

# 복잡한 환경에서 다중표적추적을 위한 고속 트랙병합 기법

## A Method of Fast Track Merging for Multi-Target Tracking under Heavy Clutter Environment

이 승 연\*                  윤 주 흥\*                  이 석 재\*\*                  정 영 현\*                  최 덕 선\*

Seung-Youn Lee          Joo-Hong Yoon          Seok-Jae Lee          Young-Hun Jung          Tok-Son Choe

### Abstract

In this paper, we proposed a method of fast track merging which is the foundation of track to track association technique. The existing method of track merging is performed throughout comparison between tracks to tracks. Therefore, it has heavy calculation time. In our research, we developed a method for fast clustering by using nearest neighbor measurement identification. The simulation results show that the proposed method is more faster than previous method about 3.3%. We expect that this method could be effectively used in multi-target tracking particularly in heavy clutter environment.

Keywords : Track Merging, Track-to-Track Association, Multi-Target Tracking, Clutter Environment

### 1. 서 론

일반적인 추적시스템에서 표적 추적은 다음 Fig. 1과 같은 프로세스를 가진다<sup>[1]</sup>.

Fig. 1에서 보듯이 표적추적은 신호처리 과정, 데이터 연관과정, 추적필터 과정, 트랙관리 과정을 거친다.

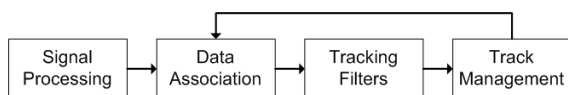


Fig. 1. Process of Target Tracking

신호처리 과정에서 레이더나 소나 등의 다양한 센서로부터 획득된 신호를 추적필터에서 사용하기 적합하게 처리하고, 이 처리된 신호들을 데이터 연관과정에서 결합하여 어떤 트랙에 연관시킬 것인지를 결정한다. 추적필터과정에서 연관된 측정치를 바탕으로 필터를 보정(Correct) 및 갱신(Update)하고, 트랙관리과정에서 트랙을 생성, 유지, 삭제하며, 형성된 트랙정보를 이용하여 데이터 연관과정에서 데이터 결합에 필요한 게이트를 계산하는 과정을 반복수행함으로써 표적추적을 수행하게 된다.

이러한 추적 시스템의 기본적인 목적은 추적 중인 각각의 물체와 관련된 하나의 트랙을 가지는 것이다. 이러한 목적을 달성하기 위해서 추적 시스템은 트랙 데이터 베이스와 관련된 일들을 처리하는 트랙 관리 프로세스를 두어야 한다<sup>[2]</sup>.

트랙 관리는 크게 세 가지로 구분될 수 있다. 첫째,

† 2012년 3월 8일 접수~2012년 6월 15일 게재승인

\* 국방과학연구소(ADD)

\*\* 대구보건대학교

책임저자 : 이승연(lee\_sy@add.re.kr)

트랙의 초기화이다. 즉, 초기 트랙 상태 행렬 추정치와 및 관련된 공분산 행렬을 할당하는 것이다. 초기 트랙 할당 기법에는 현 시점에서의 측정치를 기준으로 트랙을 할당하는 일 점 초기화(One Point Initialization) 및 현재 측정치와 이전 측정치를 이용한 두 점 초기화(Two Point Initialization) 기법이 있다<sup>3,4)</sup>.

두 번째는 트랙의 유지 및 보수와 관련된 작업이다. 이 과정에서는 트랙의 상태에 따라 유지할 것인지, 삭제할 것인지를 결정하며, 새롭게 형성된 트랙 및 기존에 있던 트랙 중 삭제된 트랙을 고려하여 트랙 재배치를 수행한다. 기존의 NNF(Nearest Neighbor Filter), PDAF(Probabilistic Data Association Filter), JPDAF(Joint PDA) 등의 데이터 연관필터에서는 트랙과 데이터 연관된 횟수를 고려하여 트랙의 관할 논리를 결정하는 m-out-of-n(n번 스캔에서 m번 연관된 정도를 살핌) 기법이 사용되며, IPDAF(Integrated PDAF)이나 LM-IPDAF(Linear-Multiple IPDA) 기법은 트랙의 존재확률을 제공하므로 이를 통해 트랙의 관할 논리를 결정하게 된다.

세 번째는 트랙간의 연관을 고려하여 하나의 물체에 대해서 중복되어 발생하는 트랙을 조정하는 기법이다. 이는 두 개 또는 그 이상의 트랙이 실제 같은 물체를 추적하고 있는 경우에 하나의 트랙으로 만들어주는 역할을 한다. 이를 트랙간 연관(TTTA : Track To Track Association) 기법이라 한다. 트랙간 연관이 이루어진 후 트랙간 융합(TTTF : Track To Track Fusion) 기법을 통해 다중센서로부터 획득된 트랙들을 결합하여 하나의 트랙을 형성하게 되면 단일센서로부터 얻어진 값보다 추정오차가 개선된 추적이 가능하다.

본 논문에서는 트랙간 연관기법 중 트랙 병합(Track Merging) 기법에 대해서 다룬다. 트랙병합기법은 센서에 의해 생성된 측정치 공간에서 중복되는 트랙을 삭제하는 기법에 해당하는 것으로, 중복된 트랙이 발생할 경우 이를 식별하고 삭제하여 하나의 표적에 하나의 트랙만이 존재하도록 하는 기법을 의미한다<sup>5)</sup>.

Fig. 2는 중복트랙이 발생하는 경우에 대한 예이다. 그림에서 ‘●’는 측정치이며, ‘○’는 클러스터이다. 또한 ‘x’는 측정치 참값에 의해 생성된 트랙의 예측치이며, ‘+’는 클러스터에 의해 생성된 트랙의 예측치이다. 실선 타원 및 점선타원은 유효 게이트를 의미한다.

그림에서 첫 번째 경우는 유효 게이트 안에 측정치가 들어올 경우 트랙이 유지되는 경우이다. 만약 유효 게이트에 측정치가 없을 경우 (N+1 시점) 트랙은 예측치를 추정치로 사용한다.

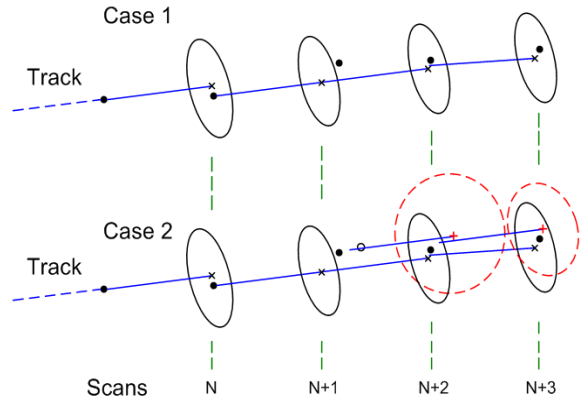


Fig. 2. In case of redundant tracks

그러나 미리 정해진 수 만큼의 측정치가 연속해서 들어오지 않거나 트랙존재확률이 문턱값 이하로 내려갈 경우 트랙은 삭제된다. 두 번째 경우는 두 점 초기화 기법을 사용하였을 때, 중복트랙이 발생하는 경우이다. N+1 시점에서 유효게이트안에 측정치가 들어오지 않았고, N+2 시점에 주변에 클러스터에 의해 새로운 트랙이 형성되고 추정되었다. 이때부터 같은 측정치를 공유함으로써 트랙 중복이 시작되었고, N+3 시점에서 하나의 측정치를 두 개의 트랙이 동시에 추적하게 되는 과정을 보여준다.

이와 같이 중복된 트랙은 병합(Track Merge)을 통해 하나의 표적에 하나의 측정치만 존재하도록 해야 한다. 이러한 중복된 트랙을 인지하고 삭제하기 위해 MHT(Multiple Hypothesis Tracking) 알고리즘에서는 N-scan back을 수행하여 같은 측정치를 연속적으로 사용하고 비슷한 상태 벡터를 사용하는 경우 트랙을 병합하여 트랙을 정확히 추정한다. 그러나 대부분 사용하는 NNF, PDAF, JPDAF, IPDAF, LM-IPDAF 등의 알고리즘들은 No-scan back이며, 이 때 중첩트랙의 병합을 위한 알고리즘은 모든 트랙의 상태벡터를 일대일로 비교하여 이루어지므로 트랙의 수를  $N_{track}$ 으로 둘 때, 최대  $(N_{track} - 1)!$ 의 연산량을 가지게 된다. 특히 클러스터가 많은 환경에서는 트랙의 수가 증가할수록 연산량은 기하급수적으로 늘어나게 되어 실시간 처리가 불가능해 질 수 있다.

본 논문에서는 이러한 문제점을 해결하기 위해 각 트랙에 최근접 측정치 고유번호를 할당하여 고속으로 클러스터링하고 클러스터링 된 트랙안에서 트랙을 병합하는 기법을 기술한다.

논문의 구성은 다음과 같다. 2장에서는 본 논문에서 제안하는 고속 트랙 병합 알고리즘을 설명한다. 3장에서는 시뮬레이션을 통하여서 제안된 알고리즘의 효과를 입증하였다. 4장에서는 결론을 제시하였다.

## 2. 고속트랙 병합기법

### 가. 트랙간 상관관계 검사기법

트랙간 상관관계 검사는 Kanyuck과 Singer<sup>[6]</sup>에 의해서 처음 제시되었으며, Kalman 필터에서 사용하는 상태벡터와 오차 공분산 행렬로부터 다음과 같이 유도된다.

시간  $k$ 에서 트랙  $i$ 와 트랙  $j$ 에서 상태벡터 추정오차는 다음과 같다.

$$\tilde{x}^i(k) = x^i(k) - \hat{x}^i(k) \quad (1)$$

$$\tilde{x}^j(k) = x^j(k) - \hat{x}^j(k) \quad (2)$$

여기서  $x^i(k)$ 와  $x^j(k)$ 는  $n$ -차원의 참 상태(True State)이고,  $\hat{x}^i(k)$ 와  $\hat{x}^j(k)$ 는 해당되는 추정상태 값이다. 두 추정치의 차는 다음과 같다.

$$\widehat{\Delta}^{ij} = \hat{x}^i(k) - \hat{x}^j(k) \quad (3)$$

참 값의 추정오차는 다음과 같이 표현할 수 있다.

$$\Delta^{ij} = x^i(k) - x^j(k) \quad (4)$$

만약 트랙  $i$ 와 트랙  $j$ 가 동일한 중복되었다면,  $\widehat{\Delta}^{ij}$ 는 0이 될 것이고, 다른 표적에서 발생한 것이라면 0이 아닌 값을 가지게 된다. 이를 다음과 같은 가설로 설정할 수 있다.

$$H_0: \Delta^{ij}(k) = 0, \hat{x}^i(k) \text{와 } \hat{x}^j(k) \text{가 같은 트랙임.} \quad (5)$$

$$H_1: \Delta^{ij}(k) \neq 0, \hat{x}^i(k) \text{와 } \hat{x}^j(k) \text{가 다른 트랙임.} \quad (6)$$

추정오차의 차이는  $\widehat{\Delta}(k)$ 로, 평균이 0이고 공분산은  $T^{ij}(k)$ 를 가진다.

$$\widehat{\Delta}^{ij}(k) = \Delta^{ij}(k) - \widehat{\Delta}^{ij}(k) \quad (7)$$

$$\begin{aligned} T^{ij}(k) &\doteq E\{\widehat{\Delta}^{ij}(k)\widehat{\Delta}^{ij}(k)'\} \\ &= E\{[x^i(k) - \tilde{x}^i(k)][x^i(k) - \tilde{x}^i(k)]'\} \\ &= P^i(k) + P^j(k) - P^{ij}(k) - P^{ji}(k) \end{aligned} \quad (8)$$

여기서  $P^i(k)$ 와  $P^j(k)$ 는 각각  $\tilde{x}^i(k)$ 와  $\tilde{x}^j(k)$ 의 공분산이며,  $P^{ij}(k) = P^{ji}(k)$ 은  $\tilde{x}^i(k)$ 와  $\tilde{x}^j(k)$ 의 교차공분산이다. 여기서 교차공분산을 Kalman 필터 관계식으로로부터 다음과 같이 유도할 수 있다.

$$\begin{aligned} P^{ij}(k) &= [I - K^i(k)H^i(k)] [F(k-1)P^{ij}(k-1) \\ &\quad \times F(k-1)' + Q(k-1)] [I - K^j(k)H^j(k)]' \end{aligned} \quad (9)$$

위 수식에서  $K$ 는 칼만 게인이며,  $H$ 와  $F$ 는 각각 측정치 행렬과 상태전이 행렬이다.  $Q$ 는 공정잡음 공분산이다. 그러나 대부분의 경우 단일 센서에서 중복되는 트랙의 경우 교차 공분산 값이 작아  $T^{ij}(k)$ 에 미치는 영향이 미미하므로 본 논문에서는 '0'으로 두고 계산하였다.

표적의 추정오차가 가우시안 분포를 따른다고 가정하면, 수식 (10)으로 둘 수 있고,  $n$ -차원의 자유도를 가지는 카이제곱 분포( $\chi_n^2$ )를 가진다.

$$D(k) \doteq \widehat{\Delta}^{ij}(k)' [T^{ij}(k)]^{-1} \widehat{\Delta}^{ij}(k) \quad (10)$$

Significance level을  $\alpha$ 라 두면, 다음과 같이 표현할 수 있다<sup>[7]</sup>.

$$\text{Accept } H_0 \text{ if } D(k) \leq \lambda_n = \chi_{n,1-\alpha}^2 \quad (11)$$

$$\text{Accept } H_1 \text{ if } D(k) > \lambda_n = \chi_{n,1-\alpha}^2 \quad (12)$$

### 나. 제안하는 알고리즘

본 논문에서는 트랙병합을 위해 다음과 같은 알고리즘을 제안한다.

- ① 센서로부터 획득한 측정치 각각에 고유번호(ID)를 부여하고, 각 트랙의 유효게이트를 산출한다.
- ② 각 트랙의 유효게이트 안에 들어온 측정치 중 트

랙의 예측치와 최근접한 측정치의 고유번호를 각 트랙의 정보속성에 추가한다.

- ③ 각 트랙의 정보속성으로부터 최근접 측정치의 고유번호가 동일한 트랙들을 클러스터링 한다. 이때, 각 트랙에 해당하는 개수의 버퍼(NN\_Buffer)를 생성하고, 여기에 각 트랙의 최근접 측정치의 고유번호를 저장한다. 여기서, 이미 다른 버퍼에 저장되어 있는 최근접 측정치 고유번호와 동일한 고유번호를 갖는 트랙에 대해서는 그 고유번호를 저장하는 대신, 고유번호의 중복횟수를 저장하여 클러스터링 작업이 고속으로 이루어지도록 한다.
- ④ 클러스터링 된 트랙 중에서(즉, 최근접 측정치의 고유번호가 같은 트랙) 공분산의 행렬식이 가장 작은 트랙을 선정한다.
- ⑤ 위에서 선정된 트랙을 기준으로 클러스터링 된 나머지 트랙에 대해서 수식 (10)을 사용하여 트랙간 거리를 산출한다.
- ⑥ 트랙간 거리가 일정범위(수식 (11)의  $\lambda_n$ )이내 일 경우 같은 트랙으로 판단하여 트랙 병합을 수행한다.

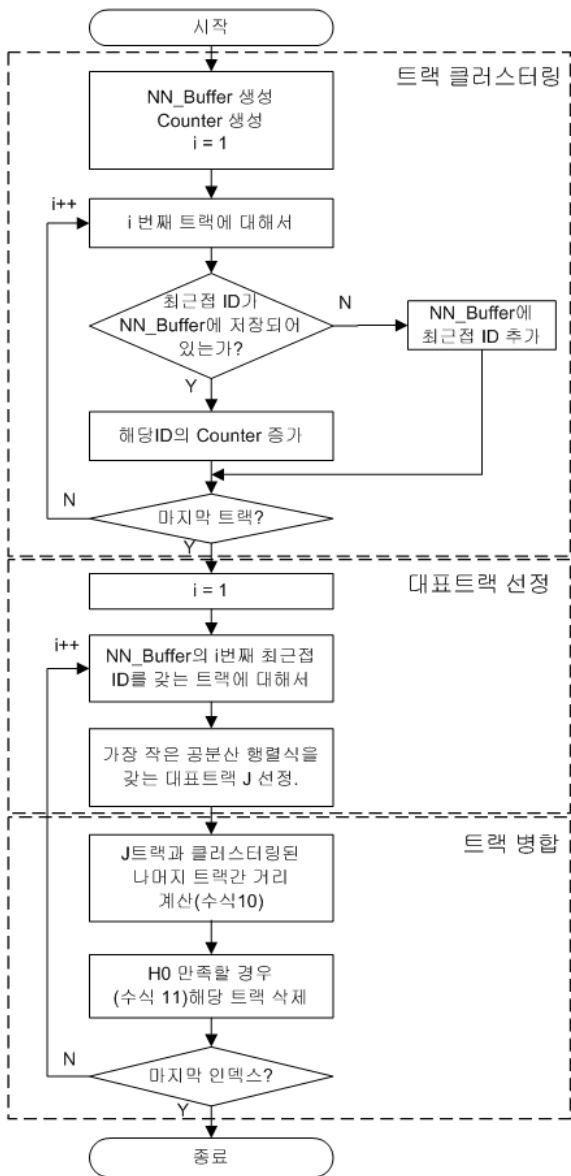


Fig. 3. Proposed algorithm of fast track merging

- 상기 알고리즘을 플로우 차트로 도시하면 Fig. 3과 같다.

### 3. 시뮬레이션

상기 제안한 기법을 IPDA 기법에 대해서 시뮬레이션 하였으며, 다음과 같이 세가지 방법을 비교하였다.

- ① 트랙병합을 수행하지 않음
- ② 기존의 트랙 병합기법
- ③ 제안한 방법

각각 100번의 몬테칼로 시뮬레이션을 수행하였고, 각 시뮬레이션은 8개의 표적이 오른쪽으로 8m/s의 등속도로 차례대로 나타났다 사라지는 시나리오를 가지도록 하였다. 측정주기는 33Hz로 설정하였고, 10초의 시뮬레이션 시간을 가지도록 구현하였다.

표적이 Gate 안에 들어올 확률은 0.92, 표적 발견 확률은 0.9로 설정하였다. 게이트 사이즈를 결정하는  $\gamma$ 는 16, 클러스터 밀도는  $0.45m^2$ 로 설정하여 중복 트랙의 발생 확률이 높도록 설정하였다. 트랙관리는 트랙 존재 확률이 0.8을 넘으면 확정트랙으로 선정하고, 0.1이하로 내려가면 트랙을 삭제하도록 하였으며, 수식 (11)에서  $\lambda_n$ 은 30으로 설정하였다.

세가지 방법에 대한 시뮬레이션 결과는 Fig. 4~6과 같다. Fig. 4는 트랙 수에 대한 확정 트랙 수를 나타내었다. Fig. 5는 트랙 수 오차를 도시한 그림으로서 트랙 병합 알고리즘이 없는 경우 트랙의 수(참값)가 많아질수록 중복되는 트랙도 많아지며, 평균 3개 정도 중복되는 것을 확인할 수 있으며, 기존의 방법과 제안

한 방법의 경우 평균 0.5개 정도 중복되어 트랙 병합의 필요성을 보여준다.

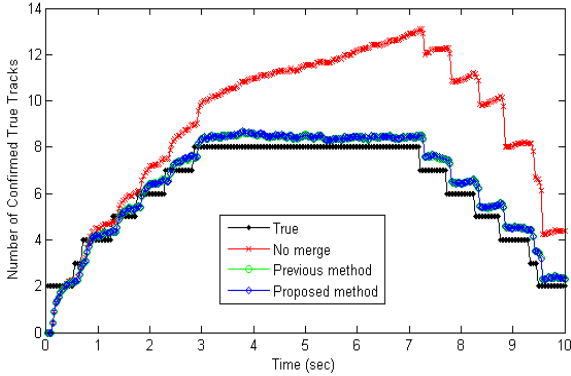


Fig. 4. Number of conformed tracks

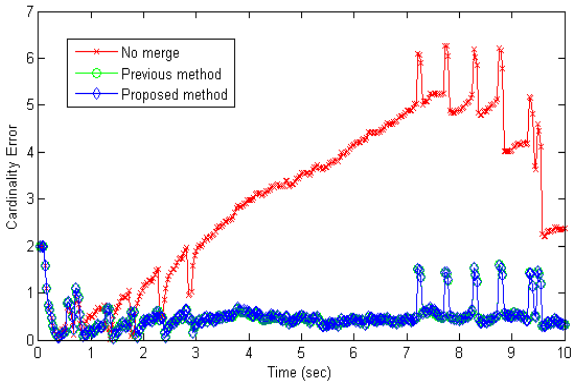


Fig. 5. Cardinality error

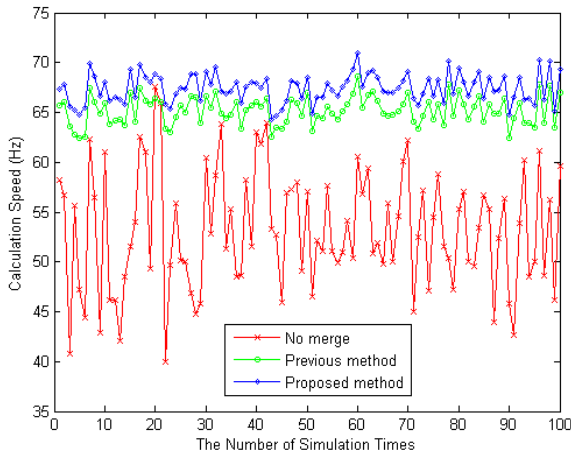


Fig. 6. Calculation speed according to simulation times

Table 1. Comparison of processing time

방법	수행시간 평균(sec)	단위연산 시간(Hz)
병합안함	6.2896	52.4673
기존의 방법	5.0637	65.1701
제안한 방법	4.8968	67.3903

Fig. 6은 몬테칼로 시뮬레이션 100회에 대한 연산 시간을 도시하였으며, Table 1에 평균 연산시간을 비교하여 정리하였다. Table 1을 통해서 볼 때, 제안한 방법은 트랙병합을 수행하지 않았을 때 보다 22.1%, 기존 방법보다 3.3% 빠른 연산 시간을 보여주고 있음을 확인할 수 있다.

#### 4. 결론

본 논문에서는 트랙연관에서 가장 기초가 되는 트랙 병합을 고속으로 처리하는 방법에 대해서 연구하였다. 트랙 병합기법은 트랙간 연관기법 중 중복되는 트랙을 제거하여 하나의 표적에 하나의 트랙이 추적할 수 있도록 해주는 기법이다. 기존의 방법은 중첩되는 트랙을 제거하기 위해 모든 트랙과 비교를 수행하여야 하므로 계산시간이 과다하게 소요되었다.

본 논문에서는 제안한 트랙 병합 방법은 측정치에 고유번호를 부여하고, 각 트랙에서 추정치와 가장 근접한 고유번호를 할당하며, 할당된 고유번호를 기준으로 클러스터링을 수행한다. 각 클러스터링 된 트랙 중에서 가장 작은 공분산을 가진 트랙을 기준으로 나머지 트랙을 비교하여 일정 조건(수식 (11))을 만족할 경우 해당 트랙을 삭제함으로써 트랙 병합이 이루어진다.

제안한 기법의 효과를 입증하기 위해 100회의 몬테칼로 시뮬레이션을 수행하였고, 기존의 기법보다 3.3% 수행시간을 단축시킴을 확인하였다.

상기 개발된 기법은 다중표적추적기법에 널리 사용될 것으로 사료되며, 특히 클러터가 많은 곳에서의 다중표적 추적에 매우 큰 효과를 나타낼 것으로 사료된다.

## References

- [1] S. S. Balckman, "Multiple-Target Tracking with Radar Applications", Artech House Publishers, 1986.
- [2] Y. Bar-Shalom, W. D. Blair, "Multitarget-Multisensor Tracking : Applications and Advances", Artech House, Vol. 3, pp. 59~66, 1989.
- [3] X. Wang, et. al. "Enhanced Multi-Target Tracking with Doppler Measurements", Information, Decision and Control, pp. 53~58, 2009.
- [4] Y. Bar-Shalom, et. al. "Automatic Track Formation in Clutter with a Recursive Algorithm", Proc. 28th Conf. Dec. Control, pp. 1402~1408, Dec. 1989.
- [5] Y. Bar-Shalom, et. al. "Multitarget-Multisensor Tracking : Advanced Applications", Artech House, pp. 6~9, 1989.
- [6] Kanyuck, A. J., R. A. Singer, "Correlation of Multiple Site Track Data", IEEE Transactions on Aerospace and Electronic Systems, Vol. AES-6, No., pp. 180~187, March, 1970.
- [7] Wei Mei, Gan-lin Shan, "Performance of a Multi-Scan Track-to-Track Association Technique, Signal Processing, Vol. 85, pp. 1493~1500, 2005.
- [8] Y. Bar-Shalom, T. E. Fortmann, "Tracking and Data Association", Academic Press, 1988.
- [9] D. Musicki, R. Evans, and S. Stankovic, "Integrated Probabilistic Data Association", IEEE Transactions on Automatic Control, Vol. 39, No. 6, June 1994.