M-007

没入型ソフトウェア可視化システムにおける 複雑構造理解に適したインタラクション手法

A comparative study on hand occlusion management methods for tracing work support system using a projector

青木 達志 † 辻 愛里 ‡ 藤波 香織 §
Tatsushi Aoki Airi Tsuji Kaori Fujinami

1. 背景

近年，社会全体でデジタル化が進んでおり，ソフトウェアの需要が増加している．しかし，開発者にとって大量のコードにおける複雑な構造及び依存関係を理解することは非常に困難である．この問題を解決する方法としてソフトウェアの構造を視覚的に表現するという方法がある．例として，統一モデリング言語（UML）ではグラフィカル表記を使用してソフトウェアを視覚的に記述することができる．しかし，この表記では 2D で構造が表されているため，表現できる情報が制限される．3D でソフトウェアの構造を可視化するという方法では 2D と比較してより多くの情報を提供することが可能となる．そのためソフトウェア構造を 3D の都市型メタファを用いて表す研究がある [1]．さらに人間の自然な動きを利用した操作を可能にするため，仮想現実（Virtual Reality : VR）や複合現実（Mixed Reality : MR）を利用して 3D ソフトウェア構造を可視化し，操作をする研究が行われている [2] [3]．しかし，構造の探索効率において有効なインタラクション手法は明らかになっていない．

これを踏まえて本研究では，VR を用いたソフトウェア可視化システムにおけるインタラクション (操作，フィードバック) の提案を行い，構造探索効率に与える影響について調査してきた．その結果，ジェスチャーの低い認識精度が原因となり，操作の難易度，直感性，そして所要時間に悪影響があることが判明した．また，システムで使用した構造は手動で作成したために単純なものになっており，開発者が理解の難しい複雑な構造には対応していないという課題も明らかになった．したがって本稿ではジェスチャーの認識精度の改善や構造作成の自動化を実現し，複雑構造への対応を行い，複雑構造における新たなインタラクションの有効性について調査する．

2. 関連研究

ソフトウェア構造を 3D で表示する研究は数多く存在し，2D 画面上に表示する研究と VR 空間上で表示する研究の 2 種類に分類することができる．

2D 画面上でソフトウェア構造を表示させる研究として Werttel らの開発した CodeCity [1] があり，都市の区画がソフトウェアのパッケージ，建物がクラスを表している．またクラス内のメソッド数，属性数はそれぞれ建物の高さ，サイズを表している．これらの構造はマウスを用いて探索可能である．さらに Balzer ら [4] の研究では構造をバブル型で表現している．パッケージは半球によって表され，サイズはパッケージやサブパッケージに含まれるメソッドや属性の数に合わせて調整される．またクラスは円で表現され，含まれるメソッドと属性の数によって表面積が変化する．このクラス内ではメソッドと属性はボックスオブジェクトとして配置される．また依存関係は青いパイプ状のオブジェクトで表現されている．数種類の関係は色で分類され，同じタイプで同一の開始点と終了点を持つものは一つの関係に結合される．このシステムでは，メソッドのオブジェクトの高さがコード行数に比例するように表現することで可視化における情報量を増加させることが課題として挙げられた．

AR，VR を用いて 3D 構造を表示する研究では，Fittkau らの ExplorViz [2] がある．このシステムでは，ソフトウェア構造を都市型で表しており，パッケージ，クラス，依存関係を表現している．ExplorViz [2] は Microsoft Kinect v2 と Oculus Rift DK1 を用いた web ベースのシステムである．フリーハンドジェスチャー操作によって構造を探索することができ，マウスなどと比較してより自然なインタラクションが可能である．しかし，被験者に対して行ったインタビューから，提案したズーミングジェスチャーのみ，好まれていなかったことが示された．また Merino らの開発した CityAR [5] や PerfVis [6] では，AR 上で Codecity [1] を利用して構造を表示させている．これらのシステムは表示するコード情報などは CodeCity [1] と同様であるが，操作方法が異なり，構造の拡大，縮小，回転などの操作は，ユーザ自身が歩き回って構造を見る角度，距離を変えることで実現している．オブジェクトの選択は，視野の中心にあるポインタとタップ操作によって行うことができ，選択したクラスの含んでいるメソッド名や属性名など詳細情報を見ることが可能である．しかし，頭の動きを安定させることが難しいため，正確なポインティングが困難という問題点がある．また，頭を動かすことで，表示テキストが見づらくなることも問題である．さらに Oberhauser らは，VR を使用して構造の探索が可能なだけでなく，VR 上でキーボードとマウスを表示することでコーディングが可能な MR-FTC [7] が開発されている．システムを使用した 3 つの比較実験として，1 つ目の実験では VR と非 VR の環境下でシステムを使用して構造探索に適しているかが調査され

†東京農工大学 大学院 生物システム応用科学府 生物機能システム科学専攻,
Department of Bio-Functions and Systems Science
Tokyo University of Agriculture and Technolog
‡東京農工大学 大学院 工学研究院 先端情報科学部門,
Department of Computer and Information Sciences
Tokyo University of Agriculture and Technolog

た．結果として VR を使用した場合，非 VR と比較して作業時間が遅くなったが，有意差は認められなかった．2 つ目の実験では，入力環境の差異による影響が調査された．仮想キーボードを VR に表示してコントローラでの文字入力，手元のキーボードとマウスを VR 上に映し出した MR 環境下でのタイピング，非 VR 環境でのキーボード入力の 3 つの条件でコーディングの適合性について調査した結果，非 VR 環境下でのキーボード入力が最も作業時間が短いことが判明した．また，コントローラと比較して，MR 環境下でのタイピングの場合作業時間が短いという結果を得られた．そして，3 つ目の実験では，音声入力，タブレット，VR コントローラを使用した際のメニュー画面操作における適合性について調査し，その結果，VR コントローラが最も作業時間が短く，ユーザに好まれており，音声とタブレットでは同程度に作業時間が遅いということが判明した．

これらの研究では，ソフトウェア構造ではコード情報としてクラス内のメソッド，属性の名前，総数やクラス間の依存関係などが表示可能であるが，その他のフィードバックを実装し，フィードバックによる構造探索に与える影響について調査した研究は存在しない．したがって，本研究では構造探索に適したインタラクション手法を提案し，その有効性について調査する．

3. 実験システム

3.1. システム概要

本研究ではソフトウェア構造を VR 空間上で表示可能なシステムを開発する．開発システム上でソフトウェア構造を探索する場合，ソースコードの依存関係を 3D で表示し，ジェスチャーによって構造を探索する．さらに特定の操作を行うことによりクラスの関係，特定のメソッドを含むクラスを強調表示するなどの，ソフトウェアの詳細情報を提示することが可能である．ユーザはソースコードの修正中，コードに含まれる属性や参照関係などを知りたい場合に，HMD を装着してシステムを使用し，ジェスチャーによる操作で構造探索を行い，クラスの構成要素や参照関係について調査する．調査完了後，ユーザは HMD を外してコードの修正を再開する．VR 上で構造を表示させることにより，構造以外の情報を遮断し集中することができるためユーザが構造探索に没入することが可能となる．

本システムは図 1 に示すように，構造分析部，3D 構造作成部，操作認識部といった主要機能で構成される．

3.2. 構造分析部

構造分析部では，ソースコード構造の分析を行う．構造の分析はソースコードを分析し，分析した情報を抽出することで可能になる．本研究では，コード分析を行う方法として J クラスレポート [8] のようなソースコード解析ツールを利用する．このツールは入力としてソースコードファイル（Java のみ）を渡すことで，コードを分析し，2 種類の Microsoft Excel ファイル（*.xlsx ファイル）作成する．一方はソフトウェア内に存在するメソッド，属性の総数や名称の一覧などの情報を含んでおり，もう一方は各クラスに含まれている属性やメソッドの名称や参照関係の情報で構成されている．出力ファイルにはソースコード内での宣言や定義されたメソッドの名称や型を含む．出力ファイルを 3D 構造作成部で分析しやすくするために，編集し，利用する．

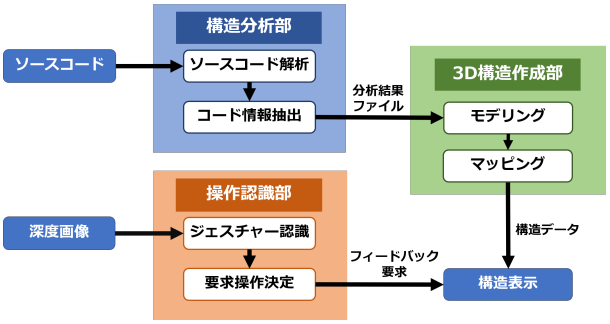

図 1: システム概要図

3.3.3D 構造作成部

3D 構造作成部ではソースコードの構造を表現した 3D 構造を作成し，VR で表示させるまでの処理を行う．これらの操作を行うために本研究では Unity3D [9] を利用する．Unity3D とは，複数のプラットフォームに対応するゲームエンジンである．このゲームエンジンでは様々な 3D オブジェクトが利用でき，オブジェクトの形やサイズなどを編集可能である．また Unity3D ではプロジェクトに VR デバイスを追加することで 3D シーンを VR 空間で閲覧することが可能であり，VR 上でソースコード構造を表示し，探索することが可能となる．これまでの構造作成部では，構造分析部で生成されたファイルを基に手動で構造を作成していた．本稿では Unity 側でファイルを読み込み，3D 構造を作成することで VR 上で構造を表示する．これにより，複雑な構造も自動的に作成することが出来る．

ソフトウェア構造は Balzer ら [4] らの研究で用いられた表示方法を参考に作成した．図 2:a で確認できる半透明の半球は，ソフトウェアのパッケージを表している．パッケージの内部には複数の青色の円が配置されている．図 2:b にパッケージ内部を拡大した画像を示す．青い円は，パッケージ内に存在するクラスを表現している．円上に配置されている緑色のオブジェクトは属性，赤色のオブジェクトはメソッド，水色のオブジェクトはコンストラクタを表している．

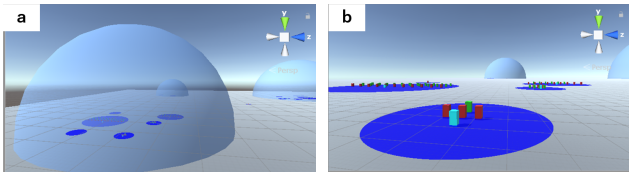

図 2: プログラム構造 (a:パッケージを表現した半球，b:クラスと構成要素を表すオブジェクト)

作成した 3D 構造では，ユーザの理解を支援するために構造表示を変化させる機能が存在する．本研究で提案する機能は 1) 参照元の強調表示，2) 参照元の抽

出，そして 3) 参照関係の切り替えの 3 種類である．

参照元の強調表示は，構造探索時にメソッドや属性などのオブジェクトを選択した場合，そのオブジェクトを参照しているクラスを強調表示する機能である．強調表示の様子を図 3:a に示す．図 3:a の中央にある黄色いボックスオブジェクトは，ユーザによって選択されたオブジェクトである．その奥にある黄色い 2 つの円オブジェクトは選択対象と参照関係にあるクラスを表している．

参照元の抽出とは，参照元を強調表示した後に参照元以外のオブジェクトを取り除く機能である．複雑な構造で強調表示する場合，参照元が遠方に配置されていると参照元の発見が困難になると考えられる．そのため，抽出機能を実装することで参照元の発見が容易になると考えられる．図 3:a の状態から実際に抽出操作を行うと黄色く表示されたオブジェクト以外は全て透明化される．

参照関係の切り替えは，参照関係を表した図 3:b のような棒状のオブジェクトが表す参照の種類を変更する機能である．本システムにおけるクラス間の参照関係は，参照の適用箇所によりインスタンス生成，メソッドの引数，メソッドからの戻り値，クラスの属性，スーパークラスがあり，5 色で区別される．しかし，クラス間で複数の関係を有するような場合には，関係を表すオブジェクトが重なることで関係性の把握が困難になると考えられる．このため，切り替え機能により把握したい種類の関係のみを把握可能となるようにする．

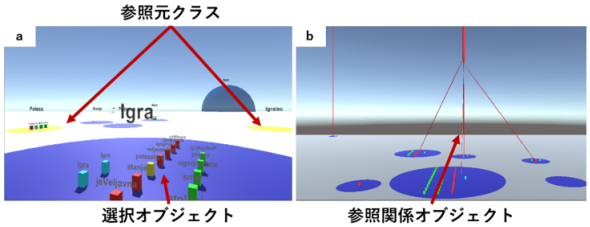

図 3: 提案機能（a:参照元の強調表示，b:参照関係の表示）

3.4. ジェスチャー認識部

ユーザのジェスチャーを認識するために本研究では Leap Motion [10] を使用する．Leap Motion は小型の USB 周辺機器で，2 基の赤外線カメラと赤外線照射 LED で構成される．赤外線 LED で手と指を照らし，赤外線カメラで撮影した画像を解析することによって 3D 空間での手と指の位置を割り出す．

Leap Motion で認識するジェスチャーは，基本的な移動操作と提案機能を実行するための操作となる．基本的な移動操作は，ユーザの前後移動を行うピンチアウト操作，上下左右の移動を行うスワイプ操作，視点の回転を行う手首の回転操作である．これらの操作方法はスマートフォンや PC のタッチパッドなどでよく用いられる操作で直感的であると考えられる．

提案機能を実行するための操作は 3.3 章で述べた参照元の強調表示，参照元の抽出，そして表示する参照関係の切り替えの 3 種類の機能を使用するための操作である．強調表示操作はオブジェクトを選択することで実行する．選択操作はタップジェスチャーによって行う．タップジェスチャーは，スマートフォンや ATM など，日常的に使用する操作であるため直感的と考えられる．抽出操作と参照関係の切り替え操作についても探索に適したジェスチャーを採用するため，20 代の 12 名を対象に機能を実際に見て実演してもらうという調査を行った．その結果，抽出操作は手で払うジェスチャー，参照関係の切り替え操作は拳を握った状態で腕を回すジェスチャーに決定した．

4. ジェスチャー精度の改善

ジェスチャーによるスワイプ操作やピンチ操作 (前後移動)，回転操作は，操作を実施した後に手を元の位置に戻すと逆方向に動作してしまう誤認識が多く見られた．この改善案として，腕を一定距離伸ばしている状態の場合のみジェスチャーを認識する方法が考えられる．多くの使用者においてジェスチャー操作をする際に，腕を伸ばすような行動がよく見られた．使用者はスマートフォン操作で，手を伸ばして画面に触れながら操作を行い，終了後に手を画面から離すといった行為に慣れており，ジェスチャー操作時にもその傾向があると考えられる．したがって，使用者から一定距離に領域を設定し，その領域内で行われた動きのみをジェスチャーとして認識する方法が考えられる．人間の腕の可動域は平面形状ではなく，扇形になっているため大きく手を動かすような操作が必要な場合には認識範囲を曲面状にする必要がある．本システムでは，複雑な構造を使用しているため，遠距離移動でジェスチャーの動きが大きくなり，弧を描くように手を動かす可能性がある．そこで操作範囲の形状が図 4:a のような平面形状と図 4:b のような曲面形状のどちらが適しているのか決定するための調査を行った．以下の節で調査内容と結果を記述する．

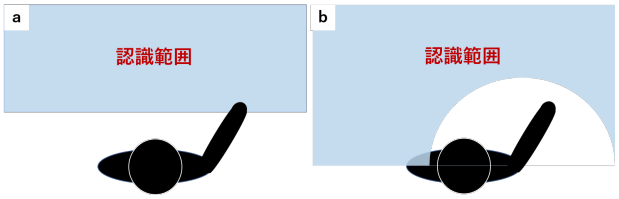

図 4: ジェスチャー認識範囲形状（a:平面形状，b:曲面形状）

4.1. 調査内容

10 種類のジェスチャー（四方向スワイプ，ピンチアウト，回転，選択，抽出，関係表示）について 2 種類の認識範囲を用いて精度計測を行った．その後，サンプルとして用意したソフトウェアの構造を平面状，または曲面状の認識範囲でジェスチャーを用いて操作するタスクを行ってもらい，その後アンケートにより各操作範囲で操作する難易度や認識されやすさ，最も好ましい操作範囲，評価した理由について調査する．被験者は 20 代の成人 6 人で，全員右利きである．

4.2. 調査結果

図 5 は 2 種類の認識範囲の形状で操作したジェスチャーの精度の平均である．精度計測の結果，左スワイプ

操作において曲面での認識範囲を用いた際の精度が高く，拡大，縮小操作では平面での精度が高くなっていた．また，得られた結果について有意確率 5%で対応のある t 検定を行ったところ，先ほどの 3 つの操作全てに有意差があることが分かった．以上のことから，左スワイプでは曲面で精度が高くなり，拡大縮小操作は平面で精度が高くなることが判明した．左スワイプだけが曲面形状で精度が高いため，精度判定の結果出力がされているファイルを確認したところ，左スワイプ操作の後半で認識範囲外に出た判定をされている場合が複数見られた．この調査では被験者が全員右利きであり，ジェスチャー操作を右腕で行っている．右腕の場合，右方向への可動域が広く右方向のスワイプ操作で大きく動かした場合でも平行移動を容易に行うことが可能であるが，左方向の場合には可動域が右方向と比較して狭く，操作の後半で平面に沿った腕の移動が困難になり，弧を描くように平行移動を行っているため平面形状の際に精度が低くなったと考えられる．

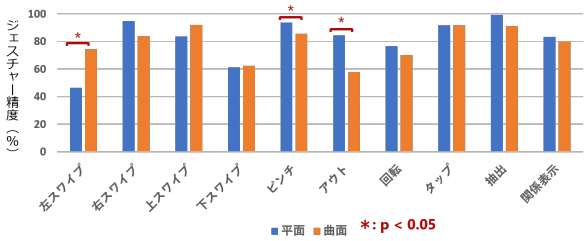

図 5: 認識形状によるジェスチャー精度

図 6 は 2 種類の認識範囲の形状で操作したジェスチャーについてユーザが 5 段階で評価した難易度の平均である．得られた垂直スワイプ（上下）の精度について有意確率 5%で対応のある t 検定を行ったところ有意差があったため，ユーザは垂直スワイプを行った際，平面形状で操作すると難易度が低く感じることが分かった．アンケートの回答では「平面の場合，手を伸ばさなくてよかった」という意見が多く見られた．曲面範囲ではユーザの肩よりも上もしくは下の位置ではユーザの正面方向での距離は異なり，肩の高さでは遠く，それよりも上もしくは下の位置では近くなる．そのため操作途中で肩と同じ高さに腕が来た時に範囲外に出ないように注意して手を伸ばす必要があるため，このような結果になったと考えられる．

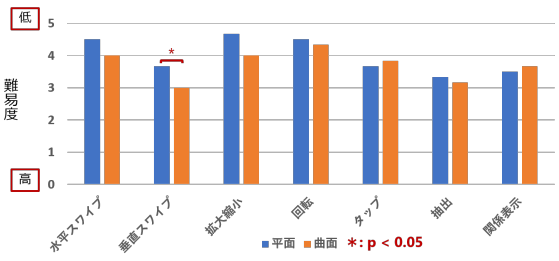

図 6: 認識形状による難易度の比較

図 6 は 2 種類の認識範囲の形状で操作したジェスチャーについてユーザが 5 段階で評価した操作の反応しやすさの平均である．この得られた結果に t 検定を行ったところ全ての操作において有意差がなかった．そのため，被験者の感じた反応のしやすさは 2 種類の操作範囲で大きく変化しないと判明した．

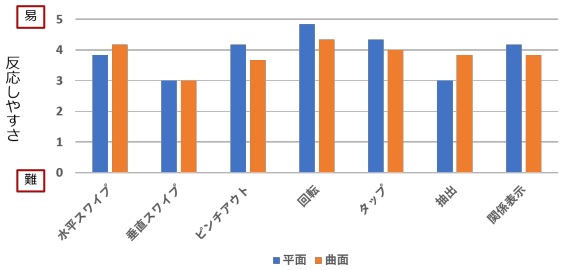

図 7: 構造探索時の所要時間

また最後に最も探索しやすかった認識形状について被験者に回答してもらったところ 6 人中 3 人が平面形状の方が「操作がやりやすい」と回答した．

以上の結果から本システムで採用する認識範囲形状は平面形状にすることに決定した．

5. システム操作に対する評価実験

複雑構造における提案インタラクションの有効性の調査を目的として，オブジェクト指向プログラミングの学習経験がある，右利きの 20 代 8 名 (男性 5 人，女性 3 人) を対象として，評価実験を行った．実験では，10 種類のジェスチャー精度を計測し，被験者はコントローラ，ジェスチャーを使用して構造を探索しつつ，構造に関する問題に回答した．実験終了後にアンケート調査を実施した．構造に関する問題は 3 種類の問題で構成されており，クラスの構成要素（メソッド，属性，コンストラクト）について回答する問題，参照元のクラスについて回答する問題，参照関係について回答する問題がある．

評価項目はジェスチャー精度，探索の所要時間，被験者による 5 段階の操作の難易度（5：操作しやすい～1：操作しにくい），操作の直感性（5：無意識的～1：意識的），機能の必要性（5：必要～1：不必要）とした．探索の所要時間は探索の開始から回答終了するまでの時間とする．コントローラを使用した場合とジェスチャーを使用した場合で有意水準 5%で行い，有意差が見られたものに*を付けて表記する．

以下の節では実験概要と結果を記述する．

6. 実験結果

計測したジェスチャーは基本的な移動操作であるピンチアウト操作（前移動・後移動），スワイプ操作（上下左右の移動），手首の回転操作（視点の回転）と提案した機能に使用する操作である強調操作，抽出操作，参照関係の切り替え操作である．

表 1 は基本的な移動操作の精度の平均であり，表 2 は提案機能に使用する操作の精度の平均である．計測した操作の中で最も精度が高いのは回転操作の 93.1%で，最も低いのは下方向スワイプ操作の 52.5%であった．スワイプ操作において，右方向のスワイプは 90.6%

という高い精度であるが，上下左スワイプ操作は他の操作と比較しても低い精度であることが分かる．

表 1: 基本的な移動操作の精度

	スワイプ				ピンチアウト		
	上	下	右	左	前	後	回転
精度 (%)	63.1	52.5	90.6	55.9	92.1	86.3	93.8

表 2: 提案機能に使用する操作の精度

	選択	抽出	切り替え
精度 (%)	70.0	82.5	71.3

図 8 は構造に関する問題をコントローラ，ジェスチャーを用いて回答したときの所要時間の平均値である．問 1(1) および 1(2) はクラスの構成要素について回答する問題で，問 2(1) および 2(2) は参照元のクラスについて回答する問題，問 3(1) および 3(2) は参照関係について回答する問題である．各小問 (1)，(2) には，難易度に差はない．図 8 を見ると全問題において，ジェスチャーよりもコントローラを用いた際に所要時間が短いことが分かる．また，全ての問題において有意水準 5%で検定を行ったところ，全ての問題に有意差があることが分かった．このことから複雑構造になったことにより目標物が遠距離となった場合にジェスチャー操作での探索時間が長くなると考えられる．

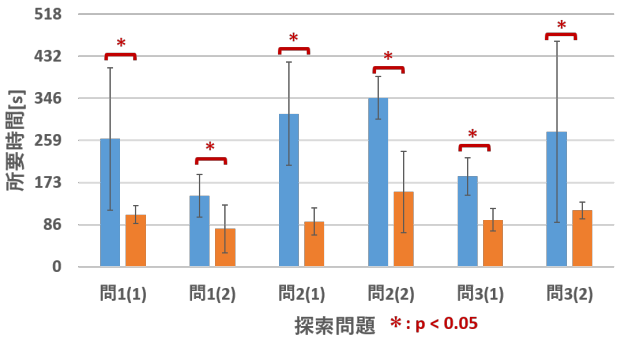

図 8: 構造探索時の所要時間

図 9 は各操作の難易度を 5 段階評価した平均値である．図 9 を見ると全操作においてジェスチャーよりもコントローラを用いた場合，難易度が低いことが分かる．また有意水準 5%で検定を行ったところ全ての操作において有意差が確認された．回答理由として，アンケートに「遠距離移動の際に複数回操作させる必要がある」,「1 回の操作が大きい」という意見が複数見られた．

図 10 は各操作の直感性を 5 段階評価した平均値である．図 10 を見ると移動操作や選択操作ではジェスチャーよりもコントローラを用いた場合に直感的となり，抽出操作と切り替え操作ではコントローラよりもジェスチャーを用いた場合に直感的となっていた．また有意水準 5%で検定を行ったところ抽出操作のみ有意差が確認された．回答理由としてアンケートでは，ジェスチャーを用いた場合の抽出操作は「払いのける動きで連想しやすい」という意見があり，コントローラを用いた抽出操作は「使用するボタンが把握できない」という回答が複数見られた．

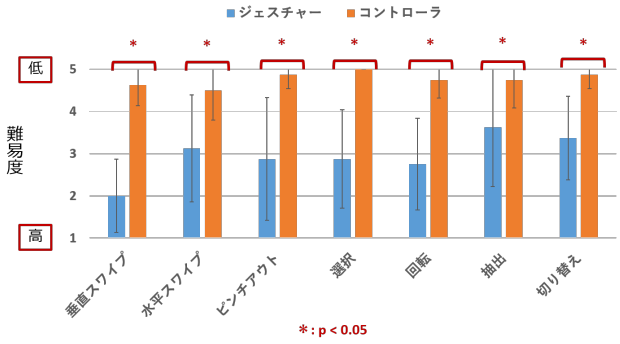

図 9: 操作の難易度

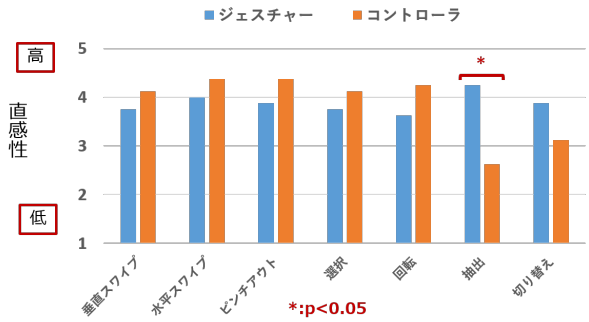

図 10: 操作の直感性

図 11 は 3 種類の提案操作の必要性を 5 段階評価した平均値である．図 11 を見ると 3 種類の全機能において，4.75 以上の高い値を記録していた．以上のことから複雑構造において提案した機能は探索に有効であると考えられる．

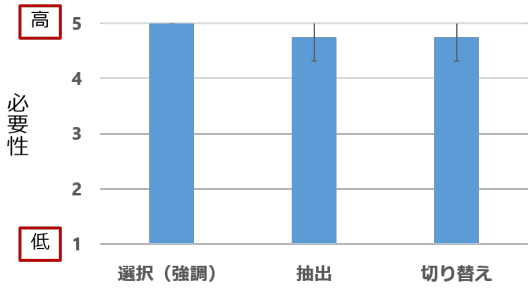

図 11: 提案操作の必要性

7. 考察

精度計測において，上下方向のスワイプは，操作時に手の甲で指が隠れてしまい，認識できなくなってしまったことが低精度の原因であると考えられる．この

実験に参加した被験者は全て右利きであったため，左方向にスワイプを行うと可動域の影響で手の動きがぶれてしまい上手く認識できなかった可能性がある．

探索の所要時間は，コントローラと比較してジェスチャー操作の場合には長くなることが分かった．このことから目標物が遠距離にある場合にジェスチャーによる探索時間が長くなると考えられる．この理由としてジェスチャーとコントローラの遠距離移動方法の違いが挙げられる．コントローラによる移動の際，スティックを倒すことで移動でき，その状態を維持することで遠距離移動が可能になる．しかし，ジェスチャーでの移動では 1 回の操作で決められた距離しか移動できないため，遠距離移動する場合には複数回操作を行う必要がある．また，精度計測の結果からスワイプ操作の一部は精度が悪いため，スワイプでの移動の際に正確な認識がされずに，ユーザが思うように移動できなかったと考えられる．以上のことから，ジェスチャー操作では複雑構造の探索時間は長時間になることが想定される．

操作の難易度は，全ての操作においてコントローラよりもジェスチャーは難易度が高いと判明した．ジェスチャーの移動方法では前述のように複数回行う必要がある．また，一部の操作についても「1 回の動作が大きい」という意見があった．これにより，ユーザの腕への負担が大きくなってしまったことがジェスチャーの難易度増加に繋がったと考えられる．

操作の直感性は，抽出操作においてコントローラよりもジェスチャーは直感性が高いと判明した．操作方法を決定するための調査を行った際には，抽出操作に適した動きとして参加者の大半が「払いのける」動作を行っており，イメージしやすく迷うことなく操作できたと考えられる．一方，コントローラでは切り替え操作を行うボタンと抽出操作を行うボタンは形状が類似しており，位置も近く混乱しやすいため無意識に操作できない．以上のことから，抽出操作においてジェスチャーを用いた場合に直感的となることが想定される．切り替え操作は，抽出操作と同様にコントローラでの操作は混乱するため，ジェスチャーの方が無意識に操作できると考えられたが，大きな差は得られなかった．ジェスチャーでの切り替え操作は，腕の回転を変えることで前の関係に戻ることができないことが問題点として挙げられていた．そのため，ユーザの意図した動作をせず，抽出操作のように直感的に感じにくかったと考えられる．移動操作（スワイプ，ピンチアウト，回転操作）も，切り替え操作と同様にコントローラとジェスチャーで大きな差は見つからなかった．移動ジェスチャーは多くのシステムに用いられており，直感的であると考えられるが，現在は家庭用ゲーム機が普及し，コントローラ操作に慣れている人にとって Oculus Rift のコントローラは直感的であるため，差異が生まれなかったと考えられる．

8. 結論

本稿では，VR を用いたソフトウェア可視化システムを複雑構造の表示にも対応させ，ジェスチャー認識の精度改善を行い，複雑構造探索における提案インタラクション手法の有効性について調査した．実験の結果，複雑な構造ではジェスチャーを使用した探索時間はコントローラを比較して延びることが判明した．全ての操作において，ジェスチャーで実施した際にはコントローラよりも難易度が高くなることが分かったが，ユーザはジェスチャーを用いた抽出操作はコントローラよりも直感的と感じることが確認された．複雑構造の探索において，ユーザは提案した 3 種類の機能は全て必要であると感じていることが判明した．

今後は，一部の操作が複雑構造の探索において適切でなく，腕の負担となることがアンケートで確認されたことからジェスチャー動作の縮小化や遠距離移動への適用を行う．また，仮想環境の情報を確認しながら実環境の作業を行うことのできる MR を本システムに適用し，VR との相違について調査する．

参考文献

[1] Richard Wettel and Michele Lanza. Visualizing software systems as cities. In *2007 4th IEEE International Workshop on Visualizing Software for Understanding and Analysis*, pp. 92–99. IEEE, 2007.

[2] Florian Fittkau, Krause Alexander, and Hasselbring Wilhelm. Exploring software cities in virtual reality. In *2015 ieee 3rd working conference on software visualization (vissoft)*, pp. 130–134. IEEE, 2015.

[3] Romano Simone, Capece Nicola, Scanniello Giuseppe, and Lanza Michele. The city metaphor in software visualization: feelings, emotions, and thinking.

[4] Michael Balzer and Oliver Deussen. Hierarchy based 3d visualization of large software structures. In *Proceedings of the conference on Visualization'04*, pp. 598–4. IEEE Computer Society, 2004.

[5] Leonel Merino, Alexandre Bergel, and Oscar Nierstrasz. Overcoming issues of 3d software visualization through immersive augmented reality. In *2018 IEEE Working Conference on Software Visualization (VISSOFT)*, pp. 54–64. IEEE, 2018.

[6] Leonel Merino, Mario Hess, Alexandre Bergel, Oscar Nierstrasz, and Daniel Weiskopf. Perfvis: Pervasive visualization in immersive augmented reality for performance awareness. In *Companion of the 2019 ACM/SPEC International Conference on Performance Engineering*, pp. 13–16. ACM, 2019.

[7] Roy Oberhauser and Carsten Lecon. Towards virtual reality immersion in software structures: Exploring augmented virtuality and speech recognition interfaces. 2018.

[8] Java ソースからメトリクス抽出・ドキュメント生成 — j クラスレポート — 無償ツール. `http://www.nextdesign.co.jp/uml_java_tools/jclassreport.html` 最終アクセス 2021/6/16.

[9] Unity. `https://unity.com/ja` 最終アクセス 2021/6/16.

[10] Leap motion. `https://www.leapmotion.com/ja/` 最終アクセス 2021/6/16.