Structural Segmentation of Line-Drawing Pictures

V.V. Matsello
Kiew

Abstract. An algorithm for extraction of characteristic parts of the object in line-drawing pictures is described. The algorithm doesn't distort the original pattern. It allows one to determine some structural relations between parts extracted, after which the structural analysis of the picture may be performed. An example of application of the algorithm for encoding of block diagrams is offered.

INTRODUCTION.

The majority of structural or linguistic methods of pattern recognition are based on the assumption that objects in the picture are represented in some specific form. It is often required to extract some characteristic components of the object (so called "primitives") and determine their relative positions to use them as terminal symbols for structural analysis algorithm.

For pictures drawing by straight lines such objects as line segments, lines intersect points or lines joint points may be chosen as characteristic components of the image.

Extraction of such primitives is not the result of some simple measurements or simple transformations effected on the original image itself (such as the measurement of the mean brightness of the individual portions of the image). It requires far-from-trivial preprocessing operations. Moreover, this preprocessing must not distort the structure or connectivity of the original pattern; as a result the analysis algorithm is feasible and effective.

At present one of the most popular preprocessing method is image skeletonization [1-3] . These algorithms are based on the iterative procedure of elimination of contour points and preservation of the final points using masks which may have different form and size. Such final points constitute the skeleton of the pattern. Unfortunately most of such algorithms have some drawbacks.

1. As a result of elimination of some points of the original image

some information about this image is lost. It leads to the following two drawbacks.

2. In the skeleton obtained the open ends of limbs get shortened, this shortening being dependent upon thickness of the limbs.

3. For patterns which consist of limbs of uniform thickness the skeleton obtained is not just composed of the straight medial lines of various limbs. More visible distortions occur at the areas of junction of the limbs and at the open ends of inclined limbs.

In this paper the algorithm of extraction of characteristic parts of objects in binary pictures with line like structures is proposed. We'll call such parts structural segments. The algorithm presents all objects in a binary picture as a set of regions which don't overlap. Most important property of the algorithm is that it does not lose any useful information about original image. That is why the algorithm does not bring any distortion of the structure or connectivity of the original picture. Moreover, the types of regions extracted and their certain structural relations, namely, adjacency of these regions in a given set of directions are determined. This information is sufficient for structural analysis of the image to be implemented.

THE ALGORITHM.

We want the algorithm to present all objects in a line-drawing picture as a set of disjoint regions - structural segments - and to define certain relations between these regions which characterize their relative positions. The segments obtained must correspond to the patterns of line segments, lines open ends or regions of lines joint or lines intersection in the original picture. For example we want the object in Fig. 1a to be devided into 15 segments as it is shown in Fig. 1b. The structural connections table (Fig. 2) must be obtained at the output of the algorithm.

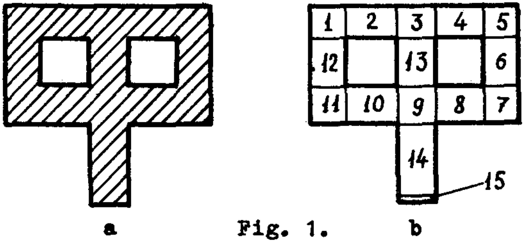

Fig. 1.

No.	Segment type	Adjacent segments				
		right	left	up	down	....
1	2	3	4	5	6	...
1	N.W. corner	-	2	-	12	
2	hor. line	1	3	-	-	
3	lines joint	2	4	-	13	
...	...	...	...	...	...	
15	bottom open end	-	-	14	-	

Fig. 2. Fragment of the structural connections table.

The algorithm of obtaining structural connections table consists of the following stages.

a). Preprocessing. At this stage the picture is encoded to provide compact storing of image data using procedure which is similar to Freeman's chain encoding tecknique. The set of brightnesses ("black" or "white") at each cell on given rectangular grid is the input information of the preprocessing. At the output image is represented by the set of closed sequences of edge line segments which are boundaries between black and white regions in the descrete picture. We'll define these sequences as contours. The concept of "edge segment" and "contour" will be thoroughly discussed later.

b). Segmentation. At this stage the image is divided into characteristic regions - structural segments. At the output of the stage each structural segment as well as the above-mentioned contour is represented by closed sequence of certain line segments. But now these segments may be both edge lines and boundary segments. The latter been able to separate one region of the object from another one. We'll call these boundary segments joints.

c). Detecting structural segments relations. At this stage for every structural segment it is neccesary to determine what structural segment is adjacent to the given one in a given direction (e.g. right, left, up, down). The result is put down into columns 1 and 3,4, ... of the structural connections table.

d). Determination of structural segments types. At this stage the type of each structural segment or type of the region of original picture which is correspondent to this segment is defined. For example on the image in Fig. 1b the type of segment No.1 is north-west corner, segment No.2 - horizontal line, segment No.9 - lines intersection, segment No.15 - bottom open end of vertical line, etc. The result is put down into column 2 of the structural connections table.

Let us consider each stage of the algorithm in detail.

PREPROCESSING.

Let the grid Q be a set of pairs (i,j), $i=1,2,\ldots,n$; $j=1,2,\ldots,m$. Pair $(i,j)\in Q$ is called a cell. The image x is the function $Q\rightarrow X$, where $X=\{\text{"black"}, \text{"white"}\}$. If n and m are large it is difficult to store image data as a set of black and white cells. Therefore the problem of compact encoding of the image arises.

We'll introduce the following concepts.

1. We'll denote by border t a pair of cells $((i,j),(i+1,j))$ or $((i,j),(i,j+1))$ on condition that one of the cells from the pair is white and the other cell is black. Furthermore it is assumed that $x(i,j)$ is white for all cells $(i,1)$ and (i,m) where $i=1,2,\ldots,n$ and for all cells $(1,j)$ and (n,j) where $j=1,2,\ldots,m$.

2. We introduce function $w(t)$ called border direction taking on values in the set $W=\{\text{right, left, up, down}\}$. For further convenience let us choose borders direction so that if one moves from the starting point of a border to its final point then object cells ("black") must be situated to the right of the border but background cells ("white") - to the left of the border.

3. If the final point of border t_1 coincides with the starting point of border t_2 then we'll say that borders t_1 and t_2 are adjacent and t_2 follows t_1 and denote this fact by $t_2=f(t_1)$.

4. Ordered sequence of borders $t_k, t_{k+1}, \ldots, t_{k+l}$ where every border satisfies the following conditions: $t_{i+1}=f(t_i)$, and $w(t_{i+1})=w(t_i)$ for all $i=k,k+1,\ldots,k+l$ will be called edge segment Z.

We associate with each edge segment Z its starting point which makes the beginning of its first border t_k (will be denoted by $BEG(Z)$), its final point which makes the end of the final border t_l (denoted by $END(Z)$), edge segment direction $w(Z)=w(t_i)$ and edge segment length $d(Z)=l$. If $END(Z_1)=BEG(Z_2)$ then we'll say that edge segment Z_2 follows Z_1 and denote this fact by $Z_2=f(Z_1)$.

5. Closed sequence of edge segments $Z_1, Z_2, \ldots, Z_p$ where $Z_{i+1}=f(Z_i)$, $i=1,\ldots,p-1$ and $Z_1=f(Z_p)$ will be called contour.

Arbitrary image can be represented by a combination of the above defined contours. To store a contour it is sufficient to store for every edge segment Z_i the value of function $f(Z_i)$ and to store either coordinates of the beginning of each edge segment, i.e. $BEG(Z_i)$, or coordinates of the first segment starting point and each segment length and direction, i.e. $d(Z_i)$ and $w(Z_i)$. For the majority of images such representation is more compact than simple storing of brightness at every grid cell.

Because of errors of digital representation contours may consist of a large number of short edge segments though real image was drown in long straight lines. In many cases such detailed information about image discrete representation is superfluous. It is useful to get rid of such superfluity. It may be done by using edge smoothing technique, for instance by means of some transformation grammar [4] The smoothing consists in replacement of some sequence of edge segments $Z_1, Z_2, \ldots, Z_r$ which is correspondent to the straight line boundary in the original picture by single segment H.

For this purpose approximation algorithm, for instance [5] , may also be used. The algorithm approximates arbitrary planar digital curve by a polygon which consists of a small number of linear segments. The polygon obtained may be represented by acontour as it has been described above.

Let contour segment number H_u coincides with its starting point number h_u. We'll denote this point by vertex h_u. If vertex h_u is both the starting point of contour segment H_u and the final point of segment H_v then we'll state that segment H_v is followed by segment H_u or that vertex h_v is followed by vertex h_u. We'll denote this fact $H_u=f(H_v)$ or $h_u=f(h_v)$.

If we know coordinates of each contour vertex we can define for every contour segment H_i its length $d(H_i)$ and its direction $w(H_i)$ from the given set of feasible directions. Contour segment direction is defined in such way. The whole plane is divided into N sectors, where N is a chosen quantity of feasible directions and $N=2^l$, where $l=2,3,\ldots,n$. Each sector is encoded by the integer number $K\in\{0,1,\ldots,N-1\}$, Then for every contour segment H_i which falls into the sector encoded as K segment direction $w(H_i)$ equals K. For instance in Fig. 3 , N=8 and for every contour segment H_i which falls into sector $-22{,}5^\circ - +22{,}5^\circ$ (shaded), $w(H_i)=0$.

SEGMENTATION.

As it was pointed before, we must divide an image represented as

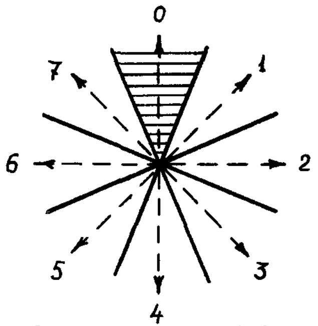

Fig. 3. An example of feasible directions choice.

a set of contours into characteristic parts - structural segmehts. We'll present each structural segment by closed sequence of straight line segments. These may be either edges or boundaries between adjacent structural segments (so called joints).

Let the image after preprocessing be represented as a set of contours which consist of line segments having one of N feasible directions. Let us consider segmentation procedure of discriminating structural segments which correspond to straight lines segments. It will be seen that structural segments of all other types will also be extracted.

Structural segments would be of "straight line" type if their contours consists of four segments which satisfy the following conditions:

1. There are two contour segments H' and H'' such that

$$|w(H')-w(H'')|=N/2 \qquad (1)$$

where w(H) is direction code of segment H and N is number of feasible directions. This requirement means that H' and H'' must have opposite directions.

2. There are two joints H^* and H^{**} such that

$$\text{a)} \quad d(H^*)\leqslant P_w \; ; \quad d(H^{**})\leqslant P_w \qquad (2)$$

where P_w - top limit of line width;

$$\text{b)} \quad |w(H^*)-w(H^{**})|=N/2$$

i.e. joints H^* and H^{**} must also have opposite directions.

3. Following relations

$$w(H_c)=(w(H_j)+N/4)\mathrm{mod}N \quad \text{or} \quad w(H_j)=(w(H_c)+N/4)\mathrm{mod}N \tag{3}$$

must be holded for every pair (H_c,H_j), where $H_c \in \{H',H''\}$, and $H_j \in \{H^*,H^{**}\}$, i.e. joints must be perpendicular to edge segments.

Let us consider algorithm operation when "horizontal straight line" structural segments are discriminated.

For the sake of convenience we introduce the following two functions. If coordinates of starting point of some line segment H_u are (i,j) we'll denote it as $I(h_u)=i$ and $J(h_u)=j$.

The algorithm consists in successive examination of horizontal contour segments and fulfillment of the common step of algorithm for every such one. In general, at every common step some new vertices h_x are defined which are either starting points or final points of a joint, and function f is changed.

The common step of the algorithm consists of two stages.

1. <u>The search of the nearest contour segment having opposite direction</u>. A recurrent horizontal contour segment H_v such that $d(H_v) \geqslant P_1$ is checked, where P_1 is minimum feasible line segment length. Let top directed segments have code K=0. Then $w(H_v) \in \{N/4,N-N/4\}$. From point h_v we construct a vertical line whose direction code is either K=0 if $w(H_v)=N-N/4$ or K=N/2 if $w(H_v)=N/4$ until it intersects the nearest horizontal contour segment H_a. Then the following conditions are checked:

a) $w(H_a)=N-w(H_v)$ i.e. H_a must have opposite direction to H_v;

b) $d(H_a) \geqslant P_1$;

c) $|I(h_v)-I(h^*)| \leqslant P_w$, where h^* is the point of intersection of contour segment H_a and the above constructed vertical line.

If contour segment H_a satisfies given conditions we must realize the second stage of the common step. Otherwise the same search procedure is implemented for point $h_u=f(h_v)$. If for h_u a sutable contour segment H_a still has not been found we must proceed to the next common step.

We have described a simplified procedure for the nearest neighbor search. In practice for N=4 the same information simultaneously with the contour detection may be obtained using one-pass algorithm. In this paper this algorithm is not considered.

2. <u>Joints construction.</u> Let at the first stage for some vertex h_v such that $f(h_v)=h_u$, the nearest contour segment H_a be found.

Let $f(h_a)=h_b$. Then

a) we introduce new vertices h_x, h'_x, h''_x which have the following coordinates:

$I(h_x)=I(h_v)$; $J(h_x)=J(h_v)$; $I(h'_x)=I(h''_x)=I(h^*)$; $J(h'_x)=J(h''_x)=J(h_v)$.

b) we change function f so that

$f(h_a)=h'_x$; $f(h_x)=h_u$; $f(h'_x)=h_x$; $f(h''_x)=h_b$; $f(h_v)=h''_x$.

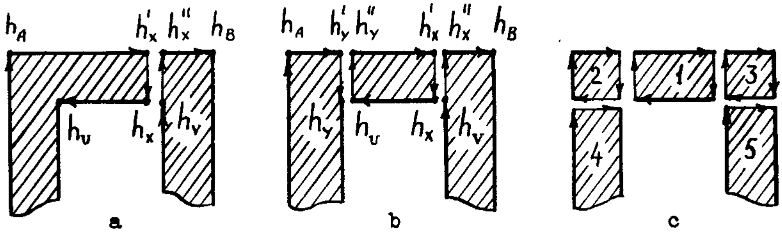

Fig. 4. The order of vertices after joints construction is shown by arrows.

The order of vertices after fulfilling algorithm common step for vertex h_v is shown in Fig. 4a by arrows. Segments which have starting points h'_x and h_v are marked as joints. Then we proceed to the next common step for vertex h_u - final point of contour segment H_v. As a result of vertex h_u processing we introduce new vertices h_y, h'_y and h''_y. The order of vertices in accordance with new function f is shown in Fig. 4b.

Having processed all horizomtal contour segments we represent regions of initual picture correspondent to horizontal straight lines as sequences which satisfy conditions (1-3). By analogy we may extract structural segments correspondent to initual straight lines segments having other N/2-1 directions. To do this it is sufficient to iterate this segmentation procedure with previous rotation of coordinate axies at the angle of $2\pi/N$ and addition 1modN to the values of function w(H).

Fig. 4c showes the result of algorithm operation in the given fragment of the image. It is seen from the figure that alongside with the structural segments of "straight line" type segments of other types have been also extracted. Thus structural segment No.2 is of "north-west corner" type while segment No.3 is of "north east corner" type.

Note that for the pictures consisting of straight lines we may choose thresholds P_w and P_l such that description of every structural segment will contain not more than one joint of each feasible direction.

DETECTING STRUCTURAL SEGMENTS RELATIONS.

As a result of segmentation initual image is represented as a set of structural segments. Each structural segment is described by close sequence of edge segments and joints. We'll call two structural segments S_1 and S_2 adjacent if their descriptions contain joints $H_1 \in S_1$ and $H_2 \in S_2$ such that

$$BEG(H_1)=END(H_2); \quad END(H_1)=BEG(H_2); \tag{4}$$

$$|w(H_1)-w(H_2)|=N/2 \tag{5}$$

If we know joints directions we may detect direction in which two structural segments are adjacent. To do this it is sufficient to subtract N/4modN from the joint direction code K. For instance in Fig. 4c structural segment No. 1 contains joint with K=4, then in direction 2 (right) it has adjacent segment No. 3.

Simple algorithm for finding segments which are adjacent to the given one in each of the feasible directions follows from the definition of adjacency. It is necessary to check all joints of the given segment and for each one find such structural segment which has a joint satisfying conditions (4-5). Then we may detect direction in which those two structural segmentsare adjacent taking into consideration the direction of the corresponding joint. As it was pointed out previously, each structural segment description may contain only one joint having certain direction. So we may construct a set of integers M which consists of N elements. The value of each element is equal to the structural segment number with the segment adjacent to the given one in the respective direction. If given segment has no adjacent one in certain direction the respective element of its set M will have value 0. For example if N=8 then set $M_1 = \{0,0,3,0,0,0,2,0\}$ is correspondent to structural segment No. 1 in Fig. 4c.

DETERMINATION OF STRUCTURAL SEGMENTS TYPES.

The type of structural segment is determined by a set of joints which the segment description suggests. For instance structural segment of "straight line" type by definition must have exactly two joints. Moreover, these two joints must have opposite directions.

"Straight line" structural segments having different inclinations are destinguished by their joints directions. Structural segments with two joints of different inclinations will be extracted at the areas of lines break. Structural segments with four joints will be extracted at the areas of lines intersection. Moreover, these joints must be perpendicular to intersecting lines. Certain directed joints in the structural segment description is referred not only nonzero elements at certain positions in its set M. It also allows one to determine the type of the structural segment using its set M.
The type of structural segment may be encoded by binary digit which has N bits. Some bits will have value 1 if they correspond to nonzero elements of set M and value 0 otherwise. For instance the type of structural segment No.1 in Fig.4c will be encoded by binary digit 00100010.

APPLICATION EXAMPLE.

Encoded data about electronic and logical scheme of the device is the input information in the process of computer-aided dezign of printed circuits. It is necessary to list all scheme elements: blocks, block contacts, points of conductors electrical junction (knots), open ends of conductors. It is also necessary to encode electrical communications between these elements.

Experiments on encoding similar block-diagrams have been carried out using structural segmentation algorithm.

Digital representation of block-diagram image is shown in Fig.5.

The same image after preprocessing and structural segmentation with N=4 is shown in Fig.6. Symbols "Ж" represents elements of edge segments and symbols " *" show joints elements.

By means of the segmentation algorithm we det the structural connections table (Fig. 7).

By vertue of structural connections information syntactic analysis of block-diagram image has been carried out using heuristic algorithm. The result of the analysis is shown in Fig. 8. Block contacts were enumerated in succession counterclockwise starting in the north-west corner of the block.

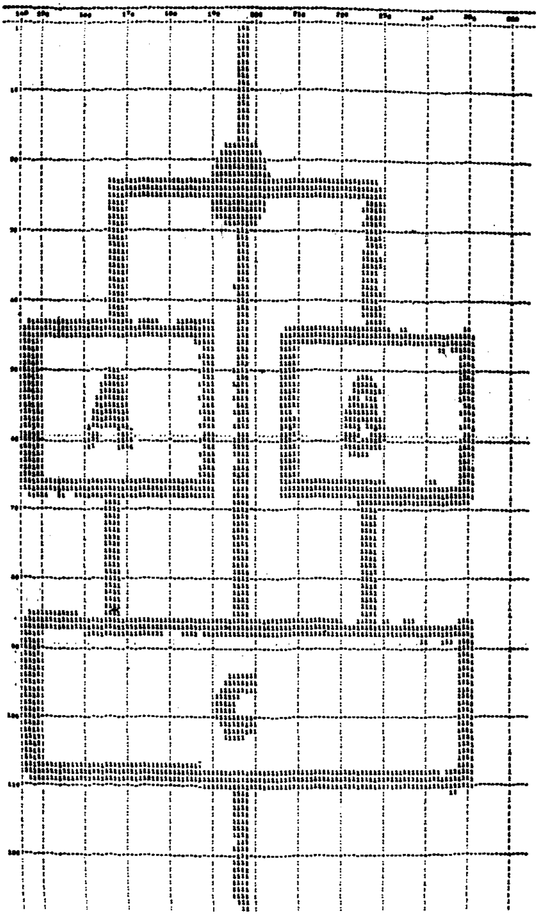

Fig. 5. Digital representation of block-diagram image.

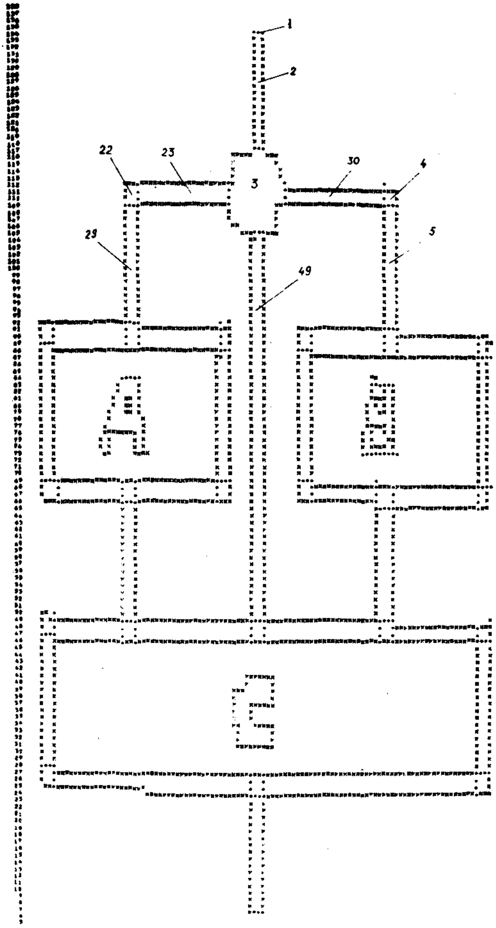

Fig. 6. Block-diagram image after preprocessing and structural segmentation.

No.	Segment type	Adjacent segments			
		right	left	up	down
1	2	3	4	5	6
1	0010	-	-	-	2
2	1010	-	-	1	3
3	1111	30	23	2	49
...	...	...	...	...	...
22	0110	23	-	-	29
23	0101	3	22	-	-
...	...	...	...	...	...

Fig. 7. Fragment of structural connections table for image in fig. 6.

```
ПРОГРАММА ОБНАРУЖИЛА
I. БЛОКОВ                3
2. КОНТАКТОВ НА БЛОКАХ   8
3. УЗЛОВ                 I
4. СВОБОДНЫХ КОНЦОВ      2
СВЯЗИ БЛОКОВ:
БЛОК СТОРОНА КОНТ.
 А    НИЖНЯЯ   I --- БЛОК С, ВЕРХН. СТОРОНА, КОНТАКТ  4
      ВЕРХН.   2 ---УЗЕЛ I
 В    НИЖНЯЯ   I --- БЛОК С, ВЕРХН. СТОРОНА, КОНТАКТ  2
      ВЕРХН.   2 --- УЗЕЛ I
 С    НИЖНЯЯ   I --- СВОБОДНЫЙ КОНЕЦ I
      ВЕРХН.   2 --- БЛОК В, НИЖНЯЯ СТОРОНА, КОНТАКТ  I
      ВЕРХН.   3 --- УЗЕЛ I
      ВЕРХН.   4 --- БЛОК А, НИЖНЯЯ СТОРОНА, КОНТАКТ  I
СВЯЗИ УЗЛОВ:
 I --- БЛОК А, ВЕРХН. СТОРОНА, КОНТАКТ  2
       БЛОК В, ВЕРХН. СТОРОНА, КОНТАКТ  2
       БЛОК С, ВЕРХН. СТОРОНА, КОНТАКТ  3
       СВОБОДНЫЙ КОНЕЦ 2
СВЯЗИ СВОБОДНЫХ КОНЦОВ:
 I --- БЛОК С, НИЖНЯЯ СТОРОНА, КОНТАКТ  I
 2 --- УЗЕЛ I
```

Fig. 8. The result of structural analysis.

REFERENCES.

1. C.J. Hildich,"Lenear Skeletons from Square Cupboards", in Machine Intelligence IV (B. Meltzer and D.Michie, Eds.), p.p.403-420.
2. R.Stefanelli and A.Rozenfeld,"Some Parallel Thining Algorithms for Digital Pictures", J.Assoc.Comput.Mach. 18,(2),1971, p.p. 255-264.
3. I.S.N.Murthy and K.J.Udupa,"A Search Algorithm for Skeletonization of Thick Patterns", C.G.I.P.,vol.3,No.3,1974,p.p.247-261.
4. В.В.Мацелло,"Выделение длинных горизонтальных и вертикальных границ на изображении", в кн.:Математические и технические средства робототехники и распознавания образов", Киев: ИК АН УССР I98I,стр.48-57.
5. U.Ramer,"An Iterative Procedure for the Polygonal Approximation of Plane Curves", C.G.I.P.,vol.1,No.3,1972,p.p.244-256.

V.V. Matsello
Institute of Cybernetics
Academy of Sciences of the
Ukrainian SSR
Kiew, 252207