
while i<k
   if ((xxp(i)==x) && (yyp(i)==y))
      ret=1;
      i=k;
   else
      i=i+1;
   end
end
```

4.5 معالجة تغير الحجم:

إن الترميز التسلسلي يتغير بتغير حجم الشكل وهذه إحدى المشاكل التي تواجه الترميز التسلسلي, إذ يجب أن يكون الترميز التسلسلي للشكل ثابتاً في حال تغير حجم الشكل زيادة أو نقصاناً بحيث يمكن استخدامه في تمثيل وتمييزها الأشكال. إن الفكرة الأساسية المتبعة في هذا البحث لحل مشكلة التغير في حجم الشكل تتمثل بـ تقسيم حجم الشكل على عدد ثابت يمثل عدد المقاطع التي يرغب فيها المستخدم, وذلك من اجل الحصول حجم الشبكة (أبعاد الشبكة) بحيث يبقى الترميز التسلسلي ثابت للشكل نفسه مهما تغير حجمه. وتحسب أبعاد الشبكة الخاصة بالشكل وذلك بقسمة أبعاد الشكل (عدد الصفوف وعدد الأعمدة)على عدد ثابت يقوم بتحديده المستخدم وليكن مثلا (10) وذلك للتخلص من مشكلة تغير الحجم, وكما في المقطع البرمجي الأتي :

```
row_dim_grid=ceil(row/10);
col_dim_grid=ceil(col/10);
```

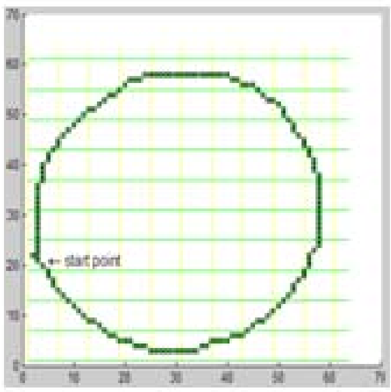
ومن ثم رسم الشبكة, كما هو موضح في الشكل (9), الذي يمثل:

row_dim_grid : عدد الصفوف في الشبكة .

col_dim_grid: عدد الأعمدة في الشبكة.

row : عدد الصفوف في الشكل.

col : عدد الأعمدة في الشكل.



الشكل (9) تحديد نقطة بداية الترميز

5.5 حذف النقاط التي لا تقع على الشبكة:

أ- تحديد النقاط :تختص هذه المرحلة بإيجاد النقاط التي تتقاطع مع الشبكة وذلك بتقسيم احداثيات كل نقطة في المتجهات (xxp) و(yyp) التي تم إيجادها سابقا على حجم الشبكة ومن ثم اخذ ناتج باقي القسمة فإذا كان ناتج باقي القسمة لا يساوي (1) أي ان النقطة لا تقع على الشبكة , سيتم تصفير ذلك الموقع كما في الايعازات الآتية :

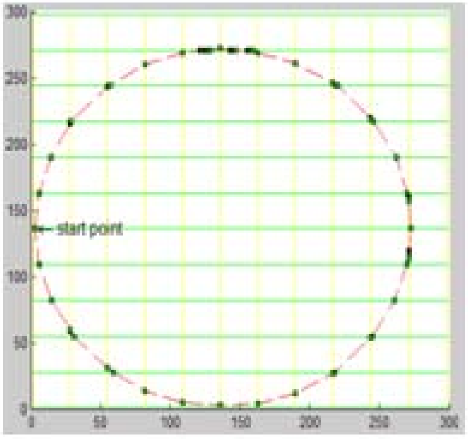
```
len=length(xxp);
for g=1: len
if  (mod(xxp(g),row_dim_grid)~=1)&& (mod(yyp(g),col_dim_grid)~=1)
       xxp(g)=0;
       yyp(g)=0;

   end
end
```

ب- حذف النقاط :بعد تصفير النقاط التي تقع خارج الشبكة, يتم حذف النقاط التي تكون قيمتها صفراً, كما في الشكل (10), وذلك بتكوين متجه لقيم x التي تساوي (1) والمتمثلة بالمتغير(xpp) ومتجه لقيم y التي تساوي (1) والمتمثلة بالمتغير (ypp) أما القيم التي تساوي صفراً فيتم إهمالها, وكما في المقطع البرمجي الأتي:

```
[cx,cy]=find(xxp);
clen=length(cx);
xpp=zeros(clen,1);
ypp=zeros(clen,1);
 m=0;
for n=1:len
   if (xxp(n)>0)
      m=m+1;
      xpp(m)=xxp(n);
      ypp(m)=yyp(n);
   end
end
 m=m+1;
```

ومن ثم خزن النقطة الأولى مرة أخرى لغلق الشكل كما في الإيعاز الأتي:

```
xpp(m)=xpp(1);
 ypp(m)=ypp(1);
```

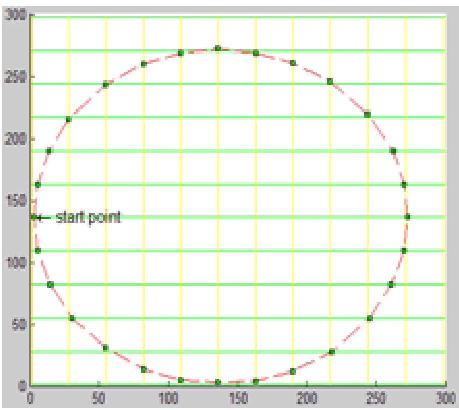


الشكل (10) حذف النقاط التي قيمتها صفر

6.5 حذف النقاط التي لا تؤثر في الشكل:

في هذه المرحلة تم حذف النقاط المتقاربة التي لا تؤثر في الشكل, وكما هو موضح في الشكل (11), وذلك بحساب المسافة بين كل نقطة والنقطة التي تسبقها فإذا كانت المسافة اقل من قيمة (min_distance) ستهمل هذه النقطة, وإلا سيحتفظ بإحداثيات كل نقطة في (xp) (yp) . وتم حساب قيمة (min_distance) وذلك بأخذ اقل قيمة بين (row_dim_grid) و (col_dim_grid) والتي تمثل أبعاد الشبكة, وكما في المقطع البرمجي الآتي :

```
e=1;
xp(e)=xpp(1);
yp(e)=ypp(1);
min_distance=min(row_dim_grid,col_dim_grid);
for g=2: length(xpp)
    D=find_distance(xp(e),yp(e),xpp(g),ypp(g));
    if (D>= min_distance)
        e=e+1;
        xp(e)=xpp(g);
        yp(e)=ypp(g);
     end
end
```



الشكل(11) حذف النقاط المتتابعة

7.5 إيجاد الترميز التسلسلي

وفيما يأتي الخطوات التي تم تطبيقها للحصول على الترميز التسلسلي للشكل:

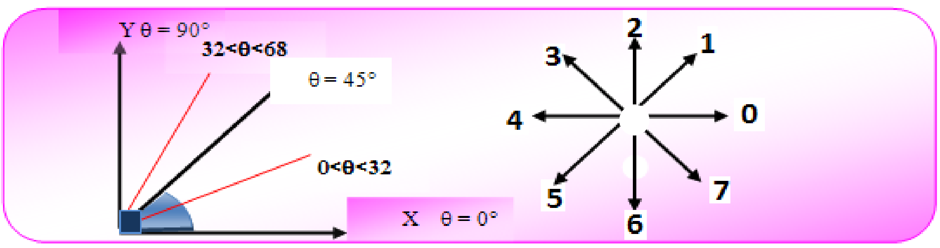
أ- إيجاد زاوية الميل:

بعد إيجاد النقاط التي ستستخدم في حساب الترميز التسلسلي للشكل نقوم بإيجاد قيمة زاوية الميل عن المحور(x), انظر المقطع البرمجي الأتي:

```
theta1=atand(dy/dx);
theta=abs(theta1);
```

يليها إجراء عملية تقريب قيمة الزاوية إلى اقرب (45°) فإذا كانت قيمة الزاوية اكبر من أو تساوي (0°) و اقل من أو تساوي (32°) عندها تكون قيمة الزاوية (0°). أما إذا كانت قيمة الزاوية اكبر من (32°) و اقل من (68°) تعد قيمة الزاوية (45°) وألا فان قيمة الزاوية هي (90°), كما في الشكل(12), كما في الايعازات الآتية:

```
if ((theta>=0)&&(theta<=23))
   theta=0;
elseif ((theta>23)&&(theta<68))
   theta=45;
  else theta=90;
end
```



الشكل (12) تقريب الزاوية في الربع الذي تقع فيه

ب- إيجاد الربع الذي يقع فيه مقطع الطول

بعد عملية تقريب الزاوية نقوم بإيجاد الربع الذي تقع فيه النقطة والرمز التسلسلي الذي سيمثل تلك النقطة في ذلك الربع انظر إلى الشكل (12), فإذا كانت قيمة المتغير dx و dy موجبة وكانت الزاوية تساوي (45°) فان الرمز التسلسلي لهذه النقطة سيكون (5) أما إذا كانت الزاوية تساوي (90°) فسيكون الرمز التسلسلي يساوي (6) في حين إذا كانت الزاوية (0°) فسيكون الرمز التسلسلي للنقطة يساوي (4). كما في الايعازات الآتية :

```
if ((dy>0)&&(dx>0))
   if (theta==45)          chain_cod=5;
     elseif (theta==90)  chain_cod=6;
     else                        chain_cod=4;
   end
end
```

وفي حالة أن قيمة المتغيرdy وdx سالبة وان الزاوية تساوي (45°) فان الرمز التسلسلي لهذه النقطة سيكون (1) أما إذا كانت الزاوية تساوي (90°) سيكون الرمز التسلسلي يساوي (2) , وإذا كانت الزاوية (0°) سيكون الرمز التسلسلي للنقطة يساوي (0) كما في الايعازات الآتية :

```
if ((dy>0)&&(dx>0))
   if (theta==45)          chain_cod=5;
     elseif (theta==90)  chain_cod=6;
     else                        chain_cod=4;
   end
end
```

أما في حالة أن قيمة المتغير dy سالبة وقيمة المتغير dy وكانت الزاوية تساوي (45°) فان الرمز التسلسلي لهذه النقطة سيكون (3) أما إذا كانت الزاوية تساوي (90°) سيكون الرمز التسلسلي يساوي (2) , وإذا كانت الزاوية (0°) سيكون الرمز التسلسلي للنقطة يساوي (4) كما في الايعازات الآتية :

```
if ((dy<0)&&(dx>0))
   if (theta==45)          chain_cod=3;
     elseif (theta==90)  chain_cod=2;
     else                        chain_cod=4;
```

```
  end
end
```

وفي حالة أن قيمة المتغير dy موجبة وقيمة المتغير dy سالبة وكانت الزاوية تساوي (45°) فان الرمز التسلسلي لهذه النقطة سيكون (7) أما إذا كانت الزاوية تساوي (90°) سيكون الرمز التسلسلي يساوي (6) , وإذا كانت الزاوية (0°) سيكون الرمز التسلسلي للنقطة يساوي (0) كما في الايعازات الآتية :

```
if ((dy>0)&&(dx<0))
  if (theta==45)          chain_cod=7;
    elseif (theta==90)   chain_cod=6;
    else                  chain_cod=0;
  end
end
```

وإذا كان المتغير dy موازياً للمحور y وكانت قيمة المتغير dx سالبة فان الرمز التسلسلي للنقطة سيكون (0) أما إذا كانت قيمة dx موجبة فسيكون الرمز التسلسلي (4)انظر إلى المقطع البرمجي الأتي:

```
if ((dy==0))
  if(dx<0)      chain_cod=0;
  else          chain_cod=4;
  end
end
```

أما إذا كان المتغير dx موازياً للمحور x وكانت قيمة المتغير dy سالبة فان الرمز التسلسلي للنقطة سيكون (2) وإذا كانت dy موجبة سيكون الرمز التسلسلي (6). انظر إلى الشكل (12).

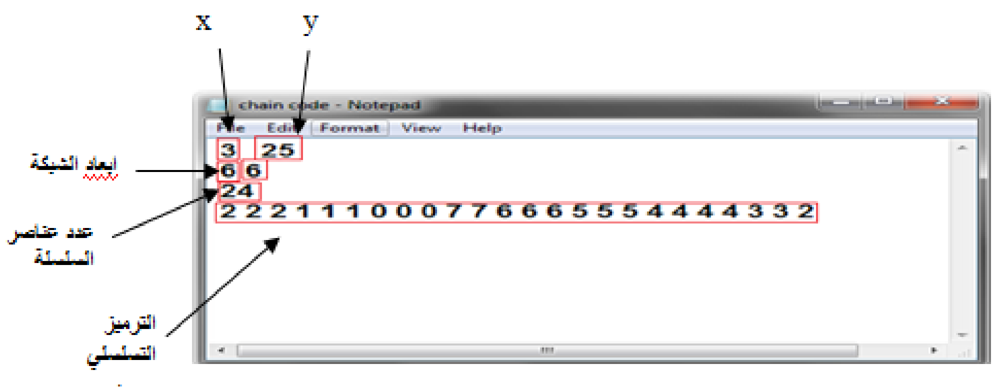
```
if ((dx==0))
  if(dy<0)    chain_cod=2;
  else         chain_cod=6;
  end
end
```

في نهاية هذه الخطوات سيتم الحصول على الترميز التسلسلي للشكل.

8.5 عملية الخزن:

في هذه المرحلة تم خزن الترميز التسلسلي في ملف يحتوي على نقطة البداية وحجم الشبكة الذي يضم المتغيرات row_dim_grid و col_dim_grid فضلا عن عدد عناصر السلسلة وعناصر الترميز التسلسلي للشكل, وكما موضح في الشكل (13):

- انظر الملحق الذي يضم الأشكال والترميز التسلسلي لها.



الشكل (13)ملف(txt.)
الذي يضم نقطة البداية ، ابعاد الشبكة ، عدد عناصر السلسلة ، والترميز التسلسلي للشكل

6. كفاءة الطريقة:

لبيان كفاءة الطريقة تمت مقارنة الطريقة المقترحة لحل مشكلة تغير الحجم مع طريقة Rehman لحل مشكلة تغير الحجم, فكانت الطريقة المقترحة أكفأ, إذ ان طريقة Rehman تغير عدد عناصر الترميز التسلسلي (طول السلسلة) بشكل ملحوظ كما في المثال في الشكل (4) اعتمادا على حجم الشكل فكلما كبر حجم

الشكل ازداد عدد عناصر الترميز التسلسلي بينما في الطريقة المقترحة في البحث فأن عدد العناصر سيكون تقريبا مساويا للشكل الواحد عند تغير حجمه. كما ان طريقة Rehman تحتاج الى حساب عدد التكرارات لكل رمز ثم حساب مجموع التكرارات ثم عملية حساب نسبة تكرار كل رمز للحصول على الترميز التسلسلي النهائي للشكل. بينما الطريقة المقترحة لا تحتاج الى عمليات حسابية للحصول على الترميز التسلسلي , وبذلك سيكون الوقت المستغرق لإيجاد الترميز التسلسلي في الطريقة المقترحة اقل بكثير من الوقت المستغرق للحصول عليه بطريقة Rehman.

7. الاستنتاجات :

1. إن تطبيق الترميز التسلسلي على الشكل الهندسي في عملية تمييز الأشكال يختصر الوقت والجهد للوصول إلى الشكل الصحيح في عملية المقارنة.
2. إن تغير حجم الشبكة لحل مشكلة تغير حجم الصورة عند تحويل الصورة إلى متجه باستخدام الترميز التسلسلي أسرع بكثير من استخدام الطرق الحسابية للباحث Rehman .
3. نلاحظ إن طريقة الترميز التسلسلي فعالة في تمييز الحدود الخارجية للشكل وبذلك يمكن الاستفادة منها في تمييز الأشكال الهندسية بكفاءة عالية.
4. إن طريقة الترميز التسلسلي تحتاج إلى معالجة أولية (عملية تنحيف) لتهيئة الحدود الخارجية للشكل.

المصادر

1. الخطيب , زينة نبيل جميل , (2011)," الوثوقية الشخصية اعتماداً على الأبعاد الهندسية لليد", رسالة ماجستير , جامعة الموصل , كلية علوم الحاسوب والرياضيات , قسم الحاسوب.
2. An, Youngeun; Lee,Jinim; Chun, Jonghun; Park, Jongan; (2011); " Frequency based on rearranged cain code"; IEEE Second International Conference on Intelligent Systems, Modeling and Simulation; PP.157-160.

3. Binary Image Analysis

www.csee.wvu.edu/~xinl/courses/ee465/binary_image_analysis.ppt; update:7/8/2011.

4. Bose, Paromita; (2000); " The Encoding and Fourier Descriptor of Arbitrary Curves In 3-Dimentional Space" , MSC Thesis, University of Florida.

5 . Boundary Description and Representation

http://morse.cs.byu.edu/60/lectures/lect07/boundary.slides.printi ; update:1/1/1970.ng.6.pdf

6. Digital Image Processing

http://telin.ugent.be/~sanja/ImageProcessingCourse/10a_Imag ; update:9/2/2007.eRepresentation.pdf

7. John, Jomy; K.V.,Pramod; Balakrishnan, Kannan; (2011); " Offline Handwritten Malayalam Character Recognition Based on Chain Code Histogram"; IEEE, Proceedings of Icetect; pp.736-741.

8. Junding, Sun; Heli ,Xu; (2009)," Contour-Shape recognition and retrieval based on chain code"; IEEE International Conference on Computational Intelligence and Security; PP.349-325.

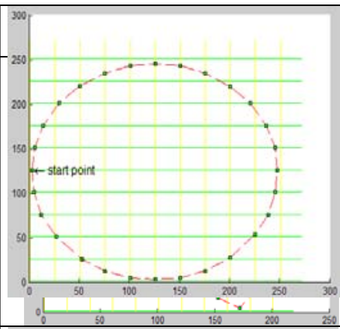
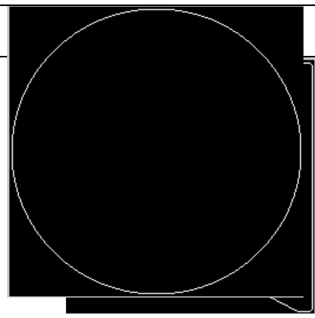
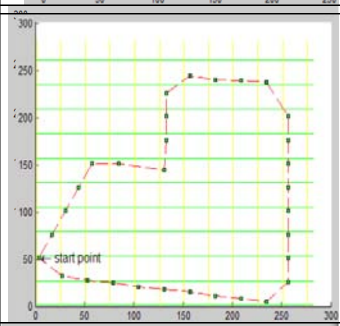
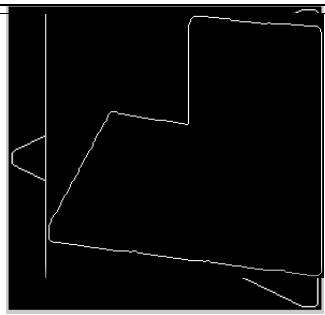
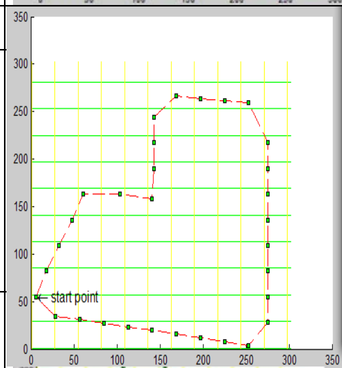
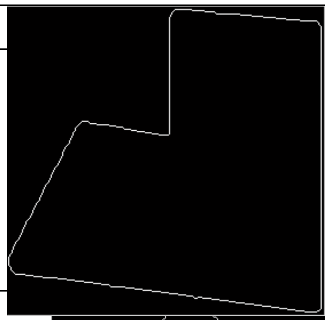
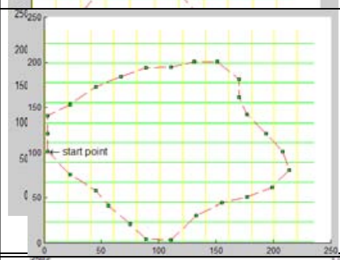
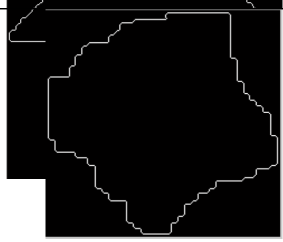
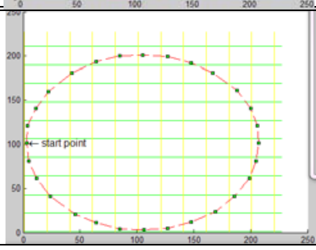
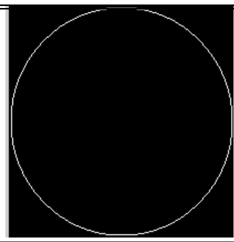
9. Lawal, Isah A.; Abdel-Aal, Radwan; Mahmoud, Sabri.A; (2010);" Recognition of Handwritten Arabic(Indian) Numerals Using Freemans Chain Code and Abductive Network Classifiers"; IEEE International Conference on Pattern Recognition; PP.1884-1887.

10. Podder, Ranjan Maity; Anjan, Bikash ; Mandal, Sandip ; " Edge Detection Using Morphological Method and Corner Detection Using Chain Code Algorithm"; IJCSI International Journal of Computer Science Issues; Vol. 8; No 1; PP.1694-0814 .

11. Raviraj, P; Angel, R.Eswin Pria; Sanavullah, M.Y; (2009); "An Efficient Approach for Shape Representation in Morphology by Combining Chain Code and Line Descriptors"; TECHNIA-

international jornal of computing science and communication Technologics; Vol.2; No. 1; PP.356-363
12. Rehman ,M.A.U; (2010);" A New Scale Invariant Optimized Chain Code for Nastaliq Character Representation"; Second International Conference On Computer Modeling and Simulation ; Vol. 4; PP.400-403.
13. Representation and Description Lecture 18 (page 1) ; http://dip.sun.ac.za/~hanno/tw444/lesings/lesing_18.pdf Update:10/10/2003.
14. Shahab, Walid; Al-Otum, Hazem; Al-Ghoul, Farouq ; (2009); " A Modified 2D Chain Code Algorithm for Object Segmentation and Contour Tracing"; The International Arab Journal of Information Technology; Vol. 6; No. 3; PP.250-257.
http://www.ccis2k.org/iajit/PDF/vol.6,no.3/7.pdf; update:7/9/2011
15. Sleit , Azzam Talal ; Jabay, Rahmeh Omar; "A Chain Code Approach for Recognizing Basic Shapes"; King Abdullah II for Information Technology College; University Of Jordan.
16. Wanitchai, Pattarapan; Phiphobmongkol, Suebskul; (2008); "TrafficWarning signs Detection and Recognition Based on Fuzzy Logic and Chain Code Analysis"; IEEE Second International Symposium on Intelligent Information Technology Application; pp.508-512.
17. Wulandhari ,Lili Ayu, Haron, Habibolah, Mohammad, Ariffin," The Mapping Algorithm of Rectangular Vertex Chain Code from Thinned Binary Image"; PP.1-9.

الملحق: أمثلة تطبيقية

2211100007776666555444433322 1111111117666666665333333333		
1111002221000766666665444444443 1111111117666666665333333333		
21000010777655666543333332 1111002221000766666665444444443		
21000010777655666543333332		
2211110007667765545543333		
2211100007776666555444433322		

22111100076677655455543333