

ドリトル言語の Kinect 対応と教育利用の可能性

兼宗 進^{1,a)} 中野 由章^{1,b)} 大西 修平^{1,c)} 野部 緑^{1,2,d)}

概要: 教育用プログラミング言語ドリトルの処理系を、マイクロソフト社の入力デバイスである Kinect の計測値を読み取れるように拡張している。現在はフリーソフトの KineX を介して身体の間節ごとの位置座標をネットワーク経由で取得しており、ドリトルから Kinect への直接的な接続についても対応を進めている。これらの実装について解説し、教育用言語からの利用可能性について検討する。

キーワード: ドリトル, プログラミング言語, Kinect, KineX

Using Kinect from Dolittle Program Environment

KANEMUNE SUSUMU^{1,a)} NAKANO YOSHIAKI^{1,b)} OONISHI SHUHEI^{1,c)} NOBE MIDORI^{1,2,d)}

Abstract: We have developed a new feature of Dolittle programming environment to communicate with the Microsoft Kinect device. Now we use the Kinect support software KineX and Microsoft Kinect SDK. KineX can send coordinate data to network periodically. We report the implementation and possibility of using physical input from student's programs.

Keywords: Dolittle, programming language, kinect

1. はじめに

学校教育でのプログラミングでは、入力装置として、当初から CUI 画面でのキーボード入力が、続いて GUI 画面におけるマウス入力が扱われてきた。一方、生徒たちが日常扱う入力装置としては、携帯電話におけるフリック入力、ゲーム機におけるコントローラー入力等のボタン操作に加え、最近では Wii における加速度センサ入力、XBox におけるモーションセンサー入力などが用いられるようになっていく。

我々は過去に、教育用言語である「ドリトル」[1] から汎

用入出力デバイスである Arduino を通して加速度センサによる入力を可能にした。その後は XBox 用のモーションセンサーである Kinect への対応を進めており、途中経過を発表した [2]。今回はその実装とともに、今後の開発の見通しを報告する。さらに、生徒が作成するプログラムからのモーションセンサーの教育利用の可能性について検討を行いたい。

2. Kinect とその支援技術

2.1 Kinect について

Kinect^{*1}は Microsoft 社のゲーム機である XBox 用に開発された入力デバイスであり、人体の動きを撮影により入力することができる。手に持ったりボタンを押したりといった形での、人体への接触が不要なことが大きな特徴である。国内では 2010 年 11 月に発売された。

主なセンサとしては、可視光センサー（ビデオカメラ）、赤外線による深度センサー、音声入力（マイクロホン）な

¹ 大阪電気通信大学
Osaka Electro-Communication University, Neyagawa, Osaka
572-8530, Japan

² 大阪府立寝屋川高校
Neyagawa Upper Secondary School, Neyagawa, Osaka 572-
0832, Japan

a) kanemune@acm.org

b) info@nakano.ac

c) ml12a004@oecu.jp

d) midori@mbox.kyoto-inet.or.jp

*1 <http://www.xbox.com/ja-JP/kinect>

どが搭載されている。可視光センサーを用いることで従来と同様のビデオ画像を用いることが可能だが、撮影された2次元平面の画像に加え、近赤外線を用いた深度センサーを用いることで、撮影された対象物の Kinect からの距離をプログラムから利用することができる。図1に、Kinectの外観と、Kinect から出力される赤外線の様子^{*2}を示す。撮影のフレームレートは、640x480の標準画像の場合、毎秒30枚(30fps)である。



図1 Kinectの外観と Kinect から出力される赤外線の様子

2.2 Kinect for Windows SDK

Microsoft 社は2012年2月に kinect for Windows SDK を公開した。このライブラリを利用することで、Kinect を利用したソフトウェアを開発することが可能になる。Kinect の発売元が公開している正式な開発環境であり、多くのソフトウェア開発で利用が進んでいる。

一方、ドリトルのようなフリーソフトウェアから利用することを考えると、「Windowsのみで動作し、MacやLinuxなどの他のOSに対応しない」「Windows7以降で動作し、XPなどの以前のバージョンには対応しない」「開発言語がMicrosoft社の提供するC++、C#、VisualBasicなどの開発環境に限定される」などの課題が存在する。

2.3 OpenNIとNITE

Kinect用のオープンソースのライブラリとして、OpenNIとNITEが知られている。OpenNIはKinectの深度センサーを開発したPrimeSense社が公開しているソフトウェアで、NITEは骨格情報を認識するソフトウェアである。

Microsoft社のKinect for Windows SDKと比較すると、国内での認知度や書籍等による情報は少ないが、ドリトルのようなフリーソフトウェアでの利用を考えると、「Windowsだけでなく、Mac、Linux等の複数のOSで動作できる」

^{*2} <http://www.engadget.com/2010/11/08/visualized-kinect-night-vision-lots-and-lots-and-lots-of-do/>

「C++、C#のほかに、C、Java等の言語でも開発を行える」ことは大きな利点である。

3. KineXを利用したKinect対応

3.1 KineX

KineXは工学ナビのサイトで公開されているソフトウェアである。KineXはKinect for Windows SDKを利用して人物を認識し、その関節位置の座標をネットワークに送出する。図2に、KineXのモデル図^{*3}と実行画面の例を示す。

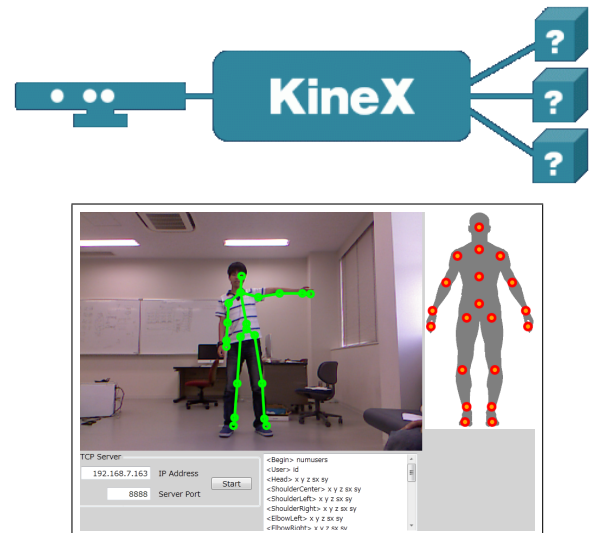


図2 KineXのモデル図と実行画面の例

通常、Kinectの入力を利用するためには、KinectのSDKを組み込むなど、ソフトウェアの改変が必要になる。しかし、KineXはKinectから入力した座標データをネットワークに送信してくれるため、KineXを用いることで、ネットワークのソケット通信が可能なソフトウェアであれば、特別な改変を必要とせずに、Kinectからの入力を利用することができるようになる。

KineXのデータはネットワークから利用可能なため、KineXを1台のマシンで動かしておくことで、同じマシンのソフトウェアだけでなく、他のマシンのソフトウェアからもKinectの入力を利用することができる。

標準のポートは8888である。図3に、KineXから送出されるデータの例を示す。「Head(頭)」などのタグは認識した身体の部位を表す。x, y, zは空間での位置情報(xは水平方向、yは垂直方向、zは奥行きを表す)、sx, syは表示される画像平面上の位置情報である。

3.2 ドリトルのKineX対応

ドリトル(V2.32)に、KineXから座標データを取得するKineXオブジェクトを実装した。図4にプログラム例を示す。

^{*3} <http://kougaku-navi.net/kinex/>

(タグ)	(x)	(y)	(z)	(sx)	(sy)
<Head>	-0.19	0.53	2.91	282	165
<ShoulderCenter>	-0.20	0.35	2.94	280	197
<ShoulderLeft>	-0.36	0.26	2.96	249	215
<ShoulderRight>	-0.04	0.24	2.92	307	217
<ElbowLeft>	-0.42	0.01	2.93	240	257
<ElbowRight>	0.01	-0.02	2.88	316	265
<WristLeft>	-0.44	-0.21	2.83	232	299
<WristRight>	0.06	-0.21	2.79	327	300
<HandLeft>	-0.45	-0.30	2.80	230	316
<HandRight>	0.08	-0.30	2.77	331	318
<Spine>	-0.20	0.01	2.90	279	259
<HipCenter>	-0.20	-0.05	2.84	278	268
<HipLeft>	-0.27	-0.11	2.83	264	281
<HipRight>	-0.13	-0.12	2.81	292	281
<KneeLeft>	-0.29	-0.55	2.79	260	362
<KneeRight>	-0.09	-0.58	2.80	297	367
<AnkleLeft>	-0.29	-0.88	2.78	258	427
<AnkleRight>	-0.09	-0.91	2.79	297	432
<FootLeft>	-0.29	-0.96	2.74	258	445
<FootRight>	-0.09	-0.98	2.76	297	447

図 3 KineX から送出される座標データの例

```

kinex!(システム:ipaddress) 接続。
表示欄=ラベル！作る。
「
    関節 = kinex !読む。
    x =関節：右手首x。
    表示欄！(x)書く。
」！100回 繰り返す。

```

図 4 KineX と通信するドリトルのプログラム例

KineX オブジェクトは「接続」メソッドにより起動し、パラメータで指定された IP アドレスの 8888 ポートの監視を開始する。

kinex!"192.126.1.24" 接続。

同じマシン内で KineX に接続する場合は、システムオブジェクトからローカルマシンの IP アドレスを参照することで、次のように書くことも可能である。

kinex! (システム:ipaddress) 接続。

KineX オブジェクトはネットワークを監視し、KineX からデータが到着するたびにそれらを受け取り、オブジェクト内に保持する。そして、ユーザーのプログラムから「読む」メソッドを送ることで最新のデータを取り出すことが可能である。

ret = kinex !読む。

このプログラムを実行すると、戻り値として、すべての関節の「x, y, z, sx, sy」がプロパティとして格納されたオブジェクトが返される。たとえば、頭部 (Head) の x 座標は、次のように取り出すことが可能である。

x = ret : 頭x。

4. デバイスと直接通信する Kinect 対応

現在は、ドリトルから直接 Kinect と通信する機能の開発を進めている。



図 5 KineX を利用したドリトルの Kinect 対応

図 5 は、前章で説明した KineX を用いた Kinect 対応の構成図である。ドリトルは KineX と通信し、Kinect で認識した身体の関節座標を受け取る。この方式で実現できる機能の特徴は次の通りである。

- 身体の関節座標を取得できる。
- 最大 2 人までの関節情報を認識できる。
- Windows7 以降の Windows のみで動作する。

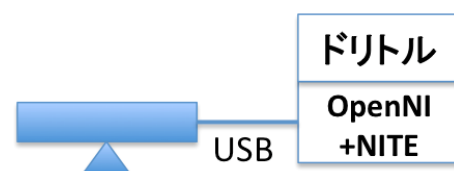


図 6 直接通信するドリトルの Kinect 対応

図 6 は、OpenNI と NITE を利用した Kinect 対応の構成図である。ドリトルは直接 Kinect と通信し、Kinect から各種の情報を受け取る。この方式で実現できる機能の特徴は次の通りである。

- 身体の関節座標を取得できる。
- 最大 8 人までの関節情報を認識できる。
- 可視光画像を取得できる。
- 深度画像を取得できる。
- 音声情報を取得できる。
- Windows、Mac、Linux など多くの環境で動作する。

可視光の画像は、ビデオカメラの画像のように、人の目で見えるカラー画像を取得する。深度画像は、ピクセルごとの、Kinect からの距離を表す情報を持つ画像を取得する。深度画像を利用することで、身体以外の物体の位置情報を認識することが可能になる。

5. 学習者に提供する機能の検討

5.1 教育用言語における提供方針

ドリトルは主に中学校から大学の教育現場で利用されている。プログラムを書く主体は生徒や学生という学習者である。教育用言語に機能を実装する場合には、利用される年齢や利用される題材を具体的に複数想定しながら、次の

ような視点で検討する必要がある。

- 扱いたいプログラムを学習者が実現できること。
- 難しすぎない難易度でプログラムを作成できること。
- 高機能になりすぎず工夫や思考の余地が残っていること。

5.2 身体のパーズ判定

身体関節座標を用いるサンプルプログラムは図4で示した。関節ごとの空間での3次元座標を学習者のプログラムの中で扱うことができる。ただし、関節ごとのx, y, zといった座標値を自在に使いこなすことは、学習者の年齢層などによっては難易度が高すぎる可能性がある。

そこで、右手を上げているかどうかといった、人間のポーズを判定するわかりやすい命令の提供も考えている。たとえば「右挙手?」という命令は、内部で「右手首が右ひじより上にある」という意味で、次のように定義される。

kinect: 右挙手? = 「右手首y > 右ひじy」。

5.3 身体ジェスチャー判定

身体が停止しているポーズだけでなく、動きによるジェスチャーを判定したい場面が考えられる。たとえば、「手を振り下ろしている」「手を降っている」といった、静止したポーズ命令では判定できない動作である。

ジェスチャーの動作は、空間での関節位置を時系列で分析することで判定できる。右手を振り下ろす動作は、たとえば次のように定義できる。

- 右手首が右肩より高い位置にある。(初期条件)
- 右手首が連続的に下がっていく。(連続条件)
- 右手首が右肩より低い位置にある。(終了条件)

また、右手を振る動作は、たとえば次のように定義できる。

- 右手首が連続的に右に移動する。
- 右手首が連続的に左に移動する。
- 上記の2つの動作を交互に繰り返している。

このような連続した動作の判定は、学習者が関節の座標値を用いて記述することは可能だが、難易度はかなり高くなる。そこで、よく使われそうなジェスチャーを想定し、あらかじめ判定する命令を用意しておくことは有用性が高いと考えられる。

5.4 深度画像の利用

通常の可視光による動画と違い、深度画像は色の情報はないものの、代りに対象物までの距離を扱える。たとえば、距離に応じて色を変化させると、前にいる人と後ろにいる人を色で区別して表示できる。物体や人物の判定に用いるなど、新しい用途が考えられる。図7に深度画像の例

を示す。^{*4}

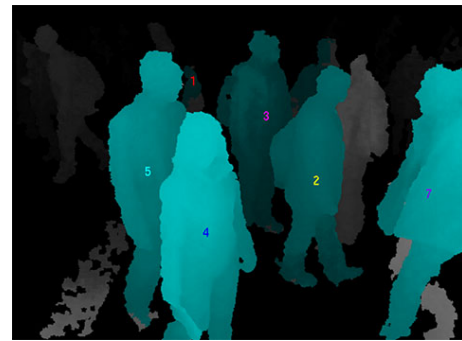


図7 深度画像の例

6. 教育利用の可能性

教育用言語での利用の可能性を検討する。

生徒のプログラムからの利用としては、Kinectの本来の用途であるゲームのコントローラーとしての利用が考えられる。筆者らは大学の子ども向け科学イベントでKinectを利用した自動車運転ゲームを出展し、多くの児童・生徒が利用できることを確認した。

Kinectはセンサデバイスであることから、計測・制御教材としての利用も期待される。身体や物体の位置を計測し、その結果を用いてロボットなどを制御する利用が考えられる。Kinect自体は1万円台と安価であるため^{*5}、学校またはクラスごとに用意することは十分可能である。

また、教師が実験やシミュレーション等の教材プログラムを開発することも可能である。ドリトルという教育用言語で記述できることでこのような利用の可能性が実現している。

その他、Kinect自体としては、今後は市販の教材アプリケーションの開発も期待される。たとえば、ジェスチャー機能を応用したモーション解析を実現することで、スポーツなど授業やクラブ活動でのフォームの確認への利用が考えられる。

その他、身体障害者や幼児を対象とした教育アプリケーションの入力デバイスとしての利用が期待される。機能の開発を進めつつ、利用の可能性についても検討を続けたい。

本研究は、科学研究費補助金(基盤研究(C))22500828の補助を受けています。

参考文献

- [1] 教育用プログラミング言語「ドリトル」
<http://dolittle.eplang.jp>
- [2] 大西修平, 野部緑, 中野由章, 兼宗進. Kinectを利用したジェスチャーによるプログラム入力の可能性. 情報処理学会, 第116回コンピュータと教育研究会, 2012.

^{*4} <http://www.pervasive.jku.at/Teaching/lvaInfo.php?key=349>

^{*5} 非商用利用の場合。