Blurred Image Detection from a Humanoid Generated Video Sequence

MASTER THESIS

Margarita Camacho Clavijo
Department of Signal Theory and Communications
Technical University of Catalonia-Barcelona Tech.

October 2015
Blurred Image Detection from a Humanoid Generated Video Sequence

Master Thesis
European Master of Research on Information and Communication Technologies

Directors
David Varas / Ferran Marques

Department of Signal Theory and Communications
Technical University of Catalonia-Barcelona Tech.

October 2015
A mis padres,
Juan y Mari
Puedes llorar porque se ha ido
o puedes sonreír porque ha vivido;
puedes cerrar los ojos y rezar para que vuelva
o puedes abrirlos y ver todo lo que ha dejado;
tu corazón puede estar vacío porque no lo puedes ver
o puede estar lleno del amor que compartisteis.

Puedes llorar, cerrar tu mente, sentir el vacío
o dar la espalda o puedes hacer lo que a él le gustaría:
sonreír, abrir los ojos, amar y seguir.

Poema Anónimo Escocés

A la memoria de mis abuelos, Antonio y Juan.
Como dijo William Arthur Ward, hay que dar las gracias cuando nos sentimos agradecidos, te hace sentir bien, pero también a la persona destinataria.

Por eso, quiero aprovechar la oportunidad que me brindan estas líneas para agradecer a todas aquellas personas que han aportado su granito de arena para construir la persona que hoy día soy.

El puesto de honor lo ocupan mis padres, Mari y Juan. Ellos lo han dado todo por mí y mis hermanos, y ahora nos toca a nosotros recompensarlos con nuestro esfuerzo y nuestros logros. Es una de tantas formas que tenemos para agradecer vuestro amor y vuestro apoyo, aunque esta deuda es impagable.

Gracias a mis hermanos, Juan Antonio y Noelia, por compartir risas y llantos, éxitos y fracasos, alegrías y enfados,... Tiene que haber de todo, sino sería un aburrimiento.

Enrique, gracias por estar en los buenos momentos, pero muchas gracias por estar en los no tan buenos. Por tu amor, por tu paciencia, por tus chistes malos, por tus madrugones,... y por animarme a seguir mi camino aunque eso implicase vuelos de dos horas. Gracias.

Quisiera agradecer también, allá donde esté, a Vicent Caselles. Si no hubiera sido por él, hoy no estaría escribiendo estas palabras, ni hubiera conocido a Ferran Marques. Algunas veces me acuerdo Ferran, de cuando me decías: "esto no lo terminas nunca." Pues parece que esto está llegando a su fin. Gracias por todo.

Gracias a los miembros del Tribunal, por ser tan permisivos con tantos cambios de última hora, y a David, por guiarme en los inicios de este proyecto.

Cómo no agradecer a todos los compañeros que hicieron las clases más llevaderas, y que pasaron de ser compañeros a amigos. Gracias a todos.

Y gracias a toda mi familia, por sus ganas de que vuelva a casa, me
hacéis las despedidas un poco más difíciles, pero qué mal si no fuera así.
Abstract

No supliques cargas más leves,
sino hombros más fuertes

Philips Brooks

The 3D shape reconstruction from a humanoid generated video sequence project deal with the development of a strategy to estimate the geometry of an interesting object from a monocular video sequence acquired by a walking humanoid robot. In order to generate the 3D model of the object, firstly, blurred image must be eliminated. I have collaborated in this pre-processing step in which the final result is a set of images that contains the object without blur. First, the presence of blur is detected by the calculus of gradient magnitude. Second, a gradient histogram is displayed with twenty bins and the latest ten bins are added to classify the images. Finally, an image is considered as clear when the addition of the bins is bigger than 0.5% of the contour pixels.
# Contents

**Acknowledgments** ix

**Abstract** xi

1. **Introduction** 1
   1.1 Motivation ............................................. 2
   1.2 Organization of the Thesis ............................. 3

2. **Pattern Recognition and Classification** 5
   2.1 Introduction to Pattern Recognition ..................... 5
   2.2 Pattern Recognition Process ............................ 7
     2.2.1 Data Acquisition Phase ............................... 8
     2.2.2 Data Pre-processing Phase ........................... 8
     2.2.3 Classification Phase ................................. 8
   2.3 A Recognition Problem .................................... 9
     2.3.1 The Initial Approach .................................. 10
     2.3.2 Improving the Classifier ............................. 14
     2.3.3 Statistical Approach to Classification ............... 14
     2.3.4 Improving the Feature Extractor ...................... 15
     2.3.5 Redesigning the Classifier ........................... 15
   2.4 Problems in Pattern Recognition .......................... 16
   2.5 Pattern Recognition Techniques .......................... 18

3. **Blurred Image Detection** 21
   3.1 Approach to the Problem ................................. 22
   3.2 Data Acquisition Phase. Analysis of Physical Space ........ 23
   3.3 Pre-processing Phase ..................................... 24
     3.3.1 Pattern Space. Gradient and Histograms ............... 24
     3.3.2 Feature Extraction ................................... 29
   3.4 Design of the Classifier .................................. 34

4. **Results** 37

xiii
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1 Feature Analysis</td>
<td>37</td>
</tr>
<tr>
<td>5 Conclusions and Future Work</td>
<td>39</td>
</tr>
<tr>
<td>5.1 Conclusions</td>
<td>39</td>
</tr>
<tr>
<td>5.2 Future Work</td>
<td>40</td>
</tr>
<tr>
<td>A 3D Shape Reconstruction from a Humanoid Generated Video Sequence</td>
<td>41</td>
</tr>
<tr>
<td>Bibliography</td>
<td>51</td>
</tr>
</tbody>
</table>
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Fixed pattern recognition process</td>
<td>7</td>
</tr>
<tr>
<td>2.2</td>
<td>Adaptive pattern recognition process</td>
<td>8</td>
</tr>
<tr>
<td>2.3</td>
<td>Conceptual representation of a pattern recognition system</td>
<td>9</td>
</tr>
<tr>
<td>2.4</td>
<td>A hand-printed B reproduced with 6, 12, 24 48 lines and unlimited resolution</td>
<td>11</td>
</tr>
<tr>
<td>2.5</td>
<td>Samples of quantized hand-printed 8’s</td>
<td>12</td>
</tr>
<tr>
<td>2.6</td>
<td>Samples of quantized hand-printed B’s</td>
<td>13</td>
</tr>
<tr>
<td>2.7</td>
<td>Empirical cumulative distribution functions for the straightness ratio</td>
<td>14</td>
</tr>
<tr>
<td>2.8</td>
<td>Patterns as points in feature space</td>
<td>16</td>
</tr>
<tr>
<td>3.1</td>
<td>Clear and blur images</td>
<td>22</td>
</tr>
<tr>
<td>3.2</td>
<td>Bird-eye view of trajectory</td>
<td>23</td>
</tr>
<tr>
<td>3.3</td>
<td>Camera position of the selected video frames</td>
<td>24</td>
</tr>
<tr>
<td>3.4</td>
<td>Left: original images. Middle: gradient magnitude histogram.</td>
<td>25</td>
</tr>
<tr>
<td>3.5</td>
<td>Different choices for histograms</td>
<td>28</td>
</tr>
<tr>
<td>3.6</td>
<td>Ideal edge (left) and real edge (right)</td>
<td>29</td>
</tr>
<tr>
<td>3.7</td>
<td>Example of feature space</td>
<td>30</td>
</tr>
<tr>
<td>3.8</td>
<td>Gradient magnitude histogram of a clear image and a blur image</td>
<td>32</td>
</tr>
<tr>
<td>3.9</td>
<td>Histograms of the features</td>
<td>35</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

*No concibo ningún matemático que no lleve en su corazón una pizca de poeta*

Karl Weierstrass

Blur is one of the most usual image quality degradations. It can be caused by different factors, thus there are different kinds of blur.

The kinds of blur in an image include out-of-focus and motion blur. When an object is placed out of camera depth of field can cause out-of-focus blur. Motion blur is due to relative motion between the camera and the scene during image exposure time.

The camera keeps the shutter open during the exposure time. If there is some relative motion between the scene and the camera while the shutter is open, the pixels receive light from more than one directions. The amount of motion blur increases with increasing exposure time or increases the relative movement between the camera and the scene.

In this project, the images have motion blur, due to these images come from a video sequences filmed by a walking humanoid robot. These video sequences are going to make a 3D shape reconstruction of an interest object.

An analysis of the recorded sequence is applied in order to determine the suitability of each frame for the purposes of contour extraction.

An efficient method for visual tracking and off-line image restoration from a continuous sequence of motion blurred images is presented in [10]. Although, the visual tracking from image restoration is separated by introducing a cost function that does not involve de-blurring.

In other paper [1], another approach is proposed to detect a partial blur and segment by a local blur estimation. The image is partitioned into blocks and the local blur kernels of the blocks are estimated. After that, a
de-blurring technique is used to measure relative blur degrees of the local blur kernels. The output of de-blurring is metric to classify the blurred and non-blurred image blocks. This approach is evaluated for out-of-focus and motion blurred images.

The detection and classification methods of blurred images by regions are shown in [15] and [11]. These methods can automatically detect blurred image regions and classify the blur types (motion or defocus blur) without either image deblurring or kernel estimation. First, in [15] the blurred image regions are detected when singular value information for each image pixels is analysed. However, [11] develops several blur features which can be modelled on image color, gradient, and spectrum information. These features are used to classify blurred images robustly in a parameter training process.

A scheme to detect blurred images and classify them into several different categories is suggested in [9], as a [15] and [11]. In this case, the blur detector uses a support vector machine to estimate the blur extent of an image. Moreover, the blurred images are further classified into both locally or globally blurred images. For globally blurred images, their point spread functions are estimated and they are classified into camera shake or out-of-focus images. For locally blurred images, the blurred regions are looked for using a segmentation method. The point spread function estimation of the blurred region can sort out the images in depth of field or moving object.

This project is based on the paper [9] on account of recommending a method to detect camera shake, a type of global motion blur because of the camera movement with respect to the scene, what happen with the video sequences filmed by the robot. This method is based on the relationship between the image blurring and image gradient.

1.1 Motivation

The Department of Signal Theory and Communications from Universitat Politècnica de Catalunya, together with the Robotics and Advanced Manufacturing Group from Centro de Investigación y Estudios Avanzados del Instituto Politécnico Nacional from Mexico have worked in collaboration to develop a new method that is capable of estimating the 3D shape of an object from analysing of a video sequence generated by a humanoid robot.

In order to solve this problem, a space carving algorithm is used, which relies on both the precise estimation of the camera pose as well as the accurate extraction of the occluding boundaries of the object for each video frame.

The multi-view 3D reconstruction is resolved by the need for the robot to surround the object of interest. To generate the robot trajectory, a monocular vision-based locomotion control is proposed what drives the camera of the robot to face towards the center of the table where the object is lo-
In this way, the robot performs a circular trajectory to keep a constant
distance from the center of the table.

Once the video has been recorded, the object must be segmented. The
object segmentation is performed with a region-based particle filter approach
and each frame is assigned a consistency score. The video frames with the
highest scores that also observe a uniform distribution of the sampled object
views are finally selected for 3D shape recovery.

Two main situations can be found in which a region-based particle filter
may not correctly recover the shape of the object. Firstly, when a part
of the object is not present in the image. Secondly, when the blurring
effect degrades the quality of the object contours. To perform a correct
object segmentation, a pre-processing data is carried out which only images
containing the entire object without blur are selected. Finally, a subset of
these images are selected to reconstruct the object in accordance with their
final segmentation quality.

One of the tasks in the pre-processing phase is to select clear images. In
this way, the main aim is to develop a classifier that sort blurred images and
remove them from the video sequence. Since the image gradient is highly
related to the image blurring [4], the blur detector computes the magnitude
of this gradient to estimate the blur present in an image.

1.2 Organization of the Thesis

The recognition of patterns theory is used to develop the image classifying.
This project is organised into the following chapters.

- In the Chapter 2 a brief review of the most significant techniques in
  the recognition of patterns process and the design of a classifying is
  made.

- In the Chapter 3 the process is described to build the classifier. The
  physic system is analysed to get some images. After that, a new mea-
  sure space is produced by using a gradient magnitude and taking heed
  its histogram. Finally, the features from these histograms are ex-
  tracted.

- The results obtained when the images are classified regarding each
  feature are described in Chapter 4.

- Finally, in the Chapter 5 the main conclusions of this master final
  project are given and the future lines of research required to develop
  a more complete classifying are included.
Chapter 2

Pattern Recognition and Classification

Quien quiere encontrará un medio;
quien no, una excusa
Proverbio Árabe

This chapter presents the fundamental concept of pattern recognition and its system configuration. After explaining some of the basic pattern recognition terminology, an example of a pattern recognition system is presented. The chapter closes by listing possible problems in pattern recognition and main pattern recognition techniques.

2.1 Introduction to Pattern Recognition

When we talk about patterns, we refer to those objects or forms that we can perceive, through which you can interpret the world. In the early days of our lives, we learn to distinguish visual patterns of our father and our mother, auditory patterns of hearing music and a speech, sensory patterns of heat and cold, etc. As we grow, we are able to further define our pattern recognition, so that you can distinguish a Picasso painting of one of Goya, or distinguish one music style from another.

Therefore, a pattern can be defined as a quantitative or structural description of an object, process or some other entity of interest. It follows that a pattern class can be defined as a set of patterns that share some properties in common. Thanks to this, we would not have any difficulty to identify alphanumeric characters even when they are of different fonts and with different orientation and size [2].
For example, analyzing a printed page and recognize their characters, fixed rules are being used, based on experience. We may not know how to define these rules are exactly, but when it comes to differentiate the letter ‘o’ and ‘a’, the rules actually exist. It may be that over time, samples of the characters ‘o’ and ‘a’ have been repeatedly observed learning to distinguish his characteristics, so that consideration of this labeled sample has produced a decision rule. This decision rule is used to decide if a new character belongs to the ‘class a’ or ‘class o’.

Therefore, pattern recognition is a process of categorizing any sample of measured or observed data as a member of one of the several classes or categories.

Due to the fact that pattern recognition is a fundamental attribute of human beings and other living things, it has been taken for granted for long time. The intention of the theory of pattern recognition is to discover the intrinsic mechanism of their recognition, simulate it and put it into action with modern technology to benefit of human beings.

The pattern recognition applications is very wide. In theory, it can be applied to any situation in which the visual or audio information is needed in a classification process. These are some of the fields of application of pattern recognition techniques:

1. Automatic inspection of printed circuits, and printed character and handwriting recognition
2. Identification of people from fingerprints, hand shape and size, retinal scans, voice characteristics, typing patterns, and handwriting
3. Recognition of photographs
4. Automatic analysis of satellite pictures to determine the type and condition of agricultural crops, weather conditions, snow and water reserves, and mineral prospects
5. Automated analysis of medical images obtained from microscopes and scanners, magnetic resonance images, X-rays, . . .
6. Automatic inspection of parts on an assembly line
7. Human speech recognition by computers
8. Classification of seismic signals for oil and mineral exploration, and earthquake
9. Macroeconomic applications
10. Market research
11. Applications in the personnel department

In general, any system can be considered a good candidate for the application of pattern recognition techniques if, knowing a labeled sample of each class, is desired to assign to one of them a new data whose class membership is unknown. Keep in mind that, the application of these techniques can produce negative results if the labeled sample is very small, the discriminant characteristics have not been correctly selected or classes are not very discriminated.

2.2 Pattern Recognition Process

In any process of pattern recognition there are two fundamental aspects: the development or creation of a decision rule, i.e., the design of the classifier, and its use.

To design the classifier, the classes are defined and the labeled sample is represented adequately of each one of them, using measures or properties called features. The classifier design problem ends when a decision rule is obtained for which a new pattern, with unknown class, can be assigned to the class with lower risk in the assignment.

The recognition is performed by making use of the rule of decision previously developed. Its use can be fixed or adaptive. In the fixed process the classifier is designed based on the labeled sample of patterns and it does not undergo any modifications once classified the patterns of unknown class. However, in the case of adaptive recognition the classifier is modified during its use. Once the classifier is built, a new pattern is assigned to a class, according to the outcome of its application. Then, an error detector indicates if the classification was correct, in which case the classifier is not modified, or if it is necessary to modify if the result was wrong.

The phases in the design of a classifier is shown in Figure 2.3. The process can be divided into three phases: data acquisition, data pre-processing and
Chapter 2. Pattern Recognition and Classification

2.2.1 Data Acquisition Phase

In the data acquisition phase, the physical system where patterns are obtained is studied. In this phase, analog data from the physical world are gathered through a transducer and converted to digital format suitable for computer processing. In this stage, the physical variables are converted into a set of measured data. The measured data are then used as the input to the second phase.

2.2.2 Data Pre-processing Phase

The most representative characteristics are obtained from measured space acquired in the previous phase, using the feature extraction technique. The data pre-processing phase includes the process of feature extraction because the amount of data obtained in the data acquisition phase is tremendous. These data must be reduced to a manageable amount but still carry enough discriminatory information for identification. If the sample pattern is not controlled, it would take using cluster analysis techniques to find ‘natural classes’ in which these patterns are grouped.

2.2.3 Classification Phase

Finally, a classifier is designed using the labeled sample pattern in each class, considering only those features estimated in the previous phase. Once built the classifier, a new pattern of unknown class can be classified.
2.3 A Recognition Problem

The problem of designing or programming machines to recognize patterns is one of the most interesting topics in the computer and information sciences. It appears in a variety of topics, and the problems range from engineering economics to the philosophy of science.

A good idea to understand the process and the outlines of a pattern classifier is through an example, which can be analyzed key stages of pattern recognition, as well as some of the techniques used.

One of the classic examples of pattern recognition is the recognition of printed characters: alphanumerics, punctuation marks, mathematical symbols, etc. For this class of the problems, only a limited number of character types must be differentiated, and each character comes from an ideal pattern or template. This means that there is a conceptually easy way of classifying an unknown character: one simply compares it with an ideal version of each of the possible characters and sees which one it most similar.

The template matching approach becomes less appropriate due to the patterns of this type are subject to significant distortions and variability. It would be difficult to classify hand-printed characters this way without using a very large number of templates, and for the recognition of cursive script this approach is useless. In such cases, some characteristics are looked for distinguishing one type of pattern from all others. For example, one could try to differentiate an 0 from a Q by looking for the presence of a tail, or to differentiate a B from an 8 by measuring the straightness of the left side.

The use of features reduces the complexity of the problem by extracting from the original data just the information needed for classification. Furthermore, if the results of feature measurement tend to be the same for patterns in the same class, but are significantly different for patterns in different classes, then that template matching can be used at the feature level with success.

This general approach to pattern recognition, i.e., the extraction of significant features followed by classification on the basis of the values of the
features, has been applied to many pattern recognition tasks, such as speech and speaker recognition, fingerprint recognition or clinical diagnostics.

Two problems must be broached for each application: the design of a feature extractor and the design of a pattern classifier. The design of a feature extractor is usually problem-dependent because all the information available about the problem must be used to select those features that seem to be most significant to differ one pattern from another. Since building a set of valuable characteristics is highly dependent on the problem, there is not much theory about design of the feature extractor.

After designing the feature extractor, the next task is to design the classifier. Unlike in the previous task, a variety of procedures are available for designing the classifier. These procedures differ in the assumptions that are made about the behavior of the features, such as assuming that the features are statistically independent. For this reason, the methods for designing the classifier are often much more independent of the special characteristics of the particular problem.

But the pattern recognition not only consists of feature extraction followed by pattern classification. For example, if common objects in a photograph want to be recognized, first, the way to locate and isolate the individual items to be recognized must be resolved, and this can be a hard task.

Another important point is that not all pattern recognition problems are classification problems. It may be interesting to reject a pattern if it is quite unlike any of the patterns. In other cases, it is also important for the classifier to provide a confidence measure with each decision. Therefore, it must be taken into account that the pattern recognition includes other aspects in addition to the feature extraction and classification.

2.3.1 The Initial Approach

Many of the ideas that have been discussing can be made clearer by considering a specific example. An example described in [13] is presented. Given a labeled sample of hand-printed B’s and hand-printed 8’s alphanumeric characters, it wants to design a decision rule under which a new character can be identified as a B or 8.

As previously stated, the pattern recognition process begins with an analysis of the physical system that patterns are extracted. In this case it is the man and the different instruments used to write. It is considering the existence of a mechanism for isolating the characters and digitizing the input data for any degree of precision desired, and it is considering ways of using this quantized data to recognize the character.

The next step is to build the measured space, which involves quantifying the characters B and 8. It is important to properly determine the level of
quantification because the results will depend on it. The Figure 2.4 shows the results when a typical hand-printed $B$ is reproduced with different degrees of resolution. In this case, a 24-line resolution is chosen because it is sufficient to tell that a character is a $B$ and not an $8$.

Then, a feature discriminant must be selected to estimate if a quantized character is a $B$ or an $8$. One characteristic that should differentiate $B$’s from $8$’s is the fact that $B$’s have a straight stroke down their left side. Thus, a plausible procedure would be to measure, for example, the straightness of that side. If it is sufficiently straight, the figure is called a $B$; if not, it is called an $8$. The straightness of a line is measured by the ratio of the distance between its endpoints to its arc length. This will yield a number $x$ between zero and one, the larger the number the straighter the line. In this way, a
perfect $B$ has a ratio equal to 1, and a perfect $8$ has a ratio less than 1 and greater than 0. If it had analyzed more features intuitively discriminating, it would have been necessary to use a feature selection technique to reduce the dimensionality.

In the last phase of the process, a classifier is designed. The selected feature, called $X$, measures the straightness of the left side of the character. Now, to decide how the line is sufficiently straight, it is needed to know feature values of $X$ for $B$’s and $8$’s. To do that, the feature $X$ is calculated for each $B$’s and $8$’s of the labeled samples shown in Figures 2.5 and 2.6 and an average value is obtained for each class, $X_B$ and $X_8$. Thus, it is reasonable to select some intermediate value $X_0$ between $X_B$ and $X_8$ as decision boundary or threshold, and use the following rule: If $X > X_0$, call the character a $B$, otherwise call it an $8$.

Unfortunately, the results obtained by applying this rule are rather di-
sappoing. About 23 percent of the characters are misclassified, with the majority of the errors being 8’s that are called B’s. The possible reasons for this result are:

1. Inadequate resolution and a better representation of the data is called for

2. The feature does not have the appropriate degree of discrimination

3. The selection of the particular measure of straightness was certainly arbitrary, and a more careful choice might yield improved results

4. The threshold $X_0$ has not been chosen correctly
2.3.2 Improving the Classifier

A wrong result of the classifier can be caused by any of the above causes. It can test if better results can be obtained changing the selection method. The empirical cumulative distribution function $F_e(X)$, derived from the labeled samples $2.5$ and $2.6$, can be analyzed and the threshold for which the smallest possible number of characters erroneously assigns, can be chosen. This has been done in $2.7$, where $F_e(X|B)$ is the empirical cumulative distribution function for the $B$’s, and $F_e(X|B)$ is the empirical cumulative distribution function for $8$’s.

Other ways to get the decision rule are for example, to assign the new pattern to the nearest pattern class, or represent the features of both classes in a histogram, and see which threshold is the optimum. This method would be equivalent to the analysis of the distribution function.

2.3.3 Statistical Approach to Classification

When samples can not be perfectly classified using the feature available, the goal may be to estimate the probability of belonging to each class. Given the feature, a sample can be classified in the class that have greater probability of belonging, or if we consider the costs of error, can be classified in the class with lower cost of error.

To do this, the conditional probabilities are estimated a priori from the data, and applying Bayes theorem, the posterior probability is estimated. According to this statistical approach, the optimal decision rule can be considered, in order to minimize the probability of error, is that which assigns
the X pattern to the class for which the probability a posteriori is greater.

To do this, the conditional densities, \( p(X|B) \) and \( p(X|8) \), and a priori probabilities, \( P(B) \) and \( P(8) \), are estimated from the data, and the a posteriori probabilities are computed using Bayes’ theorem,

\[
P(B|X) = \frac{p(X|B)P(B)}{p(X)}
\]

\[
P(8|X) = \frac{p(X|8)P(8)}{p(X)}
\]

where

\[
p(X) = p(X|B)P(B) + p(X|8)P(8)
\] (2.1)

According to this statistical approach, the optimal decision rule, in the sense that minimizes the probability of error, is one that assigns the X pattern to the class for which the probability a posteriori is greater.

### 2.3.4 Improving the Feature Extractor

If the effectiveness of the classification has not been improved, the decision rule is changed. Whereas the degree of accuracy of the measure space is enough, one of the possibilities improvement of the classifier is to research for new features.

For this particular problem, another feature that may be very significant is the ratio of the maximum width of the top half of the character to the maximum width of the bottom half, since there is a tendency to manually write the character \( B \) with larger width at the bottom, while the \( 8 \) character tends to be wider at the top.

Thus, according to this selection of features, each of the patterns which compose the labeled sample is defined by the vector \( X = (X_1, X_2) \) that reflects the values of the two measurements, the straightness ratio \( X_1 \) and the top-to-bottom ratio \( X_2 \). Therefore, each pattern can be represented as a feature vector \( X \) in a two-dimensional feature space \( 2.8 \).

Although part of the data are mixed, many of the points that represent the characters \( B \) are separated from the \( 8 \) characters. Now the problem is to design a classifier that take advantage from it.

### 2.3.5 Redesigning the Classifier

According to previous ideas, a new pattern will be assigned to class \( B \) or \( 8 \) whose a posteriori probability \( P(B|X) \) or \( P(8|X) \) is greater, with the addition that now, \( X \) is two-dimensional and therefore, the density functions are joint distributions.
Getting the decision rule, the pattern space is divided in two regions, separated by a decision boundary. If the classes are separated, the feature space can be divided in two regions by a straight line. If classes are not separated, we can continue to use the straight line if the error committed is acceptably low, or we can use a more complex decision limit, but in this case, the problem of overfitting may appear.

2.4 Problems in Pattern Recognition

In the particular application of pattern recognition techniques usually appear the following problems which, not having an adequate solution, can derail an application which in the first moment could be viable. These problems include the choice of distance or measurement of similarity of the patterns, normalization and dimensionality of the features and the estimation and treatment of complex distributions of variables [3].

Distance or Similarity Measure

The most obvious way to measure the similarity or the difference between two patterns or groups of patterns is to obtain the distance between them. If
the selected distance is a good measure of divergence, it can be considered that two patterns are more similar to the lower the value of the selected distance measurement.

Since the measurement of similarity or divergence is one of the most important topics in the pattern recognition techniques, it is essential a suitable choice of the distance. According to take one type of distance or another, the results of the problem can be completely different.

**Normalization**

One of the fundamental aspects to design a measure of similarity is the choice of the unit of measure and weighting of features. Thus, the weight can be measured in grams and kilograms, the height in meters and centimeters, etc. The result of applying for example, the Euclidean distance with one type or another scale measures is completely different. On the other hand, if the characteristics are not homogenized and weighted, it may be that a large deviation in one, mask the deviation of the other.

The measure of similarity or divergence between two patterns should not be influenced by the specific type of scale selected in the features. Groupings obtained using the Euclidean distance are invariant to translations or rotations of the values of the features, but instead are sensitive to linear transformations or, in general, sensitive to changes in scale of the features.

The most common ways to make the similarity measure to be invariant to changes of scale are:

1. Normalize the value $X_{f_j}$ of the feature $f$ of the pattern $j$ using the range of the features $a_f$, resulting $X_{f_j}/a_f$, $\forall f \in F$ for all pattern of labeled sample.

2. Normalize the value $X_{f_j}$ of the feature $f$ of the pattern $j$ using the variance of the features $\sigma_f$, resulting $X_{f_j}/\sigma_f$, $\forall f \in F$ for all pattern of labeled sample.

The above types of normalization are not sometimes the most appropriate. Considering the variance normalization, it prevents certain features can dominate over others because of the greater range of numeric values. With this normalization is achieved that dispersion of the values is only due to the own divergence of features. But it would have to study very carefully this normalization, since the divergence may be due to the presence of subclasses in the group of patterns.

**Dimensionality**

Given the difficulty that sometimes exists for obtaining reliable samples, it is not recommended to work with all available information of the pattern,
but only with the relevant information, since in general, for a large number of features it will be necessary to have a very high sample size. Therefore, it is necessary to use methods of features selection to reduce to a minimum the number of features that are used.

There is no general rule, strictly scientific, to estimate in each case the optimal size of a sample of patterns, however, there are numerous studies in the literature trying to estimate the minimum proportion of the number \( n_i \) of labeled patterns of each class in relation to the number \( F \) of features. In most cases applied, designed classifier provides very poor results if the ratio \( \frac{n_i}{F} \) (or \( \frac{n_i}{\frac{F+3}{2}} \) in the case of quadratic discrimination) is less than 5.

But it is not only interesting that ratio \( \frac{n_i}{F} \) is as high up as possible to achieve a greater representation of the sample and, therefore, that the relevant features involved throughout its range in the obtaining of the classifier, but also to avoid variability to be masked by the features that are not representative.

Therefore, as a previous step to design a classifier, we must select the relevant features for the case study object and choose a representative sample of the troubled area, not expecting great results if the previous value is less than 5.

Complex Distributions

The labeled sample of each class is not always distributed in a compact form, but that the sample may be multi-modal. For example, in a sample of the class of the first letter of the alphabet can exist patterns of subclass \( A \) and patterns of subclass \( a \). At other times, the samples form leaves instead of conglomerates, or form concentric circles with different radius, etc. As you grow the \( F \) number of features, increase the complexity of the distributions.

A large majority of the pattern recognition techniques is performed under the hypothesis that features follow the one-dimensional or multidimensional normal distribution, when in many cases the distribution is exponential, binomial, Poisson, etc. Due to the complexity of the treatment with high dimensional patterns in these cases, it is assumed that the features are independent.

2.5 Pattern Recognition Techniques

As mentioned previously, there is a tendency to consider that the design of a classifier is the only goal of the techniques knows as pattern recognition. Obtaining a decision rule is very important to assign a new pattern to a particular class, but it is not the only goal. It could indicate that any goal to be achieved with multidimensional patterns, is part of the application and research of pattern recognition techniques.
So, one of the big issues previously mentioned above consists of a suitable selection of features so that they collect the relevant information from the problem. There are basically two types of techniques for this purpose. A type of technique is to select a certain number of characteristics, according to their degree of discrimination. The other type is estimating new features through the old linear transformations, so they pick up the inherent discrimination of classes.

When a sample available is not labeled and even the classes of the problem are unknown, it may be useful to try to get the groups or natural classes in which, according to their similarity, are grouped the sample patterns. The set of methods and techniques that describe and locate these groups are called cluster analysis.

The classifier design methods can be grouped in the following way [3]:

1. Known probabilistic structure. The a priori and conditional probabilities are known, as well as the statistics of these distributions.

2. Unknown probabilistic structure.

   (a) Supervised learning methods. These methods are based on that provides a labeled sample of each class, as well as that the a priori probability is known, but if the a priori probability is unknown, it could be estimated based on the labeled sample.

      i. Parametric methods. In these methods, the conditional probability is also known, but on the other hand, the statisticians are unknown. The design of the classifier is carried out based on the maximum likelihood methods and Bayes estimation method.

      ii. Non parametric methods. In these methods the conditional probability is unknown. The classifier design is based on estimating this probability or replaced by orthonormal and orthogonal transformations.

      iii. Direct methods. These methods directly estimate the a posteriori probability that a pattern belongs to a class. The objective is to obtain the appropriate discriminant functions, apart from the conditional distribution. Some of these methods are: nearest-neighbour rule, fixed increment procedure and the gradient method.

      iv. Discriminant analysis. This method is based on the same assumptions than direct methods, but it has the advantage that, in the creation of the linear or quadratic discriminant functions, they optimize a specific criteria imposed by the user.
v. Sequential analysis. As a variant of the parametric and non-parametric methods, sequential pattern recognition analyzes one to one characteristics pattern, so that the average number of tested features that are required for the allocation of the pattern is less than the number that the above methods require.

(b) Unsupervised learning methods. These methods deal with cases in which the sample is not controlled. The difference with cluster analysis techniques is that classes to discriminate are known as well as the conditional probability function and a priori probability is unknown. The classifier is designed iteratively, since neither the statisticians are known, and to estimate them, it is necessary to estimate the class of the patterns that form the sample.
Chapter 3

Blurred Image Detection

Cada vez que cometo un error, me parece descubrir una verdad que aún no conocía

M. Maeterlinck

The example that has been considered in the previous chapter contains common element to many pattern recognition problems. The object or event to be classified is converted and represented in some way, for example as a $24 \times 24$ matrix. To work easily with this representation, the characteristics that best represent the objects are removed, and the value of these characteristics are sent to a classifier, which assigns the input object one of the possible categories of permanence.

The problem to be solved in this paper is to detect the blurred images of some video footage recorded by a humanoid robot for later removal. To do this, a classifier or detector of blurred images is built, following the pattern recognition process presented in the previous chapter.

In the first phase of the process, the data acquisition phase, the physical system in which the images are obtained is analyzed. Then a pre-processing of the data is performed. In this second phase, a measure space is created using the gradient magnitude histogram. This way, each image is represented by its gradient histogram. Afterwards, the feature of these histograms that best discriminate a blur image of a clear image are obtained. In the last phase, using the characteristics obtained in the previous stage, the classifier is designed.
Chapter 3. Blurred Image Detection

3.1 Approach to the Problem

In the most noise studies, occlusions, or illumination, made for applications such as surveillance, robotics, medical imaging, etc., it is very common to consider that the images are clear.

Blur is one of the conventional image quality degradation which is caused by various factors: instability of the camera, night scene, the movement of the object, object out-of-focus, etc. This effect can be intentional, to enhance the expressiveness of the image, but if it does not, the image quality decreases.

It might be said that there are two types of blur: out-of-focus, since the object is located outside the depth of field of the camera and motion blur, due to the relative motion between the camera and the scene during the exposure time. This latter type of blur is present in most of the real videos due to the low camera speed and quick movements of the object, destroying fundamental characteristics such as the edges of the objects contained in the image.

In this case, the blur is due to robot camera movement. The segmentation quality decreases when there is a lack of definition, producing incorrect contours and mixing the object with the pixels of the background of the scene 3.1. For this reason, it is necessary to perform a previous processing which removes the blurred images.

Since the image blurring is related to the image gradient [4], this blur detector will use the image gradient to determine if it is blurred or not. The problem is solved from the point of view of pattern recognition, where the patterns, in this case, will be magnitude gradient histograms. Certain features are extracted from these histograms to create a classifier, which determine whether an image is blurry or clear.
3.2 Data Acquisition Phase. Analysis of Physical Space

The first step in designing the classifier of images consists of the acquisition of the data. At this first phase, a study of the physical system is performed. The physical variables are converted into a digital format suitable for computer processing. So, physical variables are analyzed and transformed into a set of measures able to represent each of the images of the video sequence, without losing information about its blurriness.

Since the purpose of this work is to obtain a set of clear images of the video sequence recorded by a humanoid robot, with enough quality to achieve a good 3D reconstruction of an object, it is necessary to get multiple views of the object. To achieve this, the robot has to surround the object, recording the position and orientation of the camera and control its trajectory.

To generate the robot trajectory, a monocular vision-based locomotion control is used, that keeps the robot along a circumference of known radius while the camera orientation is directed to the center of the circle, where the object is located. The position and orientation of the camera for each video frame are estimated under a monocular vision-based framework.

To create a labeled sample of each class (clear and blur), 16 video sequences of different objects were recorded using the proposed locomotion control. The radius of the circular trajectory was set 0.6 m with a 3° angular step. For all sequences, the video acquisition rate was 9 frames per seconds with a resolution of 640 × 480 pixels, obtaining a total of $N = 2299$.
3.3 Pre-processing Phase

Due to the large amount of data obtained in the acquisition phase, data are processed to reduce the amount of data to a set more manageable but, at the same time, it must carry enough discriminatory information to identify whether an image is clear or blur.

3.3.1 Pattern Space. Gradient and Histograms

In the previous chapter, an example was presented, where the characters B and 8 were quantified with a resolution of 24 lines, enough to tell whether a particular character was B or it was an 8. So, as in this example, once the annotated sample of images with and without blur has been obtained, a new space of measures is estimated in which each image is represented by a new data, so the dimensions of the space is reduced, without losing information about the clearness.

Given that the main characteristic of a blurred image is the presence of blurred edges, the gradient is used to obtain the new measure space, since it provides information on the edges or occlusion boundaries.

Clear images have edges well defined so that, intuitively, they contain large regions where the intensity of the gradient remains constant or has soft
3.3. Pre-processing Phase

Figure 3.4: Left: original images. Middle: gradient magnitude histogram. Right: gradient direction histogram [4]

changes, interrupted by occasional large changes at the edges. As a result of those abrupt changes, the gradient achieves high values. However, in the blur images, the edges are softened, which means a more gradual change in the gradient and therefore high gradient values are not achieved.

So, the Figure 3.4 shows that, if the gradient magnitude is represented by a histogram, it is observed that, for a clear image is massive on small values but large values are also reached with some significant probability. In contrast, the gradient magnitude distribution of a blurred image is almost empty on the large values and only exists for small values of the gradient. If the gradient direction histograms are compared, for clear images all directions have almost the same probability, however, for blurred images there are some values higher than others [4]. According to this difference of the gradient distribution between a clear image and a blurred image, we can discriminate them by analyzing the gradient histogram. In particular, the classifier has been developed using only the gradient magnitude histogram.

Gradient

An image may be defined as a two-dimensional function $f(x, y)$, where $x$ and $y$ are spatial coordinates, and the amplitude of $f$ at any pair of coordinates $(x, y)$ is called the intensity or gray level of the image at that point [6]. Considering this, the image gradient of an image in pixel $(x, y)$ can be defined mathematically as the two-dimensional vector whose components are given by the partial derivatives of the function $f$ at the point, and is denoted in the following way:
Chapter 3. Blurred Image Detection

\[ \nabla f(x, y) = \left[ \frac{\partial f(x, y)}{\partial x}, \frac{\partial f(x, y)}{\partial y} \right] \]  

(3.1)

If we define

\[ g_x = \frac{\partial f(x, y)}{\partial x} \]

and

\[ g_y = \frac{\partial f(x, y)}{\partial y} \]

the vector magnitude is given by

\[ |\nabla f| = \text{mag}(\nabla f) = \sqrt{g_x^2 + g_y^2} \]

(3.2)

and the direction is given

\[ \theta = \arctan\left(\frac{g_y}{g_x}\right) \]

which corresponds to the direction of maximum variation of \( f \).

Therefore, the gradient calculation is based on obtaining the partial derivatives for each pixel. In the discrete two-dimensional case, the different approaches of the gradient operator are based on differences between the gray levels of the image, and it can be approximated through finite differences equations, one for each direction \((x, y)\).

\[ \frac{\partial f(x, y)}{\partial x} = \lim_{\Delta x \to 0} \frac{f(x + \Delta x, y) - f(x, y)}{\Delta x} \approx \frac{f(x + \Delta x, y) - f(x, y)}{\Delta x} \]

\[ \frac{\partial f(x, y)}{\partial y} = \lim_{\Delta y \to 0} \frac{f(x, y + \Delta y) - f(x, y)}{\Delta y} \approx \frac{f(x, y + \Delta y) - f(x, y)}{\Delta y} \]

\[ \frac{\partial f(x, y)}{\partial x} = \frac{f(x + \Delta x, y) - f(x, y)}{2\Delta x} \approx \frac{f(x + \Delta x, y) - f(x, y)}{2\Delta x} \]

Each of these equations can be implemented as a filter linear space invariant, in the following way

\[ \left. \frac{\partial f(x, y)}{\partial x} \right|_{x=m,y=n} \approx f(m+1, n) - f(m, n) \]

\[ \left. \frac{\partial f(x, y)}{\partial x} \right|_{x=m,y=n} \approx f(m, n) - f(m-1, n) \]

\[ \left. \frac{\partial f(x, y)}{\partial x} \right|_{x=m,y=n} \approx \frac{f(m+1, n) - f(m-1, n)}{2} \]

There are several implementations of filters that approximate the gradient in a given direction. The most commonly used operators are Roberts,
Prewitt and Sobel. In this case, the operator of Sobel has been used, which is formulated with the following convolution masks.

\[
    h_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \quad h_y = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}
\]

With this filter, an estimation of the horizontal and vertical gradient is obtained,

\[
    g_x[m,n] = f[m,n] * h_x[m,n] \\
    g_y[m,n] = f[m,n] * h_y[m,n]
\]

so,

\[
    \nabla f[m,n] = \begin{pmatrix} g_x[m,n] \\ g_y[m,n] \end{pmatrix}
\]

and the gradient magnitude and the gradient phase is calculated by

\[
    |\nabla f[m,n]| = \sqrt{g_x^2[m,n] + g_y^2[m,n]}, \quad \theta_{\nabla f}[m,n] = \arctan \left( \frac{g_y[m,n]}{g_x[m,n]} \right)
\]

Then, the gradient and the magnitude is calculated to each of the images that compose the labeled sample, using the Sobel operator. Next, these data are represented in a histogram, so there will be a histogram for each image sample.

Histogram

The histogram is a useful mathematical tool to graphically describe the data distribution. It is an easy way to obtain a density function approximation. To build it, the range of data values is divided into a finite number of intervals, called bins. Over each interval, a vertical rectangle is drawn, whose area is proportional to the number of data points falling into that interval. If a sample is directly on a border between intervals, by convention, it is assigned to the right [8].

To choose the number and location of the histogram intervals there is no definitive theoretical guide. The Figure 3.5 show possible choices for histograms to describe 50 random numbers chosen from a normal density shown in the figure 3.5a. If a small number of wide intervals is used as in the figure 3.5c, the number of samples that are in each interval will be relatively large, so the height of rectangles and therefore the area of the interval can be estimated quite accurately. However, the approximate density will be flat over large regions and any narrow fluctuation in the true distribution will be lost [8].
On the other hand, if a relatively large number of intervals is used, the fine structure of true density can be preserved, but when too many intervals are used as in Figure 3.5d, the heights of the bins will not correspond with the true distribution. Therefore, overfitting must be prevented. As an extreme example, if the number of intervals were much greater than the number of samples, most intervals would not contain samples and the rest would probably contain only one sample each. In this case, the histogram is reduced to a number of very narrow rectangles, almost one for each sample. This would not produce a useful estimate of the probability density function, it would be more similar to the representation of the actual data itself.

So, a compromise between too many and too few intervals must be achieved. There are several rules to choose the number of intervals. One of them is the rule of Sturges [14], which considers that the optimal number of intervals can be estimated from the formula

\[ b = 1 + 3.322 \log N \]  

(3.3)

where \( N \) is the total number of samples.

To process the images, they are converted to grayscale, so, the image function takes values in the range \([0, 256]\). Thus, when the gradient is calcu-
3.3. Pre-processing Phase

lated using the Sobel operator, the gradient magnitude can take values into the interval $[0, 1020]$. The maximum value is achieved if there is an ideal edge (see Figure 3.6), but due to the camera optics, sampling, lighting and other imperfections, edge points are not ideal. For this reason and based on the values of the gradient magnitude, which are obtained from the labeled sample images, the interval $[0, 700]$ is taken as the range of values of the gradient magnitude for its representation in the histogram.

After determining the range of values, the number of intervals of the histograms must be chosen. The image resolution of the labeled sample is $640 \times 480$ pixels, so there is 307200 values of the gradient magnitude for each image. Using Sturges formula described in the equation 3.3, the optimal number of intervals, also called bins, for the histogram must be 20. In this way, all histograms have 20 intervals equally distributed in the value range $[0, 700]$. The vector $C = [c_1, ..., c_{20}]$ is considered, where each element $c_i$ is the average value of each bin, also called classmark.

Since what matters is the information about contour pixels, non-null values of the gradient magnitude are represented in histograms, $|\nabla f(x, y)| > 0$. And since not all images have the same number of edge pixels, the relative frequencies are represented on the vertical axis of the histogram, i.e., the number of pixels whose gradient magnitude value is located in a particular bin divided by the total number of pixels with the non-null gradient magnitude. Thus, $\sum_{k=1}^{20} r_k = 1$ where $r_k$ is the relative frequency en el bin $k$.

In this way, each one of the images that compose the labeled sample has moved to be considered as a matrix of dimension $640 \times 480$, into being represented by a vector of dimension 20, $H_n = [h_n^1, ..., h_n^{20}]$, where the element $h_n^i$ represents the relative frequency in the bin $i$ for the image $I_n$, with $i = 1, ..., 20$. These vectors $H_n$, $n = 1, ..., N$, ($N$ is the cardinality of the labeled sample), form the new pattern space.

3.3.2 Feature Extraction

Once the space of patterns is well defined, the next step is to select the feature that best discriminate a clear image of a blur image. Features ex-
traction consists in the identification of inherent characteristics found in the object or pattern. These characteristics, or features, are used to describe the object or attributes of the object [2]. In this case, the features are used to determine the presence or absence of blurring.

It is known that when it comes an image is processed through the human vision system, the system does not visualize the image or an object, pixel by pixel. The human vision system extracts some key information created through grouping related pixels to form features. Although the features are fewer than the original data, they contain enough discrimination information for identification [2].

The correct selection of the features is essential. A set of features can be very effective for one application, but may not have use for another. Through a comprehensive study of the pattern, a preliminary set of features is obtained, from which a final selection is made to evaluate its effectiveness in the classification.

A pattern $\mathbf{x}$ can be represented as $\mathbf{x} = [x_1, x_2, \ldots, x_k]^T$, where the subscript $k$ represents the dimension of the patterns space (in our case is $k = 20$). If $k < 3$, the space can be represented graphically (see Figure 3.7). Pattern space $\mathbf{X}$ can be described by a vector of $m$ pattern vectors such that,

$$
\mathbf{X} = 
\begin{bmatrix}
    \mathbf{x}_1^T \\
    \mathbf{x}_2^T \\
    \vdots \\
    \mathbf{x}_m^T
\end{bmatrix} = 
\begin{bmatrix}
    x_{11} & x_{12} & \cdots & x_{1k} \\
    x_{21} & x_{22} & \cdots & x_{2k} \\
    \vdots & \vdots & \ddots & \vdots \\
    x_{m1} & x_{m2} & \cdots & x_{mk}
\end{bmatrix}
$$

Figure 3.7: Example of feature space [2]
where $x_i^T = (x_{i1}, x_{i2}, \ldots, x_{ik})$, $i = 1, 2, \ldots, m$ represent pattern vectors.

The purpose of the features extraction is to reduce the dimensionality without losing relevant information. It converts the original data into a form suitable, feature vectors, for use as input to the decision process for the classifier. Obviously, the feature vectors represented by $x_i^T = (x_{i1}, x_{i2}, \ldots, x_{ir})$, $i = 1, 2, \ldots, m$ are in a smaller dimension, i.e., $r < k$.

If the features have been well chosen, when the feature vectors are represented in the feature space, it will be observed how patterns form groups. Each group represents a class, where each one of the class points represents a pattern. The feature vector are inputs to the classification process.

Patterns that must be classified are gradient magnitude histograms of the images. These patterns are given by the vector:

$$H_n = [h_{n1}, \ldots, h_{n20}], \quad n = 1, \ldots, N$$

where $N$ is a total number of images, and $h_{ni}$ is the relative frequency in the bin $i$.

Analyzing the histograms of a blurred image and a clear image, certain differences are observed. These differences are used to find the features that best distinguishes these histograms.

**Median**

Firstly, in the Figure 3.8 it is observed that bars of histograms of a clear image are much higher than histogram bars of a blurred image. Therefore, a possible distinguishing feature with respect to the clear image could be the median heights of the bins. In this way, is expected that a clear image have a median higher than the median of the blurred image.

The median of a set of sorted data $x_1, x_2, \ldots, x_n$ is the value which take up the central position. If $n$ is odd, the median is the value at the position $(n+1)/2$, i.e., $M_e = x_{n+2}/2$. If $n$ is even, the median is the arithmetic mean of the two middle values, $M_e = (x_{n/2} + x_{n/2+1})/2$.

Given the nature of the data, the vector $H_n = [h_{n1}, \ldots, h_{n20}]$ is a vector of sorted data, and as the number of bins used is an even number, the median is the arithmetic mean of the $h_{10}$ and $h_{11}$ values. Therefore, this first feature is denoted by $f_1$ and is defined as

$$f_1(H_n) = M_e(H_n), \quad \text{with} \quad M_e(H_n) = \frac{h_{n10} + h_{n11}}{2} \quad (3.4)$$

Thus, if $H_n$ corresponds to the histogram of a clear image and $H_b$ to a blurred image, it is expected that

$$f_1(H_n) > f_1(H_b) \quad (3.5)$$

It is noted that, the high values of the gradient magnitude are not reached in a blur image, but that most of the data are displaced to small values,
whereas for clear images, the value range is wider. Then, another feature that could be used to differentiate the histograms is the central point of the values range that achieves each histogram. Thus, the value of this feature will be higher in a clear image than in a blur image.

To calculate this feature, consider the vector \( C = [c_1, \ldots, c_{20}] \) where \( c_i \) is the center of bin \( i \). Since all histograms have been built on the same range of values and with the same bins, in the case of a blurred image, the latest bins can be null. Then, the non-zero values of the vector \( H_n \) are considered, and a new vector \( H'_n = [h'_n^1, \ldots, h'_n^r] \) is obtained, where \( r \leq 20 \), with their respective centers vector \( C' = [c_1, \ldots, c_r] \). If \( r \) is odd, the center point of the range of values is \( \frac{c_r + 1}{2} \), and if \( r \) is even, the value of this new feature is \( \frac{c_r/2 + c_r/2+1}{2} \). Therefore, this feature is defined by the median of the \( C' \) vector, and is denoted by

\[
f_2(H_n) = M_e(C')
\]

Therefore

\[
f_2(H_n) > f_2(H_b)
\]

where \( H_b \) is the histogram of a blur image and \( H_n \) a clear image.
3.3. Pre-processing Phase

Cumulative Relative Frequency

As has been already mentioned above, the histogram data of a blurred image are concentrated on the left side of the histogram, at the smaller values. The cumulative relative frequency is computed, and the first bin which the 99% of cumulative relative frequency is achieved, is considered a new feature. This value must be less for a blurred image than for a clear one.

Since the heights of the bins represent the relative frequencies, the searched value is related with the cumulative relative frequencies. Given the pattern $H_n = [h^1_n, \ldots, h^{20}_n]$ and their respective centers of the bins $C = [c_1, \ldots, c_{20}]$, the third feature is defined as

$$f_3(H_n) = c_r$$  \hspace{1cm} (3.8)

where $r$ corresponds to the first bin for which the following condition is true

$$F_r = \sum_{i=1}^r h^i_n \geq \frac{P}{100}$$  \hspace{1cm} (3.9)

where $P$ is the percentage required. This way,

$$f_3(H_n) > f_3(H_b)$$  \hspace{1cm} (3.10)

Complementary Cumulative Relative Frequency

Finally, another feature may be obtained using the last bins of the histogram that correspond to high gradient values. In blurred images, these bins will be zero or will have low height, compared with the latest bins of a clear image. So, it can be considered a last feature as the sum of the $r$ latest relative frequencies of histograms. The accumulation of the last $r = 12$ bins of the histograms is used in this case. In this way, this sum will be smaller at blurred images.

Thus, the fourth feature is defined as

$$f_4(H_n) = \sum_{i=20-(r-1)}^{20} h^i_n = 1 - \sum_{i=1}^{20-r} h^i_n = 1 - F_{20-r}$$  \hspace{1cm} (3.11)

where $r$ is the number of bins that must be added. So,

$$f_4(H_n) > f_4(H_b)$$  \hspace{1cm} (3.12)

Therefore, each pattern $H_n$ is represented by the feature vector

$$F_n = [f_{1,n}, f_{2,n}, f_{3,n}, f_{4,n}]$$  \hspace{1cm} (3.13)

where $f_{i,n} = f_i(H_n), \ \forall i \in \{1, 2, 3, 4\}$

It is observed that the size has been reduced but the new vector retains the relevant information about the clearness of the image. Then, this vector $F$ is considered the input of the third stage, the classification problem.
3.4 Design of the Classifier

The last phase in the pattern recognition process is the design of the classifier. At this stage, the features previously estimated are used to build a rule of decision, which will determine the class of belongings of a pattern of unknown class. Therefore, inputs to the decision process are the set of features, feature vectors, and the output belongs to the classification space. The space is $M$-dimensional if the input patterns are classified into $M$ classes.

In this case, the pattern $H_n$ corresponds to the gradient magnitude histogram of a cleared image or conversely, belongs to a blurred image. So, the classification space is bi-dimensional.

Both the pre-processing phase and the decision processor are usually selected by the user of designer. There are no fixed rules that determine how the features or the decision rule should be chosen. Everything will depend on the data information available or the degree of accuracy desired in the classification.

The design of a feature extractor is usually an highly dependent task of the problem. The designer brings to bear all his knowledge about the problem, selecting the feature that best distinguish one pattern from another. As the choice of the features depend on the problem, there is little available theory to guide the design of the features extractor. Once the features extractor has been designed, there is a wide variety of procedures for designing the classifier. These procedures differ in the assumptions that are made about the behaviour of features, but these assumptions are often quite large. Thus, the methods to design the classifier are often much more independent of the special characteristics of the particular problem, and pattern classification theory is well developed [13].

The first step in choosing a classifier should be a close study of the training data. One of the best ways to do this is to build a histogram of the individual class for each available features, such as the one shown in Figure 3.9. This will give an approximate idea of which features are most useful.

In this case, four features in the pre-processing stage have been calculated. The aim is to find one feature that have greater discriminating power. That feature will be used by the classifier to determine if an image is blur or clear. By representing each feature in an histogram, you can see if the feature correctly separate classes.

When samples cannot be classified perfectly using the available set of features (see Figure 3.9), the goal may be to estimate the probability of membership in each class. Given a set of features, a sample can be classified as belonging to the class with the highest probability or, considering the costs of errors, into the class with the lower penalty. If it is predicted that an image with the feature value less than or equal to $T$ to be a blurred
3.4. Design of the Classifier

3.4.1. Design of the Classifier

(a) Feature 1

(b) Feature 2

(c) Feature 3

(d) Feature 4

Figure 3.9: Histograms of the features

image, and those with the feature value greater than T to be a clear image, then the value T is called decision boundary or threshold.

Then, the optimum threshold for each feature is searched. This optimum threshold is one that makes less mistakes when classifying.

So, the feature vector is \( F = [f_1, f_2, f_3, f_4] \). For all features, is expected that \( f_p(H_n) > f_p(H_b) \) with \( p \in \{1, 2, 3, 4\} \), being \( H_n \) the histogram of a clear image and \( H_b \) the histogram of a blurred image. Therefore, the process described below is valid for each of the features.

First of all, the maximum value and the minimum value taken by the feature \( f_p \) is calculated for all images of the labeled sample,

\[
\begin{align*}
\max_{n} \{f_{p,n}\} = \max_{n} \{f_p(H_n)\} &= \hat{F}_p \\
\min_{n} \{f_{p,n}\} &= \min_{n} \{f_p(H_n)\} = \hat{F}_p
\end{align*}
\]

(3.14)

(3.15)

Therefore, the range of values of feature \( f_p \) is \([\hat{F}_p, \hat{F}_p]\). In order to find the threshold that obtained the minimum number of images misclassified, \( T \) values are considered between \( \hat{F}_p \) y \( \hat{F}_p \), defined as
Chapter 3. Blurred Image Detection

\[ Th^t_p = \hat{F}_p + \frac{1}{T}(\hat{F}_p - \bar{F}_p)t, \quad t = 1, \ldots, T \]  

(3.16)

For each \( t = 1, \ldots, T \), \( Th^t_p \) is considered the decision boundary, so that if \( f_{p,n} < Th^t_p \) the image \( I_n \) is classified as blur image, and if \( f_{p,n} > Th^t_p \), the image is classified as clear image. A new classification vector is created \( B' = [b'_1, \ldots, b'_N] \) with \( b'_n = 0 \) if the image was classified as clear and \( b'_n = 1 \) if the image was classified as blur.

Then, the error rate obtained with the feature \( f_p \) using the threshold \( Th^t_p \) is denoted by \( TE^t_p \) and it is defined by the fraction of the misclassified images.

\[ TE^t_p = \frac{k^t_p}{N}, \quad t = 1, \ldots, T, \quad p = \{1, 2, 3, 4\} \]  

(3.17)

where \( k^t_p \) is the number of misclassified samples, and it is obtained as

\[ k^t_p = \sum_{n=1}^{N} |b_n - b'_n| \]

The most important thing is to acquire the minimum error rate, and for this, the threshold that obtained the minimum misclassified images is chosen.

The minimum error rate is denoted by

\[ TE^{t_0}_p = \min_t TE^t_p \]  

(3.18)

and it is achieved at threshold \( Th^{t_0}_p \).

This procedure is performed for each of the components of the feature vector \( F \), so every feature \( f_p \) has its minimum error rate \( 3.18 \) and the corresponding threshold \( Th^{t_0}_p \). The feature that best classifies images is the feature with lower error rate, and it is computed by

\[ \arg \min_p TE^{t_0}_p = f_{p_0} \]  

(3.19)

Thus, the optimal feature is \( f_{p_0} \) with its associated threshold \( Th^{t_0}_{p_0} \). In this way, any image of unknown class can be assigned to the class of cleared images or blurred images.

In short, given a new image \( I \) of unknown class, the magnitude of the gradient is calculated and the histogram is constructed, by eliminating null values of magnitude. The feature \( f_{p_0} \) is calculated for this histogram and, if \( f_{p_0}(I) < Th^{t_0}_{p_0} \), the image is classified as blur and if \( f_{p_0}(I) > Th^{t_0}_{p_0} \), the image is classified as clear.

\[ TE^{t_0}_4 = 0.0909 \implies 9\% \]
Chapter 4

Results

Quien nunca ha cometido un error,
nunca ha probado algo nuevo
Albert Einstein

The main objective of this Project is to design a classifier able to determine if an image is blurred or clear. In order to do that, a pattern recognition process is carried away and then, the actual design is done.

Summarizing all the process explained in chapter 3, 16 video sequences of different objects is available, being \( N = 2299 \) the total number of images with resolution equal to \( 640 \times 480 \) pixels. The gradient of each image is calculated and the data is represented in a histogram of 20 bins representing the relative frequencies. These frequencies form a vector \( H_n \) that is taken as the pattern to be classified. Therefore, at this point, each image has been replaced by a vector of length 20.

Next, the \( H_n \) patterns go through a feature extraction process. These features were explained in detail in Section 3.3.2 and they are used to design the classifier of blurred images following the procedure described in Section 3.4.

In this Chapter, the four available features are analyzed in order to determine which one of them produces the lowest error when classifying the patterns and so finding the decision threshold.

4.1 Feature Analysis

Even though a vector with the four features \( F = [f_1, f_2, f_3, f_4] \), described in Section 3.3.2 is available, the classifier will be designed taking into account
only one of them. This single feature will be the one with the highest ability to detect the blurriness of the images through its $H_n$ patterns.

As seen in Section 3.4 by looking at the histograms of each feature one can know which one discriminates the most. However, the threshold has yet to be determined. In order to do that, the ranges of the features are considered and $T$ equidistant values are chosen. In this case, $T = 20$ has been chosen. These values are the candidates for an optimal threshold.

Next, for each one of the features $f_p$, the classification of the labeled sample is carried away using each one of the $T = 20$ thresholds. In order to evaluate the classifications, the error rate $TE_{t_p}$ is calculated. Such error rate is defined as the number of images that are wrongly classified divided by the total number of images $[3.17]$. From these results, the one having the lowest quotient is chosen to classify the samples. The minimum error rate $TE_{t_0}^{f_p}$ is associated to a threshold $Th_{t_0}^{f_p}$. This threshold will be the decision boundary for the corresponding feature $f_p$. The results are shown in the following Table 4.1.

<table>
<thead>
<tr>
<th></th>
<th>$f_1$</th>
<th>$f_2$</th>
<th>$f_3$</th>
<th>$f_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min error</td>
<td>0.0926</td>
<td>0.1370</td>
<td>0.0940</td>
<td>0.0909</td>
</tr>
<tr>
<td>Threshold</td>
<td>0.2632</td>
<td>0.8421</td>
<td>0.4737</td>
<td>0.2632</td>
</tr>
</tbody>
</table>

Table 4.1: Results

Amongst the 4 features, the one of interest is the one with the lowest classifying error, being in this case the feature $f_4$, called complementary cumulative relative frequency, with an error of 9% and its corresponding threshold $Th_{t_0}^{f_4} = 0.2632$.

This optimum threshold corresponds to 0.5% of contour pixels, due to the values of the features were normalized.

Hence, if the accumulation of the last $r = 12$ bins of the histogram represents more than 0.5% of contour pixels, the image is classified as clear, and otherwise, the image is classified as blur. And finally, the classifier of blurred images is designed.
Chapter 5

Conclusions and Future Work

Next, an analysis of this Master Thesis is presented, as well as the possible modifications to be implemented from the obtained classifier.

5.1 Conclusions

In this work, a method to detect blurred images has been presented. The images were extracted from a video sequence recorded by a humanoid robot with the purpose of estimating the geometry of an object.

Given a video sequence, a category label is assigned to each frame. These labels differentiate between images that present blur from those that don’t.

In order to detect the blurriness of the images, a histogram of the gradient magnitude is built. Thus, each image is represented by its histogram, reducing the amount of data to be processed without losing information about the sharpness of the images.

The classification system uses these histograms to detect the blurry images. As discussed in previous Chapters, a set of descriptors that may contain information about the blur are extracted. In this work, the set of descriptors or features have been extracted from the histograms. Thus, the values of the extracted features are stored in the feature vector that will be analyzed by the classifier.
A classification of each feature is done and then, the feature with the highest amount of correctly classified images is chosen. The number of images that have been correctly classified represents the 91% of the database and corresponds to the accumulation of the last \( r = 12 \) bins of the histogram. If this summation represents more than 0.5% of contour pixels, the image is classified as clear. Otherwise, the blurring effect is said to be present.

After deleting the blurred images from the sequence, an object segmentation is performed using a region-based particle filter approach. Finally, a space carving algorithm is used for estimating the geometry of the object.

The programming language used to develop this project is MATLAB. MATLAB is a scientific programming language that provides strong mathematical and numerical support for the implementation of advanced algorithms. It is for this reason that MATLAB is widely used by the image processing and computer vision community.

Lastly, it is necessary to highlight the achievements of this Master Thesis:

- Learning of advanced knowledge in image processing and computer vision
- Deeper study on image processing and computer vision methods
- Use of different tools and programming languages
- Revision of the State of the art related to the study of detection and classification of blurry images

5.2 Future Work

Although this work has a specific purpose, it can also be seen as the first step towards a broader future research. The obtained classifier of this research uses only one of the four extracted features of the histogram. Another classifier could be built using the two more discriminatory features of the feature vector. Thus, the patterns would be defined by a two dimensional vector, where each coordinate points at the values of these two features.

Another option is to use the histogram of the direction gradient. As seen in Section 3.3.1, the directions of the gradient of a clear image have almost the same probability, whilst the ones of a blurry image vary in value. Therefore, the variance of the gradient direction can be used as a second feature, in conjunction with the other feature already discussed in this thesis.
Appendix A

3D Shape Reconstruction from a Humanoid Generated Video Sequence
3D Shape Reconstruction from a Humanoid Generated Video Sequence

P. A. Martínez, D. Varas, M. Castelán, M. Camacho, F. Marques, G. Arechavaleta

Abstract—This paper presents a strategy for estimating the geometry of an interest object from a monocular video sequence acquired by a walking humanoid robot. The problem is solved using a space carving algorithm, which relies on both the accurate extraction of the occluding boundaries of the object as well as the precise estimation of the camera pose for each video frame. For data acquisition, a monocular visual-based control has been developed that drives the trajectory of the robot around an object placed on a small table. Due to the stepping of the humanoid, the recorded sequence is contaminated with artefacts that affect the correct extraction of contours along the video frames. To overcome this issue, a method that assigns a fitness score for each frame is proposed, delivering a subset of camera poses and video frames that produce consistent 3D shape estimations of the objects used for experimental evaluation.

I. INTRODUCTION

Most humanoid robots rely on vision systems in order to perceive the environment and resemble human capabilities. In particular, monocular vision is preferred for small-sized humanoids that are certainly constrained to be equipped with lightweight, low-cost and low-energy consumption devices. For these robots, there is a tradeoff between a suitable camera and the quality of the images acquired during the biped march, as the stepping impacts cause jerky camera movements which generate continuous blurring along the related video sequence. Localizing the robot is an additional complex problem due to discrepancies in time among sensor readings, i.e., the orders of magnitude from the acquired frequency signals differ for each sensor and the rate of divergence from the walking reference trajectory is high for small distances.

In the context of 3D object reconstruction, analyzing a monocular video sequence acquired by a humanoid robot represents a difficult task which involves solving for camera localization as well as extracting meaningful image features under challenging motion conditions. This paper investigates the feasibility to estimate, in a multi-view fashion, the 3D geometry of an interest object from the video frames generated along the march of a humanoid. In order to capture multiple views, the robot performs a circular trajectory generated through a locomotion control that corrects the positions and orientations of the robot in accordance with vectors lying on a virtual circle of known radius.

Fig. 1: The proposed strategy. A humanoid robot records a video sequence that samples multiple views of the shape of an interest object. A visual-based locomotion control uses the monocular localization of the robot to correct its stepping and perform the required trajectory. An analysis of the recorded sequence is applied in order to determine the suitability of each frame for the purposes of contour extraction. Finally, from the selected frames, a particle filter-based object segmentation process is coupled with a space carving algorithm for estimating the geometry of the object. The figure shows the 25 camera poses of the selected video frames and the estimated 3D shape of the object.

For object segmentation, a strategy has been developed that aims at selecting a suitable set of video frames for robustly reconstructing the 3D shape of the object. In a first stage, blurred images are eliminated from the sequence as well as those frames where parts of the object appear outside the image limits. From this subset, object segmentation is performed using a region-based particle filter approach, from which a consistency score is assigned for each frame. The video frames with the highest scores that also observe a uniform distribution of the sampled object views are finally selected for 3D shape recovery. The process is illustrated in Figure 1, where the final selected video frames are shown as camera poses surrounding an object of interest. In this sense, the main contribution of this paper is a method that is capable of analyzing a video sequence generated by a humanoid robot for the purposes of 3D object reconstruction from multiple views.

The rest of the paper is organized as follows: Section II presents the relevant work that approaches both 3D shape estimation and object segmentation; Section III describes the proposed strategy to record the video sequence of the object; Section IV depicts the method developed to analyse...
the recorded video and select the image frames with the most suitable contours for 3D reconstruction; Section V shows the 3D shape recovery results for three different objects and finally Section VI provides concluding remarks and future lines of research.

II. RELATED WORK

The classical computer vision problem of monocular 3D object reconstruction has been adapted to robotic platforms in order to provide them with a way to perceive the environment and interact with it, i.e. make a decision, develop a navigation task or grasp an object. For humanoid robots provided with a stereo vision system, solutions have been proposed using next-best-view (NBV) techniques. The problem, commonly approached in a multi-objective optimization manner, consists of estimating the next camera pose that maximizes the unknown volume of the object given a current voxel grid. When coupling NBV with humanoid platforms, the new camera poses are also required to agree with the set of admissible head and body configurations of the robot.

The first attempts in reconstructing the shape of an object by a humanoid robot have probably been described in [1]. Here, an optimal set of camera and body poses were calculated in order to acquire a set of views of the object for achieving a partial reconstruction. For isolating object from background, a red table was used to place the object. The 3D model was estimated from the registration of five disparity maps obtained from stereo images of the object. While this work did not focus on estimating the complete geometry of the object, it showed how a partially reconstructed model would suffice for recognizing the object in cluttered environments. A method for achieving a complete reconstruction that considered obstacle avoidance was later proposed in [2], with successful simulation results. In this approach, a 3D occupancy grid covered the object and an updating operation of the occupancy values of the grid was performed for each captured stereo frame.

More recently in [3] a strategy was presented for acquiring monocular views of an object by a small size humanoid. From these image views, contours of the object were obtained to reconstruct 3D shape using the space carving approach of [4]. The task of extracting contours was simplified using a color based thresholding technique and, as a consequence, the objects to reconstruct were painted in red while the acquisition scenario was covered in blue. For controlling the robot, an extended Kalman filter approach was developed to estimate the position and width of the object with respect to the robot. Camera localization was achieved by solving for extrinsic camera parameters from a set of eight colored landmarks of known distance.

A common aspect to note from the above approaches is that image features and therefore the 3D shape of the object are inferred from still camera poses. In other words, in order to calculate its next state, the robot has to make a pause and interrupt its motion. Rather than being related with mechanical limitations, this pause is a consequence of the optimization process implied in the tasks of calculating the next camera pose or updating information about the object’s shape. Although these approaches profit on the versatility of head and body pose configurations of the humanoid robot, exploiting the full range of data contained in the frames of a video sequence has been neglected.

Indeed, when video information is considered, some problems arise. One of the conventional image quality degradation is the blurring of moving camera sequences [5]. In [6], a system that detects blurred images and classify them using the magnitude and the direction of its gradients is proposed. Although experiments show satisfactory results, an annotated database is needed to train an SVM classifier. Besides fast camera motion, other difficulties such as changes of the object position and shape are problems that should be handled by the system [7].

In contrast with other tracking methods, particle filters [8] can robustly track objects from a sequence of different views as they neither are limited to linear systems nor require the noise involved in the process to be Gaussian. In [9], a particle filter with edge-based features is proposed. This method has been widely used since it provides a robust framework for tracking curves in clutter. However, the space of possible deformations is limited and some transformations of the object shape may not be correctly estimated. This restriction could be critical especially in a multiview scenario. We adapt this idea using shape descriptors without any restriction in the space of possible deformation.

Image-based features for particle filters were introduced by [10]. In it, color histogram is used to robustly track objects in the scene. This feature has the advantages of being scale invariant and robust to partial occlusions and rotations. Moreover, it can be efficiently computed. In our work, we use the Diffusion distance [11] instead of the Bhattacharyya distance [12] for histogram comparison since it leads to better perceptual performance. As the color of an object can vary through time, the target model is adapted during temporally stable image observations in [13]. Note that [10], [13] do not provide shape estimation.

In our work, we propose to use the region-based particle filter presented in [14] to allow tracking and segmenting objects in sequences of different views. This is a suitable algorithm for this task as a geometrical shape is not considered to represent the object. Instead, its contours are propagated between image pairs of consecutive views.

As far as the separation of object from background is concerned, efforts have been developed for coupling 3D object segmentation with successful results in grasping tasks. In [15], a model-free algorithm was proposed to partition the surface normals of depth images acquired with an RGB-D sensor, identifying the connecting regions that belong to several objects in a cluttered scene. Other approaches have also been introduced for stereo images, as in [16], where the principle of fixation by an active observer was used to emulate foveated vision, resulting in an improved selection of the object’s contours. While both approaches are capable of performing segmentation from background as new objects are included in the observed scene, the sensors remain fixed.
and this assumption does not fit into biped robotic platforms.

It is worth commenting on the growing popularity of RGB-D sensors, which has allowed the recent development of the now called RGB-D SLAM (Simultaneous Localization and Mapping) systems [17] [18]. These methods have proved successful in SLAM tasks which include a dense 3D reconstruction of the observed scene. However, they have been tested over databases that consider smooth transitions between video frames such as hand-held camera and wheeled robot motion [19]. Unfortunately, as the march of a humanoid robot implies constant swinging, the risk of sudden changes in the motion of the camera may compromise the applicability of these approaches.

### III. Monocular Vision-based Locomotion Control

Acquiring multiple views of the object of interest is the first step in the reconstruction of its geometry. For each video frame, it is also necessary to register the position and orientation of the camera, which is estimated under a monocular vision-based framework. We have chosen PTAM (the Parallel Tracking and Mapping software of the University of Oxford [20]) to solve this problem as it is able to track hundreds of features, perform both local (incremental) and global bundle adjustments and grow the 3D map when new keyframes appear. These tasks are computed in parallel resulting in real-time applications. For the monocular case, an initialization that simulates a stereo pair to approximate the depth of the initial 3D points is crucial to obtain coherent results.

For generating the robot trajectory we propose a monocular vision-based locomotion control that drives the camera of the robot to face towards the center of the table where the object is located. Additionally, the position of the robot is constrained to keep a constant distance from the center of the table in order to emulate a surrounding trajectory.

#### A. Robot localisation

The output of the camera localization process is a rotation matrix $R_w$ and a translation column vector $t_w = [x_w, y_w, z_w]^T$ that relate the world and the camera, in other words, the rigid body transformation from the axis of the world to the axis of the camera. For controlling the march of the robot we are only concerned with the position of the robot on the $xy$-plane and its orientation angle. Note that the head (camera) of the robot has been locked to be fully aligned with its body, thus, the homogeneous matrix that maps the robot body frame to the world frame can be approximated as

$$T_b^w \approx T_c^w = \begin{bmatrix} R_w & t_w \\ 0 & 1 \end{bmatrix} ,$$

with $w,c$ and $b$ respectively standing for world, camera and body. The position of the robot in world coordinates can be directly taken from the translation vector $t_w$. The orientation angle of the robot can be found from the first two elements of the translation vector transformed into camera coordinates as

$$\theta_c = \tan^{-1}(y_c/x_c),$$

where $[x_c,y_c,z_c] = -R_c^w t_w$.

#### B. Locomotion control

In order to solve the problem of multi-view 3D reconstruction from object segmentations the robot needs to surround the object of interest. For this task, we propose a locomotion control that directs the next position of the robot to lie along the circumference of a known radius circle while its orientation is directed towards the center of the circle. For an effective translation to occur along the circumference of the circle, an angular displacement $s$ from the current to the following robot’s state has to be considered.

Let $x_{CoM}^{ref} = [x_w, y_w]^T$ be the reference position of the center of mass (CoM) on the $xy$-plane of the world taken from the translation vector $t_w$. The orientation angle $\theta_{ref}$ can be calculated as shown in Eq. 2. The current state of the robot is defined by the pair $(x_{CoM}^{ref}, \theta_{ref})$ and its projected position lying on the radius $r$ circumference may be defined as the vector $x_p = [x_p, y_p]^T = r(x_{CoM}^{ref}/||x_{CoM}^{ref}||)$. The target state of the robot at time $k+1$ is defined as $(x_t, \theta_t)$ and can be obtained by adding the angular displacement $s$ to the projected vector $x_p$. The target orientation $\theta_t$ is directly estimated from $x_t$ and its direction is inverted as the robot is facing towards the center of the circle.

The reference linear velocity of the CoM, $x_{CoM}^{ref}$ is computed considering a proportional control based on the distance between the current estimate of the robot’s CoM position and the computed target position. Likewise, for the reference angular velocity, $\dot{\theta}^{ref}_c$, the difference between the current and target orientation is used. Therefore, the errors $e_x = x_{CoM}^{ref} - x_t$ and $e_\theta = \theta_t^{ref} - \theta_t$ are regulated by imposing the exponential convergences

$$e_x = -\lambda_x e_x \quad \text{and} \quad \dot{e}_\theta = -\lambda_\theta e_\theta,$$

where $\lambda_x$ and $\lambda_\theta$ are experimentally tuned constant proportional gains. This procedure is performed while the robot does not reach the end of the surrounding trajectory and is formally described in Algorithm 1.

The input of the walking pattern generator (WPG) is given by $x_{CoM}^{ref}$ while the output considers a dynamically stable trajectory of the CoM, the position of the foot in contact and the next footstep placement. The WPG solves quadratic programs with a predefined time horizon as it is proposed in [21]. In this case, the reference orientation $\theta_t^{ref}$ is used to express the inequality constraints that define the admissible region to place the next footstep. The computation of the

---

**Fig. 2:** Three objects to reconstruct. From left to right, example images of recorded sequences Mug, Duck and Action Man.

where $[x_c,y_c,z_c] = -R_c^w t_w$. 

---
Data: Localization $x_{CoM}^{ref}$ at current time $k$, orientation $\theta_{CoM}^{ref}$ at current time $k$, radius $r$, angular step $s$.

Result: Reference linear velocity $\dot{x}_{CoM}^{ref}$ at time $k+1$, reference angular velocity $\dot{\theta}_{CoM}^{ref}$ at time $k+1$.

while $x_{CoM}^{ref}$ outside stopping region do
  
begin
  $[x_p, y_p]^T = r(\frac{x_{CoM}^{ref}}{|x_{CoM}^{ref}|})$
  $x = [x_c, y_c]^T = r\begin{bmatrix} \cos(\tan^{-1}(y_p/x_p) + s) \\ \sin(\tan^{-1}(y_p/x_p) + s) \end{bmatrix}$
  $\theta = \tan^{-1}(y_p/x_p)$
  $\dot{x}_{CoM}^{ref} = -\lambda_1(x_{CoM}^{ref} - x_c)$
  $\dot{\theta}_{CoM}^{ref} = -\lambda_2(\dot{\theta}_{CoM}^{ref} - \dot{\theta})$
  Apply a WPG given $(x_{CoM}^{ref}, \dot{\theta}_{CoM}^{ref})$
  Generate locomotion with inverse kinematics
  end

Algorithm 1: The robot performs a surrounding trajectory in accordance with a circle of radius $r$ and an angular displacement $s$.

**TABLE I:** Camera position and orientation error

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Mean position error</th>
<th>Mean orientation error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mug</td>
<td>7.4 mm ± 4.3 mm</td>
<td>3.05° ± 1.82°</td>
</tr>
<tr>
<td>Duck</td>
<td>8.9 mm ± 5.4 mm</td>
<td>3.35° ± 1.95°</td>
</tr>
<tr>
<td>Action Man</td>
<td>7.5 mm ± 4.1 mm</td>
<td>2.85° ± 1.90°</td>
</tr>
</tbody>
</table>

The footstep generation of the robot successfully achieves the expected circular trajectory, with camera orientations pointing towards the center of the table.

In order to provide a quantitative performance of our control, for each video sequence we measured the distance from every estimated camera position to the nearest position (projection vector of length 0.6 m) on reference circumference, as the aim of the control is to achieve a trajectory resembling a circle. These distances were considered once the camera poses first entered the circular area, i.e., from the red arrow in Figure 3 (b). The average error in position is shown in Table I, where the proposed scheme delivers a dismissible error of at most 1 cm for the three sequences. The orientation error was also measured considering the angular distance between the current orientation camera angle and the angle of the nearest vector on the reference circumference. The average difference reveals departures of 3° from the expected behavior, confirming the applicability of this strategy for recording video sequences to sample multiple views of the interest object.

IV. OBJECT SEGMENTATION FROM A VIDEO SEQUENCE

Once the video has been recorded, the object must be segmented in order to create a 3D model. In this work, we adapt the region-based particle filter presented in [14] to extract the 2D shape of the object from partitions (See Figure 5(b)) associated with a set of multiple views of the object. Only images containing the entire object without blur are processed and a subset of these images is finally selected to reconstruct the object in accordance with their final segmentation quality. A diagram of the proposed framework is presented in Figure 4.

A. Pre-Processing

As the shape of the object is extracted from a partition, the final quality of the model is highly dependent on the image partitions generated along the sequence. These object segmentations at different views are used by a space carving algorithm to reconstruct the 3D model. Although the smoothness of this model increases with the number of segmentations extracted from different views, the larger the number of images considered in this process the higher the probability of an erroneous object shape estimation at least in one view. Thus, a subset of images from the sequence is selected to robustly create a 3D reconstruction of the object.
Fig. 3: Applying the proposed control to record a video sequence. The blue path depicts the estimated robot positions along the circular trajectory. The virtual circle that helps controlling the actual trajectory is approximately centered at $(0.1, -0.1)$. A close up view of (a) is presented in (b), while example video frames corresponding to green arrows in (a) appear in (c).

![Image](image1.png)

Fig. 5: Discarding blur. In (a) a blurred image is presented. Images (b) and (c) show its associated partition and the best estimation of the object given this partition, respectively. The blurring effect creates erroneous contours which are not capable to represent a correct segmentation of the object. In this example, the beak of the duck is not included and, as a result, this part of the duck would not be reconstructed.

Two main situations can be found in which a region-based particle filter may not correctly recover the shape of the object. First, when a part of the object is not present in the image. And second, when the blurring effect degrades the quality of the object contours. In order to avoid erroneous estimations, two pre-processing steps select those images in which the object can be correctly segmented. These steps estimate the position of the object in the scene and the blurring of the image respectively.

1) Blurring estimation: Blur is one of the conventional image quality degradations and it can be caused by various factors. In our application, this effect arises due to the rapid camera movement of the robot. The quality of partitions decreases drastically when the blurring effect appears, producing corrupted contours and mixing object and background pixels in the same regions (Figure 5).

Since the image gradient is highly related to image blurring [5], our blur detector computes the magnitude of this gradient to estimate the blur present in an image. Then, a histogram of the gradient is built (in this work, 20 bins have been used). As the contours of a clear image are more precisely defined than the contours of a blurred image, its histogram is expected to contain some contours with large values. On the contrary, contour magnitudes of blurred images should be small.

To this extend, the accumulation of the last 10 bins of the histogram is used to classify the image. If this summation represents more than 0.5% of contour pixels, the image is classified as clear. Otherwise, the blurring effect is said to be present.

2) Position estimation: The position of the object in the scene is computed using its relative position with respect to the camera. Due to the camera movement, the object may not be completely observed, and some of its parts can be projected out of the image.

When this situation arises and the image is selected to generate the 3D model, the part which is not included in the scene will not appear in the final reconstruction even if it is correctly segmented in other views. To avoid this problem, a color-based particle filter [13] is used to estimate the position and the bounding box of the object along the sequence.

Following a conservative policy, images where the detected bounding box is closer than 25 pixels to an image border are not taken into account to extract the object contours.

B. Region-Based Particle Filter

This method segments the object along a sequence propagating its shape through time. To this end, similarities between regions are analyzed. Then, parts associated with both the object and background are put in correspondance for each pair of views.
In [14], a region-based particle filter is presented in which Monte Carlo methods and a representation of the image in terms of regions are combined. This algorithm, does not only provide the position of the object as in the color-based approach. Instead, it also estimates the shape of the object along the sequence, given an object mask of the first frame. This mask should be provided to the algorithm and in our experiments it is set by hand (See Figure 6). The object mask is used to create a color model of the object in the image that serves as a reference to weight particle estimations. In this work, a histogram has been used as object model. In the region-based approach, the measurement at time k, \( z_k \), is composed by the input image and its partition, whereas the estate, \( x_k \), is formed by a union of regions from the partition associated with the input image. In this work, partitions are obtained using [23]. An example of image partition created with this technique can be observed in Figure 6.

Each particle stands for a state represented by a union of regions that define an estimation of the object. As particles represent different estates of the tracked object, they are also represented by unions of regions. Thus, in order to form the new set of particles, not any propagation is allowed. In other words, the measurement (partition) is used in the propagation process. Then, the weight update equation can be written as:

\[
 w_k^{(i)} \propto w_{k-1}^{(i)} \sum_c p(z_k|x_k^{(i)}) p(x_k^{(i)}|x_{k-1})
\]  

(3)

where \( w_k^{(i)} \) is the weight of the \( i \)-th particle at time \( k \) and \( c \) swaps all the possible states.

This summation becomes intractable using a brute force approach. For each particle, its probability of being represented by all the possible combinations of regions of the next partition should be computed (\( p(x_k^{(i)}|x_{k-1}) \)) and evaluated (\( p(z_k|x_k^{(i)}) \)). To solve this problem, the algorithm takes advantage of the two steps of a usual tracker: prediction (movement prediction) and perturbation (particle randomness).

a) Prediction: In this step, the shape of the object in the next frame is estimated to ensure a minimum quality of the new set of particles. In order to perform this process in a robust and efficient manner, a particle support partition (PSP) is created taking into account the intersections between particles. Using this partition, all particles can be propagated with a single optimization process: label propagation. This process labels regions from the new measurement with labels from the particle support partition optimizing similarities between regions which are adjacent as it can be observed in Figure 7. These similarities are computed over the contour elements using color, texture and distance information. As it is expected that the object shape does not change abruptly between consecutive views, only regions in a neighborhood of 50 pixels are considered adjacent. This adapts the concept of adjacency presented in [24] (multiple static partitions) and in [14] (rapid changes) to a multiple view scenario. The result of the process is a labeled partition (LP) in which regions from the new partition have been labeled with labels from the PSP.

b) Perturbation: For each particle, \( N \) changes are randomly proposed separately. These changes consist on adding/removing regions that belong to the particle or its neighborhood. Then, a greedy algorithm is proposed in which those changes that improve a similarity measure (Diffussion distance) between the particle and the model, are stored and combined to form the final particle. Details of this algorithm can be found in [14].

Finally, the estimation of the object is obtained as the combination of the state of the particles. Note that in the region-based case each particle has its own associated object shape obtained through the two previous steps. Thus, the object shape is estimated combining the masks of all the particles. As a result of this combination, a certain probability of belonging to the object is assigned to each region. The final object shape is estimated considering those regions with a probability higher than a given threshold (In this work, 50% has been used). The capacity of estimating the 2D shape of the object in an image view given its shape in a similar view makes this algorithm suitable for reconstruction applications.

C. Image selection

As it has been previously commented, errors in the object shape estimation rapidly degrade the quality of the final reconstruction. In order to avoid this degradation, only a subset of the views analyzed by the region-based particle filter are used to create the 3D model.
Images are selected according to the Diffusion distance between the segmented object and the model. Moreover, the circular distribution of the cameras is taken into account to correctly represent the entire 3D object. The image with the highest coefficient is chosen first. Then, from the rest of images, the view with the highest coefficient which is not included in a temporal window of 7 frames centered in any chosen image is selected. This process is iterated until 25 frames are chosen or the coefficient falls below a given threshold. The resulting set of views is used to robustly reconstruct the object.

V. 3D RECONSTRUCTION RESULTS

As far as the 3D reconstruction of the object is concerned, we used a method based on shape from occluding boundaries known as space carving [4]. Roughly, this method uses the camera matrix in order to reproject, towards the world, the area bounded by the silhouette of the observed object in the image. The camera matrix is calculated as

\[ \mathbf{P} = \mathbf{K}[\mathbf{R}_w, \mathbf{t}_w] \]  

where \( \mathbf{K} \) is the matrix of intrinsic parameters of the camera. From a set of multiple silhouettes with known camera matrices, a 3D model is finally recovered from the intersection of all reprojected silhouettes into the voxel map. This process, which resembles sculpting (carving), is usually posed using a turntable and a fixed camera, which greatly simplifies the tasks of object segmentation and estimation of the camera matrices. On the contrary, our method is capable of dealing with a challenging video sequence recorded from a humanoid robot in motion.

In this section, we show how by coupling a robust object segmentation method with a trajectory that guarantees exhaustive sampling of the image views of the object, it is possible to estimate a visually coherent 3D geometry of the interest object.

The final results of our complete framework are illustrated in Figure 8, where we present the chosen video frames after applying the image selection process described in the previous section. In column (a), we have only included scenarios for Duck and Action Man sequences as the corresponding scenario for Mug appears in Figure 1. It is worth commenting on the differences between the set of selected images from the different scenarios. Particularly, in the Mug experiment a large region of camera poses was left out of the quality set, which can be explained as a consequence of the robot being too far from the object along certain regions of the performed path. In this case, the object was not placed in the center of the table and, as a consequence, it appeared too small for an accurate segmentation to become feasible. Nonetheless, the rendered views of the 3D reconstruction provided in Figure 8 (b) reveal that the shape of the mug was reasonably recovered. Likewise, the rendered views obtained from Duck and Action Man video sequences provide a range of object views that qualitatively agree with an expected reconstructed shape of the object.

Table II shows how the recovered 3D models are also consistent with the objects’ physical dimensions in the world, as we measured the width, depth and height of each object.
TABLE II: Departures from physical dimensions

<table>
<thead>
<tr>
<th>Object</th>
<th>Width</th>
<th>Depth</th>
<th>Height</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mug</td>
<td>5.6 mm</td>
<td>6.9 mm</td>
<td>1.0 mm</td>
</tr>
<tr>
<td>Duck</td>
<td>5.2 mm</td>
<td>5.0 mm</td>
<td>6.0 mm</td>
</tr>
<tr>
<td>Action Man</td>
<td>5.0 mm</td>
<td>4.5 mm</td>
<td>5.2 mm</td>
</tr>
</tbody>
</table>

and each reconstruction in order to obtain a degree of similarity between their corresponding bounding boxes. As shown in the table, the average error was 5.3 mm in width, 5.5 mm in depth and 4.1 mm in height. Considering the accuracy in the recovered geometry, this could be potentially used in further tasks such as grasping and identification.

Finally, rendered views corresponding to incorrect 3D reconstructions are additionally depicted in Figure 8 (c). For this experiment, we included a single video frame observing a low score in the image selection process. Note how the recovered 3D models exhibit important missing parts such as the handle, the beak and the head, respectively for Mug, Duck and Action Man video sequences.

It is worth commenting that our method presents important differences with other 3D reconstruction approaches. For example, while [1] emphasizes the full body posture of the HRP-2 in order to get still images of the object, we favor multiple view exploration by relying on a powerful computer vision segmentation strategy that works over humanoid locomotion generated video sequences. Also, while they focus on the partial reconstruction of the object for the purposes of later identification, our approach values the full 3D reconstruction of the interest object. For this reason, comparing approaches would not provide enough insight and has not been included in this paper.

VI. CONCLUSIONS AND FUTURE WORK

We have shown how a state-of-the-art technique in video analysis can be adapted to the challenging video sequences generated by a humanoid robot in motion. Particularly, the complex task of estimating the 3D shape of an interest object has been achieved by relying on a monocular visual-based control of the robot and the robust extraction of silhouettes from a selected subset of highly scored video frames. The reported results are promising and future improvements can be drawn in two directions. As far as the control of the robot is concerned, relaxing the circular supposition might generate camera views of the object that contain important information, as the robot would be able to get closer or farther from the object when needed. Also, incorporating constraints to use a wider range of body postures is desirable for the purposes of generating a richer set of camera views. The capabilities of the video analysis strategy can be as well extended and we are considering incorporating 3D information related to the sparse cloud of points generated during the march of the humanoid robot, which can be useful for a rough object-from-background separation.
Bibliography

Procure seguir a tu corazón
aunque pienses que es demasiado tarde
Antoine de Saint-Exupéry


Cuando no puede medirse o expresarse en números aquello de que se habla,
el conocimiento que se posee es pobre y defectuoso;
puede ser un principio de conocimiento, pero difícilmente
podría decirse que ha alcanzado la categoría científica,
cualquiera que se la materia de que se trate.
William Thomson Kelvin

¿Qué te parece desto, Sancho? – Dijo Don Quijote –
Bien podrán los encantadores quitarme la ventura,
pero el esfuerzo y el ánimo, será imposible.

Segunda parte del Ingenioso Caballero
Don Quijote de la Mancha
Miguel de Cervantes