Probability Estimation for People Detection

Qingjie Sun
Key Lab. of Computer Science
Institute of Software, CAS
Beijing 100080, China
sqj@ios.ac.cn

Enhua Wu
Faculty of Science and Technology
University of Macau, Macau
fstehw@umac.mo

ABSTRACT

This paper presents a probability method for detecting people in a static image. We simplify people as a torso and four limbs. The torso is fitted by a quadrangle, and each limb is fitted by one or two quadrangles depending on its pose. In order to find people, we should try to find a combination of quadrangles that must satisfy some geometric and topological constrains. Firstly, we try to detect and fit rectangle regions in the image, then we search some combinations of rectangles under certain constrains. After we get a combination, we calculate its probability of being people. If the probability is above threshold, we adjust the vertexes of rectangles so as to get a compact people model.

Keywords

people detection, image segmentation ,rectangle fitting, probability estimation

1. INTRODUCTION

Object detection is a challenging research topic. It needs us to present a good descriptor for the object class to separate it from other object class. In this paper, we mainly discuss about people detection, because it has many applications, such as surveillance system, driver-aided system, image-based human modeling and simulation.

Many factors should be taken into account for people detection［Anu01a］. People are articulate objects, so there may be a variety of shapes. In addition, people can dress in many kinds of clothes that can vary in color and texture. For people detection, there is high intraclass variability, so it is difficult to deal with.

Many systems use motion information, but they are really tracking people instead of detecting people［Har98a, Wre97a］. In this paper, we discuss people detection in a static image.

The whole shape of people can have many variations, but the variation for each of its component is smaller.

In this paper, we only consider torso and four limbs, and we try to find the rectangles corresponding to these components. After that, we start to search some combinations of these rectangles under certain geometric and topological constrains. If we find such combination, and its probability for being people is above a fixed threshold, we can say we have detected people in the image.

We will briefly review some related work in section 2. Section 3 introduces a method for image segmentation. Region analysis and rectangle fitting are explained in section 4. We present a probability function for human detection in section 5. Experimental results are demonstrated in section 6. We conclude with some discussions in section 7.

Journal of WSCG, Vol.11, No.1., ISSN 1213-6972
WSCG'2003, February 3-7, 2003, Plzen, Czech Republic.

2. PREVIOUS WORK

As the Internet has developed quickly in recent years, many researchers start to study content-based search. The people information in a color image is important, so people detection in a static image becomes a research focus.

Papageougious et al. have introduced example-based learning techniques to detect people in a static image [Ore97a]· They use Harr wavelets to represent the images and use Support Vector Machine（SVM） classifier to recognize the patterns. The system gets further improvement in [Anu01a] to make it be able to detect occluded people or people whose body parts have little contrast with the background. The system needs many transformations and match with wavelet template, and it mainly deals with pedestrian detection.

David A.Forsyth et al. present some methods to find naked people in a static image [For96a, For97a, For99a, Iof01a]. The skin of human has some properties in hue and saturation. They segment the regions with these properties, and detect edges in each region. They get rectangles by combining the edges based on local symmetry. At last, they use a complex probability function to decide which combination of rectangles can be seen as people.

D.H. Hyeon et al. detect edges in image by Laplacian operator, and they calculate curvature information for edge map [Hye01a]. They use the curvature information to detect people. The algorithm can be used for gray and color images, but it can only detect head and shoulder of people.

In this paper, we present a simple probability function to detect people. We assume the torso and four limbs are not occluded by other objects, and they are not overlapped each other. Firstly, we segment the image into regions by color and texture. For each region, we use mathematical morphology to detect and fit rectangles. If two rectangles are nearby and their union is still approximately rectangular, we combine them and get a new rectangle. We search combinations of rectangles under some geometric and topological constrains among all the rectangles. If the probability of a combination of rectangles is more than the threshold, we get a human body. We adjust the vertexes of rectangles to get a compact model, and recalculate the probability for this compact model. At last, we output the body model.

3. IMAGE SEGMENTATION

In order to detect people, we should detect the image regions of torso and four limbs at the first step. In most case, each part corresponds to one region, so we adopt region-based image segmentation. Clothes can vary in color and texture, so we must take color and texture into account at the same time. We adopt the JSEG algorithm [Yin01a]. It separates the segmentation process into two stages: color quantization and spatial segmentation. We can only consider color in the first step, and only consider texture in the second step. It can simplify the segmentation very much.

For a 24-bit color images, it can contain many colors. If we adopt the right method of color quantization, the number of colors can be reduced to 10-20 without degrading the image quality seriously.

After color quantization, we substitute each pixel's color with its class label. As a result, we get a special class map of texture composition. In the window centered by one pixel of class map, there are only a few class labels. The spatial distribution of class labels in the window can represent the texture property of corresponding pixel. We can define a J value according to the spatial distribution. If the J value is bigger, the color classes in the window are more separated, and the pixel is near the edge. If the color classes are distributed uniformly over the window, the J value is smaller, and the pixel is far from edge. We can construct a new image named J-image, and each pixel's value is the J value at the same position. If we see J-image as a 3D terrain map, the valleys and hills represent the region interiors and region boundaries respectively [Yin01a].

The J-image can be segmented by region-growing method. The oversegmented regions can be merged based on their color similarity. Details about J-image and segmentation can be seen in [Yin01a].

4. RECTANGLE FITTING

After image segmentation, we analyze the region bounded by the exterior boundary of original region by mathematical morphology. If the boundary of region consists of a long image boundary, we think the region as background and ignore it.

4.1 Distance Map

We use mathematical morphology to analyze the shape of image region. We wish the result can represent important structure [Mar97a].

3	4	5
2	i, j	6
1	0	7

Fig.1 Pixel neighbors

Let I be a binary image. The neighbors of pixel (i,j) are $(m,n)=(i+x, j+y)$, (x, y may be 0,1 or –1). If $|m-i|+|n-j|=1$,we call them direct neighbor. If $|m-i|+|n-j|=2$, we call them indirect neighbor. In Fig.1, 0,2,4 and 6 are direct neighbors of pixel (i,j), and 1,3,5,7 are indirect neighbors.

Distance map $DM(i, j)$ of image I is defined as :

$If\ I(i,j)==0,\ DM(i,j)=0$

$If\ I(i,j)!=0$

$$DM(i,j)=\min\{DM(i+x,j+y)+d[|x-y|]\} \quad (1)$$

$Where\ x,y=0,1,-1$

$d[0]=3, d[2]=3, d[1]=2$

We define the direction of distance map for each pixel that can tell where the pixel gets the minimum distance. Following the direction of distance map, we can get the shortest path from the pixel to the region boundary. We represent the direction of distance map at pixel *(i,j)* as *DMD(i, j)*. It can be calculated as: if the minimum distance comes from its *k*th neighbor pixel, set the *k*th bit of *DMD(i,j)* to 1,otherwise 0.

We define the local max point of *DM(i,j)* according to *DMD(i,j)*. We use N_m to represent *m*th neighbor of pixel *(i,j)*, and use $B(DMD(N_m),g)$ to represent *gth* bit of $DMD(N_m)$. If pixel *(i,j)* satisfy the following condition, we say it is a local max point:

$$\forall m \in [0,7], B(DMD(Nm), U(m)) = 0 \quad (2)$$

$$U(m) = (m + 4) \bmod 8 \quad (3)$$

4.2 Axis Shape Segment

For a complex shape, all the local max points may be divided into some connected component. There may be several branches for each connected component. We start from the left and down end to track. Once we meet the branch point, we stop tracking. The collected pixels during each track are called branch axis shape points set (BASPS). Keep tracking until we get all BASPS. Each BASPS is fitted by line segments: we decide the left and down point as one vertex of line segment, and right and up point as the other vertex. For any other point, we decide its distance to the line segment. If the distance is more than threshold, we cut the line segment at this point. This process is run recursively until all points are organized to some line segments. We call these line segments as axis shape segment (ASS). If adjacent ASS lie in the same line, we combine them to avoid oversegmented at the branch point. For each pixel of the ASS, we detect whether its corresponding region length has discontinuous change. When discontinuous change happens, we cut the line segment into two parts at that pixel. Each ASS should correspond to a region.

4.3 Rectangle Fitting

When we analyze a region, we fill it with foreground color f, and fill the background with the color b. From each vertex of ASS, we draw a line vertical to the ASS with color z. These lines are named potential cut line (PCL).

We need to decide whether the region corresponding to each ASS can be fitted by a rectangle. For each pixel on the ASS, we emit a line vertical to ASS, and calculate two crossing points with the region boundary. The distance between these two points is named the pixel's region length. We represent the *i*th pixel's region length as lineLeni.

$$rectLength = \frac{1}{n}\sum_i lineLeni \quad (4)$$

$$difLen = \frac{1}{n}\sum_i |rectLength - lineLeni| \quad (5)$$

$$difRatio = \frac{difLen}{rectLength} \quad (6)$$

If difRatio is less than threshold *maxLenLimit*, the region can be fitted by a rectangle. Otherwise, the region can't be fitted by rectangle.

For an ASS, if its region can be fitted by a rectangle, we try to extend it at each vertex with the direction far from the other end. When extending to a pixel, we emit a line vertical to the ASS, then we will find a pair of points whose distance most approaches rectLength between all the crossing points of the line with region boundary and the cut lines. This distance is the pixel's region length. If the region length occurs discontinuous change, we stop extending. If *difRatio*is is above threshold *maxLenLimit*, go back until *difRatio* is less than *maxLenLimit.* If part of the rectangle is outside the image boundary, we can ignore the rectangle.

4.4 Rectangle Combining

Rectangle fitting is done for each segmented image region. As we know, the clothes can be segmented into more than one part, so we should consider the combination of adjacent rectangles. For two adjacent regions fitted by different region, if their two pairs of edge directions are almost parallel, the distance of these two rectangles is less than threshold, and their union is an approximate rectangle, we combine them to get a new rectangle.

5. PEOPLE DETECTION

The people model we adopt consists of one torso and four limbs. When a limb is straight, we fit it with one rectangle. Otherwise we fit it with two rectangles corresponding to the two half limbs. The shape of torso and limb should satisfy some constrains. There are some geometric and topological constrains between the torso and four limbs.

5.1 Probability of People

When we look people in the image, the most important factor is the whole structure of people. Because of different eye direction and people pose, the shape of torso and limbs can vary. However, it can't change the decision result. As a result, we should design a probability function that can emphasize the total structure of people.

Suppose rectangle T is a possible torso, and the four limbs found nearby its four vertexes are A, B, C and D. Please see fig.2. The vertexes of T adjacent to rectangle A and B respectively form the width edge of T, and the ones corresponding to rectangle B and C form the length edge of T. The probability of this combination to be people is the following equation.

$$p(human) = pTorso * pLimb * pFit \quad (7)$$

$$pTorso = p(T = torso) \quad (8)$$

$$pLimb = \min\{ p(x = \lim b \mid T = torso)\} \quad (9)$$

where $(x = A, B, C, D)$

$$pFit = (fitDegree(A, B) + fitDegree(C, D)) / 2 \quad (10)$$

p(T=torso) is the probability for rectangle T to be torso. *P（A=Limb|T=Torso）* is the probability for T and A to be the torso-limb relation. *pLimb* is the minimum probability for T and A,B,C,D to be torso-limb relation. *fitDegree(A,B)* is the affect factor of the similarity between rectangle A and B to the total probability. *pFit* is the average affect factor of two pairs of limbs to the total probability. *p(T=torso)* and *p(x=Limb|T=Torso)* will be discussed in next section.

Let *length(A)* represent the length of rectangle A，*width(A)* represent the width of rectangle A.

$$lenDif = | length(A) - length(B) | \quad (11)$$

$$mLength = \max(length(A), length(B)) \quad (12)$$

$$lenScale = lenDif / mLength \quad (13)$$

By the same method, we can get widthScale.

$$mDif = \max(lenScale, widthScale) \quad (14)$$

$$fitDegree(A, B) = 1.0 - mDif * factor \quad (15)$$

factor is a weight less than 1.0, and it can represent the importance of match degree of limb pair to the total structure.

5.2 Learning Probability Functions From Examples

Different people have different body shape, and the clothes can vary very much, so the limb shape can change a lot, which means the ratio of limb's length divided by limb's width can vary in a big range. As a result, it is not necessary to distinguish legs and arms. We use the same function for leg and arm. In addition, because of different pose and eye direction, the ratio of limb's length (width) divided by torso's length (width) can also vary in a certain range.

We use 200 pictures as examples to get the constrain relations. Data needed to learn from examples includes: lAspect, tAspect, lenRatio, widRatio. lAspect is the ratio of limb's length divided by limb's width. tAspect is the ratio of torso's width divided by torso's length. lenRatio is the ratio of limb's length divided by torso's length. widRatio is the ratio of limb's width divided by torso's width. The histograms are illustrated in fig.3. The horizontal axis represents lAspect, tAspect, lenRatio, widRatio respectively, and the vertical axis represents percentage.

When a limb is straight, we fit it with one rectangle, as illustrated by limb A in fig.2. Otherwise we fit it

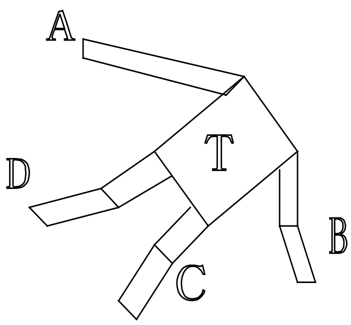

Fig2. Human model

with two rectangles corresponding to the two half limbs, then the length of limb is the sum of the two rectangles' length and the width of limb is the average width of the rectangles, as illustrated by limb B in fig.2.

We define the probability functions according to these histograms. Fig.3a (left up fig.) is the histogram of lAspect. We can see that lAspect has an interval named [a0, a3], but it mainly distributes in a interval named [a1, a2]. For the same limb, human pose and eye direction can affect the value of lAspect. When we look at people, we mainly concern on the total structure of people, and the component can change a lot. As a result, we do not need to design very complex probability function for the component. The probability function of lAspect is defined as follows:

$$pla(x = a) = \begin{cases} 0.0 & a < a0 \,\|\, a > a3 \\ y1 & a0 \le a < a1 \\ 1.0 & a1 \le a \le a2 \\ y2 & a2 < a \le a3 \end{cases} \quad (16)$$

$$y1 = p0 + (a - a0) * (1 - p0) / (a1 - a0) \quad (17)$$

$$y2 = p0 + (a3 - a) * (1 - p0) / (a3 - a2) \quad (18)$$

The value of p0 is defined beforehand, and it depends on the acceptance degree for various body shapes.

Fig.3b (right up fig.), Fig.3c (left down fig.) and Fig.3d (right down fig.) are the histograms of tAspect, lenRatio and widRatio respectively. We use similar method as lAspect to define their probability function named *pta(x=a)*, *plr(x=a)* and *pwr(x=a)* respectively.

From the examples, we know the function *plr(x=a)* and *pwr(x=a)* are independent.

We can notice that there will be discontinuous change at the a0 and a3 when using such functions. It may lead to some errors. However, we can see from examples that the probability for an example to occur nearby a0 or a3 is very small, so the probability for error is very small also.

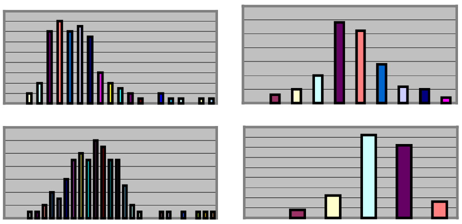

Fig.3 Histograms for people

Let the length of rectangle T is *lt*, and width of T is *wt.* We can arrive the following equations:

$$p(T = torso) = pta(wt / lt) \qquad (19)$$

Let the length of rectangle x is *la*, and width of x is *wa*. then:

$$p(x = \lim b \mid T = torso) = plr(ls) * pwr(ws) \qquad (20)$$

$$ls = la / lt \qquad (21)$$

$$ws = wa / wt \qquad (22)$$

After we get these two functions, we can calculate the probability of rectangle combinations to be people by equation 7.

5.3 Search Combination of Rectangles

People is composed of one torso and four limbs, so people detection is to find the right combination of rectangles under constrains of geometric and topological requirements. If the probability of a rectangle combination is more than threshold, we think it is people.

For complex image, we can get a lot of rectangles. If we check every rectangle using the same step, it will take too much time. We divide the search into two steps: search the torso and then search the limbs for each torso.

5.3.1 Detect Torso Rectangles

For a rectangle T, if p*(T=torso)*>0，T is a possible torso. People detection is very difficult and error prone if the size of people region is too small. We assume that the people region should not be very small. We can set the minimum length and minimum width of torso rectangle. By doing so, we can remove many unnecessary search and detection.

5.3.2 Detect Limb Rectangles

For each rectangle got by section 5.3.1, we need to check whether it is a real torso. If there is a limb around each vertex of a rectangle, and the probability of them is more than threshold, the rectangle is a real torso.

In fig.2, suppose the vertexes of rectangle T adjacent to rectangle A, B, C, D are vert0, vert1, vert2 and vert3 respectively. In order to simplify the description, we introduce several definitions.

Definition 1:For any rectangle X (different from T), if its lAspetc is less than a3，and the distance between one of its vertex named x0 and vert0 is less than threshold, we say X is the potential first half limb of T for vert0.

Definition 2:For any rectangle X (different from T), if its lAspetc is less than a3，but lAspect is more than a0, and the distance between one of its vertex named x0 and vert0 is less than threshold, we say X is the potential limb of T for vert0.

Definition 3:Let X be the potential first half limb of T for vert0, and Y be another rectangle. If Y's lAspect is less than a3, and the distance between one of its vertex named x0 and one vertex of X's *i*th edge is less than threshold, and Y's width is not more than X's width, and Y's length is more than half of X's length, we say that Y is the potential second half limb of X's *i*th edge.

Suppose X is the potential first half limb of T for vert0, and the width edge of X far from vert0 is j. If rectangle Y is the potential second half limb of X's *j*th edge, X and Y becomes the potential limb of T for vert0. The length of this limb is the sum of X's length and Y's length, and the width of this limb is the average of X's width and Y's width.

Searching all limbs of T for vert0 can be divided into two steps:

1. Search all the potential first half limbs of T for vert0, and save them into halfLimb0[].Decide the width edge far from vert0 for each rectangle in halfLimb0[], and save them in halfEdge0[].
2. For any X of halfLimbs0[], let j be its corresponding index in halfEdge0[], search each potential second half limb of X'*j*th edge which is named by Y. Calculate the lAspect of X and Y. If this Aspect is less than a3 and more than a0, we add the index of X and Y into completeLimbs0[]. At the same time, we add the length and width into limbLength0[] and limbWidth0[] respectively. If the lAspect of X is more than a0, itself also may be a potential limb, so, we add X and –1 (we set value of Y to -1 to show this limb only consist of one rectangle) to completeLimbs0[], and we add the length (width) of X to limbLength0[] (limbWidth0[]).

With the same method, we can get all the potential limbs of torso T for vert1, vert2 and vert3, and get the length and width information of each potential limb.

If rectangle T has only zero or one vertex that can find potential limb, T is not a torso. If T has only two adjacent vertexes that can find potential limb (such as vert0 and vert1), we should consider all the rectangles near T's vert2 or vert3. For a rectangle X, if one of its edge is adjacent to T's edge composed by vert2 and vert3, and the difference for length is less than threshold lenDisLimit, and the angle between these two edges is less than threshold angleLimit, we can try to combine X and T, that is to say we extend T to a new T. We then decide whether the two new vertexes have potential limbs. If T has vertexes that have potential rectangles that are not adjacent to each other, T is not a torso yet. If T has three vertexes that have potential limbs (such as vert0, vert1 and vert2), we should extend T along edge vert0vert3 and edge vert2vert3 respectively. If all vertexes of the extended T have potential limbs, we update T with extended T.

In some cases, there are some extra rectangles because of image segmentation and rectangle fitting. As a result, there may be more than one potential limb, so we should select the best potential limb. We emphasize the total structure of body model, so we should consider the match degree of limbs that lie in the same width edge of torso rectangle when we select the best limb.

Assume that rectangle T has m potential limbs for vert0 (lbs0[m]), and n potential limbs for vert1 (lbs1[n]). We use the following equations to select the best limb pair:

$$(i0, j0) = \max(lbsProb(i,j)) \tag{23}$$

where $0 \le i < m, 0 \le j < n$

$$lbsProb(i, j) = mLimb * JP * fit \tag{24}$$

$$mLimb = \min(aLim(i), bLim(j)) \tag{25}$$

$$aLim(i) = pla(x = s0[i]) \tag{26}$$

$$bLim(j) = pla(x = s1[j]) \tag{27}$$

$$JP = pi * pj \tag{28}$$

$$pi = p(lbs0[i] = \lim b \mid T = torso) \tag{29}$$

$$pj = p(lbs1[j] = \lim b \mid T = torso) \tag{30}$$

$$fit = fitDegree(lbs0[i], lbs1[j]) \tag{31}$$

where *lbsProb(i,j)* represents the probability of limb pair composed by *i*th potential limb for vert0 and *j*th potential limb for vert1. s0[i] represents the lAspect of *i*th limb for vert0，s1[j] represents the lAspect of *j*th limb for vert1.

In addition, when we define the probability of limb pair, we can consider other factors. For example, if two limbs have the similar color distributions, the probability that they belong to the limb pair increased. The relative position between the limb pair and vertexes of T (vert0 and vert1) can also affect the probability.

By similar method, we can select the best limb pair for vert2 and vert3.

To avoid making the problem too complicated, we do not consider the relation between vert0 and vert3, or vert1 and vert2.

5.4 Adjust the Vertexes of Rectangle

After finding limb for each vertex of torso, we calculate the probability for the rectangle combination by equation 7. If the probability is more than threshold, we detect people. However, because of image segmentation and rectangle fitting, the rectangles do not connect each other. We should adjust the vertexes of rectangles to get a compact model. If a limb is not straight, we connect the two half limbs of it. For each limb, we connect it to the torso rectangle.

5.5 Update Probability

After we adjust the vertexes of rectangles, we get a compact body model. Then, we should use equation 7 to recalculate the probability for the combination to be people.

6. EXPERIMENTS

We tested for sixty pictures, and successfully detected forty-one. The main detection failure is that we can't get all necessary body regions when segmenting the image. We find that we can't separate skin color from gray color very well when they are adjacent. In addition, the rectangle fitting based on mathematical morphology is prone to noise. We should improve these two problems in the future work.

Fig.4 to Fig.6 are three examples. We take the p0 of pla() as 0.8. Fig.4a is input image, Fig.4b is the segmented regions, and Fig.4c is the fitted rectangles including combined rectangles. After combination, if the length and width of rectangle are both less then threshold, we think it is too small, and delete it. Fig.4d is the detected body model. Fig.5a and fig.6a are other input images, and Fig.5b and fig.6b are detected body models.

We run the program on a PC with Celeron 952MHz, 256M RAM, and the operating system is windows 2000 (Chinese version). In the table 1, we present the size, the run time and the probability of being people for each image.

The run time is determined by the count of rectangles detected in the image, which are dependent on the complexity of image content and the image size.

Image	Size	Run-time	Probability
Fig.4	160*204	82s	0.91
Fig.5	200*268	39s	0.94
Fig.6	200*266	51s	0.78

Table 1. Experiment results

We got the images by searching on web. In our image database, the pose does not change dramatically. In the future, we should continue to collect images containing people with all kinds of pose.

When people are in some special poses, one or more limbs may change its shape dramatically, and it may lead to error.

7. DISCUSSION

In this paper, we present a method for people detection in static image. Firstly, we segment the image into regions by JSEG, and analyze each region to detect and fit rectangles. In all rectangles, we try to find the right combination satisfying some geometric and topological constrains. We present a simple probability function to calculate the probability of the combination to be people. If the probability is more than the threshold, we think it is people. At last, we adjust the vertexes of the rectangles to get a compact model.

The body model we adopted consists of one torso and four limbs. We require all the limbs and torso must appear in the image. However, many images we got from web do not satisfy this requirement. Occlusions or self-occlusions are common. So we must focus on studying some new principles to deal with the occlusions and self-occlusions. Our body model only consists of torso and limbs, and it ignores other parts, such as face, hand and foot. In the future, we should improve the body model. In addition, image segmentation is a classic difficult problem, and it may not be solved perfectly in the near future, so we need to study other detection methods that do not depend on image segmentation. In most cases, people are dressed in loose clothes, and we can’t use rectangle to fit the component, so we must consider other models to represent and detect people in the future work.

Acknowledgement

The authors would like to thank for the financial support given by the National Nature Science Foundation of China under Grant No.60033010.

REFERENCE

[Anu01a] Anuj Mohan, Constantine Papageorgiou and Tomaso Paggio. Example-based object detection in images by components. IEEE transactions on Pattern Analysis and Machine Intelligence, 23(4):349-361, 2001.

[For96a] D.Forsyth and M.Fleck. Finding naked people. In: Proceedings of European Conference of Computer Vision. Berlin, Germany: Springer-Verlag, 593-602, 1996.

[For97a] D.Forsyth and M.Fleck. Body plans. In: Proceedings of IEEE Conf. Computer Vision and Pattern Recognition. Puerto Rico: IEEE Press, 678-683, 1997.

[For99a] Forsyth,D.A. and Fleck,M.M. Automatic detection of human nudes. International Journal of Computer Vision, 32(1): 63-77,1999.

[Har98a] I.Haritaoglu, D.Harwood, and L.Davis. W4: who? when? where? what? a real time system for detecting and tracking people. In: Proceedings of International Conference on Automatic Face and Gesture Recognition. Nara, Japan: IEEE Press, 222-227, 1998.

[Hye01a] D.H. Hyeon, T.E. Kim, K.H. Yoon, J. S. Choi. People detection in a static image using curvature. In: Proceedings of International Conference on Augmented, Virtual Environments and Three Dimensional Imaging. Ornos, Mykonos , 363-366 , 2001.

[Iof01a] S.Ioffe and D.A.Forsyth. Probabilistic methods for finding people. International Journal of Computer Vision. 43(1): 45-68,2001.

[Mar97a] Marryann Simmons. 2D object abstraction: the automatic generation of hierarchical geometric representations [Master’s thesis]. University of California, Berkeley, 1997.

[Ore97a] M.Oren, C.Papageorgious, P.Sinha,E.Osuna, and T.Poggio. Pedestrian detection using wavelet templates. In: Proceedings of IEEE Conf. Computer Vision and Pattern Recognition. Puerto Rico: IEEE Press, 193-199, 1997.

[Wre97a] Wren，C., Azabayejani,A., Darrell,T.and Pentland,A. Pfinder:real-time tracking of the human body. IEEE transactions on Pattern Analysis and Machine Intelligence,19(7): 780-785, 1997.

[Yin01a]Yining Deng, B.S.Manjunath. Unsupervised segmentation of color-texture regions in images and video. IEEE Transactions on Pattern Analysis and Machine Intelligence, 23(8): 800-810, 2001.

Fig. 4a Image

Fig.4b Regions

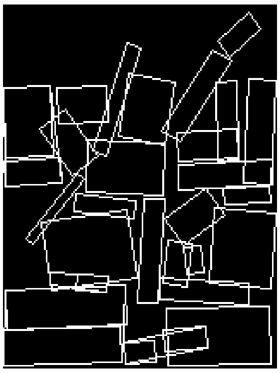

Fig. 4c Rectangles

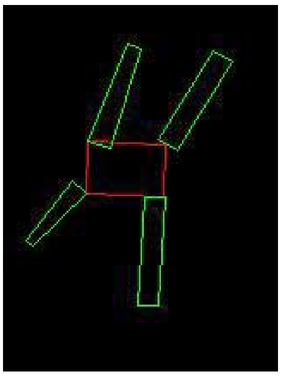

Fig. 4d People

Fig. 5a Image

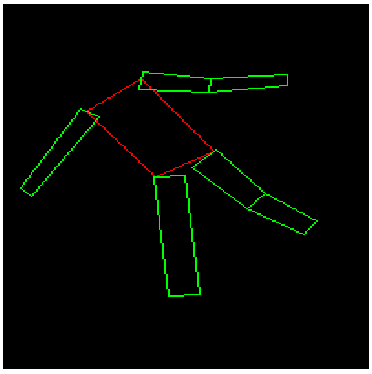

Fig. 5b People

Fig. 6a Image

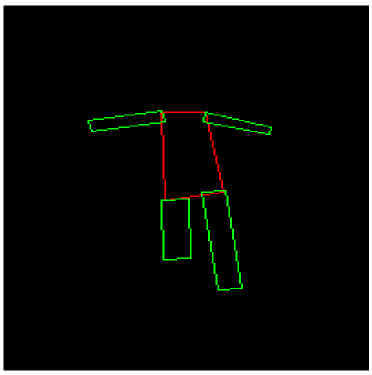

Fig.6b People