

RGB-D 카메라 및 시계열 데이터 딥러닝 기반 탁구공 타점 예측 시스템

최원철*, 손경성**, 지원**, 조현주**, 김도연**, 서동현**, 김선욱**, 최유진**, 배원규**
4D 트래블러*, 숭실대학교**

RGB-D camera and time series data deep learning-based ping-pong ball trajectory prediction system

Won-Chil Choi*, Won Ji**, Hyeon-Ju Joe**, Do-Yeon Kim**, Dong-Hyeon Seo**, Sun-Uk Kim**,
Yu-Jin Choi**, Won-Gue Bae**
4D Traveler*, Soongsil University**

Abstract - 생활체육 및 재활운동으로 많은 관심을 받는 탁구는 상대방이 반드시 존재해야만 진행 가능하다. 이에 대안으로 단방향으로 쓰는 탁구기계가 존재하지만 탁구공을 되받아 칠 수 없다. 본 논문은 탁구공을 되받아 치는 랠리 가능한 시스템을 제안하기 위하여 RGB-D 카메라에서 탁구공의 궤적을 시계열 데이터로 받아 인공지능 학습 기법을 통해 탁구공의 최종 도달 위치를 예측하는 탁구공 타점 예측 시스템을 제안한다. 제안 시스템은 RGB-D 가 실시간으로 탁구공의 움직임을 촬영하면 프레임별 탁구공의 위치를 3차원 좌표계로 나타내는 프로세스를 거쳐 시계열 데이터를 생성하고 해당 데이터를 미리 학습된 GRU 모델을 통해 최종 타점의 좌표를 탁구공이 도달하기 전에 예측한다. 정밀도 평가에서 해당 시스템은 허용 가능한 수치를 보여주었고 평균 오차는 약 7.2cm, 속도는 약 0.25초로 측정이 되었다.

SLAM 등에 사용되어 주변 환경에 대한 인식 수단으로 많은 연구가 이루어지고 있다.[2] 본 논문에서는 RGB-D 카메라를 통해 탁구공을 인지하고 xyz 3차원 좌표 값을 추출하기 위해 먼저 촬영된 RGB이미지에서 탁구공의 색상 값을 mask 하여 바이너리 이미지로 변환을 한다. 이때 조명의 영향을 최소화 하고 색상 값을 강조하기 위해 RGB 공간을 HSV 공간으로 변형을 한다.[3] 다음 <그림 1>처럼 추출된 탁구공의 바이너리 이미지는 카메라 프레임 한계로 인해 잔상이 포함된다. 이렇게 잔상이 존재하는 탁구공의 영역에서 명확한 좌표 점을 찾기 위해 무게 중심의 좌표 점으로 데이터를 추출한다. 바이너리 이미지에서 Opencv 라이브러리의 Contour 알고리즘을 통해 윤곽선을 추출하고 해당 윤곽선으로 부터 Moments 알고리즘으로 0차 1차 모멘트 값으로 중심 값을 계산하였다.[4]

1. 서 론

탁구는 타 스포츠에 비해 날씨 환경의 영향을 받지 않아 접근성이 좋고 누구나 쉽게 배울 수 있으며 활발한 신체적 능력 사용뿐만 아니라, 정신적 건강에도 영향을 미칠 수 있는 구기 종목이다. 이에 일반인에게 생활체육으로서, 노인이나 장애인들에게는 재활운동으로서 탁월한 운동이 되어 많은 관심을 받고 있다.

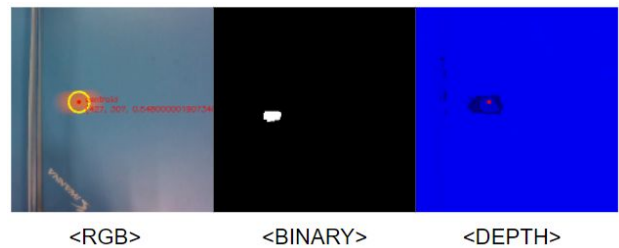
하지만 탁구 특성상 상대방이 반드시 존재해야만 진행이 가능한 문제점이 있고, 해당 방안으로 존재하는 탁구 기계는 단방향으로 공을 쏘아줄 뿐 되받아 칠 수 가 없다. 이런 단방향 탁구 기계를 활용 시 상대방과 함께 치는 랠리 형식에 비해 운동 효율이 적고 재활 운동으로서의 가치도 떨어지게 된다.[1]

단방향 탁구기계 문제점을 보완하는 rally 가능한 시스템 구현을 위해 실제 사용자가 치는 탁구공의 궤적으로 부터 탁구공을 받아낼 수 있는 위치를 알 수 있는 시스템이 선행 되어야 한다. 본 논문에서는 RGB-D 카메라로 탁구공의 궤적을 실시간으로 촬영하고 시계열 좌표 데이터로 전처리하는 프로세스, 초기 프레임의 시계열 데이터와 최종 탁구공 도달 지점의 데이터가 라벨링 된 데이터셋을 통해 GRU 순환신경망을 학습하는 프로세스, 초기 프레임의 시계열 데이터가 입력되면 탁구공 도달 지점을 예측하는 프로세스를 제안하여 최종 탁구공의 도달 위치를 실제 탁구공이 도달하기보다 먼저 예측하는 시스템을 구현한다.

2. 본 론

2.1 RGB-D 카메라를 통한 좌표데이터 추출

RGB-D 카메라는 RGB색상 카메라와 TOF 거리 센서를 동기화하여 촬영된 물체의 형상 및 거리를 동시 측정 가능하기에



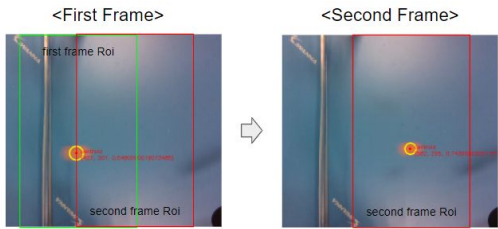
<그림 1> 왼쪽 RGB 이미지로부터 H영역이 mask된 Binary 이미지와 RGB 이미지와 동기화된 Depth 이미지

컬러 이미지로 부터 탁구공의 무게중심점을 (x, y) 좌표 값으로 추출 후 z 값을 추출하기 위해 RGB 좌표계와 동기화된 Depth 좌표계에서 (x, y)에 해당하는 z값을 사용한다. RGB카메라와 거리 센서간의 fps동기화 오차로 인해 탁구공 무게중심점이 두 좌표계에서 다르기 때문에 (x, y) 주변영역의 z값에서 가장 가까운 z 값을 추출하여 (x, y, z) 3차원 좌표데이터를 추출한다.

2.1.1 시계열 데이터

탁구공은 특정 방향으로 움직이는 운동성이 존재하기 때문에 일정 시간 간격으로 배치되는 공간 값의 수열인 시계열 데이터로 나타낼 수 있다. 본 논문에서는 카메라 프레임에 따라 특정 시간 간격으로 추출된 탁구공의 무게중심 좌표를 차례대로 나열하여 시계열 데이터로 표현 한다.

이때 RGB 값에서 발생하는 연산에 의한 딜레이를 줄이기 위해 <그림 2> 같이 프레임 별로 ROI 를 최대한 줄이고 현재 프레임에서 추출된 y 좌표 값 이후로 ROI 영역을 잡아주어 연산을 최소화 시켰다. 또한 Thread 로 작업을 나누고 Queue 구조를 이용한 동기화를 통해 연산 딜레이로 인한 프레임 누락 현상을 해결하였다. 해당 과정을 통해 초기 4프레임만을 데이터로 추출하여 최종적으로 (4,3)의 모양을 가진 배열로 시계열 3차원 좌표 데이터를 얻는다.



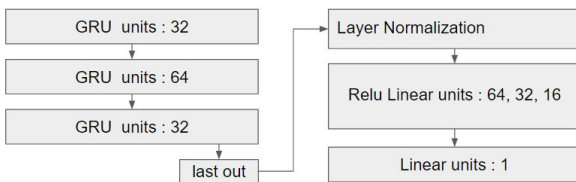
<그림 2> 이전 프레임 중심점이 시작영역이 되는 ROI 방식

2.2 GRU 딥러닝 모델

RNN 딥러닝 구조는 time step을 가지고 있어 시계열 데이터에서 시간의 순서 특성을 잃지 않을 수 있는 딥러닝 구조중 하나다. GRU 딥러닝 모델은 RNN 구조의 하나로 LSTM 과 달리 출력게이트는 없지만 기본RNN 구조에서 두개의 sigmoid 를 추가하여 vanishing & exploding gradient 문제를 줄이고 LSTM 보다 가벼운 장점을 가지고 있다.[5] 본 논문에서 추출된 데이터는 모델의 인풋 데이터의 모형이 (4,3)의 작은 모형이기 때문에 time step이 길지가 않고 입력 데이터도 크지 않기 때문에 LSTM 보다 가벼운 GRU모형을 사용하여 빠른 예측에 집중을 하였다.

2.2.1 모델 구현

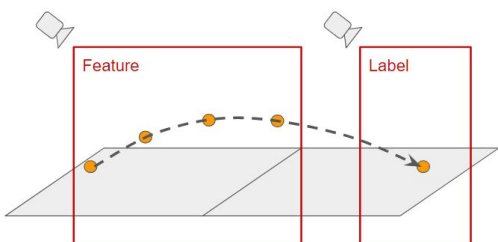
본 논문에서는 <그림 3> 같이 3개 층의 GRU cell 을 적용하였고 각각의 units 는 32, 64, 32 으로 주었다. 예측하고자 하는 값은 시계열 가장 마지막에 나타나는 탁구공의 최종 도달점에 해당하는 좌표 값이므로 3번째 GRU cell 에서 마지막 step 에 해당 하는 output 만을 사용하여 Layer Normalization 레이어와 다층 Relu linear 레이어를 통과시켜 주었다. 최종적으로 1개의 units을 가지는 linear 레이어의 결과 값을 예측 값으로 하여 meansquarederror 손실을 통해 학습을 진행하게 해주었다. 해당 모델은 결과 값이 1개인 단일 좌표 점을 예측하므로 x, y, z 값에 대하여 동일한 모델을 적용해주었고 병렬처리를 통해 3개의 예측 값을 동시에 도출하게 하였다.



<그림 3> 본 논문의 학습 및 예측에 사용된 GRU 기반 모델

2.2.2 모델 학습 및 평가

본 논문에서 구현한 GRU 모델을 학습하기 위해 총 2대의 RGB-D 카메라를 이용하여 데이터 세트를 구성한다. <그림 4>에서 feature 카메라는 탁구공의 출발 지점에 해당하는 영역을 촬영 하여 초기 4프레임에 해당하는 데이터를 추출한다. label 카메라는 탁구공의 도달 지점에 해당하는 영역을 촬영하여 한 프레임에 해당하는 단일 좌표데이터를 추출한다. 이때 label 카메라는 feature 카메라에서 탁구공이 인지된 초기 4프레임이 촬영된 후에만 촬영이 되게 프로그래밍 하여 모델을 훈련시킬 수 있는 4 frame feature - label 탁구 궤적 데이터 셋을 구축하였다.



<그림 4> 학습을 위한 데이터 세트 형성 방법 및 카메라 구도

카메라가 세팅된 환경에서 반복적으로 탁구를 진행하여 쌓은 약 3만개의 데이터세트에 고정된 평균 분포 값으로 z-score 정규화 후 GRU 모델을 훈련 시켜주었다. 학습 과정에서는 AdamOptimizer을 사용하여 256 batch-size 로 500 epochs 로 진행 해주었다. learning rate scheduler 는 0.002로 시작하여 100 epoch 마다 0.0005씩 줄여준 간단한 스케줄을 사용하였고 ModelCheckpoint를 통해 best validation loss 값을 가지는 epoch를 저장 하였다. 본 논문에서 진행한 case에서 저장된 학습 모델로 부터 약 2000개의 테스트 데이터세트에 대해 예측한 값 <표 1>을 환산한 결과 실제 도달지점 과의 평균 7.2cm 오차가 발생하고 예측시간은 약 0.05 - 0.06초가 소요된다.

<표 1> 정규화 된 테스트 데이터세트에 대한 예측 값 표본

frame1(x,y,z)... frame4(x,y,x)	y label	y predict
(8.27,-7.74,1.04)... (8,13,-0.90,3.64)	139.0	160.165
(8.62,-7.75,-5.54)... (6.78,0.64,0.40)	234.0	259.569
(1.76,-5.42,-2.19)... (0.84,2.06,1.09)	363.0	341.939
(5.02,-4.26,5.31)... (5.37,2.84,9.64)	123.0	144.100
(0.01,-7.74,-3.63)... (-0.56,1.03,1.61)	366.0	378.627

3. 결 론

본 논문에서는 RGB-D카메라와 GRU 딥러닝 모델을 통해서 탁구공의 최종 도달점을 빠르게 예측하는 시스템을 제안한다. 시스템에서 총 소요되는 시간은 탁구공을 부터 시계열 데이터를 추출하는데 0.1초, 모델로 부터 예측 값을 추출하는데 평균 0.055 초, 데이터 전처리 및 환산 과정을 포함해도 약 0.2 - 0.3 초 안에 최종 도달점을 예측 한다. 실제로 탁구공이 반대 지점으로 날아가는 시간은 평균 0.6초 정도가 소요된다. 또한 약 3만개의 데이터세트로 시스템은 약 7.2cm 오차에 해당하는 정확도를 보이고 탁구공에 회전이 최소화된 데이터인 것을 감안하면 데이터 추가 수집 및 다양화, 모델 성능향상에 대한 연구가 진행 되어야 한다. 향후 연구를 통한 정확성 향상이 이루어진다면 해당 시스템은 펠리가 가능한 탁구 로봇 등에 활용되어 혼자서도 재할, 생활 운동으로서 탁구가 가능하게 하여 국민 건강에 기여할 수 있을 것으로 예상된다.

[참 고 문 헌]

- [1] 조재훈, "장애인 재활체육의 딜레마와 생활체육과의 연계"
- [2] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., Vilamoura, Portugal, Oct. 2012, pp. 573 - 580.
- [3] W. Chen, Y. Shi and G. Xuan, "Identifying Computer Graphics using HSV Color Model and Statistical Moments of Characteristic Functions," IEEE ICME, pp.1123-1126, July 2007.
- [4] L. Rocha, L. Velho, and P. C. P. Carvalho. Image moments-based structuring and tracking of objects. In SIBGRAPI, pages 99 - 105, 2002.
- [5] Junyoung Chung, Çağlar Gülçehre, Kyunghyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. CoRR, abs/1412.3555, 2014.