AN INTELLIGENT MOBILE APPLICATION TO ANALYZE DANCE MOVEMENTS AND PROVIDE FEEDBACKS USING MEDIAPIPE AND OPENCV

Hongfan Zhu[1], Theodore Tran[2]

[1]YK Pao School, 1251 West Wuding Road, Changning District, Shanghai, China
200042 [2]Computer Science Department, California State Polytechnic University, Pomona, CA 91768

ABSTRACT

Motion Mentor is a mobile application designed to address a challenge faced by beginner dancers in improving their dancing technique-the development of improper movements and habits when practicing without a teacher's guidance. Therefore, Motion Mentor offers real-time posture correction and personalized feedback [5]. This method involves using the Mediapipen pose-detection AI model for real time posture detection, combined with advanced algorithms for accurate dance analysis, and Firebase for the storage of data and uploaded videos [6]. Users can access educational content, record their dance performances for feedback, and review their progress. During the experimentation, our system was applied to scenarios involving rapid dance movements to test the accuracy of pose estimation, comparison between the estimated and the actual real-time distance and speed estimation [7]. These scenarios suggested the limitations of our application in different dynamic and lighting conditions, providing insights into areas for improvement. Overall, this solution enhances accessibility and conveniences for all dancers in improving dance technique, offering real-time feedback and educational materials.

KEYWORDS

Dance Movement, Pose Detection, Video Processing

1. INTRODUCTION

Embarking on the intricate journey of dance often involves grappling with the nuances of movement and, unfortunately, the inadvertent cultivation of undesirable habits. This challenge is particularly pronounced for novice dancers navigating the realms of practice without the immediate guidance of a seasoned instructor. For example, even the greatest and the most renowned dancers, such as Lauren Lovette, Karida Griffth, and Melanie Moore suffered with various bad habits which required double the effort to reverse [1]. In such solitary endeavors, the absence of real-time correction leaves room for the inadvertent solidification of inaccuracies, hindering the seamless development of one's dance technique.

In response to this pervasive challenge, we proudly unveil a groundbreaking solution—a cutting-edge mobile application meticulously crafted to deliver real-time correction of dancers' postures. This visionary tool stands as a beacon for self-improvement, providing dancers with an unprecedented avenue for immediate, precision-guided refinement [15]. By eradicating the waiting game associated with traditional instruction, our application seamlessly integrates

David C. Wyld et al. (Eds): AIBD, MLSC, ACSTY, NATP, CCCIoT, SVC, SOFE, ITCSS -2024
pp. 75-85, 2024. CS & IT - CSCP 2024 DOI: 10.5121/csit.2024.140406

corrective insights into the dancer's practice, eliminating the risk of perpetuating detrimental habits.

Our commitment extends beyond mere correction; our application curates a rich repository of insights, tips, and instructional videos sourced from renowned experts and institutions. This curated content serves as a wellspring of inspiration, empowering dancers to not only rectify mistakes but also to elevate their technical prowess through supplementary knowledge.

In essence, our mobile application emerges as a transformative force in the dance arena, shattering the constraints of conventional practice. It is a testament to the marriage of technological innovation and artistic refinement—a tool that not only corrects but elevates, ensuring that dancers at every level unlock their fullest potential. Join us in redefining the dance landscape, where precision meets convenience, and self-improvement knows no bounds.
All three methodologies attempted to detect dance movements.

The 1st and 2nd proposed methods utilized various sensors on the dancer’s body to capture movement data, which are subsequently applied to a neural network model to recognize motion patterns, such as dance figures and stops. However, some potential limitations are sensor drift and calibration issues affecting the accuracy of the motion capture system. The calibration process and the requirement for many sensors makes it hardly accessible to most dancers. Additionally, they may ignore outside factors such as different dance styles, physical fatigue, etc. Although the sensor is more accurate since it analyzes based on stable and subsistent coordinates, our app improves on accessibility to a wider variety of dancers and the diversity of dancing styles, rendering our project into a practical tool for dancers world-wide.

The 3rd solution, blending video processing and classification recognition, heavily relies on multi-feature fusion for precise dance action recognition. However, it generates high-dimensional, redundant feature vectors, intensifying computational demands. Additionally, the method tends to assign similar weights to less and more impactful features, affecting accuracy. Our project targets dance action recognition, utilizing multi-feature fusion while overcoming challenges of high dimensionality and key frame extraction. Real-time posture correction and personalized feedback provide a comprehensive solution for personalized dance techniques, surpassing existing limitations.

Motion Mentor provide real-time posture correction and personalized feedback to effectively address issues and stop dancers from developing poor movements/habits in their dance(s). As opposed to trying to learn a dance by oneself, having a tool that provides instant and personalized advice will greatly help dancers learn and improve on their dance abilities more accurately and effectively. Because of its accessibility and convenience (it can be used on mobile devices anytime and anywhere), there is no need to wait for teachers or attend physical classes regularly. The app also offers a thorough learning experience that combines real-time feedback with a collection of educational materials, making it a very useful and adaptable tool for dancers of all skill levels. In essence, Motion Mentor is committed to adapting to each user’s unique abilities and situations, enriching them with the tools and feedback needed for growth and proficiency in the art of dancing.

In Experiment A, we were testing the accuracy of pose estimation, particularly during rapid dance movements. The experiment was set up by recording footage of dynamic dance movements to assess the program’s performance. Control data was sourced from videos of slower dance movements to establish a baseline. The most significant finding was the inconsistency in angle measurements during rapid movements, with a mean angle of 68 degrees and a range from 45 to 85 degrees. This highlighted some challenges in accurately tracking landmarks and calculating angles during fast movements. The dynamic nature of the movements can introduce errors and inaccuracies in the tracking algorithms. Improvements in tracking algorithms and frame rate may address these issues.

In Experiment 2, we aimed to test the accuracy and functionality of real-time distance and speed estimation. The setup involved recording video frames with measured object widths as references. The time taken for each frame to process was recorded for speed calculation. Actual distances and speeds were collected as baselines for comparison. The most significant finding was the impact of lighting conditions on speed values, which ranged from 0.2 to 1.0 meters per second. Object recognition accuracy, affected by lighting, greatly influenced distance and speed estimation. The results highlighted the need for improved object recognition algorithms to ensure accurate distance and speed estimation under diverse lighting conditions.

2. CHALLENGES

In order to build the project, a few challenges have been identified as follows.

2.1. Back End

Problem: Interpreting the output data from the Mediapipe AI model to calculate metrics poses challenges to the accuracy and relevance of dance performance analysis [8].

Solution: To address this challenge, I could develop custom algorithms or post-processing techniques specifically tailored to the output data provided by the Mediapipe model. This would involve converting key points or landmarks into 3D coordinates to calculate distances accurately, tracking key points over time to derive speed, and analyzing the relationships between specific points to determine hip angle and shoulder slope [9]. Additionally, providing clear and user-friendly visualizations or reports based on these calculations would help dancers better understand and utilize the data to improve their performance.

2.2. Tip Page

Problem: Updating the tip page directly from the source code, which necessitates a complete app upgrade whenever changes are required (a time-consuming process and not user-friendly).

Solution: To address this challenge, I would likely implement the ability to edit and update the tip page from Firebase's database rather than inserting it manually in a text file within the source code [10]. This solution would significantly improve the convenience of making updates, allowing for real-time content changes without the need for app upgrades/updates.

2.3. UI

Problem: The UI design lacks a cohesive aesthetic that should be consistent, fluid and intuitive to the user [14]. Various sections or pages feature different colors, fonts, and shapes, leading to a visually disjointed user experience.

Solution: To address the issue, I could establish a consistent theme color for each page or section, such as using overall orange for one section, green for another, blue for a third, and pink for the fourth, ensuring that these colors complement each other. Additionally, adopting a uniform set of fonts and typography styles throughout the app would help provide visual uniformity and clarity (ex. Headers would be bolded and bigger than subtext).

3. SOLUTION

The main structure of our app, Motion Mentor, is composed of three major components to provide an effective experience for ballroom dancers: the Learning Pages section, the "Just Dance" section, and the History section. Users start with the Loading Page, which initializes the app. They are then directed to the Home Page, where they can access the navigation bar. From the navigation bar, users can explore various sections of the app, including Learning Pages, Just Dance, and History.

In the Learning Pages section of the app, users can access tips and instructional videos to improve their dancing skills. These videos and tips range from a variety of different reputable sources, allowing users more versatility and options to choose from.

In the Just Dance section, users can record their dance performances. The app employs advanced algorithms and AI recognition techniques to analyze the recorded videos and provide feedback [11]. Users can also access the processed videos, complete with pose detection landmarks and other relevant data which is stored in the History section.

In the History section, users can review their processed videos, track their progress, and compare past performances. This allows them to go back and analyze even the oldest of their uploads and see their improvement/regression and give users the opportunity to take these experiences and hopefully grow/learn from them.

For this project, we utilized a combination of technologies: Media Pipe, Python, Firebase, and Dart. Google's MediaPipe was utilized for real-time pose detection and tracking in dance videos [13]. Python was chosen as the backend programming language to implement algorithms responsible for processing data from the pose detection model, offering users valuable feedback on their dance performances. Firebase served as the reliable cloud storage solution for securely storing processed videos, tutorial content, and potentially other user data. Finally, Dart was used to create the frontend of the app, facilitating the development of the user interface and navigation system as well as connect to the backend of the app and its subsequent functionalities.

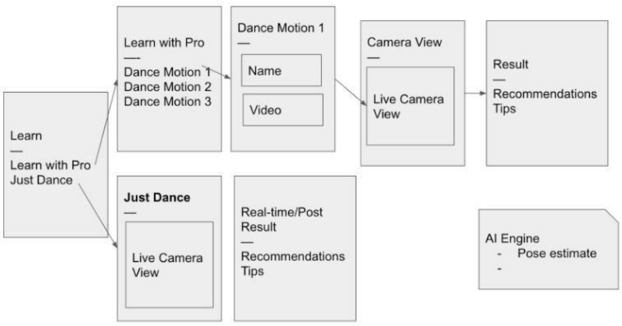

Figure 1. Overview of the solution

Within the Learning Pages section, dancers can access resources to enhance their skills. This section is further divided into two essential components: the Tip Page, which offers valuable insights on ballroom dancing techniques, and the Video Page, where users can access a library of instructional videos.

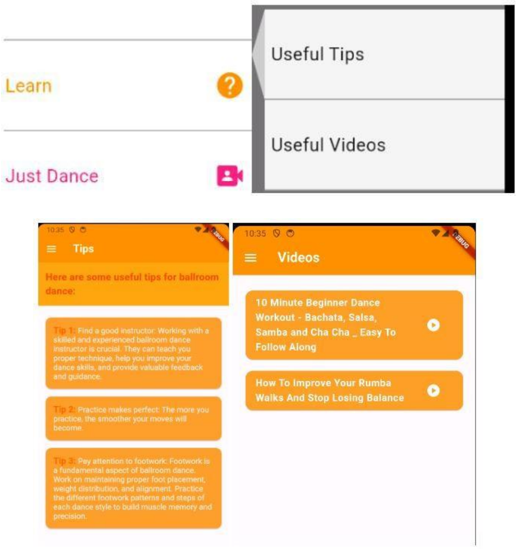

Figure 2. Learning pages section

Figure 3. Screenshot of code 1

The code selected from the “tip_page” creates stylized cards that display useful tips. It runs when the user enters the Tip Page. The “Card” widget is used to define a visually distinct container for the tips, used within the context of TipPageState. The TipPageState class consists of numerous strings of tips that can be easily edited.

The "Just Dance" function within the Motion Mentor app offers a unique and interactive experience for users. It allows users to record videos of themselves while dancing and then utilizes advanced algorithms and AI recognition techniques to analyze those videos. It then returns the processed videos to the user. These processed videos include valuable data such as pose detection landmarks, dance performance, including posture, movement, balance, and other various information.

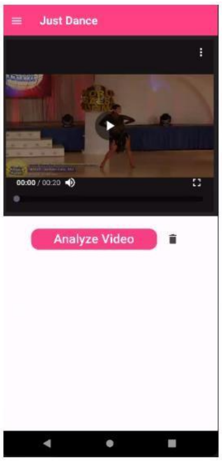

Figure 4. Screenshot of analysis page

```
void uploadFileToServer() async {
  // This url is for the local server. Change later to the public url.
  var url = 'http://10.0.2.2:5000/'; // Local host

  Map<String, String> headers = {
    "Connection": "Keep-Alive",
    "Keep-Alive": "timeeout=5, max=1000"
  };

  http.MultipartRequest request =
      http.MultipartRequest('POST', Uri.parse('$url/analyze_dance'));
  request.headers.addAll(headers);
  request.files.add(
    await http.MultipartFile.fromPath(
      'video',
      widget.video1File.path,
      contentType: MediaType('application', 'MOV'),
    ),
  );

  request.send().then((r) async {
    print(r.statusCode);

    if (r.statusCode == 200) {
      var result = json.decode(await r.stream.transform(utf8.decoder).join());
      _saveVideo(result);
      setState(() {
        analyzedVideoUrl = result;
        _setVideoController(analyzedVideoUrl);
        isLoading = false;
      });
    }
  });
}
```

Figure 5. Screenshot of code 2

The function in the result.dart file is responsible for uploading video files to the server. Initially, the function establishes the URL to the server and the HTTP headers. The MultipartRequest is utilized to generate a POST request to the server, allowing us to set a multi-part request suitable for uploading files.

Subsequently, the function requests the headers and adds the imported video files using MultipartFile.fromPath. widget.videoFile.path represents the path of the video that will be analyzed in the JustDance section.

Following this, request.send().then((r) async) sends the request to the server. A HTTP response callback (represented by r in the brackets) is then returned, which includes the statusCode. If the statusCode is 200, the request is successful. In this case, the result is saved to an ArrayList to store the information, such as the date and time. The information is later decoded and displayed to the user. Finally, isLoading is set to false, indicating that the function has completed its process.

In the "History" function of the app, users can conveniently access their processed videos at a later time, which consists of the specific time when it was generated. Dancers can review their previous attempts to consolidate their learning.

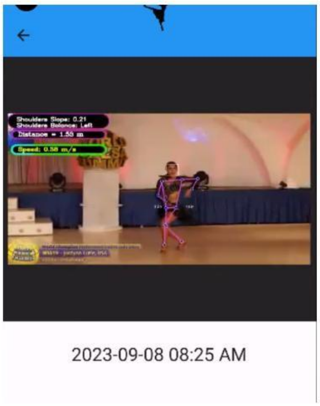

Figure 6. Screenshot of video

```
_loadHistory() async {
  SharedPreferences prefs = await SharedPreferences.getInstance();
  List dates = [];
  print(prefs.containsKey('date'));
  if (prefs.containsKey('date')) {
    dates = prefs.getStringList('date')!;
    for (var date in dates) {
      _history[date] = json.decode(prefs.getString(date)!);
    }

    print(_history.keys);
    setState(() {});
  }
}
```

Figure 7. Screenshot of code 3

The function _loadHistory loads data from SharedPreferences and populates all the information into a map (the analyzed videos are stored in the SharedPreferences).

First, the “List dates = []” initializes an empty list to store the data retrieved from the SharedPreferences. The IF statement checks if the ‘date’ key exists in the preferences. If it does, it’ll retrieve the list of date strings stored under the ‘date’ key in the SharedPreferences and assign it to the dates list (The dates are in (prefs.getString(date)!). The “for loop” iterates each date string and date list, which are decoded and stored in the _history[date] map. Finally, the ‘setState(() {})’ triggers the rebuild of the widget tree and updates the data. It confirms whether the data of the uploaded videos is prepared.

4. EXPERIMENT

4.1. Experiment 1

A possible blind spot within the program is the AI pose accuracy. Therefore, we want to test the accuracy of the pose estimation, especially when the dancer is performing rapid movements, because it determines the overall quality of real time feedback.
We will put up an experiment using footage that involves rapid dance movements to test the accuracy of our app in challenging situations. We will record experimental data obtained from dynamic movements, and source control data from videos of slower dance movements to establish a baseline for comparison. The purpose of this experiment is to evaluate the program’s performance with different movement dynamics, manifesting how well it handles faster movements compared to slower ones.

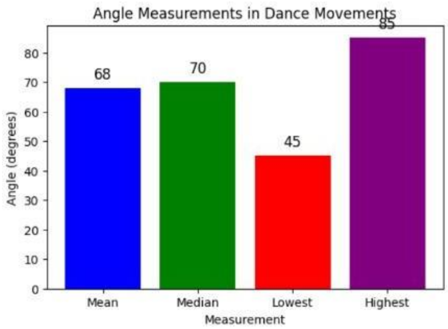

Figure 8. Figure of experiment 1

The mean angle in rapid dance movements was 68 degrees, the median was 70 degrees, the lowest measured angle was 45 degrees, and the highest measured angle was 85 degrees.

We observed the mean and median values were close despite the significant variation between the lowest and highest measured angles, which can be attributed to rapid dance movements. While rapid movements tend to introduce more errors in landmark tracking and angle calculations, the consistency of the mean and median suggest that there is a consistent spread/distribution of angle measurements with minimal extreme outliers.

Although rapid dynamic movements can introduce errors and inaccuracies to landmark tracking and angle calculations, we concluded our AI pose detection was relatively precise, but we can still work on continuing to improve the program's tracking algorithms and frame rate to further diminish these potential issues.

4.2. Experiment 2

Accurate distance and speed estimation are important because they impact object tracking and autonomous systems.

This experiment is designed to provide insights into the system's performance and limitations on the accuracy and functionality of the real-time distance and speed. The setup involved the recording of video frames with a measured width of objects as references. We recorded the time taken for each individual frame to process, which is a crucial factor in speed calculation. The actual distance and speed were collected to establish a baseline. The estimated (calculated by the code) and actual distances (measured in reality) and speeds will be compared and analyzed to assess the accuracy.

The first graph depicted the estimated distance over time in seconds, while the second graph showcased the estimated speed over time in meters per second.

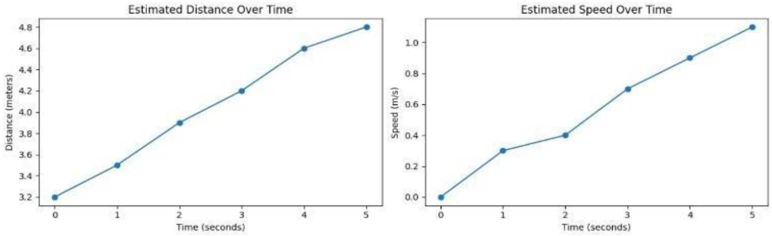

Figure 9. Figure of experiment 2

The analysis involved calculating mean and median values for estimated distances and speeds, revealing minimal variation in mean distance (approximately 4.5 meters). Lighting conditions significantly affected speed values, ranging from 0.2 to 1.0 meters per second. Object recognition accuracy, influenced by lighting, had a substantial impact on distance and speed estimation. The results emphasize the need to improve object recognition for accurate real-time distance and speed estimation under diverse lighting conditions.

Both experiments also indirectly addressed the challenge of environmental factors by training and testing the models on real-world datasets that contain a variety of environmental conditions. This approach allows us to evaluate how well the models perform in different lighting, weather, and time-of-day scenarios, as well as other factors that can impact trash detection accuracy.

Overall, by conducting these two experiments, we were able to assess the performance of two widely used computer vision models for trash detection and provide insights into their strengths, limitations, and potential areas for improvement.

5. RELATED WORK

In the paper "Detecting and Visualizing Stops in Dance Training by Neural Network Based on Velocity and Acceleration", the authors discuss how they developed a system in which they attached 18 small sensors to various parts of the dancer's body to record dance movements using motion capture technology [2]. A neural network model was employed to analyze and detect stops in dance movement. Subsequently, the system visualized the stops into a 3D model to help dancers understand the timing of stops.

The paper claims that the method achieves highly accurate stop detection results, which suggests the system is effective in identifying stops in dance movements.

The potential limitations are that the accuracy of a motion capture system may not be as high as optical, because they may suffer from sensor drift or calibration issues. The method may also ignore factors that affected the timing of stops, such as physical fatigue.

The utilization of sensors has relatively higher accuracy than AI recognition, because it relies on subsistence and stable coordinates of points. On the other hand, our project focuses on accessibility to a wide variety of people. Physical sensors aren't accessible for regular dancers with limited time and financial resources; while a single mobile device is the only requirement for Motion Mentor to function, generating similar results with the mentioned method.

The method for dance analysis uses multiple Kinect sensors (similar to methodology A) to address occlusion tracking problems and recognize dance patterns using Hidden Conditional Random Fields (HCRF). According to the source, this method demonstrates high recognition accuracy in motion patterns, such as dance figures [3].

However, there are some limitations of this solution. In its calibration the joint positions are being estimated, but the estimation can be erroneous, affecting the accuracy of the calibration and fusion process. Additionally, the method is evaluated based on experimental results, but it does not consider different dance styles or complex dance patterns.

On the other hand, our project incorporates a wider range of dance styles and complex patterns in its analysis and feedback system. Motion Mentor provides real-time feedback to dancers, leading to more effective learning and improvement.

The goal is to analyze video data using video processing and classification recognition to recognize different dance actions. This method uses multi-feature fusion, where the extracted features are combined to get a comprehensive representation of dance actions. According to the source, this system enhances the precision in information retrieval [4].

Some of the limitations are high dimensionality and redundancy of the fusion method, and the difficulty in accurately extracting key frames from dance videos. Additionally, the complexity of dance movements and the lack of research in this field pose challenges in accurately recognizing and categorizing different types of actions.

Our project improves on this by incorporating real-time posture correction and personalized feedback, addressing individualized dance techniques beyond general recognition.

6. CONCLUSIONS

Using a camera for pose detection in the app may have limitations in terms of precision and accuracy. Factors like lighting conditions and camera angles can affect the performance of the pose detection algorithm, leading to potential inaccuracies in tracking dance movements [12]. To improve this, machine learning techniques can be incorporated, and the pose detection model can be trained on a larger and more diverse dataset. This would enhance the algorithm’s ability to recognize and track different poses accurately.

While the app aims to provide feedback on dance performances, it may not be as comprehensive as feedback from an actual dance instructor. To supplement the feedback, a community platform can be implemented within the app. This platform would allow users to interact with other dancers and instructors, creating a supportive and collaborative environment. Users can share their progress, seek advice, and receive feedback from experienced individuals, enriching their learning experience.

In conclusion, by using machine learning techniques and incorporating learning materials, the app can improve pose detection accuracy and create a supportive learning environment for ballroom dancers.

REFERENCES

[1] Oliver, Wendy. "Body image in the dance class." Journal of Physical Education, Recreation & Dance 79.5 (2008): 18-41.
[2] Jin, Yuuki, Genki Suzuki, and Hiroyuki Shioya. "Detecting and Visualizing Stops in Dance Training by Neural Network Based on Velocity and Acceleration." Sensors 22.14 (2022): 5402.
[3] Kitsikidis, Alexandros, et al. "Dance analysis using multiple kinect sensors." 2014 international conference on computer vision theory and applications (VISAPP). Vol. 2. IEEE, 2014.
[4] Luo, Wanshu, and Bin Ning. "High-dynamic dance motion recognition method based on video visual analysis." Scientific Programming 2022 (2022): 1-9.
[5] Shum, Hubert PH, et al. "Real-time posture reconstruction for microsoft kinect." IEEE transactions on cybernetics 43.5 (2013): 1357-1369.
[6] Mao, Bomin, et al. "AI models for green communications towards 6G." IEEE Communications Surveys & Tutorials 24.1 (2021): 210-247.
[7] Bläsing, Bettina, Gershon Tenenbaum, and Thomas Schack. "The cognitive structure of movements in classical dance." Psychology of Sport and Exercise 10.3 (2009): 350-360.
[8] Mohamed, Nachaat. "Importance of Artificial Intelligence in Neural Network through using MediaPipe." 2022 6th International Conference on Electronics, Communication and Aerospace Technology. IEEE, 2022.
[9] Wei, Zhenzhong, Fuqiang Zhou, and Guangjun Zhang. "3D coordinates measurement based on

structured light sensor." Sensors and Actuators A: Physical 120.2 (2005): 527-535.

[10] Li, Wu-Jeng, et al. "JustIoT Internet of Things based on the Firebase real-time database." 2018 IEEE International Conference on Smart Manufacturing, Industrial & Logistics Engineering (SMILE). IEEE, 2018.

[11] Bonnet, Noël. "Artificial intelligence and pattern recognition techniques in microscope image processing and analysis." Advances in Imaging and Electron Physics. Vol. 114. Elsevier, 2000. 1-77.

[12] Rogez, Grégory, et al. "Randomized trees for human pose detection." 2008 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2008.

[13] Lugaresi, Camillo, et al. "Mediapipe: A framework for building perception pipelines." arXiv preprint arXiv:1906.08172 (2019).

[14] Ferreira, Jennifer, James Noble, and Robert Biddle. "Agile development iterations and UI design." Agile 2007 (AGILE 2007). IEEE, 2007.

[15] Hamera, Judith. "Dancing communities." Performance, Difference and Connection in the Global City (2007).