

# RFID 태그 추적을 위한 TPIR\*-Tree

이세호<sup>0</sup> 안성우 홍봉희

부산대학교 컴퓨터 공학과  
{rustyangel<sup>0</sup>, swan, bhong}@pusan.ac.kr

## TPIR\*-Tree for tracing RFID tags

Seho Lee<sup>0</sup> Sungwoo Ahn Bonghee Hong  
Dept. of Computer Engineering, Pusan National University

### 요 약

RFID 시스템은 전자태그를 상품에 부착하여 리더를 통해 태그를 인식함으로써 상품의 정보 및 위치정보를 추적할 수 있다. 태그 객체의 위치정보는 시간에 따라 궤적 정보가 누적되는 이동체와 유사한 특성을 가지지만 이동체의 위치와 달리 태그의 위치는 리더의 위치로 인식되며 위치보고가 리더의 인식영역 안에서만 이루어지므로 시간축에 평행한 interval의 형태로 나타난다. 태그가 리더의 인식영역에 들어와 나가지 않으면 궤적이 완성되지 않아 리더에 머물고 있는 태그의 궤적을 표현할 수 없으므로 질의 시 이러한 태그를 검색할 수 없다. 시공간 이동체 색인에서는 이러한 태그의 특성을 표현하기 힘들기 때문에 태그의 특성을 고려한 색인이 필요하게 되었다. TPIR\*-Tree(Time Parameterized Interval R-tree)는 시간 매개변수 간격으로 태그의 interval을 정의하여 리더안에 머무는 태그의 interval을 표현할 수 있다. 그러나 각 interval이 시공간적으로 연결되어 있지 않아 색인 상에서 태그의 궤적을 검색하는 것은 매우 높은 검색 비용을 가지는 단점이 있다.

이 논문에서는 태그 궤적 검색 시 TPIR\*-Tree의 높은 검색 비용문제를 해결한 TPIR\*-Tree를 제안한다. 제안된 TPIR\*-Tree는 색인에서 태그의 궤적 정보를 유지할 수 있도록 하기 위해서 시간적으로 연결되지 않은 각 interval을 연결하기 위한 기법을 제시하고 있다. 또한, interval을 색인에 삽입할 때 연결정보를 유지하기 위해 이전 interval을 효율적으로 검색할 수 있는 방법을 제시하고 있다.

### 1. 서 론

RFID(Radio Frequency Identification)[1]는 무선 주파수를 이용하여 태그를 부착한 객체를 리더가 자동으로 인식하고 정보를 읽는 기술로써 빠른 인식 속도, 인식을 많은 데이터의 저장능력으로 RFID를 사용하는 응용분야가 주목을 받고 있다. 이러한 기술은 재고관리, 물류의 공급망 관리, 공장 자동화 등의 다양한 응용분야에서 사용되고 있다.

태그는 이동체와 유사하게 시간에 따라 연속적으로 이동하므로 위치를 모델링하기 위해서 이동체를 위한 시공간 색인을 적용할 수 있으나 태그는 이동체와 비교하여 다음과 같은 차이점을 보인다. 첫째, RFID 시스템에서 질의는 "reader#4에 있었던 태그들은?" 같이 이동체 색인의 실제좌표와 달리 논리적 위치를 가진다. 즉, x-y 좌표계가 아닌 리더의 번호와 태그의 번호를 도메인으로 가진다. 따라서 이 도메인은 연속적이지 않고 이산적인 특성을 가진다. 둘째, 태그의 궤적은 연결되어 있지 않다. 위치를 보고하여 태그의 위치가 아닌 리더의 위치를 보고하고, 위치보고 주기가 비주기적이므로 각 보고위치를 선분으로 연결할 수 없다. 다만 태그가 어떤 리더의 인식영역에 들어갈 때와 나갈 때의 두 시점만을 연결할 수 있다. 셋째, 태그의 위치가 리더의 위치로 보고되기 때문에 리더의 인식영역에 들어가고 나갈 때 태그의 위치정보는 시간축에 평행한 이산적인 interval로 표현된다. 마지막으로 이동체의 경우 이동체의 수에 비해 상대적으로 보고 횟수가 많아 각 이동체의 궤적정보가 많다. 이에 반해 태그의 경우, 그 수가 많고 보고 횟수는 비주기적이며 보고 횟수가 적어 각 태그의 궤적정보는 적다.

RFID 시스템은 물류/유통 등의 영역에서 각 물품들의 이력정보를 자동적으로 수집, 파악하는 것에 사용된다. 이러한 환경에서 각 물품의 이력정보에 관한 질의가 중요하다. 대형 할인 매장에서 재고 관리를 위해 창고에 현재 있는 물품들에 대한 질이나 축산물의 유통경로 파악을 위한 궤적 질의를 수행할 수 있어야 한다. 최근 이러한 태그의 특성을 고려한 색인인 TPIR\*-Tree(Time Parameterized Interval R-Tree)[2]가 제안되었다. 이 색인의 경우 태그 객체의 특성을 단절된 궤적의 표현과 태그가 리더의 인식영역 안에 머물러 있는 상태를 Now interval로 표현하여 처리하는 방법을 통해 데이터의 삽입 시 효과적인 노드 선택 기법과 현재 질의 처리 방법을 제시하고 있다. 그러나 태그의 위치 정보가 이산적인 interval로 표현되어 각 interval은 시공간적으

로 연결되어 있지 않다. 이로 인해 궤적 질의를 수행할 때 검색 영역이 급격히 증가하여 검색 성능이 떨어지는 문제가 있다. 본 논문에서는 태그의 궤적 검색 성능 향상을 위해 각 태그 interval에 연결정보를 유지하고, 이런 연결정보를 효과적으로 삽입하는 기법을 제시한다. 이 논문의 구성은 다음과 같다. 2장에서는 관련연구를 소개하고, 3장에서는 대상 환경 및 태그의 특성으로 인한 문제점을 정의한다. 4장에서는 태그의 위치정보를 연결하는 방법과 삽입 정책 및 궤적 검색 기법을 제시한다. 마지막으로 6장에서 결론 및 향후 연구를 기술한다.

### 2. 관련 연구

RFID 시스템의 질의처리를 지원하기 위한 색인인 TPIR\*-Tree가 최근에 제안되었다. TPIR\*-Tree[2]의 경우 리더의 인식영역 안에 들어가서 위치정보를 보고하고 아직 리더를 벗어나지 않아서 완성되지 않은 interval을 Now interval로 리더의 인식영역을 벗어나 interval이 고정된 Fixed interval로 표현하여 데이터의 삽입과 노드의 분할 시 이를 부모노드의 시간 경계까지 시간 값을 확장해 처리하여 효과적인 데이터 저장과 현재 질의를 가능하게 하였다.

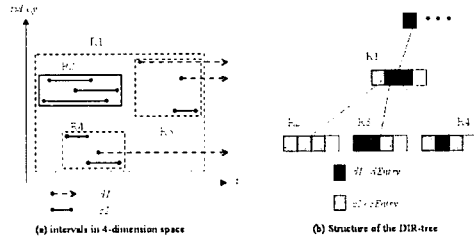


그림 1 TPIR\*-Tree의 모델링 방법 및 색인 구조

그러나 그림1에서와 같이 각 interval간의 연결 정보가 없으므로 궤적 검색 시 태그 축으로 slice질의를 통해 데이터를 검색하여 시간값으로 정렬을 해야 된다. 또한 시간값과 공간값이 도메인의 전체 영역을 되므로 검색 영역의 증가하여 궤적 질의 시 검색 성능이 떨어지는 문제점이 있다.

본 논문은 교육인적자원부 지방연구중심대학육성사업 "차세대물류 IT기술연구사업단"의 지원에 의하여 연구되었음

### 3. 대상 환경 및 문제 정의

이 장에서는 RFID 시스템의 환경을 기술하고, 태그의 특성으로 인한 궤적검색의 문제점을 분석한다.

#### 3.1 대상 환경

RFID 시스템은 리더와 태그, 그리고 서버로 구성된다. 리더는 자신의 태그 인식영역 안으로 태그가 들어오면 이에 대하여 *In\_Event*를 발생하여 서버에 전송하고, 인식영역을 나갈 때는 *Out\_Event*를 발생하여 그 정보를 서버에 전송하게 된다. 리더의 인식영역은 물리적인 리더의 인식영역이 아닌 논리적 인식영역으로 창고의 입구와 출구에 있는 물리적 리더가 입구에서 태그를 읽으면 *In\_Event*를 출구에서 읽으면 *Out\_Event*를 발생하여 창고에 부착된 물리적 리더를 논리적 리더로 매핑하고 창고를 하나의 논리적 리더의 인식영역으로 판단할 수 있다.

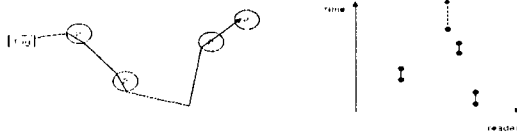


그림 2 태그의 궤적 표현

태그의 궤적은 그림2와 같이 태그가 리더의 인식영역 안에 들어갈 때와 나갈 때 보고된 두 시공간 위치를 연결한 선분으로 표현된다. 보고되는 태그의 위치는 리더의 위치를 태그의 위치로 보고하기 때문에 리더의 인식영역으로 들어가고 나갈 때까지의 위치는 변하지 않는다. 따라서 태그의 궤적은  $t_{tag} = \{(readerID, tag, time) \in R^3 \mid readerID = r, tag = t_0, time_n \leq time \leq time_{out}\}$ 인 시간축에 평행한 interval로 표현된다. 태그의 위치는 곧 리더의 위치이므로 각 보고위치가 이산적이며 태그가 리더의 인식영역 안으로 들어 왔을 때만 그 위치를 파악할 수 있기 때문에 태그의 궤적은 시공간적으로 연결되어 있지 않은 interval의 집합으로 나타난다.

#### 3.1 문제 정의

태그는 기존의 이동체와는 많은 부분에서 그 특성이 상이하게 나타난다.

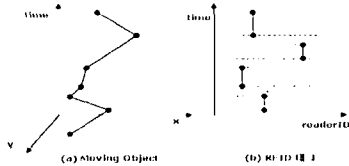


그림 3 이동체와 태그의 위치 표현 방법

그림3(a)과 같이 이동체의 궤적은 각 보고 시점을 잇는 선분으로 표현되어 하나의 polyline의 형태로 나타난다[3]. 선분의 시작점과 그 이전의 선분의 끝점은 같기 때문에 궤적 검색 시 time slice질을 통해 해당 이동체의 선분을 찾아내고 연결된 각 시점에 대한 점 질의를 통해 궤적 추적이 가능하지만 그림3(b)처럼 태그의 궤적이 이산적인 interval의 집합으로 표현되기 때문에 이동체 색인에서와 같이 각 interval간의 연결된 궤적 정보를 얻어내는 것이 어렵다. 또한 태그정보가 유효한 기간 중 시간도메인 상에 위치정보를 표현하지 못하는 구간이 생기기 때문에 이동체 색인처럼 time slice질의 시 지원하는 태그의 데이터의 검색을 보장할 수 없다. 즉, 시공간적으로 어떠한 연결도 없기 때문에 궤적추적 시 검색 영역의 증가로 낮은 검색 효율을 가진다.

### 4. TPIR\*-Tree

4장에서는 TPIR-Tree를 RFID시스템 환경에서 궤적 질의의 처리를 효율적으로 수행하기 위해 도메인과 색인구성을 변경하고 새로운 질의 처리 알고리즘 제시한 TPIR\*-Tree에 대하여 기술한다.

#### 4.1 태그 객체 모델링

TPIR-Tree의 도메인은 x-y좌표, 태그번호와 시간값을 가진다. 그러나 RFID 환경에서의 질의는 “리더#9에 있었던 태그들과 그 시작은?”, “EPC#e5가 지나온 리더들과 그 시작은?”[4]같은 태그, 리더 그리고 시간이 주요 인자이다. 따라서 효과적인 색인 구성을 위해서 기존의 x,

y, tag, time의 도메인이 아닌 ReaderID, tag, time을 도메인으로 설정한다. 그림4처럼 태그의 부분 궤적인 interval은 태그가 리더의 인식영역을 벗어나지 않은 *Now interval*과 리더의 인식영역을 벗어나 interval이 고정된 *Fixed interval*로 분류할 수 있다[2].

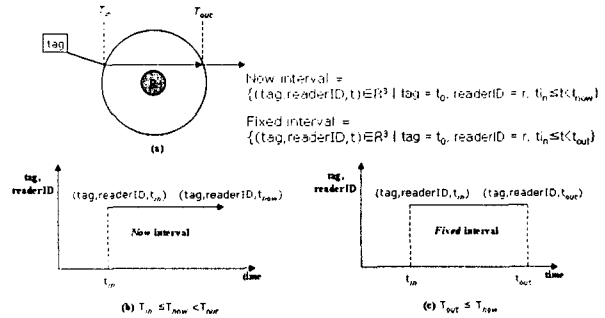


그림 4 태그 객체의 interval에 대한 모델링

#### 4.2 동일 태그에 대한 interval의 연결

TPIR-Tree는 동일태그에 대한 이산적인 interval의 연결정보를 유지하지 않기 때문에 높은 궤적 검색 비용을 가진다. 태그의 interval 자체에는 그러한 연결정보가 없기 때문에 색인 상에서 궤적 검색을 위한 정보를 유지할 필요가 있다. 효과적인 궤적 검색을 위해 태그의 이산적인 interval을 양방향 연결 리스트로 구성할 수 있다. 이러한 연결정보를 유지하는 방법은 다음과 같은 두 가지 방법이 있다. 첫째 TB-Tree[5]에서와 같이 각 태그의 interval만 같은 저장하고 각 단말노드를 연결하는 번들(bundle) 연결 기법, 두 번째 방법으로는 개별 interval끼리 직접 연결하는 태그의 수만큼의 단말노드 검색이 발생하게 되어 TPIR-Tree에 비해 낮은 검색 성능을 보인다. 또한 태그의 궤적을 구성하는 interval은 작기 때문에 노드의 공간 활용을 좋지 못하다. 따라서 태그와 리더에 관한 질의가 빈번한 RFID환경에서 적용하기 어렵다. TPIR\*-Tree에서는 그림5와 같이 TPIR-Tree의 궤적 질의에 비해 빠른 검색을 보장하면서 리더에 관한 질의 성능을 보장하기 위해 두 번째 방법인 동일 태그의 interval을 직접 연결시키는 태그 리스트를 이용하여 interval간의 연결 정보를 유지한다.

전자의 경우, 하나의 단말노드에 하나의 태그 interval만 저장하기 때문에 궤적 검색 성능은 아주 뛰어나지만 “reader#9에 있었던 태그들은?”같은 리더에 관한 질의 시 하나의 단말노드에 하나의 태그 정보만 존재하기 때문에 최소한 해당되는 태그의 수만큼의 단말노드 검색이 발생하게 되어 TPIR-Tree에 비해 낮은 검색 성능을 보인다. 또한 태그의 궤적을 구성하는 interval은 작기 때문에 노드의 공간 활용을 좋지 못하다. 따라서 태그와 리더에 관한 질의가 빈번한 RFID환경에서 적용하기 어렵다. TPIR\*-Tree에서는 그림5와 같이 TPIR-Tree의 궤적 질의에 비해 빠른 검색을 보장하면서 리더에 관한 질의 성능을 보장하기 위해 두 번째 방법인 동일 태그의 interval을 직접 연결시키는 태그 리스트를 이용하여 interval간의 연결 정보를 유지한다.

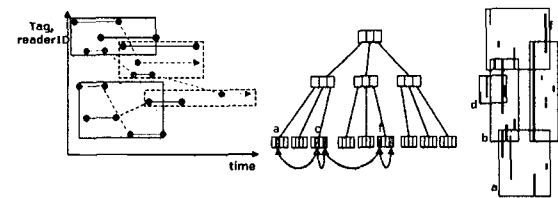


그림 5 TPIR\*-Tree의 색인 구조

#### 4.3 삽입 정책

4.2절에서 궤적의 검색을 효과적으로 지원하기 위해 태그의 interval을 연결하였다. 그러나 연결 정보를 이용하여 궤적을 검색하기 위해서는 먼저 interval을 찾아야 하며 매번 interval의 삽입 때마다 이전 interval과의 연결 정보 유지를 위해 발생한다. 언급한바와 같이 태그의 interval은 이산적으로 분포하기 때문에 기존 이동체 색인처럼 time slice질을 통해 해당 선분을 찾는 방법을 사용할 수 없다. 이 절에서는 데이터의 삽입 시 태그의 마지막 interval의 효율적인 검색을 지원하는 기법을 제시한다.

마지막 interval의 정보를 유지하는 방법은 외부의 보조 색인을 해석하는 방법과 색인 안에서 그 정보를 유지하는 방법이 있다. 보조 색인을 사용하는 경우는 유지 방법이 간단하고 구현이 쉬운 장점이 있지만 보조색인은 태그의 수만큼의 저장 공간이 필요하고, 매 삽입 때마다 보조색인의 갱신이 필요하며 주 색인의 마지막 interval을 중복 저장하는 문제가 있다. 또한 보조색인은 각 태그의 마지막 interval의 readerID와 시작시간만을 저장하고 있기 때문에 보조색인과 주 색인을

차례로 검색하고 다시 보조색인을 갱신하는 문제가 있다. 색인안에 정보를 유지하는 경우, 그림6과 같이 색인 안에 interval의 상태를 기록하는 방식으로 제한된 정보로 인해 검색단계가 복잡해진다는 문제가 있지만 별도의 저장 공간이나 갱신 비용이 없으며 TPIR-Tree에 비해 낮은 계층의 검색의 비용을 가진다. TPIR-Tree는 리더의 인식영역 안에 머무르고 있는 데이터를 표현하기 위한 Now interval검색을 위해 비단말 노드에 자식노드에 저장된 interval의 구성 정보를 저장하고 있다. TPIR-Tree이 구성 정보에 마지막 interval의 포함여부를 추가로 저장하여 데이터 삽입 시 효과적인 마지막 interval 검색을 지원하고자 한다.

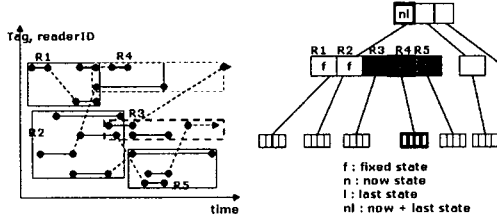


그림 6 TPIR-Tree의 색인구조 및 interval 연결 방법

TPIR-Tree에서는 기존의 Now interval과 Fixed interval이외에 각 태그의 마지막 Fixed interval을 Last Fixed Interval로 정의한다. 각 비단말 노드는 그림6 및 표1과 같이 4가지 상태 중 하나를 가진다. 자식 노드에 Now interval이 있는 경우와 Last Fixed interval이 있는 경우, Now interval과 Last Fixed interval이 둘 다 있는 경우와 Fixed interval만 존재하는 경우이다. 이중 Fixed state의 경우 자식노드에는 모두 Fixed interval만 존재하는 경우이므로 삽입 시 검색 대상이 되지 않는다.

표 1 TPIR-Tree의 노드 분류

	Now State	Last State	Now / Last State	Fixed State
자식 노드에 포함하는 interval	Now Fixed	Last fixed Fixed	Now Last fixed Fixed	Fixed

색인으로의 interval의 삽입은 태그가 리더의 인식영역 안으로 들어올 때 발생하는 Now interval의 삽입과 리더의 인식영역 밖으로 나갈 때 발생하는 Fixed interval의 삽입이 있다. Now interval을 삽입할 때는 이전의 Fixed interval을 검색해 두 interval간의 연결정보를 생성하게 된다. 이전의 Fixed interval은 위에서 정의한 Last fixed interval이다. 새로 삽입되는 태그값으로 현재질의를 통해 Last fixed interval을 검색하게 되는데 이때 해당영역과 겹치는 비단말 노드 중에서 노드의 상태가 Last state나 Now/Last state인 노드만 검색하여 Last fixed interval을 찾을 수 있다. 이 interval의 정보는 색인에 Now interval의 삽입할 때 두 interval을 연결정보로 사용된다. 이와 유사하게 Fixed interval을 삽입하는데 해당 태그의 끝 시간만 알기 때문에 이전의 Now interval을 검색하여 interval의 시작시간을 알아야 한다. 현재질의를 통해 해당 영역과 겹치는 비단말 노드 중에서 Now나 Now/Last state를 가지는 노드만을 검색하여 해당 태그의 Now interval을 검색하여 interval의 시작시간과 이전 interval과의 연결정보를 알아내고 Now interval을 삭제하고 시작 시간과 끝 시간을 연결한 Fixed interval을 삽입하게 된다.

```

FindLast(N,tag,state)
F1 Set N to be the root
F2 If state is enter then invoke F4.
   otherwise invoke F3
F3 IF N is NOT a leaf.
   FOR EACH entry E' of N whose state with the now state or
   now/last state
   invoke FindLast(N',tag,state), where N' is the childnode of N
   pointed to by E'
   ELSE IF
   N Contains an entry E' that tag id is same
   RETURN E'
F4 IF N is NOT a leaf.
   FOR EACH entry E' of N whose state with the last state or now/last
   state
    
```

```

invoke FindLast(N',tag,state), where N' is the childhood of N
pointed to by E'
ELSE IF
N Contains an entry E' that tag id is same
RETURN E'

Insert(N,I,state)
I1 Set N to be the root
I2 If N is a leaf
   invoke FindLast(N,tag,state) where tag is I tag id
   connect each interval and change node state that contains previous
   interval if the node organization is changed
   RETURN N
else
   Choose the entry E' by least new area enlargement. Resolve ties by
   least overlap enlargement. change node state if the node
   organization is changed
I3 Set N to be the childhood pointed to by the childpointer of E' and
   repeat from I2
    
```

4.4 태그의 계적 검색

태그의 계적 검색은 interval의 연결 정보를 이용하여 효과적으로 수행할 수 있다. 전체 계적 검색은 그림 7(a)처럼 노드의 구성 정보를 이용하여 태그의 마지막 interval을 검색하여 연결된 interval들을 연결정보를 이용하여 다음 interval을 바로 찾아갈 수 있다. 시간 간격 질의의 경우는 이동체 색인처럼 시작 시간으로 time slice 질의를 수행할 수 없기 때문에 질의의 시간 간격으로 영역질을 수행하여 해당 태그의 interval을 찾은 이후에 interval의 연결정보를 이용하여 바로 검색을 할 수 있다.

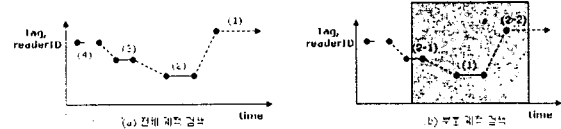


그림 7 TPIR-Tree의 계적 검색

5. 결론 및 향후연구

이 논문은 기존 이동체와 RFID시스템의 태그 환경의 특징에 대해 분석하고 기존 색인의 높은 계적 검색 비용문제를 해결하기 위해 interval간의 연결정보 유지 기법과 삽입 및 계적 검색 시 효율적인 interval 검색 위한 기법을 적용하여 태그 계적 검색 시 TPIR-Tree의 높은 검색 비용 문제를 해결한 TPIR-Tree를 제안하였다. 제안된 TPIR-Tree는 색인에서 태그의 계적 정보를 유지할 수 있도록 하기 위해서 시간적으로 연결되지 않은 각 interval을 연결하기 위한 방법을 제시하고, interval을 색인에 삽입할 때 연결정보를 유지하기 위해 이전 interval을 효율적으로 검색할 수 있는 방법을 제시하였다.

향후 연구로써 검색 성능 향상을 위한 interval의 상태에 따른 노드 재배치 전략을 연구하고 실험을 통해 TPIR-Tree, TPIR-Tree과의 삽입 및 계적 질의에 대한 성능 평가가 필요하다.

6. 관련연구

[1] K.Romér, T.Schoch, F.Mattern and T. Dubendorfer, "smart Identification Frameworks for Ubiquitous Computing applications", In Pervasive Computing and Communications Proceedings of the First IEEE International Conference, 256-262, 2003  
 [2] 반재훈, 홍봉희, "RFID 시스템에서 태그의 위치 추적을 위한 시간 매개변수 간격 모델링 기법", 한국정보처리학회 추계학술발표대회 논문집, 12권 1호, pp. 129 - 132, 2005  
 [3] Y. Theodoridis, M. Vassilakopoulos, and T. Sellis, "Spatio-temporal indexing for large multimedia applications", In Proc. of the 3rd IEEE Conf. on Multimedia Computing and Systmes, pp. 441- 448, 1996  
 [4] Mark Harrison, "EPC™ Information Service - Data Model and Queryis" 2003  
 [5] D.Pfoser, S. Jensen, and Y. Theodoridis, "Novel Approaches in Query Processing for Moving Object Trajectories", In Proc. of VLDB, pp. 395-406, 2000.