

비콘으로 얻은 이벤트 스트리밍을 통한 소비자 동선 분석

이현주*, 김성수*, 정태선*
 *아주대학교 컴퓨터공학과
 e-mail : sigran@ajou.ac.kr
sskim@ajou.ac.kr
tschung@ajou.ac.kr

Analysis of the flow of Consumer traffic through event streaming from Beacon

Hyun Joo Yi*, Sungsoo Kim*, Tae-Sun Chung*
 *Dept. of Computer Science, Ajou University

요 약

매장은 특정 구역들로 분할되어 있고, 각 구역들은 고유의 비콘을 가지고 있다. 현재 소비자가 있는 구역의 비콘이 자신의 고유 ID 를 소비자의 단말기(스마트폰)에 전송하면, 단말기는 임의의 번호를 자신의 고유 ID 로서 생성하여 [Beacon ID, 단말기 고유 ID, 현재 시간]을 서버로 보낸다. 그러면 서버쪽에선 사용자의 현재 위치를 파악하는 동시에 사용자가 그 자리에 얼마나 오래 있었는지를 계산해 저장한다. 이러한 과정의 결과로 만들어진 하나의 SEQ 가 매장 주인이 생각했던 SEQ 와 얼마나 유사한지를 비교하여 그 결과를 보여준다.

1. 서론

상점의 주인들은 소비자들의 동선에 맞추어 유행하는 상품들과 그렇지 못한 상품들을 분류하여 매장에 배치한다. 그러나 이때 중요한 실제 소비자들의 동선 파악의 경우, 대부분 직원들이 소비자들의 상품 구매를 도와주며 어림짐작해서 파악하는 것이 전부인 경우가 많다. 그렇기에 보다 정확한 소비자 동선을 파악하는 것이 필요하다. 본 논문에서는 매장을 특정 구역들로 나누어 놓은 뒤 각 구역마다 배치한 비콘(무선통신 장치로써 블루투스 4.0 (BLE- Bluetooth Low Energy) 기반의 프로토콜을 사용해 매우 작은 주파수 시그널을 그들 주위에 전달하는 전자기기)을 이용해 소비자의 동선 파악과 각 구역에 머문 시간을 계산하여 가장 인기가 많았던 구역과 이동 순서를 특정 짓는 방법에 대하여 고안해 보았다.

2. 시나리오

소비자가 한 매장의 입구를 통과하는 순간, 매장 입구에 설치되어 있던 비콘에서 소비자의 단말기로 자신의 ID 를 보낸다. 그러면 소비자의 단말기가 자신의 단말기에서 랜덤으로 생성된 고유 ID 와 현재 시간, 그리고 비콘의 ID 를 서버로 보낸다. 이때 비콘의 고유 ID 는 'Exit Beacon Id'로 인식되어 시퀀스 (이하 SEQ)의 시작을 알려주게 된다.

아래의 그림은 해당 매장을 임의의 9 개의 구역들

로 나눠놓은 것을 보여준다. 소비자는 매장 들어서자마자 구역 A 로 들어갈 것이다. 그리고 그 즉시 일정한 시간(T)을 간격으로 구역 A 의 비콘이 보내는 신호를 해당 소비자의 단말기가 캐치하여 서버에게 보냄으로서 현재 A 가 있는 위치가 매장의 어느 구역인지를 파악할 수 있게 해준다.



(그림 1) 매장의 구역분할

A 구역의 상품들을 모두 둘러본 소비자는 이제 자신의 눈길을 끄는 다른 구역으로 이동할 것이다. 이때 선택지는 가까운 B, D, H 를 포함해 멀리 있는 C, F, G, I 구역들이 있다. 사용자가 새로운 구역에 들어서게 되면 서버로 보내지는 비콘 ID 가 해당 구역의 비콘의 ID 로 바뀌게 된다. 서버는 보내진 이벤트를 분석하여 소비자의 변한 위치와 그 전 구역에 머물렀던 시간을 계산한 뒤 그 값을 특정 매트릭스에 넣는다.

사용자가 매장에서 모든 볼 일을 다 본 다음 매장

을 빠져 나가면, 또다시 매장 입구에 설치되어 있던 비콘에서 소비자의 단말기로 자신의 ID 를 보내게 되어 SEQ 의 종료를 알린다.

3. 플로우 차트

본 논문에 보여질 플로우 차트에는 4 가지 파트가 있다. Event Creator, Event Gathering, Similarity Comparing, 그리고 Scoring 이 그 4 가지 파트이다. 해당 파트들은 다음과 같이 순차적으로 연결된다.



(그림 2) 전체적인 플로우 차트

Event Creator 는 말 그대로 이벤트를 만들어서 계속 전송함으로써 스트림이 되게 하는 것을 말한다. 다음 파트인 Event Gathering 은 그러한 스트림 형태로 들어오는 이벤트들을 같은 단말기 ID 를 가지고 있는 이벤트들끼리 시간순서대로 묶어서 하나의 SEQ 를 만들어 주는 파트이다. 그렇게 묶은 SEQ 를 다음 파트인 Similarity Comparing 으로 넘겨주면 상점의 주인들이 가장 유력할 거라 예상하는 경로와 Similarity 를 비교하게 된다. 이 파트에서 다음 파트로 각 SEQ 와 그에 해당하는 Similarity 를 전송한다. 그러면 맨 마지막 파트인 Scoring 파트에서는 상점의 주인들이 생각했던 SEQ 와 실제 결과들 중에서 가장 높은 Similarity 정도를 가진 SEQ 를 같이 보여줘 비교가 가능하게 한다. 그리고 동시에 한 구역에서 다른 구역으로 넘어가는 경우의 수를 모두 비교해, 가장 많은 경우의 수로 이루어진 새로운 SEQ 를 제시해 주고, 가장 머문 시간이 높았던 구역이 어디였는지도 보여준다. 다음은 각 파트들에 대한 좀 더 구체적인 설명들이 기술되어 있다.

3-1. Event Creator (이벤트 생성기)

Event Creator 에서는 비콘에서 전송되는 각 구역에 설치된 비콘들의 고유 ID 가 단말기에 전송되고, 해당 단말기에서 [Beacon ID, 단말기 고유 ID, 현재 시간]가 서버로 전송된다. 그리고 서버는 이렇게 받은 이벤트들을 스트림 형식으로 그 다음 파트인 Event Gathering 에 넘겨주게 되는 것이다.

아래 그림은 비콘과 단말기, 그리고 서버가 어떤 형식으로 정보를 주고 받는지를 보여준다.



(그림 3) 비콘과 단말기, 서버의 통신

비콘의 최대 통신 가능 거리는 50m 이나, 본 논문에서는 그 거리를 제한하여 매장의 각 구역을 넘나들 시 서로 다른 비콘의 신호가 수신된다고 가정하였다.

그 후, 그렇게 한 비콘에서 수신된 ID 와 현재 시간을, 단말기 자체에서 생성된 고유 ID 안에 속성으로 집어 넣는다. 즉, [고유 ID].BID, [고유 ID].TIME 으로 표현될 수 있다. 좀 더 간략한 표현을 위해 다음과 같이 기호화 하였다.

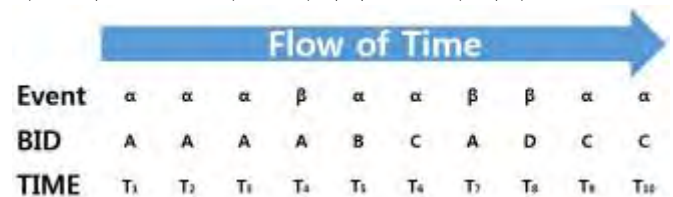
α	단말기 자체에서 생성된 고유 ID
α .BID	Beacon ID
α .TIME	현재 시간

<표 1> 각 요소들의 기호화

이렇게 만들어진 이벤트를 WiFi 를 통하여 단말기에서 서버로 보내면 Event Creator 파트는 끝나게 된다. 그리고 이제까지의 전체적인 과정을 해당 파트에서는 T 시간 간격으로 진행한다.

3-2. Event Gathering (이벤트 모으기)

소비자의 단말기에서 보내진 이벤트들은 하나의 스트림이 되어 서버로 전달되어 진다. 아래의 그림은 그러한 이벤트 스트림을 가시화 한 그림이다.



(그림 4) 시간의 흐름에 따른 이벤트들과 그 속성들 [1]를 참조하여 만들었다

이렇게 들어온 각 이벤트들은 아래의 단계들을 통해 하나의 SEQ 가 된다.

<Step 1>

1. 단말기 고유 ID (이하 event)가 들어오면 Stack 에 event 를 검색한다.
- 2-1. event 가 검색되지 않을 경우 <Step 2>의 1-1로 이동한다.
- 2-2. event 가 검색될 경우, <Step 2>의 1-2로 이동한다.

<Step2>

- 1-1. 변수 i 를 0 으로 초기화 하고, event 를 SEQ_

event[i]에 집어 넣는다.

- 1-2. event.BID 가 SEQ_event[i]와 일치하는지 체크한다. 일치하면 2 로 넘어간다. 불일치하면 3 으로 넘어간다.
2. event.BID 가 SEQ_event[i].BID 가
 - 2-1. 일치할 경우 아무것도 안한다.
 - 2-2. 일치하지 않을 경우, 변수 i를 하나 증가시켜 주고, event 를 SEQ_event[i]에 넣어준다. SEQ_event[i].TIME 에는 { SEQ_event[i].TIME - SEQ_event[i-1].TIME } 를 계산해서 넣어준다.
3. SEQ_event 를 끝낸다.

이렇게 완성된 하나의 SEQ 는 이제 다음 파트인 Similarity Comparing 파트로 넘어간다.

3-3. Similarity Comparing (유사성 비교)

비슷한 이벤트 패턴을 탐지하기 위해서는 적절한 유사성 매트릭스가 필요하다. [2] 본 논문에서는 유사성 비교 방법에 대한 자세한 사항은 다루지 않고 있다. 다만 참조논문 [2]을 통하여 해당하는 방법만 제시할 뿐이다.

여기서 위의 그림 4 를 다시 한번 살펴보도록 하자. Event 밑의 BID 는 해당 비콘이 존재하는 구역을 나타내는 기호들로 대체되어 표시되어 있다. 즉, SEQ 에서 BID 부분만 뽑아 일렬로 세우면 소비자가 실제로 움직이고 쇼핑을 한 SEQ 가 나오게 된다. 해당 SEQ 를 기존에 BID 와 TIME 모두 포함한 SEQ 와 구별짓기 위해 SEQ_BID 라고 이 파트에 한정해 부르도록 하겠다.

그럼, 가게의 주인이 제시한 이상적인 SEQ 와 실제 소비자의 SEQ_BID 의 일치 정도를 비교해 보자. 둘의 일치 정도는 상호 정보와 엔트로피의 감소 (mutual information and reduction in entropy)에 기초한 정보-이론적 방법 (Information-Theoretic Measures)을 사용하여 확률적으로 계산하여 구할 수 있다. [2] 이러한 방법을 이용하여 신규 패턴을 검출하거나 혹은 예기치 못한 변화에 대응해 변환한, 이미 존재하는 패턴의 변화를 추적하는 것도 가능하지만 [2] 유사성 비교의 차원에서는 뒤의 두 기능은 필요치 않아 다루지 않도록 하겠다.

이렇게 검사한 유사성의 정도를 수치화 하여 해당 SEQ_BID 의 상위 SEQ (BID, TIME 정보가 포함된 SEQ)와 함께 다음 파트인 Scoring 에 넘기면 이 파트의 일은 끝난다.

3-4. Scoring (점수 매기기)

Scoring 에서는 3 가지의 점수를 매기는데, 그 중 첫 번째는 가장 소비자들이 오래 머물렀던 구역의 순위를 정하는 것이고 두 번째는 한 구역에서 다른 구역으로 넘어가는 경우의 수를 모두 비교해, 가장 많은 경우의 수로 이루어진 새로운 SEQ 를 제시하는 것이다. 마지막으로 세 번째는 들어오는 SEQ 들과 Similarity 들을 들어오는 순서대로 비교하여 가장 Similarity 가 높

은 실제 소비자들의 경로를 도출해 내는 것이다.

첫 번째인 ‘가장 소비자들이 오래 머물렀던 구역의 순위’ 먼저 점수를 매겨 보도록 하자. 우선 그 전에 미리 알아두어야 할 사항이 하나 있다. Scoring 파트가 이전 파트에게서 받은 정보는 {SEQ, Similarity} 와 같은데, 여기서의 SEQ 는 자신의 속성으로서 TIME 과 Beacon ID 를 가지고 있다. 그러나 이 속성들 중 TIME 은 앞의 표 1 에서 보여진 TIEM 과는 조금 다르다. 표 1 의 TIME 이 단말기가 서버에게 정보를 보낼 때의 현재 시간을 의미했다면, 현재 SEQ 의 TIME 은 소비자가 매장 안의 특정 구역에 머물렀던 총 시간을 의미한다. 즉, 아래 표와 같이 의미가 변형되었다.

α	단말기 자체에서 생성된 고유 ID
α .BID	Beacon ID
α .TIME	현재 시간 \rightarrow 특정 구역에 소비자가 머물렀던 시간

<표 2> 각 요소들의 기호화

그렇기에 α .TIME 들을 타임 매트릭스의 각 해당하는 구역에 차례로 집어 넣어 그 수를 모두 더한다면, 첫 번째 Scoring 의 목표를 달성할 수가 있다. 그러한 타임 매트릭스의 초기 모습은 아래와 같다.

A	B	C	D	E	F	G	H	I
0	0	0	0	0	0	0	0	0

<표 3-1> 타임 매트릭스 초기

보다시피 각 구역에 해당하는 값들은 모두 0 으로 초기화 되어있다. 여기에 위의 그림 4 를 참조하여 α SEQ 의 TIME 값을 집어 넣으면 아래와 같이 변하게 된다.

A	B	C	D	E	F	G	H	I
$T_3 - T_1$	T_5	$T_6 + (T_4 - T_1)$	0	0	0	0	0	0

<표 3-2> 타임 매트릭스에 SEQ α 의 시간 값을 넣은 후

여기다가 만약 β SEQ 가 들어와서 β 의 TIME 값도 집어넣어 주게 되면 다음과 같이 표가 변한다.

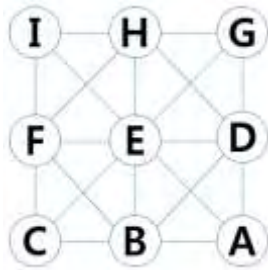
A	B	C	D	E	F	G	H	I
$(T_3 - T_1) + (T_4 + T_7)$	T_5	$T_6 + (T_4 - T_1)$	T_8	0	0	0	0	0

<표 3-3> 타임 매트릭스에 SEQ α 와 β 의 시간 값을 넣은 후

이러 방식으로 진행해서 마지막에 표가 모두 완성되면, A~I 구역의 최종 시간들을 정렬하여 소비자들이 제일 오랫동안 머물렀던 곳을 구할 수 있다.

그럼 이제 두 번째로 ‘새로운 SEQ 를 제시’를 해보도록 하자. 실제 현실에서 소비자들은 바로 옆 구역으로 이동하는 경우뿐만 아니라, 그보다 좀 더 먼 곳으로 이동해서 물건을 살피는 경우도 있을 것이다. 또한 갔었던 곳에 다시 갔다 오고 새로운 곳으로 가기도 하는 등, 소비자들이 완전히 한쪽에서 정해진

다른 한쪽 방향으로만 소비자들이 향하는 것이 아니다. 때문에 최다 방문 경로를 제시 하기 위해 그림 1을 그래프화 시키면, 방향성 없는 화살표들이 노드로 대체된 구역들을 서로 연결시켜 줘야만 한다. 해서, 위의 조건들을 모두 만족하는 그림 1의 그래프는 아래와 같다.



(그림 5) 그림 1의 그래프화

여기서 만약 A→B가 SEQ 안에 존재하면 A에서 B로 가는 경로에 1의 가중치를 준다. A→E→I의 경우에는 A→E와 E→I 둘 모두에 가중치 1을 주는 방식으로 간다. 좀 더 가시적으로 내용을 정리해 보기 위해, 그림 4의 예를 살펴해보도록 하자. α의 SEQ는 섹션 3-2의 SEQ 제조 과정을 거친 후, 아래와 같은 모습으로 변환되었다.

SEQ[4]

SEQ 중에서 BID만을 뽑아 리스트를 만들어 주면,

SEQ[0].BID → SEQ[1].BID → SEQ[2].BID → SEQ[3].BID

가 되고, 이것을 기호로 변환시켜 보면,

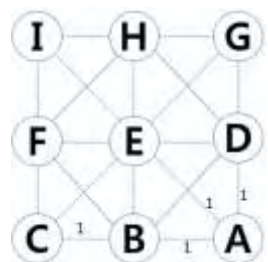
Exit ID → A → B → C

가 완성된다.

마찬가지로 β의 경우를 살펴보면 최종적으로 아래와 같은 결과가 도출된다.

Exit ID → A → D

위의 두 결과들을 그림 5의 그래프에 Exit ID를 제외시키고 나머지를 가중치로서 적용시킨 후의 모습은 다음과 같다.



(그림 6) 가중치를 적용시킨 그래프

이러한 방식으로 모든 SEQ의 경로가 가중치로 전환되고 난 후에 가장 가중치가 높은 경로들을 선으로 이으면 추천 경로가 완성된다. 그림 4의 경우 예시로 보여주기 위한 짧은 SEQ들 밖에 나오지 않아서 추천 경로를 만들어 내기에는 모자라지만, 실제로 적용시켜 더 길고 많은 SEQ들이 나와 모두 적용 시키면 충분히 추천 경로로서의 경로가 도출될 것이라고 생각된다.

마지막으로 세번째 Scoring인 ‘매장의 주인이 생각한 SEQ와 가장 Similarity가 높은 실제 소비자들의 경로’를 도출하는 것이 남았다. 이 부분은 사실 이전 파트인 Similarity Comparing에서 가장 필요한 부분인 유사성 수치화를 해주었기 때문에 어려운 점은 없다. 그저 새로운 SEQ가 들어오면 들어오는 대로 유사성 수치의 대소를 비교해 주고 작으면 데이터베이스의 것과 새로운 것을 맞바꾸고 크면 그대로 놔두는 방식으로 진행하면 된다.

4. 결론 및 향후 계획

이번 논문에서는 소비자의 동선 파악과 각 구역에 머문 시간을 계산하여 가장 인기가 많았던 구역과 이동 순서를 특정지어 보았다. 이렇게 함으로서 매장의 주인들이 자신들의 생각한 소비자들의 동선과 실제 동선의 매치 정도를 비교해 볼 수 있었다. 또한, 앞으로 매장의 주인들이 전보다 구체적이고 정확한 소비자 동선을 알아내어 그대로 응용해 볼 수도 있으며, 특히나 특정 구역에서 머문 시간을 획득하여 어느 상품이 소비자들의 관심은 많이 끌었으나 매출은 좋지 않았는지, 또 매출은 좋았으나 많은 이들의 관심을 끌어당기지 못하는지를 파악할 수 있다. 이러한 결과들을 토대로 적절한 대응 방안도 마련할 수 있어 매장의 부흥에 기여할 수 있다.

미래의 계획으로는 우선 비콘의 수신 범위 겹침으로부터 정확하게 구역을 구분해 낼 방법을 찾는 것과 SEQ의 유사성을 구하는 방법의 구체화, 그리고 마지막으로 실질적인 구현이 있다.

참고문헌

[1] Eugene Wu, Yanlei Diao, Shariq Rizvi, *High-performance complex event processing over streams*, SIGMOD 2006, June 27-29, 2006, Chicago, Illinois, USA
 [2] Samujjwal Bhandari, Susan D. Urban, Mohan Sridharan. *An Adaptive Event Stream Processing Environment*, SIGMOD/PODS'12 PhD Symposium, 2012