

관심도 레벨 측정을 위한 멈춤 행동 판단 시스템¹

양지윤^{1,0}, 최유주^{2,*}

¹ 서울미디어대학원대학교 VCAR 연구실

² 서울미디어대학원대학교 인공지능응용소프트웨어학과

njs10919@gmail.com, yjchoi@smit.ac.kr

A stationary behavior determination system for measuring the level of interest

Jiyeon Yang^{1,0}, Yoo-Joo Choi^{2,*}

¹ VCAR Lab., Seoul Media Institute of Technology

² Department of AI Software Engineering, Seoul Media Institute of Technology

*Corresponding Author

요 약

본 논문에서는 전시회의 관람객을 탐지하고 추적하여 실시간으로 속도를 측정하여 전시 부스에서 관람객의 속도가 줄어드는 것을 파악한 후 멈춤 행동을 판단하는 시스템을 제안한다. 해당 시스템은 전시회 부스에 웹 캠을 설치하여 그 관찰 영역에 들어온 관람객을 인식한 후 관람객의 속도를 예측하고 관람객의 속도 변화를 논문에서 정의한 관람객의 여러 상태로 변화시킨 후 새롭게 정의한 멈춤 행동에 해당하는 관람객의 관심도를 추정하게 된다.

1. 서론

최근 전시회의 품질을 높이기 위해 관람객들의 행동과 만족도 등을 분석하는 연구가 진행되고 있다. 관람객의 정보를 분석하는 것은 전시회를 주최하기 이전에 관람객들의 만족도를 높일 수 있는 전시 계획을 세우는 데 사용하거나 전시가 완료된 이후 어느 정도의 성과를 이루었는지 판단하는 지표로 사용할 수 있다. 그렇기 때문에 전시품에 관심이 높은 관람객에 대해 집중 탐구를 하기 위해서 관람객의 행동이나 표정, 움직임의 특성들을 분석한다. 국내 연구 사례에서 RFID 와 모바일 어플리케이션을 사용하여 전시의 데이터를 수집하는 연구를 진행하였다. [1,2] 이 방식은 관람객이 전시를 관람하면서 직접 인식 장치를 착용하거나 핸드폰에 어플리케이션을 설치해야 하기 때문에 모든 관람객에 대해서는 정보를 모으기 어렵다. 외국의 연구에서는 루브르 박물관에 온 관람객의 정보를 수집하기 위한 방법으로 블루투스를 사용하였다. [3,4] 하지만 실제 관람객의 8.2%만 블루투스를 사용했기 때문에 대다수의 관람객의 데이터를 수집하지는 못했다.

이전 논문에서는 관람객들의 행동을 추정하기 위해 실시간으로 관람객의 이동 속도를 계산하는 모듈을 연구했다. [5] 관람객이 특별한 장비 없이 웹 캠의 관찰 영역안에 들어서면 관람객의 위치를 파악하고 움직이는 속도를 측정할 수 있었다. 이를 위해서는 해당 관람객을 계속 탐지하고 id 를 부여한 후 계속해서 트래킹을 통해 위치와 속도를 업데이트 하였다. 본 논문에서는 관람객의 만족도를 판단하기 위한 지표로 관람객의 속도를 측정하였고 관람객이 해당 부스를 관찰하기 위해 속도를 줄이고 그 부스에 오랫동안 체류해 있다면 관심을 보이고 있다고 정의하고 해당 관람객의 관심도 레벨을 조정하는 시스템을 구축하였다.

2. 제안 시스템

본 연구에서 제안하는 것은 관람객의 이동 속도가 특정 속도 이하로 내려가게 되면 전시물 앞에 서서 관심을 보이는 행동으로 보고 이를 이용하여 특정 속도 이하로 내려가는 시간을 측정하여 일정 시간 유지가 된다면 그 관람객의 관심도 레벨을 올려서 전시품에 관심을 보인다고 판단한다. 이를 이용하여 관람객

¹ 본 연구는 문화체육관광부 “관광서비스 혁신성장 연구개발사업” (R2022020105)의 지원에 의하여 수행되었음

의 관심도를 실시간으로 판단할 수 있다.

2.1 관람객 탐지를 위한 새로운 객체 생성

객체 추적 네트워크를 통해서 새로운 객체 즉 관람객이 관찰되었음을 인지하였을 때, 관람객 클래스 인스턴스가 생성되어 관람객의 주요 정보가 저장되도록 한다. 관람객의 정보를 저장하는 클래스의 이름은 `trackedObject` 라고 하고 해당 객체는 `appearTime`, `speed`, `centerPoint`, `state`, `slowdownTime`, `frameNum`, `stayTime`, `accumulatedTime`, `interestLevel` 등의 속성값을 갖는다. 처음 객체가 탐지되었을 때 생성자를 통해 `appearTime`, `speed`, `centerpoint`, `state` 를 지정하고 나머지 매개변수들은 시간이 지남에 따라 값들이 변하기 때문에 우선 0 으로 초기화한다. <표 1>은 `trackedObject` 클래스의 주요속성값에 대한 설명을 보여주고 있다.

<표 1> 추적객체 클래스의 주요 속성값

속성이름	저장 정보 내용
<code>appearTime</code>	객체의 출현 시각
<code>speed</code>	객체의 평균 속도
<code>centerPoint</code>	객체 중심 좌표
<code>state</code>	현재 객체의 상태
<code>slowdownTime</code>	느려지기 시작한 시각
<code>frameNum</code>	느려지기 시작한 시점의 프레임 넘버
<code>stayTime</code>	멈춤 행동이 유지된 시간
<code>accumulatedTime</code>	현재 멈춤 행동 이전의 멈춤 행동이 유지된 시간
<code>interestLevel</code>	객체의 관심도 레벨

2.2 멈춤 행동을 판단하기 위한 알고리즘

다음은 멈춤 행동을 판단하기 위한 알고리즘을 수도 코드로 나타낸 것이다.

Algorithm 1. Tracked Object State variation

Input: `Id`, `tracked object box`, `tracked object average speed`, `current time`

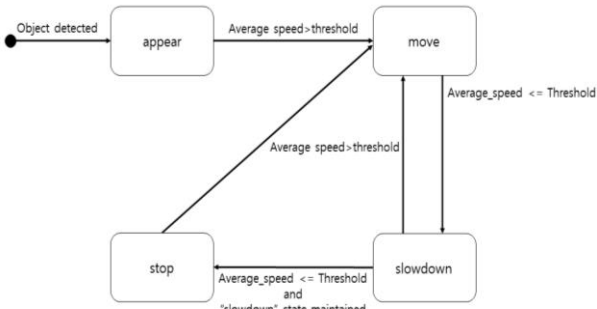
Output: Dictionary of Tracked Object (= `TrackedObject[Id]`)

- (1) `Object = tracked Object(appearTime = First_tracked_time, speed = Average_speed, centerPoint = centerPoint)`
- (2) **If** `TrackedObject` is empty:
- (3) `TrackedObject[Id] = Object`
- (4) **elif** `TrackedObject` is not empty and `Id` is not in `TrackedObject`:
- (5) `TrackedObject[Id] = Object`
- (6) **elif** `Id` is in `TrackedObject`:
- (7) **if** `Average Speed <= Threshold` and `TrackedObject[Id].state` is not “slowdown” and “stop”:
- (8) `TrackedObject[Id].state = “slowdown”`

- (9) **elif** `Average Speed > Threshold` and `TrackedObject[Id].state` is not “move”:
- (10) `TrackedObject[Id].state = “move”`
- (11) **elif** `Average_speed <= Threshold` and `TrackedObject[Id].state` is “slowdown”:
- (12) **if** `slowdown state maintained more than 10 frames`:
- (13) `TrackedObject[Id].state = “stop”`
- (14) Calculate accumulated time
- (15) **elif** `TrackedObject[Id].state` is “stop”:
- (16) `stayTime = Current_time - TrackedObject[Id].slowdownTime`
- (17) **if** `TrackedObject[Id] stayTime + accumulatedTime > 5 sec`:
- (18) Raise the Interest Level(level 1)
- (19) **if** `TrackedObject[Id] stayTime + accumulatedTime > 10 sec`:
- (20) Raise the Interest Level(level 2)
- (21) **End**

입력으로는 추적된 객체에 대한 정보와 현재 시간을 넘겨준다. 객체의 정보는 추적된 객체의 `Id`, 객체의 위치를 나타내는 `box` 정보, 객체의 평균 추정 속도이다. 딥서너리를 이용하여 `trackedObject` 클래스를 각각의 `Id` 로 구분한다. 알고리즘의 라인(1~5)에서 처음 객체들이 탐지되고 추적되기 시작할 때 딥서너리에 정보를 담는다. 라인 (6~16)에서 각 프레임마다 모든 객체들의 정보가 계속해서 업데이트되기 때문에 해당 프레임에서 평균 속도와 임계 속도를 비교해서 각 객체의 상태를 바꿔준다. 객체속도 측정 방법은 다음절에서 상세히 설명하겠다. 라인 (12~13)에서 객체의 평균 속도가 임계 속도보다 줄어든 상태에서 10 프레임 이상이 유지된다면 멈춤 행동으로 판단하고 `state` 를 “stop”으로 변경해준다. 라인 (17~20)에서는 이렇게 “stop” 상태의 누적 시간이 5 초 이상 유지되면 관심도가 level 1 로 증가하고 누적 시간이 10 초 이상 유지되면 관심도가 level 2 로 증가시킨다. 다음 그림 1 은 탐지된 객체의 네 가지의 상태전이도이다.

상태전이도는 크게 4 개의 상태를 가진다. `appear` 상태는 관찰영역에 관람객이 탐지된 상태이고, `move` 상태는 탐지된 객체가 움직이는 상태이다. 논문에서 정의한 `threshold` 이하로 속도가 내려가면 `slowdown` 상태로 변경된다. `slowdown` 상태가 10 프레임 이상 유지가 된다면 `stop` 상태로 변경된다. 상태간 이동을 발생 하는 이벤트는 화살표 위에 표시하였다.



(그림 1) 상태전이도

2.3 객체의 속도 측정 방법

객체의 속도를 계산하기 위해서 객체와 카메라 사이의 거리, 사람의 너비와 프레임 속의 너비가 필요하다. 이 실제 거리와 프레임 속의 너비를 비교하여 카메라의 초점 거리를 구한 후 초점 거리를 사용해 각 프레임마다 사람의 너비를 계산하여 실제 카메라와의 거리를 계산한 후 객체의 속도를 거리 기준으로 도출한다. [5]

3. 실험 및 결과

제안 알고리즘의 정확도를 측정하기 위하여, 서로 다른 속도로 카메라 앞에서 움직이며, 측정되는 속도와 이를 통하여 예측되는 움직임을 화면에 표시하였다. (그림 2)는 속도를 높여 지나가는 객체를 표시하고 있고, (그림 3)은 속도가 줄어들고 있는 객체, (그림 4)는 일정 시간 속도가 줄어서 멈춤 상태를 유지하고 있어서, 멈춤 상황인 인지된 상태를 보여 주고 있다.



(그림 2)빠르게 움직이고 있는 객체



(그림 3)속도가 줄어드는 사람의 상태 표시(slowdown)



(그림 4) 속도가 줄어든 후 멈춤 상태(stop)

(그림 2)에서는 객체가 탐지된 이후 움직이고 있는 상태를 화면에 표시된 정보로 확인할 수 있으며, (그림 3)에서는 속도가 줄어든 사람의 상태가 “slowdown”으로 변경된 것을 볼 수 있고 (그림 4)에서는 “stop” 상태가 5 초 이상 유지되어 관심도 레벨이 level 1로 증가한 것을 확인할 수 있다.

4. 결론

본 논문에서는 웹 캠을 이용하여 전시에서 관람객의 속도를 실시간으로 탐지하고 그 속도를 기반으로 멈춤 행동을 판단한 후 관람객의 관심도를 추정하였다. 현재는 파이썬 객체로 구현된 관람객의 정보를 따로 저장하고 있지 않아 프로그램을 종료하게 되면 추정한 관람객의 관심도를 분석할 수 없다. 추후에는 실시간으로 추정된 관람객의 속도 및 관심도 정보를 데이터베이스에 저장하여 시간대 별로 관람객이 전시회 부스에 관심을 보였는지 분석을 할 수 있도록 발전시킬 예정이다.

참고문헌

- [1] B. Kim, H. Jung, K. Bang, and Y. Ko, “The Design and Implementation of Mobile RFID System Using Indoor Location-Support System,” Journal of The Korean Institute of Information Scientists and Engineers, Vol. 33, No. 2(D), pp. 812-817, 2006.
- [2] J. Park, Y. Kim, and C. Park, “Design of Walking Management Application based on Beacon in Museum Tour Guide,” Proceedings of The Korean Institute of Information Scientists and Engineers, pp. 476-478, 2019.
- [3] Y. Yoshimura et al., “An Analysis of Visitors’ Behavior in the Louvre Museum: A Study Using Bluetooth Data,” Environment and Planning B: Planning and Design, Vol. 41, No. 6, pp. 1113-1131, 2014.
- [4] Y. Yoshimura, F. Girardin, J.P. Carrascal, and C. Ratti, J. Blat, “New Tools for Studying Visitor Behaviours in Museums: A Case Study at the Louvre,” Information and Communication Technologies in Tourism, pp. 391- 402, 2012.
- [5] J. Yang , Interest Level Measurement System based on Object Speed, ASK 2023 Annual Spring Conference of KIPS, Korea Information Processing Society , 2023, p765-767