Journal of Zhejiang University-SCIENCE C (Computers & Electronics)
ISSN 1869-1951 (Print); ISSN 1869-196X (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Extracting 3D model feature lines based on conditional random fields*

Yao-ye ZHANG†, Zheng-xing SUN†‡, Kai LIU, Mo-fei SONG, Fei-qian ZHANG

(*State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210046, China*)

†E-mail: flavor2@163.com; szx@nju.edu.cn

Received Dec. 29, 2012; Revision accepted May 21, 2013; Crosschecked June 6, 2013

Abstract: We propose a 3D model feature line extraction method using templates for guidance. The 3D model is first projected into a depth map, and a set of candidate feature points are extracted. Then, a conditional random fields (CRF) model is established to match the sketch points and the candidate feature points. Using sketch strokes, the candidate feature points can then be connected to obtain the feature lines, and using a CRF-matching model, the 2D image shape similarity features and 3D model geometric features can be effectively integrated. Finally, a relational metric based on shape and topological similarity is proposed to evaluate the matching results, and an iterative matching process is applied to obtain the globally optimized model feature lines. Experimental results showed that the proposed method can extract sound 3D model feature lines which correspond to the initial sketch template.

Key words: Nonphotorealistic rendering, Model feature lines, Conditional random fields, Feature line metrics, Iterative matching
doi:10.1631/jzus.CIDE1308 **Document code:** A **CLC number:** TP391

1 Introduction

Line drawing of 3D models is one of the most interesting issues in nonphotorealistic rendering (NPR) fields (Hertzmann, 2010), and is widely used in interactive modeling, model analysis, and other applications (Catalano *et al*., 2011). A key problem in line drawings is how to extract feature lines that give the shape and structure information needed for 3D models, a procedure similar to that implemented by artists. There have been many feature line extraction methods based mainly on the geometric features of 3D models, in which different feature lines including silhouettes (Hertzmann, 1999), suggestion contours (DeCarlo *et al*., 2003), and ridges and valleys (Interrante *et al*., 1995) are extracted and then combined to depict the shape of 3D models. These methods extract feature lines from 3D models automatically, but without user guidance, the appearance, number, and structure of feature lines may not be consistent with user perception and different requirements of various applications (Cole *et al*., 2008). As a result, the question of how to combine user guidance with geometric features, thereby making the extracted feature lines consistent with user perception, remains an important issue in line drawing (Hertzmann, 2010).

Some researchers have attempted to make feature line extraction somewhat interactive so that user guidance information can be introduced. For instance, Cole *et al*. (2008) fitted regression models of feature curve locations to a sample set of hand-drawn images; a large sample set was needed, however, due to the focus on feature curve location statistics. Kalogerakis *et al*. (2012) adopted a model based on artists' hatching methods, which can be used to render a new hatched image of an object, but this model can be

‡ Corresponding author

* Project supported by the National Natural Science Foundation of China (Nos. 61272219, 61100110, and 61021062), the National High-Tech R&D Program (863) of China (No. 2007AA01Z334), the Program for New Century Excellent Talents in University (No. NCET-04-04605), and the Science and Technology Program of Jiangsu Province, China (Nos. BE2010072, BE2011058, and BY2012190)

used only for creating hatching maps. Each of the methods above requires a large number of models be labeled for study and involves extraction processes which are difficult to manipulate. Lum and Ma (2005) implemented an interactive feature line extraction process using the user's sketch strokes. However, it depends upon existing object-space based feature line extraction methods, which are sensitive to the thresholds of key parameters and mesh quality. What is more, the sketch is used only for labeling, and features such as shape, position, and topology are ignored, so the native properties of the sketch cannot be reflected.

Sketching is one of the most intuitive and naturally interactive tools, and has been widely used in 3D modeling (Olsen *et al*., 2009), information retrieval, and other fields. One of the most effective methods of extracting model feature lines is using sketching, so that the line drawings extracted correspond to specified sketch strokes. Sketching can effectively convey position, shape, and topological features according to users' purposes, and can also be combined with model geometric features. The key problem of sketch-guided extraction is how to find a subset of model feature points that best fit the sketch strokes, from which the feature points are connected to form continuous feature lines. The difficulty of matching feature lines with sketch strokes lies in making the extracted feature lines not only depict the shape of the model surface, but also fit the sketch strokes. Matching methods that combine sketch strokes with model feature points are usually based on simple distance differentials (Mao *et al*., 2009) or a probabilistic model of specific features (Kraevoy *et al*., 2009), but existing methods always follow the 'extract-match' process so that the matching results are limited by the feature lines extracted and are also sensitive to the thresholds. Furthermore, these conventional methods consider only specific features of sketch points and model feature points, and usually focus on the correlation between single strokes and feature lines instead of global topological features. As a result, only simple sketch strokes can be processed, and more complex sketches are difficult to match (Kraevoy *et al*., 2009).

While acknowledging the problems listed above, this paper proposes a 3D model feature line extraction method guided by the user's sketch. This method has two main advantages: (1) A conditional random fields (CRF) matching model with multiple features between stroke points and candidate feature points is established, and the user template is effectively combined with model feature line extraction; (2) A metric based on topological similarity is proposed to evaluate the rationality of extracted feature lines, and an iterative optimization process is applied to obtain globally optimized model feature lines.

2 Related studies

Conventional feature line extraction methods can be divided into object space methods and image space methods. Object space methods extract feature points by computing curvatures and other differential properties of the model surfaces. Hertzmann (1999) extracted model contours with normal vectors between each vertex of the model surface, and then computed the cross product of normal vectors and view vectors to select contour points. DeCarlo *et al.* (2003) proposed suggestive contours as the extension of contours. Interrante *et al*. (1995) proposed a method to detect ridges and valleys via local curvature extremes of 3D surfaces. Judd *et al*. (2007) defined view-dependent curvature as the variation of the surface normal with respect to a viewing screen plane, and apparent ridges as the loci of points that maximize a view-dependent curvature. These existing methods directly extract feature lines of various types based on geometric features. The resultant extracted line drawings are significantly influenced both by the thresholds used, and by the mesh quality; if the mesh quality is poor, a mesh preprocessing is usually needed. Image space methods, on the other hand, extract feature lines from projected images of 3D models and do not depend on object space estimations of differential attributes. Saito and Takahashi (1990) extracted contours, creases, and other feature lines using image processing operations, while Lee *et al*. (2007) demonstrated near interactive animations of line drawings using GPU-based image processing operations. Image space methods are appealing in that they are generally simple to implement and that they show how level-of-detail abstraction occurs automatically in image space computation. However, there are some drawbacks: accuracy is limited by pixel resolution (often resulting in jagged or irregular lines), stylization options are limited (e.g., curves cannot be textured), speed is limited by hardware image-processing performance, and a careful setting of user-defined thresholds is required (Kalogerakis *et al*., 2009). Some researchers have attempted to use guidance in model feature line extraction. Lum and Ma (2005) proposed an interactive feature line extraction process using user's sketch strokes, and used neural networks and support vector machines to learn a classifier for displaying feature curves. This method is sensitive to key parameters' thresholds and mesh quality; the sketch is used only for labeling and other features of sketch such as shape and position are abandoned. Cole *et al*. (2008) learned regression models of feature curve locations from a sample set of hand-drawn images, but only the probabilities of feature curve locations are computed and a large sample set is needed. Kalogerakis *et al*. (2012) established a mapping between model features and hatching style, and new hatching maps from different views can be rendered according to this style. But this procedure can be used only for rendering a hatching map. All these methods require a large number of sample models be labeled, which makes the extraction process more difficult to manipulate.

Some researchers have studied the problem of matching between sketch strokes and model feature lines. Mao *et al*. (2009) used simple distance differences to match strokes and model feature points, but their method cannot be applied to complex sketches. Kraevoy *et al*. (2009) constructed a hidden Markov model (HMM) using stroke points and model feature points, to match the strokes and contours based on shape similarity, but the HMM model allows only local shape features to evolve.

Furthermore, existing matching methods always divide the process into two steps: feature line extraction and stroke lines matching. Feature lines are extracted using conventional line drawing methods, and thus the matching results are greatly constrained; at the same time, only local shape differences are considered, and a lack of global shape and topology features makes complex sketches difficult to deal with.

Conditional random fields (CRF), first proposed for sequence data analysis by Lafferty *et al*. (2001), has been widely used in natural language processing and image registration in recent years. Ramos *et al*. (2007) proposed a CRF-matching model based on CRFs in robotics laser scan matching, to establish pixel mapping between two different images. Arbitrary features were used to delineate various properties of the data, and these features can be taken from labeled data to combine the local and spatial properties of the data in a theoretically sound manner. This, in contrast to generative models such as HMM and Markov random fields, provides substantial flexibility in using high-dimensional feature vectors.

3 Conditional random fields based feature line extraction

Our proposed method allows users to extract model feature lines with a template. The main process of this method, as shown in Fig. 1, is composed of three steps. First, as a preprocessing, the given 3D model is projected onto a depth map, and a Canny edge detection process is applied to obtain the edge points, allowing a subset of edge points to be extracted as candidate feature points to follow the sketch strokes. Second, several feature functions are defined to establish a CRF model using the sketch and model feature points, including two types of feature functions, shape similarity feature functions and gradient feature functions. The marginal probability distributions of hidden states can be evaluated with belief propagation (BP), and then a Viterbi decoding is

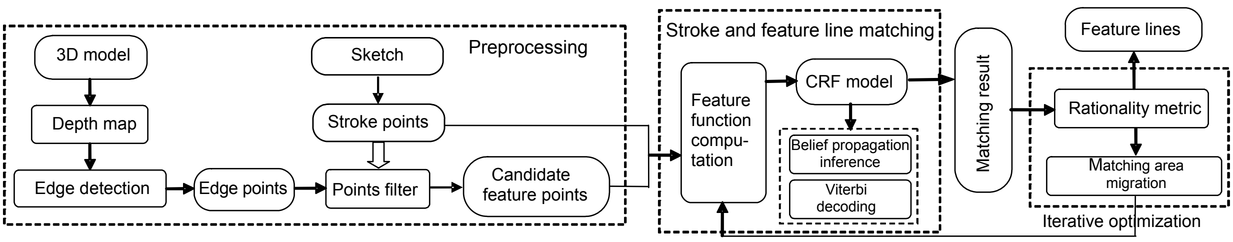

Fig. 1 The main process of our proposed conditional random fields (CRF) based feature line extraction

applied to obtain the sequential candidate feature points as feature lines. Based on shape and topology similarity, we define a rationality metric of feature lines, according to which the candidate area can be migrated to better fit the model. An iterative matching is then applied to progressively approximate the globally optimized feature lines.

3.1 Preprocessing

The feature points of 3D models usually convey the salient features on model surfaces, and are potential points for the composition of feature lines. There are mainly two kinds of conventional feature line extraction methods, object space methods and image space methods. Object space methods extract feature points by computing the differential properties of model surfaces, and usually have a higher computational complexity than image space methods. Furthermore, there are various geometric features that correspond with the different kinds of feature lines, including silhouettes, highlights, suggestive contours, valleys and ridges, making the extraction procedure more complicated. In image space methods, 3D models are projected into 2D images to extract feature points by using image processing algorithms, which greatly reduces the computational complexity, while a single threshold is required to make parameter adjustment more facile. In our proposed method, the 3D model is projected into a depth map from a specified angle of view, and a Canny edge detection algorithm (Canny, 1986) is applied to obtain the edge points of the depth map (Fig. 2).

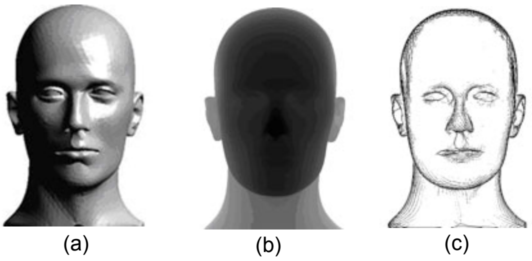

Fig. 2 Extraction of candidate feature points
(a) Model; (b) Depth map; (c) Edge points

The edge points extracted using the Canny edge detection algorithm are too many to directly construct a CRF model because the search space for a CRF model would be vast. Therefore, a filter is used based on the sketch strokes, and certain subsets of edge points are selected to construct several CRF models corresponding to the individual strokes. The convex rectangle of each stroke is computed and the inner edge points are selected as the candidate feature points. The search space is greatly reduced by applying such filters, but the matching results of CRF models will also depend on the positions of the strokes: if one stroke is further away from an actual feature line, matching processes may lead to a substandard result.

3.2 CRF-based matching

Extracting model feature lines with a template is in fact based on finding an optimized subset that best depicts the shape of the model surfaces and also fits the sketch strokes. There are several algorithms to match the sketch strokes and the model feature lines using simple relative distance (Mao *et al*., 2009) or construct an HMM model based on local shape differences between stroke points and feature points (Kraevoy *et al*., 2009), but only local shape features are considered in existing methods, and a lack of global shape and topological features means that these methods can handle only simple sketches or single strokes. Furthermore, these methods depend on conventional line drawing methods, and interactive information cannot be fully introduced. CRF modeling, however, is an extremely flexible technique for integrating different features in the same probabilistic framework, allowing features to be defined so as to capture different types of information (Ramos *et al*., 2007). In the proposed method, we construct a CRF matching model to extract the optimized model feature lines by combining the user template with the model geometric features.

3.2.1 CRF-matching model

A CRF model is an undirected graphical model developed for labeling sequence data (Lafferty *et al*., 2001). CRFs directly model $p(y|x)$, the conditional distribution over the hidden variables y given observations x. To establish correlations between sketch sample points (denoted by A) and candidate feature points (denoted by B), CRF-matching creates a graphical model based on sketch points and feature points. Each hidden variable y_i has a multinomial distribution, and each state j in y_i corresponds to the probability that feature i in A is matched to feature j in

B (Fig. 3a). The nodes y_i are connected by undirected edges which are in accordance with the connectivity structure of corresponding sketch points (Fig. 3b), and the conditional distribution $p(y|x)$ is defined by the hidden states y. The graph structure in our method can be a chain with extra links to represent long-term dependencies, which are used to ensure global consistence. The CRF conditional distribution can be factorized into a product of clique potentials, which are functions that map variable configurations to non-negative numbers. Intuitively, a potential captures the compatibility among the variables in the clique: the larger the potential value, the more likely the configuration. Local potentials can represent shape or geometric features.

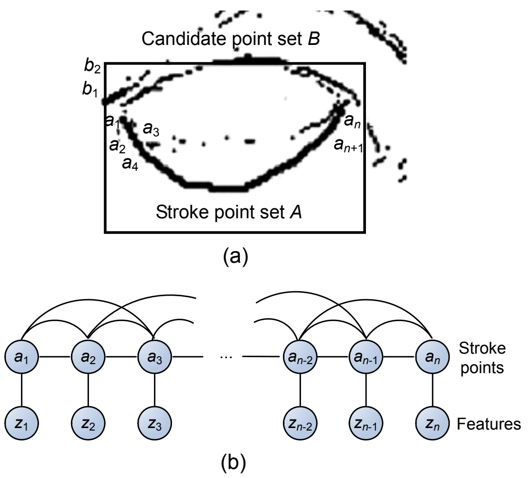

Fig. 3 Conditional random fields (CRF) matching model structure
(a) Candidate feature point set; (b) CRF chain

Inferences in CRF can estimate the marginal distribution of each hidden variable y_i, and can be solved using the BP algorithm, which works by sending local messages through the graph structure of the model (Murphy *et al*., 1999). Each node sends messages to its neighbors based on messages it receives and the clique potentials, which are defined via the observations and the neighborhood relation in the CRF. The most likely configuration of all hidden variables y can then be estimated using the Viterbi algorithm.

3.2.2 Feature function computation

CRF-matching can introduce arbitrary local potential to describe shape and geometric properties. Since the matching in the proposed method focuses on searching for feature point sequences that correspond with sketch sample points, the features should describe shape differences between sketch strokes and feature lines, and also the combination possibility of feature points based on image gradients.

Definition 1 (Shape similarity features) Shape similarity features are used to describe the shape and position differences between sketch sample points and candidate feature points. The local potentials are described as follows:

Definition 2 (2D point distance) The 2D point distance measures the Euclidian distance between sketch sample points and candidate feature point (Fig. 4a). Using a_i to denote the sketch sample point i and b_j the candidate feature point j, and $\boldsymbol{x}_{a_i}$ and $\boldsymbol{x}_{b_j}$ their 2D positions, the 2D point distance is defined as

$$f_{\mathrm{d}}(i,j,\boldsymbol{x}_A,\boldsymbol{x}_B)=\left\|\boldsymbol{x}_{a_i}-\boldsymbol{x}_{b_j}\right\|^2. \quad (1)$$

Definition 3 (Local angle difference) The local angle difference measures the local shape differences between individual sketch sample points and candidate feature points (Figs. 4b and 4c). Various features can depict local shape differences. In the proposed method, the angles between individual curve points are computed, and the differences between the angles of sketch points and depth map edge points are used to describe the local shape differences:

$$f_{\mathrm{s}}(i,j,k,\boldsymbol{x}_A,\boldsymbol{x}_B)=\left(\theta_{a_{ik}}-\theta_{b_{jk}}\right)^2, \quad (2)$$

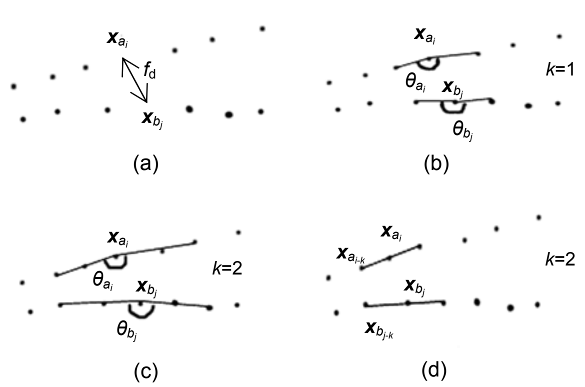

Fig. 4 Shape similarity features
(a) 2D point distance; (b) Local angle difference (k=1); (c) Local angle difference (k=2); (d) Local continuity (k=2)

where $\theta_{a_{ik}}$ denotes the angle between segments $\boldsymbol{v}_1=\overline{a_{i-k}a_i}$ and $\boldsymbol{v}_2=\overline{a_ia_{i+k}}$, $\theta_{b_{jk}}$ denotes the angle between segments $\boldsymbol{v}_3=\overline{b_{j-k}b_j}$ and $\boldsymbol{v}_4=\overline{b_jb_{j+k}}$, and k indicates the index offset to the adjacent point.

Definition 4 (Local continuity) Local continuity measures the continuous difference between points in sketches and depth maps (Fig. 4d), defined as

$$f_{\mathrm{c}}(i,j,k,\boldsymbol{x}_A,\boldsymbol{x}_B)=\frac{\left\|\boldsymbol{x}_{a_i}-\boldsymbol{x}_{a_{i-k}}\right\|^2}{\left\|\boldsymbol{x}_{b_j}-\boldsymbol{x}_{b_{j-k}}\right\|^2}-1. \tag{3}$$

Definition 5 (Image space features) Image space features are used to describe the possibility of connecting candidate feature points to feature lines, defined by the image gradient of the depth map which is computed using the Canny edge detection algorithm.

Definition 6 (Canny gradient) The Canny gradient measures the possibility of labeling candidate feature points as actual feature points. Image edge detection studies show that the pixel with a higher gradient has a higher possibility of becoming an actual edge point, and in a similar way a candidate feature point with a higher gradient is more likely to be an actual feature point. The Canny gradient is defined as

$$f_{\mathrm{g}}(j,\boldsymbol{x}_B)=g_{b_j}/g_{\max}, \tag{4}$$

where g_{b_j} is the gradient of candidate feature point b_j, and $g_{\max}$ is the largest gradient in the depth map.

Definition 7 (Gradient continuity) Gradient continuity measures the continuum of two candidate feature points, and is described by the difference of gradients between adjacent feature points:

$$f_{\mathrm{gc}}(j,k,\boldsymbol{x}_B)=\left|(g_{b_j}-g_{b_{j-k}})^2-(g_{b_{j+k}}-g_{b_j})^2\right|. \tag{5}$$

3.3 Iterative optimization

In our proposed method, a filter is performed according to the sketch strokes, greatly reducing the number of feature points used in CRF matching. The convex rectangle of each stroke is computed and the inner candidate feature points are selected as the candidate subset of this stroke. The search space is greatly reduced by applying such a filter, but as a result the matching results of the CRF model will also depend upon the positions of sketch strokes: if the stroke is too far away from the actual feature line, the matching may lead to a substandard result which deviates from the best feature line. Introducing a step of rationality evaluation and candidate feature points migration is necessary for the filter to optimize the feature lines extracted iteratively.

3.3.1 Rationality metric of feature lines

Since the convex rectangles of sketch strokes are not necessarily located near the actual feature area, a metric that can measure the differences between actual feature lines and the feature lines extracted by CRF-matching is required. In our method, a rationality metric based on shape and topological similarity is proposed to quantify the rationality of feature lines. The rationality metric is composed of three parts based on the global distance, shape similarity, and topological similarity.

Global distance: This measures the global distance between sketch strokes and feature lines. Since the actual feature lines wanted for extraction are always close to the sketch strokes, the resulting feature lines are more probable to be the actual ones if the global distance is relatively small. The global distance is defined as the average Euclidian distance of corresponding points in sketches and depth maps:

$$E_{\mathrm{d}}(A,B)=\frac{n}{\sum\left\|\boldsymbol{x}_{a_i}-\boldsymbol{x}_{b_j}\right\|^2}, \tag{6}$$

where A denotes the strokes in the user's sketch, B the extracted feature lines corresponding to A, and n the number of sample points in A.

Global shape similarity: This measures the global shape differences between resultant feature lines and their corresponding strokes. The resulting feature line is more likely to be an actual one if their global shapes are similar. In our method, a Hausdorff distance is used to determine the global shape similarity:

$$\begin{aligned}&E_{\mathrm{s}}(A,B)=\max\left(d(A,B),d(B,A)\right),\\&d(A,B)=\max_{a\in A}\min_{b\in B}d(\boldsymbol{x}_a,\boldsymbol{x}_b),\end{aligned} \tag{7}$$

where $\boldsymbol{x}_a$ and $\boldsymbol{x}_b$ denote the points in strokes A and feature lines B, respectively.

Topological similarity: this measures the structural topological similarity of sketch strokes and feature lines. Topological consistency is an important aspect for sketching and should be maintained in the matching process. Graph structure is usually used to describe topological information, but the topological similarity between the two graphs is difficult to measure quantitatively. Hence, a relative position difference metric of corresponding strokes and feature lines is used to delineate the summarized distance for corresponding lines, which is an approximation of the topological similarity:

$$E_t(S)=\sum_{i=0}^{n-1} E_t(S_i), \tag{8}$$

where

$$E_t(S_i)=\sum_{j=1,j\neq i}^{n}\left\|A_i-A_j\right\|-\sum_{j=1,j\neq i}^{n}\left\|B_i-B_j\right\|,$$

and A_k and B_k (k=0, 1, …, n) denote the corresponding stroke and feature line, respectively.

3.3.2 Migration of the matching area

The strokes in the sketch should be similar to those in a given 3D model, so that a filter based on the sketch strokes can be used to obtain candidate feature points around that stroke (Section 2.1), but it is not ensured that all the corresponding feature points can be found in the filter area. By properly defining the initial matching point and matching area, the search space of CRF-matching can be reduced, but the matching result will be largely influenced. If the sketch stroke is close to the actual feature line, then the matching can be quickly located to the actual feature line; if the stroke is further away from the actual feature line, the matching may not achieve a sound result. Therefore, an iterative optimization is needed to progressively locate the filter area closer to the actual feature line. Iterative optimization is performed in the following steps:

1. Select n initial matching points $\{\boldsymbol{p}_1, \boldsymbol{p}_2, \dots, \boldsymbol{p}_n\}$ randomly in the circle with radius r around the seed point (first as s_0 in stroke), and the matching process is performed with an initial matching point (Fig. 5b).

2. Extract n resulting feature lines $\{f_1, f_2, \dots, f_n\}$ using the CRF-matching model, evaluate the rationality metric of each feature line, and then select the most likely feature line f_i and its corresponding initial matching point as the seed point $\mathbf{sp}_k$ of the kth iteration (Fig. 5c).

3. In accordance with the displacement $\boldsymbol{d}=\mathbf{sp}_{i-1}-\mathbf{sp}_i$ between the seed points of ($i-1$)th and ith iterations, the matching area is migrated along $\boldsymbol{d}$ when the rationality value reaches the threshold, and if the rationality is lower than the threshold, the matching area should be expanded along $\boldsymbol{d}$ (Fig. 5d).

4. Steps 2 and 3 are repeated until the rationality value is high enough, and the feature lines obtained are selected as the final results.

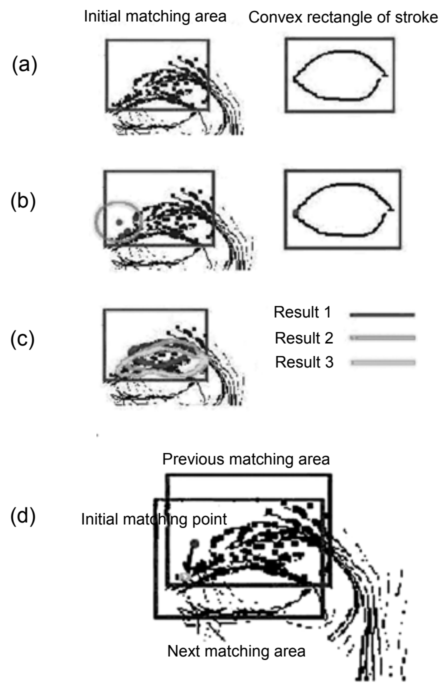

Fig. 5 Process of iterative optimization
(a) Matching area; (b) Initial matching point selection; (c) Matching result with various initial matching points; (d) Matching area migration

4 Experimental results

The proposed method is implemented with C++, and the 3D models used are selected mainly from the Princeton benchmark dataset. Given a 3D model, the viewpoint and sketch are specified by the user, and an extraction process is then applied to obtain the corresponding feature lines. The feature lines of various objects and views have been obtained, and the characteristics of sketches including the number, shape, and topology of strokes have been captured.

Fig. 6 shows the feature lines extracted from different face models based on the same sketch. Although there are differences in mesh quality and shape between the various models, the feature lines are extracted reasonably and fit the significant features well. The characteristics of the sketches, such as the number, shape, and topology of strokes, are maintained during the extraction process, and the template drawn by the user is also conveyed well.

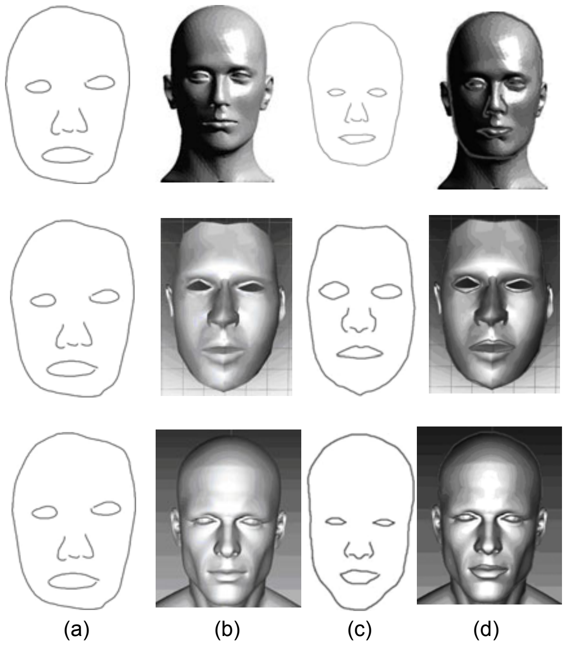

Fig. 6 Sketch drawing templates and various models
(a) Sketch; (b) Model; (c) Result; (d) Result+Model

Since the extraction process is based on a template of a given sketch, the feature lines can be computed properly only if the feature points are extracted as a Canny edge points set. As a result, the feature lines that best reflect the user's template can be extracted easily even if mesh quality is poor or model features are not so salient.

Fig. 7 shows the results extracted for the chest, collarbone, and abdomen of a body model. Some parts of the surface are smoother than the other parts (near the abdomen). If a conventional feature line extraction method is performed, such as suggestive contours (DeCarlo *et al*., 2003) and ridges and valleys (Judd *et al*., 2007), the results may not be satisfactory no matter how the threshold is set. Note that RTSC from Princeton University is used here for comparison. The shape features are quite different in each part of the surface. Some of the real feature lines are lost when the threshold is too high, e.g., SC Thresh=0.1 (abdomen feature lines in Figs. 8a and 8b), and many redundant lines are extracted at a very low threshold, e.g., RV Thresh=0.02 (Fig. 8c). The results of the proposed method show that for 3D models with various degrees of smoothness on different sections, reasonable feature lines can be obtained using the template.

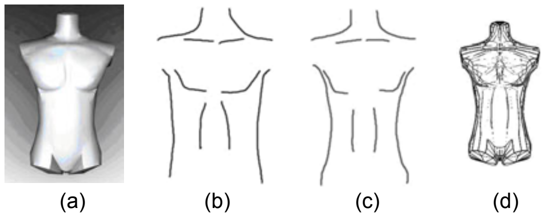

Fig. 7 Feature line extraction for the body model
(a) 3D model; (b) Sketch; (c) Result; (d) Result+Model

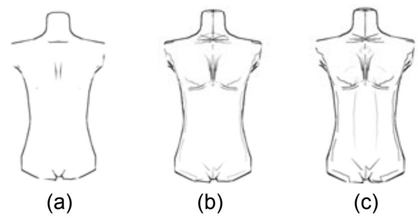

Fig. 8 Extraction results of the body model using RTSC with a high (a), medium (b), or low (c) threshold

Fig. 9 shows the results extracted for a rabbit model, involving many irregular mesh structures on the model surface. The extraction results show that for 3D models with poor mesh quality, our proposed method obtains exact feature lines which are consistent with user perception, without the need for additional preprocessing or mesh optimization. The results extracted using conventional methods are shown in Fig. 10. The inner details are lost at a high threshold, e.g., SC Thresh=0.1 (Fig. 10a), many fragmented contours are extracted at a very low threshold, e.g., SC Thresh=0.02 (Fig. 10c), and some important feature lines cannot be extracted in both situations, such as the contours under the head and around the legs.

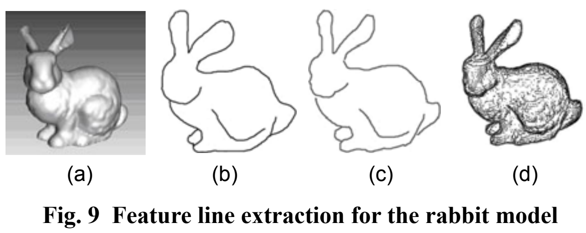

Fig. 9 Feature line extraction for the rabbit model
(a) 3D model; (b) Sketch; (c) Result; (d) Result+Model

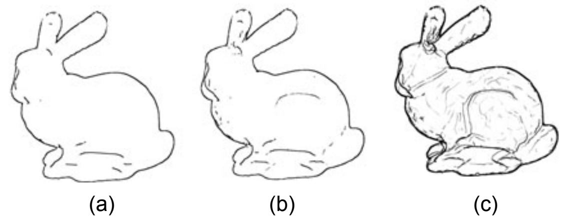

Fig. 10 Extraction results of the rabbit model using RTSC with a high (a), medium (b), or low (c) threshold

The Canny threshold significantly influences the selection of candidate feature points used in the proposed method. Figs. 11a and 11b show the template and 3D model, with some difference around the left eye and the mouth between the strokes and the actual object area. Figs. 11c–11e show the matching results of the left eye and the mouth at a low threshold (λ=0.00005). Figs. 11d–11f show the results at a high threshold (λ=0.0001). These results demonstrate that the edge points may be increasingly redundant at a low threshold; at a high threshold, the matching results will be much closer to the actual feature lines. The emphasis of each feature function in the CRF-matching model can be adjusted to make the matching results more consistent with the user template. In the proposed method the parameters are set to empirical values, and adopting a model parameterization with statistical measurements according to a specific sample would be more effective.

The main threshold in Canny edge detection is the Canny gradient threshold λ, which may alter the search space of matching between sketch strokes and feature points. The value of λ actually leverages the computational complexity and search space in the matching process: when λ is increased, the search space of matching gets smaller and the complexity of the matching process is reduced, but a too high threshold will make the matching point set (all the candidate feature points for matching) lack actual feature points. Similarly, a too low threshold will produce too many redundant matching points. The λ should therefore be set to an appropriate value to make the matching point number as small as possible, while the actual feature points are extracted. Fig. 12 shows different edge points extracted with various λ. In our proposed method, λ is set to 0.00005.

Fig. 13 shows the iterative optimization results of the human mouth. The initial matching point will leave the drawing area and the extracted feature lines

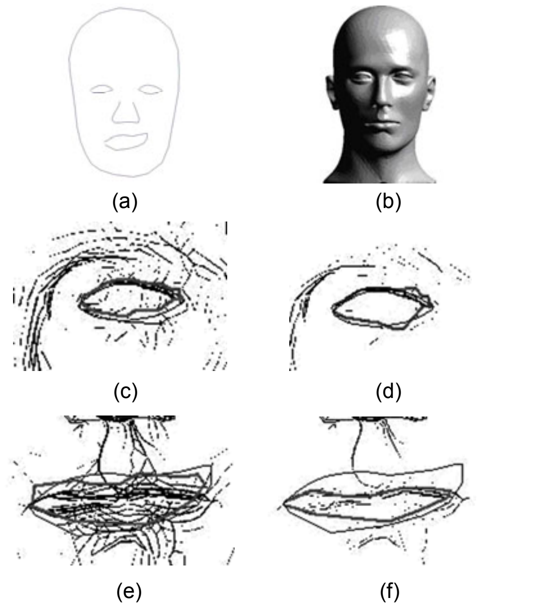

Fig. 11 The influence of Canny threshold λ

(a) Sketch; (b) 3D model; (c) The eye extracted with λ=0.00005; (d) The eye extracted with λ=0.0001; (e) The mouth extracted with a low λ; (f) The mouth with a high λ. Strokes are marked in thin lines, and matching results in thick lines

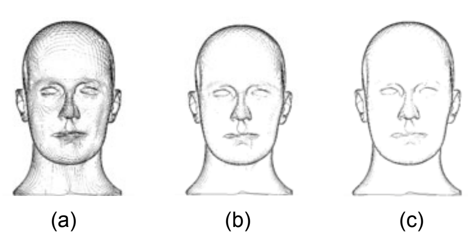

Fig. 12 Sample feature points with the Canny gradient threshold λ=0.0002 (a), λ= 0.00005 (b), and λ=0.00008 (c)

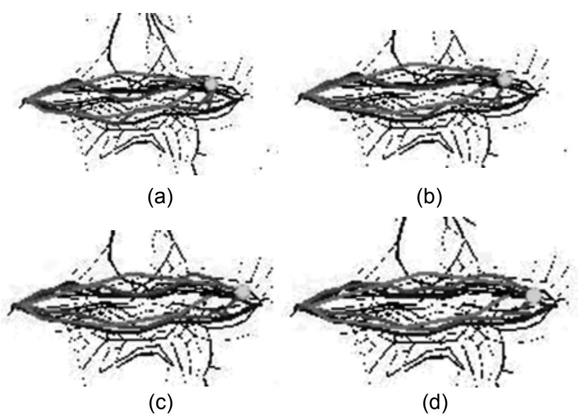

Fig. 13 Iterative optimization process

(a) Initial position; (b) After two iterations; (c) After five iterations; (d) After 10 iterations. Initial matching point is labeled on the right are closer to the actual feature lines, and are finally located in the optimization area. The iterative optimization shown above can progressively make the matching results be approximately sound feature lines, only if sketch strokes are not too far away from the actual feature area.

5 Conclusions

This paper proposes a 3D model feature line extraction method which allows users to manipulate the extraction with a template. A conditional random fields (CRF) matching model with multiple features between stroke points and candidate feature points is established, which effectively combines the user's sketch with the model feature line extraction process. To match the sketch points with corresponding feature points, a similarity metric based on topological structure is proposed to evaluate the rationality of the extracted feature lines, and an iterative optimization process is applied to obtain globally optimized model feature lines. The results show that the method is efficient in extracting valid 3D model feature lines which correspond to the guidance sketch.

This method, however, has two main drawbacks. First, the parameters used, including feature functions and the Canny edge detection threshold, are set empirically, and may not obtain the best results for certain kinds of models, and a distorted sketch drawing based on statistical measurements (such as data probabilities of a given model type) to obtain model parameters using sample data may be more effective. Second, the image space features used in this method make for a quicker extraction, but are also constrained by image resolution. Using object space features in matching would be more effective for very complex feature lines, and stylization can thus be applied to the matching results (Kalogerakis *et al.*, 2009).

References

Canny, J., 1986. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, **8**(6):679-698. [doi:10.1109/TPAMI.1986.4767851]

Catalano, C.E., Mortara, M., Spagnuolo, M., Falcidieno, B., 2011. Semantics and 3D media: current issues and perspectives. *Comput. Graph.*, **35**(4):869-877. [doi:10.1016/j.cag.2011.03.040]

Cole, F., Golovinskiy, A., Limpaecher, A., Barros, H.S., Finikelstein, A., Funkhouser, T., Rusinkiewicz, S., 2008. Where do people draw lines? *ACM Trans. Graph.*, **27**(3), Article 88, p.1-11. [doi:10.1145/1360612.1360687]

DeCarlo, D., Finkelstein, A., Rusinkiewicz, S., Santella, A., 2003. Suggestive contours for conveying shape. *ACM Trans. Graph.*, **22**(3):848-855. [doi:10.1145/882262.882354]

Hertzmann, A., 1999. Introduction to 3D Non-photorealistic Rendering: Silhouettes and Outlines. ACM SIGGRAPH Course Notes, p.15-29.

Hertzmann, A., 2010. Non-photorealistic Rendering and the Science of Art. Proc. 8th Int. Symp. on Non-photorealistic Animation and Rendering, p.147-157. [doi:10.1145/1809939.1809957]

Interrante, V., Fuchs, H., Pizer, S., 1995. Enhancing Transparent Skin Surfaces with Ridge and Valley Lines. Proc. 6th Conf. on Visualization, p.52-59. [doi:10.1109/VISUAL.1995.480795]

Judd, T., Durand, F., Adelson, E.H., 2007. Apparent ridges for line drawing. *ACM Trans. Graph.*, **26**(3), Article 19. [doi:10.1145/1276377.1276401]

Kalogerakis, E., Nowrouzezahrai, D., Simari, P., McCrae, J., Hertzmann, A., Singh, K., 2009. Data-driven curvature for real-time line drawing of dynamic scenes. *ACM Trans. Graph.*, **28**(1), Article 11, p.1-13. [doi:10.1145/1477926.1477937]

Kalogerakis, E., Nowrouzezahrai, D., Breslav, S., Hertzmann, A., 2012. Learning hatching for pen-and-ink illustration of surfaces. *ACM Trans. Graph.*, **31**(1), Article 1, p.1-17. [doi:10.1145/2077341.2077342]

Kraevoy, V., Sheffer, A., Michiel, P., 2009. Modeling from Contour Drawings. Proc. 6th Eurographics Symp. on Sketch-Based Interfaces and Modeling, p.37-44. [doi:10.1145/1572741.1572749]

Lafferty, J., McCallum, A., Pereira, F., 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. Proc. 18th Int. Conf. on Machine Learning, p.282-289.

Lee, Y., Markosian, L., Lee, S., Hughes, J.F., 2007. Line drawings via abstracted shading. *ACM Trans. Graph.*, **26**(3), Article 18, p.1-6. [doi:10.1145/1276377.1276400]

Lum, E.B., Ma, K.L., 2005. Expressive line selection by example. *Vis. Comput.*, **21**(8-10):811-820. [doi:10.1007/s00371-005-0342-y]

Mao, C., Qin, S.F., Wright, D., 2009. A sketch-based approach to human body modeling. *Comput. Graph.*, **33**(4):521-541. [doi:10.1016/j.cag.2009.03.028]

Murphy, K., Weiss, Y., Jordan, M., 1999. Loopy Belief Propagation for Approximate Inference: an Empirical Study. Proc. Conf. on Uncertainty in Artificial Intelligence, p.467-475.

Olsen, L., Samavati, F.F., Sousa, M.C., Jorge, J.A., 2009. Sketch-based modeling: a survey. *Comput. Graph.*, **33**(1):85-103. [doi:10.1016/j.cag.2008.09.013]

Ramos, F., Fox, D., Durrant, W.H., 2007. CRF-Matching: Conditional Random Fields for Feature-Based Scan Matching. Proc. Robotics: Science and Systems, p.201-208.

Saito, T., Takahashi, T., 1990. Comprehensible Rendering of 3-D Shapes. *ACM SIGGRAPH Comput. Graph.*, **24**(4):197-206. [doi:10.1145/97880.97901]