

## 図形描画とテキスト入力を用いた力学に関する質問応答システム

五十嵐 健夫<sup>†</sup> 横野 光<sup>‡</sup> 岩根 秀直<sup>‡</sup>

**概要.** 数式処理システムをバックエンドとして力学の問題を解くアプリケーションを提案する。物理演算を使って問題を解くアプリケーションとしては、状況を示す図を GUI で描いて、それを初期条件として数値シミュレーションを実行し、結果をアニメーションとして提示するものが複数存在する。しかし、方程式を立てて解析的に変数間の関係を解くようなものは存在しない。一方、方程式を立てて解析的に解くシステムとして、数式処理システムがあるが、コマンドラインインタフェースのみで、図を扱うことはできない。本研究では、図形的な情報を GUI で指定して、記号的な情報をコマンドラインで入力するようなユーザインタフェースを提案する。特に、人間にとって自然なインタラクションを実現することを目的として、入力が曖昧性がある場合に、それらをどのように処理するかに関心を当て議論を行う。

### 1 はじめに

究極のユーザインタフェースの一つとして自然言語によるインタラクションが挙げられる。すなわち、人間が他の人間に、口頭や文字で説明するように、自然言語で計算機に指示を出して、操作を行うというものである。古くはアップルの Knowledge Navigator [2]などが有名であるが、近年、アップルの Siri やマイクロソフト社の Cortana のような、会話エージェントが注目を集め始めている。背景としては、インターネットやソーシャルメディアの普及により、会話に必要な知識（ビッグデータ）を集めることが用意になってきていることが挙げられる。しかし、このような現在主流の会話型のインタフェースは主にテキストだけのやり取りを利用しており、エージェントとのやり取りに図形的な情報を利用しているものはあまりない。またアプリケーションとしても、情報検索や簡単なコマンド発行が主であり、複雑な問題を解くようなものは少ない。本研究の長期的な目標は、コンピュータと人間間のコミュニケーションにおいて人間同士のような自然なコミュニケーションを実現することを目的として、1)自然言語だけでなく図形的な情報を扱う枠組みを確立すること、2)そのようなインタフェースを通じて、ある程度難しい問題のようなアプリケーションを実現すること、である。

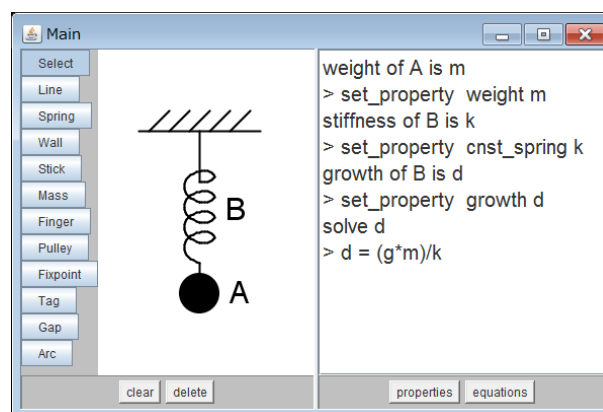


図 1. 提案システムの画面例。

上記の目標に向けた第一歩として、本稿では、数式処理システムをバックエンドとして力学の問題を解くアプリケーションについて紹介する。図 1 に画面例を示す。ユーザは左側のパネルに図を描き、右側のパネルでテキスト入力によって計算機に指示を与える。物理演算を使って問題を解くアプリケーションとしては、図で描かれた状況を GUI で描いて、それを初期条件として数値シミュレーションを実行し、結果をアニメーションとして提示するものが複数存在する [6]。しかし、方程式を立てて解析的に変数間の関係を解くようなものは存在しない。一方、方程式を立てて解析的に解くシステムとして、数式処理システムがあるが、コマンドラインインタフェースのみで、図を扱うことはできない。本研究では、図形的な情報を GUI で指定して、記号的な情報をコマンドラインで入力するようなインタフェースを提案する。

Copyright is held by the author(s).

<sup>†</sup> 東京大学

<sup>‡</sup> 株式会社富士通研究所

人間にとって自然なインタラクションを実現するにあたって重要なことは、曖昧性を許容することである。今回のアプリケーションの場合、必要な条件をすべて正しく明示的に与えれば、数式処理システムによって正しい解が得られることはある意味自明であるといえる。しかし、「必要な条件をすべて正しく明示的に」与えるためには、数式処理システムの動作の内部を理解し、計算機にわかりやすい形で入力を行わなければならない、人間にとって「自然な」インタラクションとは言い難い。一方、人間が他の人間に、問題を説明する場合には、細部を省略し、要点だけを伝えるだけで十分である。そのような指示には、計算機の側から見ると多分に「曖昧性」が含まれており、その曖昧性を解消することが重要となる。自然言語処理の分野では、このような曖昧性の解消の問題が強く意識されており、多くの研究がおこなわれている。一方、人間による図的な入力については、既存の計算機アプリケーションのほとんどが曖昧性のない入力を前提としており、曖昧性をどのように解消していくかについての議論はあまりなされていない。本研究の貢献は、このような問題に焦点を当てた点にある。

本研究では、力学の典型的な問題の例として、大学センター試験入試の問題を取り上げて議論を行う。詳細はのちに述べるが、これらの入試問題に表面的に表れている情報だけでは、数式処理システムで直接問題を解くには情報が不足しており、必要な情報をなんらかの形で推測する必要がある。今回のシステムではこのような問題を解決するために、1) 図に書かかれている情報とテキストで入力された情報を統合することによる、および2) 数式処理エンジンを外部モジュールとして利用し、さまざまな入力を数式処理システムに与えて出力をチェックする、というような動作をする高階処理モジュールを導入する、3) 入試問題において前提とされることの多い規則を明示的にヒューリスティクスと入れておく、といった3つの手法を提案している。それぞれの手法の詳細は今回のアプリケーションに特有のものであるが、考え方は他の「曖昧性を許容する自然なインタラクション」の実現に応用可能であると考えている。

## 2 関連研究

物理シミュレーションは確立された技術であり、自動車の設計やロボットの制御などに広く使われている。最近では、初心者でも使えるツールとして、画面上に2次元でおもりやバネなどの絵を描くと、そのまま物理シミュレーションが適用されてアニメ

ーションが提示されるものなども出てきている[6]。これらのツールでは、主にグラフィカルユーザインタフェースが利用されており、図形的な情報はマウスを用いた操作で図形的に指定し、物体の質量や弾性といったパラメータはダイアログボックスやパネルなどを用いて入力する。コマンドは、ボタンをおしたりメニューから選択することで実行される。

方程式を立てて解析的に解くシステムとして、*mathematica* や *maple* のような数式処理システムがある[10]。基本的な使い方としては、キーボードから対話的に式や変数などを入力すると、システムからの結果がテキストで返される。計算の結果をグラフを出力することもできるが、あくまで出力であり、入力として使うものではない。ユーザインタフェースとしては、コマンドラインインタフェースが使われており、図を入力として扱うことはできない。逆に図形に特化したものとして *Cindellera* などがある[3]。これは制約付き描画システムであり、図形に対して制約を指定すると、ユーザのドラッグ操作に追従して制約を満たすように図形がアップデートされる。しかし、テキスト入力による対話形式のインタラクションはサポートされていない。

図的な情報を扱ったインタラクションに関する研究としては、手書きスケッチで図形を描くと自動的に物理シミュレーションを適用してアニメーションを生成するもの[5]や、手書きで図形と数式を描くとその数式に従って図形がアニメーションするもの[7]などが提案されている。ただし、これらのアプリケーションでは、(手書き入力の認識以外) ユーザの入力に曖昧性は許されておらず、処理はすべて決定論的に行われている。本研究の貢献は、曖昧性のある入力を扱うための手法を提案している点にある。

視覚的な入力と言語的な入力を受け取り、曖昧性を解消するものとして、*put that there*[1]に代表されるようなマルチモーダルインタフェースにおける曖昧性解消がある[12]。これらの研究で扱われているものは、対象物をジェスチャーで指し示すポイントイングと、音声入力で動作を指定するコマンド入力の組み合わせであり、曖昧性解消は主に、これらのジェスチャーや音声の認識誤りの解消を対象としていた。本研究では、ユーザの描いた絵とテキスト入力の間での曖昧性解消を対象としており、問題の粒度が異なるといえる。

画像理解と自然言語処理に関する研究としては、たとえば、自然言語によって表現されたシーンに対応する3次元CGを自動生成する手法や[4]、位置関係を表す自然言語表現から画像中の位置関係を推測する手法など[14]が研究されている。本研究は、これらの先行研究のように画像を生成したり検索したりするだけでなく、力学の問題を解くというアプ

リケーションにつながっているところに新規性がある。

自然言語処理によってシーンを記述し、論理的な推論を行うシステムとして有名なものに SHRDLU がある[13]。また、最近の試みとして、物理の入試問題を自然言語で記述し、物理エンジンを利用して解くという試みも存在する[15]。本研究の長期的な目標は、このような問題解決システムに図形的な入出力を加えて、より自然で柔軟なインタラクションを実現することである。

### 3 ユーザインタフェース

図1に提案システムの画面スナップショットを示す。左側がキャンバスとなっており、ユーザはここに図を描く。右側がコマンドラインインタフェースとなっており、ユーザはここにコマンドを入力する。計算の結果もここに提示される。

キャンバスは標準的なドローエディタのように動作する。ユーザは、ツールパネルから描くオブジェクトを選んで、マウスクリックやマウスドラッグで場所や形状を指定する。特徴的な点としては、重りやバネといった物理的なコンポーネントを表すオブジェクトの他に、ラベルを表すものや、長さを示すものが用意されている。通常はこのようプロパティ情報はオブジェクトの中に隠されていて、ダイアログボックスやプロパティパネルの該当する部分を探してみなければならないが、必要な値だけを直接オブジェクトとして図中に含めることで、一覧性を確保することができる。

コマンドラインインタフェースは、キーボードでテキストを入力することでシステムに指示を出すことができる。命令をただでさえあればボタンでもよいが、変数の名前などの情報を入力する際には、いちいちテキストフィールドを開くよりも、そのまま入力できるコマンドラインインタフェースが適していると考えられる。また、システムとユーザのやり取りを会話ログのような形で一覧できるのもコマンドラインインタフェースの利点といえる。なお、将来的には、より自由度の高い自然言語に近い入力をサポートすることを目標としているが、現時点では決められた形式に従って入力するコマンドラインインタフェースとなっている。

具体的には、まず、`[property name] of [object name] is [label]` といった形式のコマンドで、オブジェクトのプロパティのラベルを設定する。例えば、バネのばね定数を指定する場合には、`stiffness of spring is k` と入力する。`[object name]` の部分は、`mass` や `spring` といったオブジェクトの種類でもよいが、複数存在する場合には、`stiffness of A is k` のように、タグを利用して指定する。`solve [label]` と

入力すると、その問題を解いた結果を返す。

図1は、簡単なバネのつり合いの問題を提案システムを利用して解いた様子を示している。バネが2つあるので、区別するために A および B のラベルがつけられている。またそれぞれのばね定数として、`k` および `K` が示されている。この図に対して、バネの伸びが両方とも `d` であることをコマンドラインインタフェースで設定し、`solve d` と指示することで、`d` の式が得られている。

### 4 実装

提案システムは、図形処理モジュールと、コマンドライン処理モジュール、および計算モジュールから構成されている。図形処理モジュールは、キャンバス内が編集されるたびにその内容を解析し、その結果を計算モジュールに送る。コマンドライン処理モジュールは、コマンドラインから指示がだされるたびにそれを解釈し、計算モジュールに送る。計算モジュールは、キャンバスに描かれたシーンの内部表現を保持し、コマンドラインから計算の指示が来た場合には、必要な計算を行って結果をコマンドラインに返す。計算モジュールでは、バックエンドとして数式処理システムである `maxima` [9] を利用している。

図形処理モジュールの動作の詳細を説明する。まず、テキストラベルや長さラベル、角度ラベルの内容を近くにあるオブジェクトのラベルとして関連付ける。長さや角度は、そのまま、長さや角度といったプロパティにマッピングされる。テキストラベルについては、オブジェクト自体のラベルである場合のほか、なんらかのプロパティのラベルである場合もあるので、まずはタグとして付与する。次にキャンバスに描かれているオブジェクトの接続関係を解析し、接続部を `コネクタ` として表現する。次に、各オブジェクトと各コネクタについて、つり合い関係を示す方程式を生成して計算モジュールへ渡す。たとえば、バネであれば、`spring.force = spring.stiffness * spring.growth` といった式を立てる。コネクタについては、`x` 方向と `y` 方向に分けてつり合いの式を立てる。もし、垂直方向にバネと質点がつながっている場合には、`y` 方向のみのつり合いの式として `-spring.force + g * mass.weight = 0` といった式を立てる。最後に、オブジェクトのプロパティを未知数と定数に分けて、計算モジュールに渡す。

計算モジュールでは、図形処理モジュールから受け取った方程式のリストおよび未知数のリストに対して、コマンドラインモジュールから受け取ったプロパティのラベルに関する情報を反映させる。その結果を `maxima` に送ることで解が得られる。適切な

解が得られた場合にはその旨を表示し、制約が多すぎた場合や少なすぎた場合にもその旨を提示する。

## 5 曖昧性解消

本研究において重要な点は、ユーザの入力に含まれる曖昧性を解消する手法を提案することである。本実装においては、大きく3点における曖昧性に対処しており、これらについて一つずつ説明する。

### 5.1 図とテキストの対応付け

GUIを用いた典型的なシステムでは、図形オブジェクトのラベルやプロパティなどはダイアログなどを通じて曖昧性なく入力される。しかし、いちいち対象とするオブジェクトをクリックしてダイアログを開いて、目的とする欄を見つけて入力するという手間がかかる。

本システムでは、このような手間をはぶき、ユーザがより簡便に入力できるような工夫をしている。まず、オブジェクトのラベルについては、明示的に対象オブジェクトを指定せず、単にテキストオブジェクトを近くに配置するだけでよいようにしている。例えば、A というテキストオブジェクトがバネの絵の近くに配置されていると、A というラベルがそのバネオブジェクトに関連づけられる。テキストオブジェクトの近くにあるオブジェクトが複数ある場合には、どのオブジェクトに関連づけるべきかという曖昧性があるが、現状の実装では単に一番近いものとしている。将来的には、このような曖昧性を許容し、後段で曖昧性を解消するような機構が必要と考えられる。

コマンドラインからの入力については、オブジェクトラベルを利用した指定方法(`growth of A is d`)と、オブジェクトタイプを利用した指定方法(`growth of spring is d`)の両方をサポートしている。前者については、対応するラベル(A)に該当するタグが付与されている図形オブジェクトに対して、当該プロパティを設定する。後者については、該当するオブジェクトタイプをもつオブジェクトに対して当該プロパティを設定する。現在の実装では、複数のオブジェクトが該当する場合にはすべてのオブジェクトを対象としているが、将来的にはユーザーに問い合わせるなどの曖昧性解消機構が必要と考えられる。

### 5.2 定数と未知数の区別

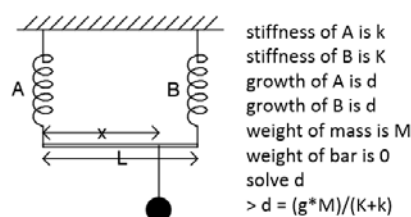
本システムでは、数式処理システムを利用して連立方程式を解いている。その際、数式処理システムへの入力として、方程式の中に含まれる変数のうち、どれが定数で、どれが未知数なのかを明示する必要がある。しかし、ユーザが単に図を描いて `solve` を要求しただけでは、どれが未知数なのか自明ではな

い。本実装では以下のような方法で判断を行っている。

まず、通常の物理現象において、初期条件として与えられるものを定数、重力がかかってバランスが取れた結果として得られるものを未知数とする。具体的には、おもりの質量やばね定数、棒の長さなどを定数として、糸の張力やバネの伸びなどを未知数とする。また、`solve x` のように、明示的に解を求められている変数についても未知数としている。

通常はこれで問題ないが、問題の状況によっては、定数となっているものを未知数として考えないといけない場合もある。例えば、図の問題において、ばね定数が異なる場合に、伸びについて特に制約がなければ、糸の固定位置は定数として与えることで全体の状態が確定する。しかし、伸びが同じであるという制約が加わると、糸の固定位置について変数にしないと過制約状態となって解が得られない。このような問題に対処するため、提案システムでは、入力を直接数式処理システムに渡すのではなく、未知数と定数の割り当てをいろいろと試しながら数式処理システムに渡して結果を観察し、もっとも妥当な結果を選ぶ、という高階モジュールを導入している。具体的には、まずデフォルトの状態で解いてみて、過制約となった場合には、決められた順に定数を未知数に変えて試している。決められた順として、現在は、まず `x` のような未知数に使われやすいラベルで表現されるプロパティを探し、それが無ければ、棒における接続位置や重さといった変動するものとして扱われやすいものプロパティを探している。

現在は、このようなルールを手作業で記述しているが、将来的には、数多くの実例から自動的にルールを見つけてくることなどが必要と考えられる。



(2009年度センター試験本試験第4問 問1より)

図 2. 定数か未知数か曖昧な例。A と B の伸びが異なれば `x` は定数として扱うべき。同じなら未知数。

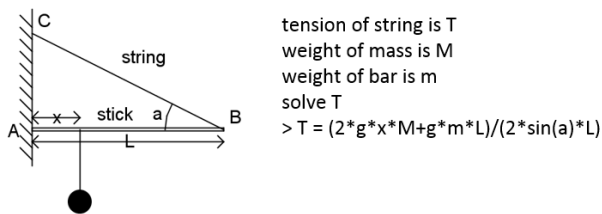
### 5.3 用語の曖昧性の解消

自然言語処理の主要な曖昧性の一つに表記の揺れがある。例えば、力学の問題に表れる「string」について、「wire」「cable」「thread」といった表記ゆれが考えられる。現在の実装では、それぞれの用語に

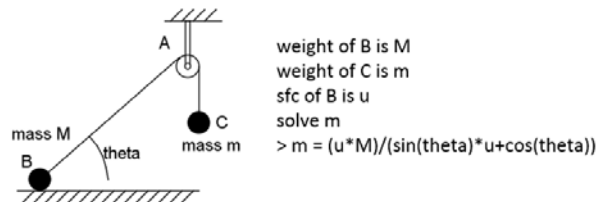
ついて表記ゆれの可能性のあるものをあらかじめ列挙しておく、それを使って曖昧性の解消を行っている。現在は、手作業で列挙しているが、将来的には、数多くの実例から自動的に表記のバリエーションを学習することなどが必要と考えられる。

## 6 結果

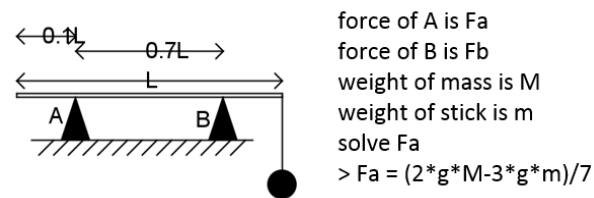
本システムの利用例として、センター試験の過去問から、力学のつり合いの問題を取り出して解いてみた。理想的には、実装に際して参考として用いた問題とは全く別の問題を取り出して解けるかどうか調べるべきであるが、センター試験の過去問のうちで現在の実装で対象としている範囲の収まるもの(力学における静的つり合いの問題で、かつバネ・棒・糸・重り・滑車などのみからなるもの)の数が少なく、とにかく解けるものを解いてみた、という例になっている。より汎用的なシステムを実装して、能力を客観的に評価することは今後の課題である。



(2005 年度センター試験追試験第 2 問 問 5 より)



(2005 年度センター試験本試験第 1 問 問 2 より)



(2005 年度センター試験本試験第 2 問 問 1 より)

図 3. 提案システムで問題を解いた例。

## 7 今後の課題

本研究の長期的な目標は、図とテキストを組み合わせることで、図形的要素を持つような問題解決を行うアプリケーションをより使いやすくしていくことである。特に、曖昧性のある入力を適切に解釈し

て動作するような、知的なユーザインタフェースを実現することを目指している。本稿で紹介したシステムはその第一歩であり、これをもとにさまざまな研究を行っていく予定である。

### - 手書きスケッチインタフェースとの融合

現状の実装では、図の描画はマウスでオブジェクトタイプを指定して部品を置いていくことで実現している。またテキストの入力は、キーボードで入力している。これらを手書きスケッチや手書き文字で置き換えることを考えている。関連研究で挙げたような手書きスケッチを利用したシステム[5][7]と同じようなユーザインタフェースを適用することで、電子黒板での利用など、人間と人間のコミュニケーションの場で利用することが可能になると考えている。さらに、紙に描かれた図や文字をスキャンして認識する機能なども実装していきたい。

### - 音声認識との融合

テキスト入力部分を音声認識によって入力することも検討している。音声入力を用いることで、より効率よく入力することが可能になると期待できる。しかし、音声認識は 100%確実に入力することが困難であるので、前述したような曖昧性解消技術をうまく利用していくことが必要である。

### - 自然言語インタフェースへの発展

現在利用しているテキストでの入力は、自然言語を模したものであるが、あくまでも決まった文法に従って記述されたコマンドである。これを自由な自然言語で入力できるようにしていくことも長期的な課題である。あらゆる自然言語表現を適切に扱うことは困難であると考えられるが、力学の問題のように問題分野を限定し、かつ図形入力という別の入力チャンネルの情報を利用することで、一般的な自然言語表現には対応できるようになると考えている。具体的にどのような自然言語表現がありうるかについては、ユーザテストやクラウドソーシングなどを用いて収集していく必要があると考えている。また、コマンドラインインタフェースの制約を減らして自然言語に近づける試みとしてキーワードコマンド[8][11]といったものも提案されており、そのようなアプローチも取り入れていきたい。

### - 他のカテゴリの問題への展開

上記にあげたような、システムとしての性能を向上していくものと並行して、力学以外の問題に対応できるようなシステムも開発していきたいと考えている。まずは、物理の他の問題、例えば電子回路の問題や、3 次元的な電磁気の問題などに自然に応用

できると考えている。

#### - 教育用アプリケーションとしての実装

現在の実装は単純に条件を与えると解くだけであるが、このような機能をバックエンドとして、教育用のソフトウェアとして実装することが考えられる。まずは、最終的な解だけでなく、その仮定を図示することで、理解を助けることができる。さらに、生徒に答えさせたあとで、その答えと正解とを比較することで、どこで間違えたのかを指摘することなども可能であると考えられる。

## 8 むすび

数式処理システムをバックエンドとして力学の問題を解くアプリケーションを紹介した。具体的には、図形的な情報を GUI で指定して、記号的な情報をコマンドラインで入力するようなユーザインタフェースを実装した。特に、入力に曖昧性がある場合に、それらの曖昧性を解消する手法として以下の3つの工夫を行った。1) 図に書かれている情報とテキストで入力された情報の統合、2) さまざまな入力を数式処理システムに与えて出力をチェックする、というような動作をする高階処理モジュールの導入、3) 入試問題において前提とされることの多い規則のヒューリスティクスとしての設定。今後は、これらの成果を発展させて、「曖昧性を許容する自然なインタラクション」の実現を目指していきたいと考えている。

## 謝辞

本研究は、国立情報学研究所で推進されている「ロボットは東大に入れるか」プロジェクトとの協力関係のもと行ったものである。本研究は JSPS 科研費 JP26240027 の助成を受けている。

## 参考文献

- [1] Bolt, R. A. Put-that-there: Voice and gesture at the graphics interface, ACM SIGGRAPH, 1980. 262-270.
- [2] Buxton, B. *Sketching user experiences: getting the design right and the right design*. Morgan Kaufmann, 2010.
- [3] Cindella. [www.cinderella.de](http://www.cinderella.de)
- [4] Coyne, B., and Sproat, R. WordsEye: an automatic text-to-scene conversion system. Proceedings of the 28th annual conference on Computer graphics and interactive techniques. ACM, 2001.

- [5] Davis, R. Magic paper: sketch-understanding research. IEEE Computer 40.9 (2007): 34-41.
- [6] Ernerfeldt, E. Phun-2d physics sandbox. Diss. Master's thesis, Umeå University, 2009.
- [7] LaViola Jr., J. and Zeleznik, R. C. 2004. MathPad2: a system for the creation and exploration of mathematical sketches. ACM Trans. Graph. 23, 3 (August 2004), 432-440.
- [8] Little, G. and Miller, R. C. 2006. Translating keyword commands into executable code. UIST '06, 135-144.
- [9] Maxima. [maxima.sourceforge.net](http://maxima.sourceforge.net)
- [10] Maple. [www.maplesoft.com](http://www.maplesoft.com)
- [11] Miller, R. C., Chou, V. H., Bernstein, M., Little, G., Van Kleek, M., & Karger, D. 2008. Inky: a sloppy command line for the web with rich visual feedback. UIST '08, 131-140.
- [12] Oviatt, S. Mutual disambiguation of recognition errors in a multimodel architecture. Proceedings of the SIGCHI conference on Human Factors in Computing Systems. ACM, 1999.
- [13] Winograd, T, and Flores, F. Understanding computers and cognition: A new foundation for design. Intellect Books, 1986.
- [14] Zitnick, C. L., Parikh, D., and Vanderwende, L. "Learning the visual interpretation of sentences." Proceedings of the IEEE International Conference on Computer Vision. 2013.
- [15] 横野光, 稲岳哲也. "論理演算と物理シミュレーションの結合による物理問題解答." 2014 年度人工知能学会全国大会 (2014).