

ВІРТУАЛЬНА ПРИМІРКА 3D МОДЕЛІ ОДЯГУ НА ВІДЕО ЗАСОБАМИ МАШИННОГО НАВЧАННЯ

Коріонов Дмитро, Юрій Гуртовий

Науковий керівник: кандидат ф.-м. наук, доцент Гуртовий Ю.В.

Центральноукраїнський державний педагогічний університет імені

Володимира Винниченка, м. Кропивницький, Україна

Стаття присвячена розгляду сучасних підходів до роботи з 3D розпізнавання і моделювання пози людини та об'єктів, що її оточують, з якими вона може взаємодіяти у віртуальній чи доповненій реальності. Оглянуто практичний приклад організації процесу розробки програмного рішення для створення "віртуальної примірки одягу", описано принципи роботи методології рішення, що базується на розпізнаванні тривимірних часово-просторових об'єктів, зокрема людей, з використанням технологій машинного навчання

Ключові слова: Машинне навчання, 3d розпізнавання, захоплення руху, регресія

VIRTUAL DRESSING OF CLOTHES IN VIDEO VIA MACHINE LEARNING

Korionov Dmytro, Yurii Hurtovyi

Supervisor: Ph.D., Senior Lecturer Hurtovyi Yu.V.

Volodymyr Vynnychenko Central Ukrainian State Pedagogical University,

Kropyvnytskyi, Ukraine

Article is devoted to study of modern approaches to tasks in 3d recognition and modelling of human pose and surrounding objects, that people can interact with in virtual or augmented reality. This article briefly reviews practical experience of developing software solution for creating a "virtual dressing room". The methodology of solution is described, which is based on motion capture of people by means of machine learning technologies.

Keywords: Machine learning, 3d recognition, motion capture, regression

Постановка проблеми. Розвиток технологій машинного навчання та світове пандемічне становище створюють попит на ширшу адаптацію та застосування комп'ютерного розпізнавання і технологій машинного навчання для вжитку у повсякденному споживанні у вигляді віртуальної або зміненої, доповненої реальності. З розвитком робототехніки та машинного зору, технологій віртуальної та доповненої реальності з'являються і нові задачі в яких необхідно класифікувати тривимірні об'єкти, відновлювати відсутню інформацію про такі об'єкти або генерувати нові, для вирішення цих задача все частіше вдаються до методів машинного навчання. Це можна пояснити тим, що для більшості класичних алгоритмів машинного навчання необхідні

аналітичні функції для маніпулювання над даними. Якщо розглядати класичні алгоритми машинного, то зазвичай у більшості випадків для роботи з ними використовують дані з регулярною структурою: текст, зображення, таблиці, сигнали але у разі, якщо представлені дані, кодують тривимірні структури, то говорять про область машинного навчання, яка називається 3D ML (three dimensional data machine learning problems) або часто зустрічається термін Geometrical deep learning, якщо мова йдеться про застосування глибоких архітектур.

Регресія пози людини в 3D-просторі засобами Frankmocap

Якщо проаналізувати процес віртуальної примірки, то головною задачею, яку необхідно вирішити є розпізнавання та моделювання пози людини в 3D просторі. Передбачення пози людини і є нетривіальною задачею з декількох причин.

По-перше, щоб регресувати позу людини спочатку необхідно детектувати саму людину на відео. Задача детекції є не нова, тому на сьогоднішній день існує багато інструментів для її рішення, загалом це вже навчені моделі ML.

По-друге, по знайденому зображенню людини необхідно регресувати її позу, щоб далі створити відповідну 3D-модель. Така задача називається 2D-to-3D та передбачає роботу з так званими неевклідовими даними. Отже також проблемою, яку необхідно вирішувати, є формат представлення даних. У 3D об'єкти часто представлені вершинами і трикутниками тобто сітками, що кодує їх 3D-форму. Чим більш деталізований об'єкт, тим більше вершин йому потрібно. На сьогодні основний формат представлення людського тіла є модель SMPL(рис.1) - це статистичні модель, яка кодує образ людини за допомогою двох параметрів[3]:

- вектор із 10 скалярних значень, що відповідає за статуру людини;
- вектор із 24x3 скалярними значеннями, що відповідають за позу людини та повороти суглобів відносно її параметрів. Кожен

оберт кодується як довільний двовимірний вектор у представленні осьового кута повороту.

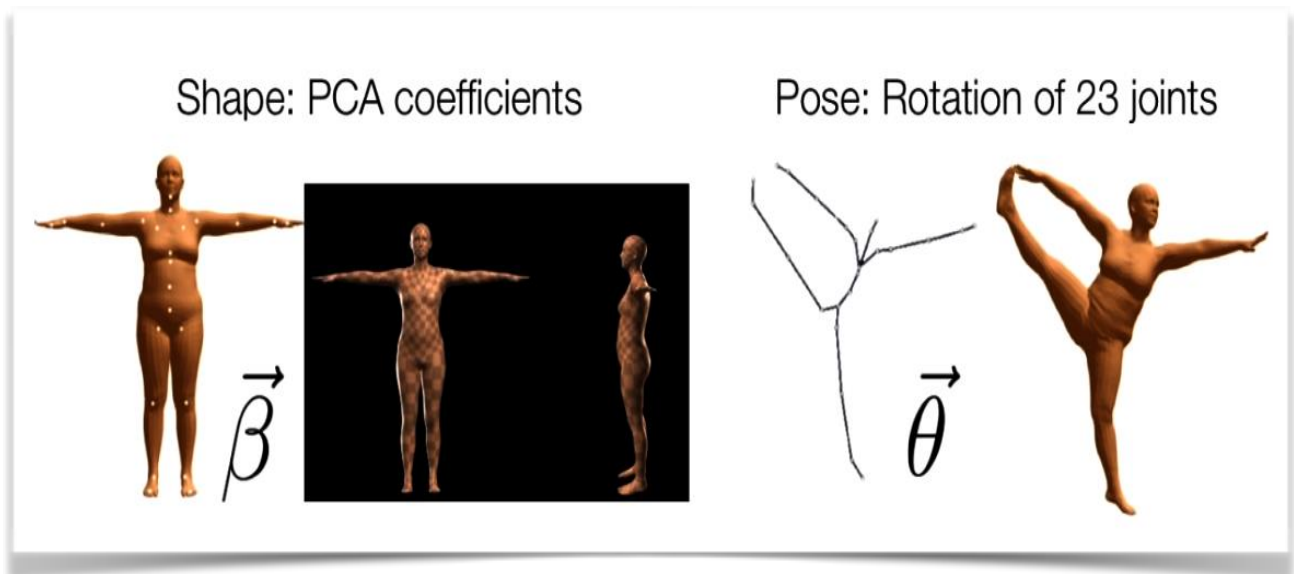


Рис 1. SMPL модель

Тренувати власні моделі для детекції, регресії, є надзвичайно важкий та довгий процес, який також потребує дуже багато ресурсів та обчислювальної потужності, тому для вирішення вище перерахованих задач можна скористатися бібліотеками FrankMocap від Meta (Facebook AI). Дана бібліотека включає в себе вже навчені моделі для регресії, детекції. Також бібліотека потребує наявності графічного процесору тобто відеокарти, тому для написання коду та роботи з бібліотекою можна використати хмарні ресурси Googlecolab або Kaggle, які дають змогу користуватися графічним або тензорним процесором.

```

cur_frame +=1
if img_original_bgr is None or cur_frame > args.end_frame:
    break
print("-----")

if load_bbox:
    body_pose_list = None
else:
    body_pose_list, body_bbox_list = body_bbox_detector.detect_body_pose(
        img_original_bgr)
hand_bbox_list = [None, ] * len(body_bbox_list)

# save the obtained body & hand bbox to json file
if args.save_bbox_output:
    demo_utils.save_info_to_json(args, image_path, body_bbox_list, hand_bbox_list)

if len(body_bbox_list) < 1:
    print(f"No body detected: {image_path}")
    continue

```

Рис 2. Читання та детектування відео

В якості детектора FrankMocap має клас BodyPoseEstimator, що по суті є сервісом натренованої моделі MobileNet для детекції об'єктів на відео, в тому числі і людей. Модель приймає параметром зображення та результатом її є координати прямокутника з людиною.

Отже першим кроком, який необхідно зробити - це зчитати відео, далі покадрово його пропустити через модель детекції, та отримати масив з координатами знайдених людей на відео. Наведений код(рис.2) ілюструє цей процес.

```

with torch.no_grad():
    # model forward
    pred_rotmat, pred_betas, pred_camera = self.model_regressor(norm_img.to(self.device))

    #Convert rot_mat to aa since hands are always in aa
    # pred_aa = rotmat3x3_to_angle_axis(pred_rotmat)
    pred_aa = gu.rotation_matrix_to_angle_axis(pred_rotmat).cuda()
    pred_aa = pred_aa.reshape(pred_aa.shape[0], 72)
    smpl_output = self.smpl(
        betas=pred_betas,
        body_pose=pred_aa[:,3:],
        global_orient=pred_aa[:, :3],
        pose2rot=True)
    pred_vertices = smpl_output.vertices
    pred_joints_3d = smpl_output.joints

    pred_vertices = pred_vertices[0].cpu().numpy()

    pred_camera = pred_camera.cpu().numpy().ravel()
    camScale = pred_camera[0] # *1.15
    camTrans = pred_camera[1:]

    pred_output = dict()
    # Convert mesh to original image space (X,Y are aligned to image)
    # 1. SMPL -> 2D bbox
    # 2. 2D bbox -> original 2D image
    pred_vertices_bbox = convert_smpl_to_bbox(pred_vertices, camScale, camTrans)
    pred_vertices_img = convert_bbox_to_oriIm(
        pred_vertices_bbox, boxScale_o2n, bboxTopLeft, img_original.shape[1], img_original.shape[0])

    # Convert joint to original image space (X,Y are aligned to image)
    pred_joints_3d = pred_joints_3d[0].cpu().numpy() # (1,49,3)
    pred_joints_vis = pred_joints_3d[:, :3] # (49,3)
    pred_joints_vis_bbox = convert_smpl_to_bbox(pred_joints_vis, camScale, camTrans)
    pred_joints_vis_img = convert_bbox_to_oriIm(
        pred_joints_vis_bbox, boxScale_o2n, bboxTopLeft, img_original.shape[1], img_original.shape[0])

```

Рис 3. Код реалізації регресії людини

Безпосередньо другим кроком є сама регресія пози людини(рис.3). В FrankMocap - це відбувається за допомогою моделі HRM, яка вчилася реконструювати повну тривимірну сітку людського тіла з одного зображення RGB. Також дана модель використовує метод оптимізації SPIN, що дає більш насичене та корисне репрезентацію сітки, параметризованою формою та кутами тривимірного з'єднання. [4]

Сервісом для цієї моделі виступає клас BodyMocap, який на вхід приймає: оригінальні зображення та масив координатів прямокутників з людиною. Результатом регресії є передбачена фактура та матриця розтошування суглобів і їх обертів відносно батьківських суглобів, кожен суглоб репрезентується матрицею 3x3, тому їх необхідно конвертувати в

кути по осі, щоб отримати матрицю 24x3, яка є представленням пози в SMPL [1]. Далі з отриманих даних необхідно створити об'єкт SMPL моделі.

Поєднання моделей одягу та людини засобами MultiGarmentNetwork

Для накладання одягу на людини була використана бібліотека MultiGarmentNetwork, яка містить сховище 3D моделей одягу у форматі .obj, який легко можна конвертувати в SMPL, оскільки їхня підгонка можлива параметрами форми тілобудови та визначеної пози. Для того, щоб поєднати моделі необхідно розібрати третій крок формування моделі SMPL.

```
tgt_params = {'pose': np.array(smpl_tgt.pose.r), 'trans': np.array(smpl_tgt.trans.r), 'betas': np.array(smpl_tgt.betas.r), 'gender': 'neutral'}
smpl_tgt.pose[:] = 0
body_tgt = Mesh(smpl_tgt.r, smpl_tgt.f)

## Re-target
ret = retarget(garment, body_src, body_tgt)

## Re-pose
ret_posed = pose_garment(ret, vert_inds, tgt_params)
body_tgt_posed = pose_garment(body_tgt, range(len(body_tgt.v)), tgt_params)

## Remove intersections
ret_posed_interp = remove_interpenetration_fast(ret_posed, body_tgt_posed)
ret_posed_interp.vt = garment.vt
ret_posed_interp.ft = garment.ft
ret_posed_interp.set_texture_image(garment_tex)
```

Рис 4. Трансформація суглоба регресованої моделі

На цьому кроці вершини пози в стані спокою трансформуються за допомогою зваженої комбінації глобальних трансформацій суглоба (поворот + зміщення). Простіше кажучи, чим ближче до вершини суглоб, тим сильніше суглоб обертає (трансформує) вершину [3]. Тобто до сітки людини в стані спокою необхідно дати сітку людини в одязі та зважену комбінації глобальних трансформацій суглоба регресованої моделі. Потім на отриману модель необхідно накласти текстуру одягу, для цього можна скористатися функцією `set_texture_image`. Наведений на рис.4 код ілюструє нашу реалізацію цього процесу.

4. Візуалізація моделі засобами Pytorch3D

Останім Кроком залишається візуалізація отриманої моделі поверх відео. Для візуалізації моделей та роботи з зображеннями було використано бібліотеку Pytorch3D.

```
# visualization
res_img = visualizer.visualize(
    img_original_bgr,
    pred_mesh_list = pred_mesh_list,
    body_bbox_list = body_bbox_list)

# show result in the screen
if not args.no_display:
    res_img = res_img.astype(np.uint8)
    imshow(res_img)
```

Рис 5. Клас Visualizer

Дана бібліотека містить в собі моделі освітлення, затінення (шейдери), аналізу кольорів, текстур та ефектів світла, саме клас Visualizer(рис. 5), який може рендерити моделі поверх зображення. Для цього йому необхідно передати оригінальне зображення, координати прямокутників з людиною, та самі моделі людини в одязі.

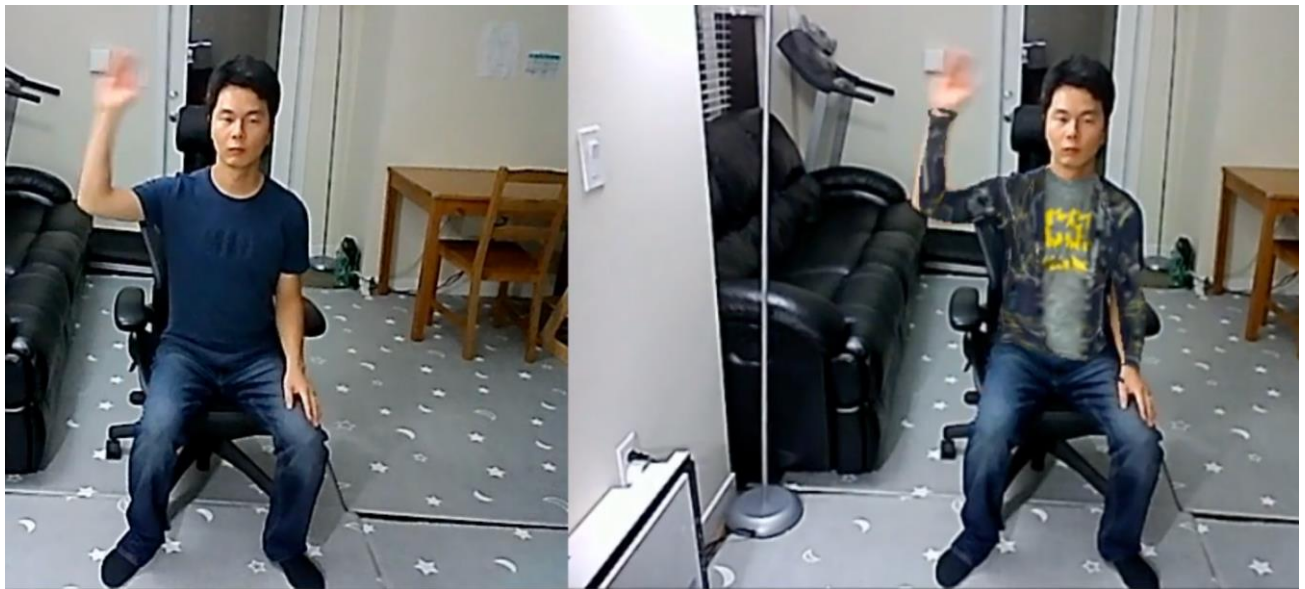


Рис.6 Віртуальна примірка одягу

В результаті було створено скрипт мовою python, який дає змогу віртуально примірювати одяг шляхом розширення функціоналу open-source проекту Facebook AI Research FrankMocap в поєднанні з проектом 3D моделювання одягу MultiGarmentNetwork(рис.6).

Висновки

Отже, у даній статті була розглянута SMPL модель, яка є стандартом в задачах регресії та моделювання пози людини в просторі, досліджено етапи побудови моделі та регресії пози людини. Також було виконано технічну

роботу результатом якої є програма, яка розширила базовий репозиторій *francosar*, додавши туди функції трансформації 3d моделей одягу, їхньої підгонки під розпізнаний та змодельований меш людини з реального відеозображення. Віртуальний одяг накладається на відео людини, адаптуючись до її тілобудови та рухів. Подальша робота в цьому напрямку має базуватися на розширенні розуміння теоретичного підґрунтя і математичного апарату технологій машинного навчання і статистики. Для кращого накладання одягу на відтворену модель людини доцільно вводити елементи фізичного моделювання - аби одяг лягав на плечі і ключові анатомічні поверхні та вигини, мав ефекти властиві матеріалам одягу. Також для поліпшення методів оптимізації для змоги накладати одяг у реальному часі.

Список літератури

1. Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J Black. Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image. In European Conference on Computer Vision, 2016
2. Rong, Yu and Shiratori, Takaaki and Joo, Hanbyul. A Monocular 3D Whole-Body Pose Estimation System via Regression and Integration. In IEEE International Conference on Computer Vision Workshops, 2021
3. Matthew Loper, Naureen Mahmood Javier Romero, Gerard Pons-Moll, Michael J. Black. SMPL: A Skinned Multi-Person Linear Model
4. Angjoo Kanazawa, Michael J. Black, David W. Jacobs, Jitendra Malik. End-to-end Recovery of Human Shape and Pose CVPR 2018.