明るさ変換手法による デジタルツイン画像と実画像間の AI 精度差の低減

池田　佳弘† 　中村　建一†† 　史　旭† 　三上　啓太† 　江田　毅晴†

† NTT ソフトウェアイノベーションセンタ
〒 180–8585 東京都武蔵野市緑町 3–9–11
†† 東京工業大学

E-mail: †yoshihiro.ikeda.ap@hco.ntt.co.jp, ††nkennichi@gmail.com

あらまし　監視カメラ向け映像解析 AI システムにおいて，運用前の作業に要するコストと負担の低減に向けた EyeTuner フレームワークを提案する．EyeTuner は，デジタルツイン上で映像解析 AI の精度が十分となるカメラの設置パラメータ（位置，向き，露出等）と映像解析 AI のハイパーパラメータ（使用する AI モデル、推論時の閾値等）を自動探索し，運用前の作業を自動化する．ここで，EyeTuner ではデジタルツインに対する AI の精度を基にパラメータを探索するため，デジタルツインに対する精度は実環境に対する精度と一致する必要がある．しかし，精度が一致するかを検証したところ精度に差が生じた．本研究では，この精度差を低減するため，デジタルツイン画像の明るさを実環境画像の明るさに近づける，輝度変換手法と GAN 変換手法を提案し，効果を検証した．結果として，精度差を平均 6.64 ポイントまで低減した．

キーワード　デジタルツイン，監視カメラシステム導入の課題低減，AI 精度差低減

1　はじめに

深層学習を用いた画像認識（以下，映像解析 AI 又は単に AI と呼ぶ）に注目が集まっており，人物検知 [1] や姿勢推定 [2]，人物照合 [3] など様々なタスクが映像解析 AI で処理されている．

このような映像解析 AI を用いた監視カメラシステムを実運用する場合，カメラのパラメータ（3 次元の設置位置，回転角度，画角，露出設定等）と，AI のハイパーパラメータ（既存のどの AI モデルを使用するか，推論時の閾値設定等）が映像解析 AI の精度に影響を与え，AI の精度が低下する場合がよくある．そのため精度の低下を防ぐために，システム導入前に試行錯誤してパラメータを調整する作業が必要となる．一般に，パラメータの調整作業は以下のような手順で行なわれている．

（1）　カメラの被写体となるエキストラを雇う

（2）　カメラの設置パラメータと，AI のハイパーパラメータを手動で変更する

（3）　エキストラをカメラで撮影し，AI の精度を確認する

（4）　（3）の精度が，ユーザ定義された閾値以上になるまで，（2）と（3）を繰り返す

しかし，調整作業を行なう場合，2 つのことが課題となる．1 つは，複雑な状況を再現するための人件費（以下，コスト）が高いことである．実際の監視現場では，監視対象は服装，性別，人数などの様々な属性を持つが，それらすべての状況を再現する場合，被写体となるエキストラを雇う人件費（以下コスト）も増え，コストは高くなる．もう 1 つは，カメラや AI のパラメータを変える度に，エキストラの撮影と，撮影後の精度確認を行なう必要があり，作業者にとっての負担が大きいことである．

図 1: EyeTuner フレームワーク

ところで，近年ではデジタルツインと呼ばれる，実環境の物理的なオブジェクトの特性，状態，動作をデジタル空間上に再現し，ソフトウェア上で実環境のシミュレーションを行なう技術が登場した．そこで本研究では，デジタルツインの考え方を参考にして，コストと負担を低減する EyeTuner フレームワークを提案する．EyeTuner では 3D 空間上に人物の 3D モデルが配置されたデジタルツイン内を撮影した映像（以下，デジタルツイン画像）に対し，人物の 3D モデルに対して映像解析 AI の精度を算出して，その精度が最も高くなるようなカメラの設置パラメータと AI のパラメータを自動で探索する．EyeTuner によってエキストラを人物の 3D モデルに置き換えることでコ

ストを削減し，探索の試行錯誤を自動化することで負担を低減することが可能になる．図 1 に，EyeTuner フレームワークを図示する．

ここで，EyeTuner ではデジタルツイン画像に対する映像解析 AI の精度を基にパラメータを探索するため，デジタルツイン画像に対する精度は実環境を撮影した画像（以下，実画像）に対する精度と一致する必要がある．しかし，デジタルツイン画像と実画像に対する AI の精度が一致するかを検証したところ，精度には差が生じることが判明し，その要因としてデジタルツインにおける 3D 空間の明るさや色味が実環境と異なったことが挙げられた．

この精度差を低減するため，本研究では輝度を調整することでデジタルツイン画像の明るさを実画像に近づける輝度変換手法と，GAN によって明るさや色味を実画像に近づける GAN 変換手法という 2 つの手法を提案し，実験により精度差の低減効果を確認した．結果として，精度差の低減効果を確認し，差を平均 6.64 ポイントまで低減した．

本章の構成は以下の通りである．2 章では，カメラの最適なパラメータの探索やデジタルツインに関する既存研究，及びデジタルツインの構築に利用する 3 次元再構成技術について説明する．3 章では，デジタルツインの作成フローについて説明し，実際にデジタルツインを構築して，デジタルツイン画像に対する AI の精度と実画像に対する AI の精度が一致するかを検証する．4 章では，デジタルツイン画像と実画像間の精度差を低減するための輝度変換手法と GAN 変換手法について説明し，5 章と 6 章で手法適用の結果と考察を述べ，7 章で EyeTuner に残された課題を述べる．

2 関 連 研 究

2.1 監視カメラの配置探索

監視カメラの適切な設置場所を自動で探索する既存研究として，カメラの視野が監視対象を死角なくカバーすることを指標として探索する手法 [4–7] がある．Niccoló ら [7] は，Unity 上に再現した仮想の室内環境において，粒子群最適化手法を用いて，室内を模した 3D モデルに対し，複数の仮想のカメラを用いて，最も死角をなくすようなカメラ配置設定を探索するデモを作成した．しかし，これまでの研究では，本研究が対象とするような映像解析 AI の精度を指標として配置探索を行なっていない．

2.2 デジタルツイン

デジタルツインの定義は様々あるが，[8] らは，「デジタルツインとは，個々の物理的なオブジェクトをデジタル空間上に表現したものである．デジタルツインにおいては，実環境の物理的なオブジェクトの特性，状態，動作を，デジタル空間上に表現する．デジタルツインでは，表現された仮想的なオブジェクトの集合を用いて，実環境の物理的なオブジェクトの挙動をシミュレートすることに活用可能である.」と説明している．

これまでのデジタルツインの適用先として，[9, 10] の調査によれば，デジタルツイン建物のレイアウトや設備構成の設計に用いること，デジタルツインと古典的な機械学習技術を組み合わせて製品の故障を予知すること，ヒトデジタルツインのような，人間の臓器の 3D モデルを作成し，医療に活用することなどが行なわれてきた．また，近年では機械学習とデジタルツインを組み合わせて，ロボットの動作をデジタルツイン上で訓練する研究も行われている [11, 12].

このように，デジタルツインと機械学習を組み合わせて活用する取組は進んでいる一方で，我々の想定するようにデジタルツインを映像解析 AI の精度シミュレータとして用いる研究はなく，その有効性や技術課題は明らかではない．

2.3 3 次元再構成技術

3 次元再構成とは，カメラ等のセンサでスキャンした映像や点群情報等から，(テクスチャ付き）メッシュと呼ばれる，(色情報付き）3 次元形状をコンピュータ上に再現する手法である．3 次元再構成の技術としては，写真から形状を構築するフォトグラメトリと呼ばれる手法や，深度センサや LiDAR センサから得られる点群から構築する SLAM と呼ばれる手法 [13] などが存在する．フォトグラメトリでは，予め撮影済みの複数の画像から特徴点を抽出し，複数の画像間で特徴点マッチングを行なって画像間で共通する特徴点を見つけ，各画像を撮影したカメラの位置関係を把握し，ステレオマッチングアルゴリズムを用いて深度推定を行ない，(テクスチャ付き）メッシュを生成する．SLAM では，センサから取得した点群や画像情報からリアルタイムにセンサの位置推定と特徴点マッチングを行なって，メッシュを構築する．LiDAR を用いたスキャンは従来は高価な機材が必要だったが，近年では iPhone や iPad に LiDAR センサが実装され，より安価に 3 次元再構成を行なえるようになった．また，深層学習モデルを用いて 3 次元再構成を行なう研究 [14] も進んでいる．本研究ではこれらの 3 次元再構成技術を用いてデジタルツインを構築し，映像解析 AI の精度が一定の閾値を超えるカメラのパラメータと AI のハイパーパラメータの探索に利用する．

3 EyeTuner とその技術課題

3.1 EyeTuner

本研究では，カメラのパラメータ（3 次元の設置位置，回転角度，画角，露出設定等）と，AI のハイパーパラメータ（既存のどの AI モデルを使用するか，推論時の閾値設定等）について，調整作業のコストと負担を低減する EyeTuner フレームワークを提案する．EyeTuner は，(1) デジタルツインの構築，(2) パラメータの設定，(3) 処理・閾値判定の機能から構成される．図 2 に，EyeTuner のフローチャートを示す．

デジタルツインの構築は，ゲームエンジン上に 3D 空間モデルを作成・配置（図 2 中 S3）と，その中に人物の 3D モデルを配置（S4）を行なう．3D 空間モデル作成・配置（S3）では，2.3 節で述べた手法を用い，カメラや LiDAR センサからの情報を入力として（S1），映像解析 AI の導入現場の空間をテク

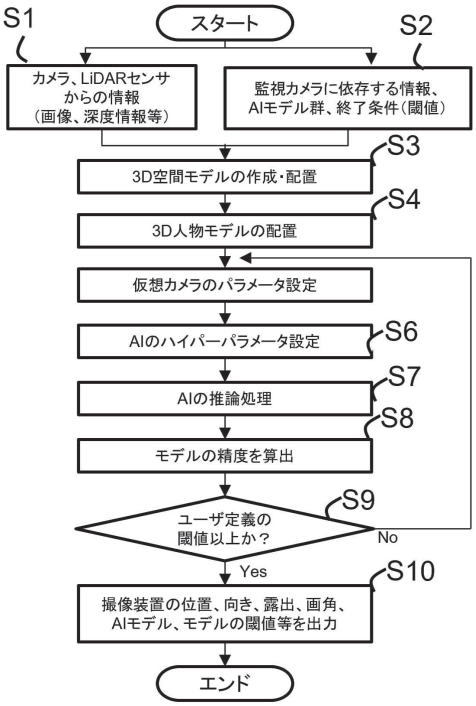


図 2: EyeTuner のフローチャート

スチャ付きメッシュ（以下，3D 空間モデル）として再構成し，3D 空間モデルを Unity などのゲームエンジン上に配置する．人物 3D モデルの配置（S4）では，予めゲームエンジン内に服装，性別，人数，姿勢など様々なバリエーションの人物 3D モデルを用意し，その中から人物 3D モデルを選択し，3D 空間モデル上に配置する．

パラメータの設定では，仮想カメラのパラメータ設定（S5）と，AI のハイパーパラメータ設定（S6）を行なう．仮想カメラのパラメータ設定（S5）では，ゲームエンジン上で仮想のカメラの 3 次元の設置位置，回転角度，画角，露出設定等を設定する．なお，実運用で使用する監視カメラに依存する設定（センササイズ，撮影解像度，焦点距離等）については，予め入力として与え（S2），仮想カメラの初期値として反映する．AI のハイパーパラメータ設定（S6）では，使用する AI モデルの候補（群）を入力として与え（S2），その中から既存のどの AI モデルを使用するか，及び推論時の閾値設定等を設定する．

処理・閾値判定では，パラメータが設定された仮想のカメラと AI モデルを用いて，仮想のカメラに映る映像，すなわちデジタルツイン画像に対する AI の推論処理を行ない（S7），精度を算出して（S8），精度がユーザが設定した閾値（S2）を超えるかどうかを判断する（S9）．閾値を超えない場合，カメラと AI のパラメータを更新し，推論処理を行ない，再度精度を算出する．この更新と精度算出を繰り返し，例えば精度が閾値を超えた時に，カメラと AI のパラメータを出力する（S10）．なお，更新と精度算出の終了条件は一例であり，例えば更新の試行回数を基に，十分な試行回数がなされたかどうかで判断することも考えられる．

EyeTuner によるパラメータ更新が完了した後（すなわちエンド以降），実環境では，この出力（S10）を実環境のカメラと AI のパラメータに人手で反映する．EyeTuner によって，エキストラを人物の 3D モデルに置き換えることでコストを抑え，かつパラメータの探索を自動化することで，作業者の負担を低減することが可能になる．

ここで，EyeTuner ではデジタルツイン画像に対する映像解析 AI の精度を基にパラメータを探索するため，同じパラメータ設定においてデジタルツイン画像と実画像に対する映像解析 AI の精度は一致する必要があるが，一致するかは不明である．例えば，デジタルツイン画像に対して精度が良いカメラの位置は，実環境では精度が悪い位置になっている可能性がある．そこで研究では，実験を行ない同じパラメータ設定時におけるデジタルツイン画像と実画像に対する AI の精度が一致するかを検証した．

3.2 デジタルツイン画像と実画像に対する AI の精度比較

本実験の目的は，パラメータ設定を揃えた時にデジタルツイン画像と実画像に対する AI の精度が一致するかを検証し，一致しない場合にはその要因を明らかにすることである．我々が想定する映像解析 AI のアプリケーション（例えば人物追跡）では，一般に物体検知 AI モデルによって人物検知することを前処理として行うため，人物検知の精度が最終的なアプリケーションの精度に強く影響する．そこで本実験では，映像解析 AI として人物検知タスクを想定し，デジタルツイン画像に対する人物検知の精度と，実画像に対する物検知の精度が一致するかを確認した（3.2.3 節）．実験に向けて，実画像のデータセット（3.2.1 節）と，3.1 節で説明した方法で構築したデジタルツイン画像のデータセットを用意し（3.2.2 節），それぞれに対する人物検知 AI モデルの精度を算出して比較した．

3.2.1 実画像のデータセット作成

実画像のデータセット作成に向けて，NTT 武蔵野研究開発センタ内の CLIC という休憩スペースに 10 台のカメラ（GoPro 9）を設置し，撮影を行なった．CLIC は縦幅約 12m，横幅約 13m，高さ約 4m の直方体に近い形状の空間であり，椅子や机，ソファ，本棚などの，人物を遮蔽するオブジェクトが多数存在する．CLIC は窓から日光を取り込んでおり，日光が強いと逆光が強くなり，日光が弱いと逆光も弱くなる．また，被写体となる CLIC の利用者の属性は成人以上であり，男女比や服装，年齢，体型，姿勢（立つ，歩く，座る等）は様々である．以上から，CLIC は我々が想定する映像解析 AI 運用時の現場に近い条件を備えていると考えられる．

今回はカメラの設置条件として，本棚や人に近すぎず，かつ CLIC の中心が画角内に映ることと設定した．そして，設置条件の下で様々な地点に設置し，各カメラに 1 から 10 までの番号を付け，後述する実験にて用いた．なお，カメラの高さの違いについては今回は考慮せず，全てのカメラはほぼ同じ高さに設置した．図 3 に，CLIC を上から撮影した図（デジタルツインを活用して撮影）と，カメラのおおよその設置場所と向きを示す．

次に，この 10 台のカメラを用いて，5 日間撮影を行なった．

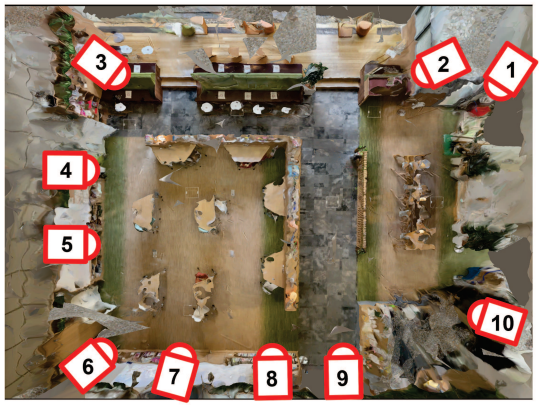


図 3: カメラの設置場所

図 4: 実画像とデジタルツイン画像の例

撮影時のフレームレートは 24fps であり，解像度はフル HD で，1 日あたり約 2 時間撮影して，1 カメラあたり約 10 時間の映像を取得した．そして，各カメラごとに映像を 5fps でフレーム化し，個々の人物を囲う矩形アノテーションを手動で付与した．このアノテーションは，後述の映像解析 AI の精度を算出するための正解データとして用いた．

3.2.2 デジタルツイン画像のデータセット作成

デジタルツイン画像のデータセット作成に向け，3.1 節で述べた方法に従い，デジタルツインを構築した．初めに，3D 空間モデルの作成・配置を行なった．2.3 節で述べたように，3 次元再構成技術は様々あるが，EyeTuner を実際に利用する場合，デジタルツインを構築するための実環境をスキャンする時間は短いことが望ましい．また，iPad の LiDAR センサを用いて既存のアプリケーションを試したところ，スキャンする空間が広くなるとアプリケーションが処理落ちすることが分かった．加えて深層学習を用いた手法は，深層学習モデルの中に格納された 3 次元形状を，クオリティを保ったまま外部に取り出すことが技術的に難しい．これに対してフォトグラメトリ技術は，動画から切り出したフレームから再構成できるため手軽にスキャンが行なうことができ，今回検証する CLIC のような広い環境であっても処理落ちせずに再構成が行なえる上，再構成結果を.obj ファイルのような標準的なモデルデータとして取り出すことができるため，本研究では 3D 空間モデルの作成にフォトグラメトリを用いた．フォトグラメトリ用のデータとして，CLIC 内を iPhone 11 Pro を用いて倍率 0.5 の動画モードで約 7 分間撮影し，撮影した映像を 1 秒あたり 1 枚のフレームに切り出した，計 418 枚の写真を用いた．そしてフォトグラメトリのアプリケーションの 1 つである 3DF Zephyr [15] アプリを用いて，写真からテクスチャ付きメッシュを構築し，ゲームエンジンの一つである Unity にインポートした．

次に，3D 空間モデル上に人物 3D モデルを配置した．使用した人物 3D モデルは，RENDERPEOPLE [16] の販売する，様々な服装，性別，姿勢（立つ，座る，歩く）の付いた人物の 3D モデルであり，予め Unity 上にインポートした．そしてその中から，ランダムに人物の 3D モデルを選択してランダムな位置に配置した．ただし人物 3D モデルの位置の制約として，実環境で起こりえない状況は排すように手動で調整した．具体的には，空間の 3D モデルの床面に人物の 3D モデルの足が接着するようにし，かつ人物の 3D モデルが机やソファなどに埋没するような状況は排した．そして，人物 3D モデルが配置された状態を 1 シーンとしてカウントし，人物 3D モデルの配置位置が異なる計 30 シーンを用意した．なお，各シーンにおける人物の 3D モデルの数は，実画像のデータセットとおおよその人数の分布を合わせた．

最後に，Unity 内の仮想カメラに対し，実環境の監視カメラ（GoPro9）と同じセンササイズ，画角，解像度を設定して，実環境のカメラと同じ位置・角度に 10 台設置して 1 から 10 まで番号を付けた．そしてカメラ 1 台につき用意した 30 シーンを撮影して，カメラ 10 台で計 300 枚の映像を用意し，映像内の個々の人物 3D モデルに対して正解データとなる矩形アノテーションを付与した．なお，矩形アノテーションは，個々の人物 3D モデルを囲う矩形をスクリプトにより自動で作成した．図 4 に，番号 1 と番号 3 のカメラから撮影した実画像と，対応するデジタルツイン画像の例を提示する．

3.2.3 実 験 条 件

本実験では，カメラのパラメータと AI のハイパーパラメータをすべて共通の設定にしたときに，実画像とデジタルツイン画像に対する AI の精度が一致するかを確認した．実験で使用した AI モデルは，推論が高速なことから監視カメラ向け映像解析で一般的に用いられる，YOLOv3 [1] の事前学習済みモデル（入力サイズは 608 × 608）を使用した．そして，実画像とデジタルツイン画像のデータセットそれぞれに対し，[17] のツールを用いて推論を行ない，出力結果として物体を囲う矩形の座標とそのラベル，及び確信度の群を得た．その後，実環境と仮想環境の各カメラについて，一意の精度を算出し，精度が一致するかを確認した．例えば，実環境側であれば 1 カメラあたり 300 シーンに対して，デジタルツイン側であれば 1 カメラあたり 30 シーンに対して一意の精度を算出した．

精度は，物体検知における精度評価の指標である Average Precision を用い，AP@[.5:.95] を使用した．具体的には，まず，YOLOv3 の予測結果のラベルとして「person」がつき，かつその確信度が 0.5 以上の結果のみを残し，それ以外の結果をフィルタした．次に，推論された矩形の正誤判定に用いる IoU 閾値

カメラ番号	実画像 AP(%)	デジタルツイン画像 AP(%)	精度差（ポイント）
1	26.29	31.83	5.54
2	19.87	27.77	7.90
3	15.48	22.79	7.31
4	23.82	30.24	6.42
5	17.35	35.97	18.62
6	15.35	20.07	4.72
7	22.04	24.39	2.35
8	21.82	36.47	14.65
9	24.53	40.02	15.49
10	13.32	23.90	10.58
		精度差の平均値	9.36

表 1: ベースライン検証の結果

図 5: 日光による明るさの違いの例（番号 8 から撮影）

を 0.5 から 0.95 まで 0.05 刻みで増やしたリストを用意した．そして，フィルタ後の結果と正解データの矩形アノテーション，IoU のリストを用いて，各 IoU 閾値ごとに [18] により Average Precision を算出し，リスト内全ての IoU 閾値に対する AP の平均値を精度とした．最後に，カメラ番号ごとに実環境画像とデジタルツイン画像に対する YOLOv3 の精度の差を取り，差があるかを確認した．

3.2.4 実験結果

表 1 にカメラ番号ごとの精度を表にまとめた結果を示す．表 1 における実環境 AP は実画像に対する映像解析 AI の精度（AP@[.5:.95]）であり，デジタルツイン AP はデジタルツイン画像に対する精度である．精度差は，カメラ番号ごとに実画像 AP とデジタルツイン画像 AP の精度の差（絶対値）を取った結果であり，表中の精度差の平均値とは 10 カメラ分の精度差の平均値である．

表 1 より，すべてのカメラにおいて，デジタルツイン画像の方が実画像よりも精度が高いことが分かった．また，精度差の平均値は 9.36 ポイントとなり，デジタルツイン画像と実画像に対する映像解析 AI の精度は一致しなかった．特に，精度差はカメラ番号 5 の時が最も大きく，18.62 となった．このことから，デジタルツイン上で AI の精度が高くなるパラメータ（カメラ位置，角度）を実環境に適用しても，実環境上では十分な精度が出ない可能性が生じた．

3.3 EyeTuner の技術課題

EyeTuner を利用する上では，デジタルツイン画像に対する映像解析 AI の精度が，実画像に対する映像解析 AI の精度と一致することが前提であり，精度差が生じることは課題である．今回の実験で AI の精度が一致しなかった要因としては，デジタルツインと実環境における日光や照明の違いと，それに付随する色味の違いによる影響が考えられる．具体的には，図 5 に示すように，実環境では時間帯によって様々な日照と照明の状況が生じ，実画像データセットには「晴れており逆光が強い」，「曇りであり逆光が弱い」,「夜に撮影」した映像のような環境の明るさが異なる映像が含まれる．これに対しデジタルツイン画像のデータセットでは，空間の 3D モデルの光や影の状況が 3.2.2 節で使用したフォトグラメトリ用データの光や影の状況に固定される．このことから，実環境とデジタルツインの日光や照明と，それに付随する色味が異なることで精度差が生じた可能性がある．そこで本研究では，(1) 輝度を変換することでデジタルツイン画像の明るさを実環境に近づける輝度変換手法と，(2)GAN によって明るさを近づける GAN 変換手法を提案し，実験によりそれぞれの手法の効果を検証した．

4 提案手法

4.1 輝度変換手法

輝度変換手法では，デジタルツイン画像の輝度を実画像の輝度に近づけることで，精度差の低減に寄与するかを確認する．その手順として，(1) 実画像に対する輝度値の平均と分散を取得，(2) デジタルツイン画像に対する輝度値の平均と分散を取得，(3) デジタルツイン画像の輝度値を実画像の輝度値に近づくように変換し，変換後のデジタルツイン画像と実画像に対する映像解析 AI の精度を比較する．輝度変換に期待する効果は，変換後のデジタルツイン画像と実画像の精度差が，変換前の精度差と比べて小さくなることである．

まず，実画像の輝度値を取得し，その平均と分散を算出した．本実験では，3.2.1 節で使用した実画像データセットを，その明るさ明るさに応じて「晴れており逆光が強い」,「曇りであり逆光が弱い」,「夜に撮影」の 3 種類に目視で分類した．そして，分類した中から，晴れており逆光が強いフレームを 1 カメラあたり 20 枚，曇りであり逆光が弱いフレームを 1 カメラあたり 18 枚，夜に撮影したフレームを 1 カメラあたり 10 枚選択したサブデータセットを作成した．その後，各サブデータセットごとに実画像を輝度情報を持つ色空間に変換し，輝度情報を取得して輝度値の平均と分散を算出した．例えば「晴れており逆光が強い」フレームの輝度を算出する場合，計 200 フレーム（1 カメラあたり 20 フレーム× 10 台分）に対する輝度画像の全ての輝度値を用いて平均と分散を算出した．なお，輝度を取得するための色空間として YCbCr，HLS の 2 種類を用いた．

次に，デジタルツイン画像も同様に，3.2.2 節で得られた 30 枚のテスト用画像（× 10 台分）全てに対して輝度値を取得した．そして，画像 1 枚ごとに輝度値の平均と分散を取得し，その値が実画像から取得した輝度値の平均と分散になるように変換を行なった．計算式で表すと，以下の通りとなる．

$$V_{DT} = \frac{V_{DT} - \overline{V_{DT}}}{\sigma(V_{DT})} * \sigma(V_{Real-all}) + \overline{V_{Real-all}} \quad (1)$$

ただし，V_{DT} はデジタルツイン画像 1 枚における輝度値，$\overline{V_{DT}}$ と $\sigma(V_{DT})$ はそれぞれデジタルツイン画像 1 枚における輝度の平均値と分散，$V_{Real-all}$ は実環境の画像群を輝度値に変換したものであり，明るさの異なる 3 種類のサブデータセットに対して算出した結果である．最後に，輝度変換後のデジタルツイン画像と実画像を用いて 3.2.3 と同様に実験を行ない，映像解析 AI の精度を比較した．

4.2 GAN 変換

GAN とは，入力画像を目標の画像に近づけるための，深層学習による画像生成手法である．GAN は Generator と Discriminator から構成される．Generator は入力画像として，ランダムなピクセル値が載った画像を受け取り，画像を変換する生成器（変換器）である．Discriminator は，Generator が変換した後の画像を受け取り，目標とする画像に近いか否かを区別する識別器である．GAN の目標は，Generator と Discriminator の学習によって，Discriminator が区別出来ない画像を Generator が生成することである．また，近年では Generator の入力画像として写真などの通常の画像を入力とし，入力画像の色味や模様，明るさ等のスタイルを，目標とする画像に近づけるように GAN で変換を行なう CycleGAN [19] のような手法も登場した．本実験では，輝度変換と同様に GAN によってデジタルツインの明るさを実環境の明るさに近づけた場合に，精度差を低減可能かを検証した．

まず，GAN のモデルとして U-GAT-IT [20] を用い，デジタルツイン画像と実画像を用いて学習を行なった．学習に用いたデジタルツイン画像は，3.2.2 節で作成したテスト用画像 300 枚に加え，新たに人物 3D モデルを含むデジタルツインをランダムな位置と向きから撮影したデジタルツイン画像を含む，約 2,000 枚の画像を用意した．また，実画像は，3.2.1 節で用意した夜に撮影したフレームに加え，3.2.1 節で撮影した動画から夜に撮影したフレームを各カメラごとに新たに取得した約 10,000 枚の画像を用意した．そして，用意した画像全てを U-GAT-IT の入力サイズである 256 × 256 にリサイズし，バッチサイズは 1 で，デジタルツインの画像が CLIC の画像に近づくように 380,000 イテレーション学習を行なった．使用した GPU は Tesla T4 であり，[20] そのままのモデルでは GPU に乗らなかったため，モデル全体のチャネル数を約半分に削減し，Generator 内の ResnetBlock を 4 個に制限した．

次に，実画像に対する精度を算出した．使用した実画像は,「夜に撮影」したフレームに属する画像のうち，1 カメラあたり 18 シーンを選択して使用した．なお，この 18 シーンは U-GAT-IT の学習用画像にも含まれる．後述するデジタルツインのテスト用画像のサイズは U-GAT-IT の変換後の画像サイズである 256 × 256 のため，実画像と正解アノテーションのサイズ（スケール）も同様に 256 × 256 にリサイズした．使用したデータ以外の条件は，3.2.3 節と同じである．

最後に，デジタルツイン画像に対する精度を算出した．使用したデジタルツイン画像は，3.2.2 節の画像ではなく，新たに実画像（18 シーン）とほぼ同じ位置に人物の 3D モデルを手動で配置したシーンを用意して撮影したものを使用した．そして，学習済みの U-GAT-IT の Generator を用いて画像を変換した．ここで，本実験では Generator による変換を 2 パターン行ない，計 2 種類のテスト用データセットを用意した．1 つは，人物の 3D モデルが映るデジタルツイン画像をそのまま Generator で変換したデータセットである．この方法は Generator の変換により，背景だけでなく人物 3D モデルの色味や明るさを実画像に近づけられる可能性がある一方で，PTGAN [21] で述べられているように人物 3D モデルの形状が破綻し，推論に失敗する可能性がある．そこで，もう 1 つの方法として，デジタルツインから人物の 3D モデルを取り除いた空間 3D モデルのみを撮影し，その映像を Generator で変換して，後から人物の 3D モデルを変換後の背景画像に合成したテスト用データセットを作成した．この手法で作成した画像は，人物の形状を保ちながら，GAN による変換のメリットも受けることが可能であると考えられる．なお，U-GAT-IT による変換後の画像サイズは 256 × 256 であるため，アノテーションも 256 × 256 サイズにリサイズした．

5 実験結果

5.1 輝度変換手法の結果

図 2 に，輝度変換後の画像に対する YOLOv3 の精度を提示する．図中の実画像 AP，変換なし AP はそれぞれ 3.2.4 節で述べた実画像と変換前のデジタルツイン画像に対する精度である．逆光が強い，逆光が弱い，夜に撮影は，それぞれデジタルツインの画像を実画像のサブデータセット（「晴れており逆光が強い」,「曇りであり逆光が弱い」,「夜に撮影」）の輝度に変換した後の画像に対し，精度を算出した結果である．精度差の平均値は，3.2.4 節と同様に，カメラ番号ごとに，実画像 AP と精度の差（絶対値）をとり平均した結果である．

変換なしと変換後（逆光が強い，逆光が弱い，夜に撮影）における精度差の平均値を比較すると，変換後の方が平均 1.36 ポイント小さくなった．また，HLS と YCbCr における精度差の平均値を比較すると，逆光が強い，逆光が弱い，夜に撮影のどのパターンであっても HLS で変換した方が精度差が小さくなり，HLS の方が YCbCr よりも平均 1.69 ポイント小さくなった．さらに，逆光が強い，逆光が弱い，夜に撮影の 3 パターンについて比較すると，デジタルツインの画像を逆光が強いパターンに変換した場合が最も精度差が低減しており，HLS で 6.64 ポイント，YCbCr で 8.27 ポイントの差となった．

5.2 GAN 変換の結果

図 6 に GAN による「変換なし」,「背景のみ変換」,「背景と人物 3D モデルを変換」した画像を示す.「変換なし」は，実画像とデジタルツイン画像それぞれを 256 × 256 サイズにリサイズし，YOLOv3 の精度算出を行なった場合の結果である．「背景のみ変換」は，デジタルツインから人物の 3D モデルを取

カメラ番号	実画像 AP(%)	変換なし AP(%)	逆光が強い AP(%)		逆光が弱い AP(%)		夜に撮影 AP(%)	
			HLS	YCbCr	HLS	YCbCr	HLS	YCbCr
1	26.29	31.83	31.16	32.32	31.89	32.8	31.71	32.59
2	19.87	27.77	24.75	25.94	25.61	27.33	25.27	27.25
3	15.48	22.79	19.74	23.06	21.21	23.01	20.91	23.28
4	23.82	30.24	27.33	28.09	28.02	29.24	27.7	29.46
5	17.35	35.97	29.35	31.77	30.63	33.05	30.98	33.14
6	15.35	20.07	15.7	17.58	17.27	19.46	17.51	19.65
7	22.04	24.39	24.25	26	24.15	25.39	24.43	25.59
8	21.82	36.47	34.3	35.16	34.36	36.05	34.15	36.29
9	24.53	40.02	37.02	38.26	37.72	39.09	37.04	39.55
10	13.32	23.90	22.64	24.42	23.49	25.28	24.06	25.11
	精度差の平均値	9.36	**6.64**	8.27	7.45	9.08	7.39	9.2

表 2: 輝度変換手法の結果

カメラ番号	実画像（リサイズ済）AP(%)	変換なし（リサイズ済）AP(%)	背景のみ変換 AP(%)	背景と人物3Dモデルを変換 AP(%)
1	33.04	58.71	44.73	5.07
2	44.42	41.5	48.13	0
3	30.4	18.66	15.81	0
4	22.11	35.48	35.74	0
5	32.12	34.46	19.3	0.65
6	15.15	32.56	24	0.58
7	22.5	36.64	23.29	0
8	29.08	43.82	33.36	0
9	27.44	33.28	28.76	2.38
10	20.96	30.46	22.25	2.86
	精度差の平均値	11.767	**7.297**	26.568

表 3: GAN 変換手法の結果

図 6: GAN 変換手法の結果例

り除き，背景のみを撮影した後に U-GAT-IT で変換を行ない，後から人物の 3D モデルの映像を合成した画像に対して精度を算出した結果である．「背景と人物 3D モデルを変換」は，人物 3D モデルを含むデジタルツイン画像をそのまま U-GAT-IT で変換して精度を算出した結果である．

表 3 に実験結果を示す．精度差の平均値は変換なしの場合 11.77 ポイント，背景のみ変換した場合で 7.30 ポイント，背景と人物の 3D モデルを変換した場合で 26.57 ポイントであった．背景のみ変換した場合は変換なしよりも精度差が 4.47 ポイント低減したが，背景と人物の 3D モデルを変換した場合には変換なしと比べて精度差の平均値が増加した．

6 考　　察

輝度変換，GAN 変換の結果から，仮説の通りデジタルツイン画像の明るさを実画像に近づけることで精度差を低減することが可能であると考えられる．また，輝度変換手法において，逆光ありの輝度値に合わせて変換を行なった場合が最も精度差が低減された．これは，3. 2. 1 節で作成した実画像のデータセットについて，晴れており逆光が強いフレームの枚数が最も多く，実画像に対する精度は逆光が多いフレームに依存した可能性があり，その結果逆光が強いフレームの輝度にデジタルツイン画像の輝度を近づけることで精度が近づいたことが考えられる．

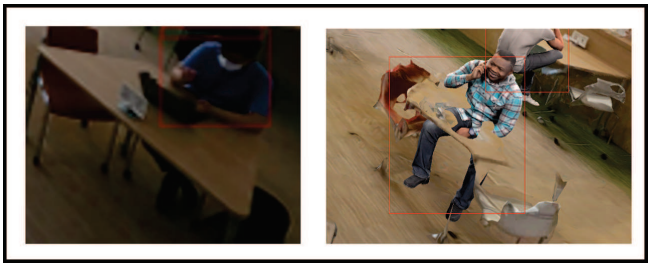
GAN 変換の結果から，背景のみ撮影した映像を変換した場合に精度差を低減する効果が見られた．一方で，背景と人物の 3D モデルをまとめて撮影した映像を変換した場合の精度差は，変換なしの場合よりも広がった．これは図 6 の点線より下側に示すように，GAN により人物 3D モデルの形状が破綻して YOLOv3 が人物として検知出来なくなったことが要因として考えられる．そのため，PTGAN などの手法のように，人間の輪郭を保つような機構を GAN に取り入れ，人物の形状を保ったままその明るさを含めて実環境に近づけることで，さらに精度差を低減できる可能性がある．

7 議　　論

今回の検証で精度差が完全には埋まらなかった理由として，空間モデルの再構成の精度が荒いことが挙げられる．これは例えば図 7a のように，実画像側は机によってオクルージョンが生じて上半身しか検知できないが，デジタルツイン画像では机の再構築が荒く人間の全身を検知できてしまうことや，図 7b のようにカメラの台座がデジタルツイン側では再現できておらず，オクルージョンが実画像よりも減ったことなどが挙げられる．そのため，今後は欠損の少ないデジタルツインを構築することで精度差が低減されるかを検討する必要がある．欠損の少ないデジタルツインの構築方法としては，空間を今回のように一括して広い空間をスキャンするのではなく，細かい領域に分割して丁寧にスキャンし，部分部分でフォトグラメトリを実行して空間モデルを構築し，最後に空間モデルを連結してデジタルツインを構築することや，NeRF [14] などの，深層学習技術を用いる方法を検討する．

また，最終的な EyeTuner の構想では，デジタルツインの中で自動的にカメラの設置パラメータと AI のハイパーパラメータを探索する必要があるが，デジタルツインの環境が大きくなるほど探索に要する計算コストや時間は増加することが想定される．そのため，まずはグリッドサーチなどの基本的なアルゴリズムを用いて探索を行ない，計算コストや時間が課題となるかを明確化する必要がある．

さらに，今回の検証では物体検知というタスクの中で，限られた映像解析 AI モデルでしか検証を行なっていないが，実用



(a) 机の欠損（番号 6）

(b) カメラの台座の欠損（番号 10）

図 7: メッシュの欠損と物体検知に対する影響の例

上は同じ物体検知というタスクであっても様々な AI モデルが存在し，かつ AI のタスクについても姿勢推定や人物照合などの様々なものが存在する．そのため，異なる AI モデル，異なるタスクについても同様に精度差が生じるかを検証する必要がある．

8 おわりに

映像解析 AI の精度低下を防ぐために行われる，カメラの設置パラメータと AI のハイパーパラメータの調整作業のコストと作業負担を削減するため，デジタルツインを活用して調整作業を自動化するシステムである EyeTuner を提案した．また，デジタルツイン画像に対する映像解析 AI の精度が実画像に対する精度と一致するかは明らかではなかったため，人物検知タスクを対象に精度が一致するかを検証したところ，精度のギャップが生じた．そこで，輝度変換や GAN による変換を用いてデジタルツインの画像の明るさを実環境の画像の明るさに近づけることで，精度差を 6.64 ポイントまで抑えることが可能なことを確認した．今後の課題として，さらに精度差を低減するため欠損の少ないデジタルツインを構築する手法の検討に加え，デジタルツインの中で効率的にカメラの設置パラメータと映像解析 AI のハイパーパラメータを探索する手法の検討，及び異なる映像解析 AI モデルを用いた場合の検証が必要である．

文　献

[1] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement, 2018.

[2] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 43, pp. 172–186, 2021.

[3] Alexander Hermans, Lucas Beyer, and B. Leibe. In defense of the triplet loss for person re-identification, 2017.

[4] MS Sumi Suresh, Athi Narayanan, and Vivek Menon. Maximizing camera coverage in multicamera surveillance networks. *IEEE Sensors Journal*, Vol. 20, No. 17, pp. 10170–10178, 2020.

[5] Xiaojian Zhu, Mengchu Zhou, and Abdullah Abusorrah. Optimizing node deployment in rechargeable camera sensor networks for full-view coverage. *IEEE Internet of Things Journal*, Vol. 9, No. 13, pp. 11396–11407, 2021.

[6] Kun Shi, Shuxian Liu, Chao Li, Haoyu Liu, Shibo He, Qi Zhang, and Jiming Chen. Towards optimal deployment for full-view point coverage in camera sensor networks, 2022.

[7] Niccoló Bisagno and Cristian Iacovlev. Camera network optimization: maximize coverage in a 3d virtual environment. In *Proceedings of the 13th International Conference on Distributed Smart Cameras*, pp. 1–2, 2019.

[8] Sebastian Haag and Reiner Anderl. Digital twin – proof of concept. *Manufacturing letters*, Vol. 15, pp. 64–66, 2018.

[9] Barbara Rita Barricelli, Elena Casiraghi, and Daniela Fogli. A survey on digital twin: Definitions, characteristics, applications, and design implications. *IEEE access*, Vol. 7, pp. 167653–167671, 2019.

[10] 野村淳一, 三輪冠奈. データ駆動型社会におけるデジタルツインに関する一考察. 情報経営, pp. 45–48. 日本情報経営学会, 2021.

[11] Tadele Belay Tuli, Linus Kohl, Sisay Adugna Chala, Martin Manns, and Fazel Ansari. Knowledge-based digital twin for predicting interactions in human-robot collaboration. In *2021 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, pp. 1–8. IEEE, 2021.

[12] Yuto Fukushima, Yusuke Asai, Shunsuke Aoki, Takuro Yonezawa, and Nobuo Kawaguchi. Digimobot: Digital twin for human-robot collaboration in indoor environments. In *2021 IEEE Intelligent Vehicles Symposium (IV)*, pp. 55–62. IEEE, 2021.

[13] Raul Mur-Artal and Juan D Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE transactions on robotics*, Vol. 33, No. 5, pp. 1255–1262, 2017.

[14] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, Vol. 65, No. 1, pp. 99–106, 2021.

[15] 3D Flow. 写真計測用ソフトウェア 3df ゼファー (3df zephyr).

[16] RENDERPEOPLE. World ' s largest library of scanned 3d people.

[17] Kai Chen, Jiaqi Wang, Jiangmiao Pang, et al. Mmdetection: Open mmlab detection toolbox and benchmark, 2019.

[18] Rafael Padilla, Wesley L Passos, Thadeu LB Dias, Sergio L Netto, and Eduardo AB Da Silva. A comparative analysis of object detection metrics with a companion open-source toolkit. *Electronics*, Vol. 10, No. 3, p. 279, 2021.

[19] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 2223–2232, 2017.

[20] Junho Kim, Minjae Kim, Hyeonwoo Kang, and Kwang Hee Lee. U-gat-it: Unsupervised generative attentional networks with adaptive layer-instance normalization for image-to-image translation. In *International Conference on Learning Representations*, 2020.

[21] Longhui Wei, Shiliang Zhang, Wen Gao, and Qi Tian. Person transfer gan to bridge domain gap for person re-identification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 79–88, 2018.