www.cadanda.com Computer-Aided Design AND APPLICATIONS™

Automatic Virtual Alignment of Dental Arches in Orthodontics

Yokesh Kumar[1], Ravi Janardan[1] and Brent Larson[2]

[1]Dept. of Computer Science and Engineering, University of Minnesota,
{kumaryo, janardan}@cs.umn.edu
[2]Division of Orthodontics, Dept. of Developmental and Surgical Sciences, University of Minnesota,
larso121@umn.edu

ABSTRACT

A key task in orthodontic treatment planning is to align the teeth in a given lower and upper arch so as to establish an ideal occlusion (i.e., contact relationship), subject to certain dental constraints. A simulation-based approach is introduced to establish a near-optimal occlusion based on certain dental constraints that are defined using features on tooth surfaces (e.g., cusps, ridges, incisal edges etc.). The alignment process is modeled as the simulation of a hypothetical spring-mass system where masses representing teeth are connected and influenced by springs representing dental constraints. The set of constraints chosen is based on well-known guidelines to achieve normal occlusion and to detect the most common type of orthodontic errors. The design and implementation of such a simulation-based system are discussed and experimental results are provided to demonstrate the efficacy of the approach.

Keywords: Digital orthodontics, dental features, dental alignment, malocclusion, surface mesh.
DOI: 10.3722/cadaps.2013.371-398

1 INTRODUCTION

Dental occlusion is the contact relationship of a given arrangement of teeth on the (maxillary) upper and the (mandibular) lower arch. Orthodontic treatment aims to optimize the dental occlusion so that proper function and esthetics of teeth are achieved. This involves the *alignment* (re-positioning) of teeth relative to one another in each arch and proper matching of the contact regions between teeth in different arches. The actual process of occlusal treatment is carried out by applying mechanical forces on teeth using dental appliances such as brackets and wires.

A key task in planning orthodontic treatment is to understand the best possible outcome (in terms of occlusion) for a given pair of opposing arches. For a given pair of arches, the treatment goal is to realize an arrangement that achieves the best possible alignment and occlusion, subject to certain constraints.

There can be significant variability in dental anatomy across different patients, including differences in the tooth size and shape, crowding, type of malocclusion, tooth condition (missing or restored teeth), and the extent of improper contacts. Therefore, across patients, this results in differences in what constitutes an optimal alignment, i.e., differences in the quality of the dental occlusions that can be realized from one patient to another.

Traditionally, the process of establishing proper occlusion has been simulated by re-positioning teeth manually in stone or plaster models cast from the dental impressions of the patient's arches. Lately, however, computer software

such as emodel® and SureSmile® have become available to do this simulation digitally. However, these tools require a fair amount of user input to plan the correct placement using an interactive visualization.

Our objective is to *automatically* compute the *optimal occlusion*, which we take to mean an arrangement of teeth in a given pair of opposing arches that best satisfies the constraints of an ideal occlusion, as envisioned by an experienced practitioner. The alignment of teeth has an *intra-arch* (resp. *inter-arch*) component that aligns teeth with respect to other teeth on the same (resp. opposing) arch. This paper addresses the problem of simulating the process of establishing proper occlusion of teeth in a virtual digital setting *before* the actual treatment starts. Thus, the aim is to enable the orthodontist to visualize and analyze the automatically-generated optimal occlusion which guides the subsequent treatment process.

The use of such a virtual tool makes it possible to study the optimal occlusion outcomes of different scenarios in planning, e.g., the optimal occlusion outcome neglecting an existing tooth (extraction). Another example is to optimize occlusion for one of the occlusion classes, Class I, II, or III (described in Section 3.2). These tools also have tremendous benefits in the training of dental students as they allow the trainees to observe and emulate different possible outcomes for a given case, as well as learn about the effectiveness of the different treatment plans on cases already solved by experts.

1.1 Challenges in Virtual Alignment

The computation of the optimal occlusion of a given set of teeth is difficult for several reasons, as discussed below.

Evaluation of dental occlusion: The orthodontic literature contains very few robust, quantitative metrics to evaluate the quality of the dental occlusion in a given arch [7, 8]. Such a metric is important in determining if an intended correction to the arrangement of teeth brings about a change in the overall functionality. A widely used evaluation metric is the American Board of Orthodontics (ABO) model grading system [17]. (The ABO grading system is described in detail in Section 3.2.) This system is used to evaluate the test cases in orthodontic board exams. It presents several types of common errors that can be evaluated to get an objective score based on well-agreed-upon constraints. For example, the absolute difference in height of the so-called marginal ridges of the posterior teeth (defined in Section 1.5) is mapped to an integer score in the interval [0,2]. Such a scheme is useful, but is very coarse, leaves a large room for approximations, and does not capture many of the finer aspects of orthodontic occlusion. There are detailed studies on the ideal arrangement of teeth in terms of the relationship to the archform, adjacent teeth, occlusal plane and the teeth from the opposing arches [3]. However, there are no established metrics to measure the relative "goodness" of a given arrangement with respect to a normal one.

Algorithms for computing occlusion: There are no known practical algorithmic techniques to search for the optimal occlusion for a given input case. This means that most of the occlusions established currently are primarily a result of "try-and-refine" procedures guided by the experience of the user. This is very tedious and has a high risk of leading to sub-optimal occlusions that may require costly re-treatment.

Properties of ideal occlusion: As mentioned above, there are many well-agreed-upon constraints that must be satisfied for a good dental occlusion. However, the relative importance of these constraints is not established and, as a result, the opinion of different practitioners regarding this may differ. This is important because many of these constraints are essentially of a conflicting nature and cannot be simultaneously satisfied in real-world cases. For example, the marginal ridges of posteriors must be at the same vertical height. However, ideal occlusal contacts require a posterior tooth to be in tight contact with the teeth from the opposing arch. Thus, there is an intra-arch and an inter-arch constraint that are in conflict with respect to the height of the teeth. Also, there is not much statistically-observed information available on which of the constraints are the most important in defining the course of alignment in real-world cases.

Special cases in occlusion: Often the types of occlusion in orthodontics are classified as Class I, II, or III, based on the relationship of the posterior teeth of the opposing arches (see Section 3.2). These classes are also determined by whether a premolar is extracted from the upper or the lower arch. It is important to know which of these classes apply to a given case, as the alignment plan is different for each of them. However, it is, in general, difficult to find the occlusion class of a given model automatically.

Computational issues: There are additional computational issues in the virtual setting as compared to a physical setup environment. For example, we need real-time collision detection and avoidance between the teeth in a virtual world, as they move from their initial positions to their final positions during the treatment simulation, whereas this is naturally "enforced" in a physical setup [6]. Also, the search-space of the possible alignments is much larger in a virtual environment. This is due to the large number of potentially sub-optimal arrangements generated by an unguided, automatic movement of teeth compared to the physical setup performed by an expert.

1.2 Our Goals and Contributions

There is considerable variation and sophistication in the manual orthodontic treatment planning techniques used in the clinic. In this paper, we consider the basic problem of aligning teeth from two opposing arches so that an optimal occlusion can be achieved on a broad range of cases. (Our notion of optimality is as stated at the beginning of Section 1.) We note that the use of advanced techniques such as tooth extraction (commonly premolar), trimming of crowns (commonly anteriors), etc. (see Class I, II, III description in Section 3.2) are beyond the scope of this paper.

1.3 Organization of the Paper

Section 2 describes the related work in this field. Section 3 describes the some known approaches to characterize the problem of alignment in orthodontics. These ideas are used to model the alignment problem in terms of satisfying certain dental constraints in Section 4. An implementation of this model along with a simulation algorithm is described in Section 5. Finally, Section 6 presents the software implementation and the experimental results obtained on real-world datasets.

1.4 Input and Assumptions

We assume that we are given a pair of opposing dental arches of a patient. These arches are given as separate 3D surface meshes that are usually obtained by laser-scanning a plaster model of the patient's dental impressions. In fact, we assume further that the arches have been pre-segmented into individual tooth objects (submeshes) and that the tooth objects are given in order along each arch. This is reasonable since tools exist for carrying out such a segmentation; see for instance [12, 14]. The constraints on the teeth can be described in terms of the relationships between certain immutable intrinsic features on tooth surfaces, e.g., cusps, marginal ridges, incisal edges, etc., as shown in Figure 1 and defined below in Section 1.5. These features can be automatically computed using the techniques given in [13].

1.5 Dental Anatomy and Dental Features

In this section, we define key elements of dental anatomy and dental features. All definitions are illustrated in Figure 1. More information on dental anatomy and dental features can be found in standard textbooks, such as [20].

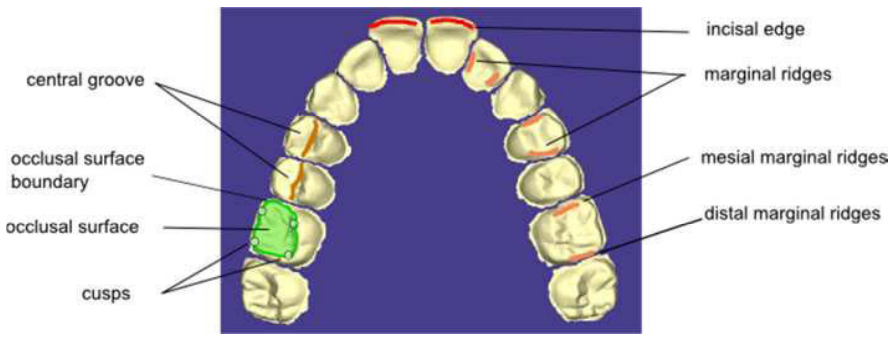


Fig. 1: Intrinsic features on the surface of teeth. (*All figures in the paper are best viewed in color.*)

1.5.1 Dental anatomy

Teeth are classified as *incisors*, *canines*, *premolars* and *molars*. Each *dental arch*, i.e., row of teeth, can be divided into a *left* and a *right* side. Each side has two incisors (*central* and *lateral*), one canine, two premolars (*first* and *second*) and three molars (*first*, *second* and *third*). The incisors and canines are collectively called *anteriors* and are used in cutting action. The premolars and molars are called *posteriors* and are involved in chewing action.

The inner (resp. outer) part of the tooth on the tongue (resp. face) side is called the *lingual* (resp. *buccal*) side. A tooth's surface towards the front (resp. back) of the arch, i.e., towards (resp. away from) the central incisors, is called the *mesial* (resp. *distal*) side.

A suitably chosen line through the mesial and distal side of each tooth defines the *mesiodistal* line of the tooth. Similarly, the lingual and buccal (or labial) sides define the *buccolingual* line. These lines are important in feature identification and in understanding tooth functionality.

1.5.2 Dental features

Next, we define some of the features on tooth surfaces that are important in orthodontic treatment planning. The surface of a tooth resembles a terrain with "mountain peaks", "ridges" and "valleys". For ease of understanding, it is convenient to define dental features in terms of these terrain-like features.

Incisal edges are the sharp ridges on the anteriors (incisors and canines) running along the mesiodistal line. The *cusps* are the mountain peak-like structures on the crown of the tooth of canines, premolars and molars. Each cusp has *cusp ridges* radiating from its tip, similar to ridges that connect mountain peaks to valleys on a terrain; these can be used to define other features such as the occlusal surface boundary. Canines have a single cusp which plays an important role in determining the overall quality of the alignment. The premolars (resp. molars) have 2 or 3 (resp. 4 or 5) cusps depending on the arch (upper or lower) and the individual in question. These cusps are named according to their position on the tooth surface. For example, a molar would have a *mesiolingual cusp* situated on the mesial and lingual side of the tooth. Similarly, for a *mesiobuccal*, *distolingual*, *distobuccal*, *lingual* and *buccal cusp*. Some of these cusps are shown in Figure 1.

The *occlusal surface* of a posterior tooth is the area of the tooth surface where chewing occurs. This is also the contact area between corresponding posterior teeth from opposing arches. The *marginal ridges* are located at the mesial and distal ends of the occlusal surface. These are the regions where the mesial or distal walls of a tooth make contact with the occlusal surface. Thus, each tooth has a *mesial* and a *distal* marginal ridge.

The occlusal surface of a posterior tooth is bounded on the buccal and lingual sides by the cusp ridges (inclusive of the cusps). On the mesial and distal sides, the occlusal surface is bounded by the two marginal ridges. This provides a boundary around the occlusal surface area called *occlusal surface boundary*. Thus, the occlusal surface boundary is a curve on the tooth surface that connects the marginal ridges, cusp ridges, and the cusp peaks.

Finally, the *grooves* are the depressions and fissures on the occlusal surface of a posterior tooth that resemble riverbeds and valleys on a terrain. There are various types of grooves and corresponding classification and naming conventions. For our study, we are interested in the long grooves running along the mesiodistal line of the tooth, called *central developmental grooves*, or simply *central grooves*. We also use the *buccal grooves* on molars, which are depressions on the buccal side of the crown running from the occlusal surface towards the gums.

In addition to the above-mentioned features, which are intrinsic to the tooth surface, there are other features that are derived from these intrinsic features and are important in alignment. These include the archform, the occlusal plane and the interproximal contact points [20]. The *archform* is defined as an appropriate smooth curve through the incisal edges of anteriors, canine cusps and the buccal cusps of molars and premolars. The archform determines the overall quality achievable by the alignment process. The *occlusal plane* is defined as a smooth surface passing through the occlusal or biting surfaces of the teeth. It is an imaginary surface at which the upper and lower arches meet and is important in establishing the vertical positioning and the buccolingual orientation of teeth in the final alignment. The *interproximal contact points* are the contact points between two adjacent teeth on the same arch; they are viewed as derived features as they are not intrinsic to a tooth but can be determined from the surfaces of two adjacent teeth. They are important in establishing the correct vertical alignment of posterior teeth from opposing arches.

2 RELATED WORK

As mentioned above, the traditional approach of alignment is to re-position the teeth on a stone or plaster model that is mounted on an articulator. However, tools such as SureSmile® [18] and *e*model® [19] provide the ability to manipulate 3D tooth models in a virtual environment. These computer-aided tools are becoming increasingly popular because of the facilities they provide to interact with, visualize, and animate treatment plans. Also, these tools allow long-term, robust storage of models and accurate measurements of tooth features and dimensions. However, the alignment process itself has not changed much and still involves users manipulating the position and orientation of individual teeth, mostly using a mouse-based interface.

We mention briefly two areas where problems related to dental occlusion arise. In prosthodontics, the goal is to reconstruct portions of an arch through the use of dentures and implants. The key task here is to align ideal tooth objects optimally along a predetermined archform which can be chosen based on the alveolar bone in the mandible [1]. The problem in orthodontics is different as one has to work with patient-specific teeth (which may not be ideally-shaped) and the current archform determined by their arrangement. Thus, the current archform cannot be altered arbitrarily and must respect certain biological limits.

In craniomaxillofacial surgery, the goal is to establish the optimal dental occlusion of a given pair of *unsegmented* upper and lower arches [6, 9]. This involves computing the translations and rotations through space that would "fit" the rigid upper and lower arch meshes together to achieve maximum intercuspation (maximum occlusal contact area). Most of these techniques try to align arches based on some intrinsic features on the teeth, e.g., the grooves and marginal ridges on the upper arch are aligned with the buccal cusp ridges and incisal edges on the lower arch in [6]. This is different from the problem we consider here, where an individual tooth can move within the arch. Also, in orthodontic cases, the optimal maximum intercuspation is not known beforehand and is related to finding the optimal alignment of teeth. For example, given a pair of opposing arches that have good intra-arch alignment (no malocclusions), one can maximize the contact area between opposing arches. However, in the presence of orthodontic errors these criteria may not apply.

Another approach is to compute a close surface match of the opposing arches by moving the arches through space. This usually involves minimizing some function of the distance between the two occlusal surfaces of opposing arches [6]. As noted in [6], this is effective if the two surfaces to be matched are close to each other, in which case such a minimization converges to a very close fit quickly. However, this technique requires establishing correspondences between the vertices of the two meshes and the errors due to this may be very large when the meshes are far apart. Also, wrong correspondences can lead to a locally optimal arrangement wherein a tooth is "trapped" among other teeth in a sub-optimal configuration. Note that this can be a frequently-occurring situation in orthodontic cases where there is essentially some amount of crowding and malocclusions to be expected.

All the techniques mentioned above are essentially "global" (single optimization function) and work on the entire arch as an individual unit, ignoring the "local" effects in the neighborhood of each tooth, thus, being sub-optimal by nature for the class of our alignment problems. Also, they completely ignore the intra-arch alignment which is very significant (e.g., the upper anterior teeth must be aligned to be esthetically pleasing). Thus, they fail to exploit the critical properties observed and distilled over the years regarding the correct dental occlusion in humans.

3 FORMULATION OF THE ALIGNMENT PROBLEM

In this section, we first describe the concepts and ideas involved in defining the correct occlusion in normal and abnormal cases (Section 3.1). Next, we present a set of constraints based on the commonly observed occlusion errors in orthodontics, which can be used to quantify approximately the quality of occlusion in a given pair of arches (Section 3.2). Finally, we discuss the inter-dependencies and conflicts among the different constraints (Sections 3.3 and 3.4).

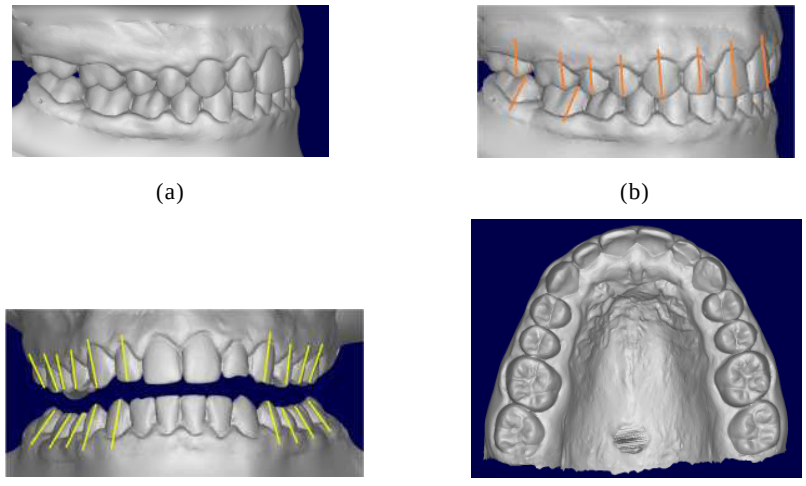
3.1 Orthodontics Background for Occlusion

This section describes the key ideas in understanding and characterizing the concepts for achieving correct alignment and occlusion. A more detailed discussion of other influences such as the underlying alveolar bone, roots of teeth, functioning of jaws, etc. can be found in texts such as [4, 20]. These properties determine the correct placement of a tooth vertically, horizontally, and along the archform; the placement depends on not only the tooth in question but

also on many nearby teeth from the same and opposing arch. A study of all these factors is beyond the scope of this paper. Here, we consider the problem of occlusion with respect to the position and surface features of teeth only.

3.1.1 Characterization of normal occlusion

There have been many studies to get an accurate characterization of correct dental alignment in humans. The main purpose of these studies was to find a set of properties that are applicable to a wide variety of normal cases. Early work [4] established the correct occlusion in terms of the position of the upper first molar. All the normal cases were observed to have the mesiobuccal cusp of the upper first molar align vertically with the buccal groove of the lower first molar. The first molars are large teeth that are among the first permanent teeth to erupt and, thus, influence the placement of other teeth on the arch. They also determine the extent of separation of the jaws (while open) and are the most consistent in assuming their correct positions on the arch. Thus, it was hypothesized that if the first molars could be locked in correctly, then this would lead to the correct relationship between other anterior and posterior teeth. However, it was noted that in spite of the importance of the first molars, their correct placement still left enough room for some abnormal occlusion to occur. These were the Class I, II and III relationships defined by Angle [4] (see Section 3.2).



(a) (b) (c) (d)

Fig. 2: Important keys to normal occlusion: (a) Molar relationship. (b) Crown angulation (shown orange). (c) Crown inclination (shown yellow). (d) An arch with no tooth rotations.

A more extensive characterization of correct occlusion was given by Andrews [3] as a set of six “keys” of normal occlusion observed in more than one thousand cases. These six keys were shown to be necessary and sufficient to describe the normal occlusion in a majority of the cases:

1. *Molar relationship:* As mentioned above, the upper first molar’s mesiobuccal cusp occludes with the buccal groove of the lower first molar. The mesiolingual cusp of the upper first molar should make contact with the central groove of the lower first molar. Also, the upper first molar should be angulated so that the distal marginal ridge aligns with the mesial marginal ridge of the lower second molar. Figure 2(a) shows an example of the correct molar relationship.
2. *Crown angulation:* We define the *long axis* of a tooth as the vertical line that passes through the approximate center of the buccal surface of the tooth crown (the crown of a tooth is the visible part of tooth above the gumline). Traditionally, the mid-developmental ridge on the buccal side of the anteriors and the premolars is taken to be the long axis. However, this ridge is difficult to compute as it may not be prominent enough, especially on lower incisors. In normal occlusion, the gingival end of the long axis of the teeth should be more

distal when compared to their occlusal end. Note that an angulated tooth will take up more space along the archform than in the normal case. Figure 2(b) shows an example of the distal angulation of crowns.

3. *Crown inclination:* The crown inclination is the labiolingual or buccolingual inclination of the tooth's crown. When examined from the mesial or distal side, if the gingival end of the crown is more lingual (resp. labial or buccal) than the occlusal end, it is referred to as positive (resp. negative) inclination. The upper (resp. lower) anterior teeth should have a mild positive (resp. negative) inclination, whereas both the upper and lower posteriors should have a positive inclination. The actual angle of inclination varies with each tooth type. Figure 2(c) shows an example of crown inclination for some of the teeth.
4. *Rotations:* The teeth should be free of any rotations with respect to the hypothetical archform. This is important for the proper occlusal contact and functionality (e.g., aligned incisal edges are more efficient in cutting). Figure 2(d) shows an arch without any rotations.
5. *Tight contacts:* The actual occlusal contacts should be tight without any space between the occluding surfaces of opposing arches. Figure 2(a) shows an example.
6. *Occlusal plane:* In most cases the occlusal surface, where the teeth from opposing arches meet, ranges from a flat plane to a slight *curve of Spee* [20]. However, a flat occlusal plane is recommended as the treatment goal given the fact that the curve of Spee increases naturally with growth and function.

3.1.2 Evaluation of orthodontic cases

Various schemes for the objective evaluation of the quality of occlusion have been proposed in the literature. The goal here is to objectively assign a score to the quality of the finished cases using certain measurements on the arches. Some of these are described in [7, 8]. A characteristic of such an evaluation is that the scores must be based on features on the tooth surfaces that can be computed easily and precisely. Also, they must be applicable to a large variety of cases and at the same time point out minor but important inadequacies in tooth positions. Based on extensive studies of thousands of cases over several years by experts, the ABO has established a scoring system to evaluate the quality of cases presented by students training in orthodontics [17]. This objective score is calculated from a set of intrinsic features on tooth surfaces (e.g., cusps, grooves, marginal ridges, incisal edges, etc.) that can be measured unambiguously in a majority of cases. The properties measured for scoring are very similar to the six keys to normal occlusion and are described below in Section 3.2.

3.1.3 Our approach to correct occlusion

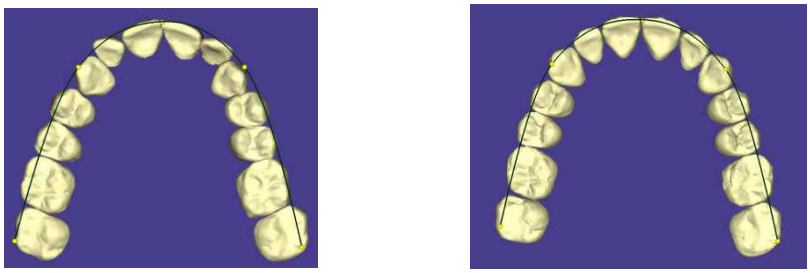
Some of the techniques to capture the main factors in orthodontic treatment were described above. For example, Section 2 described factors defining normal occlusion, whereas Section 3.1.2 described the common errors that occur in abnormal occlusions. These factors are well-rooted in the experience of experts and are justifiable in terms of tooth functionality, anatomy, and esthetics. Thus, these factors eventually influence the current practice in treatment planning. Our search for techniques to automate alignment and re-establish dental occlusion uses these ideas to guide and formulate methods that work for a majority of cases. Our hypothesis is that the re-positioning of teeth to correct as many of the common errors as possible among those defined in the dental literature such as the ABO model grading system will lead to improved dental occlusions for the case under consideration.

3.2 Constraints Influencing Occlusion

This section describes the formulation of the alignment problem in orthodontics in terms of various constraints that govern the quality of occlusion. As mentioned in Section 2, most of the related formulations of alignment problems treat the arches as individual rigid bodies that can only be translated and rotated through space as a single entity. Our formulation is to make the arches more "stretchable" by allowing individual teeth to move differently from one another. Thus, our approach can potentially explore many more arrangements of tooth positions and should lead to better outcomes.

We begin by describing the various constraints that must be satisfied by arches in correct occlusion, as given in the ABO model grading system. Some of the principles are modified to keep the computational model simple while not having a significant impact on the quality. It is worth observing these constraints in the context of the properties of normal occlusions [3].

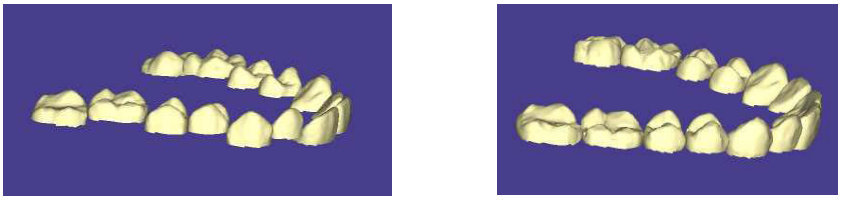
Alignment constraint (AC): The alignment constraint requires that the incisal edges, canine cusps and the buccal cusps of posteriors on the arches are aligned along a smooth archform as shown in Figure 3. The original constraint also requires the grooves of the upper arch to be aligned along a smooth curve. However, we can handle the upper arch similar to the lower arch without affecting the correctness. This is arguably the most important constraint as it affects the primary layout of teeth along the alveolar bone, which in turn influences most other constraints. Also, it determines the esthetics on the anterior region and proper cutting (resp. chewing) function on the anteriors (resp. posteriors). The upper and lower lateral incisors and second molars contribute to the majority of errors, as was observed in real-world test cases.



(a) (b)

Fig. 3: Alignment constraint: (a) Malocclusion and incorrect alignment in both anteriors and posteriors. (b) Properly aligned (buccal) cusps, grooves and incisal edges. Also, premolars with undesirable rotations are shown in (a). (The proper alignment here and in Figure 4 and Figure 5 was obtained using our algorithm in Section 5.)

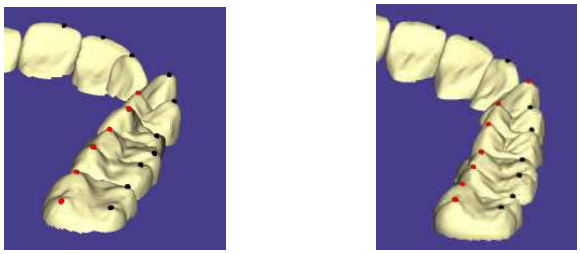
Marginal ridges constraint (MRC): Marginal ridges determine the vertical positioning of the posterior teeth with respect to their neighbors on the same arch. The marginal ridges of all properly formed teeth must be at the same height. Figure 4 gives an example of correct and incorrect marginal ridge heights. This ensures that the proper occlusal contacts will be made with the opposing arches, e.g., the distal marginal ridge of the upper first molar occludes with the mesial marginal ridge of the lower second molar as shown in Figure 2(a). The most common errors occur in the first and second molars.



(a) (b)

Fig. 4: Marginal ridges constraint: (a) Errors in marginal ridges. (b) Correct vertical positioning of teeth leads to the same height of marginal ridges.

Buccolingual inclination constraint (BIC): The buccolingual inclination of a tooth is measured as the angle between the line joining its buccal and lingual cusps and the occlusal plane. The crown inclination can be negative or positive as defined in Section 3.1.1. It is used to determine the crown inclination of the posterior teeth. Proper occlusion in maximum intercuspation requires that the buccal and lingual cusps of the posterior teeth be at the same height relative to the occlusal plane. Figure 5 shows an example. The second molars contribute to the majority of inclination errors.



(a) (b)

Fig. 5: Buccolingual inclination constraint: (a) Errors in buccolingual inclination at the posteriors. (b) Corrected inclination showing buccal and lingual cusps at the same height.

Occlusal contacts constraint (OCC): These constraints determine the extent of occlusal contact in the posterior teeth from opposing arches. Large occlusal contact areas on posteriors are vital to ensure that the chewing function is carried out well. Figure 6 shows examples of incorrect and correct occlusal contacts. This constraint is defined on the *functional cusps*, i.e., buccal cusps of the lower and lingual cusps of upper posteriors. The most common inadequacy of contacts is found in the second molars.

(a) (b)

Fig. 6: Occlusal contacts constraint [17]: (a) Errors in occlusal contacts leads to improper bites at posteriors. (b) Correct occlusal contacts are tight.

Occlusal relationship constraint (ORC): This is used to ensure the anteroposterior relationship of the posterior teeth. This includes the crucial first molar relationship discussed in Section 3.1.1. This constraint extends it and defines the vertical alignment, with respect to the archform, of each upper posterior with a corresponding interproximal contact point or groove of the lower arch. For example, as shown in Figure 7(a), the canine cusp of the upper arch must align with the interproximal contact points between the lower canine and the premolar. Similarly, the upper first premolar cusp must align with the interproximal contact point between the lower first and second premolars.

This also leads to the classification of malocclusion errors as a Class I, Class II, or Class III relationship. A Class I relationship is the one described above. A Class II (resp. Class III) relationship occurs when a premolar is extracted from the upper (resp. lower) arch thereby “shrinking” the room on the archform. Therefore, in a Class II relationship, the upper first molar is located ahead of the lower first molar and, thus, the mesiobuccal cusp of the upper first molar aligns with the interproximal contact points of the lower second premolar and the first molar (see Figure 7(b)). Similarly, in a Class III relationship, the buccal cusp of the upper second premolar aligns with the buccal groove on the lower first molar (see Figure 7(c)).

As mentioned earlier, we only handle occlusion for the Class I relationship. The consideration of Class II and III relationships is beyond the scope of this paper.

(a) (b) (c)

Fig. 7: Occlusal relationship constraint and malocclusion classes [17]: (a) Class I. (b) Class II. (c) Class III.

Interproximal space constraint (ISC): This determines the amount of interproximal space between two adjadent teeth on the same arch. Such gaps between teeth are unaesthetic and can lead to food impaction. Figure 3(b) shows an example of correct interproximal spacing between teeth.

Overjet constraint (OC): This constraint determines the transverse relationship between teeth from opposing arches when viewed from the mesial or distal side. On posterior teeth, it ensures that the functional cusps occlude with the central grooves of the opposing arches and are captured adequately by the occlusal contact constraint. However, on the anteriors, the lower incisal edges must be in contact with the lingual side of the upper anteriors. But this must not be achieved simply by the over-inclination of anteriors of a single arch. There are reasonable correct range of angles for the labiolingual inclination of the anteriors (see Figure 8). The most common errors are found among the incisors.

(a) (b)

Fig. 8: Overjet constraint [17]: (a) Anterior teeth. (b) Posterior teeth.

3.3 Modeling of Constraints: The Constraint Graph

A conceptual diagram of various constraints acting on different teeth is shown in Figure 9. We refer to this as the *constraint graph*. The nodes represent individual teeth and the edges represent a constraint that affects two different teeth. To reduce clutter, Figure 9 shows the nodes on one side (say, left) only. However, this does not mean that the left and right sides of the arches can be aligned independently, as some constraints such as interproximal space may influence both sides simultaneously (at the central incisors). Some of the constraints relate a tooth to a more global landmark of the arches, e.g., the archform and the occlusal plane. Thus, the archform and the occlusal plane are also represented as nodes that control the alignment and the overjet of teeth, respectively. The alignment, buccolingual inclination, marginal ridges, and interproximal space constraints are the *intra-arch constraints*, i.e., they influence the position of a tooth relative to the other teeth on the same arch. Similarly, the occlusal contact, occlusal relationship, and overjet constraints are the *inter-arch constraints* and determine the quality of occlusion of opposing arches. Note that all the intra-arch and occlusal contact constraints apply to both the arches. Some of these constraints are not shown in Figure 9, again to reduce clutter.

Each edge in Figure 9 defines a type of constraint between the two teeth on which it is incident, with respect to some feature. The features used at the endpoints of the edge may be of different types (e.g., a cusp of a tooth must make contact with a groove of another tooth) or may be defined using features of multiple teeth. For example, the occlusal relationship constraint defines the relationship between the cusp of the upper canine and the interproximal

contact point of the lower first premolar and the adjacent canine. (This interproximal contact point can be computed from two adjacent tooth objects, but is not a direct feature of either of the teeth.) Buccolingual inclination is the only exception in that it affects only a single tooth and, hence, both endpoints of the edge are incident on the same tooth (the edge is a loop). Two edges incident on the same tooth may act on different subsets of its features, e.g., the occlusal relationship constraint influences only the buccal cusps of a molar, whereas the buccolingual inclination influences both buccal and lingual cusps.

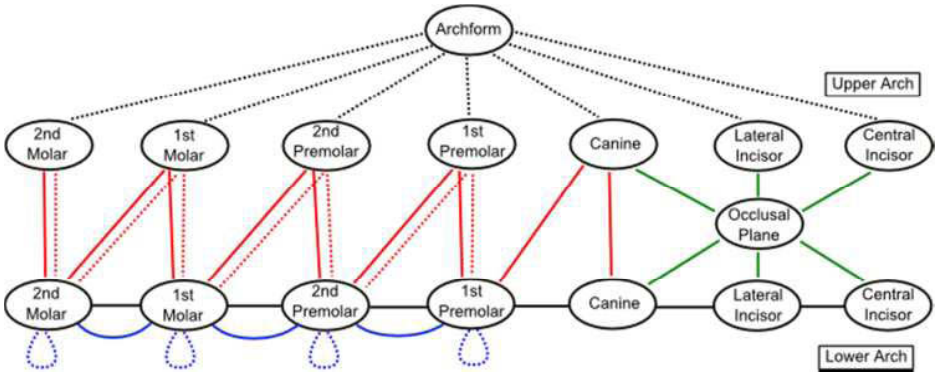


Fig. 9: A part of the constraint graph with constraints affecting the occlusion of teeth on one side (say, left) of the upper and lower arch. The different teeth are represented by nodes. Alignment constraints (shown dotted black), interproximal space constraints (shown black), marginal ridge constraints (shown blue), and buccolingual inclination constraints (shown dotted blue) constitute the intra-arch constraints. Similarly, occlusal relationship constraints (shown red), occlusal contact constraints (shown dotted red), and overjet constraints (shown green) constitute the inter-arch constraints. Intra-arch constraints are only shown for a single arch to reduce clutter. All the intra-arch and occlusal contact constraints apply to both the arches.

There are a few exceptions among the constraints such as the buccolingual inclination mentioned above. These arise due to the observed abnormal anatomical features in orthodontic cases. For example, the lower premolars often have *diminutive* (not well-formed) cusps on the lingual side, thus, resembling the adjacent canine more than the second premolar. Also, due to significant wear, restoration, or unusual anatomy, some of the features like cusps, marginal ridges and cusp ridges may not be well-represented. Thus, the particular constraint (edge) corresponding to these features is not considered while determining the occlusion. This is justified because it is difficult (and may be incorrect) to use the diminutive cusp to influence occlusal contacts or the buccolingual inclination of the tooth the cusp belongs to.

Also, it has been observed in practice that for a reasonable alignment and occlusion among well-formed teeth, the combination of occlusal relationship, occlusal contact, and alignment constraints subsumes the overjet constraint for the posteriors. (This is why overjet is not shown for posteriors in Figure 9.)

3.4 Conflicts among Constraints

It is easy to see that the satisfaction of some of the constraints described above will lead to conflict among certain others. The conflict among the constraints can be classified, according to the type of tooth movement involved, as horizontal and vertical conflicts.

- Horizontal conflicts: Consider the conflict between the alignment and interproximal space constraints. The crowding of teeth leads to improper layout of teeth on the archform. In order to correct the alignment, one needs to rotate and align a tooth along the archform. However, this is difficult as there is no room between the teeth on the archform. Thus, some teeth must be moved out on the archform (distally) so that enough room can be created to accommodate the tooth in question in its new position. In addition to these intra-arch constraints, the occlusal relation also is in conflict for the horizontal positioning of the teeth along the archform.

- Vertical conflicts: These are constraints that influence the vertical position of the tooth. Due to marginal ridges and buccolingual inclination, the intra-arch constraints try to establish a uniform height for the posterior teeth within the same arch. However, due to occlusal contact and overjet, the inter-arch constraints try to position the teeth so that the contact area is maximized.

In addition to these conflicts, there are influences due to the combination of both horizontal and vertical constraints, e.g., the correct occlusal contact (vertical positioning) depends on the establishment of correct occlusal relationships (horizontal positioning).

4 ESTABLISHING CORRECT OCCLUSION VIA CONSTRAINT-BASED ALIGNMENT

In this section, we describe an approach to achieve correct occlusion via the handling of the constraints described in Section 3.2, while keeping in mind the conflicts among constraints discussed in Section 3.4. We also discuss a general simulation-based alignment model based on a spring-mass system.

4.1 Handling of Constraints

Section 2 discussed the limitations of techniques that view the entire arch as a single object. Next, we look at the possible approaches to constraint-based establishment of dental occlusion. Given the constraints on occlusion, some natural questions are: Can the constraints be handled independent of each other? Also, is it possible to handle the constraints by partitioning them into disjoint groups that can be handled independent of each other? A little contemplation reveals that because of the natural conflicts between the constraints discussed in Section 3.4, such a strategy will not work. Also, the approach of first satisfying intra-arch constraints in isolation from the inter-arch constraints followed by handling of inter-arch constraints does not perform well (e.g., due to the vertical conflicts described in Section 3.4).

Currently, even the relative importance of individual constraints or of groups of constraints on the global alignment and occlusion problem is not well-understood. There is evidence that, in case of a conflict, some constraints such as AC and OCC are given more importance than others such as MRC. However, these are based on the observations of experts and tend to vary in practice. Finally, unusual dental anatomy may also contribute to further complications. For example, if some molar cusp is only slightly diminutive, it may still be considered for the buccolingual inclination constraint, which may cause non-tight occlusal contact.

4.2 Simulation-based Approach: A General Spring-mass Model

The discussion above clearly suggests the need for a more holistic approach to handle the constraints that considers all of them simultaneously. However, such a strategy must also be flexible enough to incorporate some of the observed wisdom about the relative importance of constraints.

A feasible way to realize the constraints and their influences on teeth is as a system of forces on a set of 3-dimensional (3D) rigid bodies. Consider each tooth as a 3D rigid body with its surface represented by a triangle mesh. The constraints as explained above act on points in 3D space that are defined either by a known tooth surface feature or which can be computed using these available features. Now, each constraint edge can be viewed as a “spring” connecting the two rigid body masses and the influence of the constraint can be viewed as a force exerted by the spring.

Note that the system described above is entirely hypothetical and is just one possible realization of the constraint graph shown in Figure 9. In the modeling of teeth as rigid bodies, properties other than their surface are chosen in a normalized fashion without consideration of their actual values (e.g., each tooth is taken to have unit mass). The situation with edges is also hypothetical as they do not have any “real” physical analogue and are only defined conceptually. Therefore, several properties assigned to a constraint edge depend on the solution approach that we employ. Some examples are: force in a spring which requires defining the non-stretched length of the spring and a suitable spring constant (which may reflect the relative importance of a constraint, as a stiff spring exerts more force). More details of a concrete realization of the constraint graph with respect to our actual implementation will be given later in Section 5.

Given such a model of the constraint graph (teeth as nodes and edges are stretched springs attached to features), one can visualize a simulation which re-positions the teeth (nodes) such that the overall system achieves a certain optimal state. This optimal state can be defined in many meaningful ways, e.g., the minimization of total energy in the

springs, or when no further tooth re-positioning is possible, etc. Various techniques exist to describe simulations on rigid-body systems, e.g., N-body simulations [2] and simulated annealing, that we can adapt to our approach. Note that a good simulation algorithm based on the approach described above must be robust across a range of actual values chosen for the properties (mass, spring constant, etc.).

The input to a simulation are the current teeth (with their features, the archform, and the occlusal plane) and a set of constraints to be satisfied, using which a set of nodes and edges for the constraint graph are created. Given this, the simulation algorithm proceeds to move the nodes in small steps in response to the forces exerted by the springs, while also updating the node and edge properties. One can also define various boundary conditions for the simulation process, e.g., teeth cannot penetrate each other at any time, maximum distance that a tooth can move, directions in which a given tooth can move, groups of teeth that must move together, etc. An important aspect of this simulation is that it must somehow detect and overcome a local minimum configuration during its execution, such as when teeth become interlocked in a sub-optimal position and are unable to move.

4.3 Advantages of the Simulation-based Approach

The simulation approach is motivated by the process that an expert follows while planning the correct occlusion on real-world cases. We believe that this is close to how an orthodontist plans the alignment, weighing all constraints "in parallel" from a global (arch-level) as well as from a local (tooth-level) viewpoint, thus satisfying both classes of constraints simultaneously.

As discussed in Section 1.1, it is difficult even for a human expert to always find the optimal occlusion. Also, it has been observed in this domain that an optimal arrangement cannot exist in a neighborhood of highly sub-optimal arrangements in the "search space". This means that there will be many near-optimal arrangements of teeth that closely resemble the optimal one. In Section 5.3, we describe how we can get past some highly sub-optimal solutions by detecting local minimum conditions in our simulation.

Recall that the goal of virtual alignment, as it stands today, is to assist in planning patient-specific treatment in the context of the best possible outcome achievable. The actual treatment may deviate from this plan for a variety of reasons, so the planned alignment may not be necessarily achievable. Given this, an arrangement with a nearly-optimal occlusion is as good as the optimal one. (This is also reflected in the coarse scoring system given in the ABO model grading system [17].)

Also, there are some practical benefits to using a simulation-based approach. It provides fast evaluations of different treatment scenarios, such as the best outcome after a given tooth is extracted, optimization for Class I, II or III occlusion, satisfying a subset of constraints only, etc. This is extremely useful as an educational tool that can quickly help visualize the final outcome of different strategies in occlusion planning.

Finally, real-world cases sometimes need very specialized alignment planning due to various reasons. However, in such cases, some of the existing dental occlusion errors (evaluated in terms of constraints) may be similar to the errors found in a majority of orthodontic cases. In such situations, a simulation-based approach can help the practitioner by generating arrangements that are in the vicinity of the optimal occlusion. Subsequently, these arrangements can be fine-tuned manually by the practitioner based on experience. (We do not envision the practitioner having to re-adjust the spring constants and re-run the simulation.)

5 IMPLEMENTATION OF THE SIMULATION: REALIZATION OF THE CONSTRAINT GRAPH

Sections 4.2 and 4.3 described a simulation-based approach to achieve occlusion and provided some rationale about the viability of this approach in comparison to some other techniques. In this section, a concrete implementation of such a simulation is described. This includes the details of representing teeth and their constraints in the constraint graph model. Also, the details of the simulation algorithm along with its various properties such as choice of time steps, boundary conditions, avoidance of local minima, etc. are discussed.

5.1 Node Properties

As mentioned earlier, each tooth is represented as a node. Thus, each node is modeled as a rigid body and corresponds to a tooth surface with specially chosen points that correspond to the surface features of interest. Each node is simply given one unit of mass. As the emphasis of the simulation is on the relative importance of constraints, there is no clear advantage to having one node weigh more than the other.

There are two special nodes representing the archform and the occlusal plane of the arches. These do not correspond to any tooth surface and are global entities using which derived features, such as interproximal contact points, can be represented. Also, as will be seen later, during the simulation the archform and occlusal plane nodes do not change their position unlike the other nodes (this is an example of a boundary condition).

Next, we describe the state of each node that is maintained during the simulation. The rigid body setup and simulation techniques used here are based on the ideas and implementation discussed in [5]. It is assumed that the nodes can undergo only translation and rotation transformations and cannot be deformed (change shape). This is important because a feature such as a cusp may not remain a cusp if the tooth surface is deformed. A node N_i has a centroid (center of mass) whose position at time t is denoted as $X_i(t)$ in the *world-space* coordinate system (see Figure 10). Each node also has a corresponding local *body-space* which has its origin (0,0,0) at the centroid of that node. Additionally, each rigid body N_i also has a rotation matrix $R_i(t)$ associated with it that describes its orientation in world-space at time t. The matrix $R_i(t)$ can be described using three column vectors as $R_i(t) = R_i^x(t), R_i^y(t), R_i^z(t))$ where, $R_i^x(t) = \left(r_{xx}, r_{xy}, r_{xz}\right)^T$, $R_i^y(t) = \left(r_{yx}, r_{yy}, r_{yz}\right)^T$ and $R_i^z(t) = \left(r_{zx}, r_{zy}, r_{zz}\right)^T$ correspond, respectively, to the directions of the x-, y- and z-axis of the body-space of N_i with respect to the world-space axes at time t.

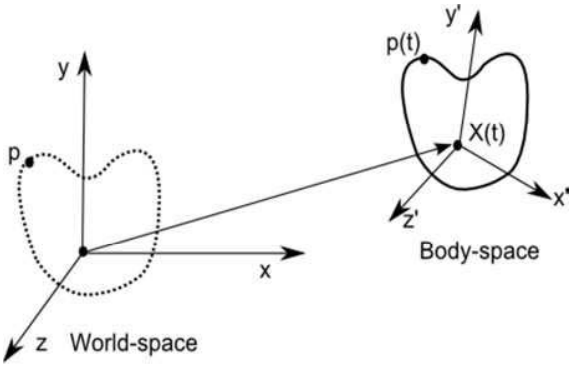


Fig. 10: The body-space axes (x', y', z') of a rigid body shown in the world-space (x, y, z). At time t, the rigid body is translated in world-space through $X(t)$ and the direction of its body-space axes is $R(t)$. At time t, point p on the rigid body is at $p(t) = R(t)p(0) + X(t)$, where $p(0)$ is the position of p with respect to $X(0)$.

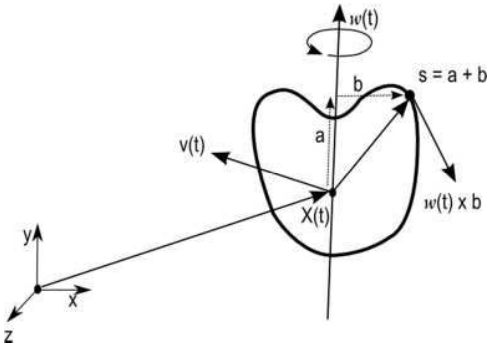


Fig. 11: A rigid body is shown with its linear velocity ($v(t)$) and angular velocity $\omega(t)$). The rate of translation of a vector s on the rigid body through world-space is $v(t)$ and its rate of change of orientation due to $\omega(t)$) is given by $\omega(t)) \times b = \omega(t) \times (a + b) = \omega \times s$.

Initially, at time t= 0, N_i's centroid C_i is positioned at $X_i(0)$ and has the x-, y- and z-axis parallel to the corresponding world axis, i.e., $R_i(t) = I_{3\times3}$, the identity matrix. For simplicity, a point p (in world-space) on the surface of N_i is described in the body-space of N_i. Thus, $p(t)$, the position of p at time t, is calculated as

$$p(t) = R_i(t)\, p(0) + X_i(t) \qquad (1)$$

where $p(0) = p - X_i(0)$ is the initial position of p with respect to the body-space origin $X_i(0)$ of N_i (see Figure 10).

The state description of each node has a position ($X_i(t)$) and an orientation ($R_i(t)$). The simulation would also move some nodes (teeth) under the influence of constraint- driven forces. This requires the tracking of the linear velocity and angular velocity of each node during the simulation. The linear velocity $v_i(t)$, is the rate of change of position of the node's centroid at time t ($X_i(t)$). Similarly, $\omega_i(t)$ is the angular velocity of the rigid body in world- space (see Figure 11). The direction of $\omega_i(t)$ gives the axis of rotation and the magnitude $|\omega_i(t)|$ gives the speed of rotation.

The rate of rotation of the axes of the body- space, denoted as the matrix $R_i(t)$, is computed from the angular velocity as described below. Consider a vector s in the body- space of N_i. In the body- space, s can be written as $a + b$ where vectors a (resp. b) are its components parallel (resp. perpendicular) to $\omega_i(t)$. The instantaneous movement of the point at the tip of s due to $\omega_i(t)$ is along a circle in the direction orthogonal to both b and $\omega_i(t)$ and its tangential velocity is given by $|\omega_i(t)||b| = \omega_i(t) \times b = \omega_i(t) \times (a + b)$ (as a and $\omega_i(t)$ are parallel) $\omega_i(t) \times s$. Thus, the rate of change of a vector $s(t)$ at time t due to $\omega_i(t)$ is $\dot{s}(t) = \omega_i(t) \times s(t)$ (see Figure 11). The rate of change of the column vectors of a rotation matrix $R_i(t)$ can derived similarly [5].

$$\dot{R}_i(t) = (\omega_i(t) \times R_i^x(t),\ \ \omega_i(t) \times R_i^y(t),\ \ (\omega_i(t) \times R_i^z(t) \qquad (2)$$

Given a vector a, define a matrix a^* as

$$a^* = \begin{pmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{pmatrix}$$

Then, the cross- product $a \times b$ can be written as $a^* b$, i.e.,

$$a \times b = a^* b = \begin{pmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{pmatrix} \begin{pmatrix} b_x \\ b_y \\ b_z \end{pmatrix} = \begin{pmatrix} a_y b_z - b_y a_z \\ -a_x b_z + b_x a_z \\ a_x b_y - b_x a_y \end{pmatrix}.$$

Using this, Equation 2 is written as

$$\dot{R}_i(t) = (\omega_i(t)^* R_i^x(t),\ \ \omega_i(t)^* R_i^y(t),\ \ \omega_i(t)^* R_i^z(t)) \qquad (3)$$

The quantities $v_i(t)$ and $\omega_i(t)$ are maintained in the state of N_i and are computed using the forces and torques acting on N_i. Given a node with linear velocity $v(t)$ and current position $X(t)$, the new position of node at time $t + \Delta t$ is computed using the Euler- forward technique [15] as:

$$X(t + \Delta t) = X(t) + \Delta t \cdot \dot{X}(t) = X(t) + \Delta t \cdot v(t) \qquad (4)$$

Similarly, the rotation matrix at time $t + \Delta t$ is computed using the current rotation matrix $R(t)$ and the rate of its rotation $\dot{R}(t)$ as:

$$R(t + \Delta t) = R(t) + \Delta t \cdot \dot{R}(t) = R(t) + \Delta t \cdot (\omega(t)^* R(t)) \qquad (5)$$

Each node may have many external forces acting on it at any given time as a result of the constraints, e.g., attractive forces (occlusal contact) and repulsive forces (minimum interproximal space). A vector $F = (f_x, f_y, f_z)^T$ describes the direction and magnitude of the force. Let there be k forces $F_{i1}, F_{i2}, \ldots, F_{ik}$ acting on rigid body N_i at time t at points $p_{i1}(t),\ p_{i2}(t), \ldots,\ p_{ik}(t)$, respectively. Then, the net external force on a node N_i at time t due to all the forces is

$$F_i(t) = \sum_{j=1}^{k} F_{ij} \qquad (6)$$

Similarly, the net torque on N_i due to all the forces is

$$\tau_i(t) = \sum_{j=1}^{k} \left(p_{ij}(t) - X_i(t)\right) \times F_{ij} \qquad (7)$$

The derivation of the linear (resp. angular) velocity via the computation of the linear (resp. angular) momentum using forces (resp. torques) for our simulation is based on [5]. The *state* $S_i(t)$ of a node N_i at a time t is the position, orientation, linear velocity and angular velocity of the node, i.e., $S_i(t) = (X_i(t), R_i(t), v_i(t), \omega_i(t))$. Given a set of nodes

representing rigid bodies, one can maintain the history of the nodes' movements during an interval of time under the influence of forces and resulting torques by using a sequence of states.

5.1.1 Collision detection among rigid bodies

It is clear that one of the important constraints that must be respected at any instant of the simulation is that no two teeth (nodes) surfaces must penetrate each other. (Unlike the orthodontic constraints discussed in Section 3.2, collision-avoidance is a constraint imposed by the simulation.) This means that the collision between teeth must be evaluated at all-time instants of the simulation. Existing solutions to collision detection between two tooth-like non-convex surfaces are computationally expensive [11, 16].

We use a simplified and efficient approach to collision detection among teeth by observing that all collisions can be classified into two groups: (a) inter-arch collisions: those between teeth on different arches (occlusal contact) and (b) intra-arch collisions: those among adjacent teeth on the same arch.

The inter-arch collisions can be checked by simply checking for penetrations on the occlusal region of each tooth involved. Consider tooth objects A and B. For each of these tooth objects, a hierarchical, tree-based bounding box data structure for their 3D mesh vertices is created; these structures are denoted as H_A and H_B, respectively. Thus, the highest level of H_A and H_B corresponds to the bounding-box of all the mesh vertices of A and B, respectively. If the surfaces of A and B penetrate each other, then, the bounding-boxes at the highest level of H_A and H_B intersect. In other words, if two bounding-boxes do not intersect, then the surfaces corresponding to the vertices in their boxes do not penetrate and the search can be terminated. Otherwise, we search recursively on the corresponding sub-bounding-boxes of these bounding-boxes in H_A or H_B, until a collision is detected, or the set of sub-bounding-boxes is exhausted in one of the trees. This is based on a popular approach to collision detection among 3D surfaces [11], but is expensive because the structures H_A and H_B must be updated (or rebuilt) when teeth vertices are moved during the simulation.

Intra-arch collision checking can be done as follows. A vertex of a tooth mesh can be projected horizontally onto its nearest point on the archform. The projections of all the vertices of a tooth define an interval on the archform, where the left (resp. right) endpoint of the interval is the projection with least (resp. greatest) x-coordinate value. We consider two adjacent teeth as colliding if their corresponding intervals on the archform overlap. This is sufficient because eventually all teeth are expected to be aligned along the arch without any rotations and an overlap of the intervals of two teeth leads to a violation of interproximal space constraint. The projection intervals are computed to find the interproximal contact points between adjacent teeth. Due to this, the intra-arch collision checks are very efficient.

5.2 Edge Properties

Each edge of the constraint graph defines a constraint between features on nodes. An edge exerts a force on the nodes it connects through the endpoints where it is connected. The various characteristics of different edge types are discussed below.

Recall that the endpoints of an edge can represent an intrinsic feature (e.g., cusp, ridge, groove, etc.) or a derived feature (e.g., the interproximal contact point between two adjacent teeth). The interproximal contact point between two adjacent teeth was defined earlier as the midpoint of the overlap between the intervals representing the projection of each tooth on the archform. This is motivated by the following observation: The occlusal relationship constraint is defined as the alignment of cusps with interproximal contact points when viewed from the buccal side. Thus, it is seen that the occlusal relationship constraint is in fact defined for the alignment of the cusp and the interproximal contact point when both are viewed (using projections) with respect to the archform.

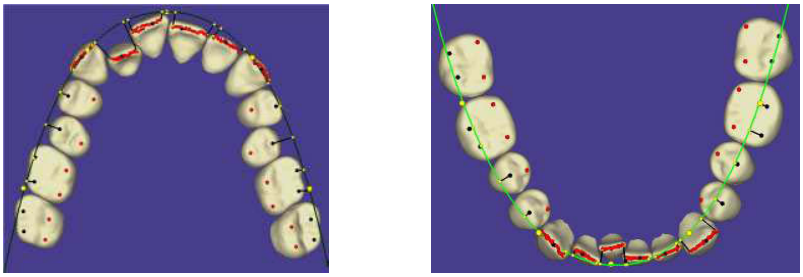
Each edge generates a corresponding force that drives the simulation. According to Hooke's law [15], the force due to an edge E (considered as a stretched or compressed spring) depends on the current length of the edge (E_L), the relaxed length (E_R), the spring constant (E_K), and some boundary conditions. In our simulation, the relaxed length (minimum energy state) of an edge is set to be very small, typically 0.05 mm. This is so because every constraint when satisfied (in isolation) would lead to a total shrinkage of the corresponding edge. Thus, any spring's length more (resp. less) than its relaxed length generates an attractive (resp. repulsive) force between the connected nodes.

The spring constants E_K are useful in defining the relative importance of each constraint and are chosen to be same for each constraint-type. This is motivated by the fact that a stiff spring (high E_K) holds more energy than a normal spring when both are stretched (or compressed) by the same length. Thus, edges corresponding to more important constraints such as alignment and occlusal contacts may be given a higher E_K than other edges. The actual value of

E_K used in our experiments was an integer in the set $\{0, 1, 5, 10, 15, 20\}$. (An E_K value of 0 on an edge will eliminate the constraint type associated with that edge).

We now provide some details on the forces due to each constraint type.

- Alignment constraint edge: Figure 12 shows examples of alignment constraint edges. The forces move the teeth so that the associated features (incisal edges of anteriors (incisors and canines) and buccal cusps of posteriors) on the teeth align with their closest points on the archform. Note that the anteriors have two endpoints on the incisal edges connected to the archform. Thus, if the tooth experiences any rotation (incisal edge not aligned with archform), the two forces couple to create a torquing effect to rotate the tooth (see the lateral incisor in Figure 12(a)).



(a) (b)

Fig. 12: The alignment constraint edges (shown as black line segments) between teeth and archform (shown green). Shown are (a) an upper arch and (b) a lower arch. The endpoints of the constraints are shown as yellow markers. Note the crowding of teeth among the incisors.

- Interproximal space constraint edge: Figure 12 also shows examples of interproximal space constraint edges. These constraints exists between every pair of adjacent teeth and are responsible for creating the needed space on the archform for rectifying any crowding among anteriors. These forces can be either mesial (attractive, closing excess interproximal space) or distal (repulsive, creating more space).
- Marginal ridge constraint edge: Figure 13 shows an example of marginal ridge constraint edges for some posterior teeth. The forces here are only in the vertical direction and their magnitude is proportional to the difference in the heights of adjacent marginal ridges.

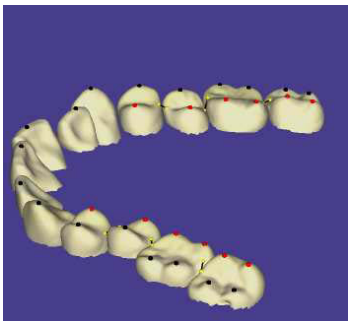
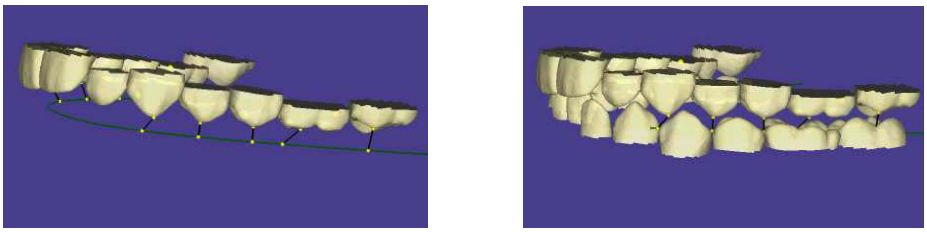


Fig. 13: The marginal ridge constraint edges (shown as black line segments) between the midpoints of the marginal ridges (shown as yellow markers) of adjacent posterior teeth. The force due to these edges depends only on the difference in the vertical height of their two endpoints.

- Buccolingual inclination constraint edge: These edges represent purely rotational torques whose purpose is to bring the lingual and buccal cusps to the same height by rotating the tooth. The torque is generated by applying

a vertically- oriented force at either the lingual or buccal cusps such that magnitude of the force is proportional to the difference in their heights.

- Occlusal relationship constraint edge: An example is shown in Figure 14. Note the position of the upper first molar and canine. The vertical angulation of these edges determines the magnitude and direction of the resulting force; more angulation results in greater force. The same applies for the actual distance between the cusp and the interproximal contact point or buccal groove.



(a) (b)

Fig. 14: Occlusal relationship constraint edges on the teeth of an upper arch. (a) Shown with archform only. (b) Shown in relation to the teeth on the lower arch.

- Occlusal contact constraint edge: Figure 15 shows some of these edges, which are attractive forces bringing the cusps and grooves closer to get tight occlusal contact.

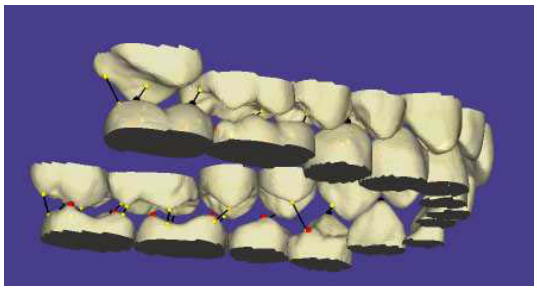


Fig. 15: Occlusal contact constraint edges shown for some of the posterior teeth. Note the edge connecting the cusp of the molar and the corresponding groove on the opposing arch in the posterior teeth.

- Overjet constraint edge: In the anteriors, these edges contribute to attractive forces between the incisal edges of lower incisors and the lingual surface of upper incisors. Also, it forces the anteriors into correct inclination by enforcing the known correct inclination angles for the long axis of anterior crowns. This is to prevent over-inclination of the upper or lower anteriors to satisfy the contact part of the constraint.

5.3 The Simulation Algorithm

The constraint graph is denoted as $G(V,E)$, where V and E are the set of nodes and the set of constraint edges, respectively. Recall that the state of a node N_i is $S_i(t) = (X_i(t), R_i(t), v_i(t), \omega_i(t))$. A *configuration* C_t is defined as a snapshot, $S = S_1, S_2, \ldots, S_{|V|}$, of the states of all the nodes at a time instant, t, of the simulation. A chronological sequence of configurations succinctly records the entire trajectory of all tooth movements and is used for the animation of the occlusion plan.

In order to drive the algorithm to produce better occlusion among nodes (teeth), a measure of quality for a configuration is essential. Such a quality measure depends on how well the constraints are satisfied in the given configuration and is a measure of the correctness of the occlusion achieved. The quality of each constraint type is given a *score* based on the ABO grading system [17]. This score is derived from the measurements made on the features of the teeth and is complete with a set of special cases (e.g., diminutive cusps are not considered for occlusal contact

constraint). The net score of a configuration, C, denoted SCORE(C), is calculated as the sum of the scores of the individual constraints due to all constraint edges in E. The goal is to compute a configuration that minimizes SCORE(·).

There are many boundary conditions imposed on the simulation. Some of these have been discussed earlier, e.g., nodes corresponding to the archform and the occlusal plane are restricted not to move. Also, there may be restrictions on the maximum distance that a node can move in a single time step to avoid abrupt, unnatural movement. There may be different restrictions on the total distance different nodes can move from their original positions, e.g., it may be useful to not move the first molars and canines too much compared to the incisors. Different edge types have different relaxed lengths E_R. These boundary conditions may also evolve over time as the simulation proceeds, e.g., certain teeth are restricted to move together (this is not handled currently by our algorithm and would require merging of nodes in G). Thus, one can appreciate that there are many boundary conditions, a subset of which a user might be interested in applying to a simulation run to study the differences in outcomes produced by initial setups. The simulation algorithm must respect these boundary conditions during its entire execution.

Section 5.2 described how forces are computed from each of the constraint edges. An outline of the main simulation procedure is shown in Algorithm 1. The output of this algorithm is a sequence of configurations leading to an optimal outcome, i.e., the final configuration, C_{final}, minimizes SCORE(·), subject to boundary conditions. Thus, based on our hypothesis (stated at the end of Section 3.1.3), this leads to the best possible occlusion of the teeth with high probability. (More discussion on optimality is provided later in Section 5.4.3.)

Algorithm 1 Constraint-based-simulation

Input: C_{t_0}, t_0 Global input variables: $G(V,E), \Delta t$, MaxIterations, Boundary conditions, δ.

Output: Sequence of configurations, $\boldsymbol{C} = C_{t_0}, C_{t_0+\Delta t}, C_{t_0+2\Delta t}, \ldots, C_{final}$ computed during the simulation.

/* C_{t_0} is the initial configuration and C_{final} is the configuration with optimal occlusion. Δt is the duration of time step by which simulation is advanced in each iteration and δ is an error threshold (used in Algorithm 2). */

```
1:  t = t_0, C_0 = C_{t_0}                                   /* Initialization. */
2:  C = C_0
3:  NumIterations = 0
4:  while (NumIterations ≤ MaxIterations) do
5:      (Ĉ, t̂) = Simulation-step (C_t, t)                    /* Run simulation until it stops. */
6:      Ĉ = Ĉ.end                                            /* Best configuration in C. */
7:      (C', t') = Check-local-minimum (Ĉ, t̂)                /* Overcome the local minimum at Ĉ. */
8:      C' = C'.end                                          /* Best configuration after local minimum check. */
9:      if score(C') ≤ score(Ĉ) then                         /* Lower score indicates better occlusion. */
10:         Append configurations in Ĉ to C                  /* Record the simulation history. */
11:         t = t'
12:         C_t = C'                                         /* Proceed from C_t. */
13:     else
14:         Append configurations in Ĉ to C                  /* Record the simulation history. */
15:         Return C                                         /* Output the best occlusion computed. */
16:     end if
17:     NumIterations = NumIterations + 1
18: end while
19: Return C                                                 /* NumIterations > MaxIterations. */
```

Algorithm 2 computes the next configuration of states such that the energy stored in the constraint edges (which is proportional to their length) is minimized or until no further tooth movement is possible. It is based on the assumption that this minimization leads to better occlusion (see Section 4.2). Also, given that a user can express the relative importance of constraints, the edges with larger spring constants will influence the computation of the next state more than those with smaller spring constants. This naturally resolves conflicts among the forces via a energy minimization approach. The parameter δ is used to check if there was any significant change in any tooth's position or orientation to

continue the simulation further. This is done by checking if the 1-norm of the position vector $X(t)$ and of the rotation matrix $R(t)$ has changed by more than δ between two consecutive iterations.

Algorithm 2 Simulation-step

Input: C_{start}, t_{start} . . Global input variables; same as in Algorithm 1.

Output: Sequence of configurations computed during simulation, $\boldsymbol{C} = C_{start}, C_{start+\Delta t}, C_{start+2\Delta t}, \ldots, C_{end}$ and t_{end}.

```
1:  t = t_start , C_t = C_start                    /* Initialization (see Sections 5.1 and 5.2). */
2:  C = C_start
3:  NumIterations = 0
4:  while (NumIterations ≤ MaxIterations) do
5:      t = t + Δt
6:      C_t = C_{t−Δt}                             /* Create a copy of previous state. */
7:      Get the active constraints in edge set E.
8:      Compute forces and torques on nodes.       /* Section 5.2. */
9:      Update linear and angular velocity.
10:     Compute new states of nodes.               /* Section 5.1. */
11:     Apply boundary conditions to new states.
12:     Record the new states in C_t.
13:     if Corresponding states in C_t and C_{t−Δt} differ by less than δ then
14:         Return C.
15:     else
16:         C = C ∪ C_t                            /* Add C_t to simulation history. */
17:     end if
18:     NumIterations = NumIterations + 1.
19: end while
20: Return C                                       /* NumIterations > MaxIterations */
```

Algorithm 3 Check-local-minimum

Input: C_t, t /* Global variables; same as in Algorithm 1. */

Output: Sequence of configurations computed during simulation $\boldsymbol{C} = C_{start}, C_{start+\Delta t}, C_{start+2\Delta t}, \ldots, C_{end}$ and t_{end}.

/* The positions of nodes in Ct are randomly displaced within the boundary condition restrictions (see Section 5.3.1) and the simulation is run. The best scoring sequence of configurations among such runs is returned. */

```
1:  BestScore = score (C_t).
2:  C = C_t
3:  t_end = t
4:  for i = 1 to k do
5:      Create C^i by displacing nodes from their positions in C_t.
6:      (Ĉ, t̂) = Simulation-step (C^(i), t)        /* Run the simulation. */
7:      C_end = C.end
8:      S = score( C_end).
9:      if BestScore ≤ S then                      /* If a better configuration is found. */
10:         C = Ĉ, t_end = t̂, BestScore = S
11:     end if
12: end for
13: Return C, t_end /* Best among the k runs. */
```

5.3.1 *Local minimum configurations in simulation*

Any optimization method must completely avoid, or detect and rectify, the situation when the search is stuck in a local minimum configuration. A *local minimum configuration* is reached when no further node movement (translation or rotation) is feasible (possibly due to collision constraints), but the total energy in the edges can be reduced. In other words, the teeth cannot move further, but a configuration with better occlusion exists. Algorithm 3 shows the algorithm used to detect and handle a local minimum configuration in order to continue the search for a global minimum configuration.

Given a configuration in which teeth are interlocked and cannot move further, Algorithm 3 introduces random perturbations to the tooth positions (particularly in the z-direction), and restarts the simulation. This is done a constant number of times (specified as k in Algorithm 3, line 4) and the best resulting configuration (the one with least score) is returned. If the initial configuration was truly a local minimum, then, with high probability, some set of perturbations will lead to a better configuration. However, if the configuration was indeed a near-optimal one, then no further improvement would likely be possible and the initial configuration is returned as the result. Note that the boundary conditions on tooth movements must be respected while creating the perturbations.

5.4 Characteristics of the Simulation Algorithm

The simulation algorithm differs from the traditional rigid body simulation methods in many ways. First, the collision detection among nodes is not handled uniformly for simplicity and efficiency. For example, the collision between a pair of nodes on the same arch, the opposing arches, and a tooth node and archform node are all handled in completely different ways (see Sections 5.1.1 and 5.4.2). Also, the exact values of mass, spring constants, termination conditions are all chosen experimentally to make the simulation work. This section describes some of the characteristics of the algorithm.

5.4.1 *Duration of time step*

A key factor that controls the accuracy and performance of Algorithm 2 is the magnitude of each time step, Δt, which controls the rate of advancement of the simulation (see Equation 4). Large values of Δt result in large changes in the translation and rotation of the nodes. If these changes are too large, they may miss the optimal solution because the search is very “coarse”. However, if Δt is too small, it may take a long time (many iterations) to converge and result in high computational cost. Most real-world simulations have a similar Δt parameter which is often best found experimentally, as we do.

5.4.2 *Residual velocities and cycling through forces*

Another major issue with the simulation, as described, is that of spurious movement of nodes due to residual velocities. As an illustration, consider a tooth node in Figure 12. Suppose that the only constraint acting on the tooth node is the alignment constraint forcing it to align along the archform. As the tooth node starts moving towards the archform, it builds up a certain momentum (and velocity). Now, consider the time when the tooth is aligned with the archform. The alignment constraint is satisfied and, hence, there should be no further tooth movement. However, the collision between a tooth node and the archform node cannot be handled in the same way as collision of two teeth (the archform is a curve and could be on a different plane than the tooth, i.e., the corresponding nodes “collide” only when viewed from top). Thus, even though the alignment constraint is satisfied, there is some amount of residual velocity in the tooth node that moves it past the target position, eventually leading to an oscillatory movement of the tooth about the archform. It is also clear that as one adds more constraints, such behavior will only multiply and get more complicated.

A simple solution to the above problem, without doing any sophisticated collision detection, is to simply reset the linear and angular velocities to zero after a certain number of iterations. This technique, while not affecting the long-term position and orientation of the nodes during simulation, suffices to handle the linear and angular momentum that is built up due to large forces on constraint edges.

In the case of multiple constraints, in addition to the reset of velocities, the simulation also selects a constraint-type that drives the computation of forces and torques on nodes for a small number of iterations. Thus, the entire time interval is divided into very small subintervals during each of which only a single constraint determines the next configuration. Thus, the effect of multiple constraints acting more or less simultaneously over the time interval is approximated by the cumulative effect of individual constraints acting independently over successive subintervals of

very small length. This simplifies the simulation considerably while still ensuring that the effect of no single constraint skews the results unfairly.

5.4.3 Optimality and convergence

Section 4.3 has already presented several arguments for why the configurations produced by the simulation approach are close to the optimal occlusion in most cases. Some discussion was also given why this is a reasonable goal given that the optimality of a configuration cannot be verified due to the lack of metrics to evaluate the quality of a given configuration of teeth in a pair of opposing arches.

Also, given the nature of the problem, the convergence of Algorithm 2, i.e., the termination of the simulation upon reaching an optimal configuration, is difficult to prove. To see this, consider a pair of equal and conflicting forces at a very small scale, e.g., interproximal space and occlusal relationship constraints acting on a second molar and influencing its placement along the archform. In such a case, it is possible that in the final stages, the simulation simply moves back and forth between two similarly-scored states, as the teeth move in an oscillatory fashion. This motivates the use of a threshold on the amount of tooth movement to be considered significant and an upper bound on the number of maximum iterations that the simulation can run (which is determined experimentally).

However, our experiments consistently show that the occlusion achieved by the simulation process is quite good and often converges quickly to near-optimal configurations. This is because the chosen constraints (see Section 5.2) guide the search for correct occlusion very efficiently.

6 EVALUATION

This section describes the software tool which implements the simulation algorithm described in Section 5.3 and presents the results of the simulation on a set of real-world test cases.

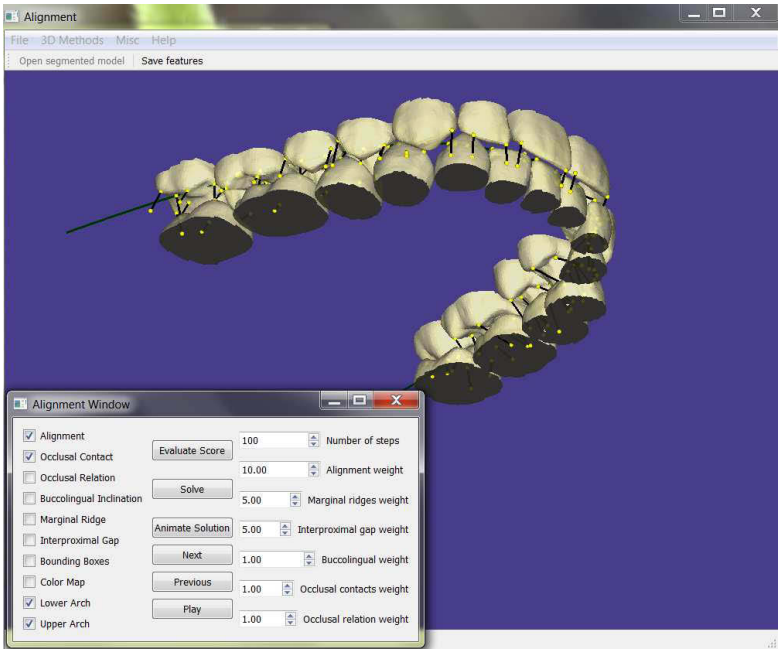


Fig. 16: Screenshot of the feature evaluation and alignment tool. An upper and a lower arch is shown along with some of the constraint edges (forces) that have been activated for the simulation. The checkboxes on the lower-left window are used to activate or deactivate different constraint edges. The numeric fields on the right side of this window can be used to set the weight of the edges of each constraint-type and the maximum number of iterations in the simulation process. The buttons in the middle are used to start the simulation, view the animation of tooth movement computed during the simulation, and evaluate a score for the quality of the occlusion achieved.

6.1 Setup

We have developed a software tool for the simulation of the alignment process using Qt Open-Source Edition 4.5 in C++. It consists of a simple, intuitive user interface that can be used to load dental models and carry out the simulation of alignment. Figure 16 shows a screenshot of the tool. In addition to this, for the purpose of evaluating the outcome of the simulation algorithm, we also use a proprietary software tool [10] that determines the discrepancies in the tooth positions and orientations in given pair of arches.

6.2 Results and Analysis

The simulation algorithm described in Section 5.3 was evaluated on a set of 8 lower and 8 upper arches. These pre-segmented arches were selected from a database of actual cases treated at the University of Minnesota's dental clinic. The models were obtained through the SureSmile® software [18] and cover a range of difficulty, including malocclusions, improperly formed teeth, and varying anatomy on posteriors. For each of these cases that we studied, the arrangement of teeth from both before and after the alignment planning phase (the latter determined by an expert) are given. (These are known as the pre-treatment and post-treatment configurations, respectively.)

The process of computing an alignment to improve occlusion is carried out by using the alignment module of our software tool described above. Note that some of the simulation parameters such as the timestep (Δt) and the error threshold (δ) are set to a value that performs well for all types of cases, as predetermined experimentally. Also, it was observed that a broad range of weight assignments for the constraint edges (representing their spring constants) performed well on a majority of the models. (An example of such a weight assignment is to set the weight of AC edges to 10, the weights of MRC, ISC, and OCC edges to 5, and all the remaining edge weights to 1.)

As mentioned in Section 1.1, there are very few well-established metrics to evaluate the quality of occlusion for a given dental model. In practice, an expert often relies on visual inspection for a qualitative analysis. However, there are some quantitative methods to grade the quality of occlusion by making certain measurements on surface features of the teeth as described in Section 3.2, and checking for commonly occurring orthodontic errors. Our implementation is based on the techniques mentioned in Section 3.2, and is used to evaluate a score for each dental model (see Section 6.2.1). Also, the availability of the post-treatment arrangement of the same teeth enables us to measure the discrepancies in the tooth positions computed by the simulation and the positions established by an expert (see Section 6.2.2).

6.2.1 Score-based analysis

Each active constraint is evaluated on the current position and orientation of the involved teeth. If a constraint is not satisfied, an integer score in the range [0,2] is assigned to it. For example, if the height of the adjacent marginal ridges differ by less than 0.5 mm, no score is assigned. However, if this difference is between 0.5 mm and 1.0 mm (resp. 1.0 mm and 2.0 mm), a score of 1 (resp. 2) is assigned. Thus, this provides a coarse grading of the quality of occlusion in terms of the error in constraints. Thus, a similar score is assigned when an incisal edge or a buccal cusp is out of alignment from the neighboring teeth (AC), and so on. More details on these score values can be found in [17]. The total of all the scores of all the constraint edges defined on the arches is taken as the total score.

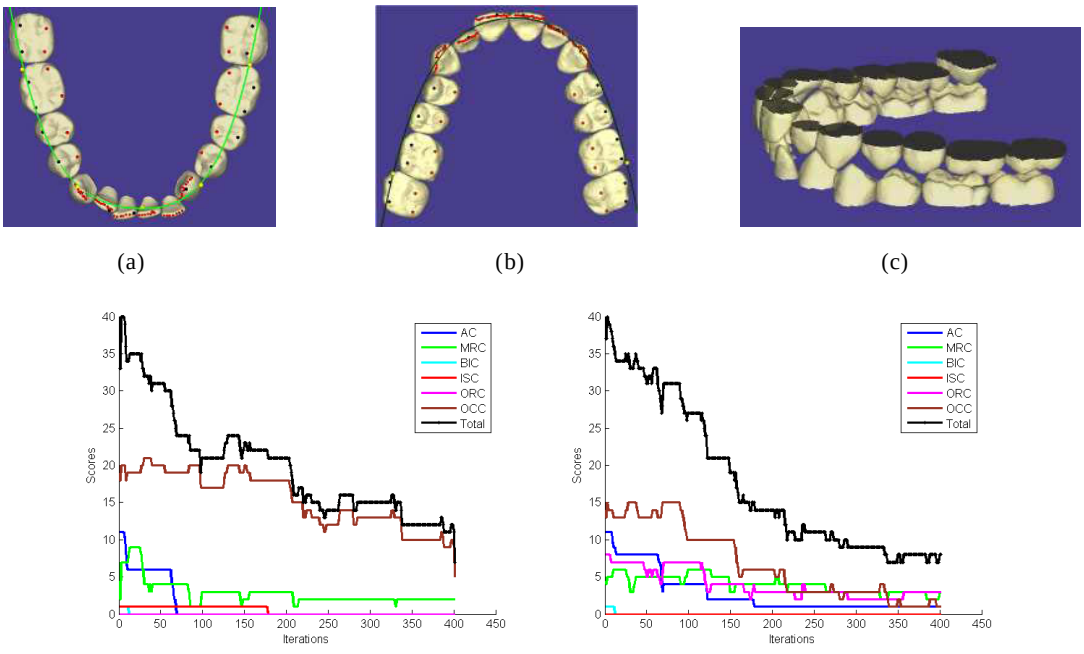
As mentioned previously, a low score for a model indicates better quality of occlusion. Figures 17-20 show some examples of models with frequently occurring errors. Also given are graphs that show the score due to each constraint and the total score. These graphs reveal how changes in the score of a constraint affect the other scores.

Figure 17 shows a normal orthodontic case with a small amount of crowding at the anteriors, whereupon the simulation process brings about a gradual and steady drop in the scores of all the constraint types. Figure 18 shows an example with too much space between the anterior teeth of the upper and lower arch. This is difficult to handle because of the conflict between the ISC and ORC. As a result, even though the teeth can be taken to their corresponding archform very quickly (AC is satisfied within a few rounds), the ORC and OCC scores are significantly affected.

Also, the lower second molars in Figure 18 have very little crown area available, especially on the lingual side. This, in turn, makes it difficult to identify the cusps and use them in determining the correct occlusal contacts and their relationships. Such problems with second molars are in general difficult to handle automatically because of the wide

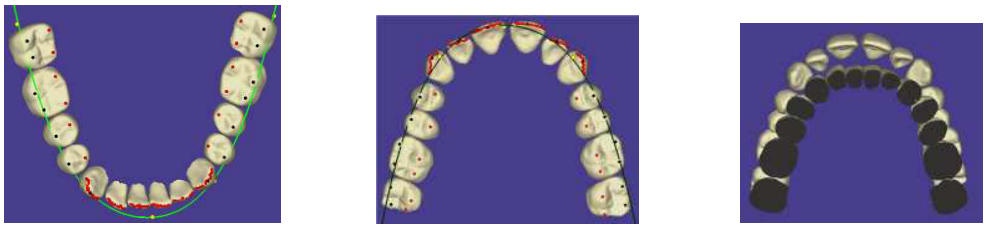
variety of the errors that occur. Thus, this is a situation where a user's experience will be needed to establish an occlusion that is acceptable.

An example where an upper arch anterior tooth is "trapped" in the lingual region of the lower anterior teeth is shown in Figure 19 (note the left lateral incisor on the upper arch). This situation may also happen in the posterior region where a tooth cannot move because of other teeth in its neighborhood. Such problems can be resolved in an efficient fashion by ensuring that there is enough room for the teeth to move horizontally, without collisions with the opposing arch teeth, at least for the initial few steps of the simulation. This usually suffices because the AC (in general, the intra-arch constraint) edges have large spring constants associated with them, and are able to play a more significant role especially in the initial iterations of the simulation. This scope for horizontal movement is provided by moving each tooth vertically away from the other arch so that there is a horizontal plane separating the two arches. This initial abrupt movement of teeth is responsible for the sudden increase in the scores in the first two or three iterations. Finally, Figure 20 shows a case where there is sufficient crowding on the archform on the upper anteriors.

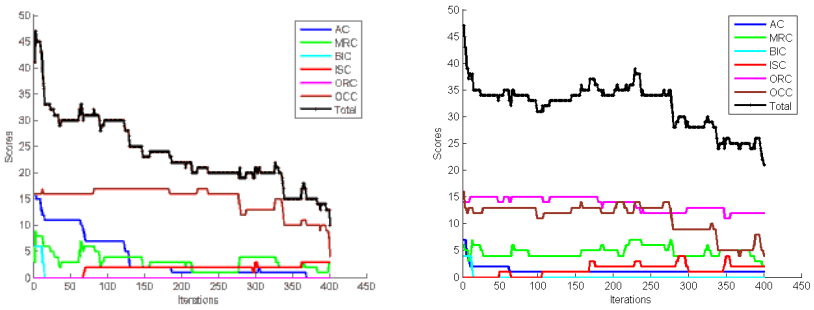


(a) (b) (c) (d) (e)

Fig. 17: The simulation of teeth under the influence of constraint-based forces. The lower arch (a), upper arch (b) and their combined occlusion from the buccal side (c) is shown. The simulation history and the scores due to different constraints are shown for the lower arch (d) and the upper arch (e). The black line at the top represents the total score due to all the constraints. The *x*-axis in the graph represent the iteration number during the simulation.

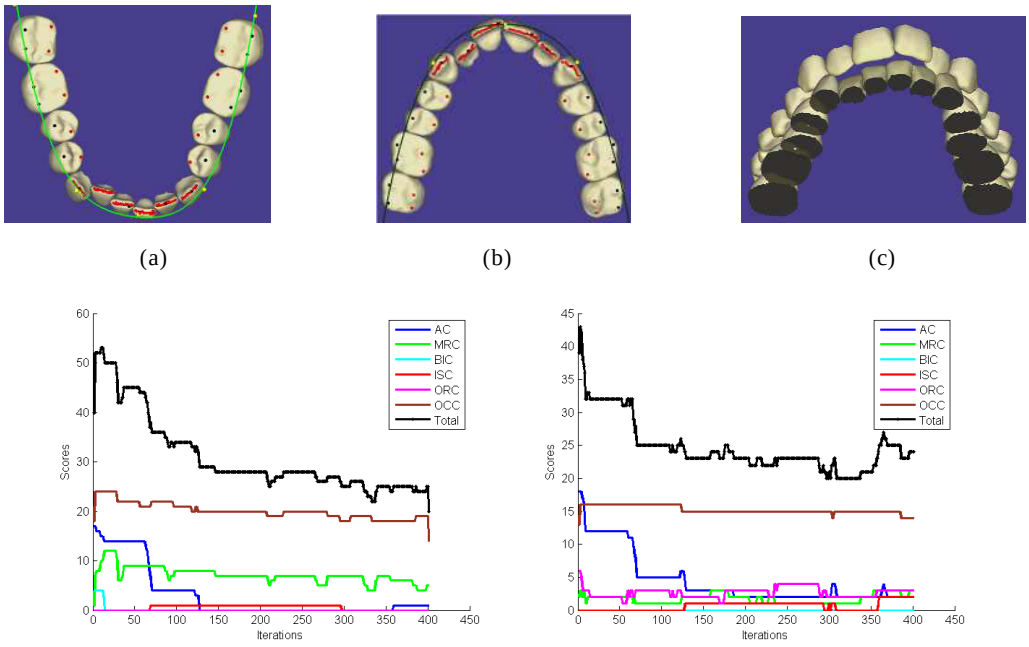


(a) (b) (c)



(d) (e)

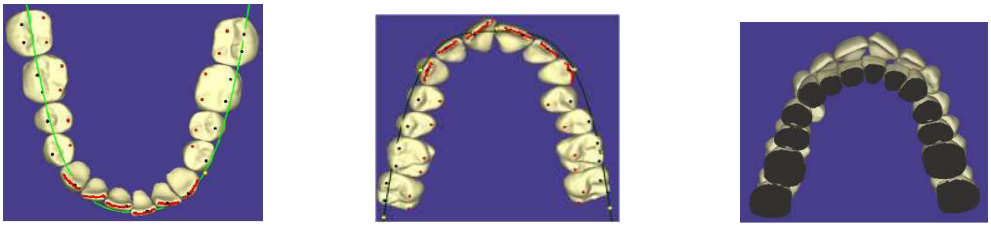
Fig. 18: The simulation of teeth under the influence of constraint- based forces. (a) Lower Arch. (b) Upper Arch. (c) Wide space between the lower and the upper anteriors. Simulation history of the lower arch (d), and the upper arch (e). (See the caption of Figure 17 for a detailed explanation.)



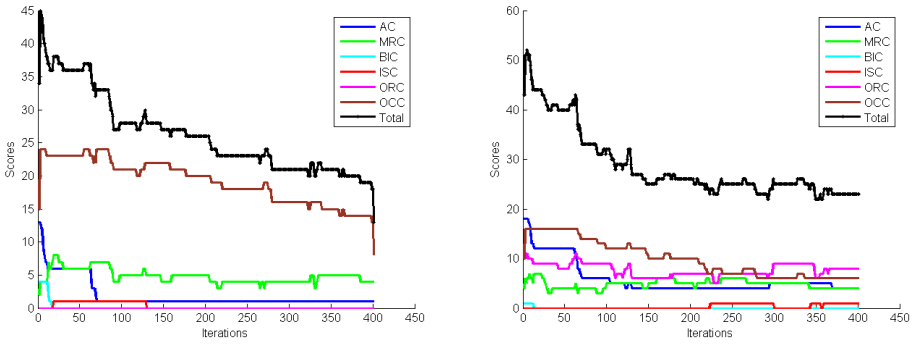
(a) (b) (c)

(d) (e)

Fig. 19: The simulation of teeth under the influence of constraint- based forces. (a) Lower Arch. (b) Upper Arch. (c) The left lateral incisor in upper arch is trapped among the lower anteriors. Simulation history of the lower arch (d), and the upper arch (e). (See the caption of Figure 17 for a detailed explanation.)



(a) (b) (c)



(d) (e)

Fig. 20: The simulation of teeth under the influence of constraint-based forces. (a) Lower Arch. (b) Upper Arch. (c) Crowding among the anteriors. Simulation history of the lower arch (d), and the upper arch (e). (See the caption of Figure 17 for a detailed explanation.)

6.2.2 *Surface-matching-based analysis*

Next, we describe a surface-matching-based analysis of the final occlusion established by the simulation algorithm. As mentioned earlier, this is done by means of the *e*model compare 8.1 software that takes as input two arrangements of the same set of teeth, one computed by the simulation and the other determined by an expert, and determines the discrepancies in the position of each tooth in one arrangement relative to the corresponding tooth in the other arrangement. This is achieved by finding a close match, which minimizes the total separation of the vertices of the surfaces in question. The surface-based method has advantages and disadvantages when compared to the score-based method discussed above. The score-based method is somewhat coarse as it assigns real-valued measurements to integer-valued "buckets", whereas the surface-based method is more sensitive small differences in the tooth positions and to factors such as the density of meshes. However, the surface-based method may provide too much detail in comparing the quality of occlusion even when it suffices to establish correct occlusion globally rather than on a tooth-by-tooth basis. Finally, the score-based method does not rely on the availability of a treatment plan created by an expert.

The errors are reported in terms of the translations and rotations needed to align each tooth with its image in the other arrangement. However, often the two arrangements are not in the same frame of reference and, due to this, a global registration of the two arrangements must be established to offset any global discrepancies. For example, consider two adjacent posterior teeth whose marginal ridges are not at the same height. In order to satisfy this constraint, the higher-placed tooth can be moved vertically down, or the lower-placed tooth can be moved up. Even though the net effect on the occlusion in both cases would be the same, the two arrangements produced by the alternative moves will be off vertically from each other, when considered in a global frame of reference.

For each of the cases, we compare the arrangement resulting from the simulation to the post-treatment arrangement established by an expert. First, the two arrangements are registered globally, after which individual teeth are compared. Tables 1 and 2 present the results on teeth in 8 upper and 8 lower arches, respectively. The mean values of their error are shown along with their standard deviation.

The discrepancies are computed based on a coordinate-system located at each tooth parallel to the archform. This enables us to assign meaningful descriptions to the differences in the tooth positions and the orientations about different axes found by the surface matching algorithm. Thus, the errors are presented relative to an orthogonal coordinate system which has its axes parallel to the mesiodistal line, buccolingual line, and the vertical z-axis. The corresponding errors in orientations are taken as the error in tooth inclination (torque), tooth angulation (tip), and tooth rotation (spin about the z-axis).

The low order errors in the mesial-distal and buccal-lingual columns indicate that the simulation does align the teeth to the archform as expected of the strong intra-arch forces. The correct vertical positioning of the teeth is little more challenging because of the conflicts among MRC and OCC. On the other hand, the errors are relatively higher in

terms of the torque, tip and rotation angles. We note that there no significant difference in the discrepancies between the upper and the lower arch teeth. Also, no significant differences were observed between teeth from the left and right side of the arches.

Note that discrepancy values with a large variance were observed for the tooth angulation comparison (shown in columns labeled as "Tip" in Tables 1 and 2). This is due to both the lack of reliable tooth features to describe the correct crown angulation and the lack of constraints to enforce the correct crown angulation in the simulation process.

Tooth type	Mesial-Distal (mm)	Buccal-Lingual (mm)	Vertical (mm)	Torque (°)	Tip (°)	Rotate (°)
2nd molar	-0.05 ± 1.94	-0.88 ± 1.41	1.25 ± 0.85	-9.53 ± 10.11	-2.45 ± 10.38	-2.22 ± 4.97
1st molar	-0.06 ± 1.28	-0.44 ± 0.61	0.57 ± 0.78	-2.62 ± 4.64	-2.07 ± 4.47	-0.97 ± 3.58
2nd premolar	-0.39 ± 2.68	-0.06 ± 1.02	1.28 ± 3.07	-2.60 ± 4.69	-11.66 ± 30.3	2.15 ± 12.36
1st premolar	-0.03 ± 1.31	0.18 ± 0.75	1.91 ± 0.94	2.18 ± 4.75	-0.31 ± 6.01	-4.93 ± 7.21
Canine	-0.66 ± 1.19	0.09 ± 0.79	1.00 ± 1.38	2.60 ± 4.41	-9.55 ± 18.9	-5.08 ± 11.21
Lateral incisor	-0.11 ± 1.46	1.15 ± 0.99	0.25 ± 1.90	3.99 ± 5.92	-7.62 ± 20.9	-7.53 ± 25.42
Central incisor	0.09 ± 0.89	1.26 ± 0.75	0.08 ± 0.75	7.13 ± 3.49	-0.24 ± 7.28	0.48 ± 10.20

Tab. 1: Discrepancies in position and orientation of the upper arch teeth across all the 8 cases (16 teeth of each type), in the outcome of the simulation compared to the arrangement established by an expert. The results are shown as mean values ± standard deviations. Negative values indicate a tooth position more distal, lingual or vertical, or with less buccolingual inclination, less mesial tip or buccal surface rotated more distally than the correct tooth in the post-treatment arrangement.

Tooth type	Mesial-Distal (mm)	Buccal-Lingual (mm)	Vertical (mm)	Torque (°)	Tip (°)	Rotate (°)
2nd molar	-0.16 ± 1.13	0.36 ± 0.72	0.71 ± 1.09	-1.67 ± 5.85	-1.57 ± 5.03	11.90 ± 6.02
1st molar	-0.33 ± 0.78	0.36 ± 0.52	-0.15 ± 1.01	-2.70 ± 3.74	-0.38 ± 2.52	11.15 ± 4.29
2nd premolar	-0.40 ± 1.09	0.08 ± 0.75	0.43 ± 1.21	-0.16 ± 4.34	4.48 ± 9.31	4.96 ± 10.72
1st premolar	0.06 ± 1.21	0.19 ± 0.95	-0.03 ± 1.10	1.37 ± 5.92	3.72 ± 6.17	-7.27 ± 10.65
Canine	0.15 ± 1.17	-0.86 ± 0.80	0.07 ± 1.66	2.09 ± 4.51	0.73 ± 7.83	3.13 ± 9.45
Lateral incisor	-0.12 ± 1.12	0.34 ± 1.04	0.78 ± 1.51	-0.73 ± 7.01	6.15 ± 5.61	10.45 ± 16.28
Central incisor	-0.05 ± 0.81	0.14 ± 1.33	0.61 ± 1.19	-3.40 ± 4.72	4.01 ± 6.56	4.20 ± 5.81

Tab. 2: Discrepancies in position and orientation of the lower arch teeth across all the 8 cases. (See the caption of Table 1 for a detailed explanation.)

7 CONCLUSIONS AND FUTURE WORK

We have presented a technique to establish dental occlusion via a simulation-based approach that models the alignment process as a spring-mass system where teeth are considered to be masses connected by springs representing dental constraints. These constraints are obtained from the experience of practitioners and well-known approaches to objectively classify commonly occurring orthodontic errors. A set of constraints guide the simulation algorithm to find an arrangement that best satisfies the given constraints subject to certain boundary conditions. The simulation algorithm can be extended to include more real-world scenarios, e.g., missing teeth, Class II and Class III relationships, etc. As mentioned earlier, there are no established metrics to evaluate the quality of occlusion in a given dental model

and this presents significant challenges in designing an algorithm to automatically find the arrangement with the optimal occlusion. The simulation can be made more biologically realistic by using the tooth's root information from an alternative source, such as a CT-scan, instead of relying on only tooth crown information from a laser-scan. Also, the simulation process itself will be greatly enhanced through the use of modeling techniques that take into account the physical properties of the jaw bone and the teeth.

Acknowledgment

This research was supported, in part, by the Orthodontics Education and Research Fund at the University of Minnesota. We thank Mike Marshall for help with the evaluation phase of this work.

REFERENCES

[1] Guide for setting up teeth on F/F dentures. http://tinyurl.com/83p45dx.
[2] Aarseth, S. J.: *Gravitational N-Body Simulations: Tools and Algorithms*, Cambridge University Press, 2003.
[3] Andrews, L. F.: The six keys to normal occlusion, *Am. J. Orthod.*, 62(3), 296–309, 1972.
[4] Angle, E. H.: *Treatment of malocculsion of the teeth*, University of Michigan Library, 1907.
[5] Baraff, D.: Physically based modeling, SIGGRAPH course notes, 2001.
[6] Chang, Y.; Xia, J. J.; Gateno, J.; Xiong, Z.; Zhou, X.; Wong, S. T. C.: An automatic and robust algorithm of reestablishment of digital dental occlusion, *IEEE Trans. on Medical Imaging*, 29(9), 1652–1663, 2010.
[7] Eismann, D.: Reliable assessment of morphological changes resulting from orthodontic treatment, *Europ. J. Orthod.*, 2, 19–25, 1980.
[8] Gottlieb, E.: Grading your orthodontic treatment results, *J. Clin. Orthod.*, 9:156–161, 1975.
[9] Hiew, L. T.; Ong, S. H.; Foong, K. W. C.: Optimal occlusion of teeth using planar structure information, *Machine Vision and Applications*, 21(5), 735–747, 2010.
[10] Hultgren, B.; Isaacson, R.; Marshall, M.: Digital Model Comparison software (*e*model compare 8.1), 2011.
[11] Jimĭnez, P.; Thomas, F.: 3D collision detection: A survey, *Comp. & Graph.*, 25, 269–285, 2001.
[12] Kronfeld, T.; Brunner, D.; Brunnett, G.: Snake-based segmentation of teeth from virtual dental casts, *Comp.-Aided Design. and Appl.*, 7(2), 221–233, 2010.
[13] Kumar, Y.; Janardan, R.; Larson, B.: Automatic feature identification in dental meshes, *Comp.-Aided Design. and Appl.*, 9(6), 747-769, 2012.
[14] Kumar, Y.; Janardan, R.; Larson, B.; Moon, J.: Improved segmentation of teeth in dental models, *Comp.-Aided Design. and Appl.*, 8(2), 211–224, 2011.
[15] Logan, D. L.: *A First Course in the Finite Element Method*. Fifth Edition, Cengage Learning, 2012.
[16] Ming, C. L.; Gottschalk, S.: Collision detection between geometric models: A survey. *In Proc. of IMA Conf. on Math. of Surfaces*, 1, 1–20, 1998.
[17] American Board of Orthodontics. Grading system for dental casts and panoramic radiographs, 2010. http://tinyurl.com/ygstyfu.
[18] SureSmile®. http://www.suresmile.com/.
[19] *e*model®8.5. http://www.geodigmcorp.com.
[20] Woelfel, J. B.; Scheid, R. C.: *Dental Anatomy: Its Relevance to Dentistry*. Lippincott Williams & Wilkins, 2011.