Statistics of 3D object locations in images*

Rafael Baur  Alexei Efros  Martial Hebert

CMU-RI-TR-08-43

October 2008

Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

© Carnegie Mellon University

* This work was supported in part by NSF Exploratory Grant IIS0745636
Abstract

The structural design of buildings and streets in man-made outdoor environments determine the possible locations of other objects like cars or people. Cars drive or park on the streets, and people walk on the sidewalks. Their space of interaction is limited and aligned to these man-made structures. In this report, we describe how 3D locations of objects can be estimated in a single view image. We gather statistics to analyze the locations of cars and people and show how their alignment to man-made structures creates regularities, which can be used to predict their locations. We show which prior knowledge of the analyzed scene is necessary for different ways of statistical prediction of object locations and evaluate the prediction performance on a data set with urban outdoor scenes.
## Contents

1 Introduction  

2 Estimation of object locations  
   2.1 Camera Model  
   2.2 Mathematical derivation of object location equations  
   2.3 Estimation of focal length  
      2.3.1 Focal length, field of view and angle of view  
      2.3.2 Focal length in pixel  
   2.4 Top-down view  
   2.5 Accuracy of estimations  
      2.5.1 Horizontal object location  
      2.5.2 Object distance in depth  

3 Initial analysis of object locations  
   3.1 Our data set  
   3.2 Absolute object locations  
      3.2.1 Interpretation of results  
   3.3 Object locations relative to other objects  
      3.3.1 Interpretation of results  

4 The importance of a reference direction  
   4.1 Getting a better understanding of the location histograms  
   4.2 Scene direction as reference direction  
      4.2.1 Labeling the vanishing point of the principal street  
      4.2.2 From vanishing point to scene direction invariant coordinates  

5 Gathering statistics  
   5.1 Statistics of relative object locations  
   5.2 Statistics of object distances to building walls  
      5.2.1 Filtering of the object distances  

6 Interpretation and evaluation of the statistics  
   6.1 Visualize the likelihoods of object locations in images  
      6.1.1 Computation of probability density functions  
      6.1.2 Back-projection of locations likelihoods into 2D image plane  
   6.2 Compare location prior with uniform distribution  

7 Automatic detection of scene direction  
   7.1 Video Compass  
   7.2 Data-driven vanishing point recovery  
   7.3 Vote for scene direction  
   7.4 Estimate scene direction with robust line fit  
   7.5 Compare video compass, voting and line fit  
      7.5.1 Summary  

III
## Results

8.1 Prediction of object locations based on statistical knowledge 39
  8.1.1 Image categories for which the statistical knowledge is helpful 40

8.2 Object location prediction in images with automatically detected scene direction 41

8.3 Object location prediction without reference direction 43

## Conclusion

9.1 Review 47

9.2 Outlook 48
1 Introduction

Humans are able to understand images within milliseconds. Objects can be recognized and classified, even when they are far away and represented by not more than a couple of pixels. In Fig. 1 we assume that the objects in the blue box are bicycles, because we see other bicycles in the image. Seeing the pixels in the blue box independently, we are not able to identify any objects. We are using the context of the image to identify objects. This stands in contrast to standard approaches to object detection [12, 21], which only take local pixel information into account. Recently, new approaches were proposed which try to analyze the image as a whole, to be able to predict what kind of objects can be expected and in what size and vertical location in the image [24, 17]. Others classify image regions by incorporating neighborhood regions [19]. What all these approaches have in common, is that they incorporate context by analyzing the 2D image plane. Hoiem et al. retrieve a camera model from a single view image to estimate the 3D height of object candidates and incorporate the knowledge about the 3D height of an object candidate in the detection process [11].

Figure 1: We assume that the objects in the blue box are bicycles because we see other bicycles in the image. Without the context, we are not able to identify any objects in the blue box.

We will use the same camera model to estimate 3D object locations. Given the fact that objects are limited to certain locations in man-made scenes, we want to analyze if there are any regularities in object locations. Cars drive and park on streets; their location is therefore not completely independent but bound to the location of streets where other cars might be located as well. People often occur in groups and are also more likely to be found in certain parts of an urban environment, for example the pavement. We will investigate if these arrangements of objects at certain parts of the scene leads to regularities in object locations relative to other objects. We will use a set of images
with urban outdoor scenes and gather statistics of the 3D locations of all objects in the
data set. With the statistics, we will find out where increased likelihoods for objects
are and what kind of prior knowledge about the analyzed scene is necessary. We will
answer the question: Given the location of one object, how much can we say about the
location of other objects? We will also gather statistics of object distances to building
walls and examine if any regularities exist. Probabilities for object locations can be
calculated with the statistical knowledge. These probabilities are used for the predic-
tion of objects. We will evaluate the prediction performance and discuss under which
circumstances the prediction is reliable.

We start in Chapter 2 with an explanation of how 3D object locations can be esti-
mated from a single view image. We will also discuss if the estimations are accurate
enough to be used for statistical analysis. In Chapter 3, some initial experiments will
be done to find out more about 3D object locations. In Chapter 4, we will find out
that it is important to know the orientation of the camera with respect to the scene,
and will show how it is used for the analysis of 3D object locations. Statistics of 3D
object locations relative to other objects and statistics of object distances to building
walls will be gathered in Chapter 5. These statistics will be interpreted and evaluated
in Chapter 6. More attention will be paid to the orientation of the camera relative to
the scene and its automatic detection in Chapter 7. We will introduce several methods
for the automatic detection of this orientation and compare the results. Finally, we will
present and discuss the results in Chapter 8 and come to a conclusion in Chapter 9.
2 Estimation of object locations

As mentioned in Chapter 1, we want to analyze 3D object locations. In this chapter, we derive equations to calculate the 3D object locations. The equations use the focal length of the lens with which the images were taken. The focal length has to be estimated and how this is done will be explained. Finally, we will analyze how accurate all estimations are.

2.1 Camera Model

Since the perspective projection makes it impossible to get 3D object locations directly from 2D pixel coordinates, a camera model has to be retrieved. A well-calibrated camera or even a multi-view camera system is needed in order to get the exact values of 3D object locations. Both are normally not available or it requires at least a big effort to set them up, therefore we want to work with a single image from an uncalibrated camera. We will show that the 3D object locations which can be retrieved are precise enough for statistical analysis.

Hoiem et al. [10] propose a zero-skew, unit aspect ratio, perspective camera model and use it to compute the 3D height of objects in images, given only the height of the camera and the horizon line in the image. The algorithm used to estimate camera height and horizon line is also described in [10].

We use the same camera model to retrieve 3D object locations. The way this is done and the camera model itself will be explained in this chapter. Based on the assumption that all objects of interest rest on the ground plane, we can estimate the horizontal 3D distance of the object to the camera center \(x_c\) and its 3D distance in depth to the camera center \(z_c\), given the horizon line in the image and the camera height. Fig.2 illustrates the two values \(x\) and \(z\) which define an object’s 3D location.

2.2 Mathematical derivation of object location equations

Some parts of the following derivation can be found in [11] as well, but because of its importance for all analysis described in this report, it is explained here explicitly and adapted to our case. The following notation is used: pixel coordinates \((u, v)\) ranging from \((0,0)\) at the top-left to \((1,1)\) at the bottom-right; world coordinates \((x, y, z)\) with \(y\) being height and \(z\) being depth; camera tilt \(\theta_x\); focal length \(f\); camera optical center \((u_c, v_c)\); and camera height \(y_c\). The origin of the world coordinates is defined by the camera center, \(z_c = 0, x_c = 0\) and the ground plane, \(y = 0\). We assume zero camera roll and define the horizon position \(v_0\) as the vanishing line of the ground plane in image coordinates. We use a perspective projection model with zero skew and unit aspect ratio.

The transformation from 3D scene coordinates to image coordinates is given by,

\[
\begin{bmatrix}
  u \\
  v \\
  1
\end{bmatrix} = \frac{1}{z}
\begin{bmatrix}
  f & 0 & u_c \\
  0 & f & v_c \\
  0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
  1 & 0 & 0 & 0 \\
  0 & \cos \theta_x & -\sin \theta_x & y_c \\
  0 & \sin \theta_x & \cos \theta_x & 0 \\
  0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
  x \\
  y \\
  z \\
  1
\end{bmatrix}.
\] (1)
Solving for $y$:

$$y = \frac{z(v - v_c \cos \theta_x + f \sin \theta_x) - fy_c}{f \cos \theta_x + v_c \sin \theta_x}$$ (2)

and for $x$:

$$x = \frac{u \cdot z - u_c (y \sin \theta_x + z \cos \theta_x)}{f}$$ (3)

We are interested in the 3D location of the objects, therefore the bottom positions $(v = v_b, u = u_b)$ of the objects need to be regarded. At $(v_b, u_b)$ the object height $y$ is zero. $v = v_b$ and $y = 0$ inserted in (2) and solved for $z$:

$$z = \frac{fy_c}{v_b - u_c \cos \theta_x + f \sin \theta_x},$$ (4)

and $u = u_b, y = 0$ inserted in (3)

$$x = \frac{z(u_b - u_c \cos \theta_x)}{f}.$$ (5)

If the camera tilt is small (e.g. if the horizon position is within the image), the following approximations can be used to greatly simplify these equations: $\cos \theta_x \approx 1, \sin \theta_x \approx \theta_x$ and $\theta_x \approx \frac{v_c - v_0}{f}$. Equations (4) and (5) then become the object location equations:

$$z = \frac{fy_c}{v_b - v_0},$$ (6)

and

$$x = \frac{z(u_b - u_c)}{f} = \frac{y_c(u_b - u_c)}{v_b - v_0}.$$ (7)
Most images are taken with a very small camera tilt which makes the approximations quite good. However, that does not apply to overhead views. Another case for which the object location equations give bad or incorrect results is when objects do not stay on the ground plane.

For clarity we introduce some notations which will be used in the remainder of the paper:

- $\hat{v} = v_b - v_0$: Distance in image coordinates from object bottom positions to the horizon line.
- $\hat{u} = u_b - u_c$: Distance in image coordinates from object bottom positions to the camera optical center.

Object location equations (6) and (7) simplify to

$$z = \frac{fy_c}{\hat{v}},$$ (8)

and

$$x = \frac{y_c \hat{u}}{\hat{v}}.$$ (9)

2.3 Estimation of focal length

Ideally, the focal length of the lens, with which the images were taken, would be read out of their EXIF Header. However, this information was not available for the images in the data set which was used (read more about the data set in Section 3.1), and had to be estimated. How this is done, is explained in the following sections.

2.3.1 Focal length, field of view and angle of view

For a 35mm film, a 50mm lens produces an angle of view which is similar to human vision (approximately 53°)[23]. Therefore 50mm lenses are called normal lenses. Most digital cameras today have a lens-chip combination which produces approximately this angle of view. However, this is not a fixed rule and the field of view can be measured horizontally, vertically or diagonally. Additionally most cameras have a zoom, if the zoom is used, the field of view vary strongly from this described "normal" field of view. Fig.3 illustrates the field of view, angle of view and focal length. The mathematical relation between the field of view, focal length $f$ and the size of the camera chip or film size (for analog photography) $d$ is

$$\alpha = 2\arctan \frac{d}{2f}.$$ 

This can be solved for $f$:

$$f = \frac{d}{2 \tan \frac{\alpha}{2}}.$$
$d$ can either be in meter or pixel, depending on which unit is needed for $f$.

![Figure 3: Illustration of focal length, angle of view and field of view](image)

### 2.3.2 Focal length in pixel

For all experiments we use a focal length estimation of $f = 1.4 \cdot d$, which corresponds to a horizontal angle of view of 39.3° and to a diagonal angle of view of 61.5°. In order to compute the object locations with equation (8), $f$ has to be in pixel units, therefore $d$ has to be in pixel units to calculate the estimation for $f$. We use the image height, which corresponds to the camera’s chip size in pixel.

The accuracy of this estimation will be discussed in Section 2.5.

### 2.4 Top-down view

With the object location equations (8) and (9) we can undo the perspective projection and get object locations in the world coordinate system. We can get a top-down view of the image scene by calculating the 3D locations for every image pixel which belongs to the ground plane. The camera model assumes that the objects stay on the ground plane, therefore it makes no sense to calculate 3D locations for other pixels than ground plane pixels. An example image and its top-down view are shown in Fig.4. The Top-down view in Fig.4 shows that the cars are aligned to two parallel lines which in this case are the curb sides. This shows that the perspective projection has been undone, because the parallelism of the curb sides was not maintained in the image. The algorithm described in [9] is used to reconstruct the geometric structure of the scene and get an estimation of the ground plane. The confidence map for ground plane pixels for the image shown in Fig.4 can be found in Fig.5.

### 2.5 Accuracy of estimations

The estimation of the object locations is based on several other estimations and approximations. First of all the camera height and the horizon line are estimated. Further, we
approximate the object location equations by assuming a small camera tilt. Another estimation is used for the focal length. In this section we want to investigate if the estimates are accurate enough for further analysis of the object locations, or if the introduced error is too big.

To measure the accuracy, we label boundary points of objects of known length. These points have to be on the ground plane, because we can not compute their 3D locations with our camera model otherwise. For example, the point where the front wheel of a car touches the ground, and the point where the rear wheel touches the ground can be labeled. Then the corresponding 3D location of the labeled points are calculated. The length of the chosen object can be found by calculating the distance between these 3D locations, see equation (10). The ratio of this length and the real object length should
be 1. In the example mentioned before, the ratio of the retrieved length and a typical wheel base of a car has to be calculated.

2.5.1 Horizontal object location

The object location equation for the horizontal location is:

\[ x = \frac{y_c \hat{u}}{\hat{v}} \]

As we can see, this is independent of the focal length. No error will be induced by a potentially flawed focal length. The equation for the distance in depth contains the focal length:

\[ z = \frac{fy_c}{\hat{v}} \]

\[ l_1 = \sqrt{(x_1 - x_2)^2 + (z_1 - z_2)^2} \]  

First of all we label objects whose boundary points have the same distance to the horizon line. That means that they have the same \( z \) values. By doing so, the \( z \) values in (10) cancel out each other and the length of the object, \( l_1 \), becomes independent of \( z_1 \) and \( z_2 \) and therefore independent of the focal length. Therefore, the accuracy of the estimation for the camera height, the horizon line and the assumption of a small camera tilt is evaluated, but not the estimation of the focal length. What has to be labeled is illustrated in Fig.6.

Figure 6: (a) is an example scene, (b) the corresponding top-down view. If the calculated object length should be independent of the focal length, object 1 should be labeled. To examine the consistency of the focal length, object 2 should be labeled.

We looked for images in our data set which contain objects of known length, whose boundary points have the same distance to the horizon line. Of course it is barely possible to find objects for which this distances are exactly the same, but the smaller the difference of the distances is, the more independent \( l_1 \) gets from the focal length. Therefore we tried to keep this distance as small as possible when we were labeling
the boundary points. One pixel close to the horizon line can correspond to more than a meter in 3D coordinates, whereas it corresponds to a couple of centimeters if it is far away from the horizon line. To keep the influence of labeling errors as small as possible, it is therefore important to choose objects far away from the horizon line. Fig. 7 shows the result for 15 objects in 15 different images. For all 15 objects their length was calculated as explained above. The ratio is computed by dividing the calculated length by the known length of the object. The mean of the 15 ratios is 0.99 with a variance of 0.03. This shows that all estimations were precise and the assumption of a small camera tilt held.

![Figure 7: Ratio of calculated object length (based on estimations) and the known object length for 15 objects in 15 different images.](image)

### 2.5.2 Object distance in depth

As we have seen before, the estimated focal length is needed to calculate the $z$ value of the object locations. The focal length is the value, which relates pixel values to values in meters, if it is inaccurate it makes no sense to compare any 3D object locations in meters. In the last section we wanted to keep the influence of the focal length very small. Now we want the influence of the focal length to be as large as possible, to be able to investigate its accuracy. Ideally, we label an objects whose boundary points have the same horizontal location ($x_1$ and $x_2$ cancel each other out in (10)). How the labeling should be done is illustrated in Fig. 6. To keep the influence of labeling errors small it is again very important to choose objects which are far away from the horizon line. We labeled boundary points for 27 objects in 27 different images. We tried to find objects whose boundary points have the same horizontal location. The ratios of
the calculated object depths and their known object depths can be seen in Fig.8. The mean of the ratios was 1.06 with a variance of 0.06. This shows that the accuracy of the estimated focal length is also very good.

Figure 8: Ratio of computed object length and the known object length for 27 objects in 27 different images.
3 Initial analysis of object locations

This chapter describes the first experiments performed to find out more about 3D object locations. In the first section, the data set, which is used for all analysis in this report will be described.

3.1 Our data set

The data set we use to gather statistics of object locations is a subset of the labelMe data set [4]. For now, we limit our investigations to locations of people and cars. The labelMe users who are labeling objects in images are not restricted to certain words when annotating the objects. For example, a car can be annotated as "car", "taxi", "blue car", "automobile", "suv", etc. and a person as "person", "man", "woman", "person upright", etc.. For our analysis we do not want to distinguish between these different kinds of cars or people. When querying for objects, LabelMe offers the option to use WordNet, an electronic lexical database [15], to combine objects with different descriptions to one category, e.g. car or person.

Further, we want to limit our data set to urban outdoor scenes. LabelMe image folders whose name contain words like "indoor", "office", "beach", "nature", etc. are excluded. Our subset contains 2667 images, with 5941 cars and 2286 people in total. The number of images which contain 1, 2, 3,...N cars or people can be found in Fig. 9.

Before any object locations can be calculated, the parameters for our camera model are needed. Namely the camera height and the horizon line in the images. We use the algorithm described in [11] to get an estimation of these parameters for every image in our data set. Once these parameters are estimated, the 3D locations for all objects in all images can be calculated (see Chapter 2), and analyzed. The initial analysis is described in the following section.

![Histograms showing number of images containing 1, 2, 3,...N cars and people](image)

Figure 9: These histograms show how many images of our data set contain 1, 2, 3,...N cars(left) or people(right)
3.2 Absolute object locations

We refer to the world coordinates $x$ and $z$ of objects as “absolute object locations”. The origin of our world coordinates is defined by the camera center, $z_c = 0$, $x_c = 0$ and the ground plane, $y = 0$. Fig. 2 in Chapter 2 illustrates $x$ and $z$ in an example scene. Fig. 10 and Fig. 11 show histograms of absolute object locations of all cars and all people in the entire data set. Objects which were too far away from the camera center were discarded (see description of the histogram figures). These plots give a first impression where the objects are located.

3.2.1 Interpretation of results

In Fig. 10 it seems that the majority of the cars appear to be closer than 30m and the majority of people appear to be closer than 20m to the camera center. This however, has to be interpreted with skepticism. The main reason for this is probably that people, who are labeling objects, tend to neglect objects which are far away. A car very far away, what means close to the horizon line, might just be represented by a couple of pixels in the image. Therefore the car is likely not to be labeled. To avoid this problem, we only regard objects in a closer range, see Fig. 11. The objects seem to be equally distributed in the scene, no peaks or regularities are present.

![Figure 10: Histograms of absolute object locations of all cars (left) and all people (right) in our data set. Objects farther away than $z = 70m$ or $x = 30m$ were discarded.](image)

3.3 Object locations relative to other objects

In this section we want to investigate if any regularity can be discovered when regarding the object locations relative to other objects. One object is selected as the center object. Then the distances in both directions ($x$ and $z$) to all other objects of the same kind in the same image are calculated. Every object in our data set was selected as center object once and the calculated distances were accumulated in a histogram. Fig. 12 shows the results for cars and people.
Figure 11: Histograms of absolute object locations of all cars (left) and all people (right) in our data set. Objects farther away than $z = 25 \text{m}$ or $x = 10 \text{m}$ were discarded.

Figure 12: Each object in the database was selected as center object (0m/0m). The object locations of all other objects in the same image, relative to the center object, were accumulated in this histogram. The result for cars is left, the result for people is right.

3.3.1 Interpretation of results

The relative car locations are still scattered and no obvious regularity can be observed in Fig.12. In the case of relative locations of people, there seems to be some regularities. The location of a person is likely to be left or right of another person. It can also be seen in Fig.12 that people seem to be located closer to one another than cars. This observation is not very surprising. For a better understanding of these
relative location histograms more investigations are described in the next chapter. Since cars are normally driving on the same road, in the same direction or are parked next to one another, it is not evident why the car locations are scattered in our plots. Later we will see, what has to be done to detect regularities in car locations.
4 The importance of a reference direction

Instead of any kind of regularity in the histograms for relative locations of cars, we got a cloud of scattered object locations (see Fig. 12). This chapter investigates and explains what additional knowledge about the scene is necessary to be able to detect regularities in relative car locations.

4.1 Getting a better understanding of the location histograms

To get a better understanding of what is really recorded by the location histograms and to find out where the cloud in the histogram of relative car locations (Fig. 12) is coming from, a closer look at two typical urban scenes is described in this section. Fig. 13 shows the histogram of two scenes separately. The combined histogram of these two scenes is depicted in Fig. 14. It can be seen that the car locations are not mapped to the same points in the histogram. Nevertheless, there is a certain similarity of the locations of the cars in the two scenes, namely the alignment along the street and the distances from one car to the others. Apparently it is not appropriate to use the same coordinate system for different scenes, when the orientation of the camera relative to the scene is different. A coordinate system which is invariant to different camera orientations has to be found.

Most urban outdoor scenes have a distinctive direction. The objects in the scene are aligned to it. Cars drive or are parked in this direction and people walk in this direction. Often it is the direction of the principal street in the image, or the direction of a building along whose side the people walk. We will call this direction the scene direction from now on. The scene directions of our example scenes in Fig. 13 are indicated in Fig. 15. We used the scene directions as reference directions to create a new coordinate system in which we calculate the relative object locations. How this is done will be explained in the following sections. The new histograms of the relative object locations are shown below the example scenes in Fig. 15 and the combined histogram of the objects of both scenes is shown in Fig. 16.

4.2 Scene direction as reference direction

Urban outdoor scenes, like the scenes in our data set, are based on a 3 dimensional Cartesian coordinate system $\vec{i}, \vec{j}, \vec{k}$ [14]. All three axis have a corresponding vanishing point in images of urban outdoor scenes. When we know the camera orientation relative to these 3 coordinates, interpretation of the scenes gets much easier [8]. Our observation in the previous section was similar: we need to know the orientation of the camera relative to the scene, to compare object locations in different images.

We assume zero camera roll, vertical lines in the world ($\vec{k}$ direction) are therefore mapped to vertical lines in the image. Building walls, street curbs and other man-made structures are typically aligned to the other two directions ($\vec{i}, \vec{j}$). One direction corresponds to what we call the scene direction and the other one is orthogonal to the scene direction. As mentioned before, the scene direction will be, for example, the direction in which the principal street in the image goes, and can therefore be defined by the vanishing point of this street.
4.2.1 Labeling the vanishing point of the principal street

All lines, which are parallel in the 3D world and are contained by the ground plane will intersect at the line at infinity of the ground plane. In the image this line at infinity is represented by the horizon line. In the previous section we explained that we want to define the direction of the principal street in the image by the street’s vanishing point. To find the vanishing point in the image, we label the street curbs or lines which are parallel to the street curbs. The vanishing point can then be calculated by intersecting the labeled lines with the horizon line. Fig.17 illustrates the labeling process.

4.2.2 From vanishing point to scene direction invariant coordinates

The vanishing point of the principal street was recovered in order to specify the scene direction. In order to represent this scene direction in the top down view and to create a coordinate system corresponding to this scene direction, we connect the bottom positions of all objects with the vanishing point. We imagine these connection lines as being contained in the ground plane. Lines intersecting at the horizon line, the line at infinity, are parallel in the real world and therefore also in our top down view. That
Figure 14: Histogram of relative car locations for two typical urban scenes. The car locations are not mapped to the same points in the histogram, even though there is a certain similarity of the locations of the cars in the two scenes, namely the alignment along the street and the distances from one car to the others (see example scenes in Fig. 13).

means that the lines from the objects to the vanishing point are parallel in the top-down view. These parallel lines specify our scene direction and define the first axis of our coordinate system. The second axis is perpendicular to the first axis. Fig. 18 visualizes what has been described above.
Figure 15: The two typical urban scenes with their histograms of relative car locations. The scene directions (the red arrows) were used as reference directions.

Figure 16: Histogram of relative car locations for two typical urban scenes. The scene direction was used as reference direction, therefore the car locations are mapped to the same points in the histogram.
Figure 17: Illustration of the labeling of the vanishing point. The horizontal blue line is the horizon line, the other blue lines are the labeled lines, which are parallel in the 3D world. The two red lines mark the intersection points with the horizon line. Due to labeling inaccuracy they are not at the exact same position. The vanishing point is calculated to be in the middle of the two intersection points.

Figure 18: Typical urban street scene - in the image on the left, the lines connecting the objects with the vanishing point are drawn in blue (shortened for better visualization), the vanishing point is where the red line intersects the blue horizon line - the right image shows the locations of the objects in a top down view, the correspondences of the blue lines are drawn in the top down view as well, it can be seen that they are parallel now. These lines define the direction of the first axis of our new coordinate system.
5 Gathering statistics

In the previous chapter we discovered that we need to know the scene directions of the images, to gather meaningful statistics of object locations. In this chapter we will present the histograms of relative object locations, which we gather from images with known scene direction. Then we will introduce a new way of analyzing 3D object locations. We will gather statistics of object distances to building walls.

5.1 Statistics of relative object locations

In this section we incorporate the knowledge of scene directions of the images in our data set to gather statistics of relative object locations. We have seen that the scene direction can be represented by the vanishing point of the street which defines the scene direction. To incorporate the knowledge of scene directions, we labeled this vanishing point in 1494 images of our data set. These images contained 3120 cars and 1368 people. To gather the statistics of relative object locations, we use the image specific vanishing point to create a coordinate system in the top down view for every image. The first coordinate axis ($\vec{x}'$) goes in the direction of the scene and the second axis ($\vec{z}'$) is orthogonal to the first one (as explained in 4.2.2). Since we are only interested in relative object locations, it does not matter where the origin of the coordinate system is. We transform all object locations into the new coordinate system. This has to be done separately for each image, because every image has another scene direction. The following procedure is the same as for our initial analysis in Section 3.3. Every object is once selected as center object and the distances to all other objects of the same kind in the same image are calculated. The distances in both directions $\vec{x}'$ and $\vec{z}'$ are accumulated in a histogram. Fig.19 shows the result for relative locations of cars and people. Distances bigger than a certain maximum distance were discarded (see figure description).

![Histograms of relative car locations (left) and relative people locations (right). Distances bigger than 10m in $\vec{x}'$ direction or bigger than 25m in $\vec{z}'$ direction were not recorded in the histogram](image)

Figure 19: Histograms of relative car locations (left) and relative people locations (right). Distances bigger than 10m in $\vec{x}'$ direction or bigger than 25m in $\vec{z}'$ direction were not recorded in the histogram
The object locations in the histograms are less scattered than they were in our initial analysis where we did not incorporate the knowledge about the scene direction (compare Fig. 19 with Fig. 12). There are distinct peaks in the new histogram for relative car and relative people locations. The peaks indicate that some object locations are more likely than others. In the histogram for cars we can see that, given a car, there is an increased likelihood that there are other cars in front of or behind this car. In the histogram for people we can see, that given a person, there is an increased likelihood that there are other people left or right of this person. In front of, behind, left and right with respect to the scene direction. Where exactly the peaks come from and how distinctive the increased likelihoods are will be evaluated in Chapter 6.

5.2 Statistics of object distances to building walls

The goal of this work is to find statistical regularities of object locations, regarding urban outdoor scenes. Due to street widths and car sizes which are not completely random, the object distances to building walls show some regularities. These regularities will be investigated in this section.

To measure the object distances to buildings, we need to label the intersection line between vertical planes in the image and the ground plane. This is not an easy task, because this intersection line is often occluded by objects, for example by cars parking on the side of the street next to the building. When the intersection line is manually labeled however, it is possible to keep the labeling errors small. Fig. 20 shows three different scenes for which labeling of the intersection line is very easy in the first case, problematic but still possible in the second case and not possible in the third case. We labeled intersection lines in 355 images of our data set, transformed the intersection lines to the top down view and calculated the distance of all objects to the intersection lines. There were 709 cars and 420 people in the 355 images which were labeled.

Statistics of object distances to building walls can only be useful in applications in which the intersection lines can be detected automatically. Different works have been published which describe how to recover surface geometry from a single view image [9, 22, 26] or how to recover a 3D depth map [2]. Once the surface geometry is recovered it should also be possible to detect the intersection line between vertical planes and the ground plane. We did not examine how reliable such a detection can be.

Figure 20: Labeling of the intersection line between vertical planes and the ground plane can be easy (left), problematic (middle) and impossible (right).
5.2.1 Filtering of the object distances

Not all object locations in an image are related to every building wall in the image. In Fig. 21 the location of the white car in the front, is not related or restrained by the building wall whose intersection line is labeled in red, but it is restrained by the wall whose intersection line is labeled in yellow. To filter out such arbitrary object distances, we labeled the orientation of all objects, and only kept distances of objects whose orientation was roughly parallel to the intersection line of a wall. Additionally we only kept distances which were smaller than a certain maximum distance. By applying these two filters the peaks in the object distance histogram became more distinct. Fig. 22 shows the resulting histogram for car and people distances to building walls with a maximum distance of 20m.

![Image of a car in front of a building wall with labels on the intersection lines.]

Figure 21: The location of the white car in front is not related or restrained by the building wall whose intersection line is labeled in red, but it is restrained by the wall whose intersection line is labeled in yellow.
Figure 22: The resulting histogram (after the filtering) for car and people distances to building walls with a maximum distance of 20m.
6 Interpretation and evaluation of the statistics

In the previous chapter we retrieved histograms of relative object locations (Fig. 19) and of object distances to building walls (Fig. 22). These histograms will be analyzed and evaluated in this chapter. The goal is to find out how powerful the statistics are. Given one object and the scene direction, or given the location of a building wall, we want to use the gathered statistics, to predict the locations of other objects.

6.1 Visualize the likelihoods of object locations in images

There are distinctive peaks in the histograms of relative object locations (Fig. 19) and in the histograms of object distances to building walls (Fig. 22). We will use the gathered statistics to compute a probability density function (pdf) and then project the likelihood of objects at certain locations, \( P(\text{object} \mid \text{relativeLocation}) \), or at certain distances, \( P(\text{object} \mid \text{distanceToWall}) \), back into 2D images. This gives a visual impression of how well we can predict object locations with the obtained statistics. In the remainder of this report we will use the abbreviations \( \text{obj} \) for object, \( l \) for relative object locations, and \( d \) for object distances to building walls.

6.1.1 Computation of probability density functions

The histograms show how often every relative object location or every object distance occurred in the scenes of our data set. We will use kernel density estimation to compute probability density functions (pdf) for the relative object locations and the object distances. With the pdf we can then calculate likelihoods for certain object locations, \( P(l \mid \text{obj}) \), or distances, \( P(d \mid \text{obj}) \).

Object distances to building walls

While analyzing our data subset with labeled building-ground intersection lines, we gathered \( N = 508 \) distances from cars to building walls and \( N = 238 \) distances from people to building walls. Some distances were filtered out before as described in Section 5.2.1. The pdf is calculated by using kernel density estimation, see equation (11). \( N \) is the number of data samples (distances in our case); \( x_1, x_2, ..., x_N \) are the data samples; \( h \) is the band with (smoothing parameter) and \( K \) is the kernel. We use a standard Gaussian function with mean zero and variance 1 as kernel (see equation (12)) and a band with \( h = 1 \). Fig.23 shows the resulting pdfs for car and people distances.

\[
f(x) = \frac{1}{Nh} \sum_{i=1}^{N} K \left( \frac{x - x_i}{h} \right) \tag{11}
\]

\[
K(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2} \tag{12}
\]
Relative object locations

We gathered 3802 data samples for locations of cars relative to other cars and 2262 data samples for locations of people relative to other people. Note: Locations contain two values, the distance from one object to another object in \( \vec{x}' \) and \( \vec{z}' \) direction. Therefore the pdf is two dimensional. A two dimensional Gaussian kernel was used. For more theory about the kernel density estimator, see [5]. Fig.24 shows the resulting two dimensional pdfs for relative car and people locations.

![pdf for distances from cars to building walls](image1.png)

![pdf for distances from people to building walls](image2.png)

Figure 23: Probability density functions for distances from cars to building walls (left) and for distances from people to building walls (right).

6.1.2 Back-projection of locations likelihoods into 2D image plane

We can visualize how well the location of an object can be predicted by knowing the location of another object and the scene direction, or by knowing the location of a building-ground intersection line. First, we choose an intersection line or an object in an image as a reference object. We calculate the corresponding 3D coordinates of every pixel below the horizon line in the image using our camera model. Then we calculate the distance to the intersection line or the relative location of every pixel’s 3D coordinates and assign a probability to the distance or relative location. In the next section (Section 6.2) we will explain more in detail how we obtain likelihoods of relative object locations, \( P(l|obj) \), and distances, \( P(d|obj) \), from the pdf. We want to visualize the likelihoods of objects at certain locations:

\[
P(obj|l) = \frac{P(l|obj) \ P(obj)}{P(l)} \propto P(l|obj)
\]

and
For the visualization we do not have to worry about the normalization factors $P(obj)$, $P(l)$ and $P(d)$ because they will be the same for all objects of the same class. Therefore we visualize $P(l|obj)$ and $P(d|obj)$. We use the color red to indicate probability values. Fig. 25 shows the likelihood of car locations, Fig. 26 shows the likelihood of people locations, and in Fig. 27 shows the likelihood of car locations depending on the distance to building walls. To find out whether the street with the cars is on the left side or the right side of the building wall, we estimate the geometric structure of the scene and compute a confidence map for ground plane pixels with the algorithm described in [9]. We add up the ground plane confidence values for all pixels in a certain area on both sides of the intersection line. The street is chosen to be on the side on which the sum of ground plane confidence values is higher.

6.2 Compare location prior with uniform distribution

With the probability density function we can assign probabilities to relative object locations and to object distances to building walls. This gives us a prior on locations and distances respectively. We now use our data set with labeled objects, to see how well the priors would have predicted the object at the given locations. To get an impartial evaluation, we calculate a second type of priors by assuming that the objects are uniformly distributed in a certain area around the reference object or uniformly distributed within a certain range of distances to building walls. With the data set, we also examine how well this second type of priors would have predicted the objects and calculate the prediction improvement which can be achieved by using the statistics.
Figure 25: Red indicates the likelihood of a car. The car whose location was chosen as reference location, is framed with a bounding box with a thick bottom line.

**Object distances to building walls**

The prediction improvement can be calculated with the following formula:

$$I(d|obj) = \frac{P(obj|d)}{P_u(obj|d)}$$  \hspace{1cm} (13)

with

$$P(obj|d) = \frac{P(d|obj) \ P(obj)}{P(d)}$$

and

$$P_u(obj|d) = \frac{P_u(d|obj) \ P(obj)}{P_u(d)}$$

(13) becomes:

$$I(d|obj) = \frac{P(d|obj) \ P_u(d)}{P_u(d|obj) \ P(d)}.$$  \hspace{1cm} (14)

$P(d|obj)$ is the prior retrieved with the statistics and $P_u(d|obj)$ is the prior retrieved based on the assumption of uniform distribution of object distances to building walls. In order to eliminate the influence of the normalization factors $P(d)$ and $P_u(d)$ we
Figure 26: Red indicates the likelihood of a person. The person whose location was chosen as reference location, is framed with a bounding box with a thick bottom line.

want them to be equal and therefore be able to be canceled out in (14). $P(d)$ and $P_u(d)$ are equal, when the distances which were used to gather the statistics and to retrieve the pdf (these are the distances which have a non-zero probability in the pdf) are in the same range $[0, \text{maxDist}]$ than the distances in case of uniform distribution. If the equality of $P(d)$ and $P_u(d)$ is assured, equation (14) can get simplified to

$$ I(d|\text{obj}) = \frac{P(d|\text{obj})}{P_u(d|\text{obj})} \quad (15) $$

We will now explain, how actual values for the prior $P(d|\text{obj})$ can be calculated from the pdf. We can estimate a probability $P(a \leq d \leq b | \text{obj})$ for any distance interval $[a, b]$. $P(d|\text{obj})$ gets the probability value of the interval in which $d$ falls. $\text{maxDist}$ in our pdf in Fig.23 is 22.7m for car distances and 23m for people distances. We divide the whole range in 100 intervals which gives an interval width of 0.227m for car distances and 0.23m for people distances.

We use the same interval width for the case of uniform distribution. To calculate probabilities we need to specify, within which range $[0, \text{maxDist}]$ the distances are uniformly distributed. After specifying a maximum distance, we can calculate the probability $P_u(d|\text{obj})$ for all intervals ($\#\text{intervals} = \text{maxDist}/\text{intervalWidth}$, $P_u(d|\text{obj}) = \ldots$)
Figure 27: Red indicates the likelihood of a car. The intersection line between ground plane and vertical building plane is indicated by the blue line.

\[ \frac{1}{\# intervals \ for \ d < \text{maxDist}} \]. For distances farther away than the maximum distance we assign the probability zero, to all other distances we assign the same probability \( P_u(d|\text{obj}) \). Fig. 29 illustrates the connection between the maximum distance and the probability value.

In order to evaluate the predictive power of the statistics of distances to building walls, we calculated all distances \( d \) to the intersection lines in images of a subset of our data set. The subset contained 112 images and was not used to gather the statistics. There were 226 cars and 101 people in the images. We calculated the prediction improvement \( I(d|\text{obj}) \) for all distances and determined the average improvement. Since \( P_u(d|\text{obj}) \) and therefore also \( I(d|\text{obj}) \) (see equation (15)) depends on the chosen maximum distance, we computed the average improvement for different maximum distances between 1m and 51m. Fig. 28 shows the result. The improvement varied between 0.98 and 3.2 for cars and 1.12 and 3.9 for people. This shows that the prediction can not really be improved for objects in a very small range, but gets significantly better when objects in a larger range are regarded.
Relative object locations

To evaluate the predictive power of the statistics of relative object locations we follow the same procedure as for object distances to building walls. We divide the range
of relative object locations into 256 bins in both directions ($\vec{x}'$ and $\vec{z}'$), which results in $256 \times 256$ bins. With the pdf we calculate the probability of every bin. We also used our data set to calculate the prediction improvement for relative car locations. We calculated the locations of all objects in all images relative to all other objects in the same image. Then we calculated the prediction improvements $I(l|obj)$ for all locations and determined the average. Fig. 30 shows the result for maximum distances between 1m and 51m. The prediction improvement is between 1.03 and 34.3 for cars and between 1.2 and 101.4 for people. Due to the limited size of the data set, we did not label the scene direction in new images, which has not been used for the gathering of the statistics. Instead, we followed the concept of leave-one-out cross-validation. For each image separately, we calculated a pdf by using the data from all available images except the actual image. This pdf was then used to calculate $I(l|obj)$ for all relative object locations in this image.

Figure 30: Improvement of the prediction of relative object locations by using the statistics.
7 Automatic detection of scene direction

In Chapter 4 we discovered that it is important to know the scene direction of an image in order to analyze the object locations. The scene direction can be specified by the vanishing point of the principal street in the image. Vanishing points or the knowledge about the relative orientation of the camera with respect to the scene, can be very useful in many different computer vision tasks from geometry recovery to robotic navigation. For example, Criminisi et al. use the vanishing point together with a certain vanishing line to measure distances between planes and to compute other 3D affine measurements [1]. In [25] Utcke describes how the vanishing point is used in a vision system for the blind. Because of this variety of applications, many different algorithms for vanishing point detection have been proposed in the past [6, 13, 8]. We will first give a brief description of J. Kosecka’s and W. Zhang’s video compass algorithm [18]. Then we describe an experiment which we did to recover vanishing points with a data-driven approach. A third described method directly detects the scene direction and requires given objects in the test image. The statistics of relative object locations are used to derive the scene direction. The last approach also requires given objects in the image and fits a line to the object locations to estimate the scene direction. The detection accuracies of the different algorithms are compared at the end of this chapter.

7.1 Video Compass

The approach is based on the fact that the majority of lines in man-made environments is aligned with the three orthogonal axes of the world coordinate frame. This is exploited to detect vanishing points. See [18] for a detailed description of the algorithm. The first part of the algorithm is the line detection stage. Canny edge detector is used to find lines and get an orientation of the gradient of every edge pixel. The edge pixels are grouped into $k$ bins according to their gradient orientation. It can happen that edge pixels which belong to the same line fall into different orientation bins (the handling of these artifacts is done in a later step of the algorithm) but basically every line is assigned to one of the $k$ bins. Lines whose length is below a certain threshold are discarded. The length threshold depends on the image size, we used 2.5% of the image diagonal. Each line longer than this threshold is represented as a list of pixels $(x_i, y_i)$. These lines need to be grouped into vanishing directions. Since every two lines which are parallel in the 3D world intersect in a vanishing point in the image, a large set of vanishing directions might be obtained initially. As mentioned before, most lines are aligned to three principal directions in man-made environments. The aim is to find the three corresponding dominant vanishing directions, and to classify lines not belonging to these directions as outliers. The grouping stage and vanishing estimation stage is addressed simultaneously as a probabilistic inference problem. The algorithm used for the estimation of vanishing point coordinates is called Expectation Maximization algorithm (EM) and was suggested by [3]. It calculates the probabilities of line segments belonging to a particular vanishing direction and maximizes this probability by regrouping lines and merging vanishing directions till its convergence. The initial grouping of lines is done based on the assumption that lines belonging to the same vanishing direction have similar orientations in the image. In fact, this is just the case.
if the vanishing point is outside the image. Lines belonging to a vanishing direction with the vanishing point inside the image will be merged by the EM algorithm later. In experiments described in the video compass paper, the EM algorithm converged successfully in 2 to 5 iterations. The output contains three vanishing points. We are only interested in the vanishing point of the principal street in the image. This vanishing point has to be on the horizon line in the image, because this is the vanishing line of the ground plane. Therefore we select the vanishing point closest to the horizon line, as the vanishing point of the principal street. We will present results which we got for images of our data set later on in this chapter.

### 7.2 Data-driven vanishing point recovery

In [20] Oliva and Torralba proposed a low dimensional representation of the structure of scenes. They call it the gist or spatial envelope of a scene. They argue that such a low dimensional representation can be sufficient to preserve certain structural features of the scene. Hays and Efros use the gist representation in an image completion application. They store images and their gist representation in a database. To complete an image, they search in the database for the nearest neighbor images based on the low dimensional gist representation. The nearest neighbor is then used to complete a certain part of the input image [16]. Divvala et al. use a similar procedure and the gist to find nearest neighbors to help single-view geometry estimation [22]. Hoiem et al. use an example-based technique and the gist to recover the horizon position in images [11]. In the following we explain an approach to recover the scene direction, which is identical to the horizon line recovery in [11].

We calculated the gist descriptor for 1250 images and labeled the scene direction specifying vanishing points in all images. We built the gist descriptor from 4 oriented edge responses at 4 scales aggregated to a 4x4 spatial resolution. This results in a gist descriptor with 256 dimensions. By applying principal component analysis (PCA) and only keeping the first principal components which contained at least 94% of the information energy, we further reduced the dimensionality. Using this descriptor for all 1250 training images, we trained a generalized regression neural networks (Matlab function: newgrnn). The neural network estimates the vanishing point of test images by

\[
\hat{v}(x) = \sum_i v_i w_i \text{ with } w_i = \exp\left[-\frac{\|x - x_i\|^2}{2\alpha}\right]
\]  

(16)

where \(v_i\) is the \(x\) coordinate of the vanishing point in training image \(i\). \(x\) is the gist descriptor of an image. The \(y\) coordinate of the vanishing point lies on the horizon line and does not have to be estimated. \(\alpha\) is called the spread. The larger the spread, the smoother will the estimated value be approximated to the values in the training set.

#### Evaluation

With a training data set of just 1250 images, no reasonable results could be achieved. The Euclidean distance in gist (\(\|x - x_i\|^2\)) was too big in most cases and a result could only be obtained when \(\alpha\) was chosen to be larger than 5. This on the other hand
increases the influence of training images which are not really similar to the test image. The estimation error was huge and the estimated vanishing points almost seemed to be randomly chosen. Hereby, we make a similar observation than Hayes et al. in [16], who got discouraging results with a data set of ten thousand images. They extended their data set to two million images which "yielded a qualitative leap in performance". Our data set is considerably smaller and increasing it is not easily possible, because the vanishing points have to be labeled manually. Hoiem et al. already got reasonable results for the estimation of the horizon line, with 2,660 training images. A reason for this could be that the standard deviation of horizon line positions is quite small. The horizon line position is usually close to the image center. The horizon line position in images which are not similar to the test image (what results in a big Euclidean distance in gist) can therefore still be similar to the horizon line position in the test image. A reasonable estimation can then still be done for test images with no similar training image in the data set.

7.3 Vote for scene direction

In this chapter we want to detect the scene direction by using the statistics of relative object locations. The motivation to do this, is on the one hand to further evaluate the predictive power of the statistics, and on the other hand to get another method to automatically detect the scene direction. The method is based on the assumption that the objects in the images are given. If there is no object in the image, the method can not be applied. First, one object in the image under scrutiny is chosen as reference object. The locations of all other objects relative to the reference object are calculated. We assign probabilities to the retrieved relative object locations by using the probability density function, as explained in Section 6.2. Every object in the image is chosen to be the reference object once. Then we add up the probabilities to get a measure of how likely the scene, with this very object locations is, according to our statistics. To calculate relative object locations, it is necessary to have a reference object and a reference direction, the scene direction. This is where the voting comes in to play. We calculate the relative object locations and the corresponding probabilities for \( N \) different scene directions. The direction for which the sum of probabilities has the highest value, is chosen to be the scene direction. Fig.31 illustrates the voting process. See the image description for explanations.

7.4 Estimate scene direction with robust line fit

To evaluate how useful the statistics are to detect scene directions, we want to compare the voting with a method which is based on a robust line fit. This method also requires that there are objects in the image. Based on the assumption that several cars in the image are driving or are parked on the principal street which defines the scene direction, we fit a line to 3D locations of the cars. If the cars are aligned and not next to each other, the direction of this line approximates the direction of the street and therefore the scene direction. We also fit a line to the 3D locations of people. We assume that people are walking next to each other along the principal street, therefore the scene direction is perpendicular to this line. If there are cars and people in the image, we can calculate
Figure 31: Illustration of the voting. The same scene is illustrated with two different scene directions which are indicated by red arrows. The rectangles are cars, the circles are people. The car and the people marked with an “R” are the reference objects. The pdfs for relative people and car locations are projected into the scenes, adapted to the respective reference object and scene direction. The numbers are the resulting probabilities for the locations of the other objects, given this reference object and this scene direction. The sum of probabilities with the scene direction assumed in the scene on the left is 20.01, the sum of probabilities with the other scene direction is 0.17. Every object in the scene has to be chosen as reference object once while the scene direction remains the same, then the sums of probabilities are compared. The direction which lead to the highest sum, is chosen as the actual scene direction (here it would be the scene direction indicated on the left).

two lines. The method usually gives better results when there are many objects in the image. Therefore take the average of the two directions, weighted with the number of cars and number of people in the image.

7.5 Compare video compass, voting and line fit

Finally we want to compare the three described methods, voting, line fit and J. Kosecka’s video compass. Voting and line fit deliver a scene direction as output. Video compass estimates vanishing points. In order to compare video compass with the two other methods, we calculate the corresponding scene direction with the vanishing point as explained in 4.2.2. Neither of the three methods is able to deliver an output in any case. Voting and line fit can not estimate a scene direction when there are less than two objects in the image. Video compass fails for images without long straight lines which are used to determine the location of vanishing points. To be able to compare the three algorithms we only used images for which all three deliver an output. Out of 939 images with at least two people or two cars, the scene direction could be detected for 604 images. We compared the detected scene directions with the labeled scene directions. The mean and the standard deviation of the errors for all three algorithms are listed in Table 1.

By doing the analysis of relative object locations, we have learned that cars tend
### Table 1: Mean error and standard deviation of the error of the scene direction detection.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Mean error</th>
<th>Standard deviation of error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Video compass</td>
<td>25.29°</td>
<td>34.34°</td>
</tr>
<tr>
<td>Voting</td>
<td>33.89°</td>
<td>36.63°</td>
</tr>
<tr>
<td>Line fit</td>
<td>36.46°</td>
<td>35.66°</td>
</tr>
</tbody>
</table>

### Table 2: Mean error and standard deviation of the error of the scene direction detection for images with at least 4 objects.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Mean error</th>
<th>Standard deviation of error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Video compass</td>
<td>24.1°</td>
<td>35.25°</td>
</tr>
<tr>
<td>Voting</td>
<td>28.5°</td>
<td>33.06°</td>
</tr>
<tr>
<td>Line fit</td>
<td>34°</td>
<td>34.77°</td>
</tr>
</tbody>
</table>

to be located behind each other in respect to the scene direction and people tend to be located next to each other in respect to the scene direction. The assumptions on which the line fit method is based come from these observations. Now we want to see how much the scene direction detection can benefit by using the exact statistics instead of just these observations. Therefore we do another comparison between the voting and line fit method. With only two people or two cars in the image, both algorithms are not very robust. If there are more objects in the image, the inference of the scene direction from the objects gets much stronger. We compare the results of voting and line fit for images with at least 4 cars or 4 people. The error could be reduced by 14.1 % for voting and 6.4 % for line fit, compared to the detection for images with at least two cars or two people. This shows that if there is a good basis on which the inference of scene directions can be done, we benefit from the exact statistical knowledge and the voting performs much better than the line fit. Table 2 shows the results.

#### 7.5.1 Summary

For all three algorithms, the error is large. Remarkable are the high standard deviations of the errors. This means that there must be very big errors in some images and very small errors in others, what implies that the scene direction detection works in some images and in others it does not work. The reason for this could be that the assumptions on which the methods are based hold in some images but do not hold in others. The assumptions for line fitting are for example, that cars are driving or parking on the principal street behind and not next to each other. In images where the voting performs poorly, the objects might not be located as predicted by the statistics. This case will be discussed more in the result chapter (Chapter 8). A reason why video compass performs poorly for some images could be that the images were cluttered and not enough long straight lines could be detected. R. Collins and R. Weiss observed in [7] that vanishing point detection is not very reliable when less than 20 lines which contribute to a vanishing direction can be found in the image, as can be the case in cluttered scenes.
Table 3: Mean error and standard deviation of the error of the scene direction detection. Images for which the error was bigger than 20° were filtered out.

Hoiem claims in [11] that vanishing point detection algorithms based on edges and perspective show good results for uncluttered man-made scenes but fail for less structured environments.

To see how well the methods would work if there were a method to filter out images for which the algorithms are not adequate, we calculated mean and standard deviation of the errors in images where the error was smaller than 20°. See Table 3. 130 images were left for this analysis.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Mean error</th>
<th>Standard deviation of error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Video compass</td>
<td>4.34°</td>
<td>4.11°</td>
</tr>
<tr>
<td>Voting</td>
<td>5.61°</td>
<td>5.38°</td>
</tr>
<tr>
<td>Line fit</td>
<td>7.43°</td>
<td>4.84°</td>
</tr>
</tbody>
</table>
8 Results

We found out that knowing the orientation of the camera relative to the scene can be crucial to analyze 3D locations of objects in images. In urban outdoor scenes, this orientation can be specified by the vanishing point of the principal street in the image. We called the direction of this street the scene direction. In the first section of this chapter, we will present results which we got for the analysis of object locations relative to other objects. The scene direction was labeled in the training images which were used to gather statistics of relative object locations. We used test images to evaluate how well object locations can be predicted with the obtained statistical knowledge. The scene directions in these test images were also labeled manually. We will show in what kind of images the statical knowledge is helpful and in what kind of images the statistics do not fit. We also present results which we got for the analysis of object distances to building walls. In the second part of this chapter, we will talk about the prediction performance for relative object locations which can be achieved when the scene direction is detected automatically in training and test images. Finally, the prediction performance for relative object locations is calculated without taking different scene directions into account and compared with the two previously achieved prediction performances. In all cases, the prediction performance is calculated by determining the average of the probabilities with which the statistics of relative object locations would have predicted the labeled objects in our data set.

8.1 Prediction of object locations based on statistical knowledge

After labeling the scene direction in 1494 images, and estimating the 3D object locations of all objects in the images, we collected information about the 3D location of objects relative to other objects. The relative object location of one object to another object is defined by distances in two directions. One direction is the scene direction, and the other direction is orthogonal to it. The collected relative object locations were gathered in a histogram in which peaks at certain locations got visible (see Fig. 19).

We also collected information about distances from objects to building walls. We labeled intersection lines of the ground plane and the vertical plane in 355 images. After estimating the 3D locations of the intersection lines, we could calculate the distances from objects to the intersection lines. The distances were also gathered in histograms in which peaks got visible (see Fig. 22). Certain distances are more likely than others. The histograms were used to calculate probabilities for relative object locations and object distances to building walls. Using the obtained probabilities, we showed that the average prediction of object locations, given one reference object and the scene direction could be improved significantly compared to a prediction based on the assumption of uniformly distributed objects. The prediction improvement depends on the area around the reference object in which the other objects are expected to be uniformly distributed. If other cars are expected to be located at a maximum distance of ten meters around the reference car, the average prediction of cars in our data set could be improved by a factor of 2.5. The average prediction of people locations, given a reference person and assuming other people in a maximum distance of five meters around the reference person could be improved by a factor of 2.75. The exact results are de-
picted in Fig.30. The prediction of object locations given the location of a building wall could also be improved by using the probabilities obtained from the histograms. The results are illustrated in Fig.28.

8.1.1 Image categories for which the statistical knowledge is helpful

By back-projecting the likelihood of object locations into 2D image planes, we can see where the most likely object locations are. Fig.32 shows visualizations of car location likelihoods. Car location likelihoods are inferred from the location of a reference car with help of the statistical knowledge of relative car locations which we retrieved. The two cars with the blue bounding boxes in the two lower images are occluded. Every normal object detector would have difficulties detecting these cars. With the increased likelihood at their locations an object detector could be warned that there might be occluded cars and an "occluded cars routine" could be prompted. In Fig.33 an example for the prediction of locations of people is shown. Thanks to the inference of location probabilities from 3D locations the red "high-probability zones" for people locations get adapted very well to the scene. In the left image the reference person is far away. The red area is much smaller than in the image on the right where the reference person is closer to the camera. Additionally the orientation of the red "high-probability zone" is adapted to the scene direction.

Fig.34 shows images in which the likelihood of car locations was inferred from the distance to building walls.

In the images where the likelihood of car locations is visualized, it can be seen that increased likelihoods of car locations are at small distances in front of and behind the reference car. Cars are that close to each other when they are parked. We were gathering statistics from a training set containing mainly urban outdoor scenes. In such scenes there are typically a lot of cars parked at the side of the street. The average prediction performance of object locations in the images of our testing data set could significantly be improved. But there were also many images in our data set for which the statistics did not fit and the prediction performance for objects in these images was poor. Fig.35 illustrates how different the prediction performance for 100 images of our data set is. The probabilities are normalized so that the value is one for the image in which the cars could be predicted with the highest probability. Fig.36 shows images with cars and people for which the statistics do not fit. No objects occur at locations where the statistics are predicting they will. The two images on top show big streets with driving cars but no parking cars. The images in the middle show highways with cars and the two images at the bottom show single people on a market place and in a park. In general, our statistics will barely apply in highway scenes. Scenes in natural environments like city parks are less structured than man-made environments and it is harder to make any prediction of object locations. It can be expected that the overall prediction performance gets better the more limited our data set is to structured man-made urban outdoor scenes. To get a sense of how much the average prediction performance can be improved, we labeled vanishing points for a new set of 651 images without highway scenes or any natural environment scenes. The average prediction of relative car locations could be improved by 45% compared to the prediction of relative car locations in our data set which contains some highway scenes or scenes in city
Figure 32: Car location likelihoods inferred from the location of a reference car. The reference car in each image is framed with a bounding box with a thick bottom line. The two cars with the blue bounding boxes in the two lower images are occluded. Every normal object detector would have big difficulties to detect these cars. With the increased likelihood at their location an object detector could be warned that there might be an occluded car.

8.2 Object location prediction in images with automatically detected scene direction

The automatic detection of vanishing points which define the scene direction would be very useful for statistics gathering of object locations since manual labeling of vanishing points for a large set of training images is time-consuming. But it is particularly necessary for any kind of application which uses statistical knowledge of object locations. This is because the vanishing point of input images has to be known and any scene understanding or scene interpretation application should work without user input. In Chapter 7 we introduced several methods to automatically detect the scene direction. The line fit and voting method for scene direction detection (see Section 7.3 and Section 7.4 for details) are not amenable if the detected scene direction should be
Figure 33: People location likelihoods inferred from the location of a reference person. The reference person in each image is framed with a bounding box with a thick bottom line. Thanks to the inference of location probabilities from 3D locations the red “high-probability zones” for people locations get adapted very well to the scene. In the left image the reference person is far away. The red area is much smaller than in the image on the right where the reference person is closer to the camera. Additionally the orientation of the red “high-probability zone” is adapted to the scene direction.

Figure 34: Red indicates the likelihood of cars. The intersection line between ground plane and vertical building plane is indicated by the blue line.

used for the gathering of object location statistics. This is because they are already based on assumptions about the object locations. Apart from that, the mean error of scene directions which were estimated with the video compass algorithm (Section 7.1) was the smallest of the three discussed methods. We followed the same procedure to gather statistics of relative object locations as described in Chapter 5, in order to see if scene directions which were estimated with the video compass algorithm are accurate enough to be used for our purposes. However, we replaced the labeled vanishing points with automatically detected ones. Fig.37 shows the histogram which was obtained by calculating all relative object locations in 1063 training images with automatically detected scene direction. Before we used 1494 images to gather statistics, we use less images here, because video compass could not find a scene direction in every of the 1494 images. Compared to the histogram which we got with labeled scene directions,
the peaks in the histogram for relative car locations are less distinct and the object locations are more scattered (compare Fig.19 with Fig.37). The difference in the histograms for relative locations of people is smaller. The resulting probability density function is depicted in Fig.38.

Using leave-one-out cross-validation (as explained in Section 6.2) we calculate the average prediction performance of relative object locations in the 1063 images. Compared to the prediction based on statistics gathered with labeled scene directions, the average prediction performance in these 1063 images dropped by 33.5% for cars and by 7% for people. Anyway, the prediction is still better than a prediction based on the assumption of uniform distribution. If cars are assumed to be located at a distance smaller than ten meters around the reference car, the improvement is still 55%. The average prediction of people locations, given a reference person and assuming other people in a maximum distance of five meters around the reference person is still improved by a factor of 2.5.

If the statistics which were gathered with labeled scene directions are used, and only the scene direction in test images is labeled automatically, the prediction performance drops by 24% for cars and only 1% for people, compared to the case where scene directions are labeled in training and test images.

8.3 Object location prediction without reference direction

Finally, we want to compare the prediction based on statistics gathered by considering scene directions in images, as we did in the cases discussed in the two previous sections, with a prediction based on statistics gathered without considering scene directions. We already got a histogram of relative object locations without considering scene directions in the initial analyses chapter (Chapter 3). The relative locations were
obtained by calculating the distances between objects in $\vec{x}$ and $\vec{z}$ direction of the world coordinate system. Fig.12 shows the histogram. By using the principle of leave-one-out cross validation (as explained in Section 6.2) we calculated probability density functions (Fig.39) and calculated the average prediction of object locations in the 1494 images. For relative locations of cars, the prediction based on statistics gathered with labeled scene directions was 70% better than the prediction without considering any scene direction. In contrast to the histogram of relative car locations, there are distinct-

Figure 36: Example images out of our data set for which the statistics do not fit. There are no cars or people at locations where they are predicted with the statistical knowledge.
Figure 37: Statistics of relative object locations in images with automatically detected scene directions.

Figure 38: Probability density function for relative object locations retrieved with an automatically detected scene direction.

tive peaks in the histogram of relative people locations, even without considering scene directions (see Fig.12). The prediction based on the statistics gathered without considering scene directions was even better than the prediction based on statistics gathered with labeled scene directions in the images. It seems to be less important to know the scene directions for analyzing the locations of people, than it is for analyzing the location of cars.
Figure 39: Probability density function for relative object locations retrieved without considering scene directions.
9 Conclusion

9.1 Review

We introduced the statistical analysis of 3D object locations in images as a tool for enhancing the performance of object detectors. We showed how the object locations can be estimated. The used camera model was explained and the object location equations were derived. We also investigated if the estimated object locations are accurate enough for further analysis. The results of initial analyses of object locations led to the observation that it is crucial to know the orientation of the camera relative to the scene in order to analyze locations of cars in images. Finally, we finally gathered statistics of object locations relative to other objects by taking the knowledge of the camera orientation into account. We also gathered statistics of object distances to building walls. The statistics are discussed and evaluated. We also paid attention to the issue of estimating the orientation of the camera relative to the scene and its detection. We introduced several methods for the automatic detection of this orientation and compared the results.
9.2 Outlook

We showed that the statistical knowledge of object locations contains a great deal of information. By using statistics, a major improvement of the prediction of object locations could be achieved, compared to the prediction based on the assumption of uniformly distributed objects. It is worth mentioning that, as far as we know, no state of the art scene understanding or object recognition system which takes single view images as input exploits the knowledge of 3D object locations at all. Just the 2D locations in the 2D image plane are used instead.

We showed that the orientation of the camera relative to the scene has to be known to gather meaningful statistics of car locations. In test images, the statistics of 3D object locations relative to other objects, combined with the knowledge of the camera orientation, can be of great help to detect occluded objects. If the objects in the test scene are located at statistically likely locations, but the camera orientation with respect to the scene is in a way that one object occludes the other, the statistics will still predict an increased likelihood for the location of the occluded object. An object detector could be pointed at this location and be advised to take special care of occluded objects.

Existing algorithms to detect the camera orientation do not necessarily deliver accurate results for all images. Another uncertainty is the fact that it can never be expected that the statistics fit for object locations in all images, even if the data set is limited to very similar scenes. There will always be outliers. Therefore it is important to incorporate the statistical knowledge to object recognition algorithms in a way that there is no penalization if the statistics do not apply. The creation of false positives should be avoided. If this is ensured, the incorporation of statistics of 3D object locations can be a very helpful extension to existing object recognition systems.

To use statistical knowledge of 3D locations of people, the camera orientation is not absolutely necessary. This excludes the uncertainty about the detection of this orientation. On the other hand, without using the camera orientation, the statistics of 3D people locations are not particularly helpful to detect occluded people.

We also showed that the statistics of car distances to building walls can help to predict car locations. Due to the fact that the exact location of building walls is hard to detect, it will be easier to incorporate the statistical knowledge of relative object locations.

The data set we used to gather statistics consisted of urban outdoor scenes. We saw that 3D object locations vary significantly in different categories of scenes. There will only be regularities in statistics of 3D object locations if the statistics are gathered with images of one specific category; and the obtained statistics can only be used to predict object locations in images of the same category.
References


