Minimum Near-Convex Shape Decomposition

Zhou Ren, Junsong Yuan, and Wenyu Liu

Abstract—Shape decomposition is a fundamental problem for part-based shape representation. We propose the Minimum Near-Convex Decomposition (MNCD) to decompose arbitrary shapes into minimum number of “near-convex” parts. The near-convex shape decomposition is formulated as a discrete optimization problem by minimizing the number of non-intersecting cuts. Two perception rules are imposed as constraints into our objective function to improve the visual naturalness of the decomposition. With the degree of near-convexity a user specified parameter, our decomposition is robust to local distortions and shape deformation. The optimization can be efficiently solved via Binary Integer Linear Programming. Both theoretical analysis and experiment results show that our approach outperforms the state-of-the-art results without introducing redundant parts, and thus leads to robust shape representation.

Index Terms—Shape decomposition, shape representation, discrete optimization.

1 Introduction

PART-based representation has shown promising results in object recognition and detection. Instead of characterizing an object as a whole, visual parts serve as intermediate components to represent an object and are robust to articulated movement and partial occlusions [2].

In this work we study the problem of decomposing an arbitrary object shape into a number of natural parts. To ensure a meaningful decomposition, each decomposed part is preferred to have a convex shape, namely a simple polygon whose interior is a convex set. Due to the nice geometric property of convex shape, it brings two advantages. First, a convex part is visually natural and geometrically simple [3] [4], thus can serve as a satisfactory primitive for recognition. Second, many complicated operations, although not feasible to apply to arbitrary shape, can be easily applied to its convex parts [5] [6].

Although strict convex decomposition has been well-studied in computational geometry [9] [10], because of its sensitivity to small variations of the shape, it cannot bring stable decomposition, thus limits its application in real problems. For example, a small local distortion on the contour, caused by imperfect image segmentation or shape deformations, can lead to completely different decomposition result. Moreover, to satisfy the strict convex requirement, it is like to result in a large number of redundant small parts, thus does not lead to stable decomposition either.

To address these problems, near-convex decomposition has been proposed [7]. As illustrated in Fig.1, instead of requiring each part to be strictly convex, it allows near-convex parts. In [7] [11], Lien et al. proposed a divide-and-conquer strategy for near-convex decomposition, which partitions the most concave feature in the shape until all the parts satisfy the convexity constraint. A recent method proposed by Liu et al. [8] formulated the near-convex decomposition as a linear programming problem by minimizing the total length of cuts. As it can tolerate local non-convex distortions, near-convex decomposition usually leads to more robust representation.

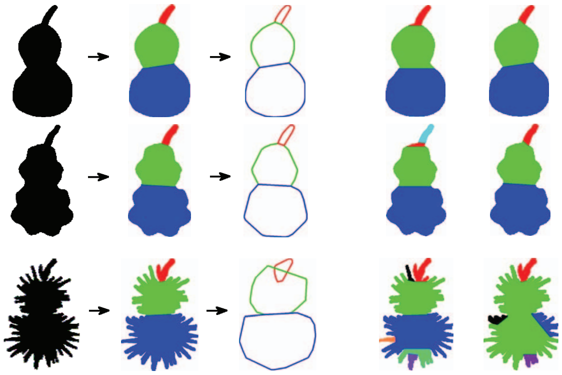


Fig. 1. The first column shows the same objects with different degrees of local distortions. The second column shows the near-convex decomposition results using our method. The third column shows the shape representations by replacing each part with its convex hull. Despite severe local distortions, as our method decomposes a shape into minimum number of near-convex parts, it avoids introducing redundant parts and thus brings consistent decomposition results. The last two columns are the results of the existing near-convex decomposition methods: [7] and [8], respectively.

Despite previous works in near-convex shape decomposition, there still remain two unsolved problems. First of all, existing methods cannot avoid introducing redundant parts. For example, the greedy algorithm proposed in [7] [11] inevitably results in redundant parts and unstable decomposition. By only optimizing the total cut length, the method of [8] generates redundant parts and is not stable either, as illustrated in the last two columns of Fig.1. Secondly, without any priori knowledge of the object, it is difficult to obtain visually natural parts through an unsupervised decomposition.

- *Zhou Ren and Junsong Yuan are with Nanyang Technological University, Singapore.*
- *Wenyu Liu is with Huazhong University of Science and Technology, Wuhan, China.*

A preliminary version of this paper appeared in the IEEE International Conference on Computer Vision [1].

Digital Object Indentifier 10.1109/TPAMI.2013.67 0162-8828/13/$31.00 © 2013 IEEE

To address these two problems, we present the Minimum Near-Convex Decomposition (MNCD) to decompose an arbitrary shape into minimum number of parts. To achieve this goal, we first construct the candidate cut set, and then decompose the shape by selecting a subset of cuts from the candidate set. In our new formulation, the decomposition is optimized by selecting the best subset which has both the minimum size and high degree of visual naturalness. To improve the visual naturalness, two perception rules, the minima rule [12] and the short cut rule [13], are imposed into the objective function. In order to solve the discrete optimization problem efficiently, we formulate our problem as Binary Integer Linear Programming (BILP), which can be solved in seconds while still guarantee the optimal solution.

Our shape decomposition method provides a compact and effective way to represent shapes. For example, as shown in the third column of Fig.1, we can approximately represent the original shapes by replacing each part with its convex hull. Based on this representation method, we have validated its superiority in the application of hand gesture recognition using Kinect sensor [14] [15], which achieves the state-of-the-art recognition accuracy.

The main contributions of this paper include: (1) we propose minimum near-convex shape decomposition which decomposes an arbitrary 2D shape into minimum number of near-convex parts; (2) our decomposition method is stable, can well handle local shape distortions and shape deformation, and is visually more natural. Extensive experiments on benchmark shape datasets and the comparisons with the state-of-the-art methods validate the advantages of our decomposition algorithm, MNCD.

The rest of the paper is organized as follows. We introduce the related work of shape decomposition in Section 2. In Section 3 we present the formulation of MNCD. Then we discuss the properties of our formulation and present the BILP solution of MNCD in Section 4. And in Section 5, experiments of MNCD on 2D shapes in terms of decomposed parts number, visual naturalness, decomposition robustness and applications are demonstrated. Finally we draw conclusion in Section 6.

2 Related Work

Shape decomposition is a fundamental step toward shape analysis and understanding [3] [16]. Such representation method is widely used in shape recognition [14] [15] [17], shape retrieval [18] [19], skeleton extraction [20] [21] [22], and motion planning [23] [24].

We can classify most shape decomposition methods into two categories. One category is based on geometric constraints. The other category is motivated by psychological studies.

In the first category, the most popular geometric constraint is convexity constraint. This is not only because convex parts have nice topological and geometric properties that allow for certain operations and improve the efficiency of algorithms, but also because convexity plays an important role in human perception [3]. There are two main indices to evaluate the performance of convex decomposition methods. One index is the time complexity. In this area, Keil et al. proved the time bound of convex decomposition to $O(n{+}r^2\min(r^2,n))$, where n is the number of vertices and r is the number of notches [9]. The other index is the number of decomposed parts. In this area, Snoeyink proposed minimum convex decomposition that can decompose 2D shapes into minimum number of strict convex parts [10]. However, strict convex decomposition always produces an unmanageable number of parts and is very time consuming. Besides, there is no need to find the strict convex parts; a certain degree of approximation is enough to satisfy practical processes and can lead to a much more robust representation. Lien and Amato proposed Approximate Convex Decomposition in [7] [11], which decomposes 2D and 3D shapes into approximately convex parts. Liu et al. proposed Convex Shape Decomposition [8] to minimize the total length of cuts. In our previous work [1] we proposed a quadratic programming formulation to decompose a shape into minimum number of parts. In these methods, they ignored small concave features and made the decomposition more robust and efficient.

The second category, motivated by psychological studies, aims to decompose shapes into natural parts. The meaning of natural parts depends on human perception and thus has no objective definition. However, there are some basic perception rules from cognitive science. In [12], Hoffman proposed the $minima\ rule$, which pointed out that human visual system was interested in boundaries at negative minima of principal curvature or concave creases. Another major perception rule is the $Short\ cut\ rule$, proposed by Singh, Seyranian, and Hoffman [13], which stated that human preferred the shortest possible cuts for decomposition.

The aim of our method is to decompose a shape into minimum number of near-convex parts. And the two major perception rules are also imposed to guide the decomposition, in order to ensure high degree of visual naturalness.

3 Problem Formulation

3.1 Overview

In near-convex decomposition, each decomposed part may not be strictly convex, thus the user has to specify a parameter ψ which indicates the near-convex tolerance of the decomposed parts. Formally, a ψ-near-convex decomposition of a shape S, $\mathrm{D}_\psi(S)$, is defined as a decomposition that only contains ψ-near-convex non-overlapping parts, i.e.:

$$\mathrm{D}_\psi(S)=\{P_i|\bigcup_i P_i=S,\forall_{i\neq j}P_i\cap P_j=\emptyset,concave(P_i)\leq\psi\},\quad(1)$$

where P_i denotes the decomposed part; $concave(P_i)$ is the concavity of P_i. We say P_i is ψ-near-convex if $concave(P_i)$ $\leq\psi$. P_i is strictly convex if $concave(P_i)$=0. According to the definition, near-convex decomposition has two constraints: *the non-overlapping constraint*, $\forall_{i\neq j}P_i\cap P_j=\emptyset$; and *the convexity constraint*, $\forall P_i$, $concave(P_i)\leq\psi$.

The partition $\{P_i\}$ is formed by some cuts. For any two vertices p, q on the contour, if the line connecting p and q locates inside the shape, line pq is a cut. As shown in Fig.2(b), the red lines are some sampled cuts. We denote the complete set of all possible cuts in shape S as the candidate

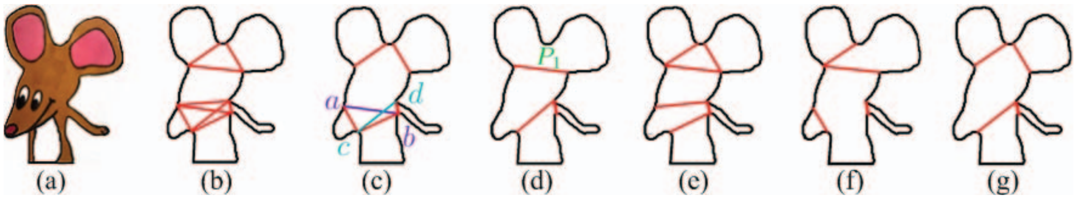


Fig. 2. Illustration of near-convex decomposition. (a) The original image. (b) The extracted shape with some sampled candidate cuts inside. (c) An incorrect near-convex decomposition, which does not satisfy *the non-overlapping constraint*, as the purple line ab intersects with the cyan line cd, causing the part abc overlaps with the part bcd. (d) An incorrect near-convex decomposition which does not satisfy *the convexity constraint*, as $concave(P_1) > \psi$. (e) A near-convex decomposition of 7 parts. (f) A minimum near-convex decomposition of 5 parts. (g) Another minimum near-convex decomposition of 5 parts, but looks more natural.

cut set, $C(S)$. Therefore, as shown in Fig.2, a near-convex decomposition of S is to select a subset of cuts from $C(S)$ to form $\{P_i\}$ such that the two constraints in Eq.1 are satisfied: 1) as illustrated in Fig.2(c), to ensure the non-overlapping constraint, the selected cuts cannot intersect with each other; and 2) as illustrated in Fig.2(d), to ensure the convexity constraint, we restrict $\forall P_i$, $concave(P_i) \leq \psi$.

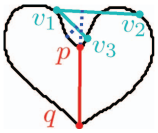


Fig. 3. At the concave contour, some lines (such as v_1v_2, v_1v_3) intersect with the contour or locate outside the contour, which form the mutex pairs; while vertices v_2, v_3 are not a mutex pair.

In order to measure $concave(P_i)$, we use the shape feature $mutex\ pair$ in proposed [8]: for any two vertices on a shape contour, v_1 and v_2, if the connecting line between v_1 and v_2 intersects with the contour or locates outside the contour, (v_1, v_2) is a mutex pair. As shown in Fig.3, (v_1, v_2) and (v_1, v_3) are two mutex pairs. The concavity of a part P_i is defined as the maximal concavity of the mutex pairs in the part:

$$concave(P_i) = \max_{(v_1,v_2)\in P_i} \{concave_m(v_1, v_2)\}, \tag{2}$$

where (v_1, v_2) denotes the mutex pair in P_i; $concave_m(v_1, v_2)$ is the concavity of mutex pair (v_1, v_2).

Hence, we can measure $concave(P_i)$ by measuring all $concave_m(v_1, v_2)$ in P_i. We use the same method proposed in [8] to measure $concave_m(v_1, v_2)$: by projecting the shape contour in multiple Morse functions, the concavity of a mutex pair is defined as the maximal perpendicular distance between line v_1v_2 and the corresponding concave contour. As in Fig.3, $concave_m(v_1, v_2)$, $concave_m(v_1, v_3)$ are shown as the blue dotted lines, and $concave_m(v_1, v_2) > concave_m(v_1, v_3)$.

To ensure the convexity constraint: $\forall P_i$, $concave(P_i) \leq \psi$, according to Eq.2, the concavities of all the mutex pairs in each part P_i must be smaller than ψ. Therefore, for a ψ-near-convex decomposition, we need to separate all the mutex pairs in S whose concavities are greater than ψ into different parts to ensure $concave(P_i) \leq \psi$. As illustrated in Fig.3, cut pq separates the heart shape into two parts, and the mutex pair (v_1, v_2) as well as (v_1, v_3) are separated. Thus $concave_m(v_1, v_2)$ and $concave_m(v_1, v_3)$ will not affect the concavities of these two parts.

3.2 Minimum Near-Convex Decomposition

As illustrated in Fig.2(e), Fig.2(f) and Fig.2(g), in order to decompose a shape into minimum number of parts with high degree of visual naturalness, we need to optimize the selection of cuts. Assume there are in total n possible cuts in a shape S, namely $C(S) = \{cut_1, ..., cut_n\}$. The final decomposition consists of a subset of the cuts from $C(S)$, denoted by $C'(S) \subseteq C(S)$. We assign a binary variable x_i to each cut_i in $C(S)$ where:

$$x_i = \begin{cases} 1 & cut_i \in C'(S), \\ 0 & cut_i \notin C'(S). \end{cases} \tag{3}$$

Thus $\mathbf{x}_{n\times 1}$ =$(x_1, x_2, ..., x_n)^\top$ is a binary vector indicating the selection/rejection of cuts from $C(S)$.

With the two constraints in Eq.1, and by minimizing the number of cuts and imposing perception rules, we formulate the ψ-MNCD as a Binary Integer Linear Programming problem as follows:

$$\begin{aligned} &\mathbf{min} \quad \parallel \mathbf{x} \parallel_\mathbf{0} + \lambda \mathbf{w}^\top \mathbf{x}, \\ &s.t. \quad \mathbf{Ax} \geq \mathbf{1}, \quad \mathbf{Bx} \leq \mathbf{1}, \quad \mathbf{x} \in \{0, 1\}^n, \end{aligned} \tag{4}$$

where $\|\mathbf{x}\|_0$ is the zero-norm of vector $\mathbf{x}$, which counts the number of the selected cuts in $C'(S)$. $\lambda \geq 0$ is a parameter introducing the visual naturalness regularization $\mathbf{w}^\top\mathbf{x}$ to the decomposition, in order to regularize the cuts selection by favoring the cuts with higher visual naturalness. We will discuss λ in Section 4.1. Different from [1], we present a new definition of intersection matrix $\mathbf{B}$, thus propose a Binary Integer Linear Programming formulation. Now we explain our formulation.

The visual naturalness regularization: $\mathbf{w}^\top\mathbf{x}$

We employ both the minima rule [12] and the short cut rule [13] to ensure high degree of visual naturalness for the decomposition. A cost is assigned to each $cut_i \in C(S)$ to evaluate its own visual naturalness, and a smaller cost means a higher degree of visual naturalness:

$$w_{pq} = \frac{dist(pq)}{1 + \beta \cdot |\min\{cur(p), 0\} + \min\{cur(q), 0\}|}, \tag{5}$$

where cut pq is a candidate cut in $C(S)$; $dist(pq)$ is the normalized distance between vertices p and q. This corresponds to the short cut rule: a shorter cut has a smaller cost. $cur(p)$ denotes the normalized curvature of the vertex p, which corresponds to the minima rule: a cut decomposing at positions with greater negative curvatures has a smaller cost. We normalize the negative curvature among concave vertices and ignore the convex vertices. β is a parameter balancing these two rules. As both rules are critical for natural decomposition, we set $\beta = 1$ in our experiments.

We denote $\mathbf{w}_{n\times 1}$=$(w_1, w_2, ..., w_n)^\top$ as the costs of n candidate cuts. From Eq.5, we know that the cuts separating at positions with greater negative curvatures and with shorter lengths have smaller costs. Thus by minimizing $\mathbf{w}^\top\mathbf{x}$, those cuts with higher visual naturalness are more likely to be selected.

The convexity constraint: Ax≥1

As mentioned in Section 3.1, to ensure the convexity constraint: $\forall P_i$, $concave(P_i) \leq \psi$, we need to separate all the mutex pairs whose concavities are greater than ψ into different parts. So we first obtain the ψ-mutex set of S, $M^\psi(S)$, which is defined as the set of mutex pairs whose concavities are greater than ψ. Then we separate all the mutex pairs in $M^\psi(S)$ with the selected cuts from $C(S)$. A candidate cut may separate several mutex pairs, such as the cut pq in Fig.3. For every candidate cut in $C(S)$, cut_i, the mutex pairs it can separate form a subset of $M^\psi(S)$, denoted by M'_i. In this way, we obtain $\{M'_i,\ i = 1, ..., n\}$.

Suppose there are m mutex pairs in the ψ-mutex set, $M^\psi(S) = \{mp_1, ..., mp_m\}$. For each mutex pair in $M^\psi(S)$, mp_i, among all the cuts that can separate it, at least one cut must be in set $C'(S)$. Thus, for each mp_i, this gives a constraint:

$$\sum_{j=1}^{n} a_{ij}x_j \geq 1,\ \text{where}\ a_{ij} = \begin{cases} 1 & mp_i \in M'_j, \\ 0 & mp_i \notin M'_j. \end{cases} \qquad (6)$$

Let us denote $\mathbf{A}_{m\times n}$=$(a_{ij}|i = 1, ..., m;\ j = 1, ..., n)$, $\mathbf{1}_{m\times 1}$=(1, ..., 1)$^\top$. Consider all the m mutex pairs in $M^\psi(S)$, we have the convexity constraint: **Ax≥1**, which is also used in [8].

The non-overlapping constraint: Bx≤1

Any two cuts in $C(S)$ may intersect with each other at an intersection. Now we define an intersection matrix, $\mathbf{B}_{t\times n}$ to indicate the intersection relations in $C(S)$, where t is the total number of intersections. Suppose cut_i and cut_j intersect at the gth intersection, then we define the gth row of matrix $\mathbf{B}$ as:

$$b_{gz} = \begin{cases} 1 & z = i \text{ or } j, \\ 0 & \text{otherwise}. \end{cases} \qquad (7)$$

As mentioned in Section 3.1, to ensure the non-overlapping constraint $\forall_{i\neq j} P_i \cap P_j = \emptyset$, the selected cuts in $C'(S)$ cannot intersect with each other, namely $\forall_{g=1,...,t}\ \sum_{z=1}^{n} b_{gz}x_z \leq 1$. Thus we have the intersection constraint: **Bx≤1**, where $\mathbf{1}_{t\times 1}$=(1, ..., 1)$^\top$.

4 Solution

4.1 Selection of Parameter λ

As mentioned earlier, λ is an important parameter introducing the visual naturalness regularization to the decomposition. If we do not consider the visual naturalness of the decomposition, while only focus on minimizing the number of parts, the problem can be reformulated by setting $\lambda = 0$, i.e.:

$$\mathbf{min} \quad \| \mathbf{x} \|_0 \quad s.t.\ \mathbf{Ax} \geq \mathbf{1},\ \mathbf{Bx} \leq \mathbf{1},\ \mathbf{x} \in \{0,1\}^n. \qquad (8)$$

The solution $\mathbf{x}$ of this formulation is not unique, but it ensures exactly minimum number of parts. Although with different objective functions, we can prove that our formulation in Eq.4 can obtain the same minimum number of parts as Eq.8 if λ is selected appropriately. Theorem 1 tells the relationship between Eq.8 and our formulation in Eq.4:

Theorem 1 Minimum Decomposition Rule

We consider two objective functions as follows:

$$\begin{cases} f(\mathbf{x}) = \| \mathbf{x} \|_0 + \lambda\mathbf{w}^\top\mathbf{x}, & s.t.\ \mathbf{Ax} \geq \mathbf{1},\ \mathbf{Bx} \leq \mathbf{1},\ \mathbf{x} \in \{0,\ 1\}^n, \\ g(\mathbf{x}) = \| \mathbf{x} \|_0, & s.t.\ \mathbf{Ax} \geq \mathbf{1},\ \mathbf{Bx} \leq \mathbf{1},\ \mathbf{x} \in \{0,\ 1\}^n, \end{cases}$$

Let:

$$\mathbf{x}' = \arg\min_{\mathbf{x}} f(\mathbf{x}), \qquad \mathbf{x}'' = \arg\min_{\mathbf{x}} g(\mathbf{x}).$$

We have $\| \mathbf{x}' \|_0 = \| \mathbf{x}'' \|_0$ *when* $0 \leq \lambda \leq 1/\sum_{i=1}^{n} w_i$.

$\mathbf{x}''$ is the solution of Eq.8 whose zero-norm is minimized, and $\mathbf{x}'$ is the solution of our formulation in Eq.4. Therefore, our formulation can decompose a shape into minimum number of parts when $0 \leq \lambda \leq 1/\sum_{i=1}^{n} w_i$. It is worth mentioning that although Eq.4 and Eq.8 both minimize the number of parts, their cuts are not necessarily the same subset from $C(S)$, since Eq.4 favors visually more natural cuts.

Proof of Theorem 1

In order to prove $\| \mathbf{x}' \|_0 = \| \mathbf{x}'' \|_0$, when $0 \leq \lambda \leq 1/\Sigma_{i=1}^{n} w_i$, first we have:

$$\min_{\mathbf{x}} f(\mathbf{x}) = \| \mathbf{x}' \|_0 + \lambda\mathbf{w}^\top\mathbf{x}' \leq \| \mathbf{x}'' \|_0 + \lambda\mathbf{w}^\top\mathbf{x}'', \qquad (9)$$

$$\min_{\mathbf{x}} g(\mathbf{x}) = \| \mathbf{x}'' \|_0 \leq \| \mathbf{x}' \|_0, \qquad (10)$$

As $w_i > 0$, so when $0 \leq \lambda \leq 1/\Sigma_{i=1}^{n} w_i$, $\forall\ \mathbf{x} \in \{0,\ 1\}^n$, $0 \leq \lambda\mathbf{w}^\top\mathbf{x} \leq 1$. Therefore, from Eq.9 we further have Eq.11, and from Eq.10 we further have Eq.12:

$$\| \mathbf{x}' \|_0 + \lambda\mathbf{w}^\top\mathbf{x}' \leq \| \mathbf{x}'' \|_0 + 1. \qquad (11)$$

$$\| \mathbf{x}'' \|_0 \leq \| \mathbf{x}' \|_0 + \lambda\mathbf{w}^\top\mathbf{x}'. \qquad (12)$$

Combining Eq.11 and Eq.12, we have:

$$\| \mathbf{x}'' \|_0 \leq \| \mathbf{x}' \|_0 + \lambda\mathbf{w}^\top\mathbf{x}' \leq \| \mathbf{x}'' \|_0 + 1.$$

As $0 \leq \lambda\mathbf{w}^\top\mathbf{x}' \leq 1$, and $\| \mathbf{x}' \|_0$, $\| \mathbf{x}'' \|_0$ are integers, thus $\| \mathbf{x}' \|_0 = \| \mathbf{x}'' \|_0$ when $0 \leq \lambda \leq 1/\Sigma_{i=1}^{n} w_i$.

	ACD [7]	CSD [8]	MNCD
Objective	a NCD without optimization	a NCD with the minimum length of cuts	a NCD with minimum number of parts and high visual naturalness
Candidate cut set	complete set of all possible cuts	incomplete set from Reeb graph	complete set of all possible cuts
Perception rules	minima rule and short cut rule	short cut rule	minima rule and short cut rule
Constraints	non-overlapping constraint convexity constraint	convexity constraint	non-overlapping constraint convexity constraint
Solution	greedy algorithm	Linear Programming	Binary Integer Linear Programming

TABLE 1
The comparison among ACD, CSD and MNCD, where NCD denotes near-convex decomposition.

```
Algorithm 1: MNCD($S$, $\psi$)
Input: A shape, $S$, and a concavity tolerance, $\psi$;
Output: $\psi$-MNCD of $S$, $\{P_i\}$.
◇ compute the candidate cut set, $C(S)$;
◇ shrink the candidate cut set, $C(S)$;
◇ compute $\psi$-mutex set of $S \rightarrow M^{\psi}(S)$;
foreach $mp_i$ in $M^{\psi}(S)$ do
    foreach $cut_j$ in $C(S)$ do
        check whether $cut_j$ separates $mp_i \rightarrow a_{ij}$;
foreach $cut_i$ in $C(S)$ do
    compute its cost $\rightarrow w_i$;
    foreach $cut_j$ in $C(S)$ do
        check whether $cut_i$ intersects with $cut_j \rightarrow b_{gz}$;
◇ obtain the optimized solution by solving Eq.13$\rightarrow \{P_i\}$.
```

4.2 Binary Integer Linear Programming

We can optimize our formulation in Eq.4 efficiently via Binary Integer Linear Programming. Now we prove Eq.4 is a Binary Integer Linear Programming problem. Note that the vector $\mathbf{x}$ is binary, thus the objective function in Eq.4 can be expressed as a linear form $\|\mathbf{x}\|_0$+$\lambda\mathbf{w}^\top\mathbf{x}$ = $(\mathbf{1}^\top$+λ $\mathbf{w}^\top)\mathbf{x}$, where $\mathbf{1}_{n\times1}$=(1, ..., 1)$^\top$.

We define a matrix $\mathbf{D}_{(m+t)\times n}$ as $\mathbf{D} = \left[\begin{array}{c}-\mathbf{A}\\ \hdashline \mathbf{B}\end{array}\right]$, and a vector $\mathbf{u}_{(m+t)\times 1}$ as $\mathbf{u} = \left[\begin{array}{c}-\mathbf{1}\\ \hdashline \mathbf{1}\end{array}\right]$, where $-\mathbf{1}_{m\times1}$=$(-1$, ..., $-1)^\top$ and $\mathbf{1}_{t\times1}$=(1, ..., 1)$^\top$, thus Eq. 4 can be represented as a Binary Integer Linear Programming problem:

$$\begin{aligned}&\mathbf{min} \quad (\mathbf{1}^\top + \lambda\mathbf{w}^\top)\mathbf{x}\ ,\\ &s.t. \quad \mathbf{D}\mathbf{x} \le \mathbf{u}, \quad \mathbf{x} \in \{0,1\}^n.\end{aligned} \tag{13}$$

We can solve this linear programming problem efficiently using standard discrete optimization techniques, such as CPLEX, Lingo, or integer relaxation. We use CPLEX in our experiments.

4.3 Implemental Details and Time Complexity

Algorithm 1 shows the overall procedure of our method. In the implementation of computing the $C(S)$, to save the memory, we shrink the candidate cut set in the first stage. Specifically, we discard the cuts whose endpoints are both convex vertices, since they cannot separate mutex pairs apart.

To compute $C(S)$, we consider all pairs of vertices on the contour. Thus the time complexity of computing $C(S)$ is $O(v^2)$, where v is the number of vertices. According to [8], the time complexity of computing $M^{\psi}(S)$ is $O(Tvr)$, where r is the number of notches in the shape, and T is the number of Morse functions we used to compute $M^{\psi}(S)$, which we set as 16 in our experiments. With $C(S)$ and $M^{\psi}(S)$, we can obtain the value of matrixes A in $O(mn)$ time and matrix B in $O(n^2)$ time, where m is the number of mutex pairs in $M^{\psi}(S)$ and n is the number of candidate cuts in $C(S)$, where $n \gg m, T, v$, and r. Thus the total time complexity of our algorithm is $O(v^2 + Tvr + mn + n^2)$=$O(n^2)$.

4.4 Comparison with Other Methods

Table 1 presents a comparison between MNCD and the state-of-the-art methods: ACD [7] and CSD [8]. Our method aims at the minimum number of parts with high degree of visual naturalness for robust shape representation.

Specifically, CSD is a special case of our formulation in Eq.4 if discarding the $\|\mathbf{x}\|_0$ term and setting $\beta = 0$ in Eq.5. The $\|\mathbf{x}\|_0$ term in our formulation guarantees the minimum number of decomposed parts, which eliminates all the redundant parts in near-convex decomposition. This term is essential for robust shape representation and can improve the efficiency of further processes, as shown in Fig.1. Parameter β in Eq.5 imposes the minima rule and short cut rule on our near-convex decomposition scheme. Setting $\beta = 0$ means discarding the minima rule. This term is essential as well because these two perception rules are introduced for high degree of visual naturalness which guarantees better recognition primitives. And the minima rule can inhibit cuts at positions with small negative curvatures or even at convex points.

5 EXPERIMENTS

In order to evaluate our shape decomposition method MNCD on 2D shapes, we test the MPEG-7 shape dataset [25], the Animal dataset [26], and the NTU-Microsoft Kinect HandGesture dataset [14]. Excluding simple shapes such as the heart shape that can be easily decomposed, we select 20 complex shape categories from the MPEG-7 dataset, in which each category has 20 shapes (20×20 = 400 shapes), 8 complex shape categories from the Animal dataset, in which each category has 100 shapes (8×100 = 800 shapes), and 9 complex hand shape categories from the NTU-Microsoft Kinect HandGesture dataset, in which each category has 100 shapes (9×100 = 900 shapes). Fig.4 shows an example for each selected category.

MPEG-7 dataset	ψ=0.005R		ψ=0.01R		ψ=0.03R		ψ=0.06R	
	ACD↓	CSD↓	ACD↓	CSD↓	ACD↓	CSD↓	ACD↓	CSD↓
bat	14.3%	8.9%	20.8%	11.3%	16.2%	6.8%	8.6%	6.5%
beetle	23.8%	10.3%	22.9%	9.0%	21.9%	16.0%	19.3%	14.4%
bird	18.5%	13.6%	23.8%	12.5%	12.8%	7.6%	17.4%	10.6%
butterfly	4.4%	5.8%	13.1%	7.2%	16.9%	8.8%	32.7%	12.9%
camel	16.1%	10.5%	15.2%	3.3%	21.1%	9.5%	21.3%	4.8%
carriage	5.5%	3.7%	13.8%	9.2%	15.6%	9.5%	18.4%	13.3%
cattle	24.9%	14.6%	24.5%	10.7%	27.4%	8.9%	23.0%	12.3%
chicken	19.0%	10.0%	23.1%	15.2%	24.0%	10.5%	3.1%	5.2%
chopper	8.9%	7.7%	16.2%	10.4%	22.1%	10.7%	17.4%	11.3%
crown	16.0%	9.2%	20.7%	11.9%	27.8%	14.6%	19.4%	16.7%
deer	18.0%	14.5%	24.2%	10.5%	15.3%	4.2%	22.6%	13.3%
dog	23.8%	15.4%	18.8%	7.6%	24.5%	9.2%	15.7%	10.5%
elephant	24.1%	12.0%	24.0%	8.9%	24.9%	9.7%	25.2%	7.8%
fly	11.9%	9.2%	8.9%	5.6%	4.2%	3.9%	10.6%	8.4%
horse	20.1%	8.0%	23.8%	5.1%	19.8%	1.1%	18.8%	6.1%
horseshoe	26.1%	18.6%	21.9%	11.7%	23.5%	14.8%	12.2%	12.2%
lizard	18.2%	10.4%	15.9%	10.0%	27.5%	15.2%	11.7%	7.3%
Misk	29.8%	30.7%	24.2%	11.9%	25.8%	20.3%	13.2%	15.4%
Mickey	24.6%	13.4%	14.0%	10.5%	19.8%	12.9%	17.3%	8.5%
spring	22.6%	12.6%	25.1%	13.7%	24.5%	15.8%	25.7%	6.9%

TABLE 2
The average reduction rate of MNCD comparing with ACD [7] and CSD [8], on the MPEG-7 dataset, where R is the radius of the shape's minimum enclosing disk.

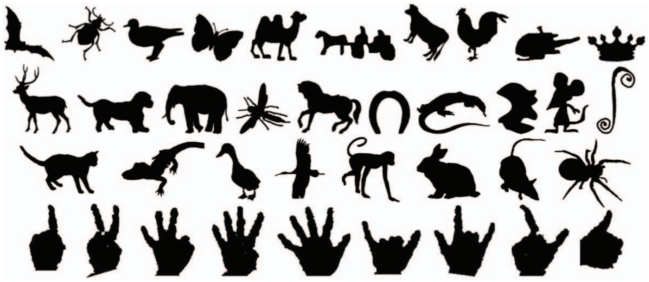


Fig. 4. An example of each shape category selected from the MPEG-7 dataset [25] (the first two rows), the Animal dataset [26] (the third row), and the NTU-Microsoft Kinect HandGesture dataset [14] (the last row) is displayed.

5.1 Evaluation of Parameters

In our algorithm, there are 2 parameters, ψ and λ, where ψ is the user specified concavity tolerance for near-convex decomposition, and λ is the parameter introducing the visual naturalness term.

The parameter ψ tells how small degree of concave features the user wants to ignore in near-convex decomposition. In Fig.5, it shows the average number of decomposed parts using our method at 4 different values of ψ, 0.005R, 0.01R, 0.03R, and 0.06R respectively, on the 1200 shapes from datasets [25] [26], where R is the radius of the shape's minimum enclosing disk. As we see, when we increase the value of concavity tolerance ψ, the average number of decomposed parts decreases. A small ψ means that the decomposed parts are almost strictly convex, which will introduce a large number of small parts to ensure the convexity constraint, thus is not robust to local distortions. When ψ increases, the decomposition can tolerate more severe distortions. Fig.6 shows the decomposition results at 4 different values of ψ.

Animal dataset	ψ=0.005R		ψ=0.01R		ψ=0.03R		ψ=0.06R	
	ACD↓	CSD↓	ACD↓	CSD↓	ACD↓	CSD↓	ACD↓	CSD↓
cat	16.7%	8.7%	24.4%	12.6%	21.9%	10.2%	22.8%	11.1%
crocodile	15.7%	14.4%	22.3%	15.9%	21.7%	11.2%	22.5%	5.9%
duck	21.1%	12.5%	25.2%	13.2%	19.0%	6.4%	15.1%	8.0%
flyingbird	7.1%	3.7%	13.2%	7.9%	21.1%	11.8%	15.6%	8.2%
monkey	18.5%	9.9%	25.5%	4.5%	21.2%	12.7%	30.5%	7.0%
rabbit	24.0%	9.4%	23.9%	6.3%	20.0%	4.9%	18.0%	11.1%
rat	24.2%	7.3%	23.5%	10.7%	12.2%	8.6%	15.3%	12.9%
spider	11.4%	5.1%	11.3%	6.2%	4.7%	5.2%	3.1%	4.1%

TABLE 3
The average reduction rate of MNCD comparing with ACD [7] and CSD [8], on the Animal dataset, where R is the radius of the shape's minimum enclosing disk.

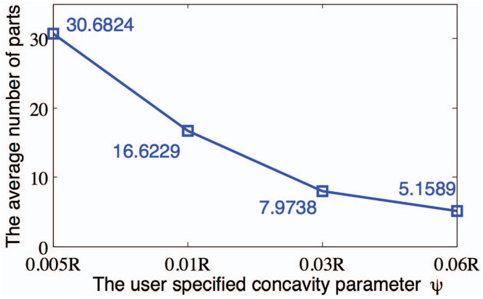


Fig. 5. The average number of parts decomposed by MNCD with different concavity tolerances.

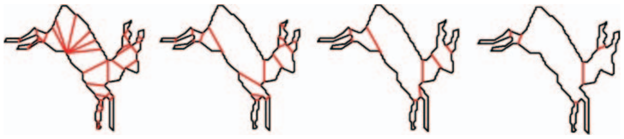


Fig. 6. The decomposition results of MNCD when λ=1/$\sum_{i=1}^{n} w_i$, with ψ= 0.005R, ψ=0.01R, ψ=0.03R, and ψ=0.06R, from left to right, respectively.

The parameter λ introduces the visual naturalness term to the decomposition in Eq.4. Fig.7 shows the decomposition results of MNCD at 3 different values of λ. If $0 \leq \lambda \leq 1/\sum_{i=1}^{n} w_i$, the number of parts by MNCD is minimized. But a larger λ leads to a more natural decomposition since it counts more weight of the visual naturalness regularization term in Eq.4. In our experiments below, we set λ as the upper bound $1/\sum_{i=1}^{n} w_i$.

5.2 Evaluation of the Number of Parts

One advantage of our method is that it does not introduce redundant part as it decomposes the shape into minimum number of parts. In terms of the number of parts, table 2 presents the *average reduction rate* comparing our method with ACD [7] and CSD [8] at 4 different ψ, on the MPEG-7 dataset and the Animal dataset respectively. The average reduction rate scores are defined as:

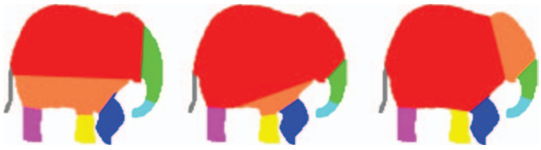


Fig. 7. The decomposition results of MNCD when ψ=0.03R, with λ= 0, λ=0.5/$\sum_{i=1}^{n} w_i$, λ=1/$\sum_{i=1}^{n} w_i$, from left to right, respectively.

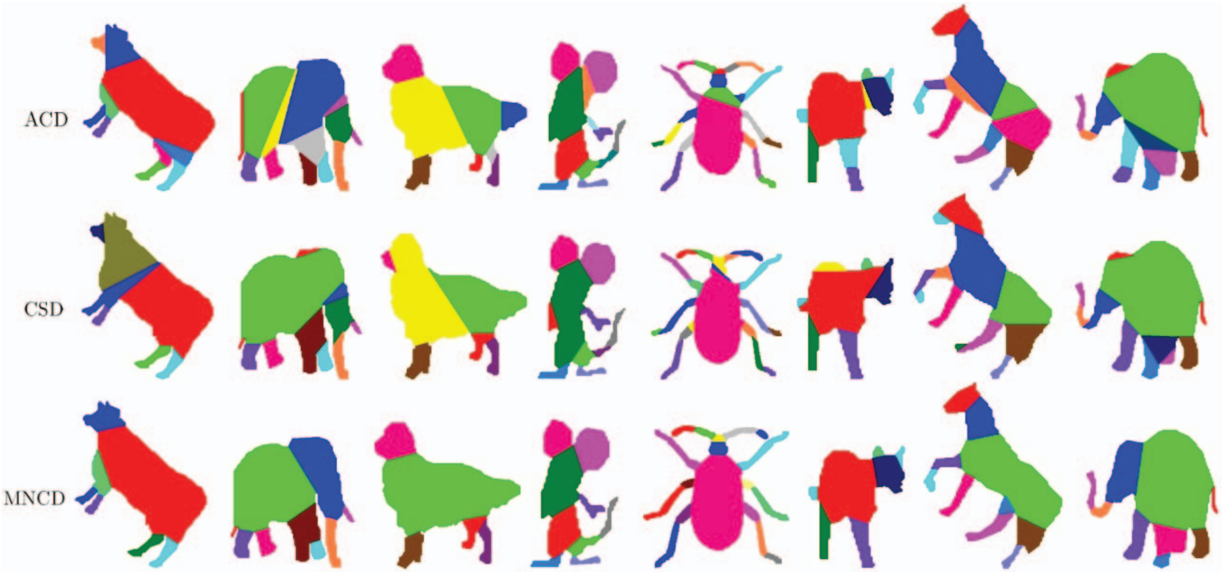


Fig. 8. Some decomposition results of ACD [7], CSD [8] and MNCD. Our MNCD method produces the least number of near-convex parts and our decompositions are visually more natural.

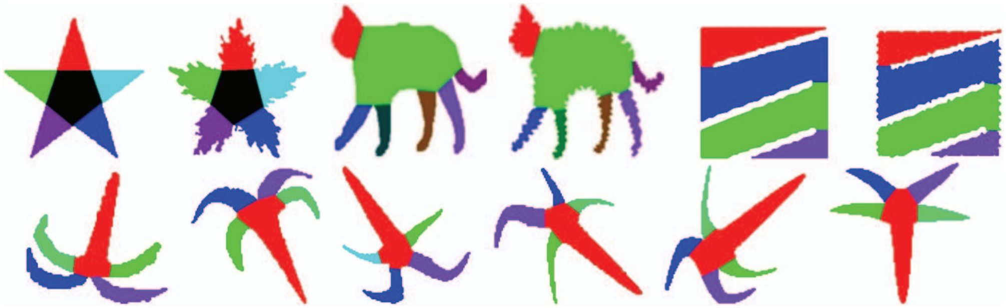


Fig. 9. The robust decomposition results of MNCD. The first row is the results of shapes with local distortions; and the second row is the results of shapes with deformation. Without introducing redundant parts and by considering perception rules, MNCD is robust to local distortions and shape deformation.

$$\text{ACD}\downarrow = (\#\text{ACD} - \#\text{MNCD})/\#\text{ACD}\ ,$$
$$\text{CSD}\downarrow = (\#\text{CSD} - \#\text{MNCD})/\#\text{CSD}\ .$$

As it shows, we produce the least number of parts. Comparing with ACD [7], up to 32.7% of redundant parts are eliminated, and up to 30.7% of redundant parts are eliminated compared with CSD [8]. On average, 18.99% of parts are eliminated compared with ACD and 10.15% compared with CSD. Thus, the efficiency of further applications on the decomposed parts can be highly improved. On the other hand, from the table, we notice that all the ACD↓ and CSD↓ scores are greater than 0 on every shape category and every ψ, which means that MNCD always produces minimum number of parts, as proved in Theorem 1.

5.3 Decomposition Results

To evaluate the visual naturalness of our decomposition, Fig.10 compares our method with the method proposed by Mi and Decarlo [27]. Note that Mi's method is specifically designed to decompose 2D shapes into natural parts. The first row are the decomposition results of their method, and the second row are the results of MNCD. As we can see, when considering the minima rule and short cut rule in our formulation, our method decomposes shapes into parts with high degree of visual naturalness comparable to the method of [27], such as the legs, head and body of the animal, the leaf and stem of the tree, etc.

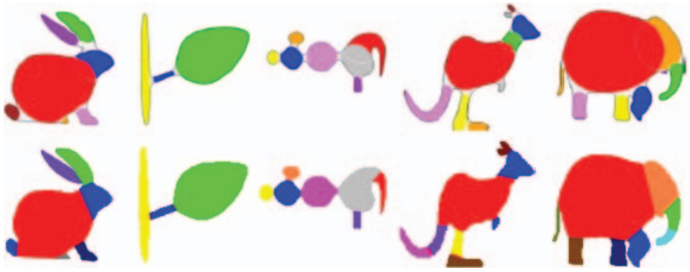


Fig. 10. The first row shows the decomposition results of [27], and the second row shows the results of MNCD.

In Fig.8, more comparisons among ACD [7], CSD [8] and our method are provided, with ψ=0.03R. The decompositions of our method produce the minimum and most natural recognition primitives. At this concavity tolerance, MNCD decomposes the animals into primitives such as head, body, legs and tail, and avoids decomposing them into redundant parts as in [7], [8].

Without introducing redundant parts, MNCD is robust to local distortions, as shown in the first row of Fig.9. The robustness of our method is demonstrated when there are large local distortions as shown in the last row of Fig.1, where the existing decomposition methods produce many redundant noisy parts. Besides, our MNCD imposes two perception rules to guide the decomposition, thus it produces parts with high degree of visual naturalness, which makes MNCD robust to shape deformation, as illustrated in the second row of Fig.9.

It is worth noting that in our method, the parameter ψ is the user specified concavity tolerance. The optimal value of ψ is application dependant.

5.4 Hand Shape Decomposition and Its Application

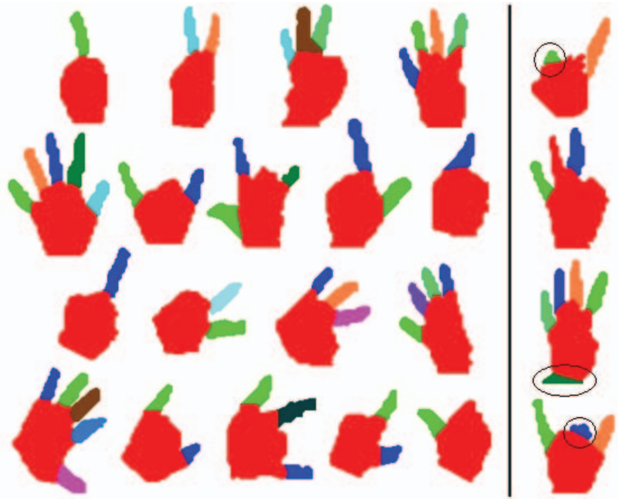


Fig. 11. The MNCD representations of hand shapes.

In Section 5.3, we illustrate the robustness of our MNCD algorithm to the local distortions and shape deformation. In order to further validate that MNCD can provide a robust shape representation, we test our method on hand shapes from the NTU-Microsoft Kinect HandGesture dataset [14].

In Fig.11, the decomposition results of hand shapes are shown, with ψ=0.06R. In the first two rows and the last two rows on the left side of the black line, we show two sets of decomposition results. As we see, our algorithm can represent the hands as finger parts and palm part consistently. The shapes in the last two rows have orientation, articulation or scale changes, and the results demonstrate the robustness of MNCD to these changes. The rightmost column on the right side of the black line are imperfect results. Specifically, the second hand shape does not decompose the ring finger out, and the rest three shapes produce one "redundant" part as shown in the ellipse. These are due to the convexity constraint. The algorithm needs to produce the part in the ellipse to satisfy the convexity constraint. And for the second shape, the ring finger and the palm make up a part that can satisfy the convexity constraint, thus it is not separated. As we see, MNCD leads to robust shape representations.

	MNCD	Our previous work [1]
Decomposition time / gesture	3.9705s	45.6132s

TABLE 4
The average computational cost of shape decomposition using the proposed MNCD and our previous work [1] on the NTU-Microsoft Kinect HandGesture dataset [14].

Based on this representation method, we have validated its superiority in the application of hand gesture recognition using Kinect sensor [14] [15], which achieves the state-of-the-art recognition accuracy, as demonstrated in [28].

In Section 4.2, we formulate MNCD as a Binary Integer Linear Programming problem. Compared with the quadratic programming formulation of our previous work [1], such a reformulation of BILP leads to significantly faster solution, which is validated by the comparison of computational costs on the NTU-Microsoft HandGesture dataset [14] shown in Table 4. The computational cost in Table 4 is the average decomposition time per hand shape, tested over all the 900 hand shapes in dataset [14].

6 Conclusion

In this paper, we present a novel near-convex shape decomposition method for robust shape representation, Minimum Near-Convex Decomposition (MNCD), which decomposes 2D shapes into minimum number of parts with high degree of visual naturalness. We formulate the shape decomposition problem as a Binary Integer Linear Programming problem, which has been theoretically proved and experimentally validated that can efficiently decompose shapes into minimum number of near-convex parts. Experiments on complex 2D shape datasets show that the proposed method is robust to shape distortion and deformation. Besides, the robustness of our decomposition is demonstrated in the representation of hand shapes in Section 5.4 and in the application of hand gesture recognition [14] [15] [28].

7 Acknowledgement

This work was supported in part by the Nanyang Assistant Professorship (SUG M58040015) to Dr. Junsong Yuan and the National Natural Science Foundation of China (grant No. 61173120) to Dr. Wenyu Liu.

References

[1] Z. Ren, J. Yuan, C. Li, and W. Liu, "Minimum near-convex decomposition for robust shape representation," in *Proceedings of IEEE International Conference on Computer Vision*, 2011, pp. 303–310. 1, 2, 3, 8

[2] K. Siddiqi, K. Tresness, and B. B. Kimia, "Parts of visual form: Psychophysical aspects," *Perception*, vol. 25, pp. 399–424, 1996. 1

[3] I. Biederman, "Recognition-by-components: A theory of human image understanding," *Psychological Review*, vol. 94, pp. 115–147, 1987. 1, 2

[4] M. Singh and D. D. Hoffman, "Part-based representations of visual shape and implications for visual cognition," *Advances in Psychology*, vol. 130, pp. 401–459, 2001. 1

[5] T. A. Cass, "Robust affine structure matching for 3d object recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 1265 – 1274, 1998. 1

[6] G. Lu and A. Sajjanhar, "Region-based shape representation and similarity measure suitable for content-based image retrieval," *Multimeida Systems*, vol. 7, pp. 165 – 174, 1999. 1

[7] J.-M. Lien and N. Amato, "Approximate convex decomposition of polygons," *Computational Geometry*, vol. 35, pp. 100–123, 2006. 1, 2, 5, 6, 7, 8

[8] H. Liu, L. J. Latecki, and W. Liu, "Convex shape decomposition," in *Proceedings of IEEE conference on computer vision and pattern recognition*, 2010, pp. 104–124. 1, 2, 3, 4, 5, 6, 7, 8

[9] J. M. Keil and J. Snoeyink, "On the time bound for convex decomposition of simple polygons," *International Journal of Computational Geometry and Application*, vol. 12, pp. 181–192, 2002. 1, 2

[10] ——, *Minimum Convex Decomposition*, 2007, www.cs.ubc.ca/∼snoeyink/demos/convdecomp. 1, 2

[11] J.-M. Lien and N. Amato, "Approximate convex decomposition of polyhedra," in *Proceedings of ACM Symposium on Solid and Physical Modeling*, 2007, pp. 121–131. 1, 2

[12] D. D. Hoffman and M. Singh, "Salience of visual parts," *Cognition*, vol. 14, pp. 29–78, 1997. 2, 3

[13] M. Singh, G. Seyranian, and D. D. Hoffman, "Parsing silhouettes: The short-cut rule," *Percept and Psychophys*, vol. 61, pp. 636–660, 1999. 2, 3

[14] Z. Ren, J. Yuan, and Z. Zhang, "Robust hand gesture recognition based on finger-earth mover's distance with a commodity depth camera," in *Proceedings of ACM International Conference on MultiMedia*, Scottsdale, Arizona, USA, 2011, pp. 1093–1096. 2, 5, 6, 8

[15] Z. Ren, J. Yuan, J. Meng, and Z. Zhang, "Robust part-based hand gesture recognition using kinect sensor," *IEEE Transactions on Multimedia*, 2013, in press. 2, 8

[16] D. Macrini, S. Dickinson, D. Fleet, and K. Siddiqi, "Bone graphs: Medial shape parsing and abstraction," *Computer Vision and Image Understanding*, vol. 35, pp. 1044–1061, 2011. 2

[17] D. Macrini, C. Whiten, R. Laganiere, and M. Greenspan, "Probabilistic shape parsing for view-based object recognition," in *Proceedings of the International Conference on Pattern Recognition*, Tsukuba, Japan, 2012, pp. 2302–2305. 2

[18] E. Zuckerberger, A. Tal, and S. Shlafman, "Polyhydral surface decomposition with applications," *Computers & Graphics*, vol. 26, pp. 733–743, 2002. 2

[19] M. Singh and D. D. Hoffman, "Completing visual contours: The relationship between relatability and minimizing inflections," *Perception & Psychophysics*, vol. 61, pp. 943–951, 1999. 2

[20] S. Katz and A. Tal, "Hierarchical mesh decomposition using fuzzing clustering and cuts," *ACM Transactions on Graphics*, vol. 2, pp. 954–961, 2003. 2

[21] F. Aurenhammer, "Weighted skeletons and fixed-share decomposition," *Computational Geometry*, vol. 40, no. 2, pp. 93–101, 2008. 2

[22] M. Tanase and R. C. Veltkamp, "Polygon decomposition based on the straight line skeleton," in *Proceedings of the Symposium on Computational Geometry (SoCG)*, 2003, pp. 58–67. 2

[23] J.-M. Lien, "Hybrid motion planning using minkowski sums," in *Proceedings of Robotics: Science and System V*, 2008. 2

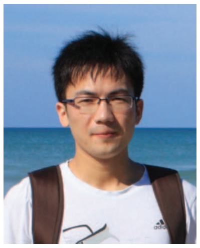
[24] J.-M. Lien and Y. Lu, "Planning motion in similar environments," in *Proceedings of Robotics: Science and System V*, 2009. 2

[25] L. J. Latecki, R. Lakamper, and U. Eckhardt, "Shape descriptors for non-rigid shapes with a single closed contour," in *Proceedings of IEEE conference on computer vision and pattern recognition*, 2000, pp. 424–429. 5, 6

[26] X. Bai, W. Liu, and Z. Tu, "Integrating contour and skeleton for shape classification," in *Proc. of IEEE Workshop on NORDIA (in conjunction with ICCV)*, 2009. 5, 6

[27] X. Mi and D. Decarlo, "Separating parts from 2d shapes using relatability," in *Proceedings of IEEE International Conference on Computer Vision*, 2007, pp. 1–8. 7

[28] Z. Ren, J. Meng, J. Yuan, and Z. Zhang, "Robust hand gesture recognition with kinect sensor," in *Proceedings of ACM International Conference on MultiMedia*, Scottsdale, Arizona, USA, 2011, pp. 759–760. 8



Zhou Ren is currently a PhD student in Department of Computer Science at University of California, Los Angeles. He received the M.Eng degree with award from the School of Electical and Electronic Engineering at Nanyang Technological University, Singapore, in 2012. Before that, he received the B.Eng with highest honor from the Department of Electronics and Information Engineering in Huazhong University of Science and Technology, China, in 2010.

From 2010 to 2012, he was a project officer in the Media Technology Lab at Nanyang Technological University. His research interests include computer vision, human-machine interaction, and statistical machine learning.

Junsong Yuan is currently a Nanyang Assistant Professor at Nanyang Technological University. He received his Ph.D. from Northwestern University, USA and his M.Eng. from National University of Singapore, in 2009 and 2005 respectively. Before that, he graduated from the special program for the gifted young in Huazhong University of Science and Technology, China, in 2002. He was a research intern at Microsoft Research Redmond, Kodak Research Laboratories, Rochester, and Motorola Applied Research Center, Schaumburg, USA. His current research interests include computer vision, video analysis, multimedia search and mining, vision-based human computer interaction, biomedical image analysis, etc. He is the program director of Video Analytics in the Infocomm Center of Excellence at Nanyang Technological University.

Junsong Yuan received the Outstanding Ph.D. Thesis award from the EECS department in Northwestern University, and the Doctoral Spotlight Award from IEEE Conf. Computer Vision and Pattern Recognition Conference (CVPR09). He was also a recipient of the elite Nanyang Assistant Professorship in 2009. He has filed 3 US patents and is a member of IEEE.

Wenyu Liu received the BS degree in computer science from Tsinghua University, Beijing, China, in 1986, and the Diploma and Doctoral degrees, both in electronics and information engineering, from Huazhong University of Science and Technology (HUST), Wuhan, China, in 1991 and 2001, respectively. He is now a professor and associate chairman of the Department of Electronics and Information Engineering, HUST. His current research areas include computer graphics, multimedia information processing, and computer vision. He is a member of IEEE.