Chapter 1

Introduction

Surfaces are important in research disciplines like Computer Aided Design, Computational Geometry and Computer Graphics.

Computer Aided Design [8, 52] is the development of a wide range of computer-based tools assisting in the design and creation of products and goods. These tools help, amongst others, engineers and architects in their design activities. In Computer Aided Design, the focus lies on construction and design of curves and surfaces.

Computational Geometry [60, 89] is the study of (computer) algorithms to solve problems stated in terms of geometry. Computational geometry has recently expanded its scope to include curves and surfaces. The research presented in this thesis is funded by EU projects dedicated to curves and surfaces and focuses on retrieval of properties of the curves and surfaces.

Computer Graphics [63] is the research field dedicated to visualization, where one utilizes computers both to generate visual images and to alter visual and spatial information sampled from the real world. Emphasis lies on visualization and output of surfaces.

Irrespective of the purpose for which the surface is used, without a suitable description a computer cannot be of any use. A surface representation is the formulation of a surface such that a human is able to reason about the surface with the aid of a computer. Many representations have been proposed [18, 49, 52] each with different advantages and disadvantages. Conversions between representations are needed in order to be able to use the advantages of different representations in the same application. E.g., piecewise linear approximations are regularly used for visualization and further geometric processing. Often it is not possible to give a representation that defines the same surface as the original surface. In such cases the surface is approximated.

Starting point of this thesis is the class of skin surfaces used in molecular biology [48]. Several methods exist that deform one skin surface efficiently into another skin surface [28, 33, 48] making them potentially suitable for animations. We show how an arbitrary surface can be approximated with a skin surface in Chapter 3 and construct a piecewise linear approximation of a skin surface in Chapter 5. Finally, in Chapter 4 we extend the class of skin surfaces to the much wider class of envelope surfaces, in particular envelopes of balls.
2. Chapter I. Introduction

Figure 1.1: The skin surface blends spheres together

The remainder of the introduction is organized as follows. We start with an outline of this thesis. We then review areas in which surfaces emerge and present surface representations. In Section 1.4 we present surface models used for molecular surface design. We conclude the introduction with a brief description of several surface approximation algorithms.

1.1 Outline of the thesis

In this thesis we propose a class of surfaces represented by a finite set of balls contained inside the surface. The surface wraps tightly around the union of the balls, making the transition between balls smooth. The size of the smooth patches in between balls can be controlled as shown in Figure 1.1.

Surfaces defined by a set of balls are used in molecular modeling to build computer representations of large proteins. In [48] Edelsbrunner defines such a class of surfaces called skin surfaces. A skin surface is formed by a set of spheres that are ‘blended’ by smooth patches. These surfaces are used to simulate chemical processes, like protein folding [25], in which molecules deform. Several algorithms have been proposed to deform skin surfaces [28, 33, 48].

Generating balls Laser scanners provide an automated way to construct surfaces of physical objects. These scanners probe an object with a laser and produce a dense set of points on or near its surface. Many surface reconstruction algorithms have been proposed that compute a surface by reconstructing the object from these points, see Section 1.5. Algorithms [6, 7, 45] exist that transform a dense point sample on a surface into a set of balls almost contained inside the surface. We describe such an algorithm in Section 2.6. The generated balls can be used to define a surface in our representation.

An example of the generation of these balls is shown in Figure 1.2. The three pictures show the real-world object (the Stanford bunny), the point sample generated by a laser scanner and the constructed set of balls contained inside the bunny.

Surface approximation Even if a surface representation has many properties, like ease of visualization and that is suitable for further geometric processing, it is
1.1. Outline of the thesis

(a) The Stanford bunny *(b) A points on the surface of the bunny (c) A set of balls contained in the bunny*

Figure 1.2: The input for our algorithm

of little use if there is no (automatic) way to construct or approximate a surface in that representation. If the representation is not exact, then original surface and its approximation should at least be ‘similar’. For example, one could say that two surfaces are similar if they are topologically equivalent, which, among other things, means that they have the same number of connected components, holes and tunnels. Another metric on the class of compact surfaces is the Hausdorff distance, which measures of the distance between the two surfaces. One could also require that not only the distance between the surfaces is small, but also that the normals to the surfaces agree. We discuss these notions in more detail in Section 2.2.

In this thesis, we use an additional criterion for good approximation. Our surfaces are represented by a set of balls. We assume that the boundary of their union forms a good approximation of the surface. Therefore, we also require that the approximating surface touches all input balls. This is a guarantee that the surface follows the boundary of the union of the balls closely.

1.1.1 Approximation by skin surfaces

We propose a method for the construction of a skin surface from an arbitrary surface that guarantees good approximation qualities.

A set of inscribed balls of a surface is dense if the boundary of the union of these balls and the surface have the same topology and a small Hausdorff distance. To become more precise, the boundary of the union of a dense set of balls is not differentiable on an intersection curve where two balls touch. To remove the tangent discontinuities, we use skin surfaces. A skin surface defines a smooth surface controlled by a single parameter called the shrink factor $s$. For a shrink factor equal to one, the skin surface is the boundary of the union of the balls. If the shrink factor is smaller than one, the radii of the balls are shrunk with a factor $\sqrt{s}$ and the balls are connected by smooth patches of hyperboloids and spheres. In order to make the
Figure 1.3: The approximating skin surface of a set of balls forming a hand. The patches between the balls are controlled by a single parameter.

skin surface wrap around the union of the balls (and not the shrunk balls), we first grow the input balls by a factor of $1/\sqrt{5}$. We call the skin surface with the grown balls the extended skin surface. An example of an extended skin surface is shown in Figure 1.3. The left figure shows the input balls and the shrink factor is decreasing from left to right.

If the shrink factor is close to one, the extended skin surface and the boundary of the union of the balls have small Hausdorff distance and the same topology. In fact, they are equal for a shrink factor one. As the shrink factor decreases, the patches become larger and they may cover (small) intrusions in the surface, viz. the region between the fingers and the palm in the third picture of Figure 1.3. Even different parts of the surface may grow towards each other and merge. Therefore it is important to have a lower bound on the shrink factor for which the extended skin surface and the union of the balls have the same topology and a Hausdorff distance smaller than some fixed constant. We propose an algorithm that derives the lower bound in Chapter 3.

By continuously decreasing the shrink factor, we are able to prove the following properties of the extended skin surface:

- the extended skin surface is topologically equivalent with the boundary of the union of the input balls;

- the Hausdorff distance between the union of the input balls and the body of the extended skin surface is smaller than some fixed constant $\epsilon > 0$;

- each ball in the input set is tangent to the extended skin surface.

In fact, for a dense set of balls, these results also apply to the surface from which the input balls are obtained if each ball contributes to boundary of their union.

1.1.2 Envelope surfaces

The extended skin surfaces have nice properties as mentioned in the previous section. The main disadvantage of the approach is that a skin surface is only controlled by
a single parameter, namely the shrink factor. For example in Figure 1.3, combining the fingers of the second picture and the palm and wrist of the third picture gives an approximation with the same properties as mentioned above.

In Chapter 4 we introduce a class of surfaces, called envelope surfaces, that forms a generalization of the class of skin surfaces. With envelope surfaces we first construct a triangulation of the centers of the balls and can change the envelope surface at the level of simplices of the triangulation yielding an approximation that is locally adaptive. Further, the interpolating patches generated by skin surfaces are always concave. For envelope surfaces these patches can be both concave and convex and, hence, allow for a better approximation. A two-dimensional example is given in Figure 1.4. The set of input circles is shown in Figure 1.4(a). Figure 1.4(b) shows an extended skin curve with a shrink factor close to one. The extended skin surface does not differ much from the boundary of the union of the input circles. For a smaller shrink factor patches between different fingers arise and the extended skin curve changes topology, as is shown in 1.4(c). Using an envelope surface to construct the surface, it is possible to interpolate in the direction of the fingers, but not in between the fingers, cf. Figure 1.4(d).

An envelope surface is defined as the boundary of the union of an infinite set of balls. We define these balls by constructing a continuous function that assigns a squared radius (weight) to each center. Under certain conditions on this weight function the envelope surface is tangent continuous. As an important special case, we analyze envelope surfaces defined by a piecewise quadratic weight function, which includes the class of skin surfaces. For this type of weight function, we develop efficient tests to validate the condition under which the envelope surface is tangent continuous.

Many desirable properties of skin surfaces hold in the generalized setting of envelope surfaces, which makes us believe that applications, like deformation schemes and meshing algorithms, developed for skin surfaces can be extended to envelope surfaces. For example, both the skin surface and the envelope surface are piecewise quadratic. However, a skin surface consists of parts of spheres and hyperboloids whereas the envelope surface consists of pieces of quadrics any type. Further, for both classes of surfaces there is a polyhedral partitioning of space decomposing the surface into these quadrics.

### 1.1.3 Meshing skin surfaces

Meshing of smooth surfaces is a first step in many numerical simulations and is often needed for visualization. In Chapter 5 we present an algorithm for meshing skin surfaces with guaranteed topology. The algorithm decomposes the skin surface into parts homeomorphic to a disk. A coarse mesh is then constructed by approximating each part with a piecewise linear patch.

The coarse mesh is not directly suitable for further processing since it has too little detail and many skinny triangles. Therefore we use several existing refinement algorithms to enhance the mesh. The first method we propose is a subdivision algorithm, which refines the mesh according to a subdivision scheme [69]. This
algorithm is very fast and particularly useful for visualization of skin surfaces. A second algorithm is based on an algorithm by Chew [34] and increases the minimal angle of a triangle. This is necessary for the stability of numerical simulations, [23].

1.2 Generating surfaces

We discuss three areas in which surfaces emerge.

In the setting of surface reconstruction the surface is not completely known. A surface is constructed that forms a reconstruction of the original surface. For example, a laser scanner generates a finite sample of points lying on the surface. Based on that point sample, surface reconstruction algorithms construct a surface that is a reconstruction of the scanned object.

In Section 1.3, we see that the implementation of operations on a surface depend on its representation. Therefore it is sometimes necessary to change the representation of a surface. In the setting of surface approximation, a surface defined in one representation is approximated with a surface in another representation. Meshing algorithms for smooth surfaces form an important class of surface approximation method. A meshing algorithm generates a piecewise linear approximation (polyhedron) of a smooth surface. In general, the faces of the approximation are triangles. These meshes are used for visualizing the surface and for further geometric processing. The algorithms proposed in Chapter 3 and 5 belong to the class of surface approximation algorithms.

The last area from which surfaces emerge that we mention is surface design. Surface design is the process in which persons like engineers and architects construct a surface interactively using a computer. Important in surface design is that the surface can be easily and intuitively manipulated. The class of envelope surfaces,
1.3 Surface representations

We mention three basic classes of surface representations.

The first class is formed by *parametric surfaces* [52]. These surfaces are defined by a function that maps a two-dimensional domain to a surface in $\mathbb{R}^3$. An example of a parametric function $f : [0, \pi] \times [0, 2\pi] \rightarrow \mathbb{R}^3$ that defines the unit sphere uses spherical coordinates: $f(u, v) = (\sin(u) \sin(v), \sin(u) \cos(v), \cos(u))$.

It is easy to generate points on a surface defined by a parameterization, which is a starting point of many algorithms. These points are necessary for meshing surfaces. On the other hand it is hard to test whether a point lies on a parametric surface. Therefore it is also difficult to prove in general that the surface does not have self-intersections, i.e., that there are no two different parameter values that map to the same surface point.

The class of *implicit surfaces* [18] defines a surface as the zero set of a scalar valued function on $\mathbb{R}^3$. In implicit representation, the unit sphere is defined by the function $f : \mathbb{R}^3 \rightarrow \mathbb{R}$ given by $f(x) = \|x\|^2 - 1$. Testing whether a point lies on an implicit surface is easy. On the other hand, generating points on the surface is harder than for parametric surfaces. In fact, it is even hard to find the number of connected components and points on each connected component. Both the skin surface and the envelope surface are implicit surfaces.

*Piecewise linear surfaces* [49] play a crucial role in visualization and in numerical simulations. The representation of parametric surfaces and implicit surfaces are often too complicated to use directly in applications, therefore the surfaces are first meshed and the computations are then performed on the piecewise linear approximation. For further geometric processing it is important that the surface and the linear approximation are topologically equivalent. In Chapter 5, we construct a mesh from a skin surface with this property.

A surface representation that also defines a surface by a set of interior balls is the *Medial Axis Transform* (MAT) [56, 95, 99], see Section 2.6. The surface is defined as the boundary of the union of an infinite set of maximal balls. The centers of the balls form a skeletal structure of the surface called the medial axis. The balls are defined by assigning a radius to each point of the medial axis.

The thesis of Vermeer [95] focusses on the construction of a parametric representation of a surface from its Medial Axis Transform. They also analyze under which conditions on the medial axis and the radii of the balls the surface is tangent continuous. The methods we propose differ from the methods in [95] because we construct balls centered on a three-dimensional domain and not restricted to the medial axis. They also start with a complete representation of the MAT and then construct a parametric representation of the surface, whereas we assume that the we are given an incomplete set of balls, from which we reconstruct the surface. An advantage of our method over theirs is that the analysis under which conditions the boundary of the union of the balls is tangent continuous becomes easier. Our condition only

introduced in Chapter 4 falls in this category.
Figure 1.5: Different kinds of molecular surfaces. Figure (a) and (b) show the Van der Waals surface (VdW), the solvent accessible surface (SAS) and the solvent excluded surface (SES) for different sizes of the solvent probe. Figure (c) shows the skin surface.

depends on the radius function while their conditions also rely on the local structure of the medial axis.

1.4 Molecular modeling

The surfaces discussed in this thesis are related to skin surfaces. We present the theory of skin surfaces in more detail in Section 2.7. Several other types of surfaces have been proposed to model molecules. Some of these models give insight in the structure of the molecule or in characteristics like the double helix structure of DNA. To sketch the context in which skin surfaces were developed for molecular modeling, we review some of the models that are related to skin surfaces. A two-dimensional example of the different surfaces is shown in Figure 1.5.

Van der Waals surface Each atom in a molecule has an electrical charge that produces a Van der Waals force. This force is strongly repulsive in close proximity, becomes mildly attractive at intermediate range, and vanishes at long distance. Due to the repulsion on the short range, atoms cannot be too close to each other. The strength of the force depends on the type of atom. For each type of atom, the radius of the ball (the Van der Waals radius) where the center of no other atom can penetrate is experimentally determined. In this way atoms are associated with balls. The Van der Waals surface is the boundary of the union of these balls.

Solvent accessible surface Small gaps and pockets in the Van der Waals surface are not relevant for chemical processes [40, 77]. For example, if these gaps and pockets are so small that not even a solvent molecule (like water) can enter, they do not influence chemical processes. The solvent accessible surface is proposed as a tool to remove these small artifacts from the Van der Waals-surface.

To construct this surface, a spherical probe is rolled over the Van der Waals surface. The trace of the center of the probe is the solvent accessible surface. It bounds the region that is accessible by the (center of the) solvent. This surface is
1.5 Surface reconstruction

Since the introduction of laser scanners, dense point sets of scanned objects are generally available. Surface reconstruction algorithms reconstruct the surface bounding the object from these point samples.

Many algorithms were developed that construct an interpolating or approximating surface from the set of points. An interpolating surface is a surface that passes through all sample points. Approximating surfaces do not interpolate the sample points, but lie in their vicinity. If the point sample contains noise, the points do not lie on the surface and algorithms that construct approximating surfaces can be used to smooth away the noise at the expense of a loss in detail.

We describe some algorithms for the purpose of surface reconstruction from point samples. This list is far from complete, but is intended to give an overview of surface reconstruction algorithms related to this thesis. Most of these algorithms have theoretical guarantees with respect to the input surface like topological equality and a bound on the Hausdorff distance under the assumption that the point sample is an $\epsilon$-sample, with $\epsilon < 0.1$. An $\epsilon$-sample means that the point sample contains points on all parts of the surface and more points in regions with more detail.

Figure 1.6: Solvent excluded surfaces for probe spheres with different radii. In light gray, the atoms, darker gray the interpolating patches between two balls and in dark gray the patches between three balls. The patches may not be tangent continuous as is the case in the second figure, where the tangent plane is discontinuous at six points. In the third figure the tangent plane of the SES is discontinuous at the intersection circle of the two probes that touch all three atoms.

equal to the boundary of the union of the balls obtained by adding the radius of the probe to the Van der Waals radius of each atom.

Solvent excluded surface Opposed to the solvent accessible surface, that bounds the region where the center of the probe enters, the solvent excluded surface [87] bounds the region where the solvent cannot enter. To construct this surface, the spherical probe is used to carve away the space outside the Van der Waals balls. This surface consists of spherical parts of the Van der Waals surface, toroidal patches between two Van der Waals balls and spherical patches where the probe touches three Van der Waals balls. This surface is also referred to as the Connelly surface [38, 39]. Note that this surface may also be not differentiable as shown in Figure 1.5(b) and Figure 1.6. Visualization algorithms for this type of surfaces are presented in [14, 15].
A mathematical definition of the notion of an $\varepsilon$-sample is given in Section 2.6. Although it is hard to satisfy or test this condition, the algorithms work well in practice even if the sampling condition is not met.

**Notation** The set of sample points is denoted by $P = \{p_0, \ldots, p_n\}$. Some of the methods mentioned below produce an implicit function, the zero set of which is the reconstructed surface. These method of surface reconstruction are also called scattered data interpolation. We denote this function by $F$ and the specified function value of $F$ at the point $p_i$ by $f_i$.

**Delaunay based reconstruction** Several algorithms reconstruct a piecewise linear surface using the Delaunay triangulation of the point set $[5, 6]$. These algorithms extend earlier work on curve reconstruction $[4]$.

The Delaunay triangulation is one of the most basic structures in Computational Geometry, and is described in detail in Section 2.5. The Delaunay triangulation is the dual of the Voronoi diagram, which decomposes $\mathbb{R}^3$ into regions closest to a certain site. The farthest point in the Voronoi cell from a site is called the pole of the site. If the point sample is dense, the normal to the surface is well approximated by the vector from a site to its pole. A polar ball is the largest ball centered at a pole, not containing any site. Each polar ball has four sample points on its boundary.

The power crust algorithm $[6]$ labels polar balls inside and outside. Two polar balls are neighbors if their Voronoi vertices are connected by a Voronoi edge. The algorithm starts by giving a large sphere outside the surface the label outside. Then it progresses to neighboring polar balls and gives these balls the same label if the outer angle in which the spheres intersect is large and the opposite label if the angle is small. The reconstructed surface is defined as the set of points with equal distance...
to interior and exterior balls. Using a suitable distance function, this surface is a subset of the facets of the Delaunay triangulation, which is piecewise linear.

The different steps of the algorithm are depicted in Figure 1.7. The reconstructed curve is the subset of the edges of the Delaunay triangulation that are emphasized in Figure 1.7(b). We describe this algorithm in greater detail in Section 2.6.

The cocone algorithm [5] uses the poles to estimate the surface normals in the sample points. It then constructs cocones around the samples. These cocones are the complement of the cone with apex at the sample and symmetry axis in the direction of the estimated normal. The Delaunay facets (triangles) lying in the cocone are selected as candidate samples. A manifold extraction step selects triangles that form the final reconstructed surface.

**Moving least squares** The idea of the moving least squares approach [1, 79, 78] comes from the well known least squares technique to fit a surface to a set of points. The term “moving” refers to the various weighting of the points in calculating their contributions to the solution at different locations. In general, sites lying close have a larger weight. Several slightly different definitions are given in literature, for assigning the weights and constructing the moving least squares function.

The approximating MLS-surface is homeomorphic to the original surface if the uniform point sample is dense enough, as is shown in [70]. In [42], the condition on the uniform sample is made adaptive, hence requiring less points. In [90], Shen et al. describe an algorithm that takes polyhedra as input instead of points.

**Subdivision surfaces** An important way of modeling and reconstructing surfaces is by means of subdivision surfaces. These surfaces are defined by a coarse control mesh which is then subdivided. Several schemes are proposed for subdivision. The schemes in [47, 100, 69] interpolate the vertices of the control mesh, whereas the schemes [81, 26] are approximating. In practice, the latter schemes produce better results.

In [65, 67, 66], Hoppe et al. describe a surface reconstruction algorithm that constructs this coarse mesh from an unorganized point cloud. The idea is to estimate the tangent plane of the surface in each sample points. This is done by a least squares fit through the samples in the neighborhood of the sample point. Then the normals are oriented consistently and the surface is extracted with a marching cubes [82] like algorithm.

This surface reconstruction algorithm is one of the first and very little is proved about their behavior.

**Natural neighbors** A natural neighbor surface [20, 53] constructs an implicit surface using natural neighbor coordinates, also called Sibson coordinates [91, 97]. The implicit surface is the zero-set of the implicit function.

The definition of natural neighbor coordinates is given in terms of Voronoi cells, which are more generally discussed in Section 2.5. In short, the Voronoi cell $v_p$ of a
point $p \in P$ with respect to a point sample $P$ is the set of points closer to $p$ than to any other sample in $P$.

The natural neighbor coordinate associated to $P$ of a point $x \in \text{conv}(P)$ are the functions

$$\sigma_i(x) = \frac{|v_i(x)|}{|v_x|},$$

where $|\cdot|$ denotes the volume of a set, $v_x$ the Voronoi cell of the point $x$ with respect to $P \cup \{x\}$ and $v_i(x)$ the intersection of $v_x$ with the Voronoi cell of $p_i \in P$ with respect to $P$. Since the Voronoi diagram partitions $\mathbb{R}^3$, it follows that $\sum \sigma_i(x) = 1$, for all $x \in \mathbb{R}^3$. Further, $\sigma_i(p_j) = \delta_{ij}$, with $\delta_{ij}$ the Kronecker delta. The points $p_i$ with $\sigma_i(x) > 0$ are called natural neighbors of $x$.

The interpolated function value $F(x)$ is defined as

$$F(x) = \sum_i \sigma_i(x)f_i,$$

or, if the normal $n_i$ of site $p_i$ is known, as

$$F(x) = \sum_i \sigma_i(x)(f_i + \frac{1}{2}\langle n_i, x - p_i \rangle).$$

The approximation is the zero-set of $F$. The function $F$ is differentiable, except at sample points.

Initially, the natural neighbor coordinates were proposed for function interpolation. For surface reconstruction, only natural neighbors lying near an estimated tangent plane in the query point $x$ are used. These points are called the T-neighbors.

**Radial basis functions** The reconstruction algorithms that use radial basis functions [86, 93, 94] construct an implicit function $F$, the zero set of which is the reconstructed surface. The implicit function is a weighted sum of radially symmetric functions $\phi : \mathbb{R}^+ \to \mathbb{R}$, depending only on the distance to a site:

$$F(x) = \sum_{i=0}^{n} w_i \phi(||p_i - x||) + P(x), \quad (1.1)$$

where $w_i$ is the weight of the radial basis function $\phi$ centered at $p_i$ and $P$ is (an optional) low degree polynomial. Common choices are $\phi(r) = r$ (linear), $\phi(r) = r^2 \log(r)$ (thin-plate spline or biharmonic radial basis function in 2D) and $\phi(r) = r^3$ (thin-plate spline in 3D).

In Equation (1.1), the unknowns are the weights $w_i$ and the coefficients $c_i$ of the polynomial $P$. Since the equation is linear in both parameters, it can be expressed as a linear system. The weights $w_i$ are used to solve the equations $F(p_i) = f_i$ and the coefficients of $P$ are used to guarantee that the system of equations has a solution.
These equations are formulated in the following linear system:

\[
\begin{pmatrix}
\phi_{01} & \phi_{02} & \ldots & \phi_{0n} & p_{0,0} & \ldots & p_{0,m} \\
\phi_{11} & \phi_{12} & \ldots & \phi_{1n} & p_{1,0} & \ldots & p_{1,m} \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
\phi_{n1} & \phi_{n2} & \ldots & \phi_{nn} & p_{n,0} & \ldots & p_{n,m}
\end{pmatrix}
\begin{pmatrix}
w_0 \\
w_1 \\
\vdots \\
w_n \\
c_0 \\
\vdots \\
c_n
\end{pmatrix}
= 
\begin{pmatrix}
f_0 \\
f_1 \\
\vdots \\
f_n \\
0
\end{pmatrix},
\]

where \( \phi_{ij} = \phi(||p_i - p_j||) \) and \( p_{i,k} \) is the factor of \( P \) with coefficient \( c_k \) evaluated at \( p_i \).

If all function values \( f_i \) are zero, then the trivial solution \( w_i = 0, c_j = 0 \) is valid. In that case \( F \) is the zero function. To avoid this solution, additional samples are placed inside (outside) the surface with positive (negative) function values. The position of these additional constraints influences the surface. Common placement rules are along the normal (inside, outside or both) or on a skeletal structure of the surface called the medial axis which is described in Section 2.6.
Chapter 1. Introduction