



블록 물류 관리를 위한 트랜스포터와 작업 블록 자동 매칭 알고리즘 연구

송진호¹ · 박광필^{2,†} · 옥진성³
충남대학교 선박해양공학과^{1,2}
대우조선해양(주) 산업기술연구소³

A Study on the Automatic Matching Algorithm of Transporter and Working Block for Block Logistics Management

Jin-Ho Song¹ · Kwang-Phil Park^{2,†} · Jin-Sung Ok³
Dept. of Naval Architecture & Ocean Engineering, Chungnam National University^{1,2}
Industrial Application R&D Institute, Daewoo Ship & Marine Engineering³

This is an Open-Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

During the shipbuilding process, many blocks are moved between shipyard workshops by block carrying vehicles called a transporter. Because block logistics management is one of the essential factors in enhancing productivity, it is necessary to manage block information with the transporter that moves it. Currently, because a large amount of data per day are collected from sensors attached to blocks and transporters via IoT infrastructure installed in shipyards, automated methods are needed to analyze them. Therefore, in this study, we developed an algorithm that can automatically match the transporter and the working block based on the GPS sensor data. By comparing the distance between the transporter and the blocks calculated from the Haversine formula, the block is found which is moved by the transporter. In this process, since the time of the measured data of moving objects is different, the time standard for calculating the distance must be determined. The developed algorithm was verified using actual data provided by the shipyard, and the correct result was confirmed with the distance based on the moving time of the transporter.

Keywords : Ship block(선박 블록), Transporter(트랜스포터), GPS(Global Positioning System), Automatic matching algorithm(자동 매칭 알고리즘)

1. 서 론

1.1 연구 배경 및 필요성

조선업에서 생산성 향상은 최근 가장 핵심적인 이슈로서, 생산 및 물류 관리에 소요되는 비용과 시간을 줄이고 경제력을 제고하기 위한 노력을 하고 있다(Nam et al., 2017). 이를 위해 물류 부분에서는 선박 블록의 상황을 모니터링하고 관리할 필요가 있다(Mun et al., 2019). 조선소에서 선박 블록의 이동을 위해 Fig. 1과 같이 트랜스포터가 사용되며, 선박 생산의 최적화를 통한 생산기간 및 건조 비용 절감을 위해서 트랜스포터와 블록의 위치 모니터링 및 관리가 필요하다(Lee et al., 2008). 현업에서는 이를 위해 트랜스포터 운전자가 소지한 모바일로 직접 작업

블록 정보를 입력하는 방법을 사용한다(Shin et al., 2006, Mun et al., 2019). 이때, 운전자가 입력을 하지 않거나 작업장에서 벗어난 채 모바일을 운용하면 블록의 이동을 정확하게 파악하기 어렵다. 또한 트랜스포터 및 블록에 부착된 센서를 통해 하루에



Fig. 1 Transporter and working block (source : DSME youtube channel)

수십만 개의 데이터가 수집되므로 이것을 수동으로 관리하기는 불가능하다.

따라서 본 연구에서는 이와 같은 문제를 해결하고 트랜스포터와 블록의 위치 모니터링 및 관리를 위해 트랜스포터 운전자가 직접 블록의 이동 상황을 입력하는 것이 아닌 트랜스포터와 블록에 부착된 센서에서 수집되는 데이터들을 활용하여 트랜스포터가 작업한 블록을 자동으로 매칭할 수 있는 알고리즘을 개발하였다.

1.2 연구 방법

본 연구의 목적은 트랜스포터가 이동한 작업 블록을 자동으로 매칭할 수 있는 알고리즘을 개발하는 것으로, 트랜스포터와 블록의 GPS 데이터를 이용해 계산된 트랜스포터와 블록들 사이의 거리를 활용하였다. 측정된 거리들의 평균값을 계산하고 최솟값을 이용해서 트랜스포터와 작업 블록을 매칭하였다.

개발한 알고리즘에 실제 조선소 현장의 데이터를 활용하여 매칭 결과를 통해 효과를 검증하였다.

본 논문은 총 5장으로 구성되어 있으며, 1장에서는 본 연구의 배경 및 필요성을 정리하였고 연구 방법을 정리하였다. 2장에서는 트랜스포터와 블록의 모니터링 및 관리와 관련된 연구들을 정리하였다. 3장에서는 트랜스포터 작업 블록 자동 매칭 알고리즘을 개발 내용을 기술하였고, 4장에서는 개발한 알고리즘을 실제 조선소 현장의 데이터를 이용해서 구현 내용을 기술하였다. 마지막 5장에서는 본 연구의 결론을 정리하였다.

2. 관련 연구

트랜스포터와 블록의 모니터링 및 관리를 위한 연구는 다음과 같다.

Shin et al. (2006)은 PDA (Personal Digital Assistants)와 GPS 데이터를 이용해서 블록을 관리하였다. 작업자의 입력에 대한 오류를 위치 보정 시스템을 이용하고 블록이 있는 지역과 입력된 위치를 분석하여 블록이 적절하게 이동이 되었는지 판단하였다.

Lee et al. (2008)은 트랜스포터의 위치를 모니터링하기 위해 GPS 센서와 IMU (Inertial Measurement Unit) 센서를 이용해서 GPS 위치를 개선시켜 트랜스포터의 위치를 추적하였다.

Shin et al. (2010)은 트랜스포터 위치 추적을 위해 RFID (Radio Frequency Identification)를 도입하는 방안을 제시하였다. 제시한 방안은 조선소 내 지번 및 도로 하부에 정해진 위치에 RFID를 설치하고 트랜스포터가 도로를 이동할 때 해당 지번의 RFID를 읽을 수 있는 리더기를 트랜스포터에 설치하여 위치를 추적하였다.

Mun et al. (2019)는 RFID 방식은 초기에 많은 투자비가 들고 유지 관리하기 위한 비용이 많이 든다는 한계를 지적하고, GNSS (Global Navigation Satellite System)과 IMU센서를 융합하여 트

랜스포터의 위치를 추적할 수 있는 시스템을 개발하였다. 또한 계산된 위치정보에서 신호 잡음이 많은 센서 신호에 칼만 필터 (Kalman filter)를 사용하여 신호 잡음을 제거하고 블록 이동 정보를 사용자가 보다 쉽게 입력할 수 있는 시스템을 개발하였다.

위 연구들은 트랜스포터의 위치를 정확하게 추종하고 사용자가 그 위치에서 블록의 위치를 입력하여 트랜스포터와 블록의 위치를 관리하는 방법으로, 트랜스포터와 작업 블록을 자동으로 매칭하여 관리를 하지 못하기 때문에 사용자의 부주의에 의한 오류가 발생할 우려가 있다.

본 연구에서 개발한 알고리즘에서 트랜스포터와 작업 블록을 자동으로 매칭시키는 방법과 관련된 연구는 다음과 같다.

Choi et al. (2004)는 도로와 컨테이너의 GPS 위치 간의 거리를 계산하여 컨테이너가 이동 중인 도로와 매칭시키는 방안을 제시하였다.

Kim et al. (2010)은 열차의 GPS 데이터를 이용해서 선로와 열차 GPS 위치 간의 거리를 유클리드 거리를 이용해 계산하여 선로 지도에 맵 매칭을 할 수 있는 방법을 제시하였다.

Saki et al. (2022)은 스마트폰, 자동차와 같이 도로에서 계속된 움직이는 물체들의 GPS 데이터와 도로를 맵 매칭시킬 수 있는 맵 매칭 open-source 엔진들을 정리하였다.

그러나 위 연구들은 고정된 물체와 컨테이너 및 열차와 같이 이동하는 물체를 매칭한 것으로, 이동하는 물체의 이동 시간동안 두 물체의 거리를 측정하는 반면에, 트랜스포터의 블록 이동은 두 물체가 모두 이동하는 동안의 GPS 데이터를 이용하기 때문에 계속되는 데이터의 시간이 다를 수 있다. 즉, 거리를 측정할 시간 기준을 정해야하는 문제가 있다.

따라서 본 논문에서는 위 연구에서 사용한 GPS 데이터를 이용한 맵 매칭 방법을 개선시켜 관련 연구들의 한계를 극복하기 위해 움직이는 두 물체를 매칭시키는 트랜스포터 작업 블록 자동 매칭 알고리즘을 제시하였다.

3. 트랜스포터 작업 블록 자동 매칭 알고리즘

3.1 트랜스포터 및 작업 블록 데이터

트랜스포터 데이터는 트랜스포터의 ID, 데이터 취득 시간, GPS 데이터를 사용하였고 블록의 상·하차를 판단할 수 있는 유압 세기 데이터를 사용하였다. 블록 데이터는 트랜스포터와 마찬가지로 ID, 데이터 취득 시간, GPS 데이터를 사용하였고 블록의 이동을 감지하는 진동 데이터를 사용하여 블록이 움직인 시간을 판단하였다. 이때, 트랜스포터 데이터는 이동 시 1초마다 수신되어 하루에 약 90만 개의 데이터가 생성이 되고 이동이 감지된 블록 데이터는 10초 간격으로 하루에 약 5만 개의 데이터가 생성이 된다.

3.2 트랜스포터 상·하차 시간 추출

트랜스포터의 상·하차 작업 시간은 트랜스포터에서 계측된 유압 세기 데이터를 이용해서 트랜스포터의 작업 중량(weight)을 계산하고 Fig. 2와 같이 계산된 중량 변화를 판단하는 방법을 사용하였다. 트랜스포터의 작업 중량이 Fig. 2(a)과 같은 경우 중량이 증가한 부분이 트랜스포터가 상차 작업을 시작한 것으로 판단할 수 있다. 그리고 중량이 감소하여 0보다 아래 값을 가지면 하차 작업을 완료한 것으로 판단할 수 있다. 즉, 중량이 증가하고 감소한 사이의 시간이 트랜스포터가 블록을 운송하고 있는 것으로 판단할 수 있다. Fig. 2(b)는 중량 데이터의 잡음을 제거하기 위해 저주파 통과 필터(low pass filter)를 사용하여 잡음을 제거한 것이다. Fig. 2(c)은 잡음이 처리된 데이터를 미분하여 중량 데이터에서 정점(peak)들을 구한 것이다. 여기서 정점들은 트랜스포터에 블록이 상차가 시작되거나 하차가 완료되는 지점이 된다. Fig. 2(d)와 같이 정점들을 이용해서 상차 및 하차 영역으로 나누게 되면 트랜스포터의 운용 시간 중에서 블록을 이동시킨 시간을 추출할 수 있다.

3.3 작업 블록 데이터 추출

추출된 트랜스포터의 상·하차 작업을 한 시간 동안에 블록의

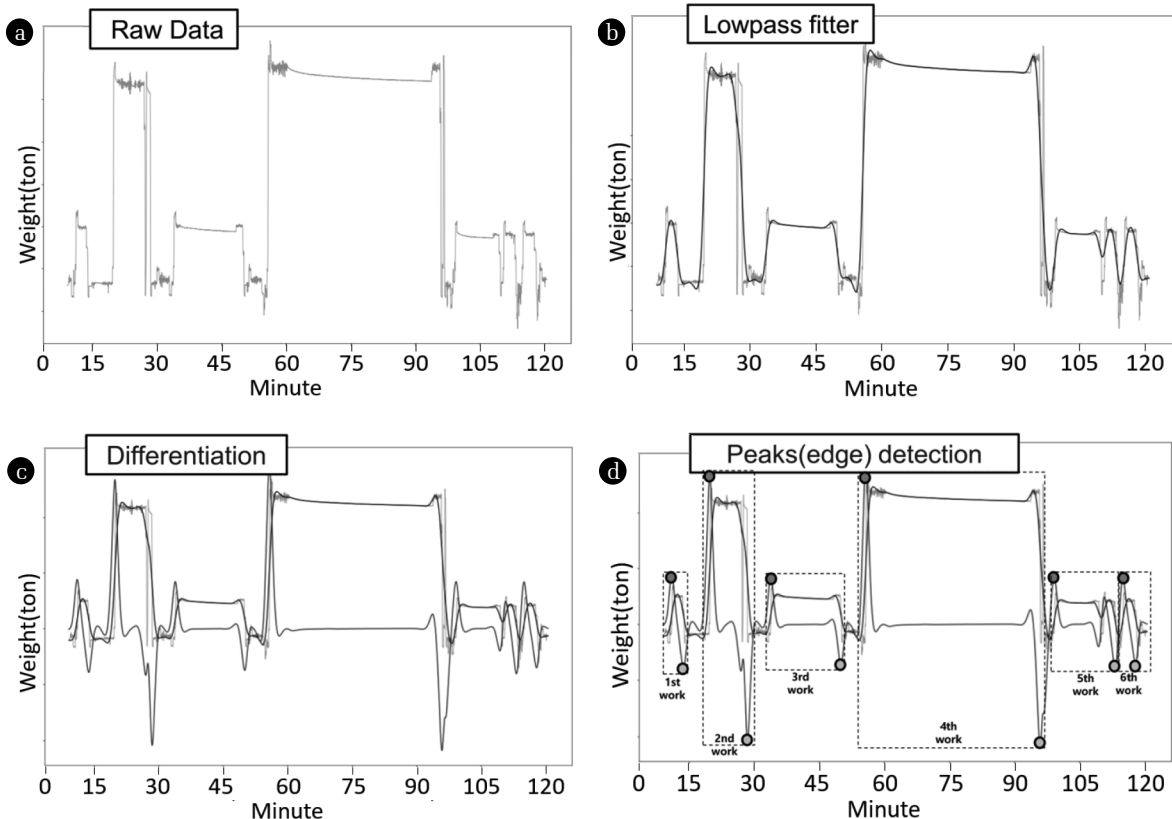


Fig. 2 Extracting transporter loading and unloading time

진동 데이터를 이용해서 이동이 감지된 블록 데이터를 추출할 수 있다.

Fig. 3과 같이 시간에 따른 트랜스포터의 작업 중량이 변화된 사례에서 5번째 블록 운송 작업 시간 동안 이동이 감지된 블록을 찾으면 Table 1과 같이 총 4개의 후보 블록을 찾을 수 있다.

이때, Table 1에 나타난 데이터는 트랜스포터 ID(tp_id), 트랜스포터의 블록 운송 작업 순서(time_order), 블록의 이동 시작 시간(r_time), 블록의 정지 시작 시간(f_time), 트랜스포터 총 운용시간(diff_time), 블록 아이디(blk_id)로 구성되어 있다. 그림에서는 트랜스포터, 블록명, 시간 데이터는 가려진 상태로 표시하였다.

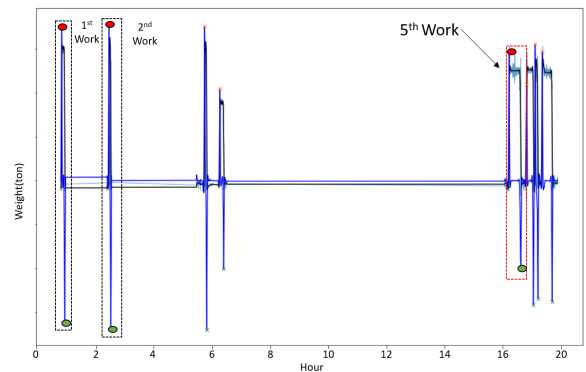


Fig. 3 Transporter working weight data according to time

Table 1 Candidate blocks for the transporter 5th work

tp_id	time_order	r_time	f_time	diff_time (s)	blk_id
TP**1	5	Y-M-D-T1	Y-M-D-T2	1461.0	B**1
TP**2	5	Y-M-D-T3	Y-M-D-T4	1461.0	B**2
TP**3	5	Y-M-D-T5	Y-M-D-T6	1461.0	B**3
TP**4	5	Y-M-D-T7	Y-M-D-T8	1461.0	B**4

3.4 트랜스포터와 작업 블록 매칭 알고리즘

트랜스포터 GPS 데이터와 트랜스포터가 블록을 상·하차하는 사이 시간 동안 이동이 감지된 블록들의 GPS 데이터들을 이용해서 트랜스포터와 해당 작업 블록을 매칭하였다. 매칭 방법은 트랜스포터와 블록들의 GPS 데이터를 이용해서 측정된 시간마다 트랜스포터와 블록 사이의 거리를 구하고 거리의 평균값이 가장 작은 블록이 트랜스포터와 가장 가까이 있다고 판단하여 해당 블록을 트랜스포터의 작업 블록으로 매칭하였다.

3.5 트랜스포터와 블록 GPS 데이터를 이용한 거리 측정

트랜스포터와 이동이 감지된 블록들의 GPS 데이터를 이용한 거리 측정은 Haversine formula를 이용해서 측정하였다.

Haversine formula는 Fig. 4와 같이 지구를 구(sphere)로 가정하여 두 지점의 좌표(위도(latitude), 경도(longitude))를 이용하여 지구 표면을 따라 측정한 거리, 즉 대원 거리(great-circle distance)를 구하는데 활용된다 (Yang et al., 2019).

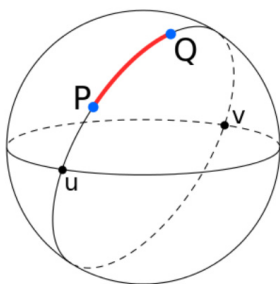


Fig. 4 A diagram illustrating great-circle distance (Yang et al., 2019)

여기서, 점 P와 Q의 위도와 경도를 각각 $lat_P, lot_P, lat_Q, lot_Q$ 라고 하고 구의 반지름을 r 이라고 하면, Haversine formula를 이용하여 두 지점 사이의 거리 d 는 식 (1)-(3)과 같이 구할 수 있다.

$$d = 2r \cdot \arcsin(\sqrt{\sin^2 A + \cos(lat_P)\cos(lat_Q)\sin^2 B}) \quad (1)$$

$$A = \frac{lat_P - lat_Q}{2} \quad (2)$$

$$B = \frac{lot_P - lot_Q}{2} \quad (3)$$

본 논문에서는 작업하는 트랜스포터와 작업하는 동안 이동이 감지된 블록의 GPS 데이터를 이용하는데, 두 물체의 계측되는 데이터의 시간이 다를 수 있다. 예를 들어, Fig. 5와 같이 트랜스포터가 13시부터 13시 20분까지 블록 운송 작업을 했고 전체 블록 중 이 시간 동안 이동이 감지된 블록이 3개가 있으며 후보 2번 블록이 운송 작업된 블록이라고 가정한다. 이때, 후보 1번 블록의 이동이 감지된 시간은 13시 10분부터 13시 15분이고 후보 2번 블록은 트랜스포터와 동일하게 13시부터 13시 20분까지 이동하였다. 마지막 후보 3번 블록의 이동이 감지된 시간은 13시 05분부터 13시 15분까지로 이동하였다. 즉, 이와 같이 트랜스포터의 블록 운송 작업 시간과 이동이 감지된 블록들의 이동 시간이 다를 수 있다. 이렇게 이동 시간이 다르게 되면 트랜스포터와 후보 블록들 사이의 평균 거리를 계산하기 위해선 시간 기준을 고려해야 한다.

먼저, 후보 블록들의 이동한 시간을 기준으로 트랜스포터와 평균 거리를 계산할 수 있다. 즉, Fig. 5에서 후보 1번 블록과 트랜스포터 사이의 평균 거리를 계산한다고 하면 후보 1번 블록이 이동한 시간인 13시 10분부터 13시 15분 사이의 GPS 데이터를 이용해서 평균 거리를 계산할 수 있다. 이때, 우연히 트랜스포터와 후보 1번 블록이 가까이 있으면 평균 거리가 작게 나와서 후보 블록 중에 최솟값인 경우가 발생할 수 있다. 따라서 후보 블록들의 이동한 시간을 기준으로 평균 거리를 계산하면 트랜스포터 작업 블록을 잘못 매칭하는 경우가 발생할 수 있다.

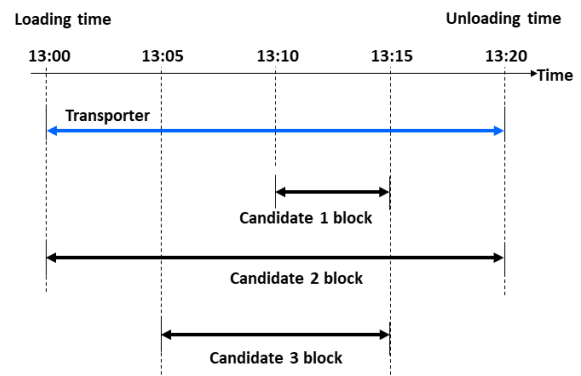


Fig. 5 Candidate blocks moved during transporter working time

이러한 오류가 발생하지 않기 위해선 트랜스포터가 이동한 시간을 기준으로 후보 블록들과 평균 거리를 계산하여야 한다. 즉, 트랜스포터가 이동한 시간인 13시부터 13시 20분 사이의 GPS 데이터를 이용해서 평균 거리를 계산하여야 한다. 이때, Fig. 6에

서 후보 1번 블록은 13시부터 13시 10분, 13시 15분부터 13시 20분까지의 GPS 데이터가 존재하지 않는다. 이러한 문제를 해결하기 위해 Fig. 6과 같이 후보 블록이 움직이기 시작한 시간 이전과 끝난 시간 이후의 데이터가 필요하다. 블록이 이동 전에는 이동 시작하는 위치에, 이동 후에는 이동 후 위치에 각각 정지한 상태로 있다고 판단할 수 있기 때문에, 이 시간들에 해당 위치의 GPS 데이터를 복사하였다. 그러면 트랜스포터는 이동하기 때문에 후보 1번 블록과의 평균 거리는 블록 이동 시간 기준으로 계산한 것보다 증가할 것이다. 따라서 앞서 발생한 한계를 보완할 수 있기 때문에 본 논문의 알고리즘은 트랜스포터 시간을 기준으로 평균 거리를 계산해서 매칭을 하였다.

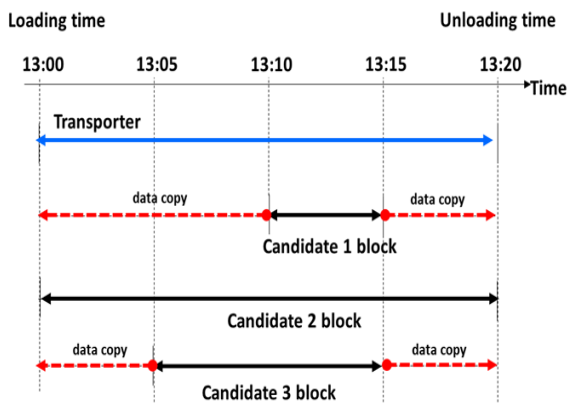


Fig. 6 Copying candidate block data

3.6 알고리즘 순서도

개발한 트랜스포터 작업 블록 자동 매칭 알고리즘의 순서도를 나타내면 Fig. 7과 같다. 트랜스포터 유압 데이터를 이용해서 트랜스포터의 상·하차 작업 시간을 계산하고 이 시간 동안 움직임이 감지된 후보 블록들의 데이터를 추출한다. 그리고 추출된 트

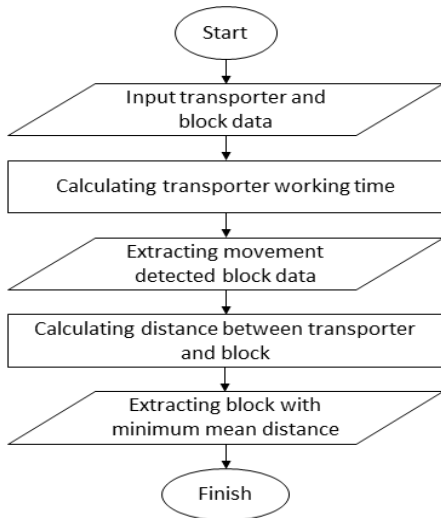


Fig. 7 Transporter working block automatic matching algorithm flowchart

랜스포터와 후보 블록들의 데이터 중 GPS 데이터를 이용해 트랜스포터와 블록들 사이의 거리를 계산한다. 계산한 거리의 평균값을 계산하고 최솟값을 이용해서 트랜스포터와 이때 작업된 블록을 매칭한다.

4. 매칭 알고리즘 검증

알고리즘을 구현한 라이브러리는 Python을 이용해 구조화된 데이터를 만들고 조작하기 위한 라이브러리로 데이터를 분석할 때 주로 사용하는 Pandas를 사용하였다 (Matt and Theodore, 2020).

개발한 트랜스포터 작업 블록 자동 매칭 알고리즘의 결과와 효과를 검증하기 위해 실제 조선소 현장의 데이터를 이용해서 2개의 Case를 구현하였다.

4.1 Case 1

Table 2는 1번 트랜스포터가 24분 21초 동안 상·하차 작업을 할 때 이동이 감지된 4개의 후보 블록을 나타낸 것이다. Fig. 8은 트랜스포터와 후보 블록들의 GPS 데이터들을 이용해서 궤적을 나타낸 것이다. x축은 경도, y축은 위도를 나타내었다. 궤적 그래프를 보면 후보 2번의 블록이 트랜스포터의 궤적과 거의 일치한 것을 확인할 수 있다. 따라서 트랜스포터가 작업한 블록은 후보 2번의 블록이라는 것을 알 수 있다.

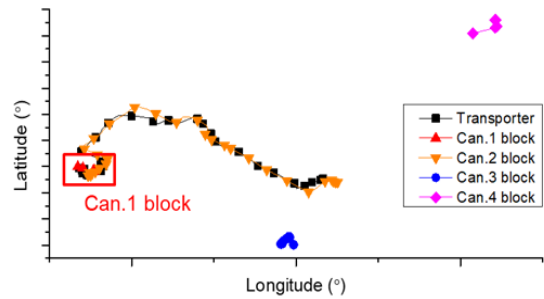


Fig. 8 Trajectory of movement of transporter and candidate blocks

개발한 알고리즘이 후보 2번 블록을 작업 블록으로 추출할 수 있는지 확인하기 위해 트랜스포터와 후보 블록들의 궤적을 Fig. 9, Fig. 10으로 정리하였고 각 후보 블록들의 평균 거리를 Table 3, Table 4로 정리하였다. Fig. 9와 Table 3은 후보 블록들의 시간을 기준으로 정리한 것이고 Fig. 10과 Table 4는 트랜스포터 시간을 기준으로 정리한 것이다.

후보 블록들의 시간을 기준으로 했을 때 트랜스포터와 후보 1번 블록의 평균 거리는 71.3 m이고 후보 2번 블록은 105.9 m, 후보 3 블록은 620 m, 후보 4번 블록은 1678.4 m의 결과를 얻었다. 후보 1번 블록의 평균 거리가 최솟값으로 계산되지만, 이것은 후보 1번 블록의 움직임이 감지될 때 트랜스포터와 가까이

Table 2 Case 1 candidate blocks

Transporter id	Working time	Matching block id	Block movement time	Number of data (ea)	
				Transporter	Block
1	24m 21s	Candidate 1 block	6 m 25s	1462	38
		Candidate 2 block	24 m 1 s		144
		Candidate 3 block	7 m 20s		14
		Candidate 4 block	4 m 20s		14

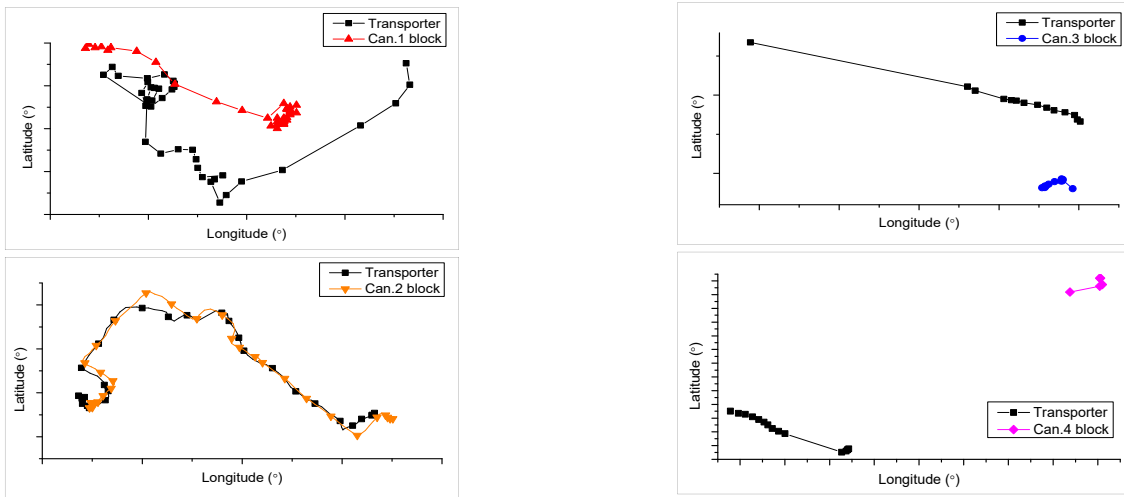


Fig. 9 Trajectory of movement of transporter and candidate blocks based on candidate blocks time (Case 1)

Table 3 Average distance calculated based on candidate blocks time (Case 1)

Candidate block	1	2	3	4
Measurement time (number of date)	6 m 25 s (38 ea)	24 m 1 s (144 ea)	7 m 20 s (14 ea)	4 m 20 s (14 ea)
Average distance(m)	71.3 (Minimum mean distance)	105.9	620.0	1678.4

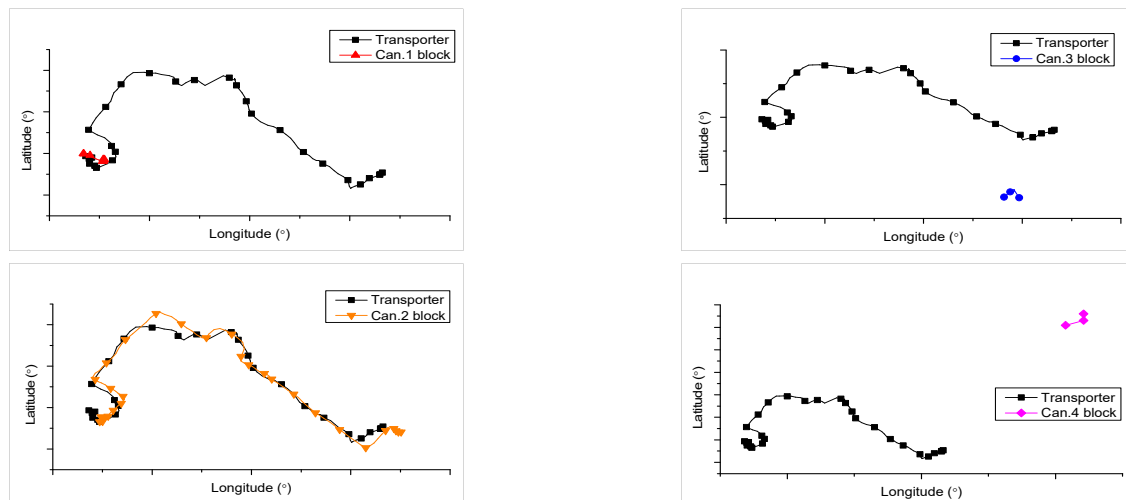


Fig. 10 Trajectory of movement of transporter and candidate blocks based on transporter time (Case 1)

Table 4 Average distance calculated based on transporter time (Case 1)

Candidate block	1	2	3	4
Measurement time (number of date)	24 m 21 s (1463 ea)	24 m 21 s (1463 ea)	24 m 21 s (1463 ea)	24 m 21 s (1463 ea)
Average distance (m)	609.3	99.3 (Minimum mean distance)	1060.2	2013.2

있어서 평균 거리가 최소인 결과가 나온 것으로, 후보 블록들의 시간을 기준으로 했을 때 오류가 발생함을 알 수 있다.

트랜스포터 시간을 기준으로 했을 때 트랜스포터와 후보 1번의 평균 거리는 609.3 m, 후보 2번의 블록은 99.3 m, 후보 3번의 블록은 1060.2 m, 후보 4번의 블록은 2013.2 m의 결과가 나타났다. 블록들의 시간을 기준으로 평균 거리를 계산했을 때 보다 후보 1번 블록을 포함한 후보 3, 4번 블록의 평균 거리가 크게 증가한 것을 확인할 수 있고 후보 2번의 블록의 평균 거리는 감소하였다. 따라서 후보 2번 블록의 평균 거리가 99.3 m로 최솟값이 됨으로 트랜스포터의 작업 블록은 후보 2번 블록으로 올바르게 매칭되는 것을 확인할 수 있다.

4.2 Case 2

Table 5는 2번 트랜스포터가 34분 38초 동안 상·하차 작업을 할 때 이동이 감지된 후보 블록을 나타낸 것이고, Fig. 11은 트랜스포터 및 후보 블록들의 궤적을 나타낸 것이다. 그래프에서 트랜스포터와 궤적이 유사한 후보 2번 블록이 작업 블록임을 알 수 있다.

Table 5 Case 2 candidate blocks

Transporter id	Working time	Matching block id	Block movement time	Number of data (ea)	
				Transporter	Block
2	34 m 38 s	Candidate 1 block	10 m 11 s	2080	16
		Candidate 2 block	31 m 33 s		137
		Candidate 3 block	3 m 50 s		21
		Candidate 4 block	10 m 41 s		61

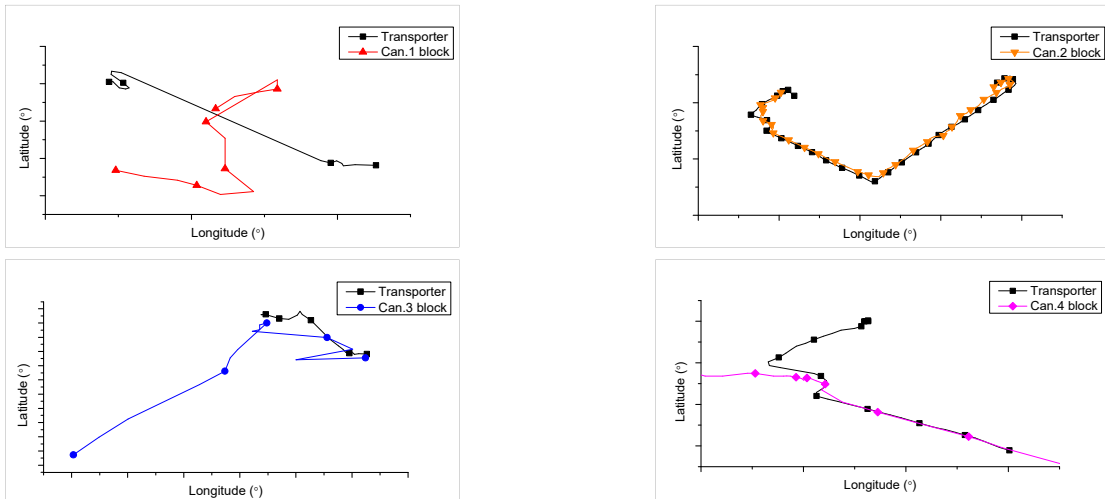


Fig. 12 Trajectory of movement of transporter and candidate blocks based on candidate blocks time (Case 1)

개발한 알고리즘이 후보 2번 블록을 작업 블록으로 추출할 수 있는지 확인하기 위해 트랜스포터와 후보 블록들의 궤적을 Fig. 12, Fig. 13으로 정리 하였고 각 후보 블록들의 평균 거리를 Table 6, Table 7로 정리 하였다. Fig. 12와 Table 6은 후보 블록들의 시간을 기준으로 정리한 것이고 Fig. 13과 Table 7은 트랜스포터 시간을 기준으로 정리한 것이다.

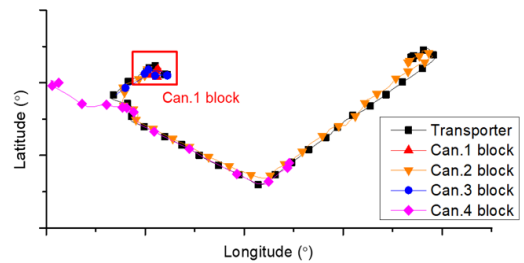


Fig. 11 Trajectory of movement of transporter and candidate blocks

결과를 분석하면, 블록 시간을 기준으로 매칭을 했을 때는 후보 1번 블록의 평균 거리는 55.36 m, 후보 2번 블록은 79.51 m,

Table 6 Average distance calculated based on transporter time (Case 2)

Candidate block	1	2	3	4
Measurement time (number of date)	10 m 11 s (16 ea)	31 m 33 s (137 ea)	3 m 50 s (21 ea)	10 m 41 s (61 ea)
Average distance (m)	55.36 (Minimum mean distance)	79.51	102.29	549.96

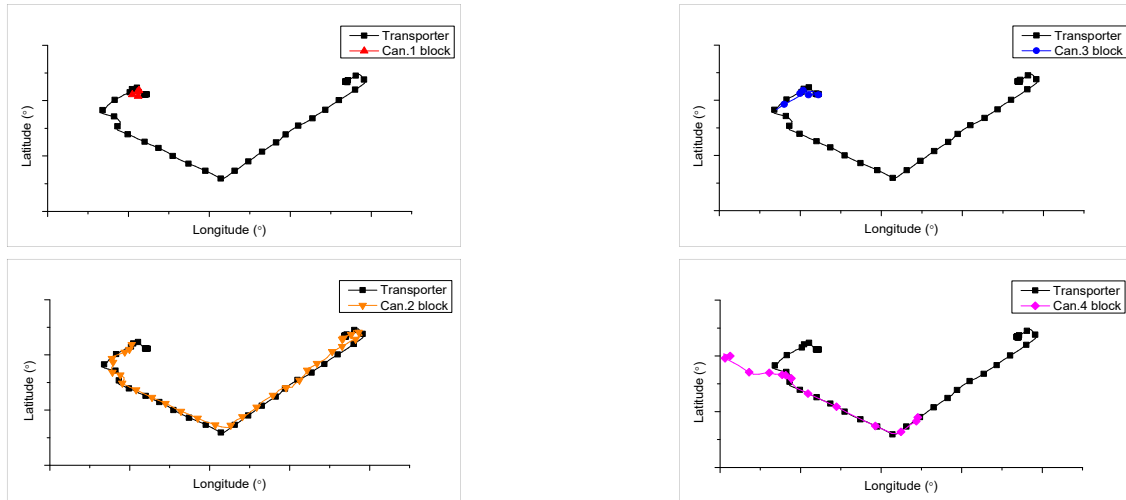


Fig. 13 Trajectory of movement of transporter and candidate blocks based on candidate blocks time (Case 2)

Table 7 Average distance calculated based on candidate blocks time (Case 2)

Candidate block	1	2	3	4
Measurement time (number of date)	34 m 38 s (1463ea)	34 m 38 s (1463 ea)	34 m 38 s (1463 ea)	34 m 38 s (1463 ea)
Average distance (m)	758.38	65.54 (Minimum mean distance)	414.67	870.85

후보 3번 블록은 102.29 m, 후보 4번의 블록은 549.96 m로 후보 1번 블록이 최솟값이기 때문에 후보 1번이 매칭이 되어 잘못된 작업 블록으로 매칭되는 것을 확인할 수 있다.

트랜스포터 시간을 기준으로 매칭 했을 때는 1번 블록의 평균 거리는 758.38 m, 후보 2번 블록은 65.54 m, 후보 3번 블록은 414.67 m, 후보 4번의 블록은 870.85 m로 후보 1, 3, 4번 블록의 평균 거리가 증가했고 최종적으로 후보 2번 블록의 거리 평균이 최솟값인 것을 확인할 수 있다. 따라서 개발한 알고리즘이 올바르게 트랜스포터 작업 블록을 매칭하는 것을 확인할 수 있다.

5. 결론

본 연구에서는 스마트 야드 인프라를 통해 수집되는 트랜스포터와 블록의 데이터를 이용하여 트랜스포터와 이를 이용해 이동한 작업 블록을 자동으로 매칭 알고리즘을 구현하였다. 개발한 알고리즘은 트랜스포터의 하중 데이터와 위치 데이터 및 블록에서 수집되는 위치 데이터들을 활용하여 평균 거리를 계산하고 최솟값을 이용해서 구현하였다. 이때, 거리를 계산할 때 블록의 움직임이 감

지된 시간을 기준으로 계산하면 오류가 발생하는 것을 확인하였고, 이를 방지하기 위해 트랜스포터의 작업 시간을 기준으로 계산이 되어야 한다는 것을 확인하였다. 실제 조선소 현장의 데이터를 사용하여 개발한 알고리즘을 적용하고 트랜스포터와 작업 블록이 올바르게 매칭되는 것을 확인하였다. 본 알고리즘은 현업에 적용되어 블록 물류 관리에 활용될 수 있을 것이라 기대된다.

본 연구는 작업 중에 실시간으로 수집되었지만, 작업이 완료된 후에 저장된 일정 기간을 대상으로 한 것이다. 따라서 실시간으로 데이터를 처리하여 트랜스포터와 작업 블록을 매칭하는 것에는 적용에 한계가 있다. 향후 연구로는 GPS 위치 정확도가 낮은 경우를 고려하는 것과 실시간으로 트랜스포터와 블록을 매칭할 수 있는 방안에 대한 연구가 필요하다.

후기

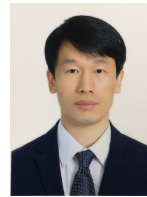
이 논문은 2022년도 정부 (산업통상자원부)의 재원으로 한국산업기술포진원의 지원을 받아 수행된 연구임 (P0001968, 2022년 산업혁신인재성장지원사업)

References

- Choi, B.G., Yun, H.C. and Yu, C.W., 2004. Construction of Container Management System Using GPS. *Journal of the Korean Society of Surveying, Geodesy, Photogrammetry and Cartography*, 22(1), pp.37-44.
- Kim, B.G., Lee, T.H., Hong, H.P. and Lee, S.B., 2010. A study on localization using GPS for train. *2010 Conference and Annual Meeting of the Korean Society for Railway*, Jeju, Republic of Korea, 28-30 October 2010.
- Lee, Y.H., Lee, K.C., Lee, K.J. and Son, Y.D., 2008. Study on the positioning system for logistics of ship-block. *Special Issue of the Society of Naval Architects of Korea*, pp.68-75.
- Matt, H. and Theodore, P., 2020. *Pandas 1.x Cookbook 2nd edition*. Packt.
- Mun, S.H., An, J.W. and Lee, J.M., 2019. Real time transporter locating system for shipyard through GNSS and IMU sensor. *Journal of the Society of Naval Architects of Korea*, 56(5), pp.439-446.
- Nam, B.W., Lee, K.H., Lee, J.J. and Mun, S.H., 2017. A study on selection of block stockyard applying decision tree learning algorithm. *Journal of the Society of Naval Architects of Korea*, 54(5), pp.421-429.
- Saki, S. and Hagen, T., 2022. A practical guide to an open-source map-matching approach for big GPS data. *SN Computer Science*, 3(5), pp.1-13.
- Shin, J.G. and Lee, J.H., 2006. Prototype of block tracing system for pre-erection area using PDA and GPS. *Journal of the Society of Naval Architects of Korea*, 43(1), pp.87-95.
- Shin, J.C., Shim, H.S., Kang, K.S. and Han, D.H., 2010. A study on the implementation of block transporter tracking management system by introducing RFID framework in the shipbuilding industry. *Korean Journal of Logistics*, 18(3), pp.69-83.
- Yang, I.C., Jeon, W.H. and Moon, J.P., 2019. A study on a distance based coordinate calculation method using Inverse Haversine Method. *Journal of Digital Contents Society*, 20(10), pp.2097-2102.



송진호



박광필



옥진성