

ラスタ図形詰込み問題に対する 局所探索法の特徴点検出を用いた効率化

大阪大学・大学院情報科学研究科 梅谷 俊治, 村上 祥平, 森田 浩

Shunji Umetani, Shohei Murakami and Hiroshi Morita

Graduate School of Information Science and Technology, Osaka University

1 はじめに

詰込み問題とは、いくつかの図形を互いに重ならないように与えられた領域に配置する問題であり、図形の次元や形状により多くのバリエーションを持つ。図形の表現方法はベクタ形式とラスタ形式の2種類が知られている。点と線分で表されるベクタ図形の詰込み問題では、ミンコフスキ差と呼ばれる図形対のたみ込み演算を用いた効率的な重なり度計算に基づくアルゴリズムが多く開発されたが、数値誤差による誤判定の回避や多様な曲線の交差判定などの幾何計算に対応する複雑な処理の実装が実用化の壁になっていた。一方で、ピクセルの集合で表されるラスタ図形は全ての幾何計算を簡単な処理で実装できるものの計算時間が図形の面積に比例するため、高解像のラスタ図形の詰込み問題に対する効率的なアルゴリズムは実現されていない。

これまで、著者らはラスタ表現された図形の詰込み問題に対して、図形対の重なり度の効率的な計算と1次元探索に基づく局所探索法を提案し、数値実験を通じて提案手法が実用に耐える性能を持つことを示した [1]。本研究では、画像処理における特徴点抽出の手法を用いて、局所探索法の基本操作となる1次元探索の効率化を実現し、高解像なラスタ図形の詰込み問題に対する効率的なアルゴリズムを実現する。

2 ストリップパッキング問題

入力として、幅 W (固定) と長さ L (可変) の大きさを持つ長方形と n 個のラスタ図形 P_1, \dots, P_n が与えられる。このとき、(1) 図形 $P_i (1 \leq i \leq n)$ は長方形 $C(W, L)$ 内に配置される、(2) 図形対 $P_i, P_j (1 \leq i < j \leq n)$ は互いに重ならない、という2つの条件の下で必要な長方形の長さ L を最小化する図形 $P_i (1 \leq i \leq n)$ の配置を求める。ただし、図形の自由な回転や反転は考えない。

ピクセルの集合で表されるラスタ図形は、図形対の重なり判定にかかる計算量が解像度に依存する欠点を持つ。そこで、本研究では、ラスタ図形を走査して有色のピクセル区間の始点と終点の座標を保持するスキャンライン形式に変換する。また、図形対の重なり判定の計算を効率化するために、水平・垂直方向に走査して2種類のスキャンライン形式で表現された集合を作成する (図1)。

3 重なり度最小化問題

図形の重なりがない解のみを探索して長方形の長さ L を直接に最小化する方法では、充填率の高い解になると1つの図形だけを動かして重なりのない解を求めることは困難となる。そこで、探

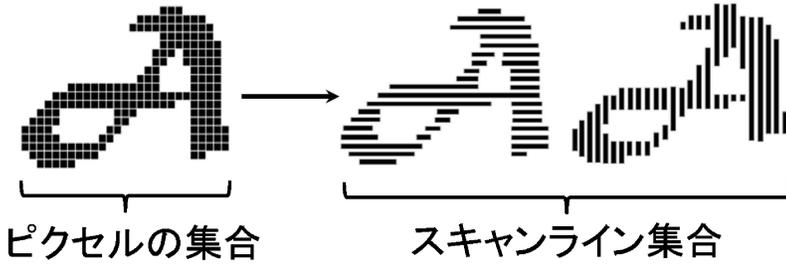


図 1: 水平・垂直方向のスキャンラインの集合

索途中では図形の重なりを許容し、長さ L を一時的に固定することで図形対の重なり度の総和を最小化する重なり度最小化問題を考える。長さ L を変更しては重なり度最小化問題解く手続きを繰り返すことでストリップパッキング問題を解く [2]。

図形 P_i, P_j の重なりを解消するために必要な水平・垂直方向の最小移動距離をそれぞれ f_{ij}^h, f_{ij}^v とする。本研究では、図形 P_i, P_j の重なり度を $f_{ij} = \min\{f_{ij}^h, f_{ij}^v\}$ と定義し、重なり度最小化問題ではそれらの総和 $F = \sum_{i,j} f_{ij}$ を最小化する。

4 ミンコフスキー差と重なり度の計算

本研究では、ミンコフスキー差を用いて図形対の重なりを判定する。図形 P_i, P_j に対して参照点を定め、 P_i の参照点 r_i が原点に一致するように配置した際に、 P_i と P_j が重なりを持つ P_j の参照点 r_j の配置全体を P_j の P_i に対するミンコフスキー差と呼び、 $P_i \ominus P_j$ と表す (図 2)。 P_j の参照点 r_j をミンコフスキー差 $P_i \ominus P_j$ の外部に置けば P_i と P_j が重ならないように配置できる。

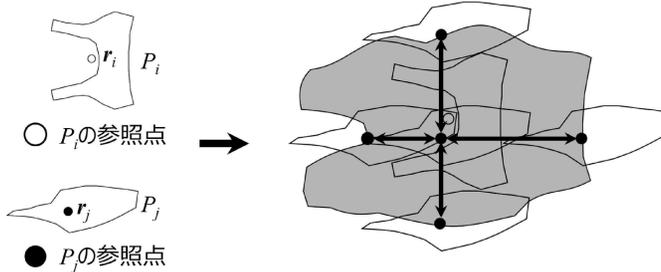


図 2: ミンコフスキー差と重なり度の計算

図形 P_i, P_j はスキャンラインの集合で表現されるため、ミンコフスキー差 $P_i \ominus P_j$ もスキャンラインの集合で表現される。水平方向のスキャンラインの集合で表されたミンコフスキー差において、図形 P_i, P_j の参照点の相対座標 $r_j - r_i$ からスキャンラインの境界までの最小距離が f_{ij}^h となる (図 3)。同じ方法で f_{ij}^v も得られる。

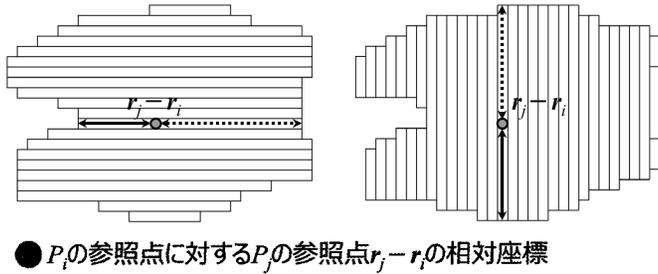


図 3: スキャンラインの集合で表現されたミンコフスキー差による f_{ij}^h, f_{ij}^v の計算

5 特徴点抽出を用いた探索の効率化

本研究では、1つの図形を水平または垂直方向に移動させて得られる任意の配置からなる集合を近傍と定義し、1次元探索により近傍内で重なり度が改善する配置が見つければ図形を移動させる手続きを基本操作とする局所探索法を提案する。

文献 [1] では、図形を1ピクセルずつ動かして重なり度が最小となる配置を求めるため、1次元探索の探索空間は長方形の幅 W と長さ L に比例する。本研究では、前処理で図形対 P_i, P_j のミンコフスキー差 $P_i \ominus P_j$ にコーナー検出アルゴリズム [3] を適用し、重なり度が最小となる可能性の高い配置のみを評価することで1次元探索の効率化を実現する (図 4)。

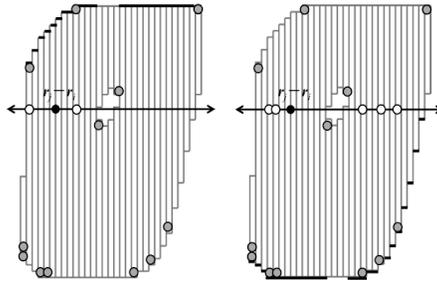


図 4: 特徴点抽出による1次元探索の効率化

6 数値実験

本研究では、多角形および円弧や穴を含む図形のストリップパッキング問題に対するベンチマーク問題例 [4, 2] を用いて数値実験を実施した。これらのベンチマーク問題例の概要を表 1, 表 2 に示す。ベンチマーク問題例のデータ形式はベクタ図形であるため、長方形領域 C の幅 $W=128, 256, 512, 1024, 2048$ ピクセルとして、各図形をラスタ形式に変換した。

本研究では、提案手法の実験結果を Burke ら [4], Umetani ら [2] の結果と比較する。Burke ら [4] は円弧を含むベクタ図形に対する詰込みアルゴリズムを提案し、表 1, 表 2 のベンチマーク問

題例に対する数値実験を実施した。また, Umetani ら [2] は多角形に対する詰込みアルゴリズムを提案し, 表 1 のベンチマーク問題例に対する数値実験を実施した。

数値実験の計算機環境と計算時間を表 3, 表 4 に示す。

表 5, 表 6 は, 多角形のベンチマーク問題例と円弧や穴を含む図形のベンチマーク問題例に提案手法をそれぞれ 10 回適用して得られた配置の充填率の平均値を示す。ここで, 充填率は $\sum_{i=1}^n (P_i \text{ の面積}) / WL$ とする。表 5, 表 6 より, 図形の解像度の増加にともなう配置の充填率の低下が特徴点抽出を用いた 1 次元探索の高速化により抑えられることが確かめられる。

表 7, 表 8 は, 多角形のベンチマーク問題例と円弧や穴を含む図形のベンチマーク問題例に提案手法をそれぞれ 10 回適用して得られた配置の充填率の最良値を, Burke ら [4], Umetani ら [2] の結果と比較している。入力する図形のデータ形式が異なるため結果に優劣を付けることはできないが, 高い解像度の図形詰込みであっても従来手法と同程度の充填率の配置を求められることが確かめられる。

最後に, ベンチマーク問題例に対する提案手法の実行結果 ($W = 512\text{px}$) を図 5, 図 6 に示す。

7 おわりに

本研究では, ラスタ表現された図形の詰込み問題に対する効率的な局所探索法を提案した。特に, 画像処理における特徴点抽出の手法を導入することで, 局所探索法の基本操作となる 1 次元探索の効率化を実現し, 高解像なラスタ図形の詰込み問題に対する効率的なアルゴリズムを実現した。数値実験では, さまざまな解像度のラスタ図形に変換されたベンチマーク問題例に提案手法を適用し, 高解像なラスタ図形の問題例に対して十分な性能が達成できることを確かめた。

参考文献

- [1] 村上祥平, 梅谷俊治, 中野雄介, 森田浩, ラスタ表現された図形の詰込み問題に対する局所探索法, 京都大学数理解析研究所 研究集会「新時代を担う最適化: モデル化手法と数値計算」講究録, **1981** (2015), 5-26.
- [2] S. Umetani, M. Yagiura, S. Imahori, T. Imamichi, K. Nonobe and T. Ibaraki, Solving the irregular strip packing problem via guided local search for overlap minimization, *International Transactions in Operational Research*, **16** (2009), 661-683.
- [3] E. Rosten and T. Drummond, Machine learning for high-speed corner detection, *Computer Vision*, **3951** (2006), 430-443.
- [4] E. K. Burke, R. S. R. Hellier, G. Kendall and G. Whitwell, Irregular packing using the line and arc no-fit polygon, *Operations Research*, **171** (2010), 948-970.

表 1: 多角形のベンチマーク問題例

問題例	図形の種類	図形数	回転角の増分
Albano	8	24	180°
Dagli	10	30	180°
Dighe1	16	16	0°
Dighe2	10	10	0°
Fu	12	12	90°
Jakobs1	25	25	90°
Jakobs2	25	25	90°
Mao	9	20	90°
Marques	8	24	90°
Shapes0	4	43	0°
Shapes1	4	43	180°
Shirts	8	99	180°
Swim	10	48	180°
Trousers	17	64	180°

表 2: 円弧や穴を含む図形のベンチマーク問題例

問題例	図形の種類	図形数	回転角の増分
Progfiles1	8	32	90°
Progfiles2	7	50	90°
Progfiles3	6	46	45°
Progfiles4	7	54	90°
Progfiles5	5	50	15°
Progfiles6	9	69	90°
Progfiles7	9	9	90°
Progfiles8	9	18	90°
Progfiles9	16	57	90°
Progfiles10	13	91	0°

表 3: 多角形のベンチマーク問題例に対する計算機環境と計算時間 (秒)

	Burke et al.[4]	Umetani et al.[2]	提案手法
	Pentium 4	Xeon	Xeon
	2.0 GHz	2.8 GHz	2.7 GHz
	10 runs	10 runs	10 runs
	time to best	time limit	time limit
Albano	299	1200	1200
Dagli	252	1200	1200
Dighe1	3	1200	1200
Dighe2	148	1200	1200
Fu	139	1200	1200
Jakobs1	29	1200	1200
Jakobs2	51	1200	1200
Mao	152	1200	1200
Marques	21	1200	1200
Shapes0	274	1200	1200
Shapes1	239	1200	1200
Shirts	194	1200	1200
Swim	141	1200	1200
Trousers	253	1200	1200

表 4: 円弧と穴を含む図形のベンチマーク問題例に対する計算機環境と計算時間 (秒)

	Burke et al.[4] Pentium 4 2.0 GHz 10 runs time to best	Umetani et al.[2] Xeon 2.8 GHz 10 runs time limit	提案手法 Xeon 2.7 GHz 10 runs time limit	
Profiles1	15	1200	1200	1200
Profiles2	295	1200	1200	1200
Profiles3	283	1200	1200	1200
Profiles4	256	1200	1200	1200
Profiles5	300	1200	1200	1200
Profiles6	171	1200	1200	1200
Profiles7	211	1200	1200	1200
Profiles8	279	1200	1200	1200
Profiles9	98	1200	1200	1200
Profiles10	247	1200	1200	1200

表 5: 多角形のベンチマーク問題例に対する提案手法の結果

問題例	特徴抽出なし					特徴抽出あり				
	128px	256px	512px	1024px	2048px	128px	256px	512px	1024px	2048px
Albano	88.48%	87.70%	86.67%	85.98%	85.48%	88.61%	87.69%	87.30%	86.80%	86.70%
Dagli	87.03%	85.97%	84.93%	84.04%	83.58%	87.09%	86.07%	85.49%	85.42%	85.14%
Dighe1	87.12%	89.99%	90.33%	86.19%	84.56%	88.48%	95.55%	97.47%	98.69%	99.20%
Dighe2	93.01%	94.84%	97.16%	98.74%	99.21%	93.01%	94.77%	97.17%	98.74%	99.20%
Fu	90.02%	89.43%	88.19%	88.17%	87.08%	90.61%	89.93%	89.33%	89.64%	88.82%
Jakobs1	84.03%	84.79%	83.58%	82.52%	82.94%	84.49%	84.32%	86.02%	85.65%	86.29%
Jakobs2	79.01%	78.42%	78.52%	76.86%	75.96%	80.67%	78.77%	78.52%	78.12%	78.23%
Mao	86.06%	84.23%	83.24%	82.64%	81.82%	86.43%	84.60%	83.48%	83.38%	82.88%
Marques	91.30%	89.47%	88.76%	87.83%	87.47%	91.39%	89.51%	89.19%	88.69%	88.14%
Shapes0	67.35%	65.71%	64.78%	63.26%	62.66%	67.38%	66.17%	65.02%	64.25%	64.25%
Shapes1	73.53%	71.78%	70.86%	69.93%	69.15%	74.04%	72.50%	71.71%	71.08%	71.00%
Shirts	86.51%	84.86%	83.67%	82.93%	82.89%	86.34%	84.86%	84.04%	83.56%	83.33%
Swim	76.69%	73.43%	72.24%	71.10%	70.24%	77.17%	73.84%	72.49%	71.81%	71.68%
Trousers	87.76%	86.72%	85.86%	85.43%	85.47%	88.13%	87.08%	86.59%	86.59%	86.76%
avg.	84.14%	83.38%	82.77%	81.83%	81.32%	84.56%	83.98%	83.84%	83.74%	83.69%

表 6: 円弧や穴を含む図形のベンチマーク問題例に対する提案手法の結果

問題例	特徴抽出なし					特徴抽出あり				
	128px	256px	512px	1024px	2048px	128px	256px	512px	1024px	2048px
Profiles1	84.74%	83.76%	83.66%	82.36%	82.01%	84.65%	84.69%	84.46%	83.37%	84.04%
Profiles2	78.96%	76.55%	75.39%	74.69%	73.21%	79.06%	76.90%	76.18%	75.91%	75.33%
Profiles3	74.53%	73.29%	71.34%	70.70%	69.40%	74.72%	73.52%	72.82%	72.22%	71.77%
Profiles4	84.95%	83.93%	83.27%	82.64%	81.71%	84.90%	85.02%	84.41%	84.38%	83.89%
Profiles5	83.12%	81.59%	80.50%	79.97%	79.04%	82.98%	81.23%	81.07%	80.72%	80.96%
Profiles6	80.66%	80.41%	78.06%	75.68%	75.75%	80.66%	80.11%	78.54%	76.78%	76.54%
Profiles7	87.28%	92.50%	95.63%	97.75%	98.27%	87.25%	92.40%	95.65%	98.18%	98.83%
Profiles8	84.29%	83.81%	82.73%	82.64%	81.87%	85.08%	84.02%	84.62%	84.90%	85.38%
Profiles9	65.78%	59.88%	56.64%	54.88%	53.34%	66.07%	60.10%	57.05%	55.53%	54.59%
Profiles10	69.80%	68.11%	66.52%	65.71%	64.64%	70.27%	68.31%	67.33%	66.85%	66.43%
avg.	79.41%	78.38%	77.37%	76.70%	75.92%	79.56%	78.63%	78.21%	77.88%	77.78%

表 7: 多角形のベンチマーク問題例に対する既存手法と提案手法の比較

問題例	Burke et al.[4]	Umetani et al.[2]	提案手法				
			128px	256px	512px	1024px	2048px
Albano	86.0%	87.56%	89.01%	88.04%	87.61%	87.22%	87.04%
Dagli	82.2%	86.35%	87.06%	86.84%	86.32%	86.27%	85.92%
Dighe1	82.1%	99.89%	91.36%	96.07%	97.60%	98.81%	99.33%
Dighe2	84.3%	100.00%	93.01%	95.01%	97.28%	98.79%	99.28%
Fu	89.2%	91.23%	91.45%	90.56%	90.33%	90.39%	89.30%
Jakobs1	82.6%	89.09%	88.58%	84.32%	88.01%	87.56%	87.98%
Jakobs2	75.1%	80.84%	81.33%	81.91%	79.51%	79.24%	79.31%
Mao	78.7%	83.73%	87.17%	85.12%	84.84%	84.03%	83.63%
Marques	86.5%	89.21%	91.39%	89.92%	90.43%	89.36%	88.52%
Shapes0	65.6%	66.50%	68.49%	66.74%	65.55%	65.10%	64.81%
Shapes1	71.5%	73.88%	75.30%	73.38%	72.35%	71.88%	71.46%
Shirts	82.8%	86.92%	86.72%	85.82%	84.33%	84.43%	84.24%
Swim	67.2%	74.54%	78.87%	74.74%	73.02%	72.21%	72.66%
Trousers	86.9%	89.40%	88.97%	87.98%	87.42%	88.02%	87.40%
avg.	80.1%	85.65%	85.69%	84.75%	84.61%	84.52%	84.35%

表 8: 円弧や穴を含む図形のベンチマーク問題例に対する既存手法と提案手法の比較

問題例	Bueke et al.[4]	提案手法				
		128px	256px	512px	1024px	2048px
Profiles1	82.5%	84.74%	86.03%	86.93%	84.32%	85.73%
Profiles2	73.8%	79.43%	77.42%	77.28%	76.27%	75.98%
Profiles3	70.8%	76.32%	74.27%	73.87%	73.51%	72.65%
Profiles4	86.8%	86.04%	85.92%	85.13%	85.31%	84.64%
Profiles5	75.9%	83.88%	81.53%	81.66%	82.10%	81.47%
Profiles6	72.1%	81.40%	81.14%	79.03%	77.50%	77.94%
Profiles7	73.3%	87.25%	92.59%	95.80%	98.27%	98.92%
Profiles8	78.7%	87.34%	84.47%	85.73%	86.65%	88.50%
Profiles9	52.9%	66.95%	61.08%	57.59%	56.64%	55.53%
Profiles10	65.0%	70.90%	68.75%	68.20%	67.91%	67.14%
avg.	73.2%	80.42%	79.32%	79.12%	78.85%	78.85%

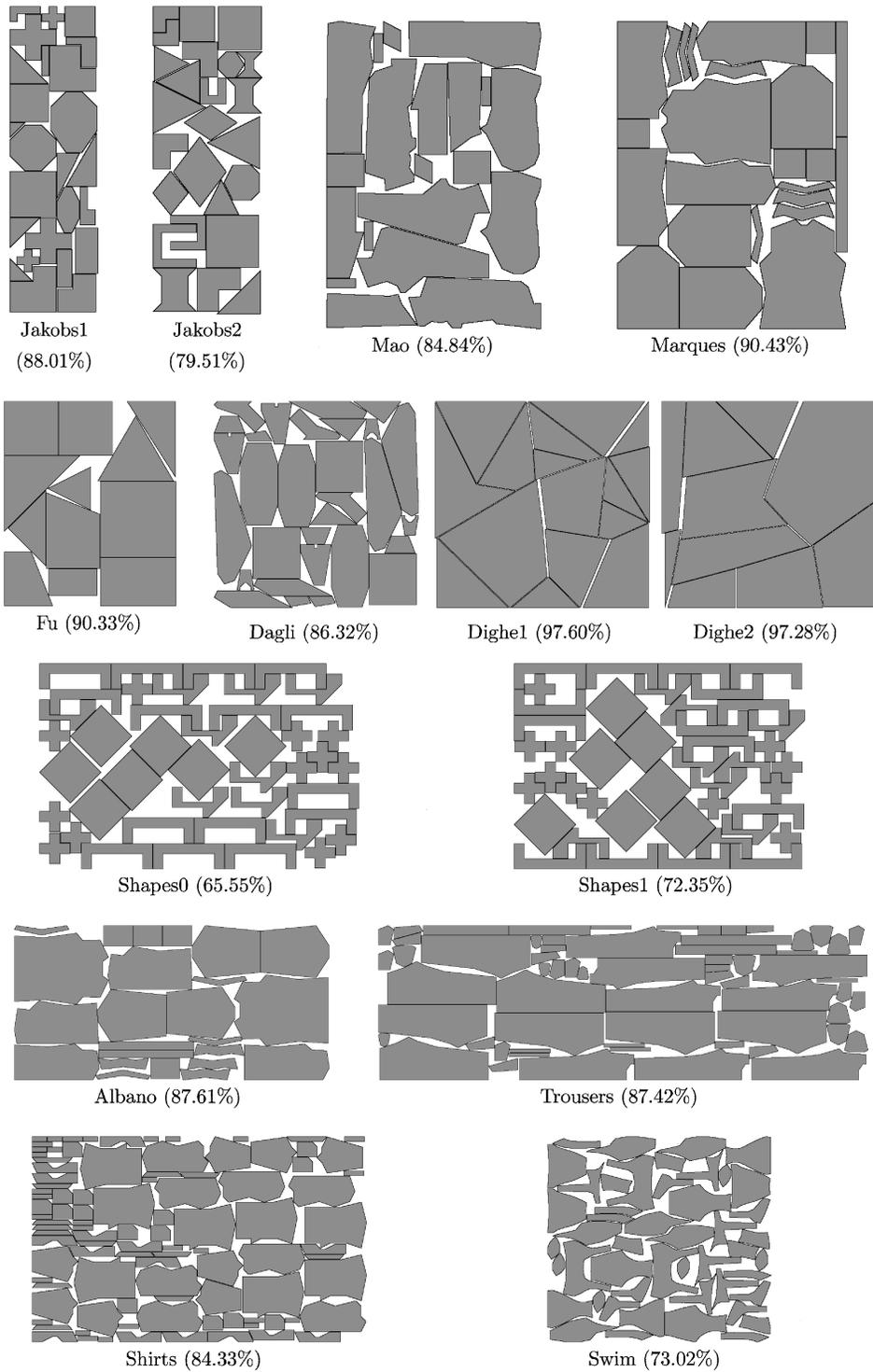


図 5: 多角形のベンチマーク問題に対する提案手法の実行結果 ($W = 512\text{px}$)

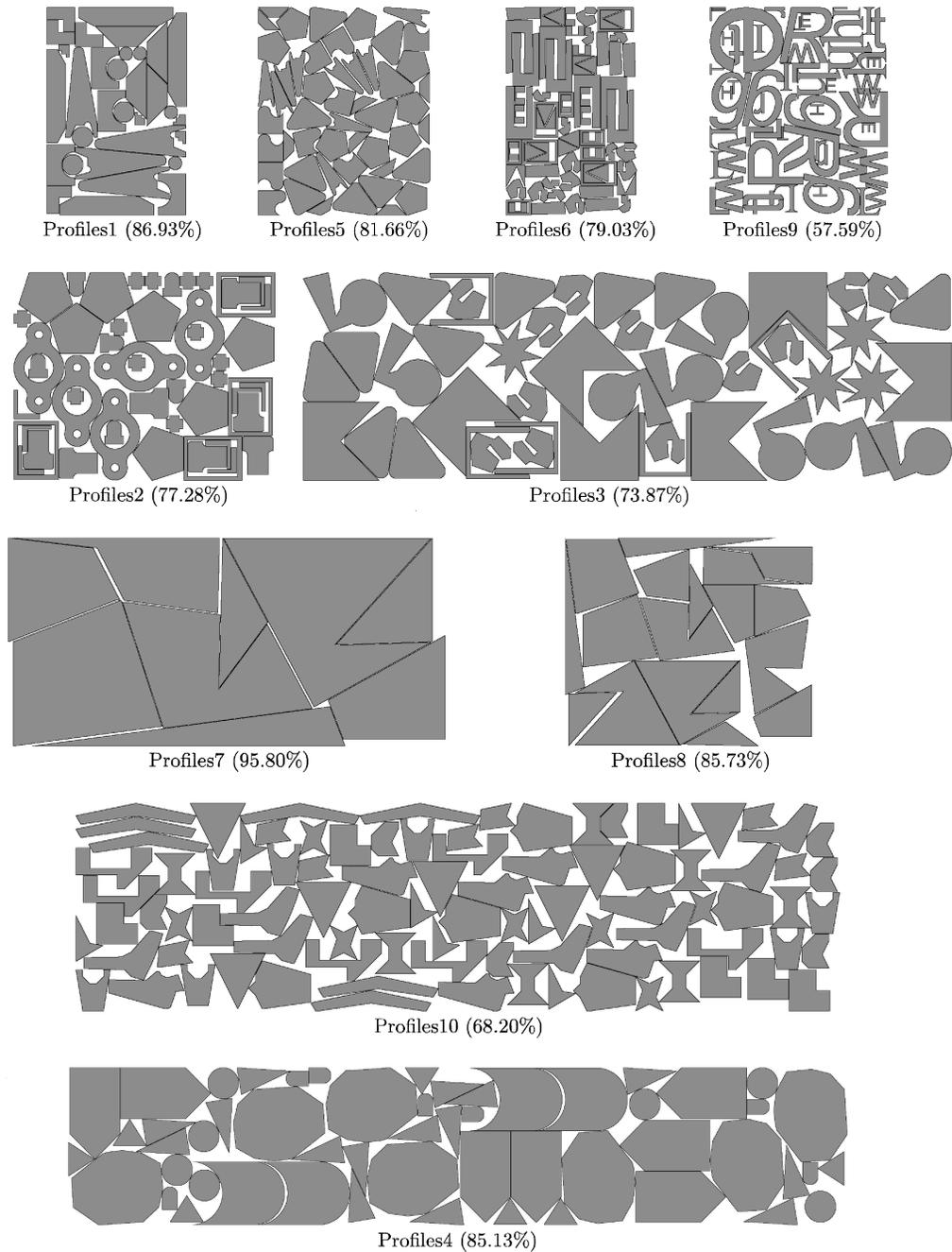


図 6: 円弧と穴を含む図形のベンチマーク問題に対する提案手法の実行結果 ($W = 512\text{px}$)