

# 特別研究報告書

## 対象物間の位置関係知識に基づいた画像解釈

指導教官 池田克夫 教授

京都大学工学部情報工学科

昭和 62 年度入学

棕木雅之

平成 3 年 2 月 15 日

## 対象物間の位置関係知識に基づいた画像解釈

棕木雅之

### 内容梗概

画像認識システムは、一般に様々な要素から構成されている。ここでは、画像認識システムを、大量の画像データを数値的に扱う画像処理部と、その結果に知識を適用して記号的に処理を行う画像解釈部に分けて、それぞれをネットワークでつながった別々の計算機上で実行する場合について考える。このような構成を用いると、それぞれの部分を処理特性に合った計算機上で実行させることができる、複数の画像解釈部が一つの画像処理部を共同で利用することができるなどの利点がある。

画像解釈部は、画像処理部に対し処理要求を出し、その結果と解釈部が持つ知識とを用いて解釈処理を行う。この場合、画像処理部から画像解釈部へ、処理結果を画像としてそのまま送ることは、通信のコストがかかるうえ、処理結果を画像解釈部で利用しにくいいため実用的ではない。本研究では、次のようなプロトコルを採用する。まず、画像解釈部からの処理要求が記号により表現されて画像処理部に送られる。これに対し、画像処理部が処理結果をトークンとして記号により表現し、画像解釈部からの処理要求をトークンがどの程度満たしているかの評価を行い、評価結果をサポート値としてトークンと共に返す。このプロトコルを採ることにより、画像解釈部が利用しやすい形で、効率的に通信を行うことができる。

画像解釈は、画像の一部分と、システムが内部に持つ対象物のモデルとの対応づけとみなせる。対象物のモデルを2次元的な位置関係で持つシステムでは、異なる視線毎に別々のモデルを持つ必要があり効率的でない。そこで本システムでは、対象物間に普遍的に成り立つ3次元的な位置関係を知識として用いる。これを位置関係知識と呼ぶ。位置関係知識を用いることにより、対象物の奥行きを持った配置をモデルに含めることができ、画像中の対象物の3次元的な位置関係の解釈を行うことができる。また、位

置関係知識は、対象物に対する視線の変化に関わらず成立するため、様々な方向から写された画像に対して、同じ知識を利用できる。

画像解釈部では、画像から得られるトークンと位置関係知識から、画像内の対象物の大まかな位置関係を取り出し（画像構成仮説の生成）、それを用いてトークンと対象物のモデルとの対応づけを行う（トークン解釈処理）。

位置関係知識は、その性質により、三つの種類に分けることができる。これらはそれぞれ、対象物間の位置関係の妥当性を表す知識、対象物間の典型的な位置関係を表す知識、対になって現れる対象物の位置関係を表す知識を表している。これを用いることにより、画像内の対象物（対象物インスタンス）の大まかな位置関係を効率的に取り出せることを示す。対象物インスタンスとその間の画像上での位置関係を画像構成仮説と呼ぶ。画像構成仮説が持つ位置関係は、3次元的な位置関係の知識から生成されるため、画像中の対象物間の3次元の関係を反映している。即ち、3次元的な位置関係の解釈が暗に行われているとみなせる。

更に、生成された画像構成仮説を用いて、トークンと対象物モデルとの対応づけを行う。位置関係を用いない場合、対象物モデルとトークンとの対応が、一意にはとれない場合がある。これに対し、画像構成仮説を用いると、その場合のトークンの解釈に含まれる曖昧さを正しい方向で改善できることを示す。

これらの考えに基づき、電車の写っている風景画像に対して、画像構成仮説の生成とトークン解釈処理の実験を実際に行った。ここで提案した位置関係知識に基づいた画像解釈が有効であることを示した。

## Image Interpretation Based on the Knowledge of Relations Between Objects

MASAYUKI MUKUNOKI

### Abstract

Image recognition system (IRS) generally consists of various functional elements. Here, the IRS is divided into two parts: one is the image processing part (IPP) which processes large amount of numerical data in images, and the other is the image interpretation part (IIP) which operates the result of the IPP symbolically. These two parts have different characteristics in processing, so it is reasonable to execute them on different computers, which has suitable architecture for its processing.

When these two machines are connected by a network, the protocol between them is essential because it decides the efficiency of the IRS. Our IRS uses the symbols and the support-values protocol. The symbol represents the request from IIP and the result of IPP (token). The support-value represents the degree of confidence of the resulting token which satisfies the request from IIP. This protocol is efficient and make it easy to handle the result of IPP in IIP.

The IIP in our IRS is described in this report. Image interpretation task is to match a part of the image with the object model in the IRS. If the object models are represented as two dimensional relations, it needs different models for each different camera directions. Therefore, the knowledge of three dimensional relations between objects (the relation knowledge) is proposed, which does not change with camera direction and enable three dimensional interpretation of two dimensional image.

The relation knowledge is divided into three types: one represents

the knowledge of appropriate relations between objects, one the knowledge of typical relations between objects, and the other the knowledge of relations in which the objects appear together. We show that the objects in the image and the rough relations between them can be efficiently extracted by using the relation knowledge. The extracted result is called the image constitution hypothesis (ICH). Since the ICH reflects three dimensional relation of the objects in the image, we can say that three dimensional interpretation of the image can be obtained implicitly.

Next, we describe the process of matching the tokens with object models (the token interpretation). We show that ambiguities in the token interpretation without concerning relations can be improved by using generated ICH.

Some experiments are done on some outdoor scene images which contain a train, and the results are shown. By these experiments, the validity of our proposal is proved.

## 目次

1	はじめに	1
2	画像認識システムの構成	3
2.1	全体の構成 . . . . .	3
2.2	画像解釈部の処理の流れ . . . . .	4
3	位置関係知識に基づいた画像構成仮説の生成とトークン解釈	7
3.1	対象物間の位置関係知識 . . . . .	7
3.2	画像構成仮説の生成 . . . . .	11
3.3	画像構成仮説に基づいたトークン解釈処理 . . . . .	12
4	実験結果および考察	17
4.1	画像構成仮説生成実験 . . . . .	17
4.2	トークン解釈実験 . . . . .	18
5	おわりに	21
	謝辞	
	参考文献	
	付録	

## 1 はじめに

人間が容易に行っている処理の多くは、処理が無意識に行われているので、その仕組みは解明されていない。そのため、計算機上で実現することが難しく、現在も多くの試みがなされている。そうした分野の一つに、画像認識が挙げられる。

画像認識には、あるアルゴリズムで物体の認識が行えることを示すことにより、人間が物体を認識する仕組みの提案を行うという認知科学的側面と、計算機により物体を認識させ、結果を利用するという工学的側面とがある。この双方の立場から、よくとり上げられる対象として、風景画像がある。風景画像の認識は、風景中に現れる様々な物体を対象に、画像の撮られた季節、時刻などの外的要因や対象物に対する視線の方向など、様々な要因を考慮して認識処理を行うことが要求される。そのため、計算機により自動的に認識を行うことが難しく、非常に興味深い研究対象である。

風景画像を対象に画像認識を行うシステムには、文献 [1][2][3] で報告されているものがある。これらのシステムでは、対象物を、色や形、関係などの属性で記述している。計算機内での対象物の記述を対象物モデルと呼ぶ。これらのシステムでは、対象物モデルと、対象物を画像中から探す処理を制御する知識とを用いて、画像認識を行っている。これらのシステムに用いられているフレームやプロダクションシステムを用いた知識表現法は、風景画像に限らず適用することができるので、汎用性が高い。しかし、対象物モデルが、画像上での位置関係を反映した2次元の関係で記述されているので、これらのシステムによる画像の解釈は、単に画像中の部分と対象物モデルとの対応づけに過ぎない。

そこで、本研究では、風景画像を対象に、対象物を実際の3次元世界で

の位置関係により記述する。これを位置関係知識と呼ぶ。位置関係知識は、現実の世界において普遍的に成立する位置関係の集まりであり、これを用いた画像解釈は、画像を参照しながら、画像に写っている現実世界を解釈することにつながる。

本研究では、画像認識システムを、

- 画像中から、ある基準で同一とみなせる部分を切り出す画像処理部
- 切り出された部分に対して解釈を行う画像解釈部

の二つに分けて考える。これにより、画像処理部と画像解釈部のモジュール性が高まり、その間のプロトコルを守りさえすれば、それぞれの内部を独立に変更することができる利点がある。本報告書では、このように分けた画像認識システムの内、画像処理部の存在を仮定した上で、位置関係知識に基づいて画像解釈を行い、結果を示す。

以下、2章では、画像認識システムの全体像と本報告書の中心になる画像解釈部の概要について述べる。3章では、画像解釈部のもつ位置関係知識について述べ、それを用いた処理について実際に作成した処理系をもとに述べる。更に、4章では、画像に対して処理を行った結果と考察を述べる。

## 2 画像認識システムの構成

### 2.1 全体の構成

本研究では、画像認識システムを、大きく画像処理部及び画像解釈部の二つの処理部分に分ける（図 2-1 参照）。画像処理部の役割は、画像解釈部からの処理要求にしたがって、領域分割など実際の画像処理を行い、処理結果を画像解釈部に返すこと、及び、処理した結果の画像を管理することの二つからなる。画像処理部から返される処理結果をトークンと呼ぶ。本システムでは、トークンは、画像解釈の画像上での単位となる。画像解釈部は、画像処理部から得たトークンに画像解釈部が持つ知識を適用し、画像の解釈を行う。

画像処理部が、大量の画像データを数値的に扱うのに対して、画像解釈部は、主として記号上の操作を行う。この二つの処理は、処理特性が異なるので、それぞれに合ったハードウェア上で実行されることが望ましい。そこで、この二つの部分を、ネットワークでつながれた別々の計算機上で処理することを考える。これにより、それぞれの処理をその処理特性にあった計算機上で実行することができる。また、一つの画像処理部を複数の画像解釈部が共同で利用することができる。

ネットワークで接続を行う場合、画像処理部と画像認識部との間のインターフェースは重要である。本システムは、画像解釈部からの処理要求に対して、画像処理部が処理結果を返す、というサーバ・クライアントモデルに基づいてプロトコルを考える。この際、画像処理部の処理結果を、画像としてそのまま画像解釈部に送ることは、通信のコストが多くかかるので、実用的でない。また、画像解釈部にとっては、画像処理部の処理結果の画像は、画像中の個々の画素に情報が分散されているので、利用しにく

い。その点を考慮して、情報の集積度が高く通信のコストが抑えられ、また、利用もしやすい、記号とサポート値（信頼度）を用いたプロトコルを採用する。

実際のプロトコルは、記号として、トークンの識別子であるトークンラベル、システムのもつ対象物モデルの識別子である対象物ラベル、及び画像上での位置関係を記号的に表した2次元の関係をを用いる。具体的には、画像解釈部からの

(対象物ラベル 2次元の関係)

の組での処理要求に対し、画像処理部が

(トークンラベル 対象物ラベルとそのサポート値  
2次元の関係とそのサポート値)

の組のリストを返すプロトコルで通信を行う。画像解釈部から送られる対象物ラベルや2次元の関係は、画像処理部では、画像解釈部に返すべき結果のトークンが満たすべき制約として働く。画像処理部が返すサポート値は、そのトークンに与えられた制約が、どの程度の信頼度で成立しているかの値である。

このように、画像処理部と画像解釈部の間のプロトコルを定めることにより、汎用性のある画像処理部を考えることができ、画像処理部、画像解釈部双方とも、そのプロトコルに準拠する限り、内部を自由に設計することができる。

## 2.2 画像解釈部の処理の流れ

前節に示した二つの部分のうち、本研究で取り上げた画像解釈部の処理の流れについて述べる。画像解釈部は、画像処理部から得たトークンと画像解釈部が持つ位置関係知識から、画像中に存在する対象物の種類と位置

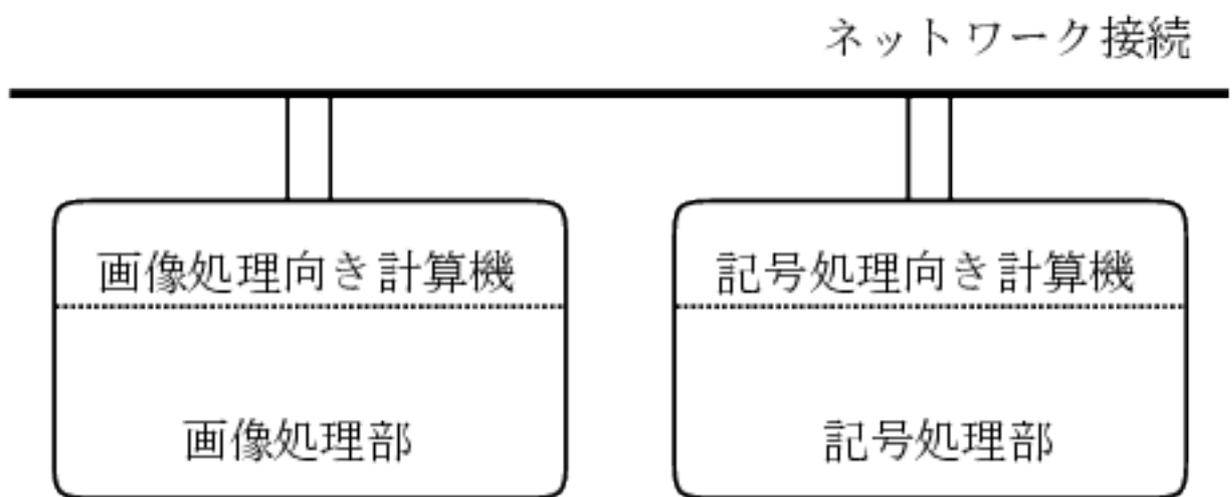


図 2-1: 全体の構成

関係を推測し、画像構成の仮説を生成する。ここでは、画像中に存在する対象物を対象物インスタンス、対象物インスタンスの種類と位置関係を画像構成と呼ぶ。また、生成された仮説を、画像構成仮説という。最初に生成される画像構成仮説は、画像中の大きな対象物インスタンスに対応するのみであるが、これを順次詳細化することにより、トークンと対象物モデルとの対応づけ、即ちトークン解釈を改善していく。画像解釈部の処理は次のステップからなる (図 2-2 参照)。

1. 画像処理部からトークンのデータを受けとり、位置関係知識の制約を満たす妥当な画像構成仮説をすべて生成する (画像構成仮説の生成)
2. 生成された画像構成仮説を画像処理部にフィードバックし、トークンの切り出しを改善する (再画像処理)
3. 改善されたトークンに対して、画像構成仮説をもとに、トークン解釈を行う (トークン解釈処理)
4. トークン解釈処理の結果により、有望な画像構成仮説を選別する (画像構成仮説の評価)
5. 選別された画像構成仮説をより詳細化し、同様の処理を行う (詳細化)

画像構成仮説は、画像中のどの位置が何の対象物モデルに対応するかの大まかな情報を与えるものである。従って、その生成時に、画像解釈部が、画像処理部から最初に受けとるトークンは、数が少ない方が都合がよい。このトークンに対し、3.1 節で述べる位置関係知識を適用することにより、画像構成仮説を生成する。画像処理部では、画像構成仮説をトークンの再処理の評価基準として用ることにより、トークンが画像中の対象物インス

タンスと1対1の対応がとれる方向に改善できる。改善されたトークンに対して、トークン解釈を行う。トークン解釈処理は、トークンと対象物モデルとの対応をとる処理である。この際、画像構成仮説を利用することにより、効率的かつトークン単体での曖昧さを減らしたトークン解釈が行える。ここで、再画像処理とトークン解釈処理は、生成された画像構成仮説毎に独立なものなので、容易に並列化できる。結果は、単にトークンと対象物モデルの対応を示すだけでなく、画像構成仮説の検証を行っているとも考えることもできる。従って、トークン解釈の結果を用いて、生成された画像構成仮説のうち確度の高いものを選ぶことで、より正確な画像解釈に進むことが期待できる。画像構成仮説の詳細化処理は、まだ検討中である。

次章以降では、これらの処理の中心となる知識である位置関係知識、及びそれを利用した画像構成仮説の生成とトークン解釈処理について、実際に作成した処理系をもとに述べていく。

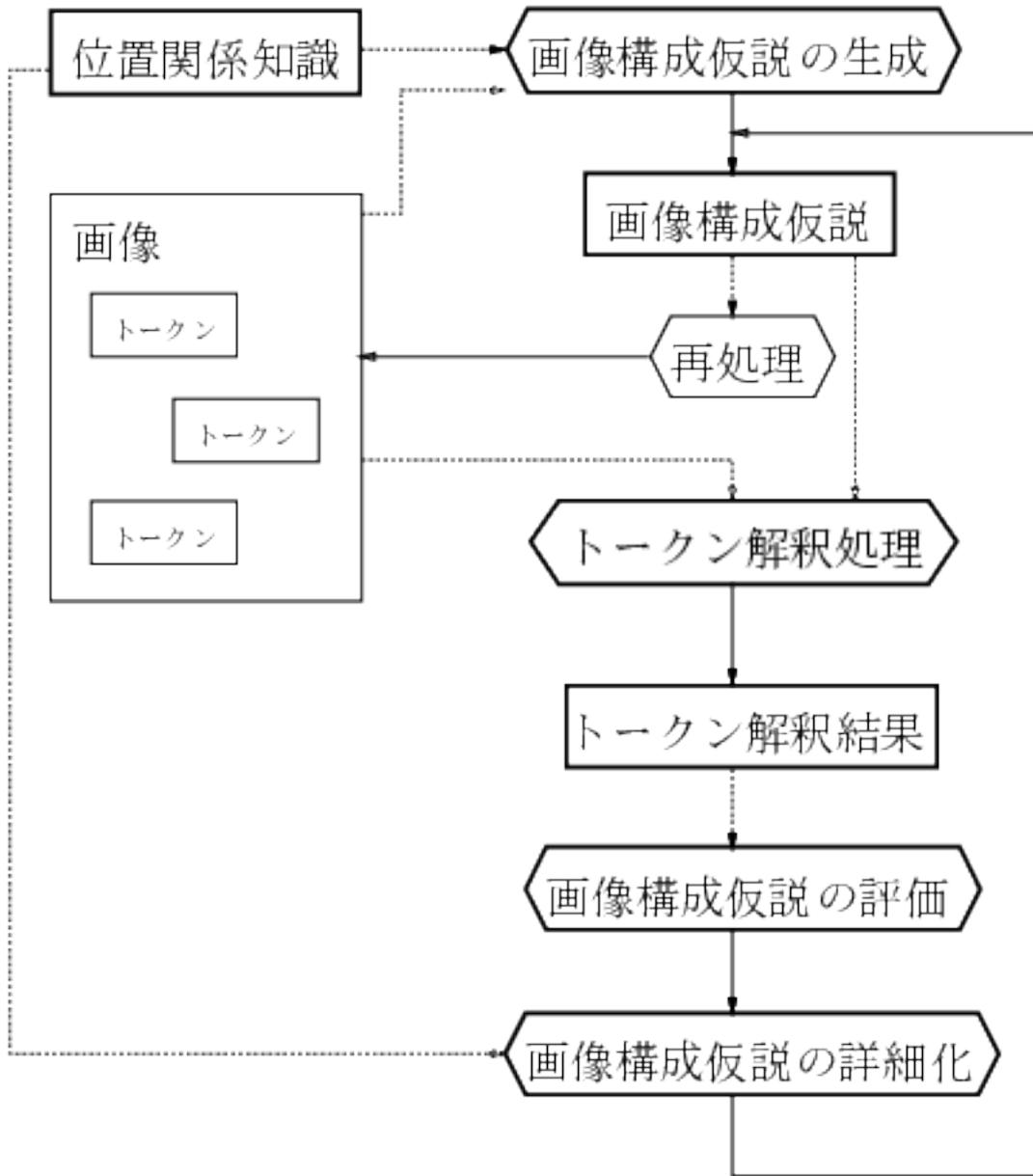


図 2-2: 画像解釈部の処理の流れ

### 3 位置関係知識に基づいた画像構成仮説の生成とトークン解釈

#### 3.1 対象物間の位置関係知識

画像解釈を行う際、対象物間の位置関係知識を用いる。画像解釈部がそれまで認識の対象に含めていなかった物体を、認識の対象に含めるには、その物体に関する位置関係知識を画像解釈部に登録するだけでよい。

位置関係知識は、対象物間に成り立つ3次元世界での普遍的な関係を表している。例えば、「電車は地面の上にある」「空はすべての上にある」などである。この知識は、画像からの画像構成仮説の生成に用いる。

位置関係知識は、3次元位置関係の集まりとして表される。3次元位置関係は

(主体対象物ラベル 3次元の関係 関係対象物ラベル)

の形式で表される。最初の項は、3次元の関係の主語に当たる対象物モデルのラベル、または対象物クラス（後述）で、最後の項は3次元関係の目的語に当たる対象物モデルのラベル、または対象物クラスである。

3次元の関係として、ここでは空間的上下関係 (over, under)、接触関係 (on-over, on-under, face)、カメラ位置からの前後関係 (before, behind) を用いる。前後関係は、主として制約として用いる。即ち、「大きなものの後ろに小さなものは存在しない（見えない）」という条件を表すことができる。

3次元位置関係は、基本的には対象物モデル個々の間で定義するが、対象物の持つ特性によって対象物モデルをクラス分けすることにより、効率的に定義できる場合がある。例えば「山は草地の上にある (mount on-over

grass)」「電車は草地の上にある (train on-over grass)」「岩は水面の上にある (rock on-over water)」などはすべて、高さを持つ物体と地平面（水平面）との間に常に成り立つ関係である。このため、対象物モデルを次の四つのクラスに分ける。

1. 背景のクラス (space)
2. 高さを持つ物体のクラス (solid)
3. 地平面・水平面のクラス (plane)
4. 平面上の模様のクラス (figure)

これらを対象物クラスと呼ぶ。任意の対象物モデルはこれらのいずれかのクラスに属する。例えば、空は space のクラス、電車は solid のクラス、草地は plane のクラス、影は figure のクラスに属する。このクラスを用いることにより先の例は、「solid は plane の上にある」と簡潔に表すことができる。

位置関係知識は、その性質により3種類に分けることができる。一つ目は、対象物が存在した時必ず成立しなければならない3次元位置関係の集合で、must-be の知識と呼ぶ。例えば、「空はすべての上にある」という3次元位置関係である。これは、逆にいえば「空が何かの下にあるのはおかしい」ということを表しており、それに反する解釈を禁止する。二つ目は、対象物が存在した時成立するかも知れない3次元位置関係の集合で、may-be の知識と呼ぶ。例えば「plane と space が接している」という3次元位置関係である。これは、常に成り立つものではないが、もし成り立つならば、画像の構成を表す重要な3次元位置関係の一つになる。三つ目は、対象物が存在した時、成立することが推定される3次元位置関係で、perhaps の知識と呼ぶ。例えば、「電車の下に影がある」という3次元位置関係である。これを用いることにより、電車が見つかった時、その下に影

の存在を推定できる。即ち、画像処理部のトークン抽出の失敗をこの種類の知識によって補うことができる。

位置関係知識は、実際には画像上の2次元位置関係として利用される。2次元位置関係は

(主体対象物ラベル 2次元の関係 関係対象物ラベル)

の形式で表される。

2次元の関係としては、画像上の上下関係 (above, below)、包含関係 (inner, outer)、距離 (near) の組を用いる。また、それぞれについて“いずれも成立しない (nil) ”、“いずれでも良い (t) ”を表す記号も導入する。例えば、(above nil near) は「上方近く、包含関係なし」という2次元の関係を、(t inner near) は「内側近く、上下関係は考慮しない」という2次元の関係を表している。

3次元位置関係と2次元位置関係の変換は、1対多の対応になる。また、この変換は、視線方向により異なったものとなる。現在のシステムでは、地表面近くからの水平方向を向いた視線での変換のみを実現している。この変換に対する変換規則は、対象物クラス間の3次元位置関係のみに対して定義される。対象物モデル間の3次元位置関係は、その対象物モデルが属する対象物クラス間の変換規則に従って、2次元位置関係に変換される。この変換規則の形式は以下のようなものである。

(主体対象物クラス 3次元の関係 関係対象物クラス  
2次元の関係 1 2次元の関係 2 ...)

前半の3項は、変換をうける3次元位置関係を表している。4項目以降は、2次元位置関係に変換された時の2次元の関係を表している。例えば、

(space on-over solid (above nil near))

という規則は、space に属する対象物モデル A と solid に属する対象物モデル B との間の on-over の 3次元位置関係が「A が B の上方近くにある」という 2次元位置関係に変換されることを表している。即ち、

(A on-over B)

は、

(A (above nil near) B)

に変換される。同様に、

(solid over plane (above nil t) (above inner t))

という規則は、solid に属する対象物モデル C と plane に属する対象物モデル D との間の over の 3次元位置関係が「C が D の上方ある」または「C が D の上方内側にある」という二つの 2次元位置関係に変換されることを表している。

これらの知識は画像によらない普遍的なものである。この知識を利用する利点としては次の点が挙げられる。

- 3次元世界での位置関係を個別に列挙しておくだけで、2次元位置関係が得られる。
- 2次元の知識では、視線の変化は適用する知識の集合の切替えになる。即ち、異なる視線毎に別々の位置関係知識が必要になる。これに対しこの方法では、適用する変換規則を切替えることにより対応できる。
- 3次元位置関係の利用により、画像の3次元解釈が可能になる。即ち、3次元位置関係から生成された解釈は、少なくとも暗に対象物の3次元世界中での位置関係を含んでいる。これを積極的に利用すれば、画像の3次元解釈も可能である。

- 位置関係知識により、画像からのボトムアップ処理では得られなかった情報の存在が効率的に仮定でき、画像処理の改善が行える。

### 3.2 画像構成仮説の生成

画像から画像構成仮説を生成することは、画像解釈の大きな目標の一つである。本節では画像処理部から初期的に得られたトークンから、位置関係知識に照らして妥当といえる画像構成仮説をすべて生成する処理について述べる。

まず、画像処理部からトークンのデータを受けとる。これは、

- トークンのリスト
- 各トークン毎の対象物モデル及びそのサポート値のリスト
- トークン間の2次元の関係及びそのサポート値のリスト

からなる。ここでは、画像の大まかな構成を得ることを目的としているので、各トークンは画像中の大きな対象物インスタンスに対応していれば良い。従って、受けとるトークンは10個程度とする。これらは、対象物の境界で正しく切り出されている必要はないが、対象物インスタンスの画像上での2次元の関係をある程度反映している必要がある。画像処理部から得られるサポート値をもとに、各トークン毎に、そのトークンに対応する対象物モデルの候補（以下単に対象物候補と呼ぶ）のリストを定める。また、トークン間の2次元の関係も、サポート値をもとに定める。この他に、3次元位置関係を2次元位置関係に変換するために、視線の方向の情報が必要であるが、現在のシステムでは、常に視線は水平方向を仮定している。

次に、得られたトークンのデータをから、トークンをノード、トークン間の2次元の関係をアークとしたグラフを考える。そして、ノードにその



トークンの対象物候補をそれぞれ割り当てて、must-be の位置関係知識に反している組合せを棄却する。このようにして得られた、ノードに対する対象物候補の割り当ては、画像全体として整合のとれた対象物インスタンスの配置を与えている。この配置に対して、may-be の位置関係知識をもとに、画像の構成を表すのに重要な 2次元の関係（アーク）のみを残して、残りの 2次元の関係を取り除く。これにより、グラフは画像中の特徴的な関係のみを表すようになる。最後に、perhaps の位置関係知識を用いて、画像処理部から受けとったトークンには含まれていない対象物モデルを推測し、グラフにその対象物モデルのノードと 2次元の関係のアークを追加する。

この処理は、整合ラベリング問題として定式化することができる [5]。整合ラベリング問題は NP-完全であることが知られており、この方法での計算量は、トークン数  $N$ 、対象物候補数  $K$  に対し最悪  $O(N^K)$  になる。しかし、本研究では、画像の大まかな構成を生成することを目的としているので、この処理でのトークン数は少数に抑えることができる。また、各トークンの対象物候補数もサポート値によりある程度抑えることができる。従って、実用的な時間で処理が行える。

### 3.3 画像構成仮説に基づいたトークン解釈処理

本節では、与えられた画像構成仮説をもとにして、トークン解釈を行う手法について述べる。最初に、トークンの一部に対して対象物モデルのインスタンス（以下単にインスタンス）を生成する。インスタンスは対象物ラベルとトークンの組及びそれに付随する情報で表され、そのトークンがその対象物として解釈されたことを示している。インスタンスに付随する情報とは、

- トークンの対象物モデルに関するサポート値

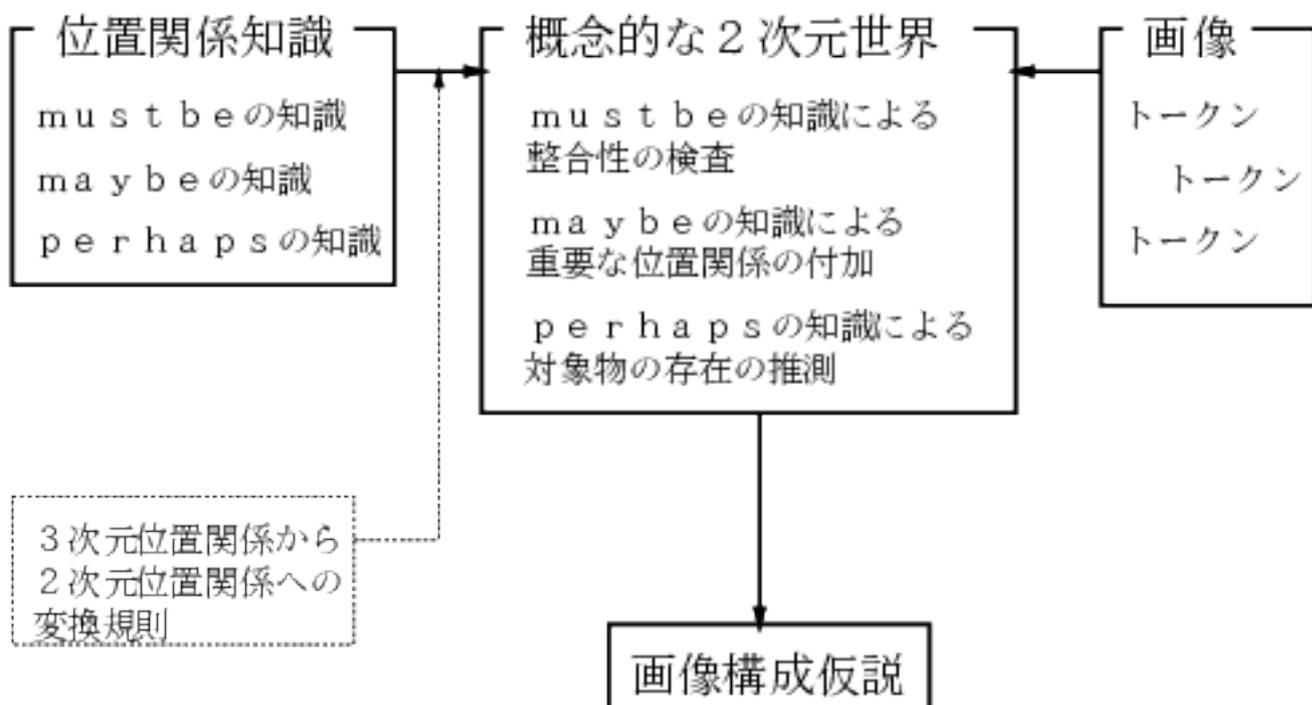


図 3-2: 画像構成仮説生成部の処理の流れ

- トークンと他のトークンとの2次元位置関係とそのサポート値の組のリスト

である。最初のインスタンスの生成は、もっとも信頼度の高い対象物を選ぶ。例えば、各トークン毎のサポート値が最大の対象物を選べば良い。最初のインスタンスが生成されると、それを手掛かりに、画像構成仮説の中に現れる対象物モデルに対し、対応するトークンの探索を行う。処理は、次のような流れで実行する。

1. 探索のための制約条件の生成

画像構成仮説をもとに制約条件を生成し、画像処理部に渡す。

2. インスタンスの生成

画像処理部から返されたトークンとサポート値のデータをもとに、トークンと対象物ラベルを結び付け、2次元位置関係とサポート値をつけてインスタンスとして生成する。

3. 衝突検出

画像中の同一のトークンに対するインスタンスの集合を検出し、衝突集合とする。

4. 衝突解消

各衝突集合毎に、最大のサポート値を持つインスタンスをそのトークンの解釈として採用する。

5. トークン解釈の更新

採用されたインスタンスをトークン解釈とする。もとのトークン解釈から、削除されたインスタンスとの関係を抹消する。

以上の処理を、トークン解釈が更新されなくなるまで繰り返す（図 3-3 参照）。

探索のための制約条件（以下単に制約条件）は、対象物ラベルと 2 次元位置関係の組で、画像構成仮説をもとに生成される。生成された制約条件は、画像処理部に送られる。画像処理部は、この制約条件を満たすトークンのリストを画像解釈部にサポート値をつけて返す。

画像処理部から返されたトークンは、インスタンス間の 2 次元位置関係とサポート値をつけて、インスタンスとして生成される。この時生成されるインスタンスは、一般に複数である。また、各々の制約条件からインスタンスが生成されるので、画像中の同一部分に対して複数のインスタンスが生成されることがある。これをトークン解釈の衝突と呼ぶ。トークン解釈の衝突は、別々の 2 次元の関係を持った同じ対象物モデルに対するインスタンスの場合と、矛盾したトークン解釈を含む場合とがあり、その解消が必要である。そのためまず、その時点でのトークン解釈のうち、衝突が生じているもの同士を集めて、衝突集合とする。現在のシステムでは、同一の識別子を持ったトークンに対する複数のインスタンスを矛盾としているが、実際には、トークンは領域の集合であるので、異なった識別子を持つトークン同士でも画像上で重なった部分を持つことがあり得る。

衝突の解消は各衝突集合毎に行われる。まず、衝突集合内で、同一の対象物に対するインスタンスを一つにまとめる。この際に、それぞれのインスタンスが持つ 2 次元位置関係をすべて引き継ぐ。これにより、衝突集合内では、一つの対象物に対して、高々一つのインスタンスのみが存在することになる。次に、このインスタンスそれぞれに対してインスタンスのサポート値  $SUPP_{inst}$  を計算する。これは、次式で計算する。

**begin**

$$SUPP_{inst} := W_{obj} * SUPP_{obj};$$

**for**  $i := 1$  **to**  $k$  **do**

$$SUPP_{inst} := SUPP_{inst} + (1 - SUPP_{inst}) * W_{rel} * SUPP_{rel_i};$$

**end.**

$SUPP_{rel_i}$  : インスタンスの2次元の関係の制約のサポート値

$k$  はそのインスタンスの持つ位置関係制約の数

$SUPP_{obj}$  : インスタンスの対象物に関する制約のサポート値

$W_{obj}$  : 対象物に関する制約のサポート値に対する重み

$W_{rel}$  : 2次元位置関係に関する制約のサポート値に対する重み

この式は、Dempster の結合規則 (Dempster's rule of combination) の応用である [4]。ここでは、サポート値をファジーな述語に対するメンバシップ関数の値とみなし、その統合を行っている。これは、トークン単体のみ、または、他のトークンとの関係からだけでも、そのインスタンスの対象物らしさは得られるが、両方が揃えば更に確からしさが増す、という考えに基づいている。即ち、上式では、

- どの一つのサポート値のみでも、インスタンスのサポート値を与えることができる。
- 多くの関係が成り立っていた場合、個々のサポート値が小さくても全体としてある程度のサポート値が得られる。

という特徴がある。

こうして得られたサポート値のうち、最大値を与えるインスタンスをトークンの解釈として採用する。最大のものが複数あった場合は、それらをすべてトークンの解釈として採用し、衝突の解消は次の処理まで延期する。

採用されたインスタンスにより、新たなトークン解釈を作る。以上を1サイクルとする。この処理を、トークン解釈が更新されなくなるまで続け

る。しかし、トークン解釈の更新が循環して起こり、処理が終了しない場合もある。これを避けるために、繰り返しが増すにつれて、 $W_{obj}$  を大きく、 $W_{rel}$  を小さくしていく方法をとっている。

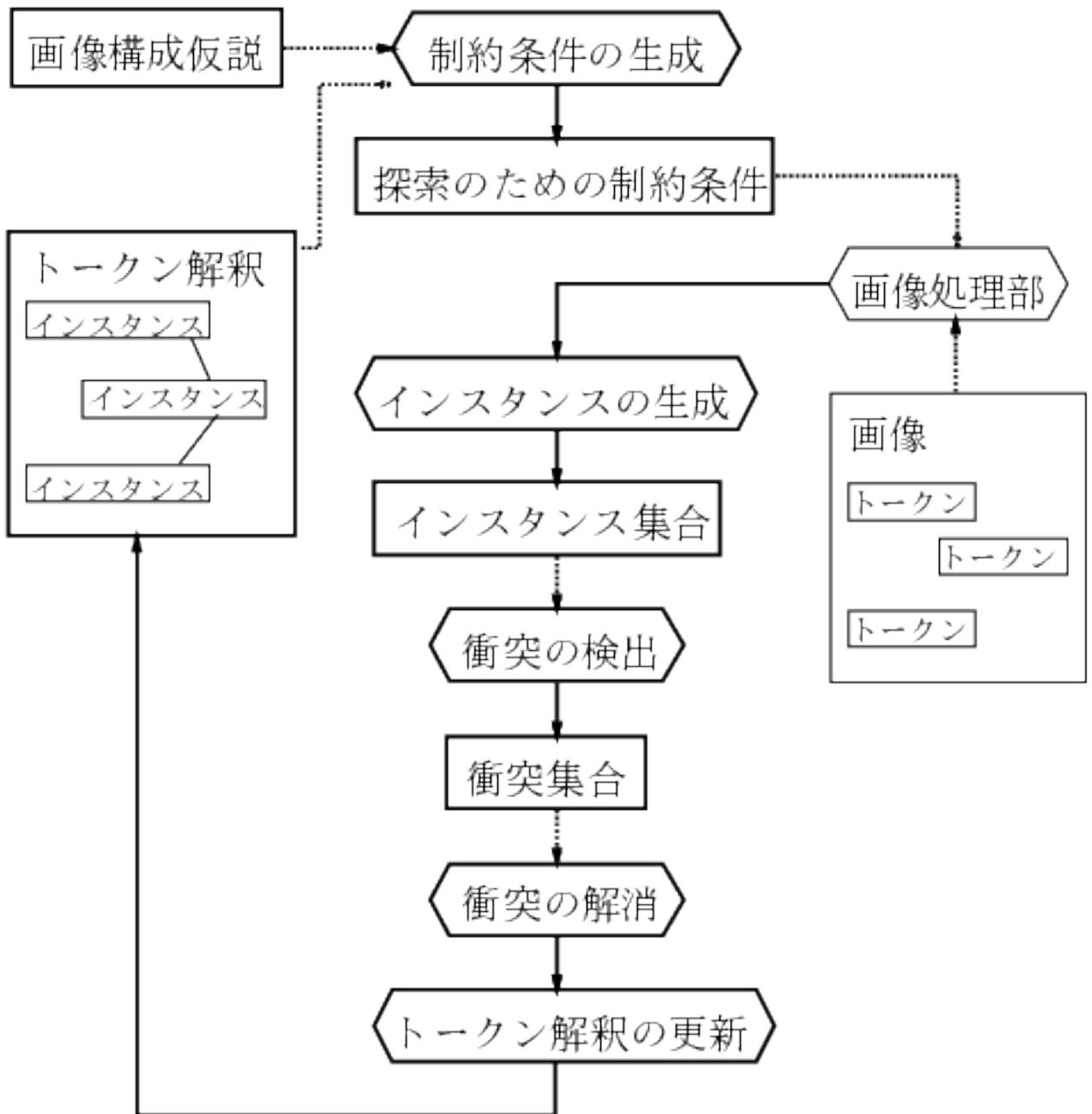


図 3-3: トークン解釈部の処理の流れ

## 4 実験結果および考察

### 4.1 画像構成仮説生成実験

位置関係知識を用いた画像構成仮説の生成を、実際の画像に対して行った。システムが持っている対象物とそのクラスを表 A-1 に、3次元位置関係知識を表 A-2 に、3次元位置関係から2次元位置関係への変換規則を表 A-3 に示す。

実験は電車の写っている3枚の風景画像に対して行った(図 A-1)。視線の方向は、すべて水平方向である。原画像は、スキャナで取り込んだカラー画像である。これを濃淡画像に変換して、局所的なヒストグラムを用いた領域分割を行った後、小領域を併合して領域分割画像を得た(図 A-2)。この際の領域分割のパラメータや小領域併合のパラメータは、結果の領域がほぼ画像の大まかな構成に対応するように定めた。この領域をそれぞれトークンとみなし、各トークンの対象物候補を、領域の色の平均値をもとに定めた(表 A-4)。また、トークン間の2次元位置関係も妥当だと思われるものを選んだ。ここでは、サポート値は直接利用しないので、算出は行わず、対象物候補、2次元位置関係は、リストにしてあらかじめ用意しておいた。

プログラムは Common Lisp (KCL) で記述されており、シークエント社の計算機 S-81 上で処理を行った。実験の結果を表 4-1 に示す。また、それぞれの画像に対して、生成された画像構成仮説の一部を図 A-3, A-4, A-5 に示す。

処理結果は、ほぼ想定した通りのものになった。いずれの場合も妥当な画像構成仮説が得られている。原画像を正しく反映していない画像構成仮説でも、トークンの位置関係のみからならば人間が見て納得できる画像構

成仮説を生成している。

原画像 train05 に対する処理では、可能な組合せ数が他の二つに比べ非常に多いが、生成された画像構成仮説の数は、他の二つとそれほど変わらない。また、処理時間も、組合せ爆発を起こすまでには至っていない。位置関係知識が適切な場合、生成される画像構成仮説の数は、システム内の対象物モデルの数と、画像中に現れる対象物インスタンスの数に依存すると考えられる。

この処理の問題点は、領域間の位置関係が、直接対象物間の位置関係を反映していなければならない点である。粗い画像分割を行うことで、この問題を回避しているが、常にこの仮定が成立するとは限らない。その場合には、処理を保留する必要がある。即ち、must-be の知識に反したノードが生じた場合、その中間構成を捨てずに処理を続けなければならない。しかし、この方法では、生成される画像構成仮説が爆発的に増えてしまう。どの時点で、中間構成を捨てるかが大きな問題になる。

また、この処理では、画像処理部から得られるサポート値は利用していない。最初に画像処理部から得られる初期的なトークン切り出しには誤りが多く、これによる対象物モデルに関するサポート値は、あまり重要視すべきではないと考えたためである。しかし、概整合ラベリング問題 [6] として扱えば、先の問題も含めて解決できる可能性があり、今後の検討課題と考えられる。

## 4.2 トークン解釈実験

画像構成仮説の一つを用いて、4.1 節の図 A-1 の 3 枚の画像に対して、トークン解釈を実際に行った。

まず、この対象画像を 4.1 節とは別の方法（領域成長法）で領域分割し、図 A-6 を得た。図 A-6 の領域は 4.1 節で用いた領域より細かく分割

表 4-1: 実験の結果

原画像	train05	train06	train07
トークン数	12	6	5
可能な組合せ数	27648	64	108
生成された画像構成仮説の数	10	6	8
処理時間 (秒)	19.95	2.85	4.77

されている。画像解釈部の実際の処理の中でも、この段階で対象となる領域は、モデル生成に用いた領域より詳細化されているため、領域数は増えるはずである。ここに示した全ての領域をトークン集合として、その解釈を行った。

このトークン集合に対して、2次元位置関係のサポート値を定義した。上下関係のサポート値  $SUPP_{above}$ ,  $SUPP_{below}$  は、領域の外接矩形の中心間距離を用いて算出した。距離のサポート値  $SUPP_{near}$  は、二つのトークンが接している時 1.0、接していない時 0.0 とした。包含関係の評価は、このように細分化された画像上では難しいため、ここでは利用していない。位置関係制約のサポート値の算出は、上下関係と距離のサポートの最小値とした。これは、上下関係、距離をファジー集合とみなした場合のファジー論理積になっている。上下関係、距離は3次元位置関係から変換されたもので、この二つの間は論理積で結ばれている。これは、一方でもサポート値が低ければ、この3次元位置関係が成立している可能性は低くなるためである。また、このトークンそれぞれに対して、対象物毎のサポート値を最低 0.4、最高 0.8 とし、この間を 0.1 毎に分けた 5 段階で人間が評価し、与えた。

使用したモデルは、図 A-3, A-4, A-5 に示した画像構成仮説に、領域の隣接関係 ( $t t_{near}$ ) を付加したものである。最初のインスタンスは、表 A-5 に示すものである。これは、図 A-2 と図 A-6 とでもっとも良く対応する領域を選んでいる。

実験結果を、表 4-2 に、トークン毎の対象物のサポート値と解釈結果を表 A-6, A-7, A-8 に、トークン解釈の結果を図 A-7, A-8, A-9 に示す。

トークン単体ではサポート値最大の対象物が複数あるもの（表 4-2 中の“複数候補を持つトークンの数”）が、位置関係を用いることにより、そのほとんどに対して、正しい解釈を行うことができている。train05 では、他に較べて正解数が少ないが、誤った解釈をされたトークンのほとんどは、

原画像中の砂浜を土手と解釈したもので、誤りとは言い切れない。

この処理の問題点としては、まず、処理時間がかかったことが挙げられる。処理対象の画像に沿ったモデルに基づいた処理であるから、処理の途中ではインスタンス同士が互いに解釈を強め合い、すぐにインスタンスの更新が起こらなくなるはずである。実際、生成されたインスタンスは、高々 2000 個程度であり、処理可能な個数である。原因は、制約条件をもとにトークンを選んでくる処理に時間が掛かっているためであった。これを改善するために、トークンのデータの持ち方を改善する予定である。

また、この処理では、「山の中に空がある」というような、矛盾した位置関係を排除できない。これは、二つの対象物間の関係を、直接トークンに適用しようとしたためである。対象物が多くトークンに分割されている場合には、モデルの持つ 2 次元の関係は、トークン間には直接は成り立たない。モデルを適用する際に、トークンを併合し、新たなトークンを生成する処理を加える必要がある。

得られた処理結果は、画像解釈部の次の段階で、画像構成仮説の評価と詳細化のために用いる。この際、モデルにない対象物に対しては、トークン解釈を保留しておくことが、以降の処理を行う上で重要になると思われる。このようなトークンを除外して、処理を進める方法も検討する必要がある。

表 4-2: 実験の結果

原画像	train05	train06	train07
トークンの数	42	42	29
生成されたインスタンスの数	1469	2234	364
生成された制約条件の数	360	383	110
正解数／複数候補を持つトークンの数	7/14	14/16	17/18
処理時間 (秒)	2389	2996	248

## 5 おわりに

本報告書では、ネットワーク分散型の画像認識システムにおける画像解釈部という位置づけのもとに、位置関係知識を用いた画像構成仮説の生成と、画像構成仮説をもとにしたトークン解釈について述べた。特に、位置関係知識を用いることによって、効率的に画像の構成の巨視的な情報、即ち画像構成仮説を取り出せることを示した。画像構成仮説は、3次元の知識をもとにしているため、画像構成仮説の詳細化が行えれば、最終的に、画像の3次元解釈が行えると考えられる。モデルを用いたトークン解釈には、まだ多くの問題点があるが、解釈処理自体は可能であることを示した。ここで用いた Dempster の結合規則は、同一の対象に対するいくつかのサポート値から、その対象の総合的なサポート値を算出する理論に基づいたもので、より広い応用が考えられる。

この手法での利点の一つは、画像構成仮説を用いたトークン切り出しの改善であるが、この部分はまだ実現されていない。また、トークンの対象物の評価値算出も自動的には行えず、実験が十分に行えなかった。処理方式の改善とともに、より多くの画像での実験が必要である。

この画像認識システムはまだ開発途上にあり、多くの課題を残している。画像解釈部においては、今後の課題として、

- トークン解釈処理の改善、及び高速化。
- 位置関係知識とトークン解釈結果を利用した画像構成仮説の生成。
- 位置関係知識の詳細化。他の位置関係の導入。

などが挙げられる。今後は、画像内の対象物の3次元位置関係の記述を行っていく予定である。

## 謝辞

本研究に際し、終始御指導、御鞭撻を賜わった池田教授に心から感謝の意を表します。

熱心な御指導、有益な示唆を下された美濃助教授に感謝の意を表します。

また、数々の御助言、御忠告を下された、広瀬助手、天野氏ほか、池田研究室の皆様に感謝の意を表します。

## 参考文献

- [1] Bruce A.Draper, Robert T.Collins, John Brolio, Allen R.Hanson, and Edward M.Riseman:  
The Schema System,  
International Journal of Computer Vision, 2 (1989), 209-250.
- [2] Yu-ichi Ohta:  
A Region-Oriented Image-Analysis System by Computer,  
Ph.D.thesis, Dept. of Information Science, Kyoto University,  
March, 1980.
- [3] 松山隆司, ビンセント・ハンク:  
画像理解システム SIGMA,  
情報処理学会論文誌 26-5(1985-9), 877-889.
- [4] 石塚満:  
エキスパートシステム構築のための Dempster&Shafer 理論の  
ファジィ集合への拡張,  
生産研究 34-7(1982-7), 312-315.
- [5] 西原清一, 原智亨, 池田克夫:  
拘束ネットワークを用いた整合ラベリング法,  
電子通信学会論文誌 67-7(1984-7), 745-752.
- [6] 西原清一, 松尾嘉和, 池田克夫:  
概整合ラベリング問題における併合法の最適化と効率評価  
人工知能学会誌 3-2(1988-3), 196-205.

## 実験資料

表 A-1: 画像解釈部の持つ対象物モデルとそのクラス

対象物ラベル	対象物クラス
sky	space
train	solid
mount	solid
bank	solid
rock	solid
grass	plane
water	plane
shadow	figure

表 A-2: 画像解釈部の持つ位置関係知識

	主体対象物	3次元の関係	関係対象物
must-be	solid	over	plane
	train	before	mount
	mount	behind	train
	bank	before	mount
	mount	behind	bank
	bank	under	train
	train	over	bank
may-be	space	on-over	plane
	plane	on-under	space
	space	on-over	solid
	solid	on-under	space
	plane	on-under	solid
	solid	on-over	plane
	plane	face	plane
	solid	before	solid
	solid	behind	solid
	train	on-over	bank
	bank	on-under	train
	train	on-over	shadow
	shadow	on-under	train
	bank	on-over	grass
	grass	on-under	bank
	bank	on-over	water
water	on-under	bank	
perhaps	train	on-over	bank
	train	on-over	shadow
	bank	on-over	water

表 A-3: 3次元位置関係から2次元位置関係への変換規則

3次元位置関係	変換された2次元の関係
(space over plane)	(above nil t)
(space over solid)	(above nil t)
(space over figure)	(above nil t)
(space on-over plane)	(above nil near)
(space on-over solid)	(above nil near)
(plane under space)	(below nil t)
(plane under solid)	(below nil t) (below outer t)
(plane on-under space)	(below nil near)
(plane on-under solid)	(below nil near) (below outer near)
(plane face plane)	(t t near)
(plane on-under figure)	(t outer near)
(solid under space)	(below nil t)
(solid on-under space)	(below nil near)
(solid over plane)	(above nil t) (above inner t)
(solid on-over plane)	(above nil near) (above inner near)
(solid before solid)	(below nil t) (below inner t)
(solid behind solid)	(above nil t) (above outer t)
(solid over solid)	(above nil t)
(solid under solid)	(below nil t)
(solid on-over solid)	(above nil near)
(solid on-under solid)	(below nil near)
(solid on-over figure)	(above nil near)
(solid on-under figure)	(t outer near)
(figure under space)	(below nil t)
(figure on-under solid)	(below nil near)
(figure on-over solid)	(t inner near)
(figure on-over plane)	(t inner near)



train05

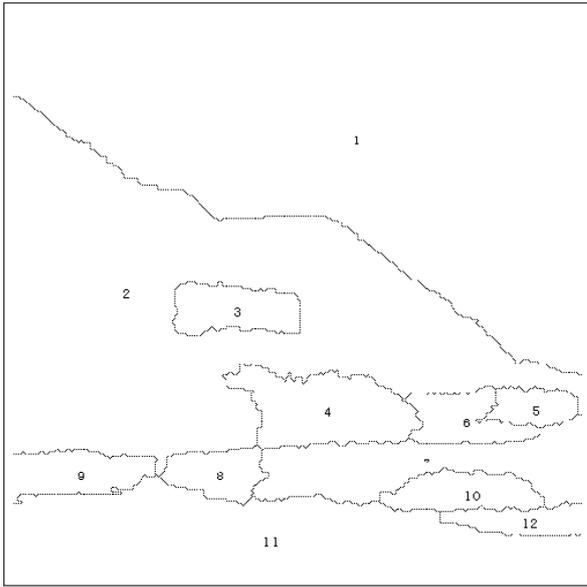


train06

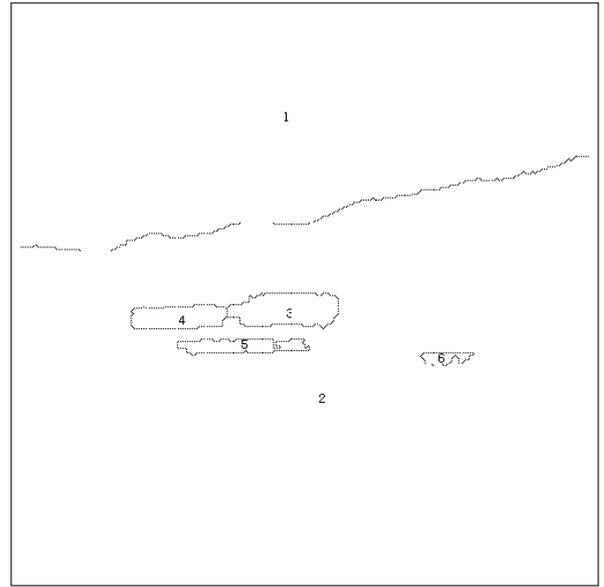


train07

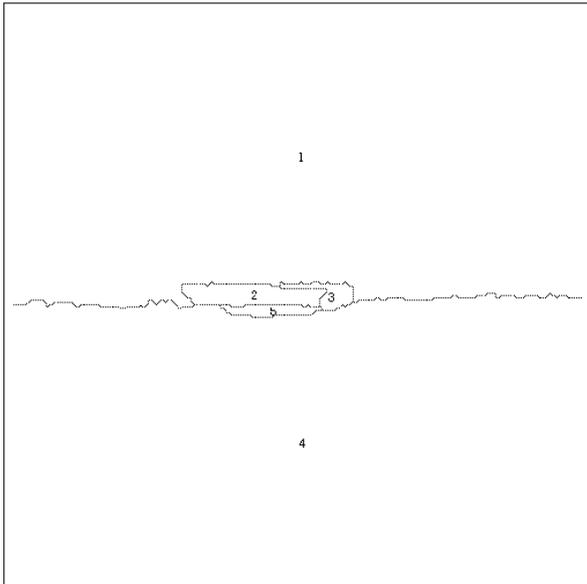
図 A-1: 原画像



train05



train06



train07

図 A-2: 画像構成仮説生成に用いるトークンの画像

表 A-4: 各トークンの対象物候補

原画像	token	対象物候補
train05	:1	sky water train
	:2	mount grass
	:3	sky water train
	:4	bank water
	:5	bank water
	:6	bank water
	:7	sky water
	:8	bank water
	:9	mount grass bank water
	:10	rock shadow
	:11	water sky
	:12	sky water train
train06	:1	sky water
	:2	mount grass
	:3	train bank
	:4	train bank
	:5	shadow rock
	:6	train bank
train07	:1	sky water train
	:2	sky water train
	:3	sky water train
	:4	grass mount
	:5	shadow rock

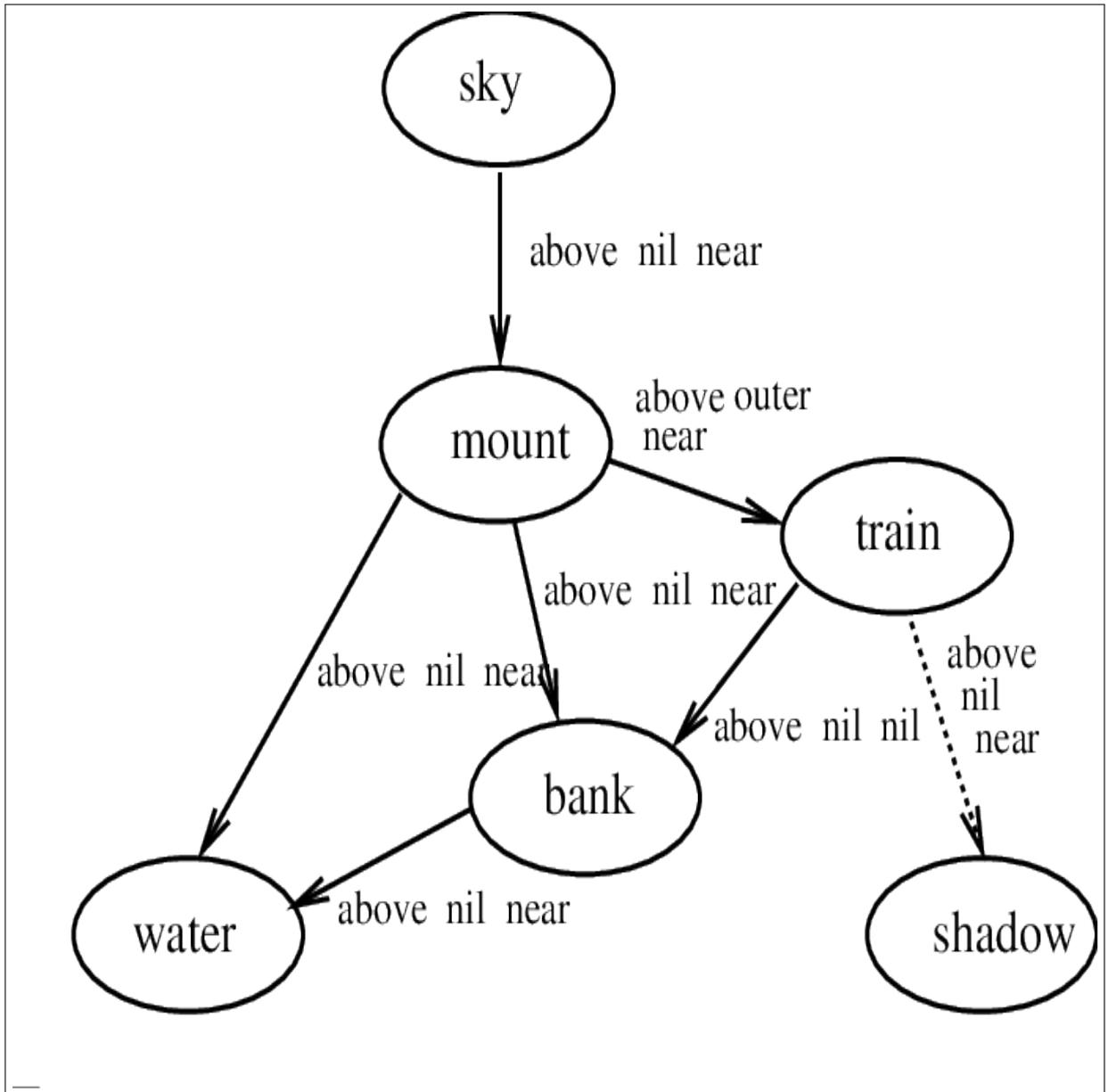


図 A-3: 生成された画像構成仮説 train05

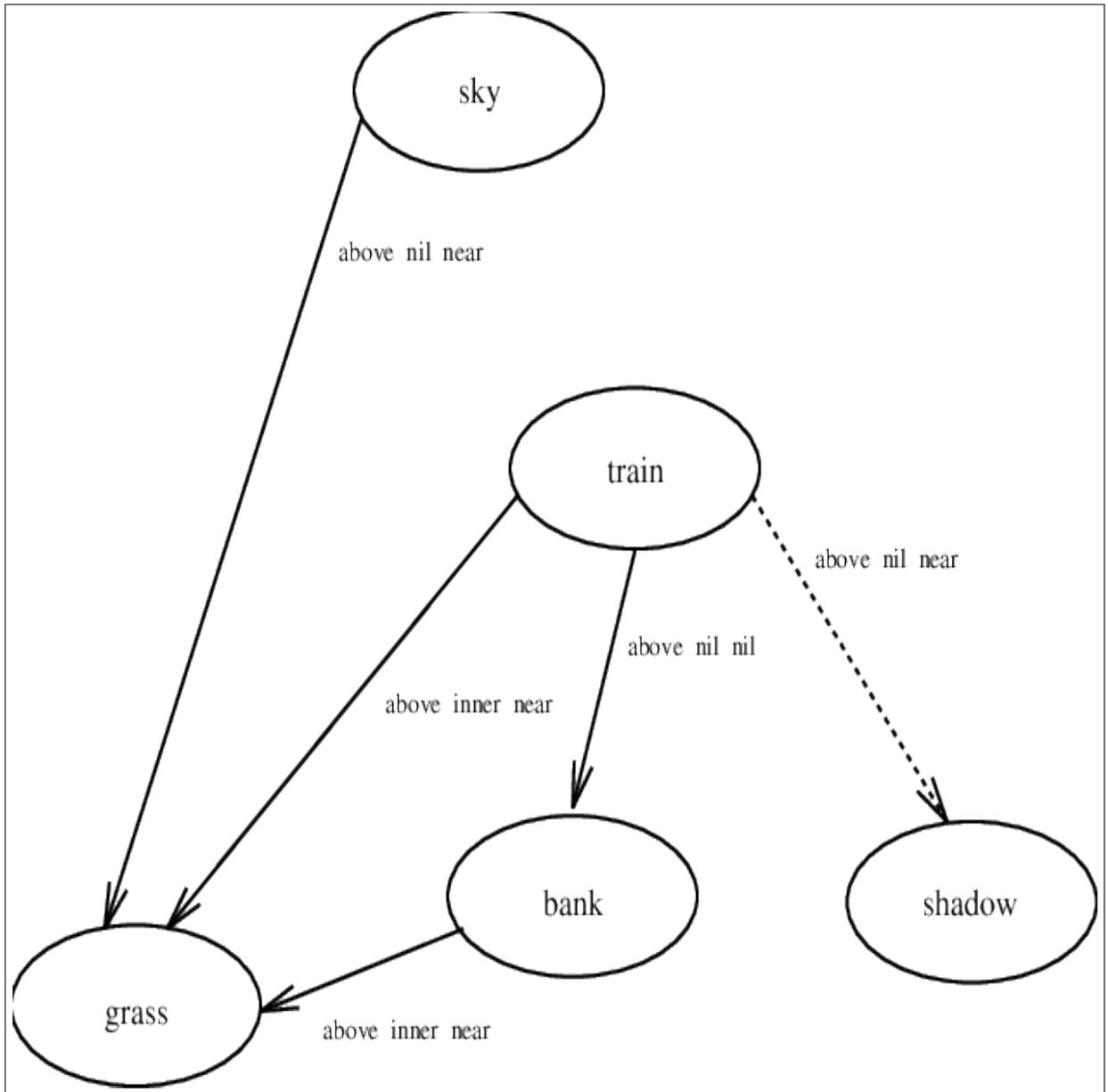


図 A-4: 生成された画像構成仮説 train06

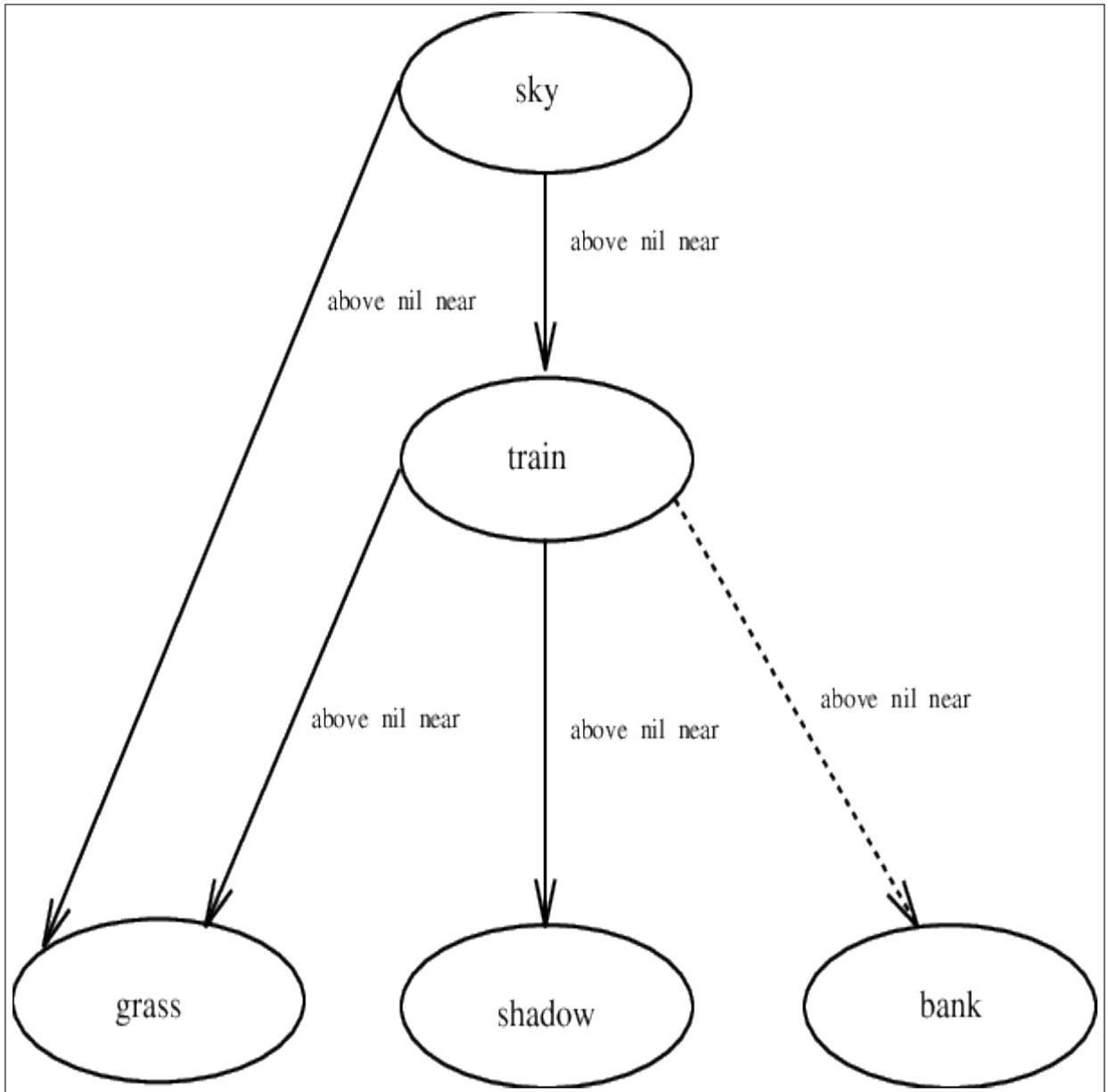
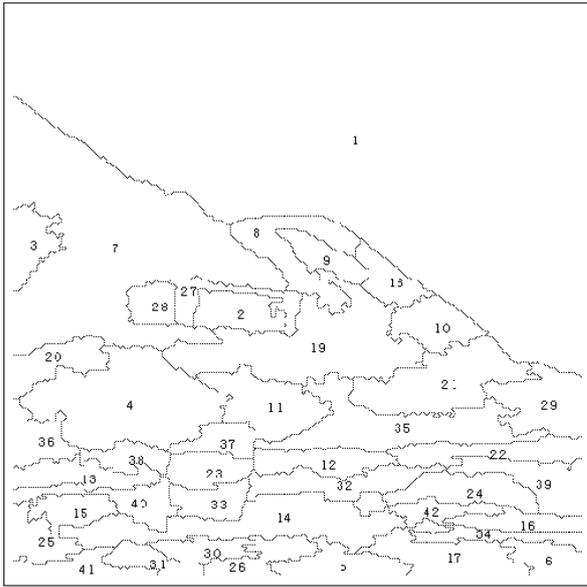
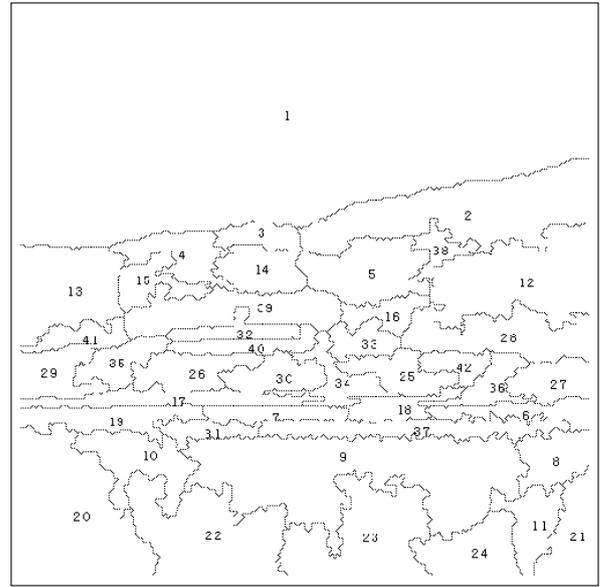


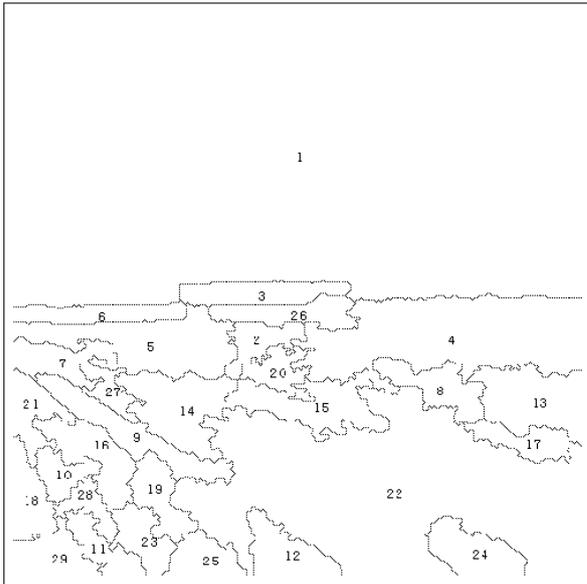
図 A-5: 生成された画像構成仮説 train07



train05



train06



train07

図 A-6: トークン解釈に用いるトークンの画像

表 A-5: 最初に生成されるインスタンス

原画像	token	対象物
train05	:1	sky
	:7	mount
	:2	train
	:35	bank
	:14	water
	:24	shadow
train05	:1	sky
	:9	grass
	:39	train
	:42	bank
	:40	shadow
train07	:1	sky
	:22	grass
	:3	train
	:26	shadow

表 A-6: train05 に使ったトークンの評価値と解釈結果

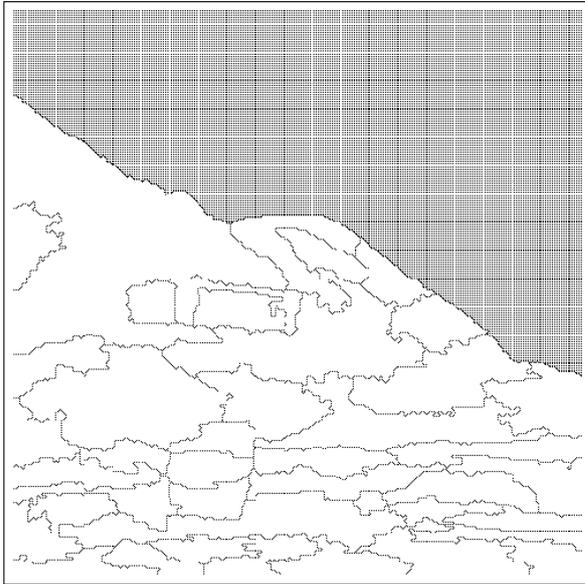
token	sky	train	mount	shadow	water	bank	result
:1	0.8	0.7	0.6	0.5	0.7	0.5	sky
:2	0.7	0.8	0.7	0.4	0.5	0.5	grass
:3	0.4	0.7	0.8	0.6	0.7	0.6	train
:4	0.6	0.4	0.5	0.7	0.7	0.6	grass
:5	0.7	0.6	0.5	0.5	0.8	0.5	grass
:6	0.6	0.7	0.5	0.6	0.8	0.4	grass
:7	0.6	0.4	0.8	0.6	0.7	0.4	grass
:8	0.7	0.7	0.7	0.5	0.7	0.7	grass
:9	0.4	0.6	0.7	0.7	0.4	0.4	grass
:10	0.6	0.7	0.7	0.4	0.6	0.7	grass
:11	0.5	0.6	0.4	0.7	0.5	0.8	grass
:12	0.7	0.7	0.4	0.7	0.8	0.6	grass
:13	0.5	0.7	0.5	0.5	0.8	0.4	bank
:14	0.7	0.7	0.5	0.4	0.8	0.7	grass
:15	0.7	0.7	0.6	0.4	0.7	0.7	grass
:16	0.8	0.5	0.6	0.5	0.8	0.7	grass
:17	0.4	0.5	0.6	0.5	0.8	0.7	bank
:18	0.7	0.4	0.7	0.5	0.6	0.5	grass
:19	0.4	0.6	0.8	0.7	0.7	0.7	bank
:20	0.4	0.7	0.4	0.5	0.4	0.7	grass
:21	0.6	0.4	0.4	0.6	0.7	0.7	grass
:22	0.7	0.5	0.4	0.5	0.8	0.6	grass
:23	0.7	0.7	0.5	0.7	0.8	0.6	grass
:24	0.4	0.4	0.5	0.8	0.5	0.6	grass
:25	0.7	0.4	0.7	0.5	0.7	0.7	grass
:26	0.6	0.4	0.6	0.4	0.8	0.5	grass
:27	0.7	0.8	0.7	0.4	0.6	0.6	grass
:28	0.5	0.8	0.5	0.4	0.4	0.6	grass
:29	0.5	0.4	0.6	0.6	0.4	0.8	grass
:30	0.5	0.4	0.5	0.7	0.8	0.4	grass
:31	0.5	0.7	0.6	0.8	0.7	0.8	bank
:32	0.6	0.6	0.6	0.6	0.7	0.6	train
:33	0.7	0.6	0.6	0.4	0.8	0.4	grass
:34	0.6	0.4	0.7	0.7	0.8	0.6	bank
:35	0.4	0.6	0.5	0.6	0.6	0.8	bank
:36	0.5	0.6	0.6	0.4	0.4	0.7	grass
:37	0.6	0.4	0.6	0.6	0.6	0.7	bank
:38	0.6	0.6	0.6	0.5	0.7	0.7	sky
:39	0.7	0.5	0.6	0.5	0.8	0.7	train
:40	0.7	0.6	0.7	0.4	0.7	0.7	shadow
:41	0.5	0.5	0.4	0.6	0.7	0.7	bank
:42	0.6	0.6	0.5	0.7	0.8	0.4	bank

表 A-7: train06 に使ったトークンの評価値と解釈結果

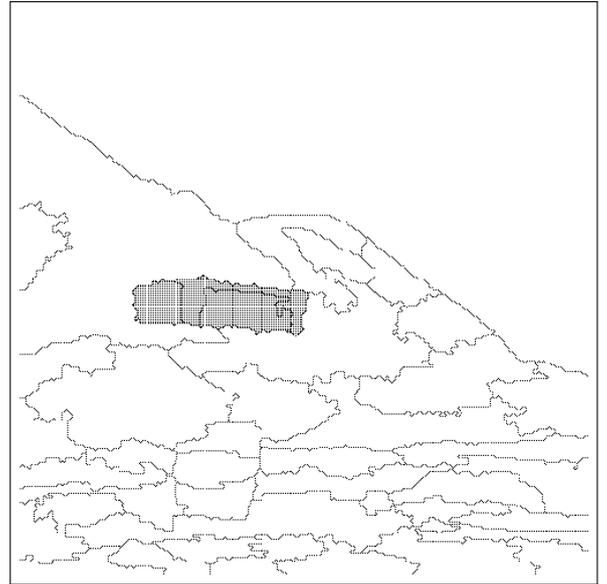
token	sky	train	grass	shadow	bank	result
:1	0.8	0.7	0.6	0.6	0.4	sky
:2	0.7	0.4	0.7	0.6	0.4	grass
:3	0.7	0.7	0.7	0.7	0.6	train
:4	0.7	0.4	0.7	0.6	0.5	grass
:5	0.7	0.6	0.7	0.6	0.5	grass
:6	0.6	0.6	0.7	0.4	0.4	grass
:7	0.7	0.4	0.7	0.6	0.6	grass
:8	0.6	0.6	0.8	0.6	0.7	grass
:9	0.5	0.7	0.8	0.5	0.6	grass
:10	0.5	0.7	0.8	0.7	0.6	grass
:11	0.7	0.4	0.8	0.5	0.5	grass
:12	0.7	0.4	0.7	0.6	0.6	grass
:13	0.7	0.6	0.7	0.5	0.7	grass
:14	0.7	0.4	0.7	0.5	0.5	grass
:15	0.7	0.6	0.7	0.7	0.5	grass
:16	0.7	0.5	0.7	0.5	0.4	grass
:17	0.6	0.5	0.7	0.4	0.7	bank
:18	0.5	0.6	0.7	0.4	0.5	grass
:19	0.5	0.6	0.7	0.6	0.8	bank
:20	0.6	0.5	0.8	0.5	0.7	grass
:21	0.6	0.5	0.8	0.6	0.4	grass
:22	0.7	0.7	0.8	0.6	0.7	grass
:23	0.7	0.4	0.8	0.6	0.7	grass
:24	0.6	0.5	0.8	0.6	0.5	grass
:25	0.6	0.7	0.8	0.6	0.7	grass
:26	0.6	0.5	0.8	0.6	0.4	grass
:27	0.7	0.5	0.8	0.4	0.7	grass
:28	0.7	0.6	0.8	0.6	0.5	grass
:29	0.6	0.7	0.8	0.6	0.7	grass
:30	0.7	0.5	0.8	0.7	0.4	grass
:31	0.7	0.6	0.7	0.7	0.8	bank
:32	0.5	0.8	0.5	0.5	0.4	train
:33	0.7	0.6	0.7	0.5	0.6	grass
:34	0.4	0.7	0.7	0.5	0.7	bank
:35	0.6	0.6	0.6	0.6	0.8	bank
:36	0.4	0.4	0.7	0.7	0.5	grass
:37	0.5	0.6	0.7	0.5	0.8	bank
:38	0.7	0.7	0.7	0.4	0.4	train
:39	0.4	0.7	0.5	0.7	0.7	train
:40	0.5	0.5	0.6	0.8	0.5	shadow
:41	0.7	0.6	0.7	0.5	0.6	grass
:42	0.4	0.7	0.6	0.5	0.7	bank

表 A-8: train07 に使ったトークンの評価値と解釈結果

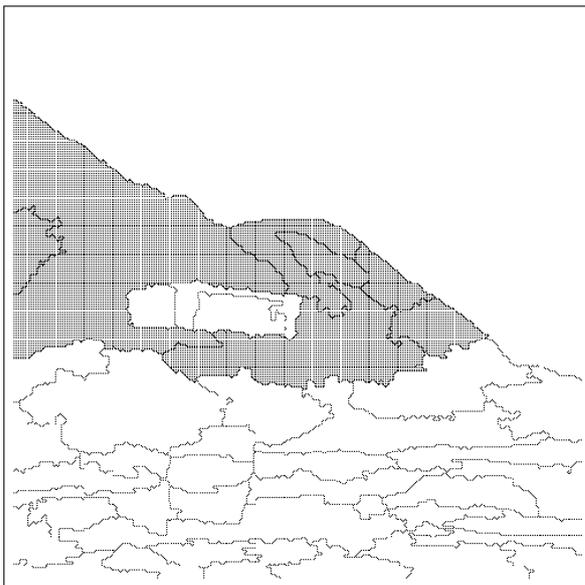
token	sky	train	grass	shadow	bank	result
:1	0.8	0.6	0.7	0.6	0.7	sky
:2	0.6	0.5	0.8	0.5	0.7	grass
:3	0.8	0.8	0.4	0.5	0.5	train
:4	0.4	0.7	0.8	0.6	0.7	grass
:5	0.6	0.7	0.7	0.4	0.5	train
:6	0.7	0.6	0.8	0.7	0.6	grass
:7	0.5	0.5	0.8	0.5	0.6	grass
:8	0.7	0.5	0.8	0.6	0.4	grass
:9	0.6	0.4	0.7	0.6	0.7	grass
:10	0.7	0.4	0.7	0.7	0.7	grass
:11	0.7	0.5	0.7	0.7	0.7	grass
:12	0.7	0.4	0.7	0.6	0.7	grass
:13	0.7	0.4	0.8	0.4	0.6	grass
:14	0.7	0.4	0.8	0.6	0.7	grass
:15	0.5	0.6	0.8	0.4	0.7	grass
:16	0.5	0.7	0.8	0.4	0.7	grass
:17	0.6	0.5	0.8	0.7	0.5	grass
:18	0.5	0.4	0.8	0.7	0.7	grass
:19	0.5	0.4	0.7	0.7	0.7	grass
:20	0.7	0.6	0.8	0.4	0.7	grass
:21	0.5	0.6	0.7	0.5	0.7	grass
:22	0.7	0.4	0.8	0.5	0.6	grass
:23	0.5	0.7	0.8	0.7	0.4	grass
:24	0.5	0.6	0.8	0.5	0.4	grass
:25	0.6	0.7	0.7	0.7	0.7	grass
:26	0.4	0.4	0.5	0.8	0.4	shadow
:27	0.7	0.6	0.8	0.4	0.4	grass
:28	0.6	0.5	0.7	0.4	0.7	grass
:29	0.5	0.6	0.8	0.7	0.7	grass



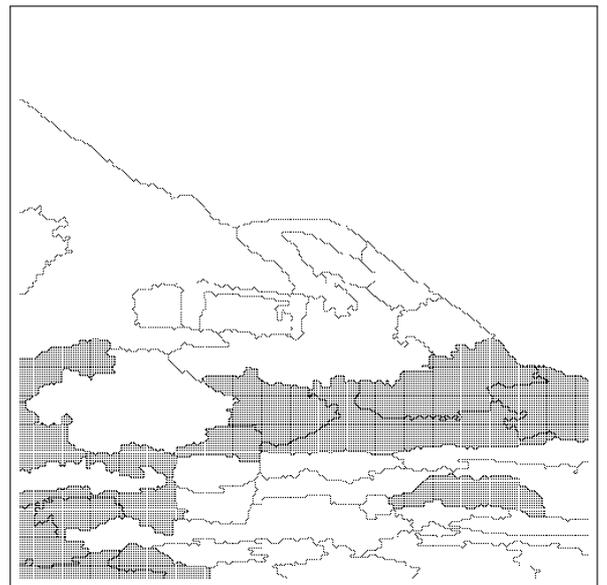
空と解釈されたトークン



電車と解釈されたトークン

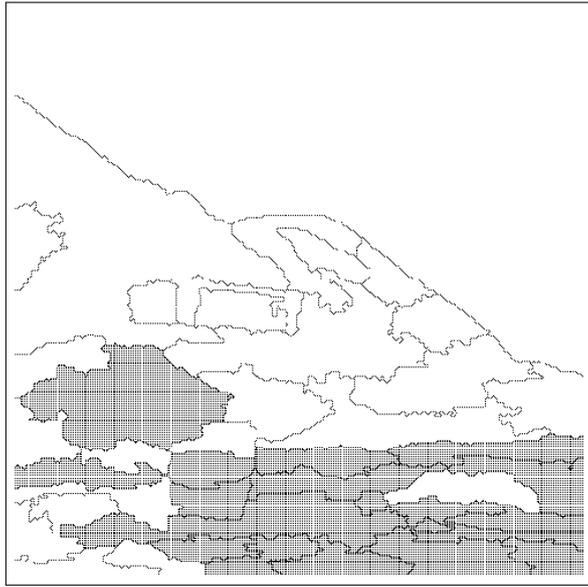


山と解釈されたトークン



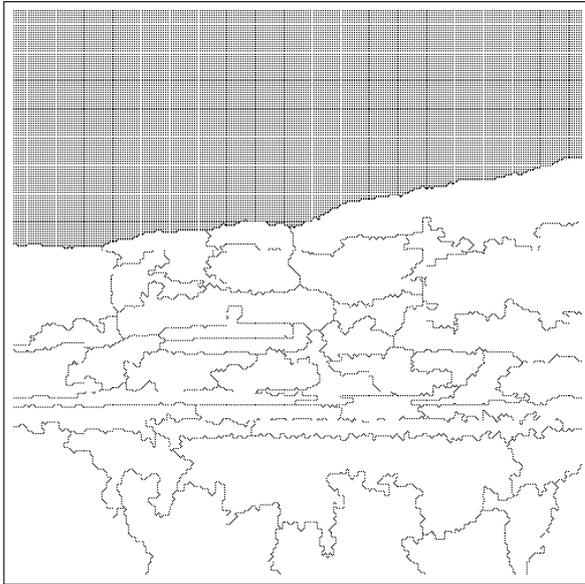
土手と解釈されたトークン

train05 のトークン解釈結果 (続く)

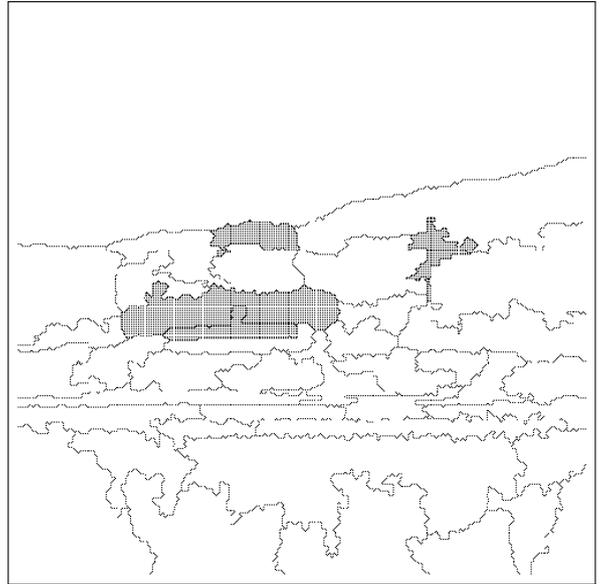


水面と解釈されたトークン

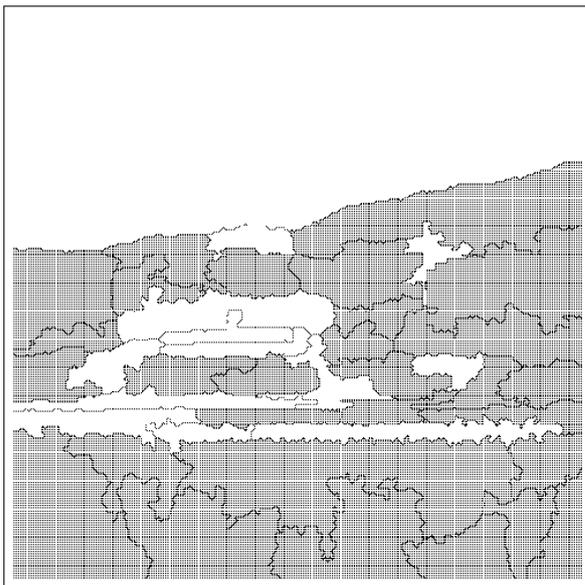
図 A-7: train05 のトークン解釈結果



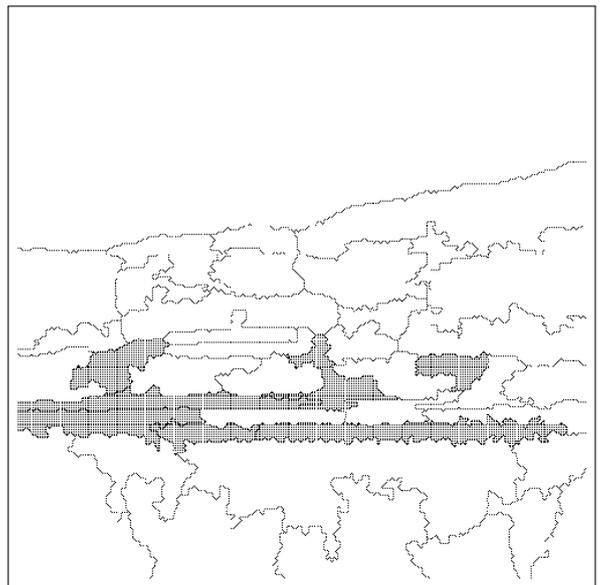
空と解釈されたトークン



電車と解釈されたトークン

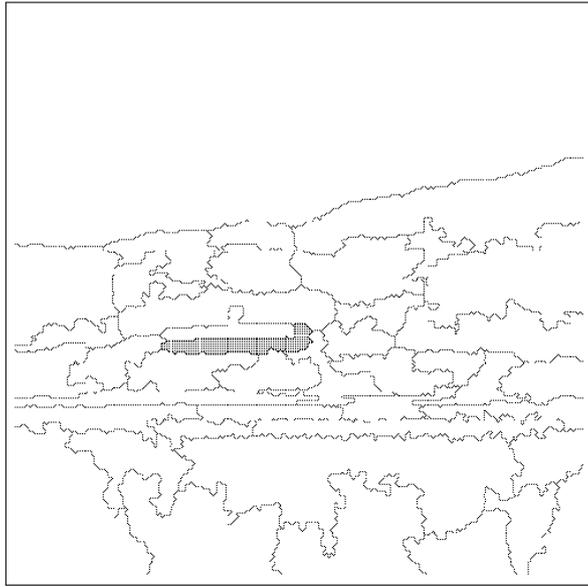


草地と解釈されたトークン



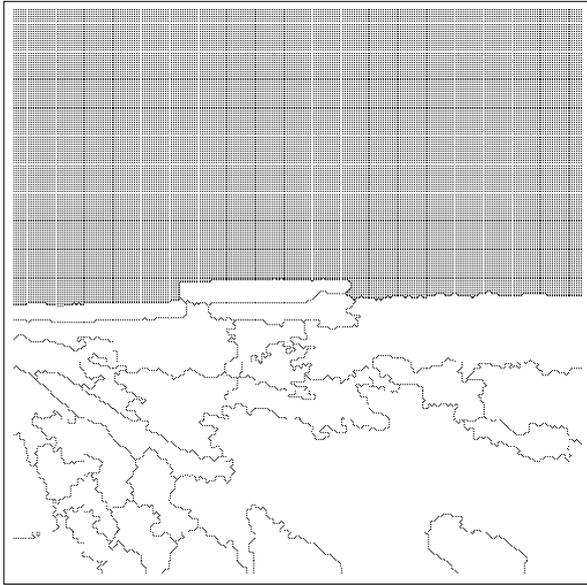
土手と解釈されたトークン

train06 のトークン解釈結果 (続く)

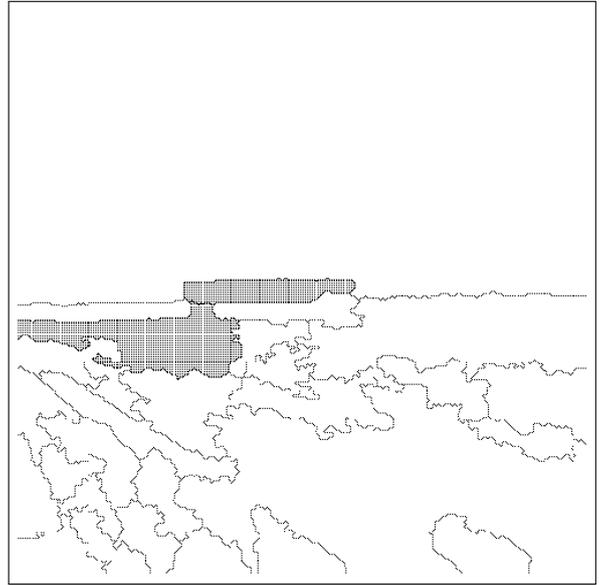


影と解釈されたトークン

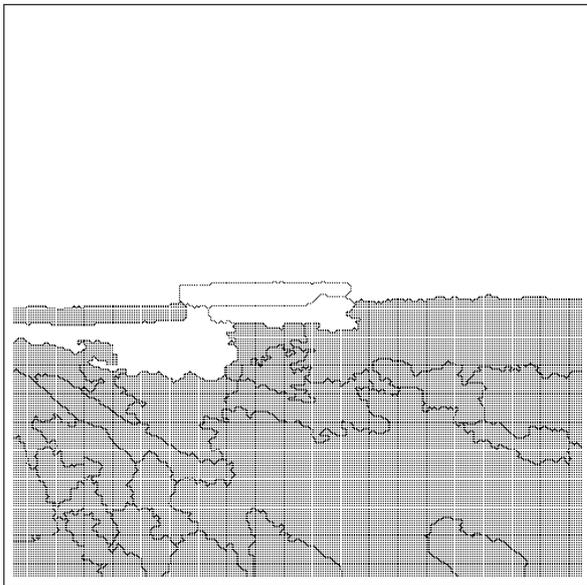
図 A-8: train06 のトークン解釈結果



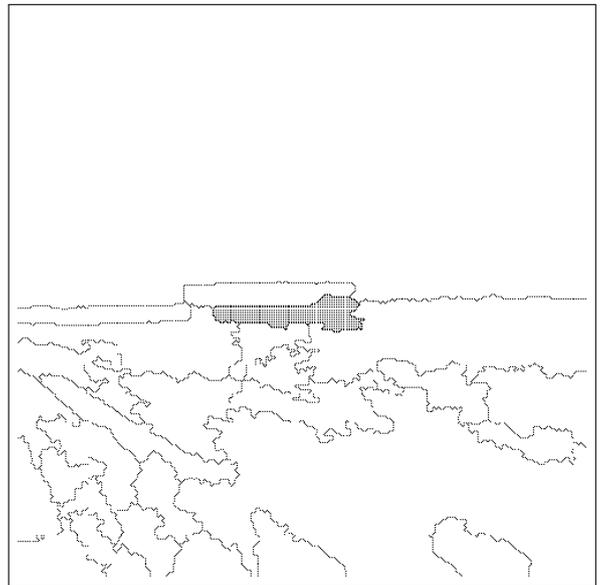
空と解釈されたトークン



電車と解釈されたトークン



草地と解釈されたトークン



影と解釈されたトークン

図 A-9: train07 のトークン解釈結果