VL-SensorIDE: VLM を用いた仮想センサ IDE による インタラクション設計の対象拡張と容易化

戸田 壱星 * 宮下 芳明 *

概要. 本研究は，VLM をユーザがプロンプトで再定義できる仮想センサとして活用し，そのままアプリコードへ配線できる IDE「VL-SensorIDE」を提案する．ユーザは，測定値のフィードバックを見ながら「食事の進行度を 0–1 で返す」といった記述を試行錯誤しながら修正し，新たなセンサを構築できる．これにより，物理センサの設置が困難な対象や，従来の認識モデルでは実装コストが高かった主観的基準・抽象的事象の測定が可能となる．8 名の評価では 20 分で 53 件の多様なセンサが作成された．本研究の成果は，VLM を仮想センサとしてコーディング環境に直結させることで，センシングの対象を拡張し，その設計と実装の自由度を高める点にある．

1 はじめに

インタラクション設計におけるセンシングは，何をどの粒度で測るかという意味的な要件と，どのように測るかという実装手段を適切に結び付けることが要点である．しかし実際の開発では，複数の障壁が存在する．まず，多くのシステムは固定ラベルや固定閾値を前提としており，導入後も含めて個人の主観や用途に合わせて測定基準を柔軟にカスタマイズするのが難しい．また，思い出のぬいぐるみのような侵襲が許されない対象や，センサの取り付けが負担となる生物，液体・影など取り付け自体が不可能な対象では，物理センサを能動的に取り付けることができない．さらに，空間の雰囲気，ユーザの作業負荷のような構成要素が不明瞭な抽象概念を測ろうとすると，要素の洗い出し，個別センサやコンピュータビジョン（CV）モデルの調達，データ統合という長い開発ループを踏む必要があり，試行錯誤を阻んできた．

近年の Vision-Language Model（VLM）は，複雑な視覚入力に対して自然言語で応答できる．この性質を利用し，本研究では VLM をプロンプトの書き換えによって測定基準を再定義できる仮想センサとして活用する．以後，VLM を用いた仮想センサを Vision-Language-based Sensor (VLS) と呼び，視覚入力から数値・真偽・文字列の値を返すものとして定義する．VLS は従来は要素分解や個別実装が前提だった主観的・複合的概念も同一の枠組みで扱える．加えて，非接触であるため，直接取り付けが難しい対象でも測定対象にできる．

この着想に基づき，VLS の設計から実プログラムでの活用までの手順を簡略化し，試行錯誤のサイクルを高速化するために，IDE として VL-SensorIDE を設計・実装した．VL-SensorIDE は，(1) VLM に渡す映像ソースの選択，(2) テーブルで VLS の設計，(3) 任意の VLS をドラッグ&ドロップでコードへ変数として配線，(4) プログラム実行時，配線された VLS の測定値を変数へ代入，の手順を IDE 内で完結させる．

本稿の貢献は，測定基準を自然言語で定義できること，出力を数値・真偽・文字列として同じ枠で扱えること，作成した VLS を IDE のコードに直接配線できる点にある．これにより，インタラクション設計で扱えるセンシング範囲を広げ，その実装を容易にする．さらに，本アプローチの適用可能性を実証するためユーザスタディを実施し，20 分で 53 件もの多様な仮想センサが構築されることを確認した．この結果から，VL-SensorIDE が，センシングの適用範囲を拡張できる有効なアプローチであることが示唆された．

2 関連研究

VLM 登場以前のセンシング支援は，主にアルゴリズムの実装と分析の効率化に主眼が置かれていた．Eyepatch [11] や Exemplar [4] は GUI での例示により分類器を対話的に構築したが，出力は主に分類ラベルや検出領域であり，基準変更には再学習を要した．プログラマ向けの IDE も探求され，VisionSketch は GUI によるデータフロー構築を [5]，Gestalt は機械学習の実装と分析の統合を [12]，DejaVu はカメラベースのプログラムの記録・再生機能を提供した [6]．これらの IDE はアルゴリズムの実装やデバッグを支援したが，処理ロジックの定義はコード記述などに依存していた．特に実世界由来の非再現的なデータはデバッグが困難であり，Exemplar や DejaVu は時系列データの可視化や記録再生機能でこの課題に取り組んだ．エンドユーザ向

* 明治大学

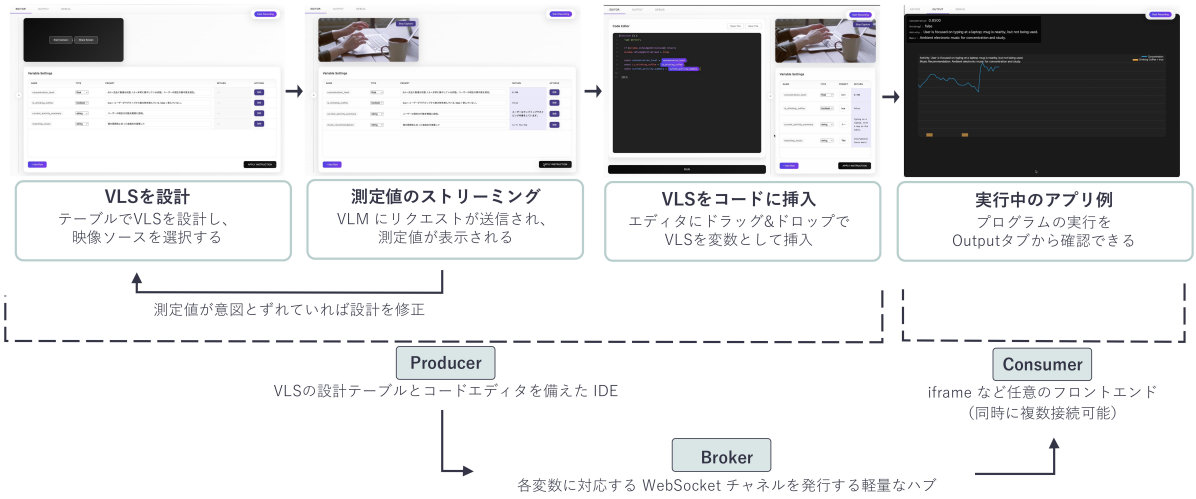

図 1. VL-SensorIDE のシステム構成と処理フロー．

けには Teachable Machine が GUI での学習から JavaScript をエクスポートするワークフローを普及させたが，明示的なデータ収集と学習が前提であり，単一プロンプトでの定義には対応しておらず，また出力も主に分類ラベルのため連続値などを扱う設計ではなかった [2]．

VLM の登場は，実装中心のアプローチから，より宣言的なセンシング定義を可能にした．この潮流に先立ち，Synthetic Sensors は複数の環境センサ信号から高位の状態，例えば電子レンジの完了，を後付けの仮想センサとして抽出可能にした [8]．これにより，従来のスマートカメラの硬直性に対して，後からセンサの定義を追加できる柔軟さを示した．Zensors は自然言語の質問に対しクラウドワーカーが人力で即時応答するハイブリッドな手法を提示したが，常時運用には人件費が課題として残った [7]．この人力による解釈という役割を VLM は部分的に置き換える可能性を持つ．Gensors は，「子どもが家具によじ登ったら」のような任意プロンプトを真偽値で判定したが，外部アプリ連携は手作業であった [10]．GPT-4V をスマートホーム自動化へ流用する事例でも，文脈理解の利点とともに実装や運用上の摩擦が報告されている [14]．一方で，VLM を連続値の計測に使う試みも進んでいる．CrowdCLIP は教師無しで群衆人数を数値として推定し，NumCLIP は数的順序の理解を強化して画像から数値を得る利用範囲を広げた [9, 3]．ただし，いずれもタスク定義は固定で，運用中に測定対象や基準を自由に差し替える前提ではなかった．

以上を踏まえ，本研究は VLM を，プロンプトの書き換えで再定義できる仮想センサとして扱う．そしてこの仮想センサを IDE に直結し，開発フローの工数を抑える運用設計を提案する．本稿の貢献は，測定基準を自然言語で定義できること，出力を数値・真偽・文字列として同一の枠で扱えること，作成した VLS を IDE のコードに直接配線できる点にある．これにより，短周期の更新で基準の記述・出力確認・修正のループを容易に回せる．導入後も，基準変更や出力型の切り替え・組み合わせをコードと一貫した環境で試せる．

3 提案システム

本研究は，VLM を用いた VLS の定義からアプリケーションへの接続までを同一の IDE 内で完結できる IDE「VL-SensorIDE」を提案する．図 1 に示すように，本システムは（1）VLS の定義，（2）測定値の確認と修正，（3）コードへの挿入，（4）アプリの実行という一連の流れを同一の IDE 内で完結できる．本章では，まずシステムの UI と基本的な操作フローを説明し，次いでそれを支えるアーキテクチャと実行サイクルの詳細を述べる．

3.1 UI と操作フロー

本システムの UI と操作フローを図 2 に示す．ユーザはまず，図 2(d) で入力映像ソースを選択する．次に，図 2(e) のテーブルで名前，型，プロンプトを宣言し，「Apply Instruction」ボタンを押すことで VLS の定義をシステムに反映させる．定義が反映されると，システムは 1.0 s 周期でその VLS の測定を開始し，結果を同テーブルの Return 列に表示する．プロンプトを編集して再度ボタンを押すと，次の更新周期から新しい定義が反映されるため，ユーザは Return 列の値の変化を見ながら対話的に測定基準を試行錯誤できる．また，図 2(f) を押下することで図 3 のようにエディタを折りたたみ，VLS の定義に特化したビューで作業することも可能である．

定義した VLS は，図 2 に示すように，変数テー

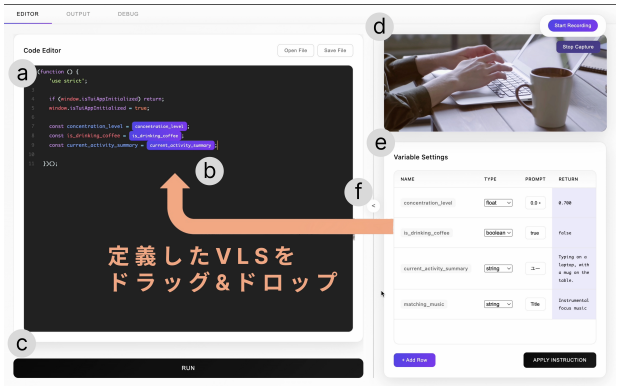

図 2. VL-SensorIDE の画面構成．a：エディタ，b：変数ブロック，c：Run，d：映像ソース，e：変数テーブル，f：エディタ切替．

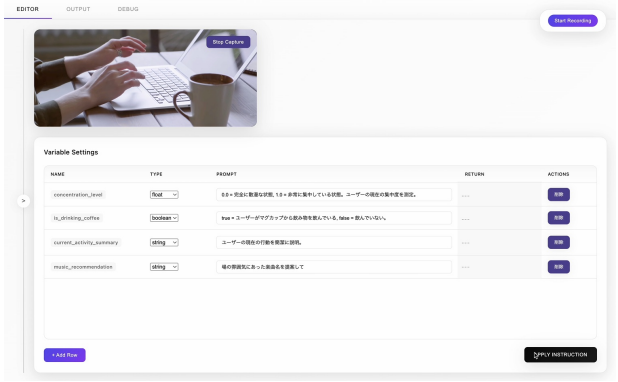

図 3. エディタ折りたたみ時の変数定義ビュー．

ブルからエディタへドラッグ&ドロップすることでコードに挿入できる．挿入された VLS は，エディタ内で図 2(b) のようなブロックとして表示される．最後に，図 2(c) の「Run」ボタンを押すと，これらの VLS が配線されたアプリケーションが Output タブ内で起動する．

3.2 アーキテクチャ

本システムは，役割を分けた Producer-Broker-Consumer の三層構成を採用している．Producer は，ユーザが操作する変数宣言 UI とコードエディタを備えた IDE 本体であり，入力映像の取得と VLM への推論要求の送出を担う．Broker は，Python で実装された WebSocket サーバとして機能し，Producer から受け取った測定値を送信する．そして Consumer は，その値を受け取り描画や反応を行う Web アプリケーションであり，複数クライアントの同時接続をサポートする．この三層構成によって，システムの拡張や再利用を行いやすくなっている．

3.3 実行サイクルと通信

システムのストリーミング処理は，映像キャプチャ，推論，送信，受け取りの順で構成される．このサイクルは 1.0 s 周期で実行される．前の処理が周期内に完了しない場合，そのサイクルは棄却され，常に最新のフレームが次周期の評価対象となることで，システムの応答性を維持している．

入力フレームの推論には，Gemini-2.0-Flash-Live を採用した．このモデルは永続的な WebSocket セッションをサポートしており，一度の接続で続けて推定を行えるため，リクエストごとの HTTP/TLS ハンドシェイクを省ける．また，セッション継続により前回値との整合を保ちやすい．アプリケーション実行では，初回ロード時にコード内のプレースホルダ変数をその時点の値で展開し，以降の更新は新しい測定値のみを差し替える軽量更新で行う．この構成により，編集時の把握しやすさと実行時の更新速度を両立する．

4 評価

4.1 手順

本システムの評価のため，プログラミング経験を持つ学部生 8 名（男性 7・女性 1，平均 20.5 歳）を対象にユーザスタディを実施した．

評価実験は三つのフェーズで構成した. 実験環境として，参加者は全員が同一のハードウェアとブラウザを使用した．まず導入（約 3 分）で VLS をコードへ挿入する操作も含めた IDE の基本操作を実演・習熟させた後，探索・発想フェーズ（20 分）に移った．このフェーズで参加者は，任意の映像ソースを対象に自由に仮想センサを定義・修正し，発想した変数と応用アイデアを記録した．なお，本評価は仮想センサの作成と，それをどのようなプログラムで用いるかの回答までを対象とし，コーディングは評価範囲に含めない．これは，実装の成否や所要時間が個々の技能に依存し，本研究の検証対象を VLS の定義と活用構想に置いているためである．この記録から発想成果を評価した後に，事後フェーズ（約 15 分）で半構造インタビューを実施し質的知見を得るとともに，日本語版 SUS [1, 13] を回収してユーザビリティを測定した．

4.2 定量的結果

表 1 に示すように，20 分のタスクで合計 53 件（平均 6.6）の仮想センサが生成された．型は float が 66% と最多で，boolean と string が続いた．SUS は平均 61.4 と，初期版 IDE として許容域に達していることを示す．

表 1. 定量的結果

項目	結果
生成センサ総数	53（平均 6.6/人，SD 1.8）
型内訳	float 35 (66%)/boolean 8 (15%)/string 10 (19%)
SUS (n = 8)	平均 61.4，SD 8.8

表 2. ユーザが作成した仮想センサの分類

クラス	例	件数
A. 主観・情動系	かわいさ，盛り上がり，論理一貫性，幸福度	16
B. 環境/生活モニタ	明るさ，机の汚さ，風力，混雑度	10
C. TUI/物体操作	コップ水量，水筒保持，目線座標，お絵描き支援	10
D. 自己管理・ヘルスケア	集中度，体調，食事健康度	7
E. eSports/開発支援	FPS スキル，カメラワーク，CPU 負荷	6
F. 生成・創作系	ストーリー生成，レシピ提案	4

4.3 仮想センサの作例

参加者が作成したセンサは，表 2 に示すように多様なカテゴリに及んだ．主観・情動系が全体の 30%を占めた一方で，62%は環境センシングや自己状態の定量化など，既存ハードセンサの代替・補完を狙う実利指向であった．さらに 8%は「ストーリー」や「レシピ」など，文字列型の出力を試みる例で構成された．

例えば，主観・情動系のカテゴリでは,「キモ可愛さ」やイベントの「盛り上がり度合い」といった，個人の主観に基づき測定基準を柔軟にカスタマイズする必要がある対象が多数試された．ある参加者 (P8) は，Return 列のライブ値を見ながらプロンプトを対話的に修正することで自身の感覚に近い出力を得られたと述べた．こうした主観的指標が容易に得られることから,「作品の格付けチェック」や「共同作業の活性度を可視化する」といった応用アイデアが構想された．

TUI/物体操作のカテゴリでは,「コップ内の水量」や「ティッシュが使用されたか」を測定する VLS が作成された．これらは，物理センサの取り付けが困難な液体や，軽量かつ使い捨てであるため従来のセンサでの検知が難しい対象に対し，非接触でセンシングを行える本システムの特徴を示している．参加者からは，これらの VLS を Tangible User Interface（TUI）における入力手法として活用するという応用が提案された．

生成・創作系のカテゴリでは,「卓上の物体を登場人物とした物語」や「映像に映っている食材からのレシピ提案」といった文字列型の VLS が試された．前者の例では，物体が動いたり新しい物体が登場したりすると，その状況変化を反映した物語が生成されていき，子供向けのインタラクティブなストーリーテリングへの応用が考案された．後者は，料理動画と冷蔵庫の中身を組み合わせた献立提案アプリへの応用が想定された．これらは，数値データを返す従来のセンサとは異なり，視覚的文脈の解釈そのものをセンサ値として直接テキストで出力する応用例である．これらの事例は，VL-SensorIDE が，物理量の測定から主観的な事柄の定量化，さらには文脈的なテキスト生成まで，広範な対象を同一の枠組みで扱えることがうかがえた．

4.4 質的分析

インタビューの質的分析から，主に以下のテーマが浮かび上がった．参加者からは，プロンプトによって測定基準を柔軟に定義できるため，かわいさや論理一貫性といった従来は定量化が困難であった主観的概念も計測対象にできた点が指摘された．また，ハードウェアを導入せずに仮説検証が行えることから，本格的な実装前の試行段階で有用なプロトタイピングツールとしての価値も確認された．一方で，測定基準のテンプレートや変数命名補助といったプロンプト設計支援機能の必要性が示唆された．さらに，VLM にプロンプトを提案してほしいといった，より対話的な設計支援への要望も複数聞かれた．加えて，医療や安全が関わる高リスクなドメインにおいては，VLM による単独判定に依存することへの懸念も示された．

作成された仮想センサの有効性については，測定対象の性質によって傾向が見られた．解釈の幅を許容できる主観的な概念，例えばキモ可愛さやストーリー生成では，自身の感性と合っていたなど肯定的な評価が得られた．また，人の数やティッシュの使用有無のような要素が単純な判定も概ね安定して機能した．これらのケースでは，出力が短時間で更新されるため，参加者は自身の意図と結果を迅速に照合できる点を評価していた．これに対し，fps の上手さや面白さのように，複数の要素を統合して判断する高次な概念をプロンプトで定義した場合は，意図と出力の間に乖離が顕在化した．しかし複数の参加者は，これはシステムの限界ではなく，プロンプトに具体的な観点を示すことで改善できると考えていた．

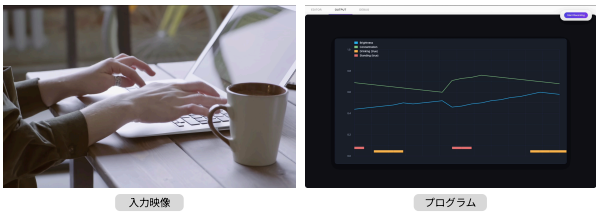

図 4. デスクワークモニタリングの実行画面.

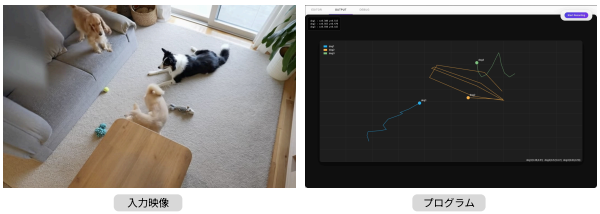

図 5. 複数オブジェクト追跡の実行画面.

5 応用事例

本章では，VL-SensorIDE の具体的な活用例として，性質の異なる 3 つのユースケースを詳述する．各事例では，その目的と実装プロセスを述べ，得られる考察を示す．

一つ目に，スマートオフィスなどにおいて，従来は複数の物理デバイスを要した環境・ユーザ状態の把握を，単一の映像ソースで低コストに実現する応用例を構築した（図 4）．ウェブカメラ映像のみを入力とし，「brightness (float)」，「concentration_level (float)」，「is_drinking (boolean)」といった，性質の異なる複数の仮想センサを同時に定義する．これにより，追加ハードウェアなしで作業環境とユーザの状態を統合的にモニタリングできる．本事例は，単一の非接触センサから，物理量，ユーザの内的な状態，断続的な行動といった，複数の異なるレイヤーの情報を同時に抽出できることを示している．

二つ目に，センサ装着が困難または負担となる対象に対して，非接触で行動追跡を行う応用例を構築した（図 5）．本アプローチの特徴は，CV 手法のように事前に定義された検出対象や基準に縛られない点にある．CV モデルは学習済みの特定カテゴリしか認識できないが，本システムではプロンプトを「青い首輪の犬を追跡して」のように修正するだけで，マーカーや特定の種に依存することなく，その場で柔軟に対象を再定義できる．

最後に，専用ハードウェアを用いることなく，TUI を即席に構築する応用例を示す（図 6）．紙に描いた十字キーへの指の接触を検出するため，「`is_up_arrow_touched (boolean)`」や「`is_down_arrow_touched (boolean)`」といった仮想センサ群を定義した．これらの出力をキー入力にマッピングする

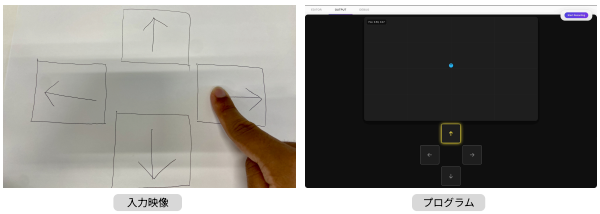

図 6. 紙を用いた即席 TUI の実行画面.

ことで，ウェブカメラが紙に描かれた十字キーを機能的なコントローラーとして認識し，操作入力へ変換する．特別な機材を必要とせず，身近な素材でインタラクティブな装置をその場で具現化できる手軽さは，物理デバイスの形態を多様にする可能性がある．

6 議論

6.1 設計判断とトレードオフ

本研究は，専用学習器や追加ハードウェアではなく，VLM へのプロンプト定義だけで視覚入力から数値・真偽・文字列の値を得る設計を採った．この方針により，導入後でもプロンプトの再定義だけで測定基準を切り直せる．ユーザスタディでは，参加者がプロンプトを修正しつつ自らの要求に合う測定基準へ収束していく様子が観察され，この再定義の容易さが試行錯誤のサイクルを加速させていることがうかがえた．また，VLS をコードへ直接配線できるため，抽象概念に対しても長い開発ループを経ずに試行を開始できる．

一方で，非接触で測定が可能であるため，センサの装着自体が困難な対象でも，まずは測定の当たりを付けることができる．ただし，本方式は視覚手掛かりに限定され，視野外・遮蔽・外観が似た対象に弱く，VLM の出力揺らぎへの対処も不可欠である．さらに，曖昧な基準では値が不安定になりやすく，高リスク領域で単独判定に依存することには慎重さが求められる．また，本システムの VLS 測定は使用した VLM の推論処理時間の制約に基づいて 1.0 s 周期で実行している．そのため，視線移動や高速ゲームプレイのような短時間の事象は取りこぼしうる明確な制約がある．

また，VLM API を直接利用する場合は連続実行や複数プロンプト管理，コード反映といった負担が大きいが，本手法は IDE がこれらを隠蔽することで，開発者がプロンプトの試行錯誤やプログラムの作成に集中できる点に利点がある．その反面，本 IDE の設計は，API リクエストのタイミングやエラーハンドリングといった低レイヤの自由度を制約する側面もある．

以上より，本設計は初期の実装負担を最小化し，基準探索と適用開始までの時間を大幅に短縮する利

点をもたらす一方で，視覚情報への依存，出力の安定性，IDE 設計に起因する適用範囲・信頼性といった制約がある．

6.2 VLS の妥当性

4.4 節の所見から，妥当性は測定対象の性質に強く左右される．人数や有無，簡単な程度のように要素が単純な事象は多くの場面で安定して扱えた．一方で，厳密な精度が求められる場面では，VLM の性能や視覚条件に起因する出力の不確実性が課題となる．対照的に，感情や雰囲気といった主観的・抽象的な事象は，プロンプトを整えていくことで出力を意図に近づけやすい．プロンプトについては，測定するにあたっての観点や条件などを与えるといった一定の労力を必要とする．しかし，全てのユーザが即座に的確なプロンプトを記述できるわけではない．加えて，毎回プロンプトを人手で詰めるのは負担が大きい．ユーザが効果的なプロンプトを効率的に見つけ出すためのデバッグ機能や補助ツールが重要である．この課題は，6.4 節で述べる VLS の設計補助機能の重要性へと直結する．また，VLS が実務でどの程度の水準を満たすかを示すための定量的な性能確認は今後の課題とする．

6.3 文字列出力の位置づけ

本稿では，数値型のセンサを単一軸を単一変数で測る一次元の測定と捉える．これに対し文字列型のセンサは，明るさ・色味・にぎやかさ・落ち着きのような複数の意味軸を同時に含む多次元の測定として位置づける．この前提のもとで，数値と文字列の関係を圧縮と抽出という変換で整理できる可能性を示す．圧縮は，複数の数値を一つの文字列にまとめることを指す．これは，個別の数値を並べて表示するだけでは捉えにくい，全体的な状況や文脈を一つの意味的なまとまりとして表現する試みである．多数の数値を個別に解釈せずとも全体像を直感的に把握でき，文字列は単なるキーワード検索を超え，意味的な近さに基づく検索の手がかりとなりうる．抽出は，文字列から必要な数値だけを取り出すことを指す．例えば「西日の差す，穏やかな午後」という一つの測定値から暖色度，明るさ，落ち着きといった複数の指標を推定する．これにより，当初測定対象として想定していなかった観点，例えば「集中しやすさ」といった指標を，後から同じ文字列データを用いて推測できる可能性がある．なお，可逆性と手続きの一般化は今後の検証課題とする．

6.4 VLS 設計補助

現行プロトタイプは，変数定義 UI と値プレビュー，ログ記録を備える一方で，型推定，変数名補完，測定基準テンプレート，および対話的にプロンプトを作成できる仕組みは未実装である．ユーザスタディではこれらへの要望が複数挙がり，基準づくりの初動を支援することが試行速度と精度安定の双方に寄与する示唆が得られた．加えて，評価参加者からは「プロンプトを改善すれば，より望んだ値が得られそう」という声が複数聞かれた．これは，本アプローチの中心的な作業が，プロンプトを対話的に洗練させるプロセスであることを示唆している．しかし現状の値のプレビューだけでは，予期せぬ値が出力された際に，どの瞬間の入力映像が原因だったのかを特定したり，過去のプロンプト修正との比較を行ったりすることが難しい．このプロンプトの洗練プロセスをさらに深化させるため，関連研究で示された記録・再生のようなアプローチに基づき，入力映像，プロンプトの変更履歴，そして出力値の時系列を統合的に可視化・再生できるタイムラインベースのデバッグ機能が有効だと考える．

6.5 限界と展望

本評価は 8 名・短時間の探索的条件にとどまり，長期運用や効果量を伴う統計的検証は今後の課題である．また Gemini-2.0-Flash-Live 以外のモデルでの検証も未実施である．さらに本システムが扱えるのは視覚的手掛かりに限られ，他モダリティを必要とする事象は測定できない．今後は，設計補助，デバッグ支援，時間分解能の向上，定量的検証，および文字列出力と数値出力との往還に関する可逆性と手続きの一般性の検証に取り組む．これにより，本システムの適用可能な領域と使用上の前提を明確化し，再現性と運用上の信頼性を段階的に高める．

7 結論

本研究は，プロンプト定義という統一されたインタフェースを通じて，従来は個別の専門技術を要した多様なセンシングタスクを，エンドユーザが専門知識なしに定義・実装できる IDE「VL-SensorIDE」を提案した．8 名による初期評価では，20 分という短時間で 53 件の仮想センサが生成され，本アプローチの即時性と実用性が示された．本研究の貢献は，測定基準を自然言語で定義できること，出力を数値・真偽・文字列として同じ枠で扱えること，作成した VLS を IDE のコードに直接配線できる点にある．これにより，従来の物理量や状態の推定に留まらず，文脈的な解釈を文字列として出力するという，新しいセンシング形態をも扱う，新たな設計思想を提示した．

参考文献

[1] J. Brooke. SUS: A Quick and Dirty Usability Scale. In *Usability Evaluation in Industry*, pp. 189–194. Taylor & Francis, London, UK, 1996. Originally published by Digital Equipment Corporation in 1986.

[2] M. Carney, B. Webster, I. Alvarado, K. Phillips, N. Howell, J. Griffith, J. Jongejan, A. Pitaru, and A. Chen. Teachable Machine: Approachable Web-Based Tool for Exploring Machine Learning Classification. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI EA '20, pp. 1–8, New York, NY, USA, 2020. Association for Computing Machinery.

[3] Y. Du, Q. Zhai, W. Dai, and X. Li. Teach CLIP to Develop a Number Sense for Ordinal Regression. In *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, Sept. 29–Oct. 4, 2024, Proceedings, Part LXXXV*, Vol. 15143 of *Lecture Notes in Computer Science*, pp. 1–17. Springer, Cham, 2024.

[4] B. Hartmann, L. Abdulla, M. Mittal, and S. R. Klemmer. Authoring Sensor-based Interactions by Demonstration with Direct Manipulation and Pattern Recognition. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '07, pp. 145–154, New York, NY, USA, 2007. ACM.

[5] J. Kato and T. Igarashi. VisionSketch: Integrated Support for Example-centric Programming of Image Processing Applications. In *Proceedings of Graphics Interface 2014*, GI '14, p. 115–122, Toronto, Ontario, Canada, 2014. Canadian Information Processing Society.

[6] J. Kato, S. McDirmid, and X. Cao. DejaVu: Integrated Support for Developing Interactive Camera-Based Programs. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*, UIST '12, pp. 189–196, New York, NY, USA, 2012. ACM.

[7] G. Laput, W. S. Lasecki, J. Wiese, R. Xiao, J. P. Bigham, and C. Harrison. Zensors: Adaptive, Rapidly Deployable, Human-Intelligent Sensor Feeds. In *Proceedings of the 33rd Annual CHI Conference on Human Factors in Computing Systems*, CHI '15, pp. 1935–1944, New York, NY, USA, 2015. Association for Computing Machinery.

[8] G. Laput, Y. Zhang, and C. Harrison. Synthetic Sensors: Towards General-Purpose Sensing. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pp. 3986–3999, New York, NY, USA, 2017. Association for Computing Machinery.

[9] D. Liang, J. Xie, Z. Zou, X. Ye, W. Xu, and X. Bai. CrowdCLIP: Unsupervised Crowd Counting via Vision–Language Model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, CVPR '23, pp. 2893–2903, Piscataway, NJ, USA, 2023. Institute of Electrical and Electronics Engineers.

[10] M. X. Liu, S. Petridis, V. Tsai, A. J. Fiannaca, A. Olwal, M. Terry, and C. J. Cai. Gensors: Authoring Personalized Visual Sensors with Multimodal Foundation Models and Reasoning. In *Proceedings of the 30th International Conference on Intelligent User Interfaces*, IUI '25, pp. 755–770, New York, NY, USA, 2025. Association for Computing Machinery. arXiv:2501.15727.

[11] D. Maynes-Aminzade, T. Winograd, and T. Igarashi. Eyepatch: Prototyping Camera-based Interaction through Examples. In *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology*, UIST '07, pp. 33–42, New York, NY, USA, 2007. ACM.

[12] K. Patel, N. Bancroft, S. M. Drucker, J. Fogarty, A. J. Ko, and J. A. Landay. Gestalt: Integrated Support for Implementation and Analysis in Machine Learning. In *Proceedings of the 23rd Annual ACM Symposium on User Interface Software and Technology*, UIST '10, pp. 37–46, New York, NY, USA, 2010. ACM.

[13] K. Sato, N. Mitomi, K. Kon, and H. Haruna. Reliability of the System Usability Scale in the Field of Prosthetics and Orthotics. *Journal of the Japanese Academy of Prosthetists and Orthotists*, 30(1):32–37, 2022.

[14] S. Yun and Y. kyung Lim. What If Smart Homes Could See Our Homes?: Exploring DIY Smart Home Building Experiences with VLM-Based Camera Sensors. In *CHI Conference on Human Factors in Computing Systems*, CHI '25, pp. 1–22, New York, NY, USA, 2025. Association for Computing Machinery.