What’s in an Image?
Towards the computation of good views for three-dimensional objects.

Oleg Polonsky
What’s in an Image?
Towards the computation of good views for three-dimensional objects.

RESEARCH THESIS

Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in Computer Science

Oleg Polonsky

Submitted to the Senate of the Technion — Israel Institute of Technology
HESHVAN 5766 HAIFA DECEMBER 2005
The research thesis was done under the supervision of Prof. Craig Gotsman in the Computer Science Department.

I thank Prof. Craig Gotsman for his guidance throughout this research.

I would like to thank our colleagues at the Max-Planck-Institut für Informatik (MPII), Saarbrücken, Germany and at the Istituto di Matematica Applicata e Tecnologie Informatiche (CNR-IMATI), Genova, Italy for our joint work and fruitful discussions.

I thank my wife Anna, my children Lea and Roni and my parents Inna and Leonid for their love, continuous support and encouragement.

The generous financial help of the Technion is gratefully acknowledged.
Contents

Abstract

1 Introduction 3
  1.1 The Goal .................................................. 3
  1.2 Previous Work ............................................. 6
  1.3 Problem Setting ........................................... 9

2 Problem Analysis 13
  2.1 Good Views and Canonical Views .......................... 13
  2.2 Canonical View Properties ................................ 16
  2.3 Possible Pitfalls .......................................... 18
  2.4 Proposed Framework ....................................... 21

3 View Filtering 25
  3.1 Three-quarter Views ....................................... 25
  3.2 A Simple Case Study ....................................... 27
  3.3 Handling a General Object ................................. 29

4 View Descriptors 33
  4.1 Surface Area .............................................. 36
  4.2 Projected Area ............................................ 37
  4.3 Projected Area Entropy ................................... 39
  4.4 Semantic Entropy .......................................... 40
  4.5 Visible Curvature ......................................... 42
  4.6 Curvature Entropy ......................................... 42
  4.7 Silhouette Length ......................................... 44
  4.8 Silhouette Entropy ......................................... 45
  4.9 Evaluation of Results ..................................... 46

5 Discussion and Conclusions 51

Bibliography 53
Hebrew Abstract
# List of Figures

1.1 Different kinds of *good* views .......................... 4  
1.2 Good and Bad View examples ............................ 5  
1.3 Effects of different lighting ............................ 11  
1.4 Viewpoint parametrization .............................. 11  

2.1 Canonical coordinate system ............................. 15  
2.2 Two views of an elongated box ........................... 17  
2.3 A “beautiful” view of a torus ............................ 17  
2.4 Maximal *mesh saliency* viewpoint ...................... 20  
2.5 The proposed framework scheme ......................... 21  

3.1 Profile drawings .......................................... 26  
3.2 Examples of three-quarter views ........................ 26  
3.3 Views of a rectangle in 2D .............................. 28  
3.4 Automatically generated views ........................... 31  

4.1 Reference set of views ................................. 35  
4.2 Surface Area ............................................. 38  
4.3 Projected Area ........................................... 38  
4.4 Projected Area Entropy ................................ 40  
4.5 Semantic Entropy ........................................ 41  
4.6 Visible Curvature ........................................ 43  
4.7 Curvature Entropy ....................................... 44  
4.8 Silhouette Length ........................................ 45  
4.9 Curvature estimation .................................... 45  
4.10 Silhouette Entropy ...................................... 46  
4.11 Optimal view sequences ................................ 48
Abstract

There are many possible two-dimensional views (or images) of a given three-dimensional object and most people would agree that some views are more aesthetic and/or more “informative” than others. It would be very useful, in many applications, to be able to automatically compute these “good” views.

Although all measures of the quality of a view will ultimately be subjective, hence difficult to quantify, we believe that there are some general computational principles that may be used to address this challenge.

Previous research on this topic dealt with only some aspects of this question. In our work we review these results and then present a general analysis of the problem. We propose a two-phase method for the computation of a “good” view. The first phase—view filtering—performs a global view-independent analysis of the object in order to generate a small number of candidate views. In the second phase we apply a view-dependent measure—a view descriptor—to select the most “informative” view among these candidates. For both phases, a number of heuristic methods were examined. We elaborate on these and show results.
Chapter 1

Introduction

Beauty is a form of genius—is higher, indeed, than genius, as it needs no explanation. It is of the great facts in the world like sunlight, or springtime, or the reflection in dark water of that silver shell we call the moon.

Oscar Wilde

In this introductory chapter, we start with the discussion of our general objective in Section 1.1. A survey of related work is presented in Section 1.2. The chapter concludes with Section 1.3, where we describe some assumptions about the inputs to the problem and the solution constraints.

1.1 The Goal

The real world consists of three-dimensional objects. The human visual system, however, is limited by optics to view only their two-dimensional images. Stereo vision and perspective only partially overcome this limitation. Thus, a significant component of the geometric information about a 3D object is lost during the viewing transformation.

This unfortunate fact is also reflected in traditional computer graphics applications, where we commonly see rendered 2D images. Although all the three-dimensional information about the shape is known \textit{a-priori} to rendering, much of it is lost when the shape is projected onto the image plane. The amount of preserved information depends on the particular \textit{view}—the position of the virtual camera relative to the shape. The aim of our work
is to find optimal, or nearly-optimal, views—those that maximize the visual information present in an image of a 3D object.

As the amount of accessible visual content continues to grow, and this content becomes more and more complex, there is an increasing demand for automatic non-interactive methods that are able to perform good camera positioning. This need is now understood by the research community, and consequently, the “good” view problem has gained much attention in recent years.

The notion of view goodness is subjective, and this constitutes the main obstacle when trying to formalize it. First of all, diverse viewers may demand different kinds of information in their view—the desired perspective often depends on the functional requirements presented by the particular visual task or application. Within an action-oriented framework, for example, the most relevant information describes how we interact with an object and how objects interact among themselves. Moreover, for many classes of objects the geometric structure itself is defined by their functional characteristics (e.g., mechanical tools, such as a screwdriver). The examples in Figure 1.1 illustrate that, in general, when the task is not precisely defined, it is virtually impossible to decide whether one view is better than another.

Besides this obvious task dependency, there is an aesthetic aspect to the question, which makes it difficult to address the problem scientifically. The decision may be quite subjective. For example, some people tend to prefer a more “beautiful” view, meaning one more pleasant and natural, such as the one depicted in Figure 1.1(c). Other observers might choose a view that is

![Figure 1.1: Different kinds of good views. Displaying a telephone: the view in (a) was chosen for the advertisement booklet, whereas (b) is an illustration from the operations manual. To the right, we compare an ordinary picture of an Eskimo in (c) with the well-known Eskimo-or-Indian picture in (d), which demonstrates an interesting effect of ambiguity in recognition. Determining which view in this case is better is a very subjective question.](image-url)
more challenging, or “interesting”, such as that in 1.1(d).

Nonetheless, despite these difficulties, we believe that there exists some common basis among all these issues. In particular, it seems that for many cases, the average human observer can easily make a clear distinction between “good” and “bad” views. Consider the examples in Figure 1.2—we doubt anyone would object to the classification proposed there.

In our work we are guided by the following rationale. Briefly, our objective is to answer the question: What is the “best photograph” that a person could take for the given object? Obviously, here we assume that the “photographs” are judged only by their visual quality, and not by any other more specific factor. Therefore, we think that the most suitable definition of a “good” view can be derived from this human-oriented task.

At first glance, this statement of the issue seems not so well-defined. Indeed, it is unlikely that two different persons would choose exactly the same viewpoint. However, it has statistical implications: instead of considering single viewpoints, we can analyze sets of viewpoints that have similar view directions, and associate a view with each of these sets. Then, in order for the “good” views to exist, we must be sure that similar viewpoints are consistently chosen by different people. Intuitively, we know that for many objects there are some views that are preferred by the majority, whereas other views are most often rejected. It turns out that this observation has a solid psychophysical background, as we will describe later.

Solving this problem presents a significant challenge in the field of visualization and shape understanding. A solution would be useful in many applications such as automatic camera positioning in CAD tools, thumbnail generation of large 3D object repositories, automatic scene composition, technical illustrations, and object recognition.

Figure 1.2: Good and bad view examples. These extremal cases show that the common sense classification may be quite unbiased. Pictures (a), (b) show good and bad views of the David head model, respectively. Pictures (c), (d) show good and bad views of the Elephant model.
1.2 Previous Work

The question “What is a good view of an object?” dates back to the Greeks and Romans. They proposed some simple rules of thumb, e.g., the golden ratio, the rule of thirds, the rule of fifths, etc [18].

In the 1930s, the mathematician G. D. Birkhoff [5, 6] tried to quantify the notion of an object’s beauty. He defined the beauty $B$ of an object as $B = O/C$, where $O$ is order and $C$ is complexity. He tested this formula on simple geometric figures, but was unable to provide a general notion of order and complexity. The main contribution of his work was an attempt to find the connection between the two diverse areas of aesthetics and mathematics.

Since the 1970s, much psychophysical research has been dedicated to the study of human visual perception. In this context, good views are presumed to be ones that make an object more readily recognizable by humans. In the well-known work by Palmer et al. [31], it was experimentally shown that for many objects there exists a small set of select views that are preferred by most people—these views are, therefore, called the canonical views of an object. The experiments were conducted to check the following kinds of visual tasks involving familiar objects:

(a) **Goodness Rating** Participants were asked to rate a set of different views of a given object, according to how “good” the object was depicted.

(b) **Visual Imagery** Participants were given the name of each object and were asked to imagine it mentally.

(c) **Active Selection** Participants used a camera to take the “best” photograph of each object.

The results of these experiments indicated that participants consistently preferred the same viewpoints independent of the task. Thus, evidence was obtained for the canonicity of a small set of views.

Additionally, as observed by Palmer et al. [31] and by Blanz et al. [7], canonical views often correspond to the off-axis “three-quarter views” of an object. According to Blanz et al. [7], canonical views are stable, and expose as many salient and significant features as possible. In computer graphics, Gooch et al. [18] tried to use the results of [7] to perform automatic scene composition, where finding the viewpoint was one of the three stages of the composition process (namely: image format, viewpoint, and shape layout). They started from a manually determined three-quarter view and then optimized it to find the most stable view. The approximate stability measure used in this method tried to eliminate coincident silhouette lines—more
specifically, the sum of squared distances between all silhouette midpoints was maximized, while silhouette occlusions were disregarded.

Many automatic object recognition methods in computer vision and robotics attempt to imitate theoretical models of perception. In this context, two main competing theories have been developed. The first one is object-centered, maintaining that human vision represents objects as 3D entities consisting of 3D components, and this representation is viewpoint independent ("Structural Description" by Marr [29], "Recognition-by-Components" by Biederman [4]). The second theory is viewer-centered "Multiple-View Description" (Koenderink and van Doorn [26], Bülthoff et al. [9]), stating that the object is best represented and processed as a set of 2D images from different viewpoints.

As described by Koenderink and van Doorn [26], all possible views of an object can be arranged in an aspect graph. The nodes of the graph correspond to equivalence classes of views, and edges join one node to another if the two differ by a single visual event. This way the nodes can be associated with stable aspects of an object, while the edges are associated with accidental views. Weinshall and Werman [43] gave a theoretical proof of equivalence between view stability and view likelihood for a given aspect and showed that this view can be computed from the aspect’s auto-correlation scatter matrix.

Tarr and Kriegman [39] conducted psychophysical experiments investigating the influence of the aspect of an object on the quality of recognition. Their experiments revealed that humans are indeed sensitive to certain types of visual events captured by the aspect graphs. Unfortunately, the set of all visual events is too large to be handled directly. The complexity of the aspect graph for line drawings containing $n$ lines has been shown to be $O(n^6)$, which is quite prohibitive. This is caused by the perfect resolution of the graph, which is actually impractical. For this reason, Eggert et al. [13] suggested the concept of scale-space aspect graph, which allows control over the level-of-details and thus can effectively limit the number of aspects.

In the field of computer graphics, to the best of our knowledge, the first results were published in 1988. Colin [10] developed a view optimization method for scenes modeled by octrees, which is based on a "direct approximate computation" paradigm. Each element of the scene is assigned a viewpoint-specific weight, which is the ratio between the visible part of the element to the entire element. Then six directions corresponding to the coordinate axes are examined and the three best directions are chosen. A good direction is computed in the pyramid defined by the chosen directions.

Kamada and Kawai [21] treated objects drawn in wire-frame in an orthogonal projection. A degenerate projection is one for which an edge is
projected to a point, or a polygon is projected to a line. The objective is to minimize such degeneracies, and the optimal viewpoint is called the *general position*. The complexity of the analytical solution is $O(n^3 \log n)$, but this does not take occlusions into account. The work of Gómez et al. [17] was similar in spirit to [21], incorporating perspective projections. Various “niceness” criteria were defined: regularity, simplicity, minimum crossing, and monotonic projections.

Plemenos and Benayada [32] suggested that a good view is that which maximizes the number of visible triangles and the visible projected surface area. These two measures are weighted and summed to an objective function and the optimal value is heuristically searched for by hierarchically subdividing the viewing sphere surrounding the scene. This measure does not take into account all the amount of *invisible* (occluded) surface area.

Barral et al. [3] added coefficients to the formula of [21] in order to cope with perspective projection. They introduced other heuristic balance coefficients, yet admitted that it is difficult to determine optimal weights for the different components. They also pointed out that often a single view (even if it is *good*) is not sufficient for scene understanding. They suggested generating a *good movie* by automatic movement of the camera around the scene. A heuristic method was proposed to generate such smooth camera paths, where the camera is driven by the previously derived cost function.

Vázquez et al. [40] extended the measure of Plemenos and Benayada [32] so as to operate on a per-face basis. This gives more detailed information about the view. A “probability” is associated with each face, defined as the fraction of its visible projected area relative to the total visible projected area. These probabilities are then combined using the information-theoretic entropy function. The cost function, called *viewpoint entropy*, is defined to be the entropy of this distribution. Hence, a good view is one for whom the faces are exposed as uniformly as possible. Note, however, that this cost function also does not take into account the behavior of the occluded surfaces. Additionally, Stoev and Straßer [37] pointed out that the method of Vázquez et al. generates “flat” views for scenes where all the normals point in similar directions (e.g., digital terrain models). As a result, the good view direction (maximal viewpoint entropy) typically is vertical; hence, although the number of visible triangles and the projected area are maximized, most depth information is lost. Their proposed workaround is to add to the cost function another (weighted) term that measures the maximal depth in the frame.

Sokolov and Plemenos [36] suggested a curvature-based measure for the viewpoint quality, where the total mesh curvature visible from a viewpoint is considered. A paper by Lee et al. [27] defined the most informative view
as that which maximizes the visible saliency of an object. The saliency is defined using a multiscale curvature measure. A gradient-descent algorithm is used to optimize this measure over the viewing sphere.

Bordoloi and Shen [8] and Takahashi et al. [38] solved the optimal view selection problem for volumetric rendering. The first measure is based on the information entropy composed of the transfer function, the data distribution and the visibility of the voxels. The second method employs interval volumes and their combinations that characterize the topological transitions of isosurfaces according to the scalar field. The globally optimal viewpoint is found by a compromise between the locally optimal viewpoints for each of the components.

Another class of problems closely related to ours is the optimization of lighting parameters. However, it seems that these type of problems are easier to address, and several well performing methods have already been proposed. Shacked and Lischinski [34] defined a perception-based quality function, formed by a weighted linear combination of six terms. Brute force optimization is used to find the optimal configuration. Gumhold [19] suggested the lighting entropy measure, which calculates Shannon entropy from the brightness of the pixels. Another entropy-based metric was proposed by Vázquez and Sbert [41], which is applied to the colors of the pixels in CIE LUV format.

In the field of robotics, Arbel and Ferrie [2] and Roberts and Marshall [33] attempted to find good views that simplify object recognition. The approach in [2] was based on a learning process for entropy maps on the viewing sphere, where each entropy value indicates the expected ambiguity of recognition. During the recognition process, these maps are navigated in order to minimize the chance of expected ambiguities. Roberts and Marshall [33] selected a minimal number of views that allow adequate representation for every face of the object. In the context of image-based rendering (IBR), Fleishman et al. [16] searched for views that are suitable for the set of reference images, where the total coverage could be achieved. They required that every polygon in the scene must be seen in the coverage at a sufficient sampling rate. The quality criterion is based on the projected area.

1.3 Problem Setting

In this section we describe some assumptions about the structure of the input and the restrictions we want to apply on the set of potential solutions.

An input scene, in general, is a collection of geometric objects. Some previous techniques (e.g., see [40]) claim to be able to handle general scenes
containing more than one object, either indoors (such as an interior of a room), or outdoors. We think that the declaration of success is premature, since very few positive results have meanwhile been obtained even for single objects. Typically, a complex scene implicitly contains various semantic relations among its objects—as a very simple example, the objects might be ranked by their relative importance—and this obviously should be reflected in the resulting image. These semantic attributes are usually completely independent of the scene’s geometric properties and some external knowledge is required in order to quantify them. Therefore, we find it practical to restrict our attention exclusively to handling standalone objects.

Furthermore, we expect that the input objects will have a well-organized spatial shape. Although nonsense objects are an interesting research target, in particular in the study of human perception, we must acknowledge that they are never encountered in real life. In computer graphics, we generally deal with objects that are familiar, or at least have a close resemblance to some familiar objects. Specifically, in our work we assume that all input objects have some object-centered coordinate frame, which can be naturally defined—the reasons for this assumption will become evident from the discussion in the next chapter (see Section 2.1).

For a given object, the image changes as a result of variation in two parameters: the viewpoint and the illumination. Theoretically, to find an optimal configuration, these factors should be considered as changing simultaneously. However, this degree of freedom seems to be superfluous—we feel that the main focus should be concentrated on the geometric structure of the object, while the attributes of lighting play a secondary role (see Figure 1.3 and accompanying remarks). On the other hand, as we have already mentioned in Section 1.2, there exists a number of techniques that are able to compute the optimal light placement for a fixed viewpoint ([34, 19, 41]). Consequently, we require that the optimal viewpoint should be illumination invariant, that is, to be independent of any lighting conditions. This way, light placement may be postponed until the optimal viewpoint is found, and then the lighting can be computed from this viewpoint. We also notice that in many cases the simple “headlight” setting is sufficient—this means that a single light source is placed at the viewpoint.

Finally, we address the properties of the solution space—the set of all possible viewpoints. Without loss of generality we assume a perspective projection. Then a viewpoint is parameterized by its position \( p \) in 3D space and two unit vectors: the line-of-sight direction \( d \) (two degrees of freedom) and the up-vector \( u \), which is orthonormal to \( d \) (leaving it only one degree of freedom). Thus the resulting search space is potentially six-dimensional. For our purposes we can, however, reduce this unnecessary freedom when
Figure 1.3: Effects of different lighting. A typical problem of non-optimal lighting is that it overshadows details of an object, as happens for the body of the car model on the right. This leads to the loss of information that could otherwise be preserved for a given view.

choosing a viewpoint if we use the conventional viewing sphere model (e.g., see Koenderink [25]).

We bound an object with a sphere and then place viewpoints at the constant distance from the center of the sphere. The minimal distance is chosen so that the sphere would be completely inside the field-of-view of the camera (thus it is not clipped by the edges of the resulting image). The viewing direction is set to point directly toward the center of the sphere.

In this way the viewpoint is completely parameterized by its direction $d$, and the up-vector $u$, as shown in Figure 1.4(a). It should be noted that under the perspective projection, moving the camera in the plane, which is orthogonal to $d$ (also known as “panning”), may change the visibility of some features, but we assume that this effect is negligible.

The unit up-vector $u$, as follows from its name, defines an up direction of the resulting image. It has one degree of freedom that corresponds to the two-dimensional rotation of the image around its optical axis (see Figure 1.4). In the quantitative sense, this rotation does not change the visual information presented by the image. However, for many inputs, nature dictates its own

Figure 1.4: Viewpoint parametrization. (b) and (c) denote different image orientations caused by varying the direction of $u$. 

11
up direction, which should be respected in any view. So, for example, an animal should not be rendered upside down, but rather standing on its feet. For the time being, automatic image orientation is an open problem, and it is not considered here. We assume that all views can be caused to have the correct orientation after the optimal view has been computed.

Summing up, in our problem we search over the space of all possible unit directions $d$. This is topologically equivalent to a sphere, therefore, it is called the viewing sphere.
Chapter 2

Problem Analysis

What is it indeed that gives us the feeling of elegance in a solution, in a demonstration? It is the harmony of the diverse parts, their symmetry, their happy balance; in a word it is all that introduces order, all that gives unity, that permits us to see clearly and to comprehend at once both the ensemble and the details.

HENRI POINCARÉ

In this chapter we describe the general considerations that lead us to define the strategy for our solution.

In Section 2.1 we show that the well-known canonical views provide an adequate formalization for our objective. The essential characteristics of these views are described in Section 2.2. Based on these characteristics, Section 2.3 analyzes the reasons for possible drawbacks of the existing methods. Finally, a new framework is proposed in Section 2.4.

2.1 Good Views and Canonical Views

In order to be able to solve our problem we must answer two major questions:

1. What is the formal definition of a “good view”?

2. How can these views be automatically calculated?
We start with a detailed investigation of the first question. As mentioned in Section 1.1, we have chosen the “best photograph” approach, which is based on informal human judgement. To be more precise, to arrive at the “best photograph”, we use the following hypothetical procedure:

A large group of people is given an input object $O$, which can be thoroughly examined by everyone in the group. Afterward, each individual is asked to take the “best” picture of $O$ with a still camera, and the information about all chosen viewpoints is gathered. Then this information is statistically processed in order to identify clusters of viewpoints on the viewing sphere. The “good” views are defined to be the centers of the resulting clusters.

These “good” views are no more than the canonical views of $O$ that were discovered by Palmer et al. [31] (see Section 1.2). Often there is more than one canonical view for a given object. Since these views have statistical meaning, there is no clear preference of one such view over another. This indicates that usually there is no such thing as the “best” view, and one can talk only about “good” views. Thus the issue at hand becomes the problem of finding some view from the set of the canonical views, and we denote this view as a “good” view of an object.

For a wide variety of objects, the canonical set is unambiguous and well-defined. Psychophysical research shows that, typically, there is a tight connection between the canonical set to the canonical coordinate system of a given object. The canonical coordinate system is the object-centered coordinate system that may be naturally determined from the object’s salient geometric properties, such as natural axes of elongation, symmetry, etc. ([29]). For example, consider the teapot model in Figure 2.1, whose left side, right side, top and bottom can easily be identified. Moreover, an object may have more than one natural coordinate frame, as happens for a torus or a complicated mechanical part; yet, in these cases it is usually no problem to choose one coordinate system due to symmetry considerations, or, if the number of such coordinate systems is limited, to regard each one in a sequence. For this class of objects the canonical views usually exist and are well-defined. On the other hand, for some objects there is no canonical coordinate frame at all, for example a crumpled newspaper. It has a large number of poorly defined axes. As a result, such an object is likely to have ill-defined canonical views, and there is no reasonable definition for a “good” view. Therefore, we assume that our input is limited to objects for which at least one canonical frame exists. In practice, most regular objects are included in this range.
Figure 2.1: Canonical coordinate system for a teapot. (b)–(e) are axis aligned views.

So far we have more fundamental insight into the nature of good views. Nevertheless, it is not clear how these views may be calculated automatically. The definition of the canonical views is not mathematical. Moreover, it involves “men in the loop”.

Despite of these problems, we can think at least about one method that may employ the direct definition of good views. This method is based on the concept of similarity and generalization. Suppose one already knows the canonical views for a model of a horse. It seems reasonable to assume that the same views will be valid also for models of a cow, an elephant, a cat, and generally for all four-legged animals. If a new model of an animal is introduced, it may be simply assigned the canonical views of the horse. So theoretically, the procedure may be as follows:

As a preparatory off-line stage, a sufficiently rich database of 3D objects is established. For each object in the database, its canonical views are determined experimentally (for example, via a web poll) and are stored together with the object.

Then, for an input object, we may use one of the existing techniques for shape matching (e.g., [14]) in order to find the closest match in the database. Usually, shape matching algorithms also contain an aligning step for normalization purposes. After two objects have been aligned, the canonical views of the object that was found in the database may be assigned directly to the input object.

This almost straightforward approach is impractical for many reasons. It is too cumbersome and still requires much human intervention in its off-line
It is obvious that a more simple and more “analytical” solution is desirable. To be fully automatic, the solution should rely only on the geometric properties of an object and should not involve human intervention, if possible. For this we must analyze the objective properties of the canonical views to determine what makes them so desirable in contrast to others.

2.2 Canonical View Properties

Even when we know that canonical views exist, their properties are difficult to determine. Perhaps the only way is to analyze many experimental results in order to derive them empirically.

Since canonical views were discovered, very few attempts have been made to recognize the object attributes that determine them. In the psychophysical literature, the most notable is the work of Blanz et al. [7]. In computer graphics various propositions have also been submitted, mostly in the field of perception-based viewpoint optimization. Summarizing this research, at the broadest level the three most important characteristics of canonical views that can be distinguished are: representativeness, beauty and detail. We will now describe each of these aspects in depth, in the order of their apparent importance.

Representativeness (Ease of Recognition) This is probably the most important characteristic of a good view. Looking at an object, the observer must first of all, by simply glancing at it, be able to grasp what he is looking at, and understand to which general class the object belongs. Is it human, animal, or mechanical part? At this stage any auxiliary details are likely to be ignored, e.g., whether the tail of the animal is visible or not.

For objects having a natural coordinate frame, it seems crucial that this frame should be evident in the picture, at least in a coarse manner (see Marr [29]). As a very basic requirement for this, we demand that true object proportions must be preserved in a view. Consider, for example, two views of the same elongated box in Figure 2.2. In the first image, the proportions of the box are preserved and this causes no difficulty in recognition. In contrast, the second image will give the viewer the false impression that a cube is being shown, due to the fact that the depicted dimensions are equal.

Beauty of a View Aesthetic criteria play an important role in deciding which views are preferred. Obviously, aesthetically pleasant views are
considered to be more appealing and cause more visual appreciation. According to Birkhoff [6], the beauty of a view may be measured by a mathematical formula $B = O/C$, where $O$ is an “order” in a picture and $C$ is its “complexity”. By the notion of “order” one often means the balance, symmetry and salience of important shape features. Moreover, some additional ideas can be borrowed from art studies, where different rules of thumb exist (for example, the “golden ratio”, rule of thirds, etc.) For a view to be beautiful, it must not be over-complicated, e.g., must not be burdened with a large amount of irrelevant details. An example of a beautiful view for a torus, judged by Birkhoff’s measure [6], is given in Figure 2.3.

A beautiful view should always be stable with respect to small viewpoint transformations. Accidental views, which are defined as unstable views in which even a small change of the viewpoint causes a great qualitative change in the image, cannot be considered beautiful. This requirement is similar to the notion of temporal coherence in computer graphics, which is considered one of the most important characteristics

Figure 2.3: The most “beautiful” view of a torus according to the Birkhoff’s measure. This view can be effectively described by two concentric circles, so it is the most symmetric and the least complicated one.
for good animation sequences (e.g., see DeCarlo et al. [11]).

**Detail** A good view should be interesting. Presumably, after the observer has made his/her first coarse identification of an object, he/she may want to concentrate on smaller details. The mental effort that is needed for the concentration is proportional to the number of these details. It is conventional to think that more detailed pictures attract more viewer attention, leading to an impression that these pictures are more interesting.

In addition, a good view should be informative. There is an inherent loss of information in each particular view opposed to the 3D structure of an object. Object features may be made indistinct due to the degeneracies of the projective transformation, or they could be partially or completely obscured by other parts. By maximizing the number of details in a view, more information can be preserved.

Obviously, these general characteristics are not fully independent of each other. For a view to be representative, the most salient shape features should be exposed, so that a direct relation to the detail property exists. On the other hand, there is somewhat contradictory requirement as to the number of details for an aesthetically beautiful view. Nevertheless, for a good view, all three properties should be satisfied, albeit with necessary trade-offs between them.

### 2.3 Possible Pitfalls

As we have seen in Section 1.2, a number of techniques have already been proposed for solving the “good” view problem. In spite of the fact that every method presents nice results, it has became apparent that as yet, no comprehensive solution exists. In this section we summarize what all these methods share in common and analyze the reasons for their possible drawbacks.

The most straightforward way of finding a “good” view is to represent it as an optimization problem. For this, an objective function $D(v)$ is defined, whose purpose is to describe the quality of a view for each possible viewpoint $v$ on the viewing sphere $S^2$ by a real number. We will call this function a view descriptor. Then a “good” view is found as a solution to an optimization problem

$$\arg \max D(v)$$

on the sphere. Depending on the complexity of the descriptor $D$, this optimization can be solved either exactly (as in Kamada and Kawai [21]), or
approximately, by an exhaustive or heuristic search over the target sphere (e.g., Lee et al. [27]).

There are several potential problems with this approach. The main one is typically in the design of the view descriptor \( D \), which is the core of all the methods. The descriptor should be general enough to capture the desired properties of good views, and it should map all these properties to a single real number in some analytic way.

In one scenario, attempts are made to synthesize such a universal descriptor exclusively from some specific kind of geometric attributes, such as projected area ([3, 40]) or surface curvature ([27]). However, it is unlikely that a simple geometric property alone may appropriately reflect the different factors outlined in Section 2.2.

Additionally, descriptors often involve the notion of “details” of an object. Indeed, the primary purpose of a “good” view is to convey more 3D information to an observer. In the scientific community information is usually associated with “details”, probably due to the well-known results in information theory. Nonetheless, the question of what is the right scale of details to be identified remains. Can triangles, for example, be considered as “details”, or should they be grouped together to form a larger “detail”? Marr in [29, p. 270] discussed these difficulties, in relation to image segmentation problem:

> What is an object, and what makes it so special that it should be recoverable as a region in an image? Is a nose an object? Is a head one? Is it still one if it is attached to a body? What about a man on horseback?

> These questions show that the difficulties in trying to formulate what should be recovered as a region from an image are so great as to amount almost to philosophical problems. There really is no answer to them—all these things can be an object if you want to think of them that way, or they can be a part of a larger object. Furthermore, however these questions were answered in a given situation did not help much with other situations.

A partial solution might be to use a multiscale measure, similar to the curvature-based one by Lee et al. [27], but still this does not appear to be enough. The problem is evident from the example in Figure 2.4, where the optimal view with respect to the mesh saliency measure [27] is shown. While this picture exposes as many salient features as possible, the overall head proportions are strongly distorted.

Other researchers recognized these weaknesses of the simplistic descriptors and tried to find a compromise by adding missing factors to the de-
All these “bug fixes” usually introduce some kind of amendment coefficients. For example, Barral et al. in [3] attempted to blend together the measure of non-degeneracy proposed by Kamada and Kawai [21] (which they called Kamada’s coefficient) and an additional “balance coefficient,” which is responsible for a proper area balance between the faces (more exactly, it is the standard deviation of the series produced by the ratio of the face’s projected area to its real area in 3D). In another case, Stoev and Straßer [37] observed that the views of maximal viewpoint entropy, as proposed by Vázquez et al. [40], often result in unacceptably flat pictures, so they tried to repair the problem by adding an additional “frame depth” coefficient to the existing formula. Furthermore, in order to assemble various coefficients into a single expression, their proper weights should be somehow determined, and this is usually done in an empiric manner. The bottom line is that all these facts make us uncomfortable with such descriptor definitions. Indeed, it seems unlikely that Nature may be dealing with so many “unnatural” factors.

Finally, even if we assume that a suitable descriptor definition has been already set up, there remains the important question as to how the optimal (maximal) solution can be found. The set of all possible viewpoints, which is topologically equivalent to a unit sphere $S^2$ (a viewing sphere), comprises an infinite number of points.

For very simple descriptors, in particular for those that disregard visibility and occlusions such as the one of Kamada [21], an exact solution can probably be found using computational geometry methods. Yet, this class of “visibility-independent” descriptors is too limited to provide a comprehensive solution, since obviously the quality of a view depends mostly on information that is present in a picture, and so is explicitly visible. Therefore, we feel it
necessary to require that the final solution must take visibility into account.

Analytical computation of visibility is a complicated problem. Existing object-space algorithms, e.g., Appel’s well-known Quantitative Invisibility [1], are extremely inefficient, and so are almost never used nowadays. Thus it seems hopeless to find an exact algorithmic solution for visibility-aware descriptors. For this reason, various search procedures are typically used to find an approximate answer.

The first possibility is a brute-force exhaustive search for a global maximum over the viewing sphere. For this search to be accurate, sufficiently dense sampling of the sphere must be made, obviously with a strong impact on the total performance. The other possibility is to apply a heuristic search, as it is prevalently done in many tasks in artificial intelligence. This usually improves the performance, but there is some risk that the search can get stuck in a local maximum, never getting to the globally optimal one. Both these approaches are problematic, so in our opinion it is highly undesirable to use a standard search procedure for solving the optimization problem.

In summary, the main disadvantages of the previously proposed methods are:

• Limitedness of descriptors;
• Failure to handle an input object at different scales;
• Inefficiency of descriptor optimization.

In the next section we suggest a solution that attempts to deal with all these issues in a single framework.

2.4 Proposed Framework

So far we have seen that a comprehensive solution should be able to handle an input object both at coarse and fine levels, and there is little chance that a single view descriptor function will be sufficient for this purpose. Therefore, we propose to build a framework that solves the problem in two stages, as depicted in Figure 2.5.

```
3D model → View Filtering → View Descriptor → Output view
```

Figure 2.5: The proposed framework scheme.
We introduce an additional view filtering phase as the first stage of our method. Given a 3D shape as input, this filtering procedure calculates a small number of candidate views for further processing. Thereafter, we proceed as usual, rating the candidate views by the view descriptor of choice $D$. Finally, we output the view with the highest score as being the “best” one. Moreover, more than one good view may be selected as the “best” one, in the case there are views for whom the values of $D$ are sufficiently close to the maximal one.

The first and the most important purpose of this separation is the division of functionality between the stages. An object is handled consecutively on two extreme levels of the scale. The responsibility of the view filtering stage is to handle the input at the most coarse level, whereas the descriptor stage is responsible for the finest level. Moreover, there is a clear order of precedence in this flow, giving the main decisive power to the filtering stage. Besides that, there is an obvious benefit in that the search space is limited to a small number of possibilities, resulting in a significant speed-up of the subsequent search procedure. It should be noted here, however, that this reduction is intended to restrict possible views, with no relation to scores that are further given by a descriptor $D$. This should not be misinterpreted as an enumeration of all potential points of maxima for a function $D$, as is sometimes done in computational geometry.

View filtering performs an a-priori global analysis of high-level shape properties in 3D, such as the object’s dimensions or its general orientation in space. This analysis can also be called view-independent, since it does not rely on some particular viewpoint. We assume that at such a high level, no superior precision is needed, and so approximate estimations can be made. In particular, the visibility issues are typically ignored at this stage (but are taken into account by a view descriptor in the following stage). Referring back to Section 2.2, the filtering is usually responsible for the most prominent characteristics of a good view, namely its “representativeness” and its “beauty” (in particular, global view stability).

The design goals for a view descriptor are apparent from our previous discussion. Furthermore, for now the scope of a descriptor’s responsibility may be freely limited in order to concentrate on the fine-grained features of a shape, up to some necessary minimal level. In terms of Section 2.2, the purpose of this stage is to maximize the “detail” property of a view. At this point each candidate view is examined in order, and in this sense a descriptor should be view-dependent. Obviously, in a good view the visibility of an object’s parts should be respected, so we require that a descriptor should be able to handle it accurately. Since the number of views is limited, and view parameters are fully determined, accurate calculations pose no problem.

In our work we have tested this approach in a number of configurations.
More specifically, a method to perform *view filtering* had been proposed. Then, a few possible candidates for a *view descriptor* were examined, every candidate being tested under the proposed filtering scheme. In Chapters 3 and 4 we provide a more detailed explanation of these experiments.
Chapter 3

View Filtering

Although this may seem a paradox, all exact science is dominated by the idea of approximation.

Bertrand Russell

The quality of a view depends crucially on its ability to properly express the general shape of an object. This goal is achieved by a view filtering procedure, which performs a global analysis of 3D shape geometry. The calculation provides an approximation, and in particular, does not account for occlusions.

In this chapter, we propose a filtering scheme based on an approximation of the so-called “three-quarter views” of an object. This idea is inspired by the empirical observation made by Blanz et al. [7] that, for many objects, their canonical views correspond to “three-quarter” views of these objects. Hence in order to estimate the canonical views, we are interested in finding these “three-quarter” views.

In Section 3.1 we provide a short exploration of the origins and the meaning of the “three-quarter” terminology. Then, this term is formally defined for a very simple case of a 3D box, as described in Section 3.2. Finally, an extension for more complex objects is proposed in Section 3.3.

3.1 Three-quarter Views

The concept of three-quarter view originates in art, mainly in portrait drawing. In ancient times, portraits were drawn in straight position, either a full
face or a profile (perhaps the latter being more popular). There are numerous examples for this type of drawings, ranging from Egyptian rock carvings to Greek vase paintings, as can be seen in Figure 3.1.

![Figure 3.1: Ancient drawings. The profile pose was usually preferred.](a) Rameses III (b) Theseus and the Minotaur

Cimon of Cleonae was probably the first Greek painter who broke with this tradition. He started experimenting with three-quarter, or “foreshortened” views, where the pose of a subject was turned away from the straight views. This technique became even more popular in the Renaissance period. One example is Leonardo da Vinci’s painting named “Head of the Virgin in Three-Quarter View Facing Right” shown in Figure 3.2(a). Artists usually claim that this pose breaks the barrier between the subject and the viewer, allowing the subject to “look out” from the portrait toward the viewer. They also observed that these views are more balanced than profiles or frontal ones.

Later, this type of view was adapted to other kinds of drawings, not

![Figure 3.2: Three-quarter views. (a) courtesy of the Metropolitan Museum of Art, New York.](a) (b) (c) (d) (e)
only to portrait paintings, and “three-quarter” poses have been associated with objects other than heads of persons. Quite naturally, the same term is commonly used in modern photography. Some real-life pictures in three-quarter views are shown in Figures 3.2(b) – (e).

We were able to find a few informal definitions for three-quarter views as they are used by artists. The following list summarizes these definitions:

(a) A slightly turned pose;

(b) A view of a face which is halfway between a full and a profile view;

(c) A portrait on which the near portion of the face up to the middle of the nose occupies roughly three quarters of the picture, while the other part is a quarter of the picture.

It can be seen that some definitions are more clear-cut than others. Furthermore, the definition in (c) can probably explain the origin for the “three-quarter view” terminology. It is also worth noting that the camera should be rotated both horizontally and in elevation, as is especially evident from Figures 3.2(d) and (e). Moreover, for the cup in Figure 3.2(d), the horizontal rotation is irrelevant due to its rotational symmetry.

To compute three-quarter views automatically, precise mathematical definitions should be given. We start by examining the simplest case first, and then extend our definition to more general objects.

### 3.2 A Simple Case Study

Three-quarter views can be defined quite naturally if we consider a simple three-dimensional box, such as the one depicted in Figure 2.2.

First of all, we observe that for a box we can easily produce a canonical coordinate frame. This object-centered Cartesian frame has axes parallel to the box’s edges, and its origin is located in the center of the box. Actually, the canonical frame defined this way is not unique, since there is still a degree of freedom for assigning axis labels ($x$, $y$ or $z$) and for choosing between two opposite axis directions. As we will see shortly, our computation is indifferent to any choice among these 24 possibilities, yet we pick some of them as a reference frame without loss of generality.

We define *straight views* to be axis-aligned viewpoints—that is, the viewpoints that are located on the coordinate axes. There are six such views, namely the vectors $(±1, 0, 0)$, $(0, ±1, 0)$ and $(0, 0, ±1)$. Intuitively, straight views correspond to a frontal view, a profile (or side), and the back, top and
bottom views of an object. Clearly, for a box the straight views are those exposing exactly one of its six faces.

Finally, three-quarter views can be defined as being exactly halfway from the straight views. More precisely, these are the eight combinations of the straight-view vectors: \((\pm 1, \pm 1, \pm 1)\), or in normalized form \((\pm \frac{1}{\sqrt{3}}, \pm \frac{1}{\sqrt{3}}, \pm \frac{1}{\sqrt{3}})\). One of the eight three-quarter views of a box is shown in Figure 2.2(a).

These views have several nice properties. One such property, most important for our purposes, is that the proportions of a box are preserved under parallel projection. Consider a box having dimensions \(a \times b \times c\), and its image, where the projected edges have lengths \(a_p\), \(b_p\) and \(c_p\) respectively. Then, for a three-quarter view, the ratio \(a/a_p = b/b_p = c/c_p\) is preserved, namely the following equations hold: \(a/a_p = b/b_p = c/c_p\). This is due to the fact that the edges are projected onto the image plane under the same projection angle \(\alpha = \arccos(\frac{1}{\sqrt{3}})\).

![Figure 3.3: Views of a rectangle in 2D. (a) is the three-quarter view. (b) is diagonally-aligned view.](image)

It is worth noting that the three-quarter view, as it is defined above, does not depend on the particular box’s dimensions \((a, b\) or \(c)\), but only on the orientation of its coordinate axes. This is best illustrated in Figure 3.3, where for simplicity the 2D case is considered. Instead of a box, we have a planar rectangle \(a \times b\), which is projected onto the “image line” \(\pi\). View directions are depicted by a black arrow. In the first case in Figure 3.3(a), a three-quarter view \((-1, -1)\) is shown. Since the right-angled triangles \(ax\) and \(by\) are similar, the relation \(a/x = b/y\) holds. In the second case, in Figure 3.3(b) we consider a view direction that depends on the rectangle dimensions, namely the vector \((-a, -b)\) that is aligned with the diagonal of the rectangle. Here both projected edges are of equal length \(x\), as can be clearly seen if we draw the height of the upper diagonal-based triangle, and then observe that the resulting small triangles are equal. In this case, the
“image” will probably lead to a false conclusion that the projected rectangle was originally a square. This is exactly a 2D equivalent of the 3D effect that was discussed in Figure 2.2(b), where the view in the direction of the box diagonal was used.

Yet another property of the three-quarter views of a box is related to view stability, which is essential for a good view.

An aspect of an object is the set of views that expose the same set of the object’s features. Following Weinshall and Werman [43], the flattest view of an aspect is the one that is obtained from the viewing direction along which the three-dimensional shape of the aspect has its minimal spread. For a box, being a convex object, we are interested in the aspects where three faces of the box are visible (and correspondingly, the set of the box’s vertices remains the same); there are exactly eight such aspects.

First, we notice that for the chosen aspects, the flattest views correspond to the three-quarter views of the box. Then we use the results from [43] that prove the equivalence of the flattest and the most stable views in the given aspect (where stability is defined with respect to the 2D image similarity metric). So transitively we conclude that our three-quarter views are the most stable views of the box in the corresponding aspects. Moreover, in a similar manner, these views are also the most likely views of the box (using the notation of [43]).

In summary, for this very simple case, the three-quarter views seem to be an ideal choice for the canonical views.

3.3 Handling a General Object

In this section we extend the previously developed definition of the three-quarter views to be applied to more general type of objects. The basic idea is to approximate an entire object at its very coarse level by an oriented three-dimensional box. Note that this probably corresponds to the lowest acceptable “resolution”, or level-of-details of the object for our task.

In order to estimate the natural Cartesian coordinate frame for an input object, we apply Principal Component Analysis (PCA, [20]) to the object geometry. This statistical technique is commonly used in numerous computational fields such as signal processing, face recognition and pattern analysis. Moreover, it has been successfully used in computer graphics for object matching algorithms as an aligning and normalization method (e.g., see Elad et al. [14, 15]).

The primary purpose of PCA is to provide a compact and optimal description of a data set. It calculates a linear transformation that chooses a
new coordinate system for the data set such that the largest variance by any projection of the data set lies along the first axis (called the first principal component), the second largest variance is on the second axis, and so on. Due to its statistical nature, PCA is fairly robust and stable in the presence of noise.

Next we describe the standard procedure for PCA calculation as it is applied to the object 3D geometry. In our implementation we use geometry sampled at the object vertices \( \{v_i\}_{i=1}^n \), where \( n \) is the total number of vertices. For fairly tessellated meshes this point cloud provides quite precise description of the overall object geometry.

Prior to the calculation, the centroid of the points should be brought to the origin (so that the resulting set will have zero mean value). This is done by subtracting the centroid \( \bar{v} = \frac{1}{n} \sum_{j=1}^n v_j \) from every original point \( v_i \). We get a translated set of points \( \{p_i\}_{i=1}^n \), where \( p_i = v_i - \bar{v} \), and the resulting coordinates of \( p_i \) are \((x_i, y_i, z_i)^T\).

Then the column vectors of these data points are organized into a \( 3 \times n \) matrix \( P \):

\[
P = \begin{pmatrix} p_1 & p_2 & \cdots & p_n \end{pmatrix} = \begin{pmatrix} x_1 & x_2 & \cdots & x_n \\ y_1 & y_2 & \cdots & y_n \\ z_1 & z_2 & \cdots & z_n \end{pmatrix}
\]

The covariance matrix \( C \) is a symmetric \( 3 \times 3 \) autocorrelation scatter matrix:

\[
C = \frac{1}{n} PP^T = \frac{1}{n} \begin{pmatrix} \sum x_i^2 & \sum x_i y_i & \sum x_i z_i \\ \sum y_i x_i & \sum y_i^2 & \sum y_i z_i \\ \sum z_i x_i & \sum z_i y_i & \sum z_i^2 \end{pmatrix}
\]

Since \( C \) is symmetric, it can always be diagonalized with an orthogonal matrix, so that the set of its eigenvectors \( \{e_1, e_2, e_3\} \) forms an orthogonal basis in \( \mathbb{R}^3 \). This enables us to choose an object-centered coordinate system with the axes defined by these eigenvectors, which are the principal components of \( P \). Intuitively, the orientations of the axes describe three orthogonal principal “dimensions” of the object. In the geometric context, the first principal component \( e_1 \) is sometimes called the primary axis of elongation. Furthermore, this choice of coordinate axes is optimal in the least-squares sense; more formally, \( e_1 \) is the solution of the optimization:

\[
e_1 = \arg \max_{||e||=1} \sum_{i=1}^{n} (e^T p_i)^2,
\]

which minimizes the average squared distance to any axis that passes through the centroid.
The computational complexity of PCA is extremely good. The covariance matrix $C$ is computed in $O(n)$ time, and the diagonalization is performed in $O(1)$ time, bringing the total complexity to $O(n)$. Moreover, only $O(1)$ additional memory is used.

In this new coordinate system $\{e_1, e_2, e_3\}$, the definitions of the designated views are identical to those we gave in Section 3.2. Straight views are defined as six axis aligned views $\pm e_1, \pm e_2, \pm e_3$. Similarly, the three-quarter views are eight combinations of the straight views $(-1)^i e_1 + (-1)^j e_2 + (-1)^k e_3$, where $i, j, k \in \{0, 1\}$.

Since we think of a box as a coarse approximation of an input object, hopefully the properties of the three-quarter views for a box will be approximately preserved for the object itself, at least at the highest level. Two such properties—proportion abidance and view stability—were discussed in Section 3.2. We also note that the approximation method we used here is similar to the “direct approximate computation” proposed by Colin [10].

According to our empirical observations, this simple technique leads to surprisingly good results for many input objects. Actually, some examples of automatically calculated straight views for the teapot model have been already given in Figures 2.1(b) – (e). The complete set of the three-quarter views that was generated for the teapot will be shown in the next chapter, where the experimental results are presented (see Figure 4.1). Notice that since the orientation of the coordinate frame is properly estimated by the PCA decomposition, the object’s symmetry is correctly presented. One can observe pairs of almost symmetric views, where the counterparts are mirrored versions of each other.

![Figure 3.4: Selected three-quarter views that were automatically generated by our method.](image)
After the view filtering phase we are left with a small number of views (eight in three-quarter filtering) that approximate the representative set of views at the most coarse level of detail. The remaining question is which one of these should be chosen as the “best” view.

The purpose of a view descriptor is to rank filtered views by measuring the amount of information available from these views, as outlined in Chapter 2. Two important characteristics of a view descriptor are:

(a) The ability to exploit a given object’s fine features;

(b) Its value should be accurate; in particular, the visibility of the features should be taken into account.

In this chapter, we describe a number of different candidates for view descriptors. The design of our descriptors is based on a number of principles. The first principle is to exploit an accepted measure of geometric complexity for a 3D shape. This could be based on various features in the shape, its surface area, its curvature distribution, etc., and is obviously view-independent. The view descriptor would then assign to a view a score that is the feature’s contribution to the complexity from the portion of the shape that is visible in that view (see Sections 4.1 Surface Area, 4.2 Projected Area, 4.3 Projected Area Entropy, 4.5 Visible Curvature and 4.6 Curvature Entropy). So, in effect, the best view is that which exposes as much of the geometric complexity of the object as possible.
The second principle is to define descriptors that are based on inherently view-dependent features, such as object silhouettes (see Sections 4.7 Silhouette Length and 4.8 Silhouette Entropy). Here again we would like to expose as much of these features as possible.

The third principle is to build a descriptor that, instead of assigning values to the primitive elements of the 3D model (e.g., vertices, faces, and edges), assigns values to larger portions of the model that have some semantic meaning. Such model portions may be obtained from segmentation algorithms (see Section 4.4 Semantic Entropy). This affords a higher level visual appreciation of the model.

Another issue is the way in which the values of primitive elements are combined to form a single value of a descriptor. Typically we use one of two methods, depending on the design objective. The first method is to simply sum the values of the primitives. In this case we assume that each value directly describes the informational contribution of the corresponding primitive, and the total information in the image is accumulated. A more sophisticated method is to use the information entropy formula of Shannon [35]. For a discrete probability distribution of \( n \) events \( D = \{p_1, p_2, \ldots, p_n\} \), where \( p_i \geq 0 \) and \( \sum_{i=1}^{n} p_i = 1 \), the Shannon entropy is defined as:

\[
H(D) = -\sum_{i=1}^{n} p_i \log_2 p_i.
\]  

(4.1)

For a fixed \( n \), the maximal value of \( H(D) \) is achieved when \( p_1 = p_2 = \cdots = p_n \) and it equals

\[
H\left(\left\{\frac{1}{n}, \ldots, \frac{1}{n}\right\}\right) = \log_2 n;
\]

that is, the entropy value is maximal when the distribution is uniform. It also follows that if uniform distributions are considered, then \( H(D) \) increases when the number of events \( n \) becomes larger. Therefore, this measure is useful when we strive to get a uniform distribution of the values measured on the primitives, and additionally to maximize the number of primitives.

In order to evaluate the performance of the proposed descriptors, and furthermore, to be able to compare different descriptors, we set up a series of experiments where the same reference set of input objects was used. The view filtering was applied on each object to get a fixed set of three-quarter views, as described in Chapter 3; the resulting reference set of views is depicted in Figure 4.1. Then, the views in this set were ranked by each descriptor. The objective was to see whether those views which ranked highest according to some descriptors were indeed those which are most informative to a human observer.
Here it should be mentioned that we have intentionally chosen reference models with right-to-left symmetry. Notice that in Figure 4.1 the views are pairwise symmetric, and it is quite reasonable that any descriptor will rank these pairs almost the same. So we always present only one view of each pair, thus allowing easy visual comparison between qualitatively different views. The resulting four views are depicted in decreasing order from left to right.

A final remark concerns the implementation of descriptors. Recall that
one of the requirements from a descriptor is that it should accurately handle visibility of features in a given view. Solving visibility in the object space is complicated and cumbersome. Therefore, to calculate visibility, we use a popular image-space technique called “Reference ID image” (e.g., see [3, 40]). In this procedure, each face of the model is assigned a unique color (which is used as the face ID). The model is rendered to the off-screen buffer on the GPU, and the resulting image is read back to the main memory. For each face $i$ we denote by $\text{Ref}(i)$ the number of pixels that show $i$’s color on the image. This value can be used, for example, to discard faces that do not show any pixels—such a face is either completely occluded or it is too small to be at the image precision. Conversely, if a face shows one or more pixels, it is partially or completely visible. Moreover, if object faces are sufficiently small, we can ignore partial face occlusions, assuming that faces with $\text{Ref}(i) > 0$ are always fully visible. So we define the visibility indicator $\delta(i)$ for the face $i$ as:

$$\delta(i) = \begin{cases} 
0, & \text{when } \text{Ref}(i) = 0; \\
1, & \text{otherwise}. 
\end{cases}$$

Theoretically, a similar technique might be used for primitives other than faces. However, for zero-area primitives (such as lines or points), this method is very imprecise. For this reason, in these cases we were forced to use object-space methods.

### 4.1 Surface Area

The first rather simplistic idea is to measure the geometric complexity of an object as the area of its surface. Then the viewpoint measure would involve the amount of the area that is actually exposed in the image. Two variations of this idea are to use either the true three-dimensional surface area (as described in this section), or the area that is projected on the image plane (described in Sections 4.2 and 4.3).

The rationale behind the Surface Area descriptor is as follows. If for one view we could say that “80% of the object is visible”, while in another view only “50% is visible”, then the first view would definitely be ranked better than the second one.

Here we associate this measure with the three-dimensional (unprojected) area of the object’s surface. The value of the descriptor is a ratio

$$D = \frac{A_{\text{vis}}}{A},$$
where $A_{\text{vis}}$ is the surface area of the visible portion of an object and $A$ is the total surface area. Notice that the normalization by the factor $1/A$, which maps all possible values of $D$ to the interval $[0, 1]$, is performed solely to make the value of $D$ more natural. Obviously, for comparison purposes, this constant factor may be omitted.

This descriptor has the important property that does not always hold for other descriptors—it takes into account the behavior of the invisible portion of the surface. Indeed, instead of saying “80% is visible”, one could always say “20% is invisible”. The objective of the descriptor is to maximize the visible surface portion and to minimize the invisible portion.

One way to estimate the amount of information that is exposed in a view is to attempt to reconstruct the 3D object from it and then compare how close this reconstruction is to the original object. Presumably, we might consider the invisible surface portion as the measurement for the uncertainty in the reconstruction, so that the reconstruction error would become smaller if the visible surface area is maximized.

In our implementation we approximate the discrete version of $D$. Specifically, $A_{\text{vis}}$ is approximated as:

$$A_{\text{vis}} = \sum_{i=1}^{n} \delta(i)A_i,$$

where $A_i$ is the area of three-dimensional face $i$ and $\delta(i)$ is the visibility indicator of face $i$ computed from the Reference ID image. The ranking of the reference views by this descriptor is shown in Figure 4.2.

### 4.2 Projected Area

Instead of measuring the visible surface area of three-dimensional shapes, it is possible to consider the area of its flat projection in a given view, or what is termed the projected area. In other words, since the viewer actually sees just a two-dimensional image of a shape, we measure the information gain in the image as the size of this image (specifically, its area).

This simple descriptor was proposed with slight modifications by Plemenos and Benayada [32] and Barral et al. [3] as an extension of the measure for non-degeneracy of faces by Kamada and Kawai [21].

The value of this descriptor can be calculated approximately in pixel precision, if we count non-background pixels in the Reference image. In fact, this value equals the sum $\sum_i \text{Ref}(i)$.

Unlike the Surface Area descriptor, here the invisible part of the shape is not considered explicitly. Since the descriptor value is a two-dimensional
area, it is normalized with respect to the total area of the image: $0 \leq D \leq (m \times n)$, where $(m \times n)$ is the image size in pixels. The ranking of the reference views by this descriptor is shown in Figure 4.3.

Figure 4.2: Surface Area.

Figure 4.3: Projected Area.
4.3 Projected Area Entropy

This descriptor extends the Projected Area measure by an additional assumption that the faces of an object constitute meaningful object features. Consequently, the final objective is, besides maximizing the total projected area, to maximize the number of visible faces also. This can be achieved by using the information entropy formula.

We consider the distribution of the projected area over all the object’s faces. For faces that are invisible in a view, the projected area is zero. If we denote by \( n \) the number of the object’s faces and by \( A'_i \) the projected area of face \( i \), then the distribution consists of \( n \) events with probabilities \( p_i = A'_i / A' \), where \( A' = \sum_i A'_i \) is the normalization factor (actually, this is the total projected area of the image). The value of the descriptor is then calculated applying the entropy formula 4.1.

This is essentially the viewpoint entropy measure that was proposed by Vázquez et al. in [40], but with the following modification. In the original work Vázquez et al. [40] suggested using the projected area of the background as the additional value \( p_0 \). The argument was that this factor allows distances to be handled—the entropy value becomes smaller when the camera moves away from the object. On the other hand, doing this introduces an additional bias in the final results. We recall that for our task the distance from the object remains fixed—this is actually the radius of the viewing sphere. Therefore, the contribution of the background is not taken into account.

The effect of this descriptor is to expose as many faces as possible so that the areas they contribute to the image are almost the same. For this to make sense an additional assumption should probably be that the tessellation of the object is uniform, that is, all the faces of the object are roughly the same size.

This descriptor has several weak points. First of all, since it explicitly uses object faces, its value is tessellation-dependent. If a mesh is triangulated, then different triangulations can give rise to different descriptor values, and even affect the relative order of these values. The second point is, again, that the invisible part of the shape is not handled explicitly. And finally, the normalization is done by the area projected on the image, which is not constant when the viewpoint changes.

In the experimental setting the projected areas are computed at pixel precision as \( A'_i = \text{Ref}(i) \). The ranking of the reference views by this descriptor is shown in Figure 4.4.
4.4 Semantic Entropy

It is possible to apply the projected area entropy method to geometric elements larger than the primitive elements (e.g., vertices or faces) of a 3D mesh. One way to achieve this is to use semantically important segments of the model. Since the segmentation of a model is required to have semantic meaning, this will also solve the problem of the previous descriptor, which is tessellation-dependent.

The descriptor value is calculated in a way similar to that described in Section 4.3. However, it is unlikely that different parts of an object have the same size. For example, a leg of an animal is typically much smaller than its body, so it does not make sense to require that both should be equally visible on the image. It seems more reasonable to assume that the proportions of the part sizes should be preserved.

In this work we measure the size of a part as its three-dimensional surface area. The relative size can be obtained by dividing this value by the total surface area of the object. Let $n$ be the number of segments in a model and $S_i$ be a surface area of segment $i$, then the relative size of segment $i$ is $s_i = S_i / \sum_j S_j$. The projected areas $A'_i$ of the segments are weighted by a factor that is inverse-proportional to the segment’s size: $w_i = 1/s_i$. This results in a set of “weighted” probabilities

$$p_i = \frac{w_i A'_i}{\sum_j w_j A'_j}. \tag{4.2}$$

Figure 4.4: Projected Area Entropy.
These probabilities are plugged into the entropy formula 4.1 to get the descriptor value.

Similarly to the previously described *Projected Area Entropy*, the effect of this descriptor is twofold. First, it seeks to maximize the number of visible semantically-important segments. Moreover, it attempts to achieve the uniform distribution of input probabilities. For an ideal solution the probabilities on the resulting image would be equal, so for two segments \( i \) and \( j \),

\[
p_i = p_j.
\]

Substituting 4.2 we get \( w_i A'_i = w_j A'_j \). Consequently, \( A'_i/A'_j = s_i/s_j \), meaning that the proportion of segment sizes is the same in 2D as in 3D.

This descriptor depends on the specific choice of the segmentation method, and its quality is critically affected by the quality of the segmentation. There exist many mesh segmentation algorithms and new ones continue to appear ([23, 12, 28, 22] just to name a few). In our experiments, we used the method proposed by Dey et al. [12], which seems to be able to identify parts of the model that are semantically meaningful.

To calculate the descriptor, we use the same technique as above, except that in the Reference image the faces belonging to the same segment are assigned the same color. The ranking of the reference views by this descriptor is shown in Figure 4.5.

![Figure 4.5: Semantic Entropy.](image)


4.5 Visible Curvature

The descriptors that were discussed in the previous sections use surface area as a measure of shape complexity. This characteristic is very flat—it does not give enough cues about the spatial behavior of object’s surface. The differential properties of the surface should be used to provide more precise descriptions.

For a long time in the literature it has been observed that high values of Gaussian curvature give a strong hint about the location of object’s features. For this reason Gaussian curvature is extensively used in computer graphics where the perceptual value of the results is important, for example, in mesh refinement, progressive compression and semantic segmentation algorithms.

For the descriptor in this section we measure the average of the absolute value of Gaussian curvature in the image. Thus the big descriptor values indicate a great number of peaks and/or saddles on the visible portion of the surface.

Following Page et al. [30], we use an indirect measure of the Gaussian curvature by the angle excess formula—for a vertex \( v \) the angle excess is:

\[
\Phi(v) = 2\pi - \sum_k \phi_k,
\]

where \( \phi_k \) are the apex angles in the one ring of triangles incident on \( v \). Then, the average value on a face \( f \) is estimated by the value at \( f \)’s barycenter:

\[
\Phi(f) = \frac{1}{|V_f|} \sum_{v \in V_f} \Phi(v),
\]

where \( V_f \) is the set of \( f \)’s vertices. Finally, the average value in the image is calculated as

\[
D = \frac{\sum_f A_f |\Phi(f)|}{\sum_f A_f},
\]

where \( A_f \) is the projected area of face \( f \), and it is approximated in pixel precision by \( A_f = \text{Ref}(f) \). The ranking of the reference views by this descriptor is shown in Figure 4.6.

4.6 Curvature Entropy

A more sophisticated curvature-based measure of shape complexity was suggested by Page et al. [30]. They proposed comparing the complexity of different objects by measuring the variation of the Gaussian curvature over the object surface. Thus a simple plane or sphere, for whom the curvature is constant everywhere, is considered to be the simplest type of shapes. Furthermore, this measure is able to cope with repetitive object features, such
as scales on a dragon’s skin, which do not contribute much to the total object complexity.

The *shape information* value is calculated as the information entropy of Gaussian curvature distribution on the surface. So it is assumed that the most complex shapes are those introducing as much variation as possible in the curvature whereas the contributions of every small range of curvatures are roughly equal.

We adopt this measure for our purposes by making it view-dependent. For this, only the visible part of the surface is considered and then the entropy is measured there. To estimate the probability distribution function we compute the histogram of pixel values on the Reference image. As in the previous section, the angle excess measure is used to estimate the average value $\Phi(f)$ for every pixel belonging to face $f$. These values are then placed in bins of equal width, and the entropy value of the resulting histogram is calculated. In our implementation, we use bin width $\Delta \Phi = 10^{-4}$. The ranking of the reference views by this descriptor is shown in Figure 4.7.
4.7 Silhouette Length

All previous descriptors were based on view-independent shape features. We now define descriptors that are inherently view-dependent.

Silhouettes (sometime called “occluding contours”) seem to provide an accurate and compact depiction of the shape of a 3D model, and, for this reason, are often used in non-photorealistic rendering (NPR). A silhouette is a view-dependent feature—an edge is a silhouette edge if the faces adjacent to that edge have an opposite orientation with respect to the viewing direction. More formally, the following relation should hold:

\[ \langle n_1, d \rangle \langle n_2, d \rangle < 0, \]

where \( n_1 \) and \( n_2 \) are corresponding face normals, \( d \) is the viewing direction and \( \langle \cdot, \cdot \rangle \) is the scalar product.

A simple version of this descriptor measures the projected lengths of all silhouette edges, similarly to the Projected Area descriptor. In this case, however, the Reference image could not be used, since it is quite impractical to calculate the length of a line that has been rasterized. To overcome this problem, we computed the silhouette edges in the object space. Then, in order to calculate the visible parts of the edges, Appel’s [1] Quantitative Invisibility algorithm was applied. Finally, the visible edges were cast to the projection plane and their lengths were summed together. The ranking of the reference views by this descriptor is shown in Figure 4.8.
4.8 Silhouette Entropy

A simple measure in the previous section may be extended similar to what was done for the curvature-based descriptors in Section 4.6. Here we use the notion of shape information for 2D planar contours as suggested by Page et al. [30].

The shape information measure for a smooth curve in 2D is the entropy of the curvature distribution along the arc length of the curve. The information is zero for a straight line and circle, since these curves exhibit no variation in shape. In the discrete case, we assume that a continuous curve is sampled uniformly by arc length. This results in a polyline, as illustrated in Figure 4.9.

Figure 4.9: Curve sampling and curvature estimation.

The curvature of the original curve in the sampled vertices can be estimated by the turning angle \( \phi \) formed by adjacent line segments of the
polyline. The required probability distribution is approximated using the histogram of these values, and then the entropy of the distribution is calculated.

In our implementation we collected statistics on all turning angles between adjacent silhouette edges. The angles were measured in their projected version, as they actually appear on the image plane. For the same reasons mentioned in the previous section, all calculations were carried out in object-space. Despite this, we observed that in some cases spurious silhouette edge crossings made the result quite unstable.

The ranking of the reference views by this descriptor is shown in Figure 4.10.

![Figure 4.10: Silhouette Entropy.](image)

### 4.9 Evaluation of Results

In this section we will evaluate and compare the performance of different descriptors for each reference model.

Since a good view computation is intended to imitate human intuition, the following evaluation strategy is proposed. First, the reference set of views for each model is ranked manually to obtain an optimal view sequence for this model. Then, each descriptor’s ranking are compared to this optimal sequence—the “closer” two rankings are, the better the performance of a given descriptor is.
To obtain such a comparison, a suitable way should be defined to measure a “distance” of any view sequence from a given optimal sequence. In fact, this problem can be formalized in terms of permutations. Given an optimal view sequence, each view is assigned its positional number 1, . . . , 4 in the sequence, so that number 1 stands for the best view and 4 for the worst one, respectively. While these numbers stay wired to the specific views, every view sequence can be represented as a permutation. In particular, the optimal sequence can be represented by a permutation

\[ o = (1 \ 2 \ 3 \ 4). \]

A “distance” measure should satisfy the following apparent properties:

- The distance will be small when the relative element order is preserved as much as possible;
- Elements with smaller numbers are considered more important, so that the best view (the first view in the optimal sequence) is the most important one.

For this purpose we found it useful to utilize the notion of inversion tables (e.g., see Knuth [24]). Recall that an inversion table \( \tau \) of a permutation \( \pi = (a_1 \ a_2 \ a_3 \ a_4) \) is

\[ \tau = (b_1 \ b_2 \ b_3 \ b_4), \]

where \( b_i \) is the number of elements to the left of \( i \) in \( \pi \) that are greater than \( i \). By this definition the following inequalities hold:

\[ 0 \leq b_1 \leq 3, \quad 0 \leq b_2 \leq 2, \quad 0 \leq b_3 \leq 1, \quad b_4 = 0. \]

Moreover, it is worth noting that an inversion table uniquely determines corresponding permutation, so it can be seen just as an alternative representation for that permutation.

For instance, the optimal view sequence \( o \) has the inversion table

\[ \tau_o = (0 \ 0 \ 0 \ 0), \]

while for the permutation

\[ \pi_1 = (1 \ 2 \ 4 \ 3) \]

the inversion table is

\[ \tau_1 = (0 \ 0 \ 1 \ 0). \]
Next we define a total order $\prec$ over the set of all possible permutations via corresponding inversion tables. The most natural and intuitive way to compare inversion tables is to use lexicographic order, from left to right. If we denote

$$\tau_0 = (0 \ 0 \ 0 \ 0), \quad \tau_1 = (0 \ 0 \ 1 \ 0), \quad \ldots, \quad \tau_{23} = (3 \ 2 \ 1 \ 0),$$

then

$$\pi_0 = (1 \ 2 \ 3 \ 4), \quad \pi_1 = (1 \ 2 \ 4 \ 3), \quad \ldots, \quad \pi_{23} = (4 \ 3 \ 2 \ 1),$$

and the order is

$$\pi_0 \prec \pi_1 \prec \cdots \prec \pi_{23}.$$

The element $\pi_0$ (or $\tau_0$) is the minimal element w.r.t. relation $\prec$, and it corresponds to the optimal permutation $o$. Now for a permutation $\pi$ we define the measure $d(\pi)$ as the integer distance between $\pi$ and $\pi_0$, or more formally as the number of permutations that are less than $\pi$ in the given order. Obviously, $d(\pi)$ can be calculated by the formula

$$d(\pi) = (b_1 \cdot 3 + b_2) \cdot 2 + b_3.$$

Our choice for view sequences, subjectively believed to be optimal, is depicted in Figure 4.11. Then, the descriptor comparison results for the given optimal view sequences are shown in Table 4.1. Several conclusions can be made by the examination of these results:

---

Figure 4.11: Optimal view sequences for the reference models.
Table 4.1: Descriptor comparison results. Lower distance-to-optimum values indicate higher descriptor quality.

<table>
<thead>
<tr>
<th></th>
<th>teapot</th>
<th>hippo</th>
<th>gargoyle</th>
<th>Average ↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>Projected Area Entropy</td>
<td>1</td>
<td>12</td>
<td>1</td>
<td>4.7</td>
</tr>
<tr>
<td>Visible Curvature</td>
<td>0</td>
<td>14</td>
<td>0</td>
<td>4.7</td>
</tr>
<tr>
<td>Semantic Entropy</td>
<td>6</td>
<td>12</td>
<td>0</td>
<td>6.0</td>
</tr>
<tr>
<td>Curvature Entropy</td>
<td>6</td>
<td>12</td>
<td>0</td>
<td>6.0</td>
</tr>
<tr>
<td>Silhouette Length</td>
<td>7</td>
<td>20</td>
<td>2</td>
<td>9.7</td>
</tr>
<tr>
<td>Surface Area</td>
<td>17</td>
<td>16</td>
<td>9</td>
<td>14.0</td>
</tr>
<tr>
<td>Silhouette Entropy</td>
<td>1</td>
<td>21</td>
<td>20</td>
<td>14.0</td>
</tr>
<tr>
<td>Projected Area</td>
<td>10</td>
<td>17</td>
<td>19</td>
<td>15.3</td>
</tr>
</tbody>
</table>

- It is evident that *Projected Area Entropy*, *Semantic Entropy*, *Visible Curvature*, and *Curvature Entropy* descriptors consistently show relatively good performance for the reference models.

- *Semantic Entropy* and *Curvature Entropy* descriptors show identical results in all the occurrences. We hypothesize that this is not a coincidence, but because these descriptors are roughly equivalent. Probably it can be explained by the fact that both descriptors are based on the analysis of the curvature distribution. Thus, the latter descriptor could always be used instead of the former, especially since it is very efficient.

- In general, the results for the *hippo* model are much worse than for the other models. This happens because the best view in the *optimal sequence* (see Figure 4.11) was chosen irrationally by a human from the pragmatic reasons, despite it having much less detail than the two consequent views. Since the descriptors are built to maximize the amount of detail, this causes an inconsistency with the empirical best view.

The summary of the experimental results for all the descriptors is presented in Table 4.2. Here only the “best” view is depicted for each descriptor.
<table>
<thead>
<tr>
<th>Surface Area</th>
<th>Projected Area</th>
<th>Projected Area Entropy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Semantic Entropy</td>
<td>Visible Curvature</td>
<td>Curvature Entropy</td>
</tr>
<tr>
<td>Silhouette Length</td>
<td>Silhouette Entropy</td>
<td></td>
</tr>
</tbody>
</table>

Table 4.2: Summary results for all the descriptors.
Chapter 5
Discussion and Conclusions

We have presented a framework for automatic calculation of good view directions for three-dimensional objects. The proposed framework involves two phases. The first phase is view filtering, which estimates a set of candidates for a good view. The second phase is the ranking of the candidates by a view descriptor. For view filtering, a scheme based on three-quarter view approximation has been proposed. This scheme was then experimentally tested with a variety of view descriptors.

The problem of finding a good view for an object seems to be quite difficult. It is becoming painfully obvious that there is no panacea. No one descriptor does a perfect job. It is probably possible to improve the descriptors described here and fine-tune them a little more, but we do not believe that any improvement achieved this way will be significant. However, since each descriptor does a reasonably good job on a majority of input objects, we are confident that it is possible to combine them to amplify the advantage that each has. Possible combinations are linear, where the optimal weights will have to be determined by some learning process, or non-linear, e.g., by a voting process.

In practice, three-quarter view estimation performs quite well on a large variety of synthetic and real 3D models. Its results have good correlation with those expected from a human, and therefore, this method is suitable for the view filtering phase. Nonetheless, because of its approximate nature, and in particular since the occlusions are completely ignored, the results are only approximations of the optimal ones. It was observed that in some cases the number of details in a view can be significantly improved by a slight viewpoint “adjustment” (within a small region of about 10 degrees) around the computed three-quarter view. Additionally, the three-quarter view method considers only spatial extent metrics of the object’s geometry. Competitive filtering schemes based on other shape properties, for example,
on surface normals, may exist.

The methodology presented here attempts to handle the problem at two
different levels of details—the view filtering operates at the most coarse scale
while the descriptor is computed from the fine-scale object properties. In a
paper by Lee et al. [27], a multiscale curvature measure—mesh saliency—was
proposed. Using a multiscale measure seems to be useful, and it might be
possible to enhance our descriptors to accommodate this type of information
as well.

In this work we did not solve the problem of automatic 2D image ori-
entation (see Section 1.3). To the best of our knowledge, this question has
not yet been addressed in the context of computer graphics, although similar
problems are currently being investigated in digital image processing. Wang
and Zhang [42] used statistical learning support vector machines (SVMs) to
classify digital photos into one of four correct orientations (which correspond
to those aligned with the frame of the picture). The classification is based
on low-level visual content, such as luminance and chrominance. For a syn-
thetic image, the classification is more complex—the main reasons being that
it lacks the environment information surrounding the object of interest and
that the set of possible orientations is continuous. Nevertheless, we think
that a similar idea of statistical learning might be applied in our case.
Bibliography


54


מה בתרומת
לקראת וישוב מבטום ליעם ביים חולת-ממדים.

אולג פלורנסקי
מה ב (_.) בהיותם עולים ל-mutedים לת-מודרים.

תנור על מתקר

לשם مليフィ חלקי של הדירוג של כלבת התאר

cוגניטי למדוע במדעי המחשב

אולג מְלָלֹנְסֶק

והם לסתנ תכנית — מקו טכנולוגיה לישראל

dצמר 2005 שן תמסי

HyFiهFi

Technion - Computer Science Department - M.Sc. Thesis MSC-2006-03 - 2006
המתקב גרעינ מסדרה פר visita ובעטר שמקולי דלקתי לתמוך במועש.

אנו מתודה הפרור והמצפה לозвращ לארץ המקור.

ברצוני להעזרת עפעפיICI- MPII ביצירת מענה-CNRI-MATI על הדמות.

המשורפת תנשא.

יתדה מזוהה שלושי 이것은, שלדני לא擴 והורי הלוחמי אתיה השלוחין על מספר.

ה垿יכו המשורר העידון.

אנו מודים לנטון על הטענה השפיט הלדרה בברית-achment.
תרוכת עניינים

התקציר大全ית
1  כ multipart,.
3  הסברת הקדמה 1.1
6  מבחר הקבוצות 1.2
9  מבחר האלפמטיים 1.3

نظירות הבסיס
13  מטסינים תובים וממתקים ק.statusTextים 2.1
16  מטסינים של מטסינים ק.statusTextים 2.2
18  מטסיקים פוקסטของเขาים 2.3
21  הפרוזה המגיעה 2.4

סימוכים נוספים
25  מטסיקים של שילוט-רבני 3.1
27  המורכבות הפוקסט 3.2
29  סימול ביביינטים כלליים 3.3

פריקט התוכנה
33  שיטת תלת-מודרני-גראה 4.1
34  שיטה בטל 4.2
35  אטרופיה של שיטת בטל 4.3
36  אטרופיה של אטרופיה סימוני 4.4
37  אטרופיה של אטרופיה טבלת 4.5
38  אטרופיה של אטרופיה הטבלת 4.6
39  אטרופיה של אטרופיה סימוני 4.7
40  גירוד קיים סילאנס 4.8
41  גירוד קיים סילאנס 4.9
51  סיכום ומסקנות
רשימת מקורות
tקציר

ב
רשימת אורכים

1. 
1.1 סוויפת שולחן על מכתשים טופיק
1.2 מכתשים טופיק מכתשים ריק
1.3 השפעת של יציאת תאורה טופיק
1.4 פרטיזיאלי של מכתש

2. 
2.1 מטרחת צירב קגדית
2.2 של מכתשים על תכנית טנקה
2.3 הלינק ה- " varargin " בירור של טיקטוט
2.4 תכניות של " מstadt החיסות "
2.5 תרשים של התרעמות המ抉וע

3. 
3.1 צויר פרופיל
3.2 דוגמאות של מכתש שילוב-רבעי
3.3 " מכתשים " של Melania B-
3.4 מכתש שלוש-רביעי מבוקסיס אפרטימיט

4. 
4.1 קבוצת מכתש ייחוס
4.2 שチン לתל-מזרחי-
4.3 שチン-
4.4 אטרופיה של שチン-
4.5 אטרופיה ספגטי-
4.6 עקומות התוצר
4.7 אטרופיה של עקומות-
4.8 אטרופיה של כף-
4.9 דגיום עקומות כף-
4.10 אטרופיה של כף-
4.11 סדרות מבטיס אפרטימיט

5. 
5.1 המטástזים

6. 
6.1 פעולות מחקר ופיתוח
6.2 ייעוץ ו尻וף
6.3 תובלה ושיקום
6.4 תכנון ו_NATIVE
6.5 המпередים
6.6 תכנון ו_NATIVE
6.7 מתקנים
6.8 תכנון ו_NATIVE
6.9 מתקנים
6.10 תכנון ו_NATIVE
6.11 מתקנים

7. 
7.1 תקיפות והבנת
7.2 תקיפות והבנת
7.3 תקיפות והבנת
7.4 תקיפות והבנת
7.5 תקיפות והבנת
7.6 תקיפות והבנת
7.7 תקיפות והבנת
7.8 תקיפות והבנת
7.9 תקיפות והבנת
7.10 תקיפות והבנת
7.11 תקיפות והבנת

8. 
8.1 תקיפות והבנת
8.2 תקיפות והבנת
8.3 תקיפות והבנת
8.4 תקיפות והבנת
8.5 תקיפות והבנת
8.6 תקיפות והבנת
8.7 תקיפות והבנת
8.8 תקיפות והבנת
8.9 תקיפות והבנת
8.10 תקיפות והבנת
8.11 תקיפות והבנת

9. 
9.1 תקיפות והבנת
9.2 תקיפות והבנת
9.3 תקיפות והבנת
9.4 תקיפות והבנת
9.5 תקיפות והבנת
9.6 תקיפות והבנת
9.7 תקיפות והבנה
9.8 תקיפות והבנת
9.9 תקיפות והבנת
9.10 תקיפות והבנת
9.11 תקיפות והבנת

10. 
10.1 תקיפות והבנת
10.2 תקיפות והבנת
10.3 תקיפות והבנת
10.4 תקיפות והבנת
10.5 תקיפות והבנת
10.6 תקיפות והבנת
10.7 תקיפות והבנת
10.8 תקיפות והבנת
10.9 תקיפות והבנת
10.10 תקיפות והבנת
10.11 תקיפות והבנה
טכניון - קמפוס ז"ח - מכון מחקרי טכנולוגיה הצגנית - מ Duchess - מ.ס. 2006.03 - 2006

תכנית

העלאס האופתי_- מרוכב עם ממדות המצלמה. לעומת זה, מורים רחבי
הארטיסטבה המכללת להמחק קי הב omdatיה- הממדות של העלאס על רישויות עד.
ה millennית של אדום- ובית הדואר. לצורך להנפקת תצלומי
על העלאס ושל ממדות על מסך המדגים, התפרקה התרא, שמעורר המדיניות
טכנייה-צגנית של העלאס הצגנית. כקרוב במגדים חקל הממסד של האינפראפקט
המדוקדוק מצגת האינפראפקט המ gratuita сфורה בישה הבורש המבעד
הonica — מיקוםเซמה בין לספרים הЛенин

被执行 ממישור נגה ירושלמי שהנושא מסолучס כבלדה בבלדה创新创业
'טורבוס' מבלטס 'רוריסט' של רוזי ראוניק, נזר על ק. מכירות האופנה
ההרבעיטין לשתייתו. ברדר כל התראתה ונאבקי הצלחת הסדר
שלח. לפגיע מהאולאמות הצלחתו גדל, ברית uyבייקת תצלום-ממודים, אפשי לאלפא
אף והתכניות הצלחתו של בורס. פורגר ליבליי או אומר לחד השמיש
היפרמיסיסי רביי. כנ纶 הצילום גדול הראינו מתכניות במתכנית
של מודלים במדגמיظهور 3D שלוה. קומפוזיציות ארוטיות והצלחת של צקת-תצלום

המדוקדוק לעור

בודה היא זנות מעונכים לפרסיםشوים bastyla-בסטלת בטיסカフェיהתורמים, "דו אייל וرأ
היה מתבצת צלסה "תמונה צבע בורה" של האופקיט הצגנית. תלמידיו שלארטיסטמה
היאולאמות האופקיט של ממדת צבע בוהל איפאמה-סטויק. יש לי 컷ס
המקס שפעים זכרים ספארקיה, ב-1951 חלקה בשתייתו. שכי רובה
מר מחשבים מוכריםميון ייחוד הם (מכים דדננים Cities, סולטאי מוכרים
אמ קריפטוס מטרים מטרים שומי טיקם ביאר אוקבי "ידור, בונדס רד גל
אמ קריפטוס מטרים מטרים שומי טיקם ביאר אוקבי "ידור, בונדס רד גל
של כל מודלים חברתיים ספוגים ביאר אוקבי "ידור, בונדס רד גל
המדוקדוק לעור — נקולראם מינימום — בטיחות זו לטח(Element)

המדוקדוק לעור

לבריק השובש אתהתכניות לשמצה קוגניט קודהכל דרוס לה빔 מק
האפרותרה הייחודיים של המפעילות האולאמות. בודר כל שתייתו בשתי ערמא
ככלים שמתממשים על ממדת האולאמות בשתייתו: אופרטוניותה ורכיב אונסני
באופית את אופריים многих האינפראפקט ז WoWquir
לשם אולאמות הביאר ברומח. בול אינפראפקט
שאנדרט לפני מעוף ליושפ fåתת פורס מיםBLE הריצוף, מתקיים
האולאמות ואת ברורם בחומドイツ, "ידור היפסטר". בחומ্ום
המדוקדוק של השיקולים
המדוקדוק הכח ויוזם איי "יפורים היפסטר". המטריה השמשת טבל

h
The method of releasing the brakes is to turn off the power supply after the emergency stop is activated.

The brakes are released by turning off the power supply.

The brakes are turned off by activating the emergency stop.

The emergency stop is activated by turning off the power supply.
הוחלף צירוף התחメイン של בלוק שגוי
(Principal Component Analysis, PCA)
(1 ± 1,1)
לפרך השואות ובין מודדים שווים נבחה מודדים תהל-ומדידים שמידות
נלקחת צורת כבדיוק. אך שב�отת המבטים המסוימים על פאזה האורחמה (מכני
שלופים-רבועים) חזרה אחת במדיד ציוני. לכל מודד את מרכיב התיאור דירוג
המכונים כפי שוחטכתל調べCAF
לסכים, ארז מגדלים come השקן שבעית החושב והשומעים של מוכנים טובים
היא בעיה יד קשת. בכל הגזרה ישון פırken מקים שיוול ספק את את האיפון. }
cפי שרטוטי. אלא אם המבודדים שבוביש אל היה משולח. "תمساب שפרס סוף
ולחומ את המבודדים ח新三板 כי אם. אך נראיה שחי אל ישרר את התונים ברחר
שמשקיתות الشريفת זאה. ל tecnologíaشعبôn כל אחד מבודדים והטכני
על בוב סקולס. ארזVES ששרותיהם שלחה מידע להбережה אחר התונים
המכהל.iráויה יכו חיתה ליאיר. כל הערים המשקפים ואתפיטים כיון לקבוצה
"א. ממצאת במדות. או לא-ליאור, למשל על ידי הנקודות הבאות.