

## 圧縮率を利用した画像からのオブジェクト自動抽出

杉内 崇浩 渡辺 俊典 古賀 久志  
電気通信大学 大学院情報システム学研究所  
〒 182-8585 東京都調布市調布ヶ丘 1-5-1  
E-mail: { sug, watanabe, koga }@sd.is.uec.ac.jp

### 概要

近年、オブジェクト認識は、セキュリティ認証、ロボットビジョンなどさまざまな場面で使われるようになってきた。我々は、画像オブジェクトをカラー画像の領域分割により直接得られる領域の集合と考えることにより、オブジェクト定義の自動抽出、認識手法について検討している。本稿では、画像内に複数回出現するオブジェクトを圧縮して得られる圧縮率を利用した自動オブジェクト抽出手法を提案する。  
キーワード オブジェクト認識、画像解析、近傍集合、知識発見

## Automatic Extraction of Objects from Images Using Data Compression Ratio

Takahiro Sugiuchi Toshinori Watanabe Hisashi Koga  
Graduate School of Information Systems, University of Electro-Communications  
Chofugaoka 1-5-1, Chofu city, Tokyo, 182-8585 Japan  
E-mail: { sug, watanabe, koga }@sd.is.uec.ac.jp

### Abstract

In recent years, object recognition has come to be used in various scenes, such as security authentication and robot vision. We study automatic extraction and recognition of objects from images by considering an object as a set of the regions directly obtained by image segmentation. In this paper, the automatic object extraction method using data compression ratio in compressing the objects which appears multiple times in image is proposed.

**Keyword** Object Recognition, Image Analysis, Neighbor Set, Knowledge Discovery

## 1 はじめに

近年、オブジェクト認識は活発に研究され、監視カメラを用いた警報システム、顔画像の自動認識、工場内でのカメラによる不良品の自動判別など様々な応用がなされている。

オブジェクト認識の代表的な手法の1つに、オブ

ジェクトの構成要素を見つけることによりボトムアップで画像内のオブジェクトを見つけることを目指す手法がある。この手法は、オブジェクトの構成要素の位置関係を定義し、そのモデルを手で計算機に与えておく必要があり、手間がかかる [1]。

よってオブジェクト定義の自動生成機構の実現が望まれる。それも姿勢などが多様に変化する悪条件

の下でごく少数の画像例のみを用いて実現する必要がある。

そこで本稿では1枚の画像内からオブジェクト定義を自動的に形成し、それをモデルとして利用してオブジェクトを発見する手法を検討する。我々は、オブジェクトは近接するコンポーネント(部品)から構成されるという考えに基づき、コンポーネントの適切な近接関係を発見することにより、オブジェクト概念を発見することを目指してきた [2, 3, 4]。

今回、我々は発見されたオブジェクトの妥当性を表すコスト関数を定義し、このコスト関数が最大となるようにオブジェクトのモデルを選ぶことで自動的にオブジェクト概念を発見する手法を提案する。そして、コスト関数には圧縮率を利用した。また、これまでは領域分割済み画像のみを対象にしていた [2, 3, 4] のに対し、今回は実画像への適用も試みた。

## 2 近傍集合を用いたオブジェクト認識

ここでは、オブジェクト自動抽出手法の基礎的考え方として、太田ら [2] による近傍集合を用いたオブジェクト認識手法について述べる。以下ではコンポーネントの違いは色と大きさで判断する。

### 2.1 近傍集合

コンポーネントの位置関係を定義するために近傍集合を用いる。近傍集合とは、モデルとなるオブジェクトの各コンポーネントに対して各々1つ定義され、そのコンポーネントから見た他のコンポーネントすべての存在し得る範囲を表す。近傍集合は2種類あり、位置関係が安定な近傍集合、位置関係が不安定な近傍集合と定義する。位置関係が安定な近傍集合は、モデルとなるオブジェクトと位置関係が変動しないオブジェクトを認識するために用い、オブジェクトとその外接長方形(MBR)(図1a)との位置関係をそのまま用いる(図1b)。図1b-1の点線の範囲がC1の位置関係が安定な近傍集合を、図1b-2が

C2の位置関係が安定な近傍集合を、図1b-3がC3の位置関係が安定な近傍集合を各々表す。位置関係が不安定な近傍集合は、人間の姿勢といったようなモデルとなるオブジェクトと位置関係が変動する可能性があるオブジェクトを認識するために用い、各コンポーネントを中心として4つのMBRを組み合わせたものを用いる(図1c)。図1c-1の一番外側の点線の範囲がC1の位置関係が不安定な近傍集合を、図1c-2がC2の位置関係が不安定な近傍集合を、図1c-3がC3の位置関係が不安定な近傍集合を各々表す。近傍集合を用いることにより、位置関係の異なる2つのオブジェクトを近傍集合の変更のみにより統一的に表現することができる。

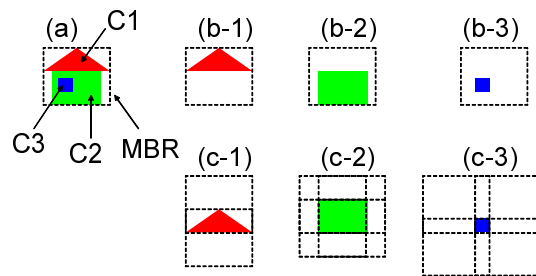


図1: (a) オブジェクトと MBR、(b) 位置関係が安定な近傍集合、(c) 位置関係が不安定な近傍集合

### 2.2 オブジェクト認識の例

近傍集合を用いて、教示されたモデルと類似したオブジェクトを全画像内から探す手法について述べる。以下に教示オブジェクト  $\{C1, C2, C3\}$  (図1a) と位置関係が変動する可能性のある類似したオブジェクトを探す例を示す。(図2)この例では不安定な近傍集合を使用する。モデルと窓コンポーネントの位置がずれたオブジェクトも探すことができる。

1. 教示オブジェクトのコンポーネントの1つ( $C1$ )と類似したコンポーネント( $C4$ )を全画像内から見つける。そのコンポーネントに $C1$ の近傍集合を当てはめる。その近傍集合内に自分以外のコンポーネント( $C2, C3$ )と類似したコンポーネント( $C5, C6$ )がすべて存在するかを調べる。(図2a)
2. すべてが見つかったならば、見つかった各コンポーネント( $C5, C6$ )に対して、各近傍集合を当てはめる。(図2b, c)
3. 教示オブジェクトのコンポーネントすべての近傍集合の重なる部分の内側に教示オブジェクトのコンポーネントと類似したコンポーネントがすべて存在すれば、類似したオブジェクトであると判断する。(図2d)

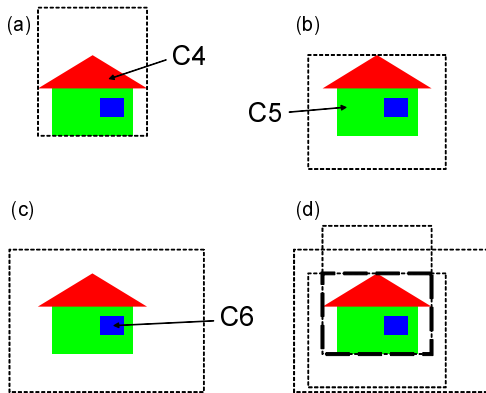


図 2: (a)  $C1$  の近傍集合、(b)  $C2$  の近傍集合、(c)  $C3$  の近傍集合、(d) すべての近傍集合の重なり

### 3 オブジェクト自動抽出手法

提案手法である人手による教示オブジェクトなしに妥当なオブジェクトを自動的に抽出する手法について述べる。本手法の概要は、モデルとなるオブジェ

クトを決めるために画像を格子状に分割し、各格子からモデルを取り出し探索を行う。太田ら [2] の手法では、格子を固定サイズにしてモデルを取り出し、自動抽出を行っていた。しかし現実的にはオブジェクトの大きさ、位置が特定の格子サイズに納まるとは限らないので、今回の提案手法では格子の大きさを変動させる。画像内から抽出されたオブジェクトに対してオブジェクトの妥当性を表すコスト関数を定義し、格子サイズを変える毎にコストを計算し、最大コストの時の格子を用いてオブジェクトを自動的に抽出するというものである。なお、コスト関数としては圧縮率を利用したものをを用いた。

#### 3.1 圧縮率を利用したコスト関数

まずはじめに本手法で用いるコスト関数について述べる。コスト関数とは画像内から抽出されたオブジェクトに対して、オブジェクトとしての妥当性を評価する関数である。オブジェクトの妥当性として以下のものが良いと考えられる。

- 1つのオブジェクト内のコンポーネントの個数が多いほうが良い。
- 類似したオブジェクトが多く出現したほうが良い。

そこで、コスト関数として一次元文字列における圧縮率を用いた。

##### 3.1.1 一次元文字列における圧縮率

一次元文字列  $abcdabceabc$  を圧縮する例を用いて圧縮率の説明をする。

図3の例では、部分文字列  $abc$  を  $A$  に置き換えることにより圧縮を行っている。この圧縮を行うことにより、 $abcdabceabc$  の  $abc\ abc\ abc$  の9文字を  $A\ A\ A$  の3文字と  $A$  の1文字と  $abc$  の3文字の合計7文字で表現している。よって圧縮により2文字分の得をしたことになる。一般に、置き換える部分

( 圧縮前 )  
 $S \rightarrow abcdabceabc$

---

( 圧縮後 )  
 $S \rightarrow AdAeA$   
 $A \rightarrow abc$

図 3: 一次元文字列の圧縮の例

文字列の長さを  $W$ 、その出現頻度を  $N$  とすると圧縮により得をする量は以下ようになる。

$$\begin{aligned} saving &= WN - (W + N + 1) \\ &= (W - 1)(N - 1) - 2 \quad (1) \end{aligned}$$

図 3 の例では、(1) 式の上段の  $WN$  が  $abc\ abc\ abc$  の 9 文字に、 $W$  が  $abc$  の 3 文字に、 $N$  が  $AAA$  の 3 文字に、1 が  $A$  の 1 文字に対応している。また、 $W = 3$ 、 $N = 3$  となり 2 文字分の得をしていることがわかる。このように置き換える部分文字列の長さ、出現頻度が多い方が圧縮率は大きくなる。

### 3.1.2 コスト関数の説明

一次元文字列の圧縮率を用いたオブジェクトの妥当性を評価するコスト関数を定義する。画像内から抽出された 1 つの種類のオブジェクトのコンポーネントの個数を  $W$ 、オブジェクトの出現頻度を  $N$  とする。すべてのオブジェクトの種類において合計することによりコスト関数を

$$cost = \sum_{\text{複数回出現した Obj}} \{(W - 1)(N - 1)\}$$

と定義する。

画像内にコンポーネントの個数が 2 個、出現頻度が 2 回のオブジェクトが抽出された場合、(1) 式をそのまま用いるとマイナスの値になってしまうので、コスト関数では (1) 式に 2 を足した値を用いている。

次に簡単な例を用いてコスト関数の有効性を示す。図 4a の画像内において自動抽出されたオブジェク

トに対して、コスト関数を計算する。自動抽出された結果として "家と木" という 1 種類のオブジェクトが抽出された場合 ( 図 4b ) オブジェクトのコンポーネントの個数  $W = 5$ 、出現頻度  $N = 2$  となりコスト関数の値は以下ようになる。

$$cost = (5 - 1)(2 - 1) = 4$$

自動抽出された結果として "家" と "木" という 2 種類のオブジェクトが抽出された場合 ( 図 4c ) "家" オブジェクトのコンポーネントの個数  $W = 3$ 、出現頻度  $N = 3$ 、"木" オブジェクトのコンポーネントの個数  $W = 2$ 、出現頻度  $N = 2$  となりコスト関数の値は以下ようになる。

$$cost = (3 - 1)(3 - 1) + (2 - 1)(2 - 1) = 5$$

今回の例では "家と木" よりも "家" と "木" のほうがオブジェクトとしての妥当性が高いという結果になった。このように一番圧縮率が高いオブジェクトの組合せが、オブジェクトとして妥当なものであると考えられる。

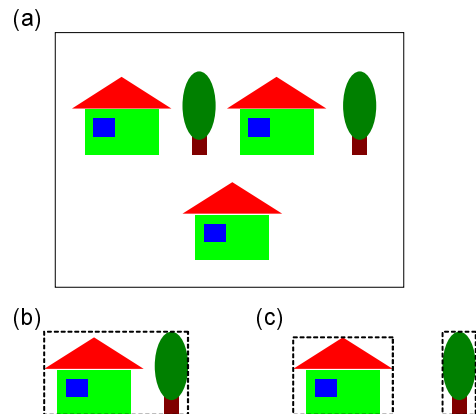


図 4: (a) 簡単な図、(b) オブジェクトの分け方 1、(c) オブジェクトの分け方 2

### 3.2 自動抽出手法のアルゴリズム

格子変動によるオブジェクトのモデルを自動的に発見する手法について述べる。アルゴリズムは以下の通りである。

Step1. 画像内に均等になるように格子を作る。(図 5a) 各格子ごとにモデルを作り探索を行い、抽出されたオブジェクトに対してコストを計算し、初期コストとする。

Step2. 1つの格子を拡大しオブジェクトを抽出する。

1. 1つの格子を拡大し、その格子内からモデルを作り探索を行う。(図 5b)
2. 残りのすべての格子においても探索順序(初めは左上から順に)に従い同様にモデルを作り探索を行う。

Step3. 抽出されたオブジェクトに対してコストを計算し、今までで最大ならば拡大した格子サイズを保持し、その格子を探索順序の一番始めに持って来る。

Step4. 1つの格子において拡大作業(Step2、Step3)を一定回数繰り返す(図 5c, d)

Step5. すべての格子に対して拡大作業(Step2、Step3、Step4)を行う。

Step6. 最大コストの時の格子サイズ、探索順序で探索を行いオブジェクト自動抽出を行う。

まずStep1として、モデルを自動的に抽出するための格子を作る(図 5a)。この格子に含まれるコンポーネントの集まりをオブジェクトのモデルと考える。拡大順序、初期の探索順序として左上から右下までの順で順序を設定する。初期格子の大きさで、探索順序1番目の格子でモデルを作り、このモデルと類似したオブジェクトを全画像中から探索を行う。同様に、探索順序に従い、すべての格子においてモデルを作り探索を行い、抽出されたオブジェクトに対してコストを計算する。なお、すべてのコンポー

ネントはただ一つのオブジェクトにのみ属するという考えにより、一度探索で抽出されたオブジェクトのコンポーネントは一回のオブジェクト抽出が終わるまでは使用できないこととする。

Step2として、拡大順序1番目の格子を拡大し(図 5b) その拡大した格子でモデルを作り探索を行う。その次に、拡大した格子以外の格子を探索順序に従い各格子でモデルを作り、探索を行いオブジェクトを抽出する。

Step3として、抽出されたオブジェクトに対してコストを計算する。コストが今までで最大の値であれば、その拡大した格子サイズを保持し、探索順序を一番はじめに設定する。これは、Step2において拡大する格子を一番はじめに探索することで、コストが最大になったのでその順番を保持するために行う。

Step4として、拡大順序1番目の格子において拡大作業(Step2、Step3)を一定回数繰り返す(図 5c, d)

Step5として、拡大順序2番目の格子に対しても同様に拡大作業(Step2、Step3、Step4)を行う。ただし、このとき拡大している格子以外の格子のサイズは今までで最高のコストであった時に保持した格子サイズを使用する。そして、同様に格子拡大をすべての格子に対して行う。

最後にStep6として、コストが最大の値である時の格子サイズ、探索順序を用いてモデルを作り、探索を行い、それをオブジェクト自動抽出結果とする。

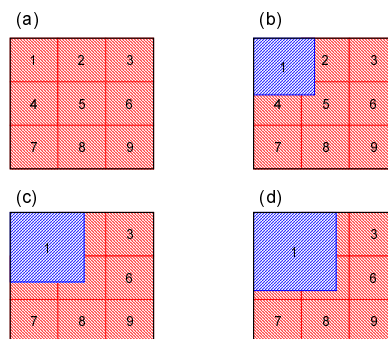


図 5: (a) 初期格子、(b)(c)(d) 格子拡大の様子

## 4 実験

本手法の有効性を示すために模擬画像、実画像を用いて実験を行った。入力として画像を1枚のみ与え、自動抽出された結果を黒枠に囲まれた部分として表示する。

### 4.1 模擬画像を用いた実験

模擬画像(図6)を用いて実験を行った結果を示す(図7)。ここでは位置関係が不安定な近傍集合を使用した。家が3個、木が8個、赤い服の人が2個、青い服の人が3個の4種類のオブジェクトが自動的に抽出された。

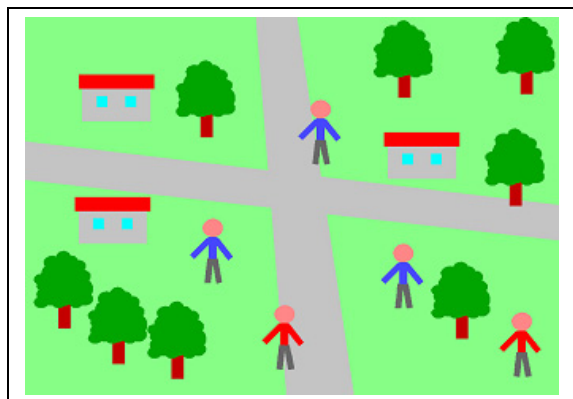


図 6: 実験画像(模擬画像)

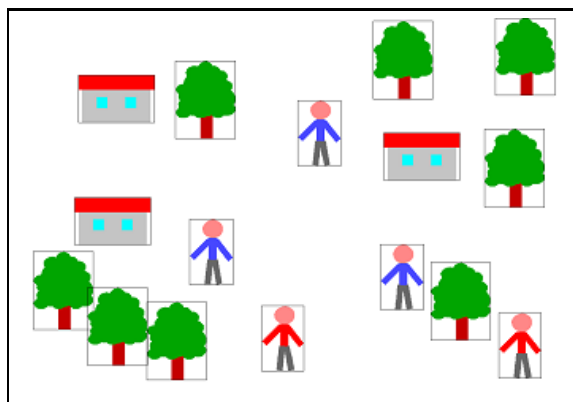


図 7: 自動抽出結果

### 4.2 実画像を用いた実験

実画像をそのまま本手法に用いるのは困難なので、実画像を領域分割、減色処理を施すことにより、提案手法を取り扱える画像に変換する。また、ノイズによる影響を少なくするために、格子から取り出されるモデルのコンポーネントを画素数の大きいほうから90%分のみを取り出して使用している。

#### 4.2.1 前処理

今回行った前処理の手順は以下の通りである。

1. MST を用いた領域分割
2. K-means 法による色合わせ
3. 背景除去
4. ノイズ除去

前処理として、はじめに本研究室で研究した最小全域木(MST: Minimum Spanning Tree)を用いた領域分割手法[5]を用いて領域分割を行った。次に、K-means法を用いて各領域の色の数を減らした。今回の実験では $K=10$ とした。また、RGB空間に均等になるように16個の点を配置し(図8)、それらに距離が最も近い画素を画像から選び、その中からランダムで初期点を選んだ。そして4回程度の試行を行い良好な結果のものを用いた。最後に、より扱いやすい画像にするために背景、ノイズ除去として画像内の領域の画素数が極端に多いものと少ないものを除去した。

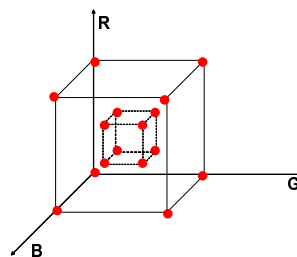


図 8: RGB空間における16個の点の配置



#### 4.2.2 実験結果

実画像(図9)を前処理した画像(図10)を用いて実験を行った結果を示す(図11)。ここでは回転した類似オブジェクトを取り出すために、位置関係が不安定な近傍集合を使用した。1種類、3個のオブジェクトが自動的に抽出された。

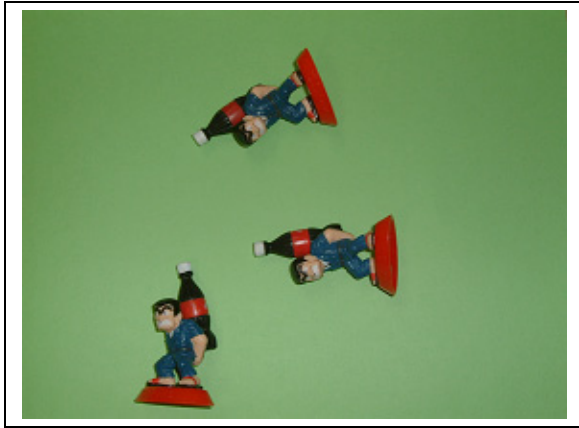


図 9: 実験画像(実画像)

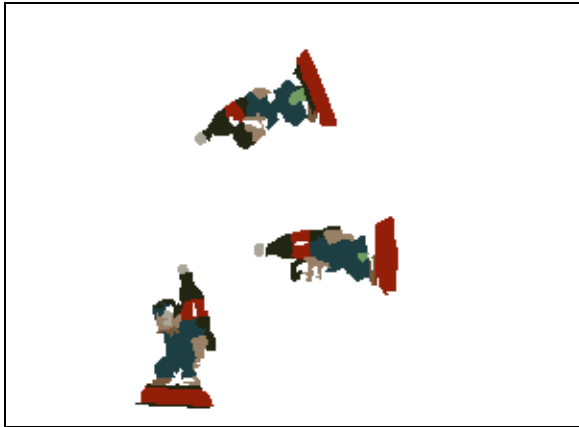


図 10: 実験画像(前処理後)

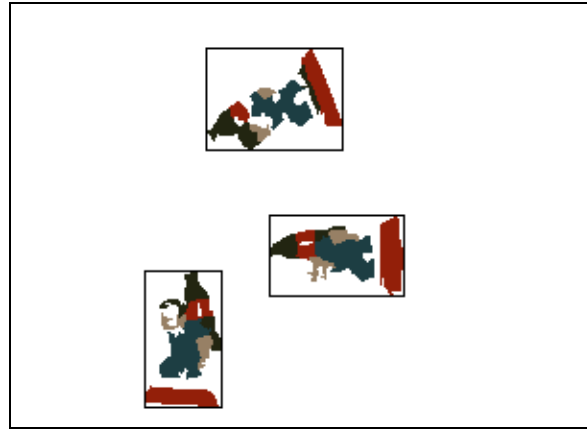


図 11: 自動抽出結果

実画像(図12)を前処理した画像(図13)を用いて実験を行った結果を示す(図14)。ここでは位置関係が安定な近傍集合を使用した。2種類、2個ずつのオブジェクトが自動的に抽出された。

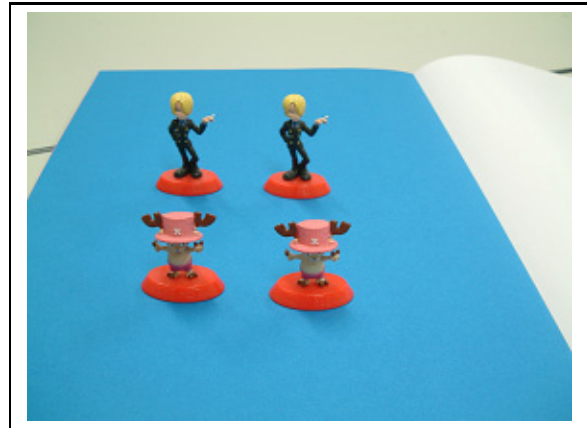


図 12: 実験画像(実画像)

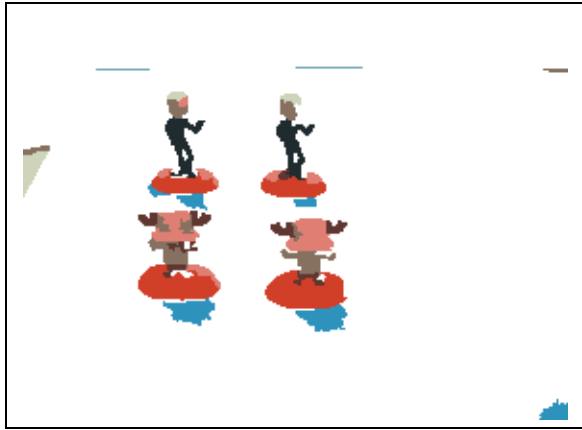


図 13: 実験画像 (前処理後)

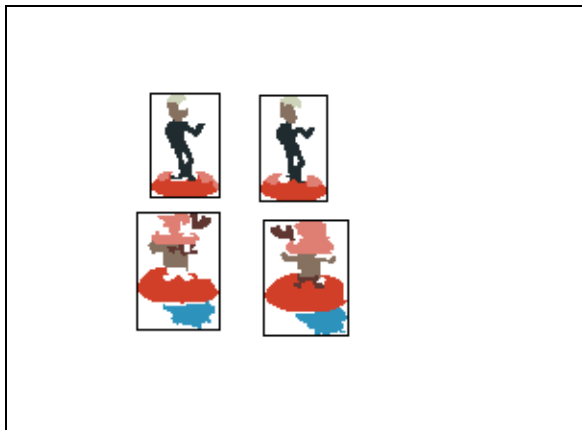


図 14: 自動抽出結果

## 参考文献

- [1] A. Mohan, C. Papageorgiou, T. Poggio : “Example-Based Object Detection in Images by Components”, IEEE Trans.PAMI, Vol.23, No.4, pp.349-361, 2001.
- [2] 太田貴彦, 渡辺俊典, 菅原研 : “画像からのオブジェクト定義自動抽出方法の研究”, 信学技報, PRMU2002-258, pp.93-98, 2003.
- [3] 久保孝弘, 太田貴彦, 渡辺俊典, 菅原研 : “航空写真からの複合オブジェクト認識手法の研究”, コンピュータビジョンとイメージメディア 132-2 (情報処理学会研究会), pp.9-14, 2002.
- [4] 西川雄二, 久保貴弘, 渡辺俊典, 菅原研 : “航空・衛星画像内のオブジェクト認識手法の研究”, 日本写真測量学会平成 13 年度年次学術講演会発表論文集, pp.149-152, 2001.
- [5] 角田夏貴, 渡辺俊典, 菅原研 : “最小全域木の適応しきい値切断による画像領域分割方式”, 電子情報通信学会論文誌, Vol.J87-D-II NO.2 (Feb2004), pp.586-594.

## 5 まとめ

1 枚のみの画像からのオブジェクト自動抽出能力の向上という課題に対して、画像内に複数回出現するオブジェクトを圧縮して得られる圧縮率を利用した自動抽出手法を提案した。そしてコンポーネントの位置変動に対応した妥当性のあるオブジェクトを自動的に抽出できることを実験により示した。今後はより複雑な画像への適用法を検討していく。