

農業機械利用의 效率提高를 위한 耕地整理의  
基礎設計技術開發研究 (I)

A Study on the Development of Basic Design Criteria of Land  
Rearrangement for Effective Use of Farm Machinery (I)

1 9 9 2 . 1 1 .

研 究 機 關

서울大學校 農業生命科學大學 附設

農業開發研究所

農 林 水 產 部

農 漁 村 振 興 公 社



# 提 出 文

농어촌진흥공사 사장 귀하

본 보고서를 “農業機械利用의 效率提高를 위한 耕地整理의 基礎設計 技術開發에 관한 研究” 研究과제의 제1차년도 보고서로 제출합니다.

1992년 11월 30일

研究機關名 : 서울大學校 農業生命科學大學校  
附設 農業開發研究所

責任研究員 : 鄭 昌 柱 (서울大學校 農業生命科學大學 農工學科 教授)

研 究 員 : 柳 寬 熙 (서울大學校 農業生命科學大學 農工學科 教授)

趙 成 仁 (서울大學校 農業生命科學大學 農工學科 教授)

補助研究員 : 崔 種 栢 (서울大學校 農業生命科學大學 農工學科 大學院生)

禹 相 河 (서울大學校 農業生命科學大學 農工學科 大學院生)

郭 柄 哲 (서울大學校 農業生命科學大學 農工學科 大學院生)

여 백

# 要 約 文

1. 研究課題名 : 農業機械利用의 效率提高를 위한 耕地整理의 基礎設計  
技術開發에 관한 研究

2. 研究期間 : 1992. 5. 1 - 1992. 11. 30 (총 6개월)

3. 研究의 必要性 및 目的

- 1) 耕地의 條件別, 區劃形狀別 農業機械의 利用效率 究明
- 2) 農業振興地區의 農業機械化 推進을 위한 適正 機械化 體系의 模型 開發
- 3) 農機械 利用效率 提高를 위한 耕地基盤造成의 設計基準設定 提示

4. 研究의 內容 및 範圍

가. 1次年度

- 1) 農作業計劃에 必要한 農業機械, 農業施設 및 裝置, 農資材, 栽培作物,  
圃場條件, 土壤條件, 氣象條件, 作業體系에 대한 데이터베이스化
- 2) 地域別, 作業別에 대한 作業可能日數率 決定
- 3) 作業別, 耕地 規模別 農機械의 作業效率의 決定
- 4) 映像處理 技術을 利用한 未耕地整理 地區의 田畝形狀, 面積 및 高低  
등의 데이터베이스 開發

## 나. 2次年度

- 1) 農業振興地區의 機械化 體系의 模型開發
- 2) 機械化 體系別 勞動生産性과 利用費用分析
- 3) 映像處理 데이터를 利用한 傾斜度, 形狀, 面積別 圃場區劃의 適正 要因 變數의 究明
- 4) 映像處理를 통해서 機械利用效率를 提高하기 위한 耕地의 條件別 區劃, 農路 등의 設計方向 設定
- 5) 機械利用效率提高를 위한 耕地 整備 設計基準 設定을 위한 전문가 시스템 開發

## 5. 研究結果 및 實用化方案

### 가. 研究結果 (1次年度)

- 1) 農作業의 耕地規模別 理論 및 實際 圃場作業效率의 分析
- 2) 農機械, 農業施設 및 裝置, 栽培作物, 作業體系 등에 대한 데이터베이스構築
- 3) 收穫作業의 地域別 作業可能日數率의 決定
- 4) 映像處理를 통한 耕地整理 對象地域의 地形情報 獲得

### 나. 實用化方案

農業機械 利用效率 極大化를 위한 耕地整備 基礎設計 方案을 提示해 줄 수 있는 耕地整理 전문가시스템을 개인용 컴퓨터를 이용해서 사용할 수 있도록 실용화할 계획이다. 이 전문가 시스템을 운용하기 위해서는 경지의 諸般與件에 따른 農機械 利用效率分析, 適定 機械化 體系 模型提示 및 農作業計劃에 필요한 제반 데이터베이스 構築, 映像處理를 이용한 地形情報獲得 등이 先行되어야 한다.

# S U M M A R Y

1. Title : A Study on the Development of Basic Design Criteria of Land Rearrangement for Effective Use of Farm Machinery(I)

2. Periods of Study : 1992. 5. 1 - 1992. 11. 30 (Total 6 Months)

3. Object and Necessity of The Study

- 1) Study on the farm machinery utilization efficiency related to area shapes and conditions of farm field.
- 2) Development of appropriate farm mechanization models in agriculture promotion regions.
- 3) Establishment of design criteria for cultivated land consolidation to improve operation efficiencies of farm machinery.

4. Contents and Scope of the Study

A. The First year

- 1) Database construction on farm machinery, farm facilities, farm equipments, farm materials, crop, soil types, weather conditions, field operation systems.
- 2) Estimation of possible working days according to various field operations in different regions.

3) Determination of operation efficiencies according to different field operations and field sizes.

4) Development of an image processing technique to determine field shape, area and altitude of the farm field where a land consolidation is required.

#### B. The Second year

1) Development of appropriate farm mechanization systems in the agriculture promotion regions.

2) Analysis of labor productivity and utilization cost according to different farm mechanization systems.

3) Study of various parameters for optimal field readjustment under different land slopes, shapes and areas.

4) Establishment of design criteria for readjusting field size and opening roads to improve operation efficiencies of farm machinery using image processing techniques.

5) Development of an expert system to establish basic design criteria for cultivated land readjustment to improve operation efficiencies of farm machinery.

### 5. Results of The Study and Suggestion for Application

#### A. Results of the Study (first year)

1) Theoretical and experimental analysis of field operation efficiencies according to various cultivated land sizes and field operations.



- 2) Database construction on farm machinery, farm facilities, farm equipments, farm materials, crop, field operation systems.
- 3) Determination of possible working days according to various field operations in different resions.
- 4) Development of an image processing technique to determine field shape, areas and altitude of the farm field where a land consolidation is required.

#### B. Suggestion for Application

An expert system to establish basic design ceriteria for cultivated land readjustment to improve operation efficiencies of farm machinery will be developed on the basis of personal computers. Analysis of farm machinery operation efficiency, model for appropriate farm mechanization, database construction on farm machinery, crop, possible working days, weather, etc. and image processing technique to obtain geographical information should support the expert system.

여 백

# 目 次

第 1 章 序 言 .....	15
第 2 章 耕地規模 및 作業別 圃場作業效率의 分析 .....	19
2.1 耕地整理에 關聯된 農業機械化 變數 .....	19
2.2 農作業計劃樹立의 關聯變數 .....	21
2.2.1 圃場作業能率 .....	21
1. 理論圃場作業能率 .....	22
2. 有效圃場作業能率 .....	23
2.2.2 圃場作業效率 .....	25
2.2.3 하루의 作業시간과 실作業시간을 .....	25
2.2.4 作業적기와 作業가능일수를 .....	26
2.2.5 부담면적 .....	26
2.3 主要 作業別 作業效率分析을 위한 作業패턴과 數學的인 模型開發 .....	29
2.3.1 경운·정지 作業 .....	29
1. 경운·정지 作業의 分類 .....	29
2. 경운·정지 作業기의 선회방법 .....	29
가. 90° 선회 .....	30
1) $\Delta$ 선회 .....	30
2) $\alpha$ 선회 .....	31
나. 180° 선회 .....	31

1) $\Omega$ 선회 .....	31
2) U 선회 .....	32
3. 耕耘整地作業模型開發 .....	33
가. 1행정 왕복순차경법 .....	33
나. 2행정 왕복순차경법 .....	34
4. 作業效率계산을 위한 수학적인 模型 .....	35
가. 作業效率分析模型의 변수설정 .....	36
나. 1행정 왕복순차경법 .....	37
1) 손실시간을 포함한 총작업시간( $T_t$ ) .....	37
2) 순작업시간( $T_f$ ) .....	37
다. 2행정 왕복순차경법 .....	38
2.3.2 收穫作業 .....	39
1. 收穫作業의 分類 .....	39
2. 콤바인 수확작업패턴 .....	40
가. 콤바인의 선회의 종류 .....	40
1) $90^\circ$ $\Delta$ 선회 .....	40
2) $180^\circ$ 선회 .....	40
가) U 선회 .....	41
나) 변형 U 선회 .....	41
나. 콤바인수확작업패턴 .....	42
3. 作業效率分析을 위한 수학적 模型 .....	43
가. 作業效率分析模型의 변수설정 .....	43
나. 손실시간을 포함한 총작업시간( $T_t$ ) .....	44

다. 순작업시간(Tf) .....	45
4. 콤바인 作業能率요인의 포장실험 .....	45
가. 포장실험방법 .....	45
1) 예취작업속도(T0) .....	46
2) 90° Δ선회시간(T1) .....	46
3) 180° U 선회시간(T2) .....	46
4) 변형 180° U 선회시간(T3) .....	46
5) 능률측정실험시의 작업조건 .....	47
나. 능률실험측정결과 .....	47
5. 作業效率과 能率의 分析 .....	49
가. 기종별의 作業效率分析 .....	49
나. 단변을 변화시켰을 때의 각 機種別 作業效率分析 .....	52
다. 동일한 경지규모에서의 장단변비를 변화시킬 때의 作業效率分析 .....	54
라. 콤바인의 作業能率分析 .....	56
 第 3 章 地域別 作業可能日數率 決定 .....	59
3.1 收穫作業의 作業可能日數率 .....	59
3.1.1 收穫作業可能日數決定을 위한 模型開發 .....	60
1. 收穫作業可能日數의 決定基準 .....	60
가. 외국의 收穫作業可能日數決定의 基準 .....	61
나. 收穫作業可能日數 決定模型 .....	61
1) 作業可能日數의 決定基準 .....	61

2) 收穫作業可能日數率의 分析 .....	63
가). 대상지역의 선정 .....	63
나) 氣象資料의 選定 .....	64
다. 作業可能日數率의 分析方法 .....	64
1) 地域別 및 年度別 作業可能日數分析 .....	64
2) 作業可能日數率의 確率分布 分析 .....	65
3) 日數別 確率分布의 統計處理方法 .....	66
2. 地域別 收穫作業可能日數確率分布曲線 .....	67
3. 地域別收穫作業可能日數率 .....	79
第 4 章 農作業計劃의 데이터베이스化 .....	81
4.1 데이터베이스의 概要 .....	81
4.2 데이터베이스化의 目的 .....	82
4.3 데이터베이스의 具現 .....	83
4.4 데이터베이스 構成 .....	84
4.5 데이터베이스化 作業 .....	87
4.5.1 農機械, 農業施設, 裝置에 대한 데이터베이스化 .....	87
4.5.2 栽培作物과 作業體系 데이터베이스化 .....	91
4.5.3 圃場條件 및 土壤條件 데이터베이스化 .....	95
第 5 章 映像處理를 통한 地形情報 獲得 .....	97
5.1 地圖의 映像獲得 .....	97
5.1.1 映像處理 .....	97

5.1.2 디지털 映像 .....	98
5.1.3 映像處理 시스템構成 .....	98
5.1.4 測度 設定(Calibration) .....	100
5.1.5 異值化 .....	100
5.1.6 細線化(thinning) .....	101
5.2 地形情報 獲得 .....	102
5.2.1 境界線地形圖 製作 및 入力 .....	102
5.2.2 獲得可能 地形情報 .....	104
5.2.3 高度粘 入力 .....	107
5.2.4 對象地域만의 映像 抽出 및 細線化 .....	108
5.2.5 各 地域(필지)의 區分 및 面積 計算 .....	109
5.2.6 高度地圖表示 및 匹地別 平均高度 計算 .....	110
5.2.7 데이터 貯藏 및 出力 .....	113
第 6 章 結言 및 2次年度 研究 方向 .....	115
參 考 文 獻 .....	117
附 錄 .....	119

여 백



# 第 1 章 序 言

우리나라 農業은 耕地基盤의 지속적인 造成과 農業機械化의 계속적 발전에 따라 수도작의 機械化가 이미 完成段階에 접어들어가고 있다. 즉, 수도작 일관 작업체계수립의 큰 제약 작업이었던 移秧作業과 收穫作業이 1991달 현재 각기 85% 와 80%의 機械化面積比率을 나타냄으로서 일부의 傾斜地 栽培地域을 제외하고는 機械化營農이 가능하게 되었다.

그러나, 農業機械의 포장작업성 이용 측면에 대한 심도 있는 고려가 미비한 상태로 耕地基盤이 정리된 결과 農路와 耕地形態 및 規模가 農業機械化의 추진에 큰 制約條件이 되고 있는 실정이다. 즉, 農業機械의 자유로운 포장출입을 제약받는 농로가 많으며, 耕地整理가 안된 논은 물론 기정리지역도 구획의 협소, 경지의 분산 등의 여건때문에 기계의 利用效率이 낮아질 수 밖에 없다. 따라서, 전국적으로 불경지의 면적이 해마다 증가하고 있으며 農業機械의 能率的, 效率的 運用에 문제를 제기하고 있다.

따라서, 耕地基盤의 개선없는 農業機械化의 계속적인 투입은 기계의 이용효율의 감소를 더욱 가속시키고, 기계이용비용을 크게하여 기계화의 장점을 살리지 못할 가능성마저 있다고 할 것이다. 더욱이 작업능률의 향상과 노동력의 절감을 기하기 위하여 대형기계의 보급이 확충되고 있는 현시점에서, 이러한 耕地基盤조성에 있어서의 문제점은 農業機械 투자효과를 감소시키게 될 것이므로 耕地基盤의 조속한 개선이 더욱 절실하다고 할 것이다.

우리과 비슷한 일본에서 막대한 투자에 의하여 점차 대구역포장정비를 서두르고 있는 이유도 여기에서 연유되어 진다고 볼 수 있다. 우리나라에서도 大區劃 圃場整備計劃을 수립하고 있으며 이 분야의 소요기술의 연구개발을 서두르고 있

는 것은 다행한 일이라고 아니 할 수 없다.

農業機械化의 效率을 極大化할 수 있는 耕地整理를 고려함에 있어서 다음 몇 가지를 고려하여야 할 것이다. 첫째로, 경지의 구역이 크면 클수록, 또 장단변비가 크면 클수록 기계의 이용효율이 일반적으로 커진다는 것은 상식상의 말이 될 수 있다. 그러면, 기계효율을 극대화 할 수 있는 경구-포구-농구로 이어지는 대구획으로 耕地基盤을 조성할 수 있는 면적비율이 우리나라에 과연 얼마나 존재하는가?

일부의 대평야지, 간척지, 등을 제외하고는 그런 지역을 쉽게 찾아볼 수 없다. 따라서, 耕地基盤의 조성은 대구역은 물론이고 우리나라에 널리 산재하는 평탄지, 경사지를 대상으로 하고 이에 대한 적정기술의 개발이 요구된다고 할 수 있다.

둘째는, 耕地基盤의 적정 설계기준을 마련함에 있어서 農業機械化의 이용효율뿐만 아니라 지형의 경사도, 기반조성경비, 용배수조작의 편의성, 사회경제적요건 등도 고려하여야 한다는 점이다. 따라서, 耕地整理의 計劃은 하나의 종합적 기술이 될 수 밖에 없으며, 일본의 농업기술센터에서 農業機械, 농업토목, 토양, 작물 분야 등의 전문인이 참가하는 제 3연구팀을 만들어 장기간에 걸친 연구사업을 추진하고 있는 이유도 여기에서 찾아 볼 수 있을 것이다. 그러나, 본 연구에서는 경지구역의 기본설계기술을 제공한다는 측면에서, 주어진 지역의 農業機械 체제 및 작업효율, 지형의 형상 및 경사도, 용배수로조건 등의 주로 農業機械 투입효과 제고를 위한 기술적인 제반인자만을 고려한 경지의 설계기준을 연구대상으로 삼았다.

특히, 耕地整理에 관한 農業機械化 관련정보는 우리나라에서 아직 체계적으로 연구되어 제시한 바 없다. 우리나라의 농업적 제반특성을 고려함이 없어 외

국의 관련정보를 그대로 원용하는 것은 결코 바람직하지 못하다고 할 것이다. 따라서, 본 연구는 農業機械化의 관련정보를 체계적으로 제시하여 기계화의 효율과 능률을 극대화할 수 있는 자료를 제공하고, 아울러 이러한 기계화 연관자료가 耕地整理의 計劃 수립에 어떻게 활용할 것인가를 찾아내고자 하였다. 여기에 부가하여, 기계화의 여건을 판단하고 耕地整理 시행의 기초가 될 수 있는 未耕地整理지역의 지형과 경지의 형태를 알아낼 수 있는 지형의 정보가 요구된다. 경지의 형태는 주어진 지역의 필지별 면적과 고저차, 그의 분포등이 포함될 수 있다. 이러한 지형에 대한 정보가 알려질 수 있다면 耕地整理의 확정설계 전단계로서 대체적인 전체 윤곽과 소요공사비의 추정에도 기여하게 될 것이다. 따라서, 本研究內容의 일부로서 映像處理技法을 이용하여 획득가능한 지형정보를 찾아내고 이것을 耕地整理 기본計劃 수립에 활용할 수 있는 방법을 제시하고자 하였다.

기계화를 위한 耕地基盤을 조성함에 있어서, 특히 강조할 것은 각 지역 특성에 맞는 기반 조성을 해야 한다는 점이다. 또한, 耕地整理대상지역의 경제적, 지리적, 사회적 특수성 등을 감안하여 토지이용 및 機械投入效率을 높일 수 있는 경지정비 설계기준을 제시해야 한다는 것이다. 이러한 문제의 해결책은 일본 등 외국의 사례와 우리가 하고 있는 시행착오적 방법 등을 참고로 하여 지형정보데이터, 農業機械 作業效率分析, 農業機械 데이터베이스 등을 綜合하여 耕地整理 전문가시스템을 構築함으로써 가능해질 수 있다고 생각된다.

따라서, 이러한 耕地整理 전문가시스템은 耕地整理對象地域의 모든 제반여건 및 정보를 고려하여 지역특성에 맞는 가장 적절한 경지정비를 위한 耕地形狀 및 規模, 耕地傾斜度, 機械化作業體系 및 豫想作業效率, 用排水路方向, 農路形態 등과 같은 基礎設計資料를 제공해 줄 수 있다.

그림 1-1은 本研究에서 수행되는 경지정비 기초설계기술 제시를 위한 전체 시스템 구성을 나타내고 있다. 이러한 전체 시스템 구성은 소형 컴퓨터를 이용하

용하여 가능하게 할 계획이며 耕地整理 전문가시스템을 컴퓨터 그래픽스와 연결 시켜 시각적으로 圖示하는 연구가 앞으로 계속 수행되어야 할 것이다.

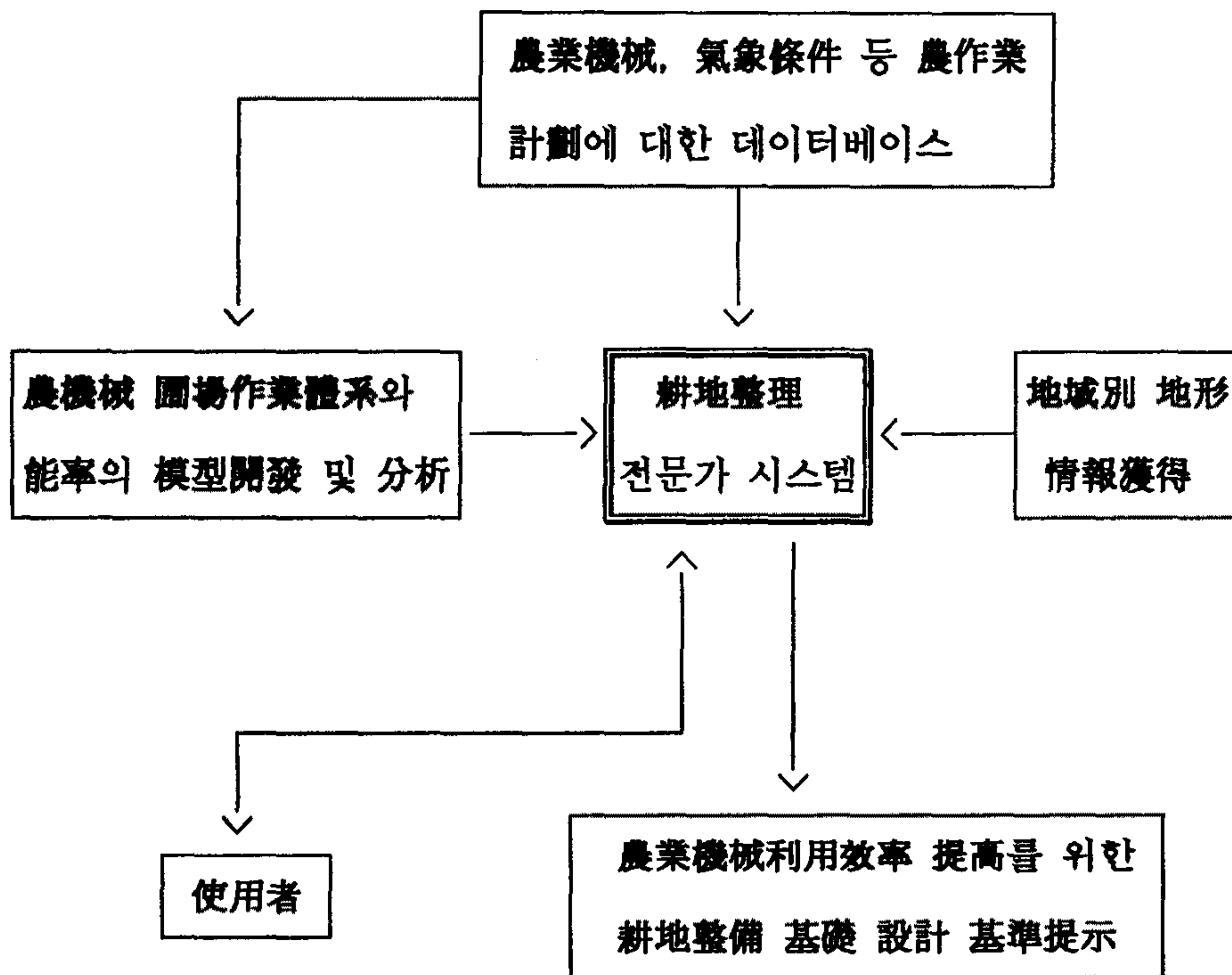


그림 1 - 1. 耕地整理 基礎設計設定을 위한 전문가시스템 構成圖

## 第 2 章 耕地規模 및 作業別 圃場作業效率의 分析

### 2.1 耕地整理에 關聯된 農業機械化 變數

耕地整理를 計劃함에 있어서, 農業機械化의 측면에서 고려되어야 할 變數가 무엇인가를 우선 規定할 필요가 있다. 그것은 다음 몇 가지로 要約하여 나타낼 수 있다. 즉,

① 作業別 所要機械의 種類와 能率 : 같은 작업에 사용되는 기계라고 하더라도 動力源으로는 동력을 달리하는 엔진, 모터, 보행용트랙터, 승용트랙터 등이 있고, 作業機로는 이들 原動機에 장착되는 作業幅을 달리하는 각종 작업기가 있다. 주어진 耕地의 規模와 條件에 대한 諸員, 특히 작업기의 작업폭과 走行速度에 관한 情報가 필요하다.

② 각종기계작업에 있어서 경지의 규모별, 형태별로 기계의 圃場作業效率과 포장작업능률 : 이것은 포장작업효율과 포장작업량 이라는 지표로 나타낼 수 있다.

③ 경지의 분산에 따른 이동거리와 이동빈도, 농업기계의 격납고에서 경지까지의 주행 또는 운반거리, 농로의 조건, 작업기착탈 등 포장밖에서의 손실시간, 이것은 실작업률이라는 지표로 나타낼 수 있다.

④ 농작업이 기계고장, 강수, 토양의 지내력 등 자연적 환경조건에 의하여 작업이 불가능한 일수 : 이것은 직접 경지의 규모나 조건에는 큰 관계는 없으나 농업기계의 작업가능면적을 추정하는데 필요한 정보이며 보통 작업가능일수율로



규정한다. 작업가능일수율은 작업별, 시기별, 지역별로 규정되어야 한다.

⑤ 작업적기종의 작업기간 : 주어진 작업은 작업적기가 존재하며, 그 작업적기종의 작업기간은 기계작업의 적기수행을 위해 구명되어야 한다. 이것은 영농규모의 바탕이 되는 부담면적결정의 기초적인 지표이다.

⑥ 작업기의 주행가능거리 : 작업기에 따라 작업중 자재의 그 적재량에 대한 제약때문에 증도에 보급 또는 하역을 하지 않고 주행이 가능한 거리가 있다. 자재의 보급과 하역은 지선농로에 인접하거나 개자리에서 이루어지는 것이 일반적이기 때문에 관련작업기별 최대 주행가능거리는 경지의 구획결정의 참고자료가 된다.

이러한 농업기계화의 제반 관련변수는 耕地整理계획의 기초자료일 뿐아니라 농업기계의 선택, 효율적이용을 위해 구명되어야 한다. 따라서, 本 研究에서는 ① 작업별로 현실적으로 이용가능한 농업기계나 앞으로 이용가능한 각종 기계에 대한 관련 정보를 Database로 구축하고, 수도작을 중심으로한 ② 각작업의 작업방법을 모향화하였으며, ③ 작업별, 경지규모 및 형태별의 圃場作業效率과 포장작업량을 규정하는 일차적작업으로서 콤바인수확작업의 圃場作業效率과 포장작업량을 추정하여 나타내었으며, ④ 부담면적결정의 기초자료가 되는 작업가능일수율에 대하여 우선 전국의 주요지역별로 콤바인 수확작업가능일수율을 分析하였다.

## 2.2 農作業計劃樹立의 關聯變數

### 2.2.1 圃場作業能率

포장에서 농기계를 사용하여 농작업을 수행할 때 포장에서 작업기가 단위시간에서 본래의 목적을 수행하며 작업할 수 있는 작업량을 圃場作業能率이라 한다. 이러한 圃場作業能率은 성능이 우수한 농기계의 개발·개량이나 작업방법의 개선 또는 기계의 구입계획이나 농업기계화계획을 수립하는 지표가 된다.

대부분의 농기계에 있어서의 作業能率은 단위시간에 수행할 수 있는 작업면적 즉, ha/hr 로서 표시된다. 하지만, 목초수확기나 콤바인과 같은 수확용기계 등은 단위시간에 수확할 수 있는 곡물의 무게 즉, ton/hr 로서 나타낼 수 있다. 이와는 반대로 단위면적을 작업하는 데 걸리는 시간 즉, hr/ha 또는 hr/10a 을 사용하여 표시할 수 있다. ha/hr 와 hr/ha 는 역수 관계에 있기 때문에 쉽게 환산이 가능하다. 기계의 성능이나 작업별 부담면적을 산출하는 경우에는 ha/hr 을, 기계이용경비나 기계이용계획의 설계 등에는 hr/ha 을 이용하는 것이 더 용이하다.

圃場作業能率은 기계의 종류, 형식, 크기 또 작업시의 기상 조건, 포장 조건, 작물 조건, 기계 운전 조건, 운전자의 조건 등에 영향을 받는다. 또한, 원하는 작업의 質에 있어서 요구하는 수준이 달라질 때 일반적으로 동일한 작업 환경하에서는 質이 높아지면 圃場作業能率은 저하되는 관계가 있다.

## 1. 理論圃場作業能率

작업기가  $W$ 의 이론작업폭과  $S$ 의 이론작업속도로 1시간을 연속적으로 작업할 때 산출되는 작업면적을 理論圃場作業能率(theoretical field capacity)이라 한다. 理論圃場作業能率は 손실이 전혀 없는 작업조건을 가정하였기 때문에 한 작업기에 대한 최대의 圃場作業能率이 된다. 理論圃場作業能率을 수식으로 표현하면 식(2-1)과 같다.

여기에는 주목해야 할 점이 있다. 작업폭과 작업속도를 적용하는 데 있어서 학문적인 혼돈이 일어날 수 있다. 本 研究에서는 이론작업폭과 이론작업속도를 다음과 같이 정의하였다.

이론작업폭은 작업기가 본래의 목적을 적절히 수행하며 작업을 할 때의 작업폭을 말한다. 실제 포장에서의 적정작업폭을 적용하는 것이 원칙이지만 本 研究에서는 일반적으로 작업기의 제원에 나타나 있는 수치를 이론작업폭으로 적용을 하였다.

$$C_t = \frac{1000 W S_h}{10000} = 0.1 W S_h = 0.36 W S_s \quad (2-1)$$

여기서,  $S_h$  : 이론작업속도 ( $km/hr$ )

$S_s$  : 이론작업속도 ( $m/sec$ )

$W$  : 이론작업폭 ( $m$ )



포장에서의 실작업폭은 이론작업폭과 상당한 차이가 있다. 이것은 포장의 조건과 작업시의 제반 작업조건에 의해서 달라질수 있다. 따라서, 이론작업폭과 실제작업폭과의 비를 실작업폭률로 정의하고 실작업폭은 이론작업폭과 실작업폭률로서 나타낸다.

이론작업속도의 결정에 있어서도 고려해야 할 사항이 많다. 작업속도는 운전자의 의지에 따라 선택할 수 있는 폭이 아주 넓고 또한 기종에 따라서도 상당히 다르다. 따라서, 本 研究에서는 토양·작물조건, 사용자의 숙련도 등 모든 조건을 고려하여 최고의 작업정도를 발휘할 수 있는 최대속도를 이론작업속도로서 정의한다. 여기에서, 적정한 작업속도를 선택하는 것은 아주 중요하다.

## 2. 有效圃場作業能率

기계가 실제로 작업을 수행하는 1시간당의 평균면적을 有效圃場作業效率 (effective field capacity)이라고 말한다.

농업기계가 그 企圖하는 기능을 실제로 발휘하는 시간을 有效작업시간 (effective operating time)이라고 한다. 예를 들면, 트랙터에 끌리는 플라우를 생각하면 새머리에서 회전하는 사이에는 흙을 경운하는 작업을 수행할 수 없으며, 직진할 때만 작업이 이루어진다. 따라서, 1ha당 有效작업시간은 이론작업량에 相應되는 이론작업시간보다 더욱 커진다.

유효포장능률이 이론포장능률보다 작은 것은 포장에서 손실시간이 있기 때문이며, 또 규정된 작업기의 폭보다 감소된 폭으로 작업하기 때문에 나타난다. 理論作業能率에 대한 유효포장능률의 비율을 포장효율(field efficiency)이라고 한다.

어떤 기계의 포장작업량은 그 기계의 작업폭, 작업속도, 작업 중 손실시간 및 작업 중 작업폭의 중복되는 정도 등에 영향을 받는다. 콤바인 · 목초수확기 · 플라우 등과 같은 작업기는 작업의 누락을 감수하지 않는한 전체의 작업폭을 실제작업에서 전부 이용한다는 것은 대단히 어렵고, 따라서 지면이 고르며, 운전자가 숙련되어 있다고 하더라도 작업폭이 약간 중복되면서 작업이 이루어지는 것이 일반적이다. 최대의 허용작업속도, 즉 이론작업속도는 작업의 종류, 포장조건, 이용하는 원동기의 크기 등에 영향을 받는다.

작업시간 중의 손실시간은 규정하기 어려운 변수이다. 이것은 기계의 조정, 윤활유의 급유, 고장·침지에서의 선회, 수확물의 하역, 비료 또는 농약의 공급 등의 여러가지 이유 때문에 나타난다.

포장작업기의 유효圃場作業能率은 다음과 같이 나타낼 수 있다.

$$C_e = 0.1 \varepsilon_f W S_h = 0.36 \varepsilon_f W S_s \quad (2-2)$$

여기서,  $C_e$  : 유효포장작업능률 ( $ha/hr$ )

$\varepsilon_f$  : 포장작업효율 (소수로표시)

$S_h$  : 이론작업속도 ( $km/hr$ )

$S_s$  : 이론작업속도 ( $m/sec$ )

$W$  : 이론작업폭 ( $m$ )

식(2-2)에서 0.1과 0.36은 단위의 변환에서 비롯된 상수들이다. 또한,  $\varepsilon_f = 1$  일 때  $C_e$  는 理論圃場作業能率이 된다는 것을 알 수 있다.

## 2.2.2 圃場作業效率

실제 작업에 있어서 선회와 같은 손실시간을 고려한 유효圃場作業能率は 손실시간을 고려하지 않은 理論作業能率보다 작게 나타난다. 각 농작업에 있어서 理論圃場作業能率에 대한 유효圃場作業能率에 대한 비를 圃場作業效率(field efficiency)이라고 말한다. 圃場作業效率을 다음과 같은 식(2-3)으로 나타낼 수 있다.

$$\varepsilon_f = \left( \frac{C_e}{C_t} \right) 100 \quad (2-3)$$

여기서,  $\varepsilon_f$  : 포장작업효율 (%)

$C_e$  : 유효포장능률 (ha/hr)

$C_t$  : 이론포장능률 (ha/hr)

## 2.2.3 하루의 작업시간과 실작업시간을

농작업을 수행하는 데 있어 하루에 작업을 할 수 있는 최대의 시간이 존재한다. 하루의 작업가능시간은, 주로 일조시간과 관계가 있다. 또한, 작업별로도 약간의 차이는 있다. 대부분의 농작업은 일몰이 지난후에는 포장에서의 작업은 거의 하지 않는다. 따라서, 농작업별 및 지역별로 일작업가능시간이 존재한다.

실작업시간을이란 하루의 작업가능시간대한 실제로 수행한 작업시간의 비율을 말한다. 실작업시간율에 영향을 미치는 요인으로서는 포장의 분산에 따른포장

과 포장사이의 이동시간, 기계고장시간, 격납고에서 포장까지의 거리, 휴식시간 등과 같은 요소들이 있다.

## 2.2.4 작업적기와 작업가능일수율

농업생산의 특수성에 기인하여 각 농작업별로 작업이 이루어져야 하는 적절한 시기가 존재한다. 이와 같은 각 농작업이 수행되어야 할 적절한 시기를 작업적기(timeliness)라고 한다. 따라서, 이러한 작업적기에 작업을 수행하지 못하였을 때는 수확량의 감소 등과 같은 크고 작은 손실이 뒤따르게 마련이다.

또한, 대부분의 포장에서의 농작업은 기상조건에 많은 영향을 받는다. 따라서, 작업적기중에서 기상조건으로 인하여 작업을 수행하지 못하는 일수가 생기게 마련이다. 주어진 작업적기와 작업적기중에서 작업을 할 수 있는 일수와의 비율을 작업가능일수율이라고 한다.

## 2.2.5 부담면적

농업생산의 한 가지 특징은 농작업의 適期性(timeliness)이 존재하는 데 있다. 농업생산은 공업생산과 같이 연중을 통하여 어느 때나 작업을 수행할 수 있는 것이 아니고 알맞는 작업시기가 있으며, 그 시기를 놓치면 收量의 감소를 초래하게 된다.

따라서, 농업기계화계획을 수립하기 위해서는 사용될 기계의 能率과 작업의

적기성에 바탕을 두고 구해지는 負擔面積을 추정하여 이것을 경영면적과 비교할 필요가 있다. 여기에서, 부담면적이란 농작업을 수행하는 데 작업적기, 기상 등의 제약과 주어진 포장의 조건 하에서 기계의 능력을 충분히 활용할 때의 작업할 수 있는 면적을 말한다.

부담면적과 경영면적이 합치되는 경우가 가장 이상적이며, 어느 한 쪽이 상대적으로 과소하거나 또는 과대할 때에는 불합리한 기계화경영이 불가피하게 된다.

부담면적의 결정에는 기계의 作業能率이 하나의 기초가 된다. 이와 더불어 1일의 작업시간, 포장 외에서의 機械移動에 따른 損失時間, 作業適期 중의 作業日數, 이 기간 중에 강우나 고장 등으로 인하여 작업을 수행하지 못하는 日數 등이 부담면적에 영향을 끼친다.

기계의 고장, 강우로 인한 작업의 불가능, 포장 사이의 이동 등은 부담면적을 감소시킨다. 기계가 하루에 책정된 작업가능시간에 대한 포장 내에서의 작업시간의 비율을 실제작업시간율이라고 한다.

예를 들면, 작업시간 10일 중에서 기계고장이나 기후상의 이유 때문에 작업이 불가능한 날이 2일이라면 작업가능일수율은 80%이다.

本 研究에서는 이러한 모든 조건을 고려한 작업기의 作業能率을 나타내는 부담면적을 계산할 수 있는 수학적인 模型을 개발했다.

부담면적은 다음 식(2-4)와 같은 식으로 계산될 수 있다.

식(2-4)에서 유의해야 할 점은  $\frac{1}{10}\epsilon_{1}SW$ 는 유효作業能率이고,  $\epsilon_{u}U$ 는 하

투의 실제작업시간이며,  $\epsilon_d D$ 는 작업적기 중에 실제로 기계작업이 가능한 일수를 나타낸다는 것이다.

부담면적을 계산하는 식에서의 圃場作業效率( $\epsilon_f$ ), 실제작업시간율( $\epsilon_u$ ), 작업가능일수율( $\epsilon_d$ )은 기계의 능률, 즉 부담면적을 지배하는 중요 요인으로서, 일부 선진국에서는 규정되어 있으며 本 研究에서 이들 효율계수들을 우리나라의 현실에 맞게 실제의 데이터를 이용하고 실제의 포장의 실험을 통하여 규정했다. 계속적으로 연구를 수행하여 각 농작업별로 이러한 효율계수의 규정작업을 완성할 것이다.

$$A = \frac{1}{10} \epsilon_f \epsilon_u \epsilon_d S W U D , ha \quad (2 - 4)$$

여기서,  $A$  : 부담면적 ( $ha$ )

$\epsilon_f$  : 포장작업효율 (소수로나타냄)

$\epsilon_u$  : 실제작업시간율 (소수로나타냄)

$\epsilon_d$  : 작업가능일수율 (소수로나타냄)

$S$  : 이론작업속도 ( $m/sec$ )

$W$  : 이론작업폭 ( $m$ )

$U$  : 1일 작업시간 ( $hr/day$ )

$D$  : 작업가능일수 ( $day$ )



## 2.3 主要 作業別 作業效率分析을 위한 작업패턴과 數學的인 模型開發

### 2.3.1 경운·정지 작업

#### 1. 경운·정지 작업의 분류

경운·정지기계는 토양에 직접 작용하여 토양의 파괴·이동·압축·성형 또는 혼합 등을 수행하는 기계로서 그 작업목적에 따라 다음과 같은 여러가지가 있다.

① 쟁기작업 : 쟁기작업은 쟁기 또는 플라우를 사용하여 굳어진 흙을 절삭·반전·파쇄하여 큰 덩어리로 파쇄하는 작업으로서 이 작업을 보통 1차경(primary tillage)이라고도 한다.

② 쇄토작업 : 쇄토작업은 1차경으로 경기된 흙을 다시 작은 알맹이로 파쇄하는 작업으로서 이 작업을 2차경(secondary tillage)이라고도 한다. 이 작업에는 로터베이터·씨레 및 헤로우가 이용된다.

③ 구동경운작업 : 구동경운작업은 로터리경운으로 1차경과 2차경을 동시에 실시하는 작업으로서 이 작업은 1차경과 2차경을 별도로 하는 방식에 비하면 흙의 반전상태가 좋지 못하다. 이 작업에는 로터베이터가 이용이 된다.

#### 2. 경운·정지 작업기의 선회방법

작업기를 부착한 트랙터가 작업중에 선회를 수행하는 방법에는 크게 90° 선

회와 180° 선회가 있다. 90° 선회는 회전경법에 사용이 되며 180° 선회는 왕복순차경법에 사용이 된다. 손실시간을 줄이기 위하여 작업기가 선회를 시작할 때 작업기를 들어올리고 선회를 끝마치면서 작업기를 다시 내린다.

#### 가. 90° 선회

주로 회전경법을 수행할 때 포장의 모서리에서 사용되는 일반적인 선회방법이며 선회종류는 크게  $\Delta$ 선회와  $\alpha$ 선회 두 종류가 있다.

##### 1) $\Delta$ 선회

이 선회방법은 전진 -> 정지 -> 후진 -> 정지 -> 전진 의 순서로 선회가 이루어진다. 이 방법은 선회할 장소가 좁을 때 주로 사용이 되며 후진으로 인한 정지상태가 존재하기 때문에 시간이 많이 걸리는 단점이 있다. 그 구체적인 방법은 그림에서 보는 바와 같다.

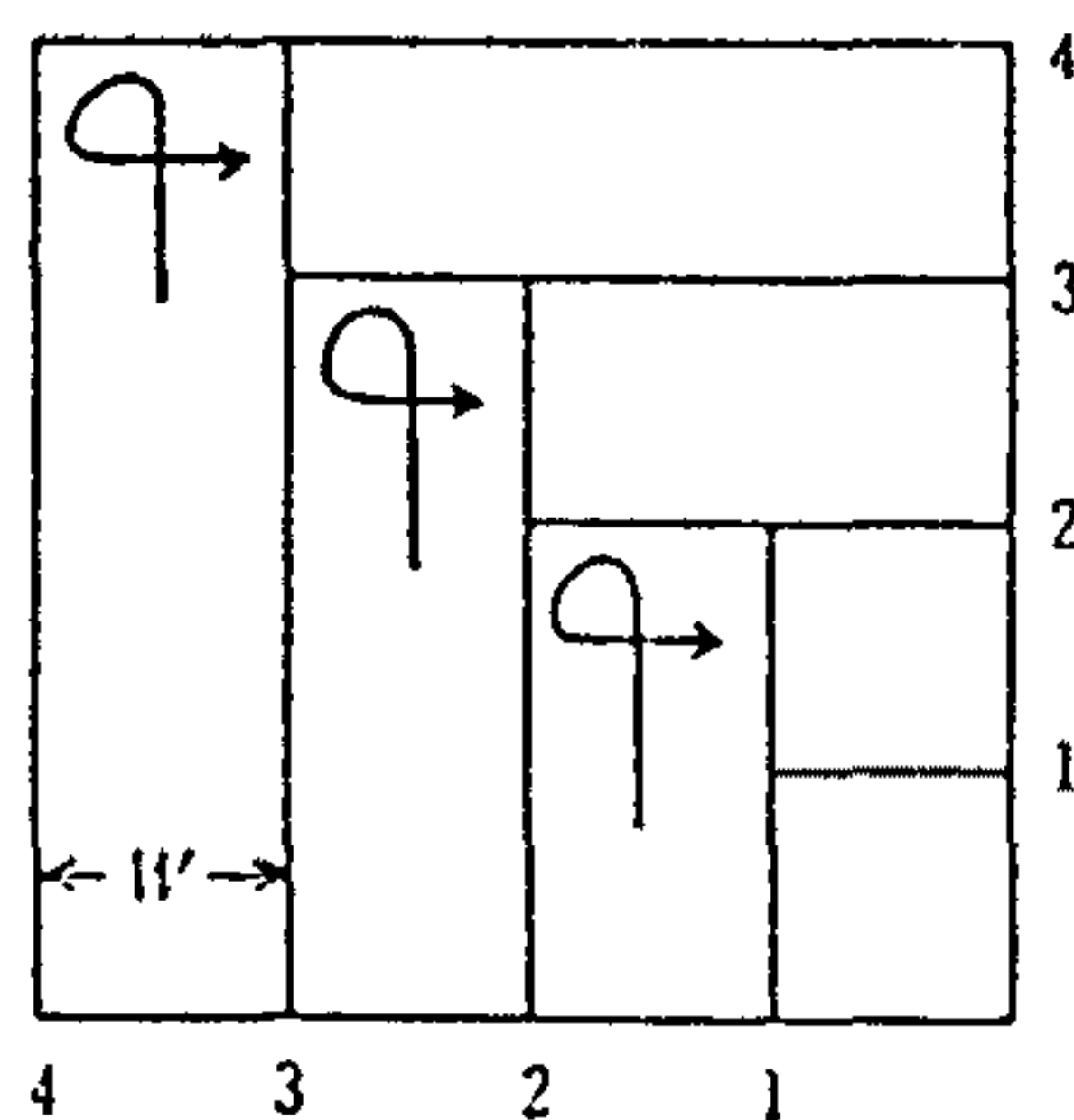


그림 2 - 1. 경운정지작업기의 90°  $\Delta$  선회방법



## 2) $\alpha$ 선회

이 방법은 전진상태에서 선회를 처음부터 끝까지 수행한다. 선회장소가 넓어야 하며 후진이 없기 때문에 시간이  $\Delta$ 선회보다 적게 소요되며 새로운 작업열로 진입할 때 정확히 작업열을 맞출 수 있는 長點이 있다.

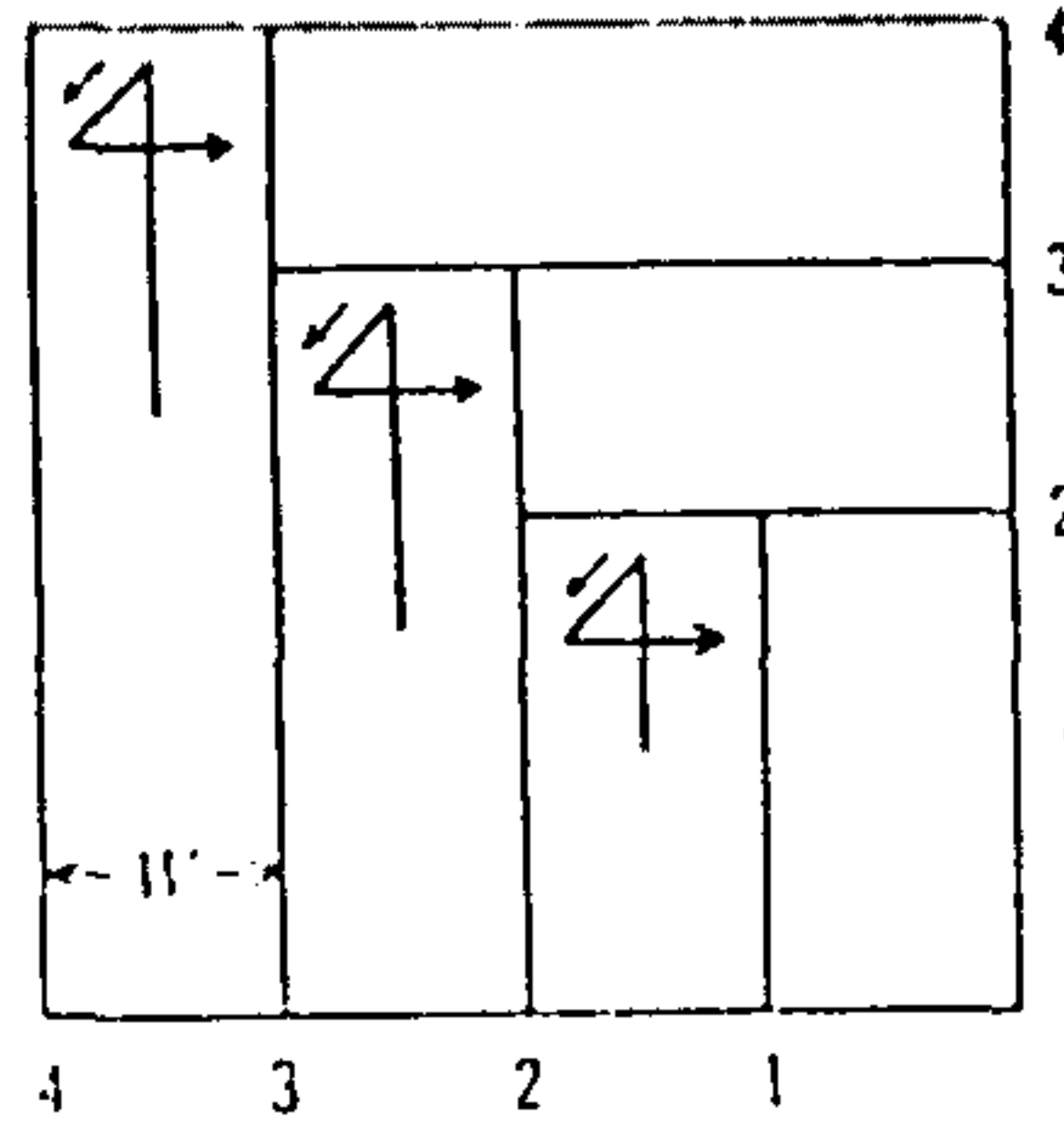


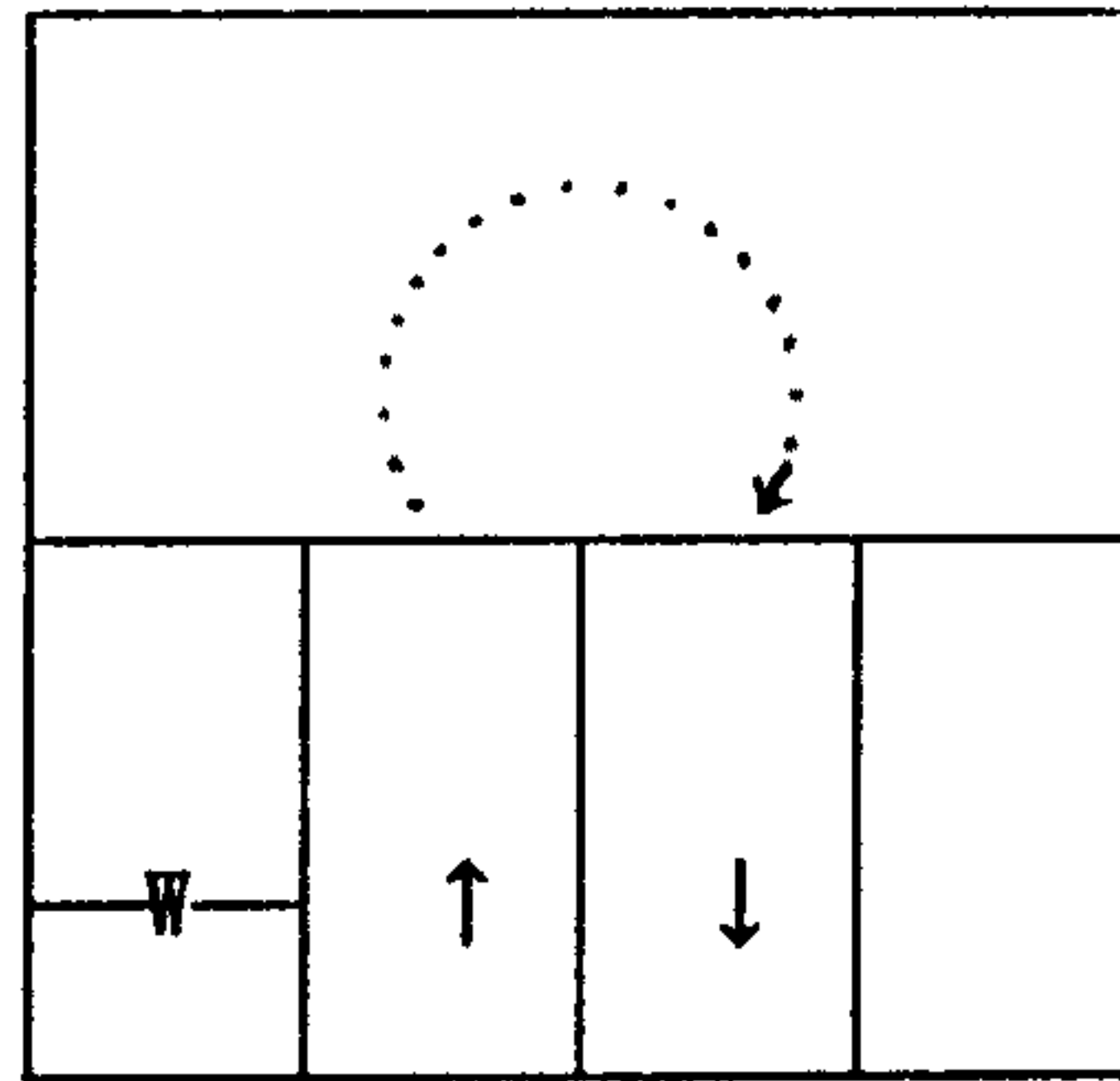
그림 2 - 2. 경운정지작업기의  $90^\circ \alpha$  선회방법

### 나. $180^\circ$ 선회

이  $180^\circ$  선회는 왕복연접하여 경운정지작업을 수행할 때 주로 많이 사용이 되는 방법이다. 선회방법에는 크게 U 선회와  $\Omega$  선회 두 종류가 있다.

#### 1) $\Omega$ 선회

$\Omega$ 선회는 U선회가 불가능할 때 불가피하게 사용이 된다. 구체적인 방법은 그림 2-3에서 보는 바와 같다.

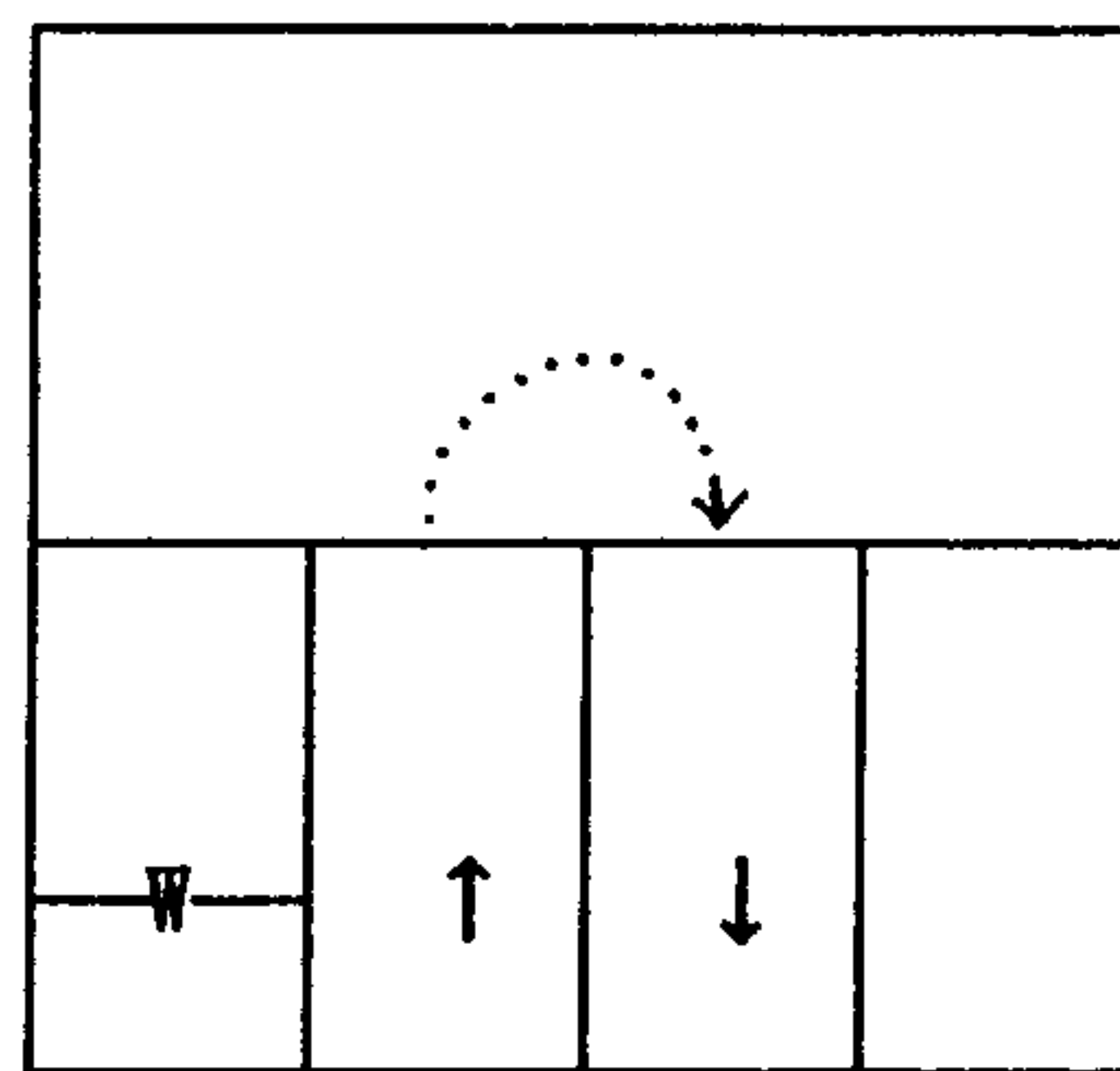


→ 는 작업진행방향

그림 2 - 3. 경운정지작업기의 180° Ω 선회방법

2) U 선회

이 선회방법은 회전반경이 작은 작업기와 회전반경이 충분한 곳에서의 180° 선회에 사용이 된다. 구체적인 방법은 그림 2-4에서 보는 바와 같다.



→ 는 작업진행방향

그림 2 - 4. 경운정지작업기의 180° U 선회방법

### 3. 耕耘整地作業模型開發

트랙터를 이용하여 경운·정지 작업을 수행할 때의 작업방법은 부착작업기의 종류에 따라서 다르게 나타난다. 토양의 반전방향이 한쪽으로만 가능한 단용플라우를 제외한 양용플라우나 로터베이터 등과 같은 작업기는 동일한 작업방법을 사용한다. 전자는 주로 회전경법을 사용하고 후자의 작업방법은 회전경법과 왕복경법의 복합적인 형태이다. 가장 큰 차이점은 전자는 연접하여 작업을 할 수 없는 반면에 후자는 토양의 반전방향에 관계가 없기 때문에 연접작업이 가능하다. 현재, 대부분의 작업은 양용플라우나 로터베이터 등을 사용하여 경운·정지 작업을 수행한다. 따라서, 본 研究에서는 양용플라우와 로터베이터에 초점을 맞추어 연구를 수행했다.

트랙터를 이용한 경운작업방법을 크게 1행정 왕복순차경법, 2행정 왕복순차경법 이 두 가지의 대표적인 유형으로 분류하였다.

#### 가. 1행정 왕복순차경법

이 작업방법은 중형이하의 트랙터에 사용되는 작업방법으로서 바로 인접한 작업열로 180° U 선회가 가능하도록 작업기의 회전반경이 작은 경우에 사용하는 방법이다.

작업 방법은 처음 시작할 때 그림 2-5에서 보는 바와 같이 회전하면서 경운을 한다. 직각모서리에서는 주로 90°  $\Delta$  선회를 한다. 또한, 90°  $\alpha$  선회를 사용하기도 한다. 그 다음에는 바로 인접한 작업열로 180° U 선회를 하여 순차적으로 왕복경운을 수행한다. 가장자리의 경운작업열은 작업을 마무리 지으면서 가

장 마지막에 수행한다.

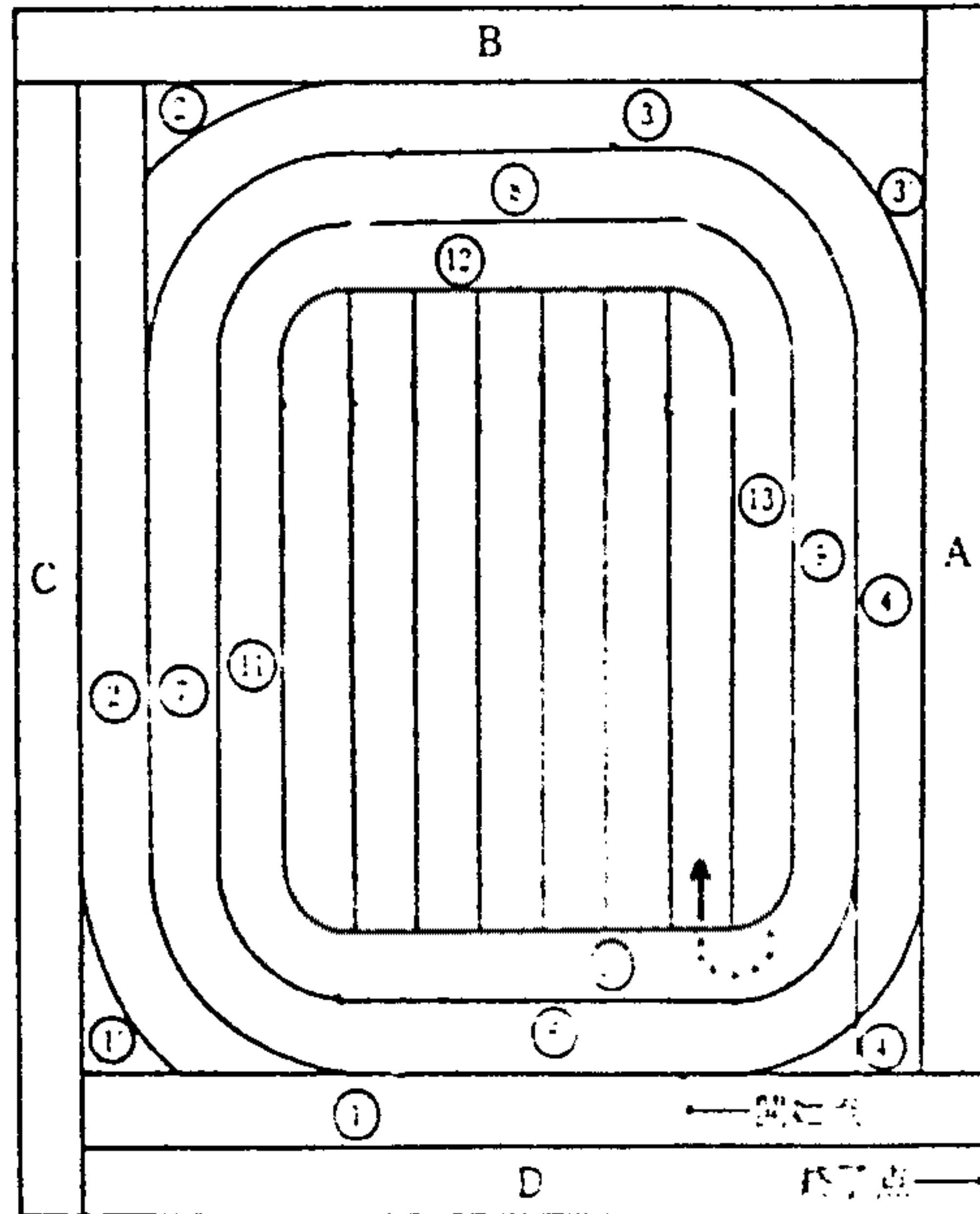


그림 2 - 5. 1행정 왕복순차경법의 작업패턴

나. 2행정 왕복순차경법

이 작업방법은 중형이상의 트랙터에 사용되는 작업방법으로서 작업기의 회전반경이 작업폭보다 커서 바로 인접한 작업열로 180° U 선회가 불가능한 경우에 사용하는 방법이다.

작업방법은 처음 시작할 때 그림 2-6에서 보는 바와 같이 회전하면서 경운

을 한다. 이 때에는 주로  $90^\circ$   $\Delta$  선회를 한다. 그 다음에는 하나의 작업열을 건너뛰어서 그 다음작업열로  $180^\circ$  U 선회를 하여 순차적으로 왕복경운을 수행한다. 이 경법에서는 그림에서 보는 바와 같이  $180^\circ$   $\Omega$  선회를 불가피하게 한번 수행한다. 가장자리의 경운작업열은 작업을 마무리 지으면서 가장 마지막에 수행한다.

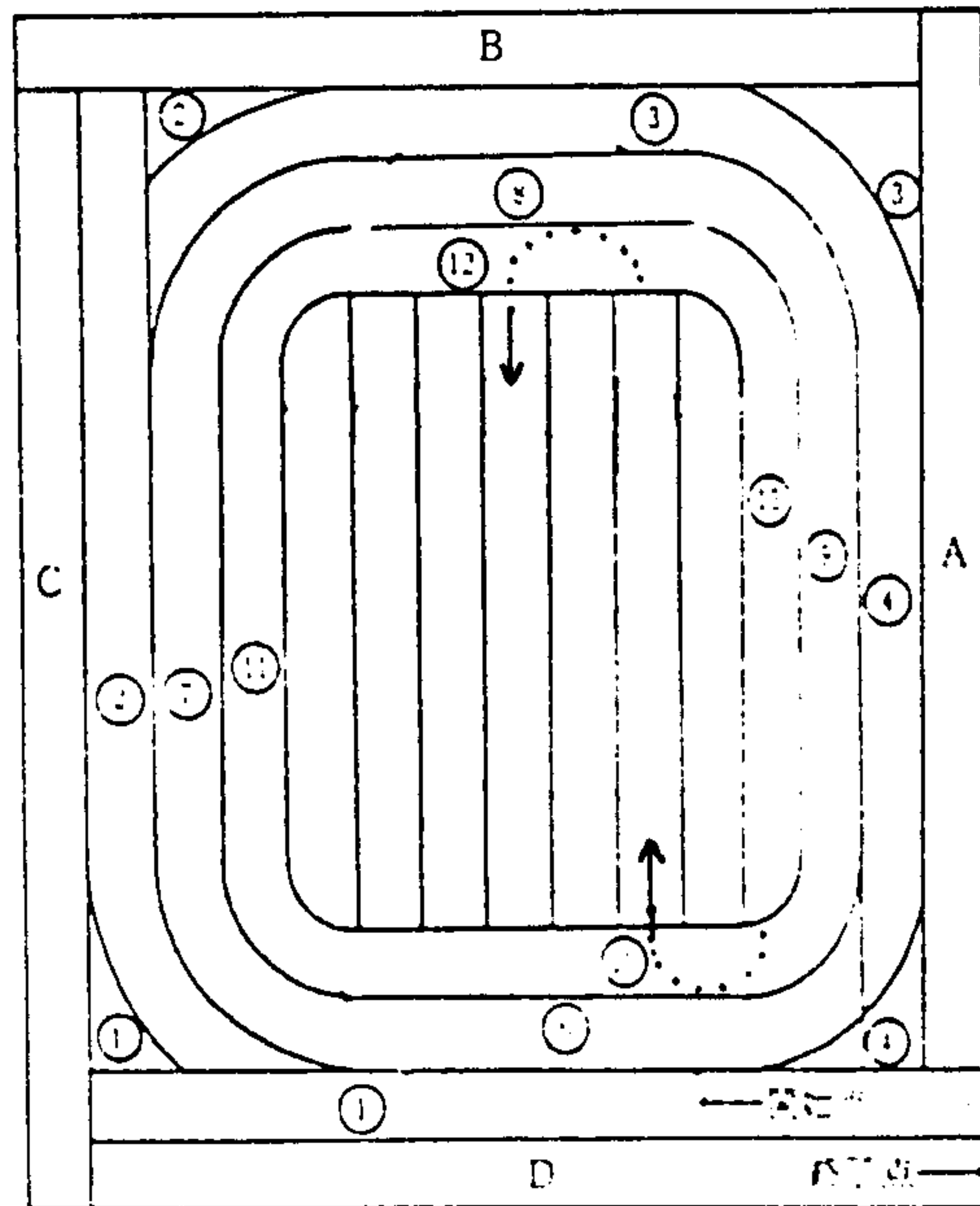


그림 2 - 6. 2행정 왕복순차경법의 작업패턴

#### 4. 作業效率계산을 위한 수학적인 模型

作業效率은 선회시간과 같은 손실시간을 포함한 전체경운작업에 소요되는 시간과 경운작업에만 소요되는 시간과의 비로서 나타낼 수 있다. 따라서, 전체경

운작업에 소요되는 시간과 경운작업에만 소요되는 시간을 계산하기 위한 수학적  
인 模型이 개발했다.

#### 가. 作業效率分析模型의 변수설정

이 模型에서는  $180^\circ$  U 선회와  $90^\circ$   $\Delta$  선회에 걸리는 시간만을 손실시간으  
로 간주했다. 유헴유의 공급, 작업자의 교체, 기계의 고장수리시간등은 고려하지  
않았다. 이러한 시간들은 명확하게 수식화하여 규정화하기 매우 힘든 변화폭이  
아주 심한 변수들이기 때문이며 또한, 本 研究에서 究明하고자 하는 경지규모별  
作業效率의 分析과 독립적인 요소들이기 때문이다.수학적인 模型에 필요한 주요  
변수들은 다음과 같이 설정하였다.

X (m) : 포장의 장변의 길이

Y (m) : 포장의 단변의 길이

V (m/sec) : 작업기의 평균작업속도

W (m) : 실작업폭

$M_w$  : 실작업폭 효율계수

W' : 작업기의 이론 작업폭

M : 왕복경법에서의 작업행정수

N : 회전경법에서의 작업 행정수

T<sub>1</sub> (sec) :  $180^\circ$  U( $\Omega$ ) 선회 시간

T<sub>2</sub> (sec) :  $90^\circ$   $\Delta$ ( $\alpha$ ) 선회 시간

$L_T$  (m) : 작업기의 전장

$N_L$  : 작업기 전장에 대한 개자리폭 계수

### 나. 1행정 왕복순차경법

#### 1) 손실시간을 포함한 총작업시간( $T_t$ )

다음 식(2-5)는 손실시간을 포함한 전체경운작업시간을 계산하기 위한 수학적 모델이다.

$$T_t = \frac{XY}{VW} + T_1(M-2N-1) + 4NT_2, \text{ sec} \quad (2-5)$$
$$= \left( \frac{XY}{VW} + T_1(M-2N-1) + 4NT_2 \right) / 3600, \text{ hr}$$

여기서,  $\frac{XY}{VW}$  : 경운 작업에 소요되는 시간, sec

$T_1(M-2N-1)$  : 180° 선회에 소요되는 시간, sec

$4NT_2$  : 180° 선회에 소요되는 시간, sec

$$M = \frac{Y}{N_w W'}$$

$$N = \frac{N_L L_T}{M_w W'}$$

$$W = M_w W'$$

#### 2) 순작업시간( $T_f$ )

다음 식(2-6)은 경운작업만에 소요되는 시간을 나타내며 圃場作業效率은 식(2-7)에 의해 계산된다.

$$T_f = \frac{XY}{VW} \quad , \text{sec} \quad (2-6)$$

$T_f$  : 경운작업에만 소요되는 시간 (sec)

### 3) 作業效率( $\epsilon_f$ )

作業效率은 총작업시간에 대한 순작업시간의 비로서 나타낼 수 있다.

$$\epsilon_f = \left( \frac{T_f}{T_t} \right) \times 100 \quad (2-7)$$

$\epsilon_f$  : 포장작업효율 (%)

### 다. 2행정 왕복순차경법

수학적인 模型에 필요한 주요변수들은 1행정 왕복순차경법과 같고 다만,  $180^\circ$   $\Omega$  선회에 소요되는 시간이 추가된다.

#### 1) 손실시간을 포함한 총작업시간( $T_t$ )

다음 식(2-8)는 손실시간을 포함한 전체경운작업시간을 계산하기 위한 수학적 模型이다.

식(2-6)에 의해서 작업에만 소요되는 시간이 계산되며 圃場作業效率은 식(2-7)에 의해 계산된다.



$$T_t = \frac{XY}{VW} + T_1(M-2N) + T_2(4N-1) + T_3, \text{ sec} \quad (2-8)$$

$$= \left( \frac{XY}{VW} + T_1(M-2N) + T_2(4N-1) + T_3 \right) / 3600, \text{ hr}$$

여기서,  $\frac{XY}{VW}$  : 경운 작업에만 걸리는 시간 (sec)

$T_1(M-2N)$  : 180° U 선회에 걸리는 시간 (sec)

$T_2(4N-1)$  : 90° Δ 선회에 걸리는 시간 (sec)

$T_3$  : 180° Ω 선회에 걸리는 시간 (sec)

$$M = \frac{Y}{N_w W'}$$

$$N = \frac{N_L L_T}{M_w W'}$$

$$W = M_w W' \quad (m)$$

## 2.3.2 收穫作業

### 1. 收穫作業의 分類

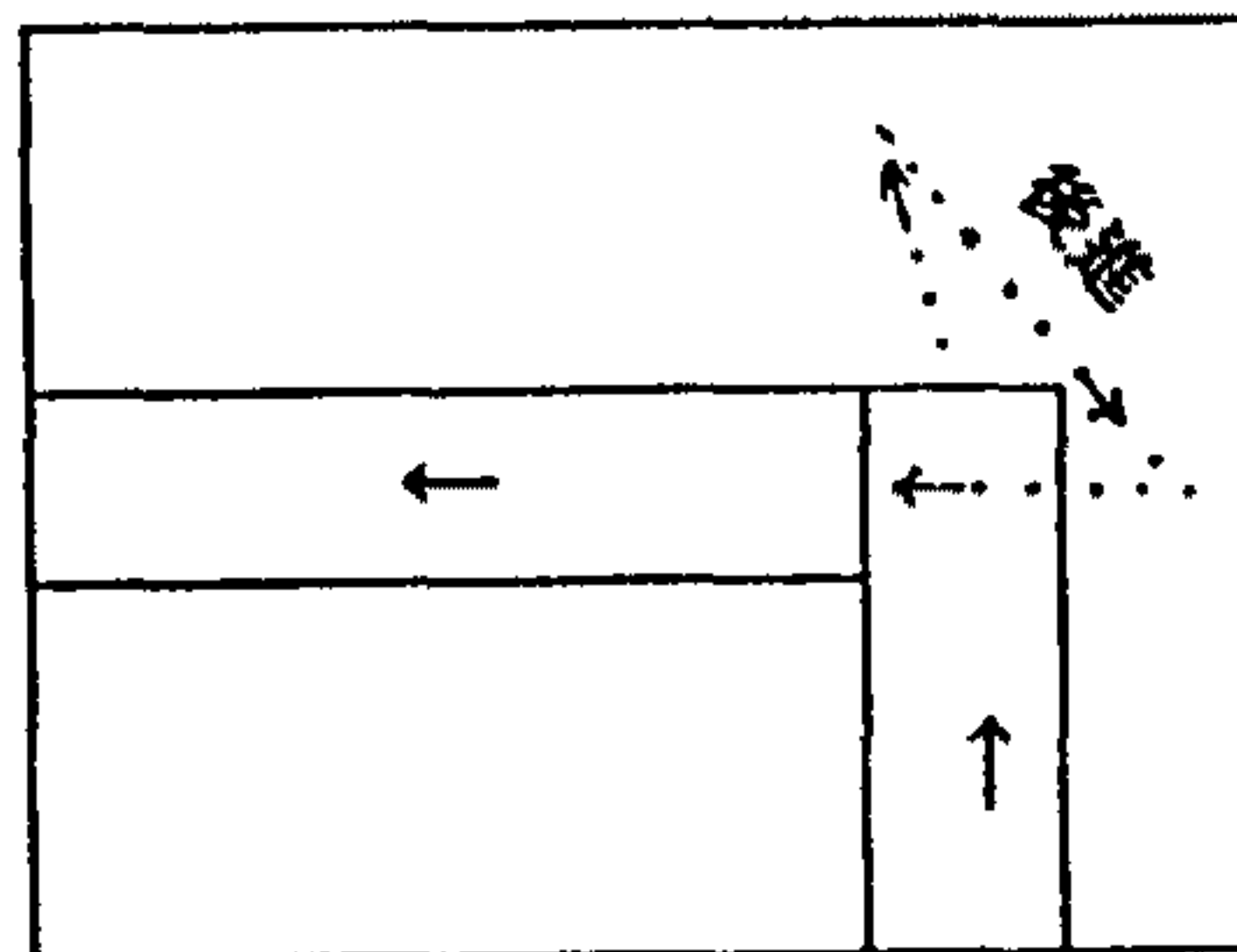
곡물의 수확작업은 예취작업에서 탈곡작업까지의 일련의 작업을 말한다. 관행의 수확작업체계에서는 예취와 탈곡을 독립적으로 수행한다. 현재의 기계수확작업체계에서 수확작업의 주종을 이루고 있는 콤바인은 예취와 탈곡을 동시에 수행을 한다. 일부에서는 아직도 바인더를 사용하고 있다. 이것은 콤바인이 작업을 수행할 수 없는 지역적인 특성수성에 기인한다. 따라서, 本 研究에서는 수확작업效率分析의 고려대상을 콤바인에 국한한다.

## 2. 콤바인 수확작업패턴

### 가. 콤바인의 선회의 종류

#### 1) 90° △ 선회

콤바인이 회전에취를 수행하기 위해서는 직각모서리에 있는 개자리에서 90° 선회를 수행해야 한다. 이 때에 그 선회의 模型은 그림 2-7에서 보는 바와 같이 △형 선회를 한다. 전진선회 -> 정지 -> 후진 -> 정지 -> 전진 의 순서로 진행이 된다. 2조형, 3조형 및 4조형 과 같은 모든 종류의 콤바인이 이와 같은 선회를 수행한다.



→ 는 작업진행방향

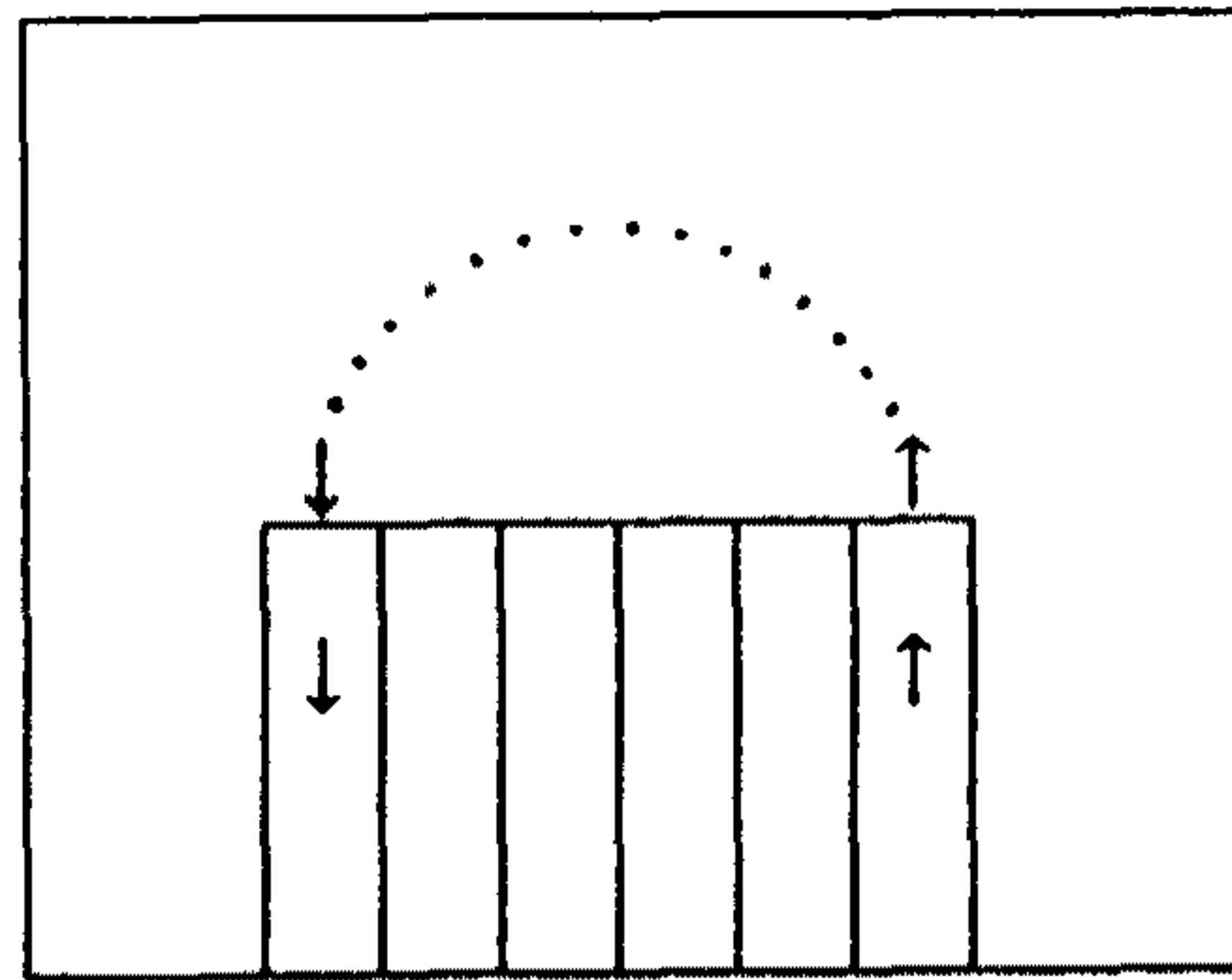
그림 2 - 7. 콤바인의 90° △ 선회방법

#### 2) 180° 선회

콤바인의 180° 선회에는 U선회와 변형U선회가 있다. 회전에취를 수행하고 난 이 후에 왕복에취를 수행할 때 180° 선회를 수행한다.

### 가) U 선회

남은 예취작업열의 폭이 콤바인의 회전 반경보다 클 때에 콤바인은 그림에서 보는 바와 같이 U선회를 수행한다. 여기에서는, 후진과 정지동작이 없기 때문에 시간이 다른 종류의 선회 시간보다 비교적 적게 소요된다.

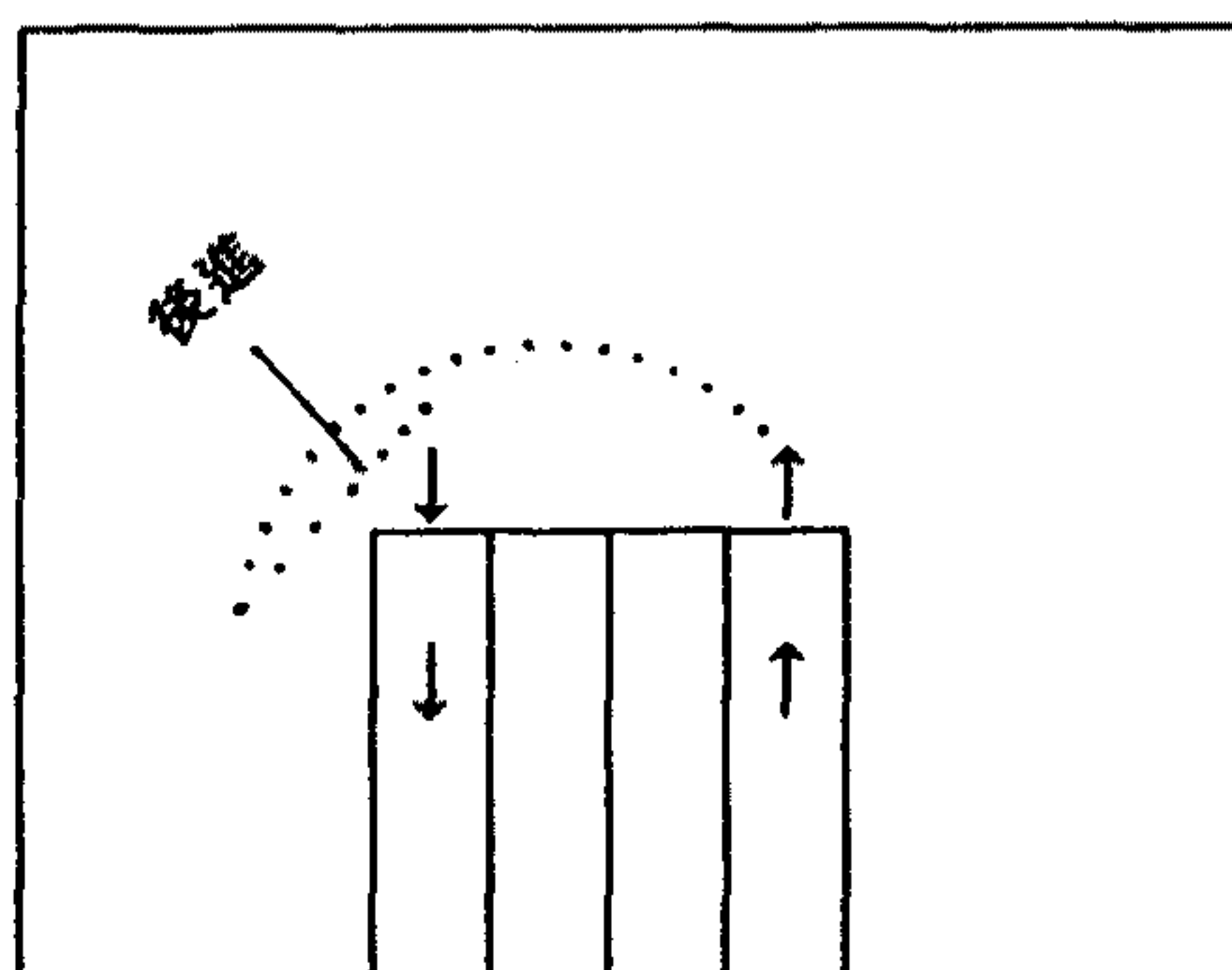


→ 는 작업진행방향

그림 2 - 8. 콤바인의 180° U 선회模型

### 나) 변형 U 선회

콤바인의 왕복예취작업에서 남은 예취작업열이 콤바인의 최소 선회반경보다 작을 경우에는 콤바인은 불가피하게 그림에서 보는 바와 같이 U선회 -> 정지 -> 후진 -> 정지 -> 전진 의 순서대로 변형된 U선회를 수행한다. 本 研究에서 이러한 선회를 변형U선회로 규정을 하였다. 여기에서는 후진과 정지동작으로 인하여 U선회보다 소요되는 시간이 많다.



→ 는 작업진행방향

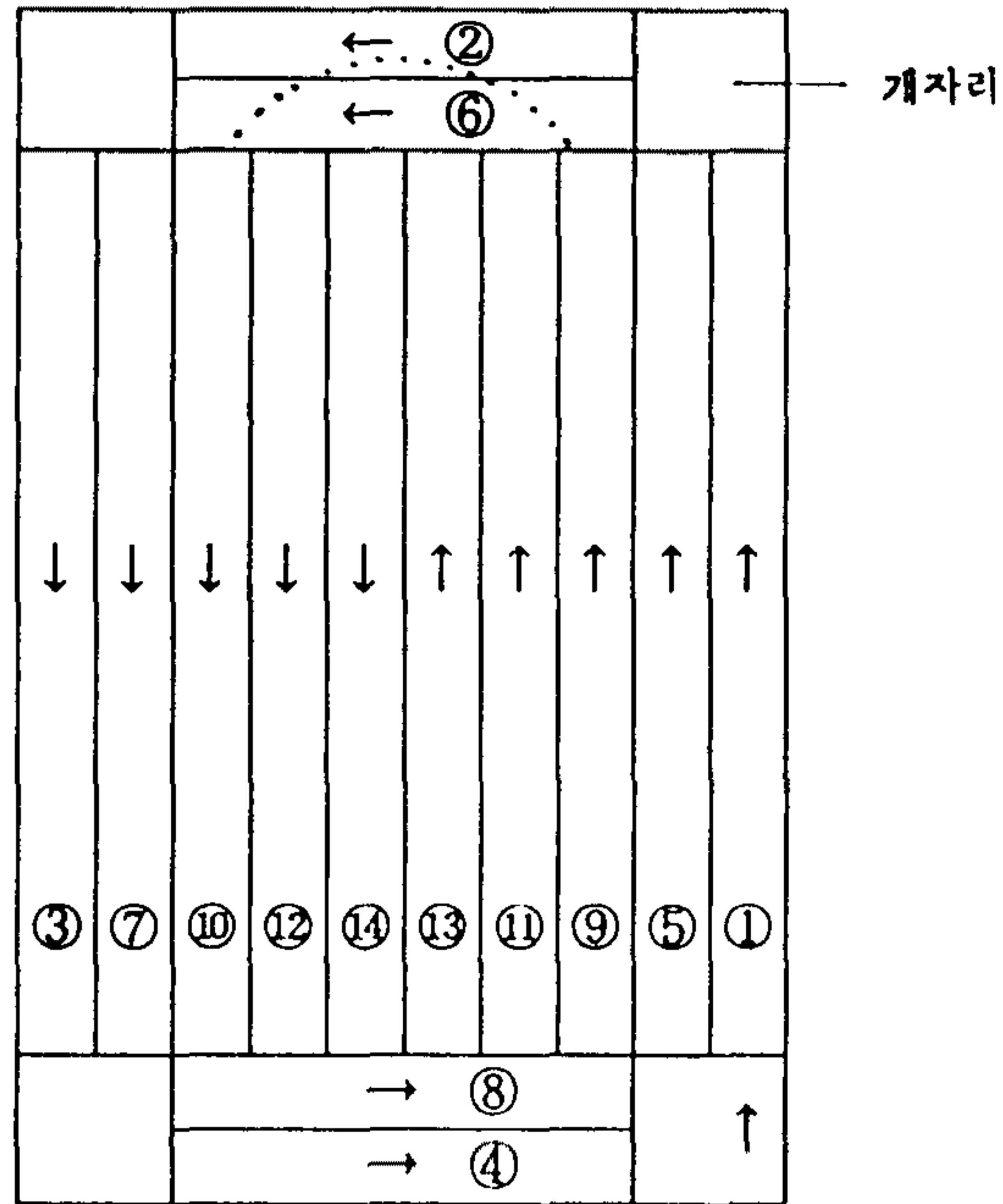
그림 2 - 9. 콤바인의 변형 U 旋回模型

#### 나. 콤바인수확작업패턴

콤바인을 사용한 수확작업방법은 트랙터를 사용하는 경운작업방법처럼 부착작업기에 따라서 다르지 않다. 콤바인은 자주식 작업기이기 때문에 부착작업기를 사용하지 않는다. 따라서, 작업방법도 콤바인의 기종에 관계없이 동일한 작업패턴으로 작업을 수행한다.

수확작업은 곡물이 포장에 딱 들어찬 상태에서 작업을 수행하므로 작업방법에 제약이 많이 따른다. 콤바인이 작업을 시작하기 전에 기계가 들어갈 자리와 선회할 자리가 필요하다. 이것을 개자리라고 한다.

수확작업을 처음 시작할 때는 직각코너에서는  $90^\circ$   $\Delta$  선회를 사용하면서 반시계방향으로 회전예취를 수행한다.  $180^\circ$  U 선회를 원할히 수행할 만큼 예취가 진행이 된 다음에는  $180^\circ$  U 선회를 이용한 왕복예취를 수행한다. 2 - 4 개의 예취작업열이 남았을 경우에는  $180^\circ$  U 선회를 수행할 수 없다. 그것은 남은 작업열의 폭이 콤바인의 U 선회반경보다 작기 때문이다. 이 때에는 콤바인의 후진이 일어나는 변형 U 선회를 이용하여 왕복예취를 작업이 완전히 끝날 때 까지 수행한다.



번호는 작업이 이루어지는 순서  
 → 는 작업진행방향

그림 2 - 10. 콤바인收穫作業方法

### 3. 作業效率分析을 위한 수학적 模型

모든 콤바인의 수확작업의 패턴은 위에서 제시된 패턴을 이용하여 수학적인 模型을 개발하였다.

가. 作業效率分析模型의 변수설정

模型에서 사용되는 각 변수들은 다음과 같다.

X (m) : 포장의 장변의 길이

$Y$  (m) : 포장의 단변의 길이

$M_w$  : 실작업폭 효율계수

$W'$  : 작업기의 이론 작업폭

#### 나. 손실시간을 포함한 총작업시간( $T_t$ )

실제로 콤바인이 포장에서 작업을 하기 위해서는 네모서리에서의 개자리와 가장자리의 작업열 등은 사람이 직접 낮으로 예취를 하나 이러한 인력작업시간은 模型에서는 고려하지 않았다.

$$T_t = \frac{XY}{W' M_w} T_0 + [2 \frac{Y}{W' M_w} - M - N] T_1 + M T_2 + N T_3 \quad (2-9)$$

$T_t$  : 포장 전작업 시간 (sec)

$T_0$  : 수확작업시 기체가 1m 전진하는 데 걸리는 시간 (sec)

$T_1$  : 콤바인의 90°  $\Delta$  선회에 걸리는 시간 (sec)

$T_2$  : 콤바인의 180° U 선회에 걸리는 시간 (sec)

$T_3$  : 콤바인의 변형 180° U 선회에 걸리는 시간 (sec)

$M$  : 180° U 선회 총회수

$N$  : 변형180° U선회 총회수

#### 다. 순작업시간( $T_f$ )

선회와 같은 손실시간을 제외한 수확작업에만 소요되는 시간은 다음 식에 의해서 계산된다.

$$T_f = \frac{XY}{W' M_o} T_0 \quad (2-10)$$

$T_f$  : 수확작업에만소요되는시간 (sec)

#### 라. 圃場作業效率( $\epsilon_f$ )

作業效率은 식(2-7)에 의해서 계산된다.

$$\epsilon_f = \left( \frac{T_f}{T_t} \right) \times 100 \quad (2-7)$$

$\epsilon_f$  : 포장작업효율 (%)

### 4. 콤바인 作業能率요인의 포장실험

#### 가. 포장실험방법

실제 콤바인수확작업을 수행하는 포장에서 作業能率요인 실험을 수행하였다. 공시기로는 대동 RX195 3조형 콤바인과 대동 Ri-43 4조형콤바인을 사용하였다. 포장의 크기는 100m(장변)×30m(단변), 100m(장변)×40m(단변)에서 실험을 수행하였다. 3조형콤바인의 경우는 4번을 실험하였고, 4조형의 경우는 3번을 수행하였다.

하나의 포장에서 수확작업의 시작에서부터 수확작업이 끝날 때까지를 기준



으로 필요한 요인들을 측정하였다.

#### 1) 예취작업속도( $T_0$ )

콤바인의 예취는 콤바인이 직진할 때에만 이루어진다. 따라서, 포장의 장변 방향으로 포장의 가운데에 10m 간격으로 한쪽 면에 두 곳, 총 4곳에 marker를 세웠다. 측정의 신뢰도를 위해서 한쪽면의 marker도 연속하지 않게 설치하였다. 그 다음에, 작업을 수행하고 있는 콤바인이 marker를 지나갈 때 걸리는 시간을 측정하였다. 단위길이를 주행하는 데 걸리는 시간의 역수는 속도로 환산이 가능하다. 이와 같은 방법으로 하나의 포장에서 작업이 시작할 때부터 작업을 종료할 때까지 측정을 여러번 수행하였다. 측정은 Digital Stop Watch를 사용하였다.

#### 2) $90^\circ$ $\Delta$ 선회시간( $T_1$ )

회전예취를 수행하면서 네개의 개자리에서  $90^\circ$   $\Delta$ 선회가 이루어진다. 따라서, 각 개자리에서 선회가 이루어질 때마다 예취를 끝내고 선회를 하기 위해 예취날을 들어올릴 때부터 선회가 완료한 후 다시 예취날을 내릴 때까지의 시간을 측정하였다. 이 측정도 하나의 포장작업을 단위로 하여 작업이 시작에서부터 종료할 때까지 모두 측정하였다. 측정은 Digital Stop Watch를 사용하였다.

#### 3) $180^\circ$ U 선회시간( $T_2$ )

회전예취가 끝나고 왕복예취를 할 때는  $180^\circ$  U선회를 한다. 따라서,  $90^\circ$   $\Delta$ 선회측정방법과 동일하게 예취를 끝내고 선회를 하기 위해 예취날을 들어올릴 때부터 선회가 완료한 후 다시 예취날을 내릴 때까지의 시간을 측정하였다. 측정은 Digital Stop Watch를 사용하였다.

#### 4) 변형 $180^\circ$ U 선회시간( $T_3$ )



2 - 4 개 정도 예취작업열이 남았을 때부터 변형 180° U선회를 하는데 이 시간의 측정도 U 선회측정방법과 동일하게 측정하였다.

5) 능률측정실험시의 작업조건

콤바인 수확작업의 능률측정실험을 할 때의 포장조건, 벼의 상태, 작업자의 숙련도 등을 표 2-1에 나타내었다.

표 2 - 1. 콤바인수확작업의 능률측정실험 작업조건

항목		수확작업조건	
벼의 도복율		0 - 20 %	
벼의 품종		아끼바레	
토양 상태		콤바인이 작업을 하면서 주행장치의 침하가 작업에 방해를 주지 않을 정도 : 적어도 작업 2 일 전부터 비가 오지 않은 상태	
작업자의 숙련도		보편적인 콤바인 조작자 : 임작업 수행	
개자리	조수 방향	3조	4조
	장변	4.25 m 이상	5.15 m 이상
	단변	3.55 m 이상	4.30 m 이상

나. 능률실험측정결과

3조형 및 4조형콤바인에 대해 각각 4회, 3회 능률측정실험을 하였다.

표 2 - 2. 3조형콤바인 작업시간의 포장실험치

	실험 I			실험 II			실험 III			실험 IV			총평균 (sec)
	평균 (sec)	추정수	표준편차	평균 (sec)	추정수	표준편차	평균 (sec)	추정수	표준편차	평균 (sec)	추정수	표준편차	
T <sub>0</sub>	17.95	21	1.65	20.30	20	1.98	12.83	54	0.89	15.37	58	2.08	15.4
T <sub>1</sub>	19.06	18	3.94	18.81	16	5.31	13.86	38	2.25	14.34	48	2.76	15.5
T <sub>2</sub>	23.25	4	2.95	23.0	4	2.74	15.60	5	3.93	17.5	4	1.12	19.6
T <sub>3</sub>	29.67	3	8.22	23.25	4	6.53	16.5	2	0.5	16.0	2	1.0	22.5

표2 - 3. 4조형콤바인 작업시간의 포장실험치

	실험 I			실험 II			실험 III			총평균 (sec)
	평균 (sec)	추정수	표준편차	평균 (sec)	추정수	표준편차	평균 (sec)	추정수	표준편차	
T <sub>0</sub>	9.19	15	0.85	9.62	18	1.05	9.41	46	2.31	9.4
T <sub>1</sub>	9.33	25	1.32	9.9	25	1.61	11.39	33	5.07	10.3
T <sub>2</sub>	13.7	1	0	14	1	0	14	3	0.82	14.0
T <sub>3</sub>	16.20	2	1.4	13.37	3	0.17	15.33	3	0.94	14.8

3조형과 4조형콤바인의 능률측정한 결과를 총괄 요약하여 다음 표 2-4에 나타내었다.

표 2 - 4. 콤바인 수확작업에서의 기본작업시간의 총괄표

측정항목	평균 소요 시간 (sec)				비 고
	3 조	표준	4 조	표준	
T <sub>0</sub>	12.83 ~ 20.3	15.4	9.19 ~ 9.62	9.4	예취를 수행하면서 10 m 전진하는 데 걸리는 시간, sec
T <sub>1</sub>	13.86 ~ 19.06	15.5	9.33 ~ 11.39	10.3	예취를 하지 않고 90° Δ선회하는 데 걸리는 시간, sec
T <sub>2</sub>	15.60 ~ 23.25	19.6	13.7 ~ 14.0	13.9	예취를 하지 않고 180° U 선회하는 데 걸리는 시간, sec
T <sub>3</sub>	16.0 ~ 29.67	22.5	13.37 ~ 16.20	14.8	예취를 하지 않고 변형 180° U 선회하는 데 걸리는 시간, sec
M	4 ~ 5	4	1 ~ 3	2	180° U 선회하는 총 회수
N	2 ~ 4	3	2 ~ 3	3	변형 180° U 선회하는 총 회수
W'	1.2	1.2	1.5	1.5	이론작업폭, m
M <sub>w</sub>	0.92 ~ 0.95	0.94	0.93 ~ 0.96	0.94	실작업폭계수

표 2 - 4 에서 보는 바와 같이 각 측정 값들이 상당히 큰 범위를 가지고 있다. 이것은 포장의 조건과 작업자의 숙련도 그리고 벼의 품종과 도복상태등의 영향으로 인한 것이다.

## 5. 作業效率과 能率의 分析

### 가. 기종별의 作業效率分析

단변이 30m, 장변을 장단변비 2 - 5 의 범위인 30m에서 150m로 변화시켜

3조와 4조 콤바인의 작업효율을 비교分析하였다.

그림 2-11에서 보는 바와 같이 3조와 4조의 효율의 차이는 명확하다. 장변이 30m인 경우에는 作業效率은 3조와 4조가 각각 61%, 54%이고 100m인 경우에는 3조와 4조는 각각 84%, 79%이다. 장변이 150m인 경우에는 3조와 4조형 콤바인의 경우는 88%와 85%이다.

3조와 4조형 콤바인의 作業效率은 경지의 크기가 커질수록 다같이 커지며 3조가 4조에 비해 作業效率이 크게 증가한다는 것을 알 수 있다. 作業效率의 차이는 장변이 30m일 때 7%, 100m일 때 5%, 150m일 때 3%로 줄어드는 경향을 나타내고 있다.

이것은, 대형이나 고속기계일수록 포장내에서의 손실시간이 상대적으로 큰 비중을 차지하기 때문이다. 대형 또는 고속기계일수록 포장의 규모가 커야 된다는 것을 단적으로 나타내는 좋은 예라고 할 것이다.

단변이 40m, 장변을 장단변비 1 - 5 의 범위인 40m에서 200m로 변화시켜 3조와 4조 콤바인의 作業效率을 비교分析하였다.

그림 2-12에 3조와 4조의 作業效率이 단변 40m인 경우에 장변의 변화에 따라 作業效率이 어떻게 변화하는 것을 나타낸 것이다. 또한, 그림 2-13에는 단변이 50m인 경우의 作業效率을 나타낸 것이다.

경지의 크기가 커질수록 3조와 4조형 콤바인의 作業效率의 차이는 장변이 40m일 때 7%, 100m일 때 4%, 200m일 때 3%로 줄어드는 경향을 나타낸다.

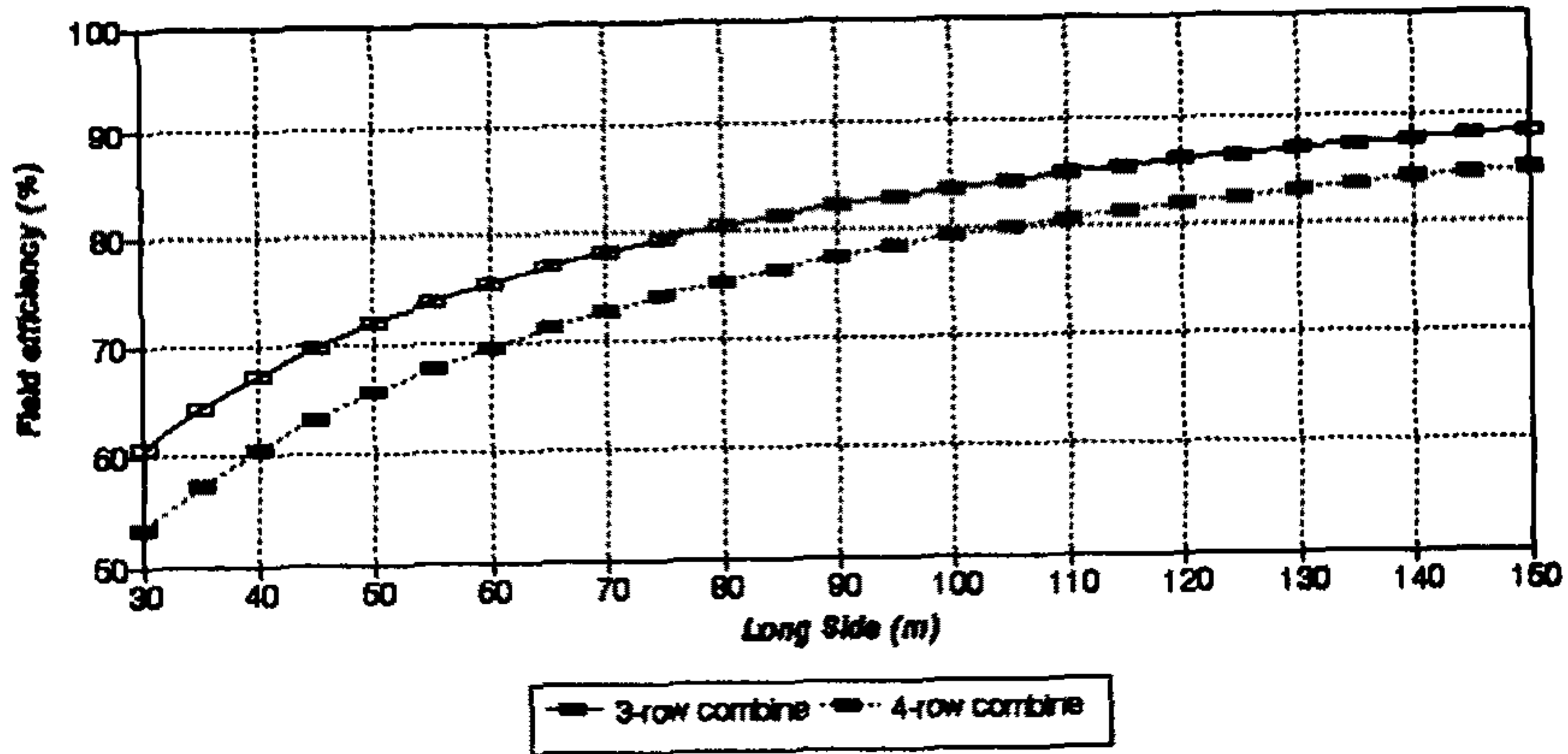


그림 2 - 11. 단변이 30m이고 장변을 변화시킬 때 기종별의 作業效率비교

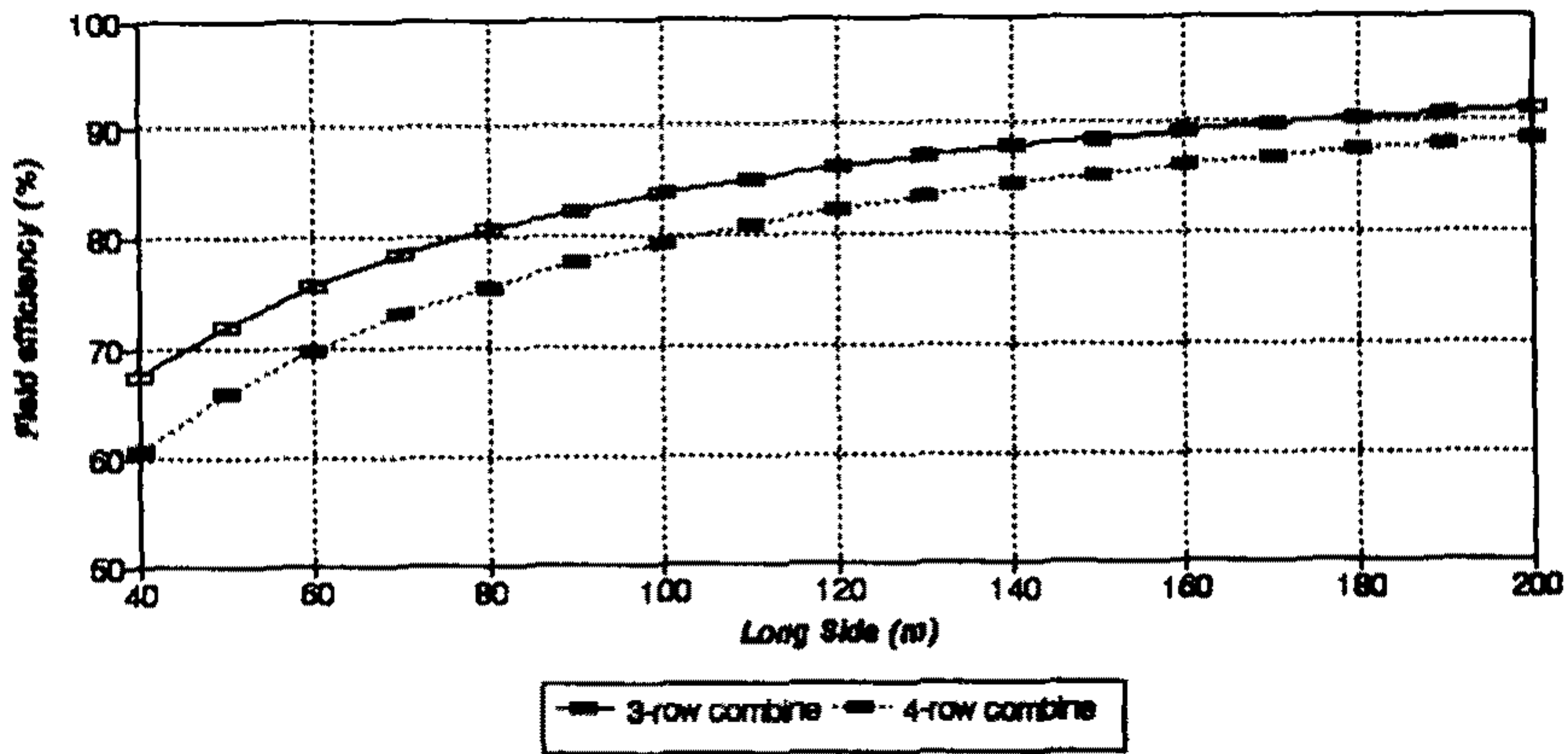


그림 2 - 12. 단변이 40m이고 장변을 변화시킬 때 기종별의 作業效率비교

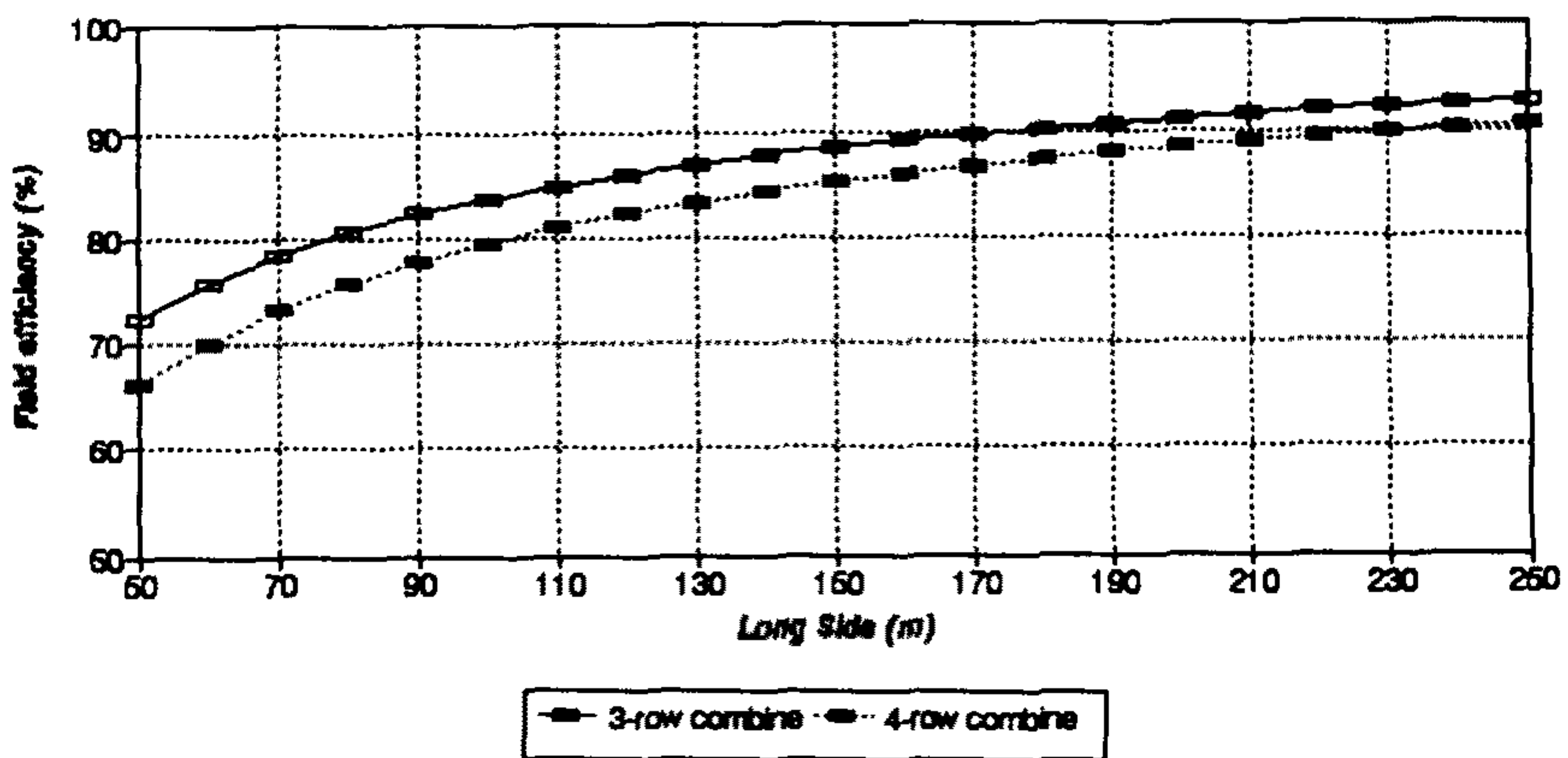


그림 2 - 13. 단변이 50m이고 장변을 변화시킬 때 기종별의 作業效率비교



#### 나. 단변을 변화시켰을 때의 각機種別 作業效率分析

3조형 콤바인을 장변이 각각 100m, 150m, 200m 일 때 단변을 30m에서 100m로 변화를 시켰을 때의 효율을 分析하였다. 그림 2-24에서 보는 바와 같이 단변을 변화시켰을 때의 각 장변에서의 효율변화는 거의 없이 일정했다.

장변이 100m일 경우에는 효율은 약 84%, 150m일 경우에는 88.5%, 200m일 경우에는 91%를 약간 초과하였다.

그림 2-15에는 4조형 콤바인에 대하여 장변이 각각 100m, 150m, 200m 일 때 단변을 30m에서 100m로 변화를 시켰을 때의 圃場作業效率를 分析한 것을 나타낸 것이다. 그림에서 보는 바와 같이 단변을 변화시켰을 때의 각 장변에서의 효율변화는 거의 없이 일정했다.

장변이 100m일 경우에는 효율은 단변에 관계없이 79%, 150m일 경우에는 약 85 %, 200m일 경우에는 88.6%로 나타났다.

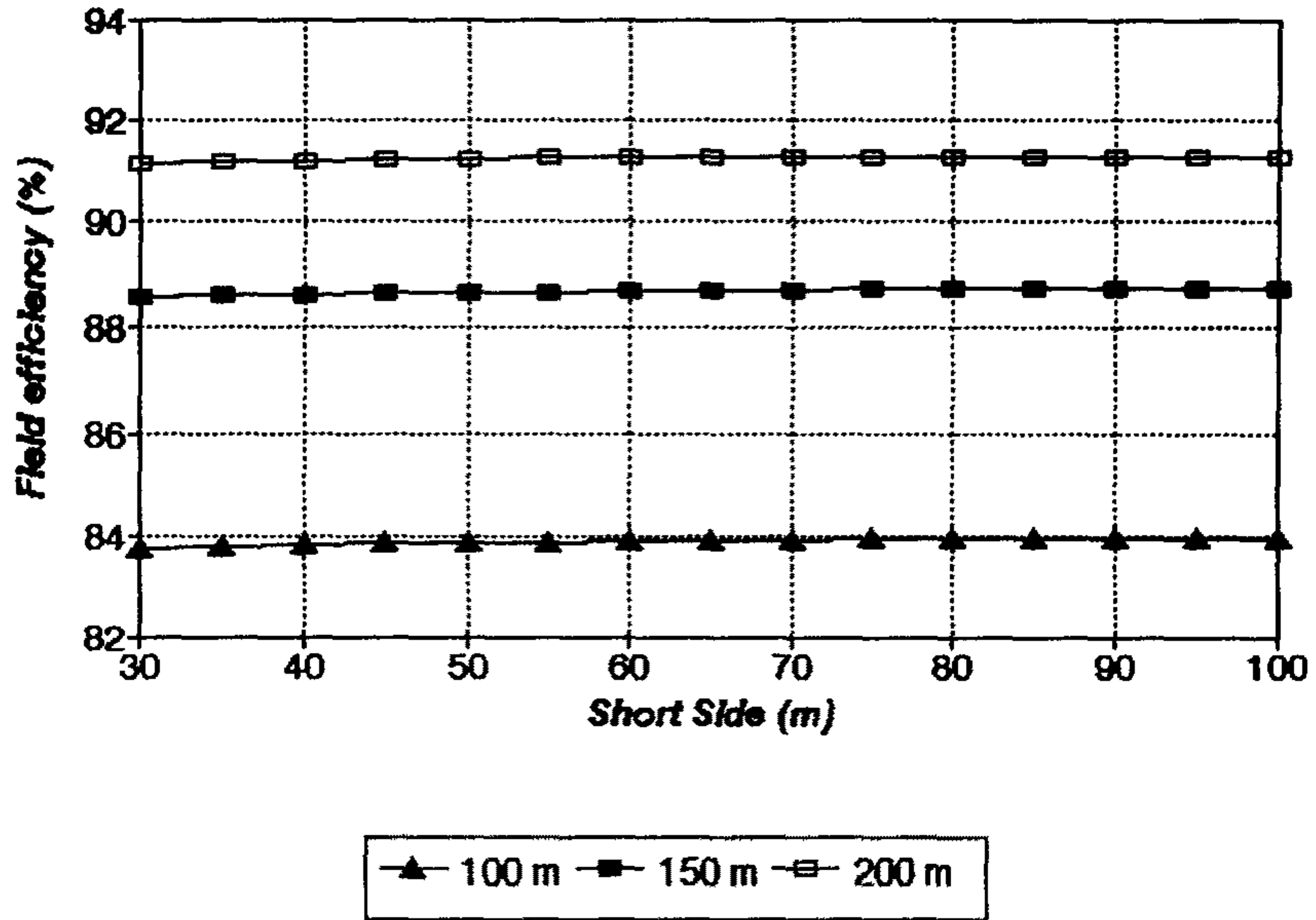


그림 2 - 14. 단변의 변화에 따른 圓場作業效率(3조)

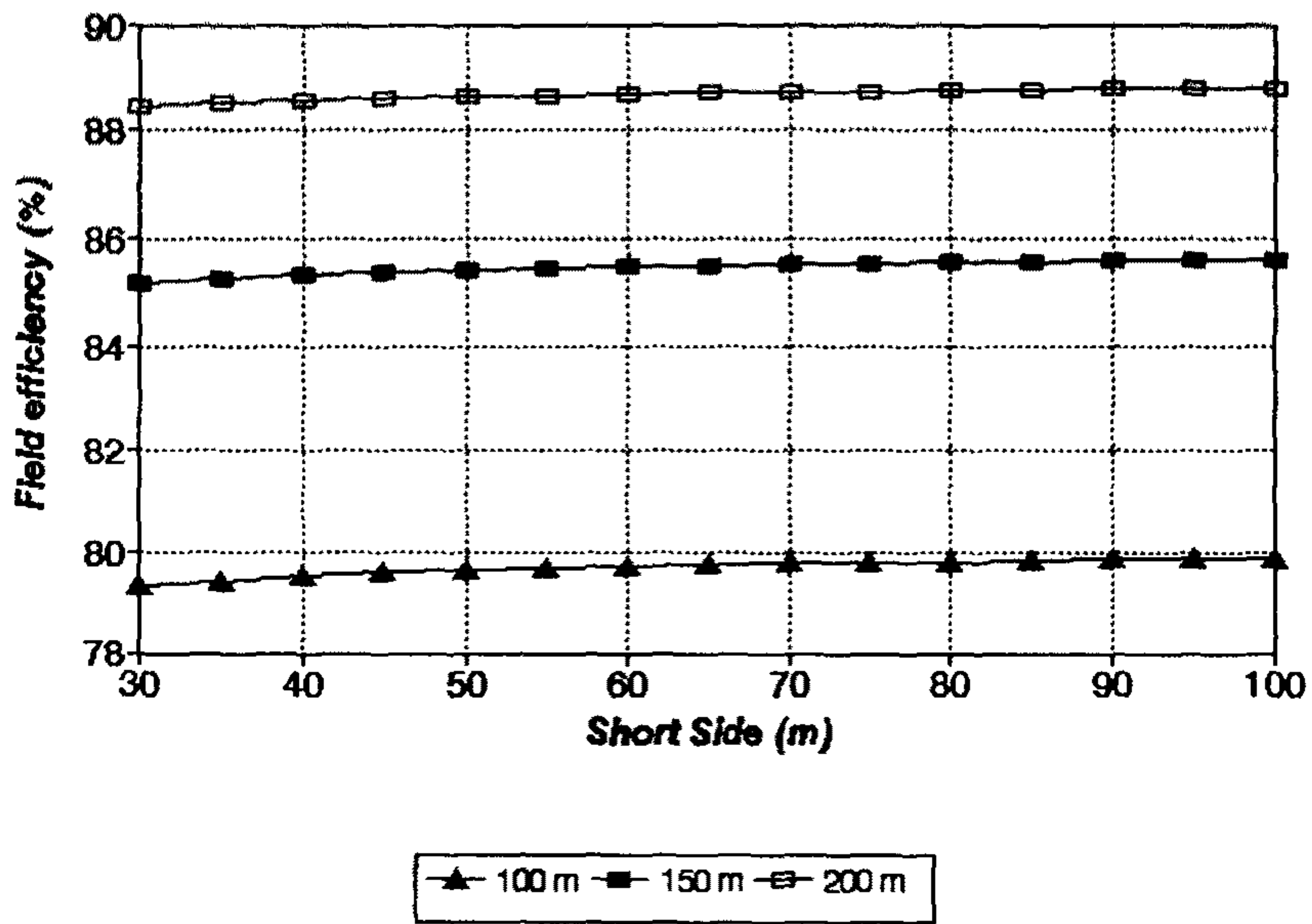


그림 2 - 15. 단변의 변화에 따른 圓場作業效率(4조)



#### 다. 동일한 경지규모에서의 장단변비를 변화시킬 때의 作業效率分析

그림 2-16에는 동일한 경지규모에서 장단변비를 증가시켰을 때 作業效率의 변화형태를 나타낸 것이다. 여기에서, 경지규모는 1 - 5ha 장단변비( $M = X/Y$ )는 1 - 5 까지 변화시켜 그에 따른 作業效率을 分析한 것이다.

장단변비 1 - 5 까지 증가시켰을 경우에 1ha 경지에서는 효율이 84 - 92 %, 2ha 에서는 88 - 94%, 3ha 에서는 90 - 96%, 4ha에서는 91 - 95%, 5ha 에서는 92 - 96 %의 범위에 있으며 장단변비가 증가할수록 효율은 증가하나 그 변화범위는 줄어들을 알 수 있다. 이것은 作業效率을 향상시키기 위하여 장단변비를 증가시킬 때 그 적절한 범위가 존재함을 의미한다. 또한, 같은 장단변비에 있어서는 경지의 규모가 커질수록 효율은 증가한다. 그리고 경지규모가 커질수록 효율의 증가범위가 줄어든다.

장단변비의 변화에 따른 作業效率의 변화는 4조도 3조의 경우와 거의 비슷하다. 그림 2-17에는 경지규모를 1 - 5ha로 하여 장단변비( $M = X/Y$ )를 1 - 5 까지 변화를 주어서 作業效率을 分析하여 나타낸 것이다.

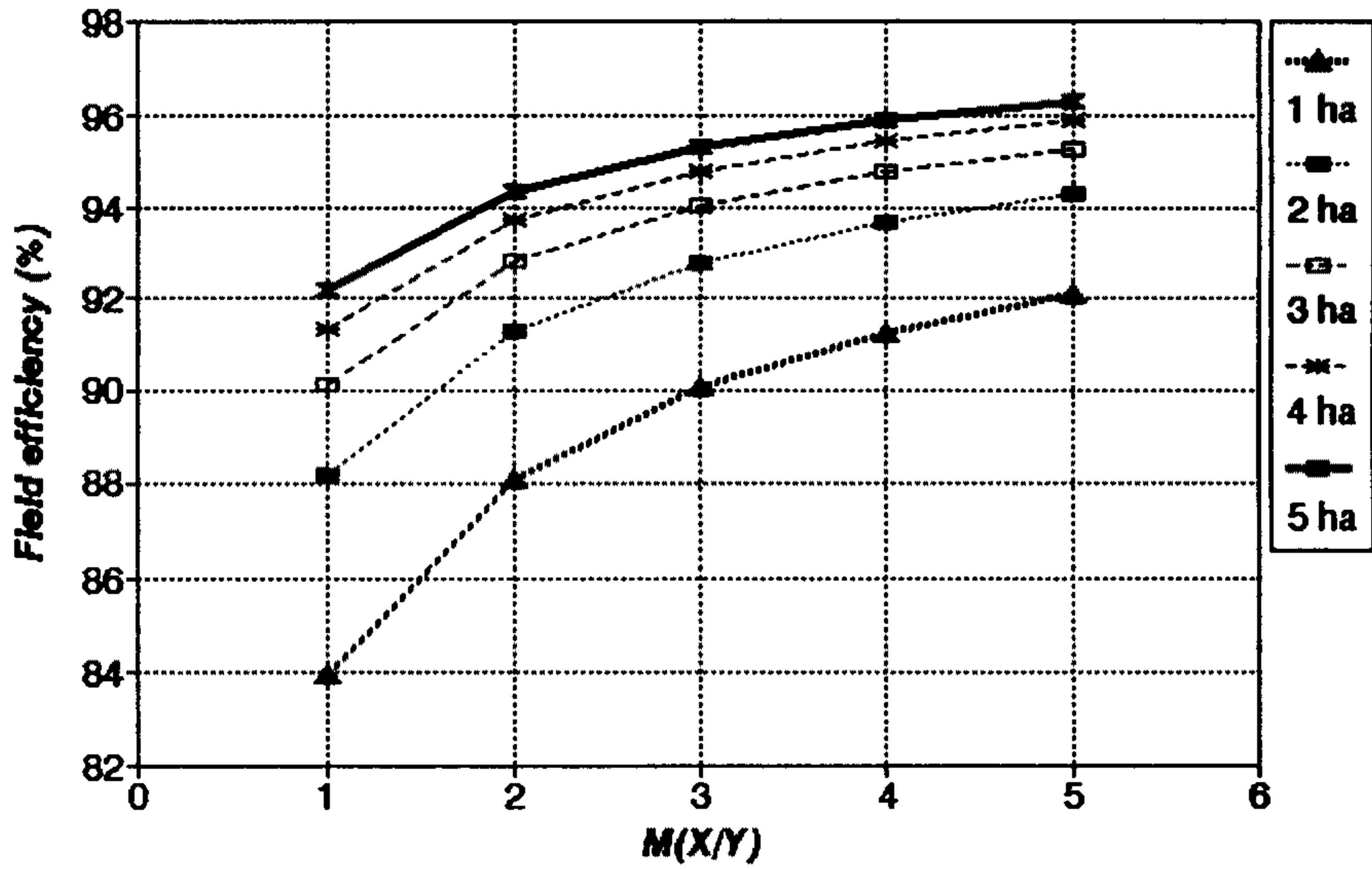


그림 2 - 16. 동일한 경지규모에서 장단비(M)를 변화시켰을 때의 作業效率(3조)

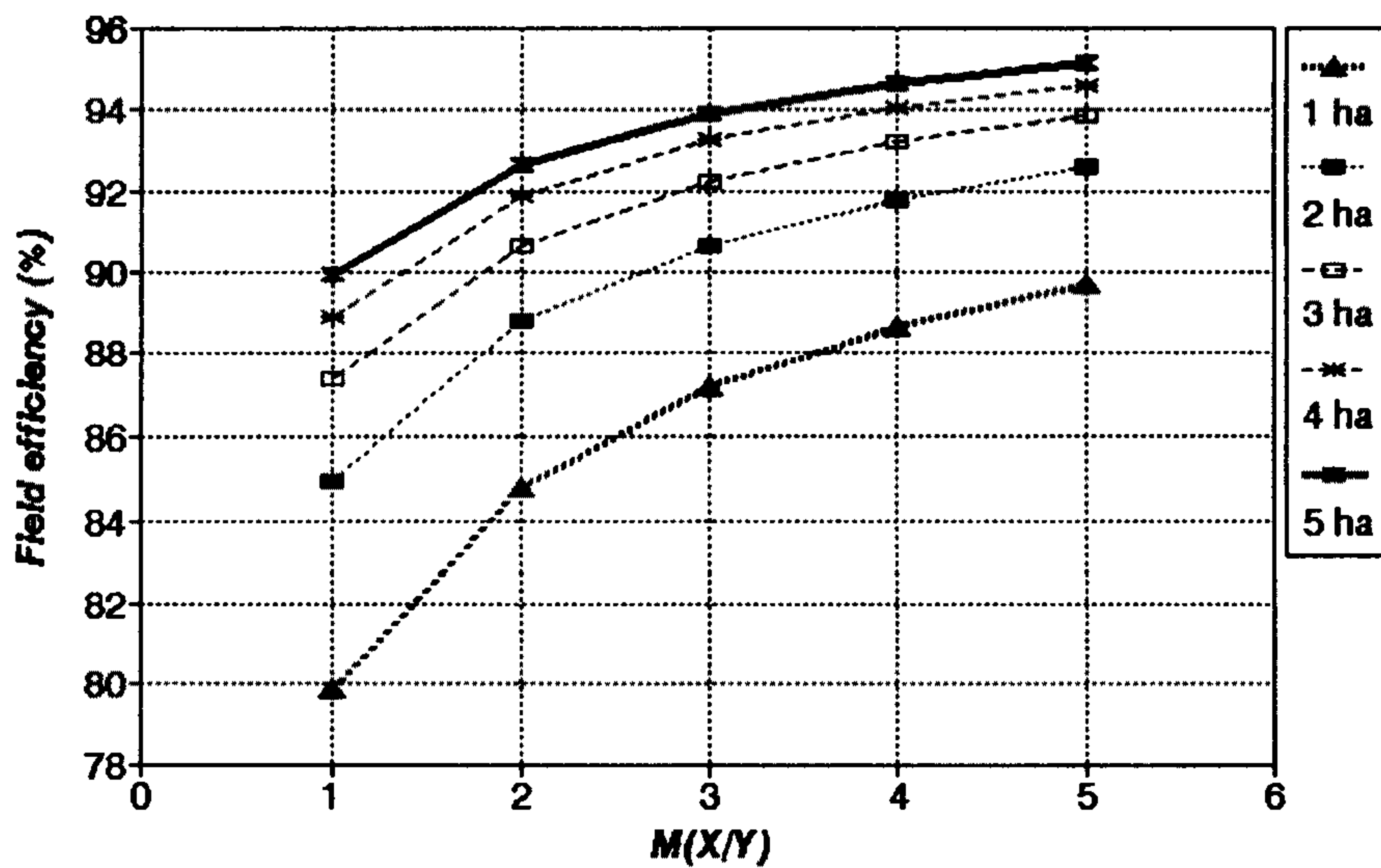


그림 2 - 17. 동일한 경지 규모에서 장단비(M)를 변화시켰을 때의 作業效率(4조)

라. 콤바인의 作業能率分析

3조형 콤바인에 의한 수확작업의 作業能率을 각기 식 (2-9), 식(2-7), 식 (2-10)과 식(2-2)를 이용하여 기술된 작업방법과 포장에서 측정된 실험치를 적용하여 分析 하였다. 그림 2-18은 단변이 30m 의 경우와 40m의 경우에 장변을 50m 에서 150m 의 범위로 변화시키면서 分析한 것을 나타낸 것이다.

그림 2-18에서 보는 바와 같이 단변이 30m와 40m일 때 장변이 50m 일 때 作業能率은 각각 0.239 ha/hr, 0.267 ha/hr에서 장변이 150m 일 때는 각각 0.292ha/hr, 0.298 ha/hr로 증가했다. 경지의 규모가 커질수록 作業能率은 단변이 30m 일때는 0.053ha/hr 증가하였고 40m 일 때는 0.031 ha/hr 증가하였다.

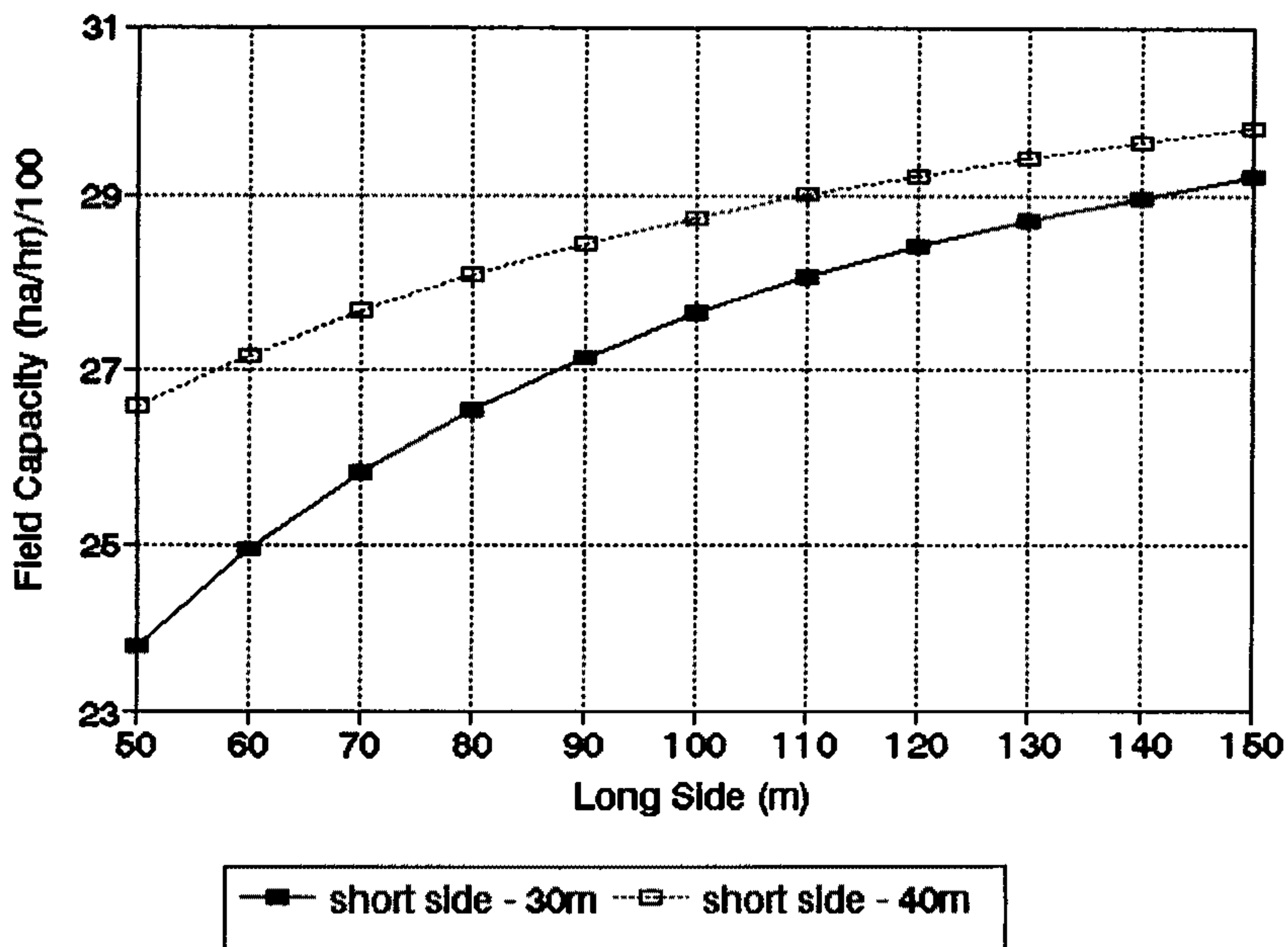


그림 2 - 18. 단변 30m, 40m일 때 장변을 변화시켰을 때의 作業能率(3조형)

3조콤바인에서와 같은 경지구모에서의 4조 콤바인의 作業能率을 分析하였다. 分析에 사용된 식과 방법은 3조형의 경우와 같다. 그림 2-19는 단변이 30m 의 경우와 40m의 경우에 장변을 50m 에서 150m 의 범위로 변화시키면서 分析한 것을 나타낸 것이다.

그림 2-19에서 보는 바와 같이 단변이 30m와 40m일 때 장변이 50m 일 때 作業能率은 각각 0.421 ha/hr, 0.482 ha/hr에서 장변이 150m 일 때는 각각 0.544 ha/hr, 0.560 ha/hr로 증가했다. 경지의 장변이 50m 에서 150m로 커질수록 作業能率은 단변이 30m 일때는 0.123 ha/hr 증가하였고 40m 일 때는 0.078 ha/hr 증가하였다.

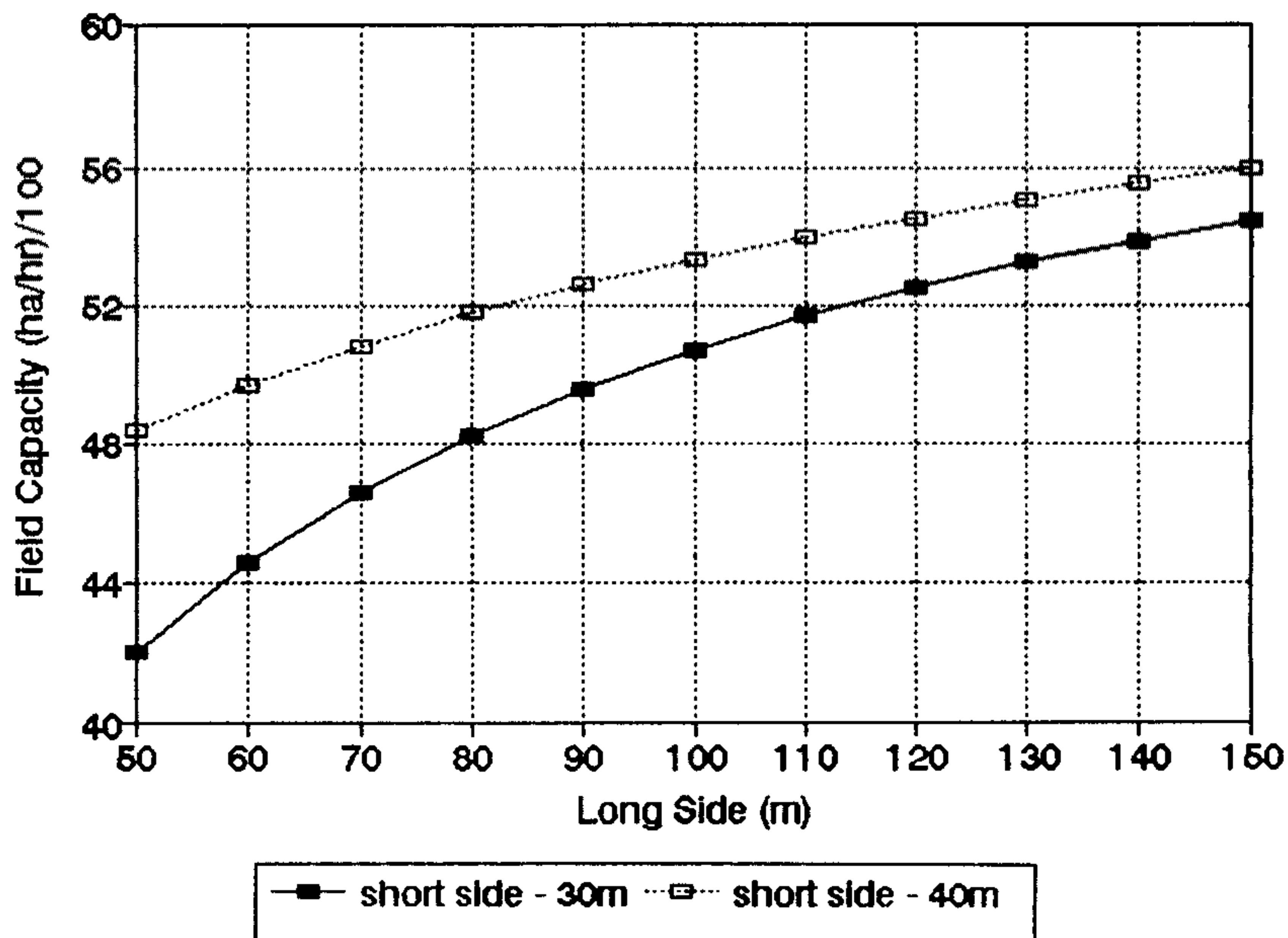


그림 2 - 19. 단변 30m, 40m일 때 장변을 변화시켰을 때의 作業能率(4조형)

여 백

## 第 3 章 地域別 作業可能日數率 決定

農業機械化의 側面에서 고려되어야 할 변수중의 하나가 地域別 및 作業別 作業可能日數率의 決定이다. 作業可能日數率은 營農規模와 기계의 負擔面積결정의 基礎的인 자료가 된다.

農作業은 각 作業별로 作業適期가 존재하며 또한, 作業適期는 地域別로 상당한 차이를 나타낸다. 따라서, 地域別 및 作業別의 作業可能日數率은 다양한 형태로 나타난다. 本 研究에서는 우선, 主要 地域別로 콤바인收穫作業에 대한 作業可能日數率을 分析하였다.

### 3.1 收穫作業의 作業可能日數率

콤바인收穫作業可能日數率의 결정에 가장 큰 영향을 미치는 요인은 강수량이다. 또한, 收穫作業은 강수에 의하여 콤바인의 습지주행성뿐 아니라 탈곡선별 작업에 큰 영향을 주기 때문에 다른 농작업에 비하여 기상조건에 아주 민감하게 영향을 받는 작업이다. 특히, 우리나라는 남북으로 길게 뻗어 있어 각 地域別로 기상조건이 큰 차이를 나타낸다. 따라서, 地域別로 수확적기가 조금씩 다르게 나타난다. 또한, 기계화수확작업의 진전에 따라 강수로 인하여 그 작업이 불가능한 일수를 규정할 필요가 있으며, 지역별로 어떻게 변하는가 알 필요가 있다.

작업가능일수율( $\varepsilon_d$ , %)은 다음과 같이 정의한다.

$$\varepsilon_d = \left( \frac{D_a}{D} \right) \times 100 = (1 - \varepsilon_i) \times 100 \quad (3-1)$$

여기서,  $D$  : 작업기간중의일수

$D_a$  :  $D$ 에서 강수로 인하여 작업이 불가능한

일수를 뺀 일수

$\varepsilon_i$  : 작업불가능일수율

$D$ 는 분석의 정확도와 이용의 평의성을 고려하여 보통 10일을 잡는다. 현재 대부분의 收穫作業은 콤바인으로 작업을 수행한다. 따라서, 본 研究에서 收穫作業可能日數率을 콤바인의 경우를 기준으로 하여 分析하였다. 본 研究에서는 作業可能日數率을 결정하는 모형을 우선 개발하고 그 개발된 모형을 기준으로 하여 전국의 주요 地域別로 收穫作業可能日數率을 分析하였다.

### 3.1.1 收穫作業可能日數決定을 위한 模型開發

#### 1. 收穫作業可能日數의 決定基準

收穫作業可能日數率의 決定에 가장 큰 影響을 미치는 要因으로서 降水量을 들수 있다. 따라서, 氣象資料중 일별강수량을 作業可能日數의 결정의 가장 기본적인 資料로 채택하였다.

외국에서의 收穫作業可能日數率決定基準을 分析하였고 나아가서 우리나라의 특수성을 고려하였으며 실제의 分析의 正確性을 기하기 위해서 실제작업을 수행



하는 농민들의 의견도 調査하여 참조를 하였다.

#### 가. 외국의 收穫作業可能日數決定의 基準

일본의 경우 실외 노동 가능 시간을 추정하고 이외에 일강수량 1~10mm 까지의 日數 혹은 10mm 이상의 일수 및 積雪 일수를 총합해서 월별 作業 가능 일수를 算定하고 이것을 각 월의 일수로 나누어서 각 월의 作業가능일수율로 표시하고 있다.

미국에서, Donaldson등은 특히, 맥류의 Combine收穫作業이 가능한 限界日 강수량을 0.01inch 로 보고 있으며 Jeffers등은 forage收穫이 가능한 한계 일강수량을 當日은 0.05inch 이하, 전일은 0.5inch 이하로 규정하고 있다.

따라서, 미국의 경우는 하루의 강우가 어느 때부터 시작되었든 일강우량이 '0.01inch' 이상이면 그 날은 收穫作業이 불가능하다고 가정하였다.

미국의 경우는 주로 맥류와 같은 전작의 경우가 대부분이었고 일본의 경우는 收穫作業에 관하여 특수하게 정의하지 않고 농작업 전반의 作業可能日數를 분석한 것이다. 따라서, 외국의 경우를 그대로 우리나라에 적용할 수는 없다.

#### 나. 收穫作業可能日數 決定模型

##### 1) 作業可能日數의 決定基準

本 研究에서는 收穫作業이 가능한 한계강수량을 1mm로 삼았다. 이 기준은 일본과 미국의 결정기준과 비슷하며 실제 收穫作業에 조사한 결과도 이와 비슷하게 나타났다.

표 3-1에는 연속강우일경우에 콤바인수확작업이 불가능한 일수를 결정하는 기준을 마련한 것이다.

표 3-1. 收穫作業可能日數의 決定基準

연속강우일수	강 수 량	작업불가능일수 결정	작업불가능 총일수(일)
1 일	1 mm 이하	당일가능	0
	1 - 20 mm	당일만 작업 불가능	1
	20 - 60 mm	당일과 그 다음날 작업 불가능	2
	60 mm 이상	당일과 그 다음 2일간 작업 불가능	3
2 일	<sup>1)</sup> 2 mm - 30 mm	당일만 작업불가능	1
	<sup>2)</sup> 30 mm - 60 mm	당일과 그 다음날 작업 불가능	2
	<sup>3)</sup> 60 mm 이상	당일과 그 다음 2일간 작업 불가능	3
3 일	<sup>1)</sup> 3 mm - 30 mm	당일만 작업 불가능	1
	<sup>4)</sup> 30 mm - 75 mm	당일과 그 다음날 작업 불가능	2
	<sup>5)</sup> 75 mm 이상	당일과 그 다음 2일간 작업 불가능	3

주) 1)은 일최소강우량 1mm를 의미함 2)는 일최소강우량 15mm를 의미함  
 3)은 일최소강우량 30mm를 의미함 4)는 일최소강우량 10mm를 의미함  
 5)는 일최소강우량 25mm를 의미함

作業可能日數 결정모델에서는 당일작업이 가능한 최소한계일강우량을 1mm

로 정했으며, 또한 최대작업불가능일수를 강우당일을 포함한 그 이후의 연속 2일간으로 하였다. 또한, 연속 강수일 경우에도 적용하였으며 마지막 날의 강우량이 그 전날의 강우량보다 많을 경우는 연속 강우에 먼저 적용하고 그 다음에는 당일 강우에 다시 적용시켜서 실제 작업가능일과의 오차를 줄였다.

예를 들면 10월 11일에 2 mm가 오고 12일에 2mm 그리고 13일에 비가 25 mm가 왔을 경우에는 연속 강우에만 적용할 경우 작업을 할 수 없는 날은 강수 당일인 11일, 12일, 13일 밖에 되지 않는다. 하지만 실제로는 14일에도 작업을 할 수가 없다. 따라서, 마지막 날인 14일은 독립적으로 당일 강우에 적용을 시켜 그 다음 날도 작업을 할 수 없는 날로 간주하였다.

## 2) 收穫作業可能日數率의 分析

우리나라의 모든 지역의 收穫作業可能日數率을 분석한다는 것은 매우 힘들 뿐만 아니라 무의미한 일이기도 하다. 그것은, 우리나라의 모든 지역이 수도작 지역이 아니기 때문이다. 하지만, 지역선정에 있어 수도작중심지를 중심으로 지역적인 분포를 최대한 고려하였다.

또한, 분석된 氣象資料를 통계처리하여 분석자료의 정확성을 기했다.

### 가). 대상지역의 선정

우리나라는 남북으로 길게 뻗어 있으므로 같은 시기라 하더라도 위도와 지형에 따라서 그 지역의 기후가 상당히 다르다. 작물을 재배하는 데에 있어서도 기후에 따른 재배시기와 수확시기가 명확하게 다르다. 특히, 벼의 경우는 남부지

방과 중부지방의 이앙시기 뿐만 아니라 수확시기도 큰 차이를 나타낸다. 따라서, 지역 마다 수확적기가 다르다. 그렇기 때문에 대상 지역 선정에 있어서 각 도마다 세 곳을 지역적인 분포와 아울러 벼의 곡창지인가를 고려하여 선정하였다. 그리고, 여기서는 강원도와 제주도는 지역적인 특수성과 실제 주 재배 작물이 벼가 아니기 때문에 제외를 시켰다. 따라서, 경기도 지방에서는 평택, 김포, 이천과 충청도 지방에는 서산, 공주, 논산을, 경상도 지방에서는 김해, 영일, 칠곡 및 전라도 지방에서는 나주, 김제, 정읍을 선정하였다.

#### 나.) 氣象資料의 選定

地域別로 약간의 분석氣象資料의 년도와 기간의 차이는 있지만 평균적으로 1965년에서 1991년까지의 氣象資料를 이용하였고, 地域別로 평균 20개년간의 氣象資料를 이용하였다. 그리고, 콤바인 收穫作業이 주로 이루어지는 9월 하순에서 11월 중순까지를 선정하여 그 시기의 일별 강우량을 기록한 자료를 분석대상이 되는 氣象資料로 선정하였다.

#### 다. 作業可能日數率의 分析方法

##### 1) 地域別 및 年度別 作業可能日數分析

地域마다 年度별로 분포되어 있는 氣象資料를 수확적기에 해당하는 월별 및 旬別로 세분하여 氣象資料를 재구성하였다. 각 해당년도마다 旬別로 세분되어진 자료에서 旬別作業可能日數를 결정했다. 그리고, 그 年度별·旬別로 분석된 作業可能日數 데이터를 각 순에 해당하는 作業可能日數를 年度별로 표 3-2에서

보는 바와 같이 나타냈다. 표 3-2는 논산지역의 10월 상순의 년도별 작업 가능일수 데이터를 예를 들어 나타낸 것이다.

표 3 - 2. 논산 地域의 年度別 作業可能日數 (10월상순)

년도	65	66	67	68	69	70	71	72	73	74
작업가능일수	10	9	10	5	10	10	9	7	6	7
년도	75	76	77	78	79	80	81	82	83	84
작업가능일수	7	8	10	10	7	6	9	9	8	8

## 2) 作業可能日數率의 確率分布 分析

표 3 - 2를 보면 최소 作業可能日數가 5일간임을 알 수 있다. 따라서, 10월 상순에서 作業可能日數가 1일에서 5일까지일 확률은 100%인 1이 된다. 그 다음에 6일이 될 확률은 작업일수가 5일인 년도가 1개이므로 전체 총 합인 20에서 1을 뺀 19년이 6일간 작업을 할 수 있는 년도가 된다. 따라서, 확률은 19/20, 즉 0.95가 된다. 이러한 방법으로 일수별로 작업가능확률을 구하면 표 3-3 에서 보는 바와 같다.

표 3 - 3. 논산 地域의 作業可能日數別 確率 (10월 상순)

가능일수	1	2	3	4	5	6	7	8	9	10
확률	1	1	1	1	1	0.95	0.85	0.65	0.50	0.30

3) 日數別 確率分布의 統計處理方法

표 3 - 3 에서 分析된 일수별 確率 데이터를 統計學的인 方法을 사용하여 데이터를 분석하였다. 本 研究에서는, 통계 Package인 SAS를 이용하여 Polynomial Regression을 사용하여 데이터를 통계처리하였다. 다항식의 차수를 증가했을 때  $R^2$ (least square)의 값이 1에 가까운 다항식의 분석의 모형으로 삼았다. 논산지역의 작업가능일수율의 누계분포다항식의 차수에 따른  $R^2$ 의 값은 표 3-4에 나타내었다. 여기에서,  $R^2$ 의 값이 1에 가장 근접한 차수는 4차이며, 다른 지역에 대해서도 비슷한 결과를 나타내었다.

표 3 - 4. 各 次數別의 Regression  $R^2$  값의 비교

다항식의 次數	$R^2$ 값
1	0.86218
2	0.94685
3	0.98345
4	0.98542
5	0.96682



## 2. 地域別 收穫作業可能日數確率分布曲線

그림 (3 - 1)에서 그림 (3 - 24)까지는 각 지역의 旬別 作業可能日數率의 確率 分布를 4차의 Regression으로 分析하여 確率分布曲線으로 나타낸 것이다.

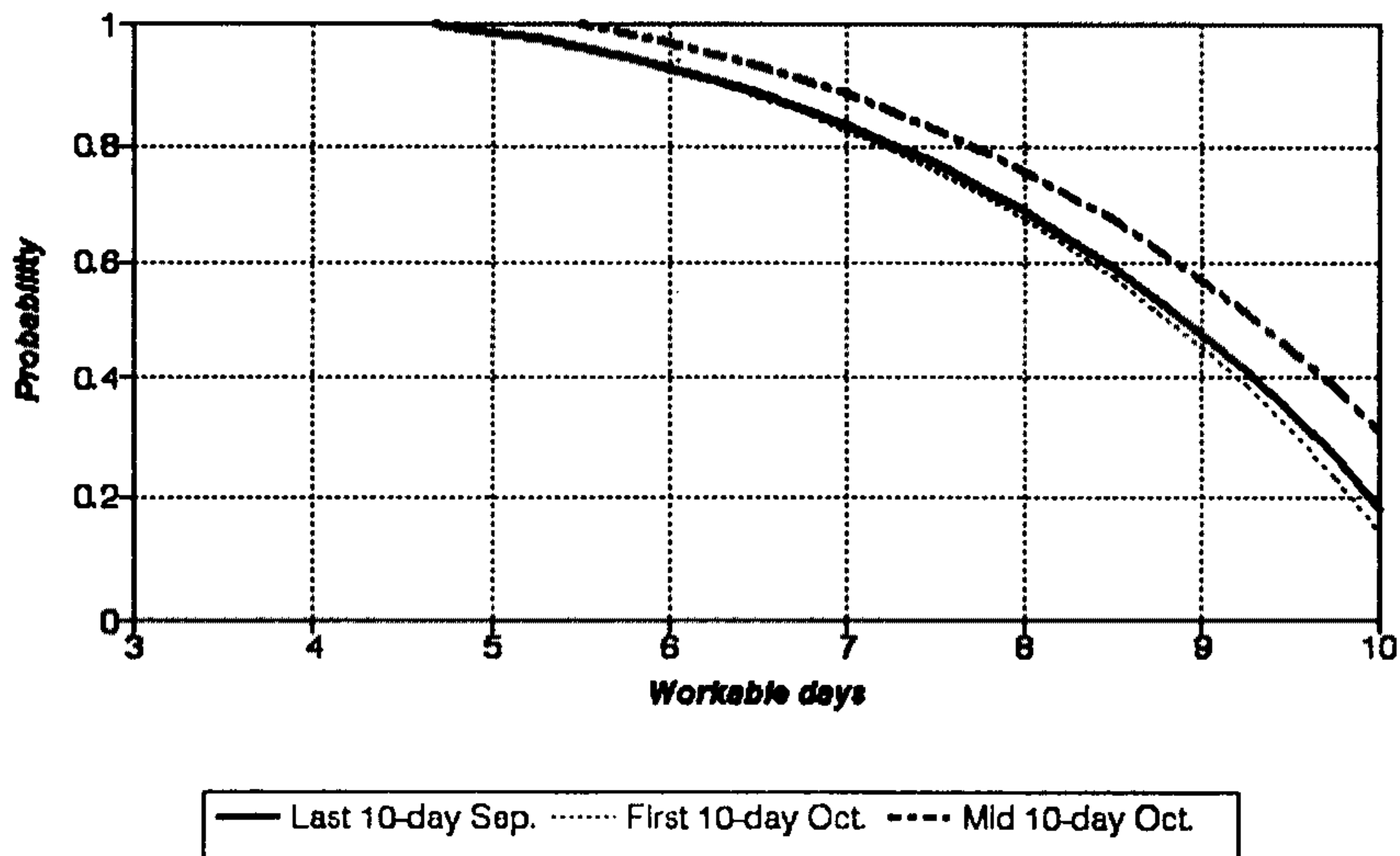


그림 3 - 1. 평택地域의 收穫作業可能日數確率分布曲線 (9월하순 - 10월중순)

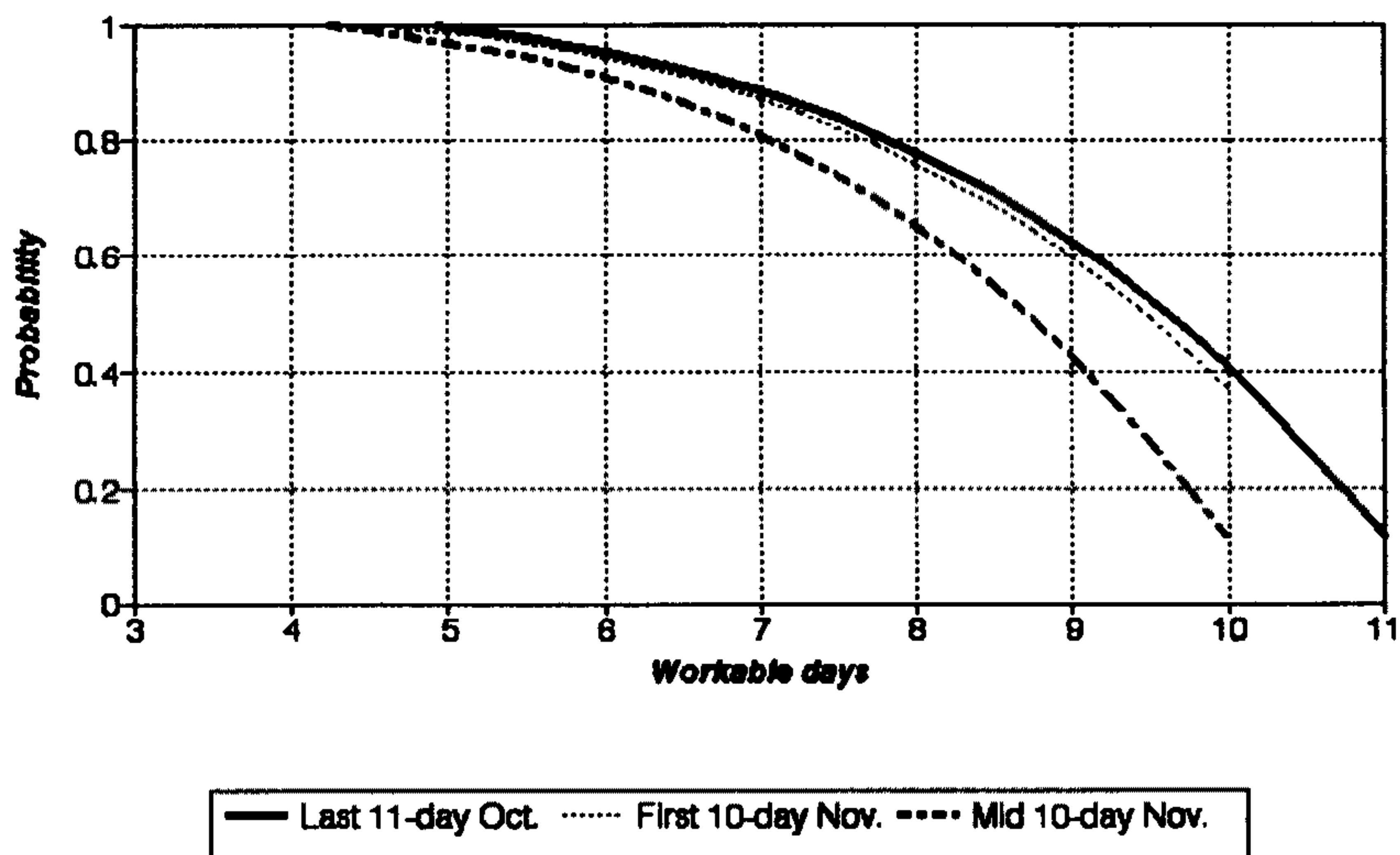


그림 3 - 2. 평택地域의 收穫作業可能日數確率分布曲線 (10월하순 - 11월중순)



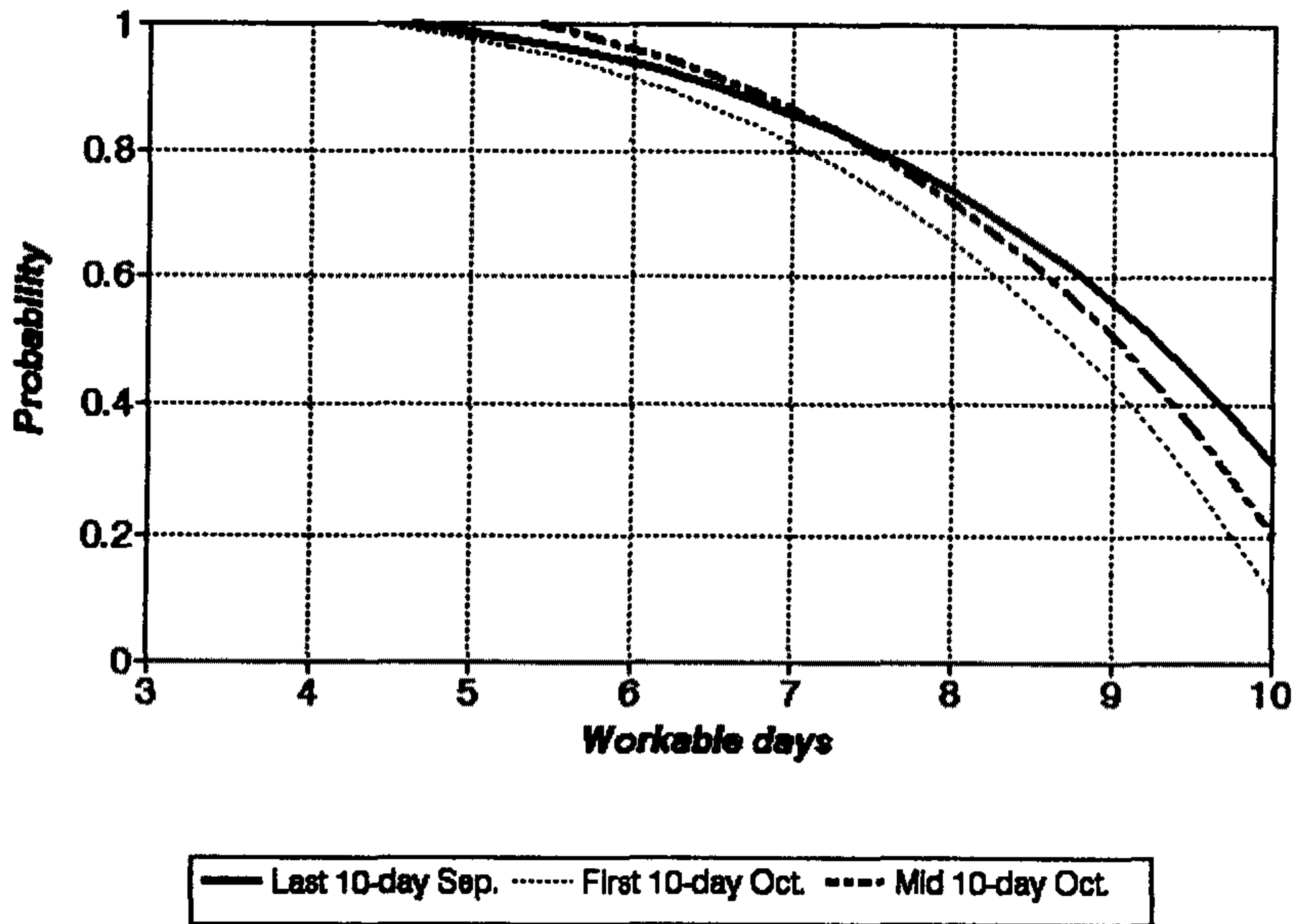


그림 3 - 3. 이천地域의 收穫作業可能日數確率分布曲線

(9월하순 - 10월중순)

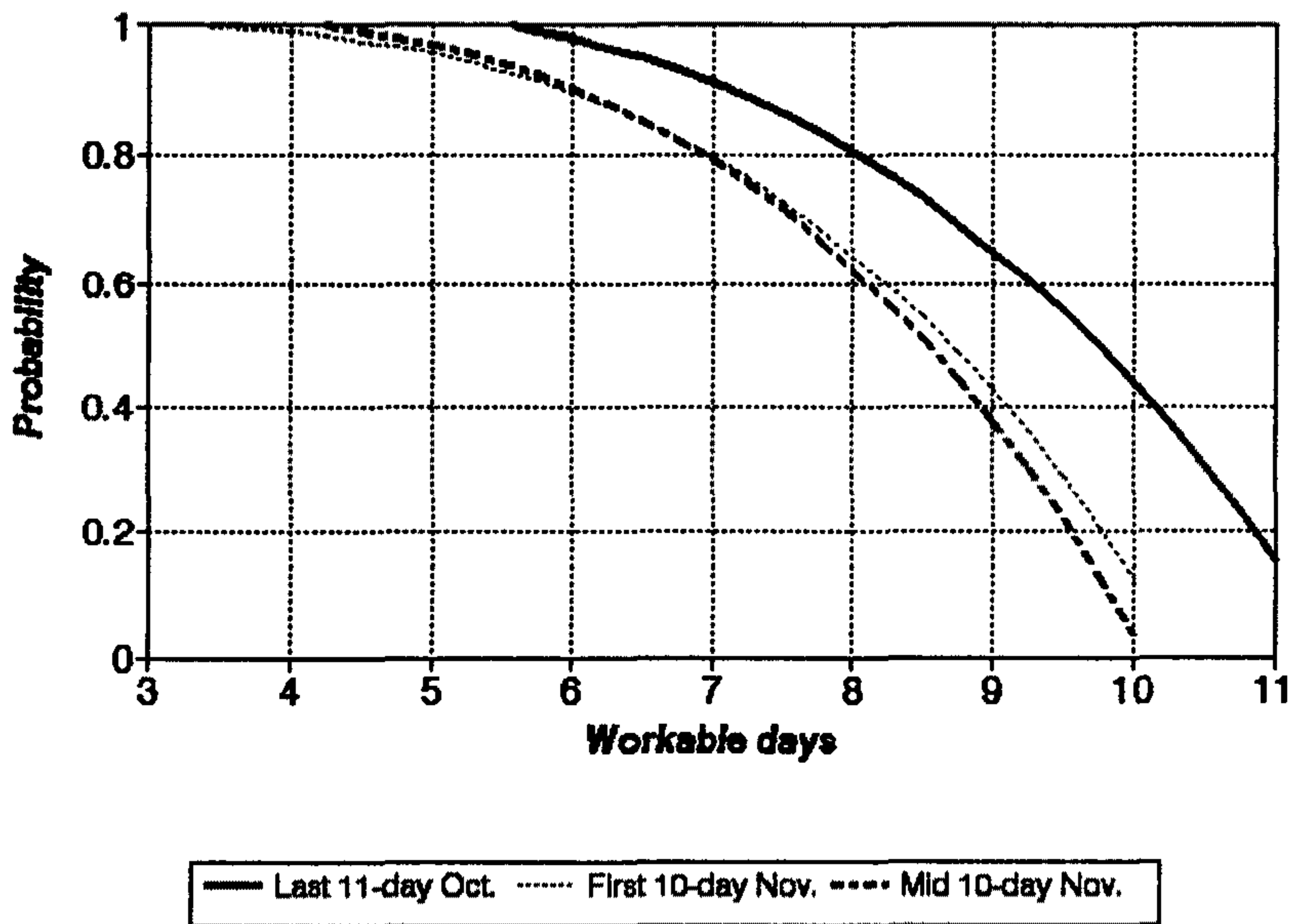


그림 3 - 4. 이천地域의 收穫作業可能日數確率分布曲線

(10월하순 - 11월중순)

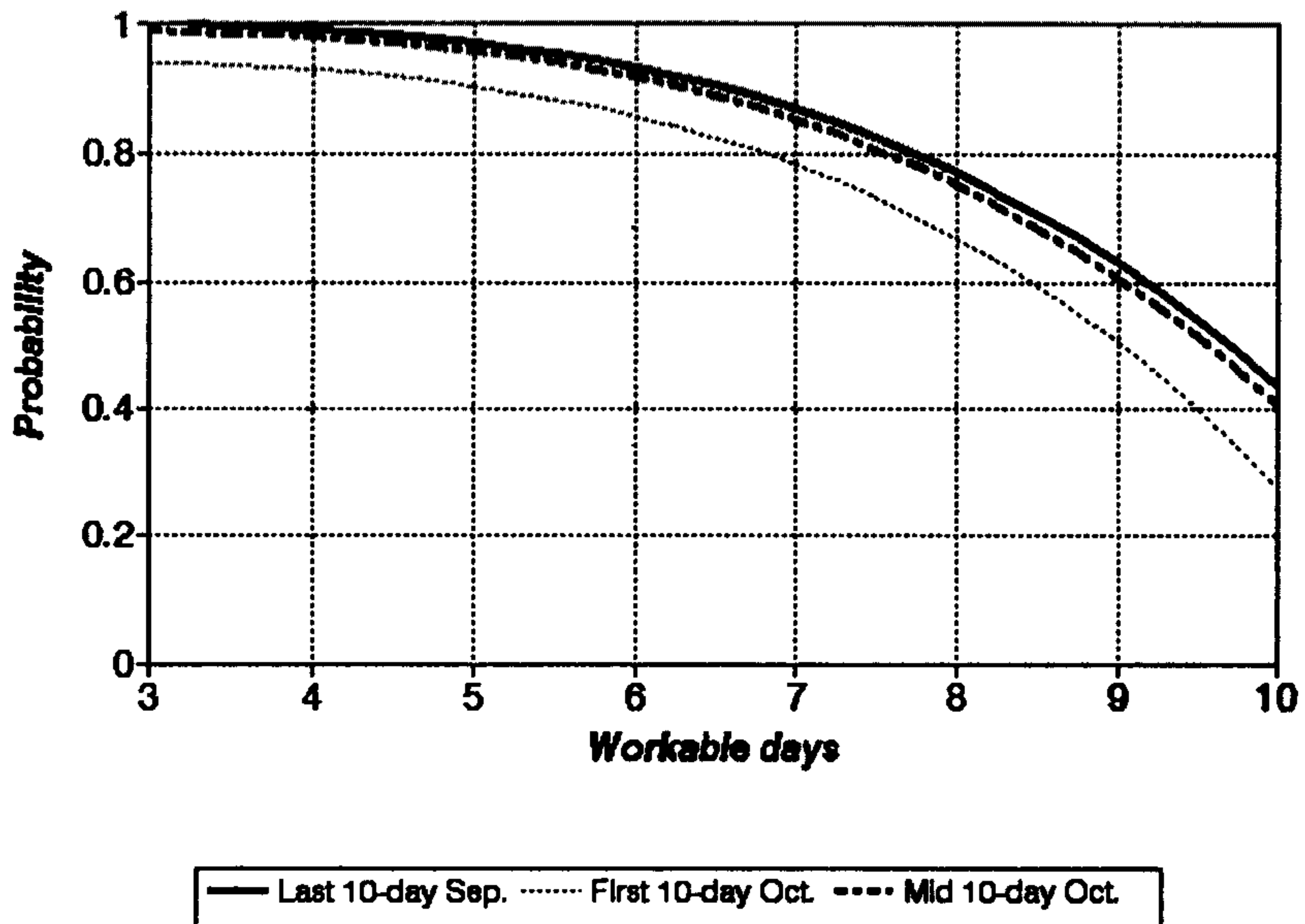


그림 3 - 5. 김포地域의 收穫作業可能日數確率分布曲線

(9월하순 - 10월중순)

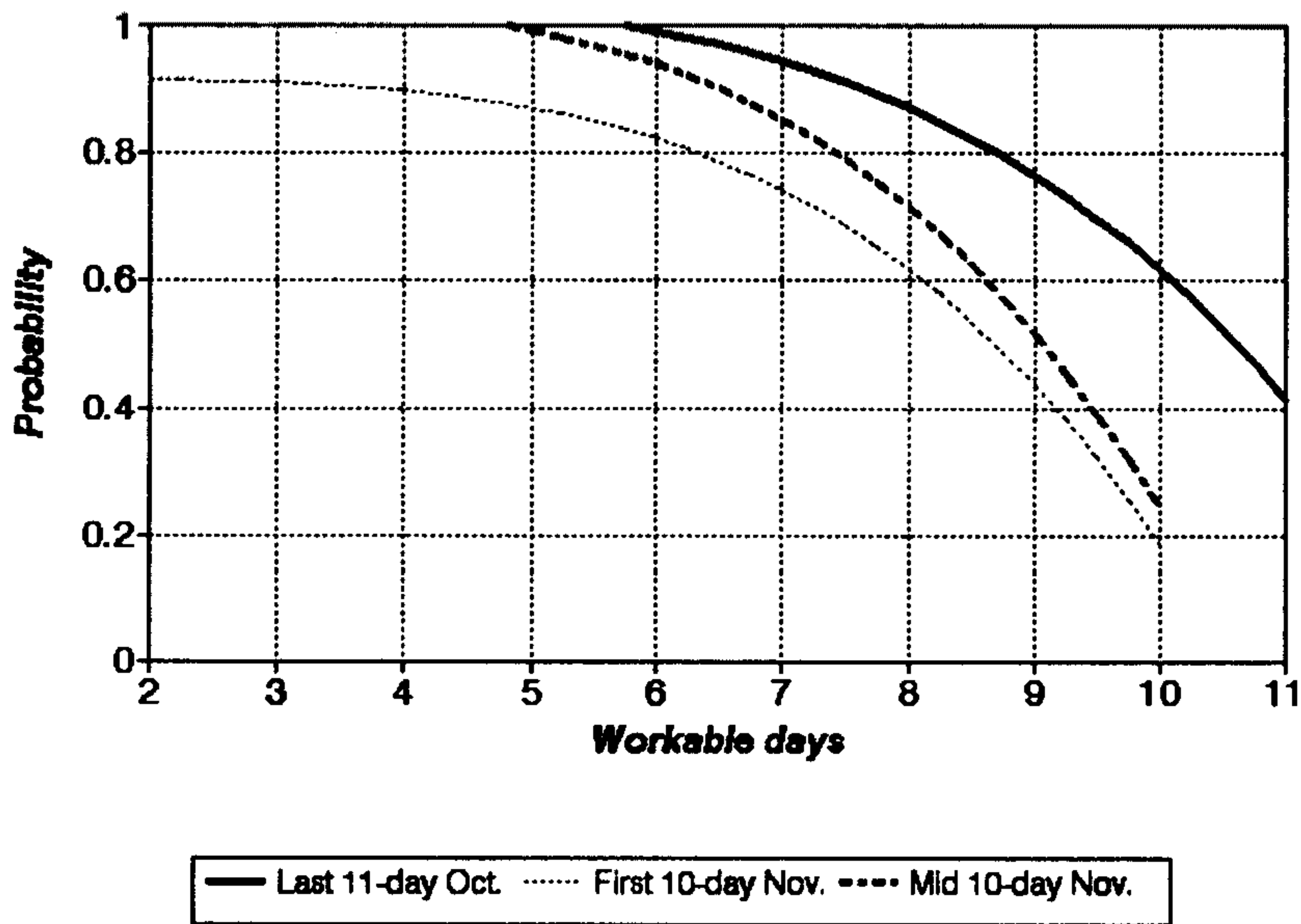


그림 3 - 6. 김포地域의 收穫作業可能日數確率分布曲線

(10월하순 - 11월중순)

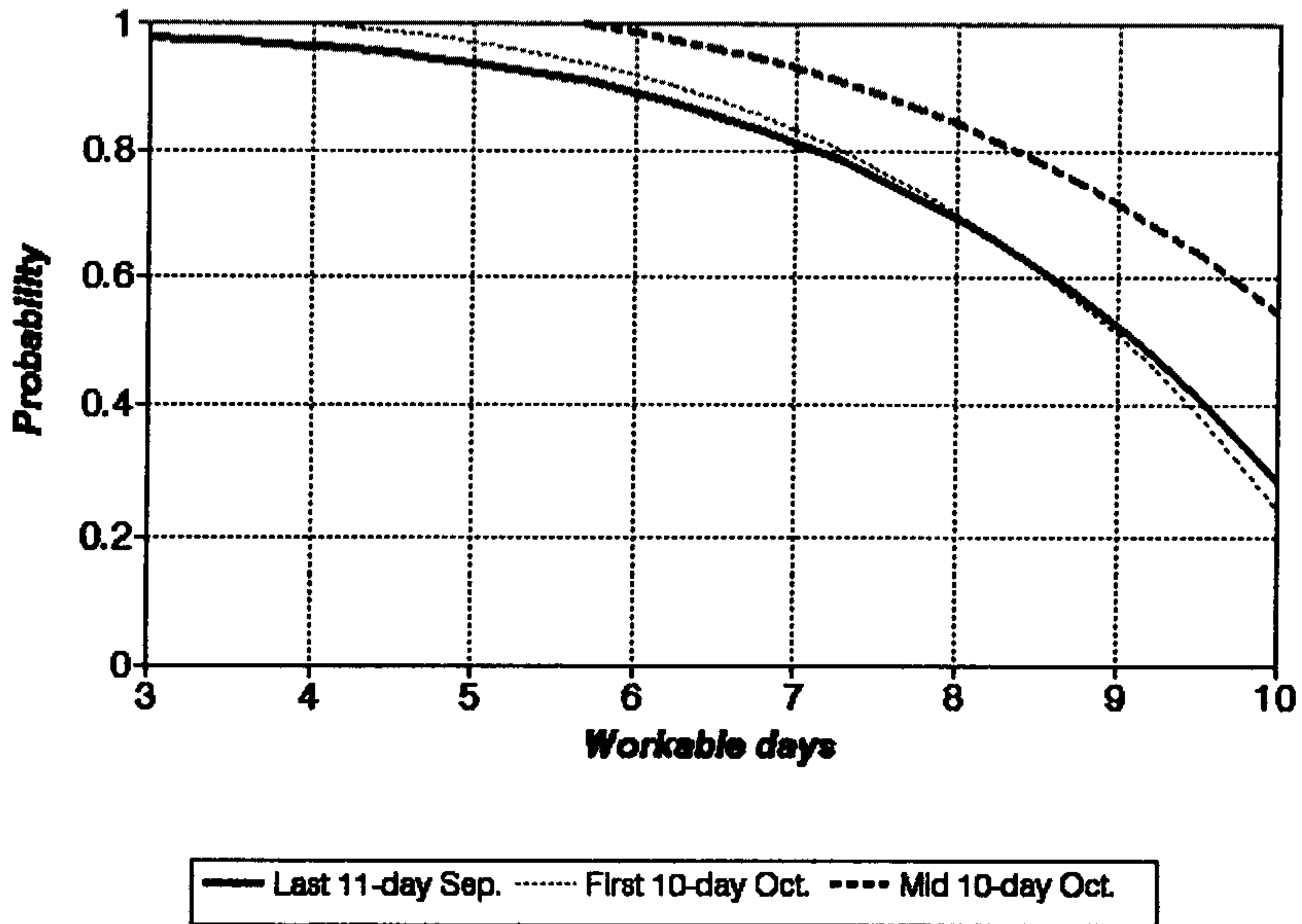


그림 3 - 7. 공주地域의 收穫作業可能日數確率分布曲線

(9월하순 - 10월중순)

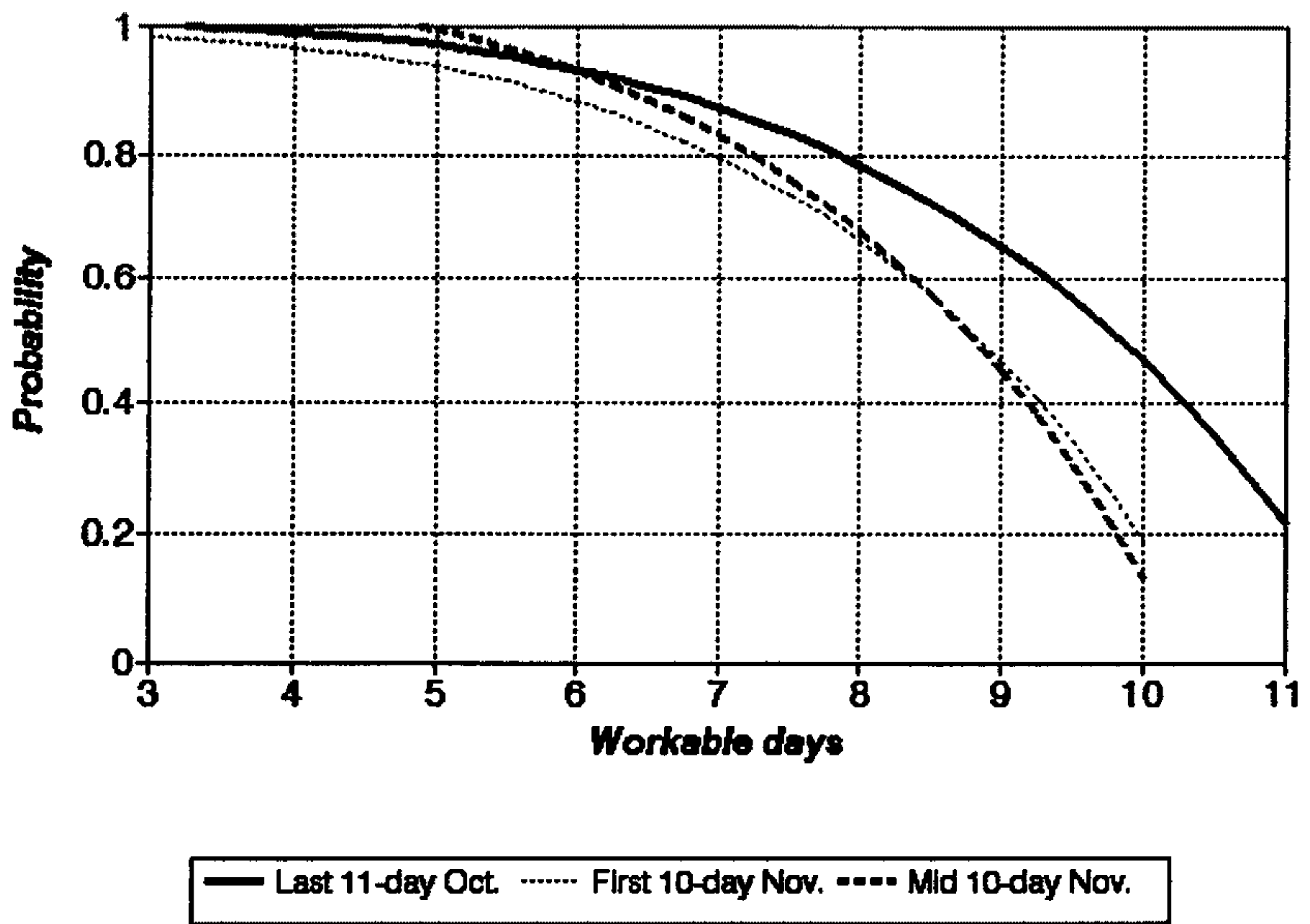


그림 3 - 8. 공주地域의 收穫作業可能日數確率分布曲線

(10월하순 - 11월중순)

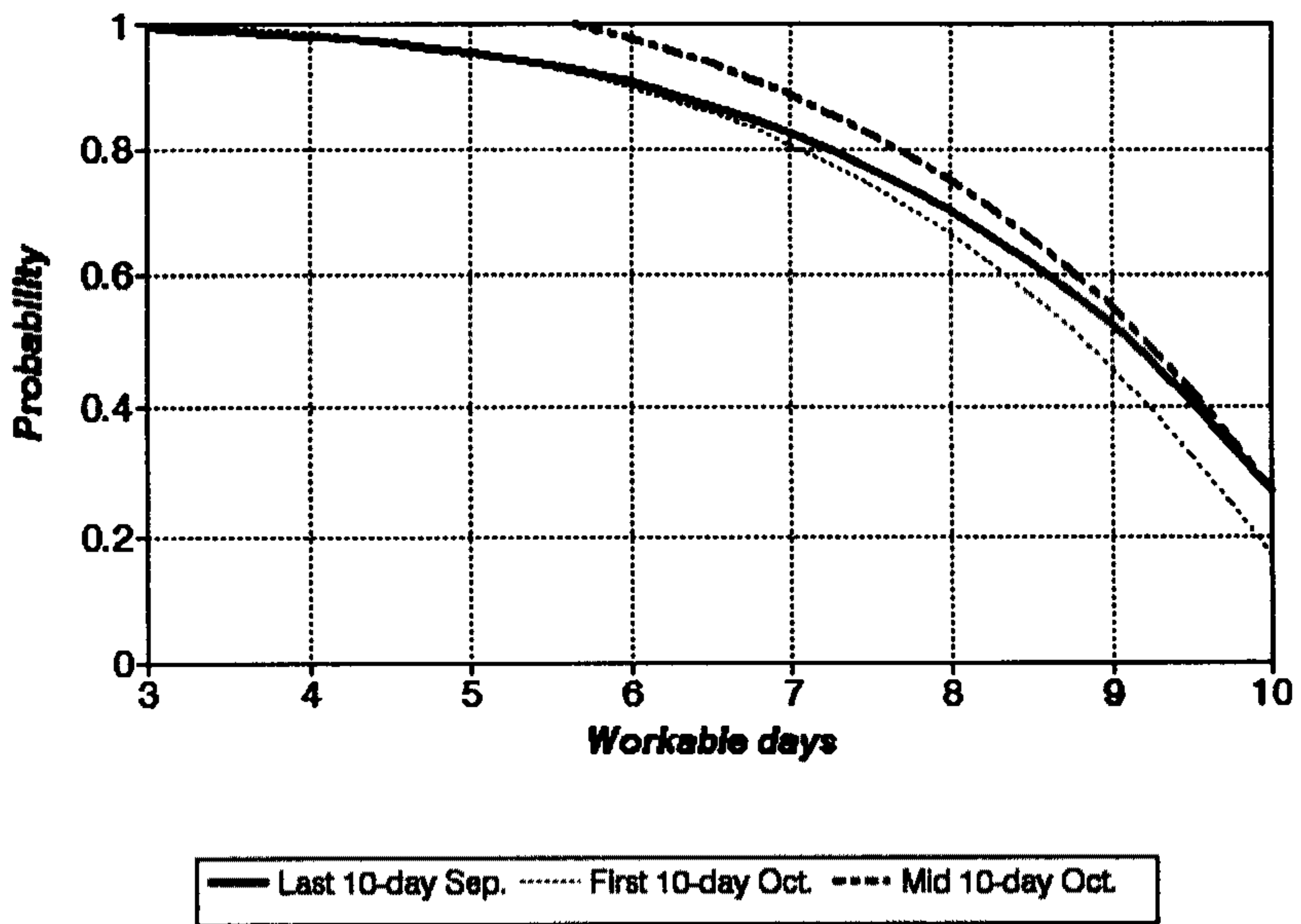


그림 3 - 9. 서산地域의 收穫作業可能日數確率分布曲線

(9월하순 - 10월중순)

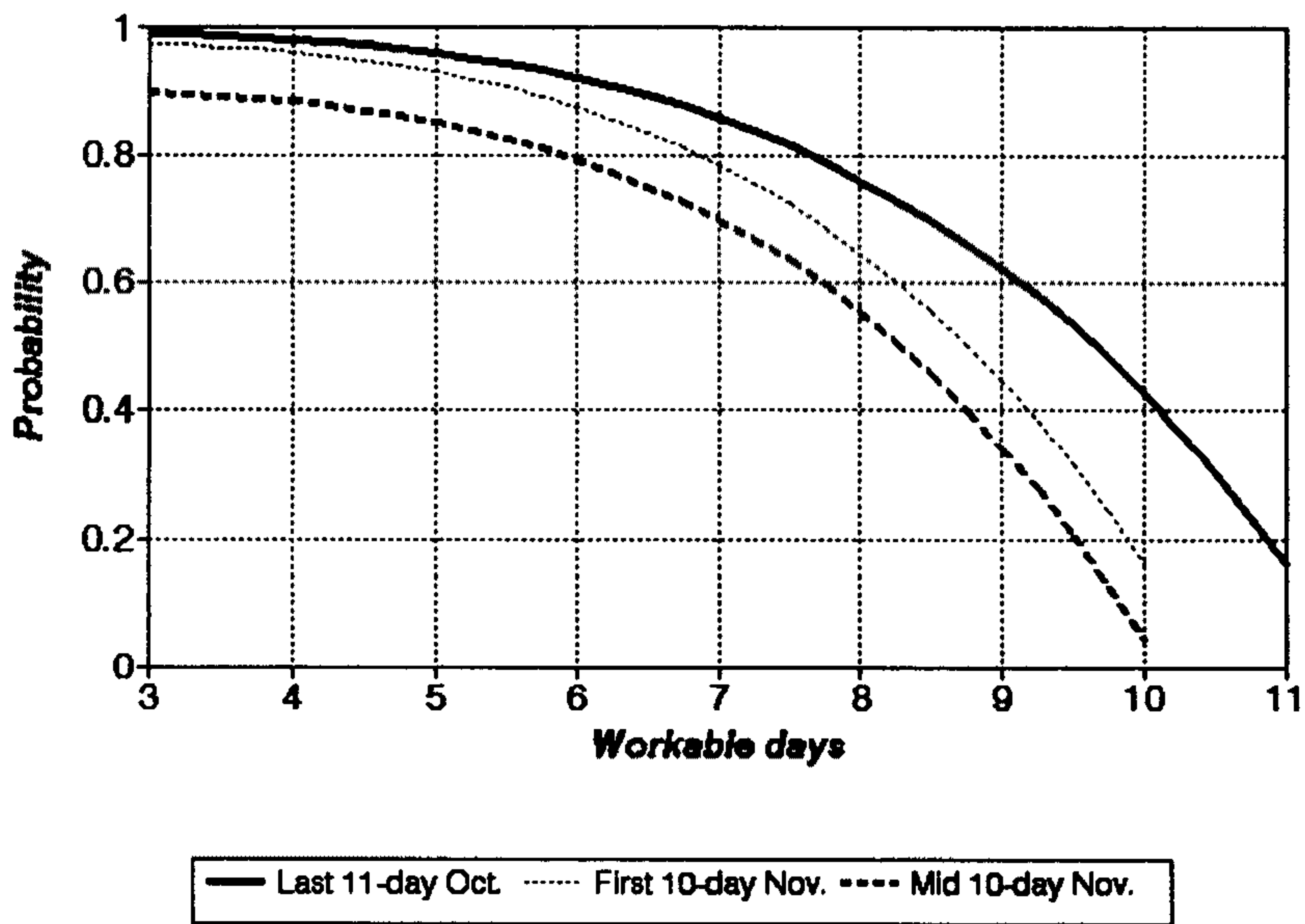


그림 3 - 10. 서산地域의 收穫作業可能日數確率分布曲線

(10월하순 - 11월중순)

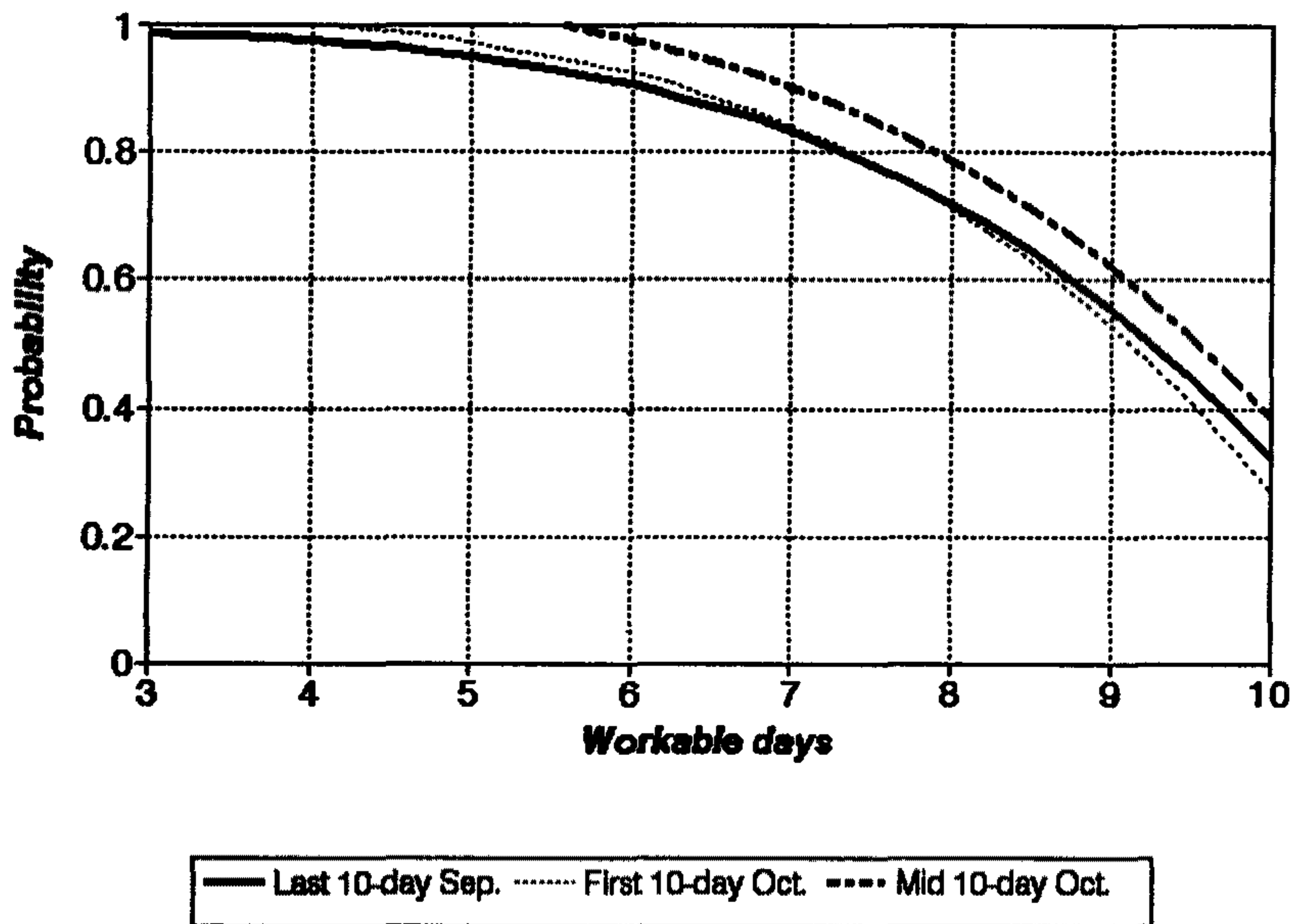


그림 3 - 11. 논산地域의 收穫作業可能日數確率分布曲線

(9월하순 - 10월중순)

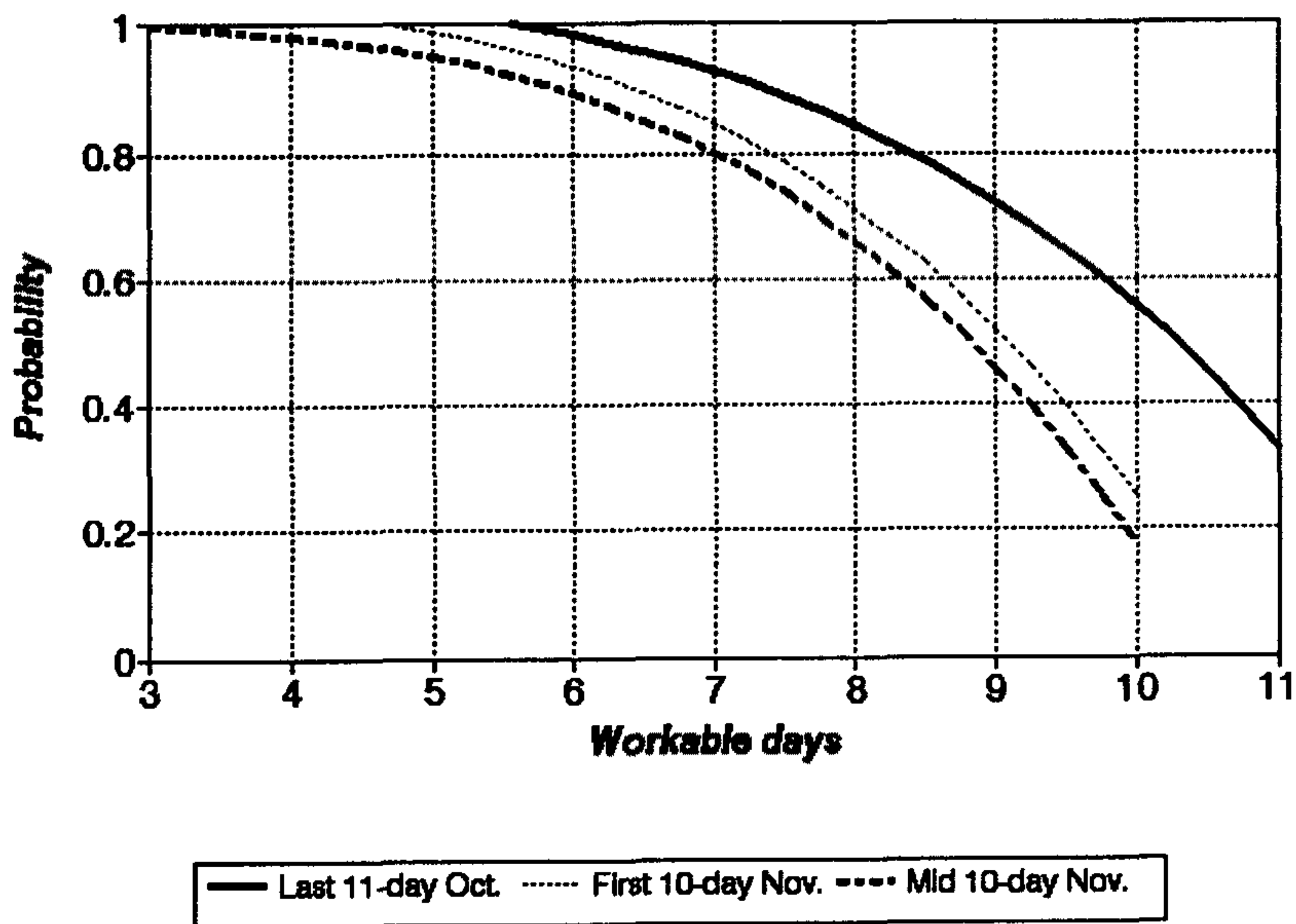


그림 3 - 12. 논산地域의 收穫作業可能日數確率分布曲線

(10월하순 - 11월중순)

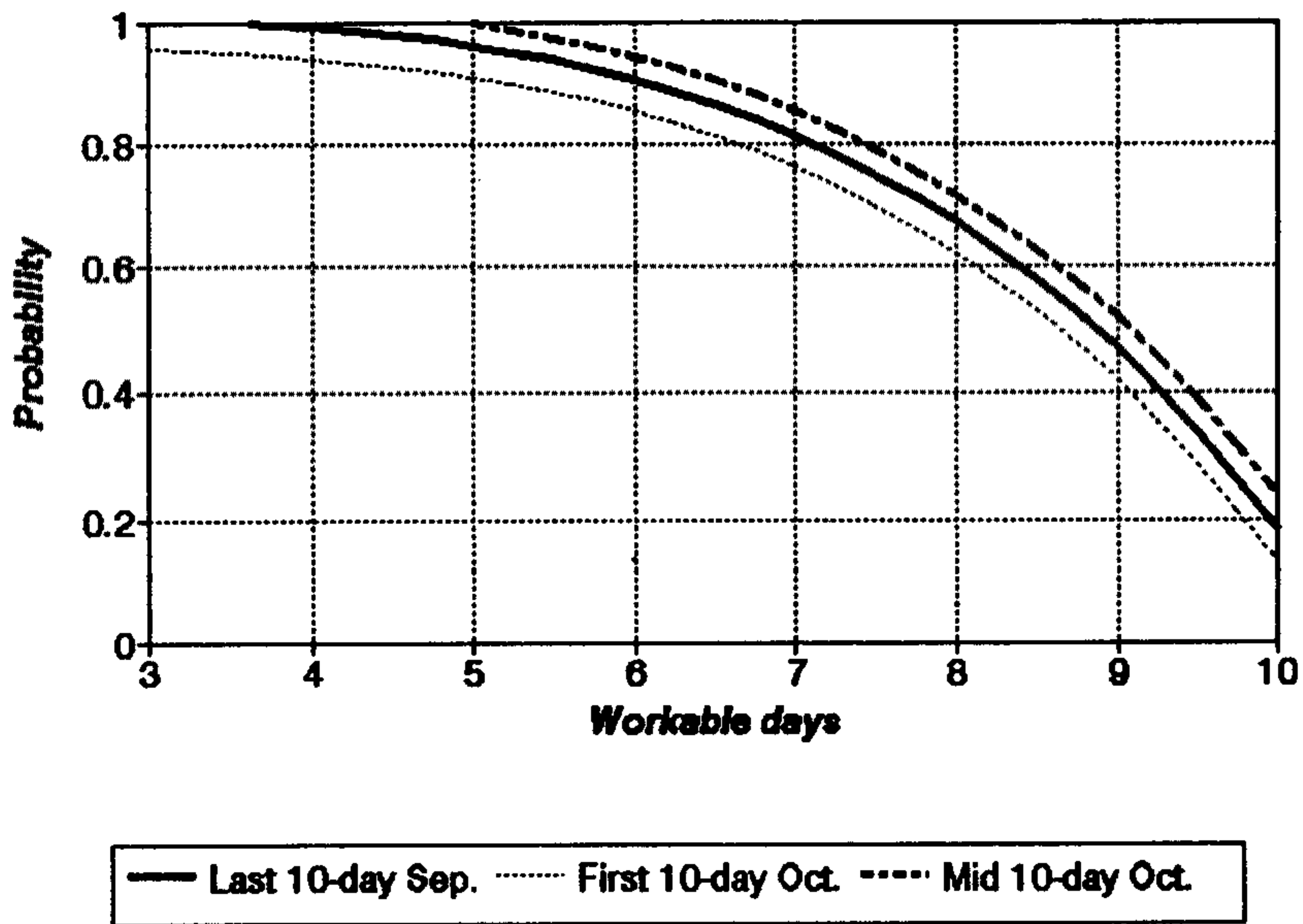


그림 3 - 13. 정읍地域의 收穫作業可能日數確率分布曲線  
(9월하순 - 10월중순)

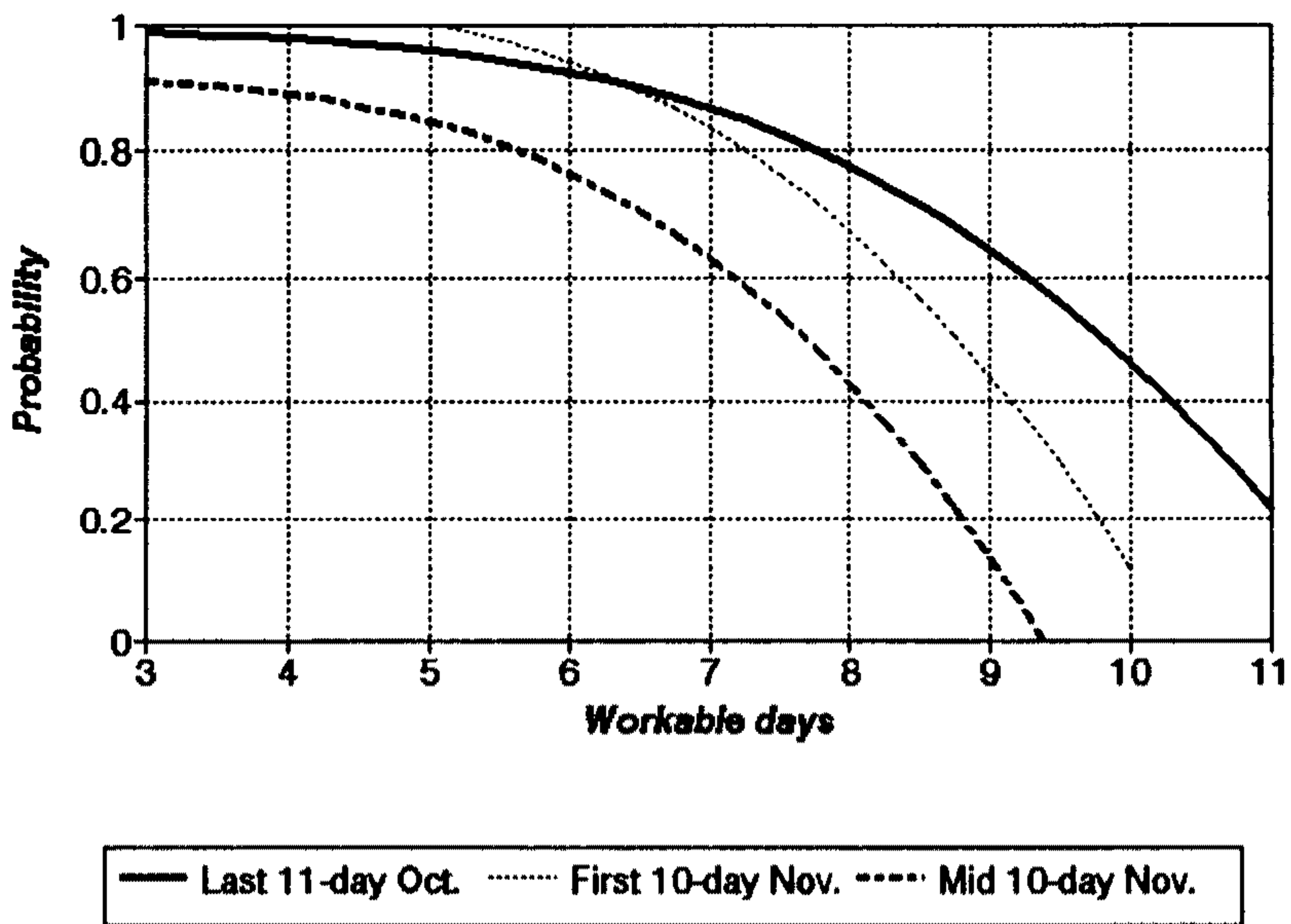


그림 3 - 14. 정읍地域의 收穫作業可能日數確率分布曲線  
(10월하순 - 11월중순)



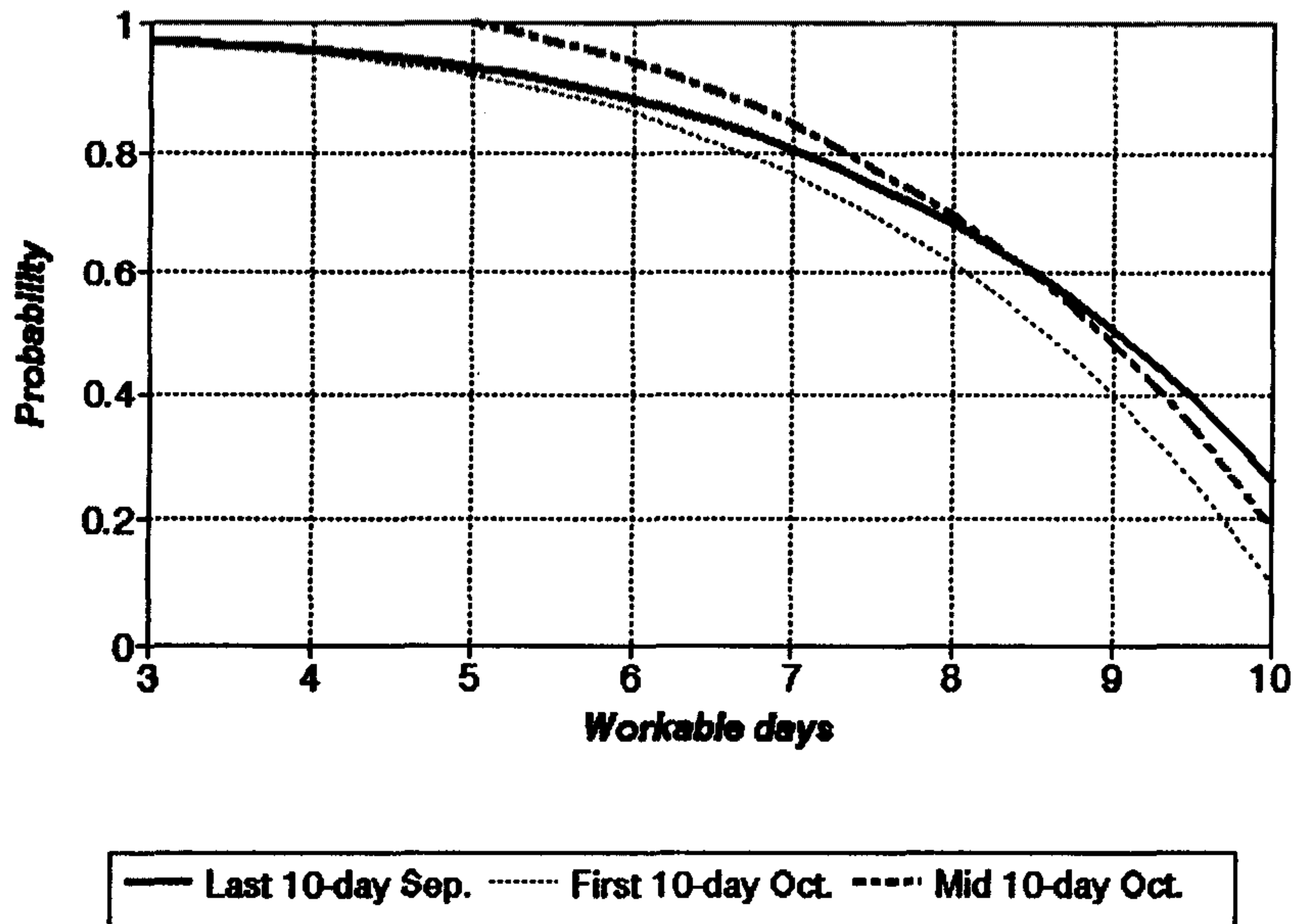


그림 3 - 15. 김제地域의 收穫作業可能日數確率分布曲線

(9월하순 - 10월중순)

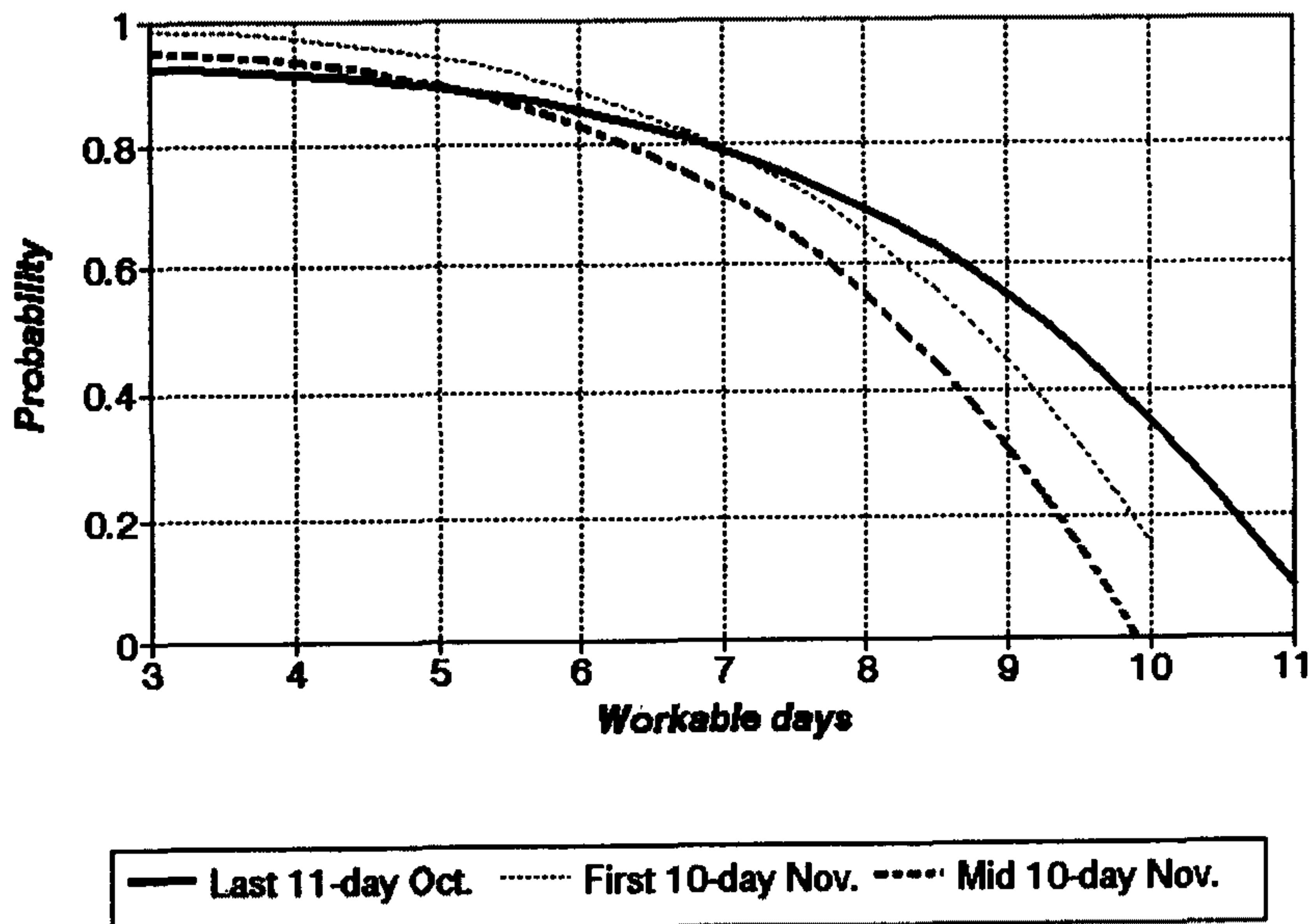


그림 3 - 16. 김제地域의 收穫作業可能日數確率分布曲線

(10월하순 - 11월중순)



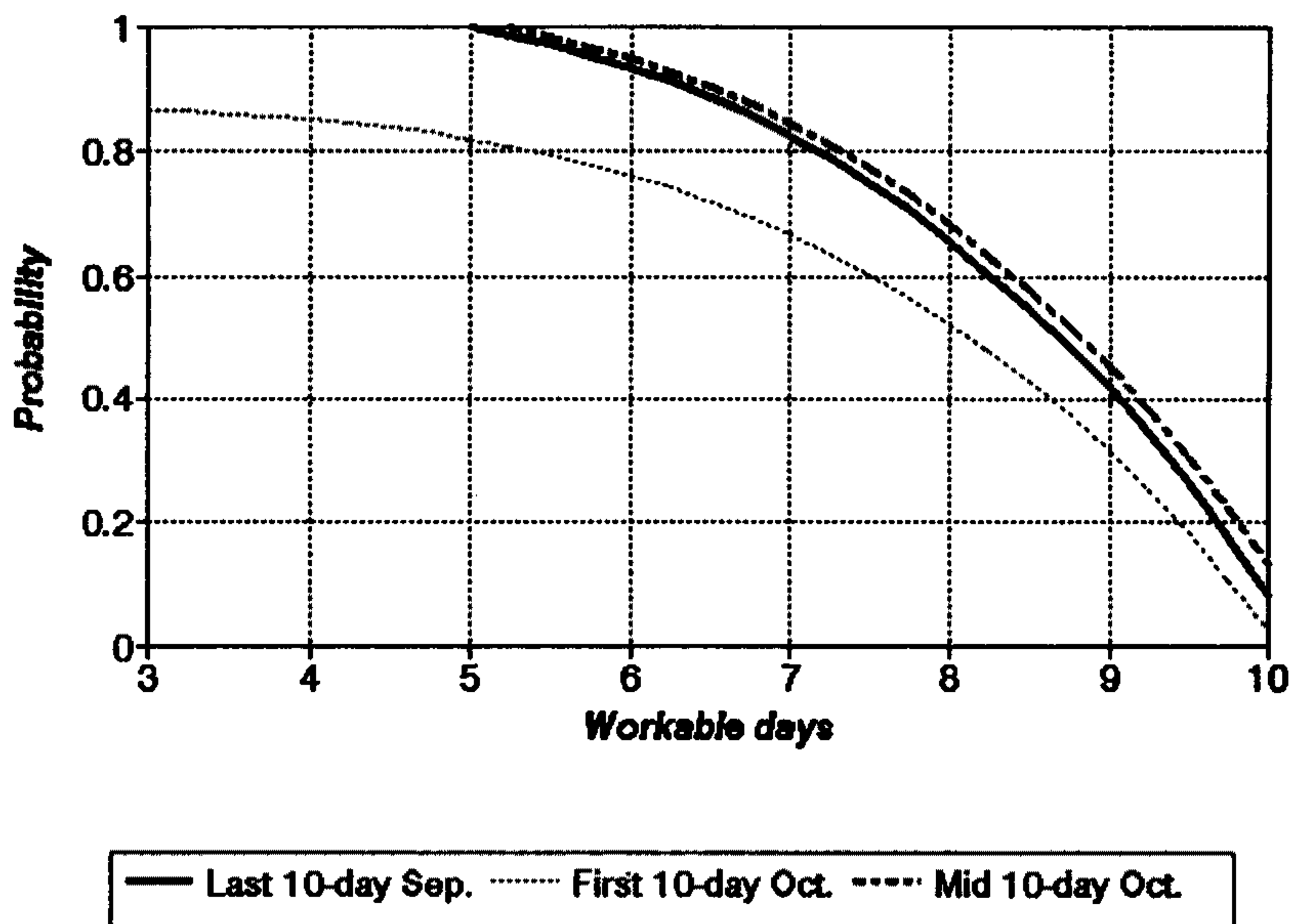


그림 3 - 17. 나주地域의 收穫作業可能日數確率分布曲線

(9월하순 - 10월중순)

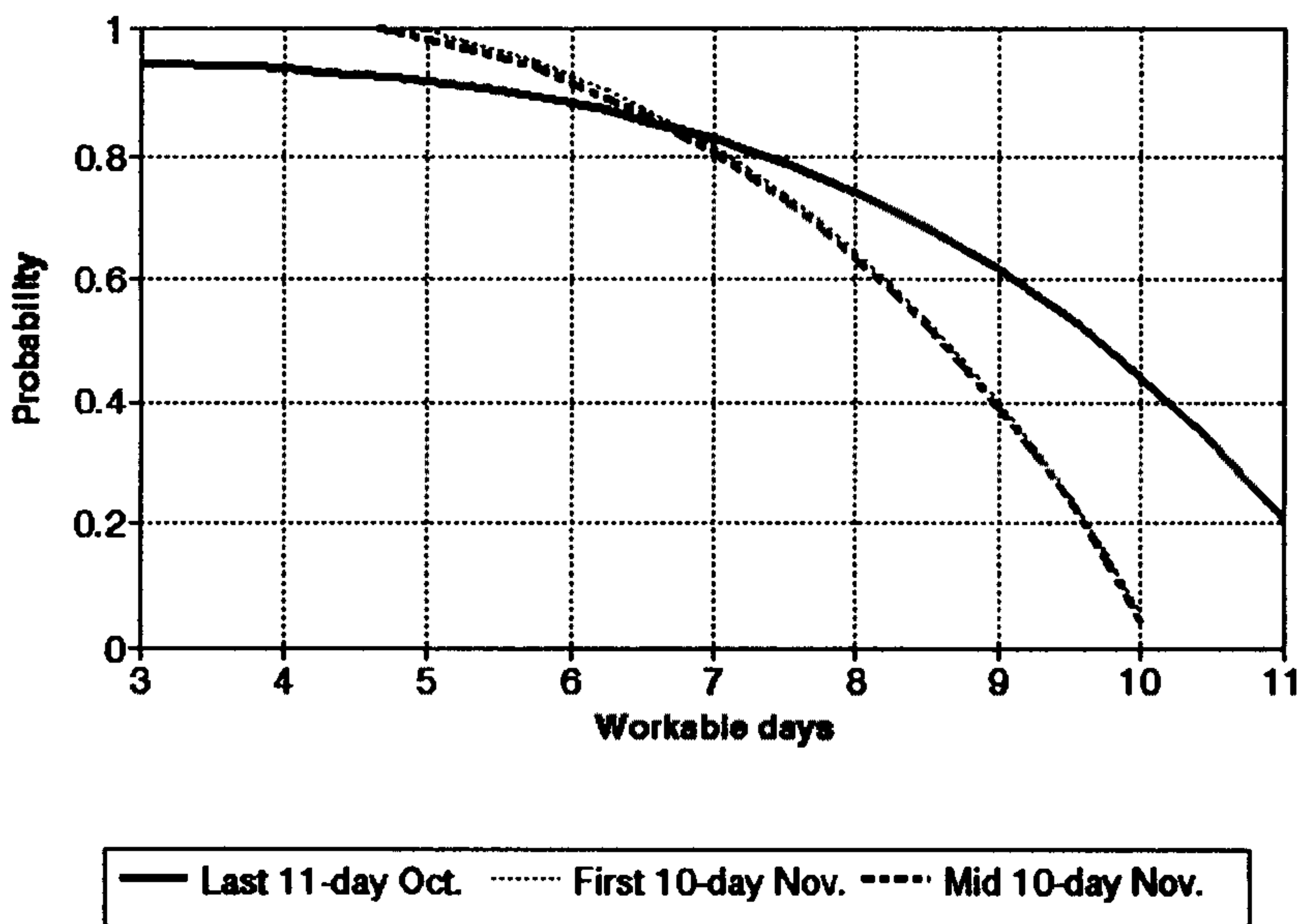
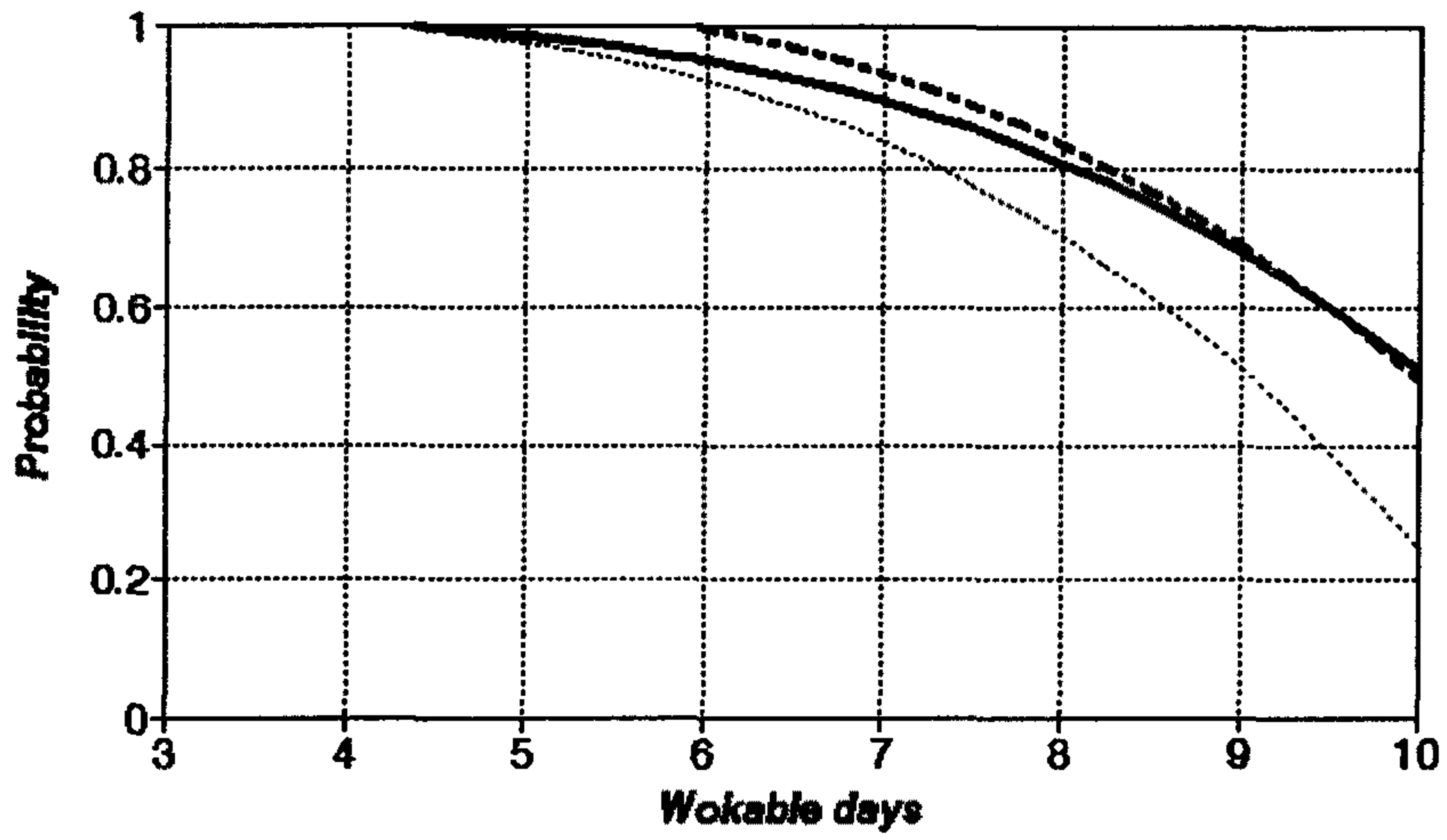


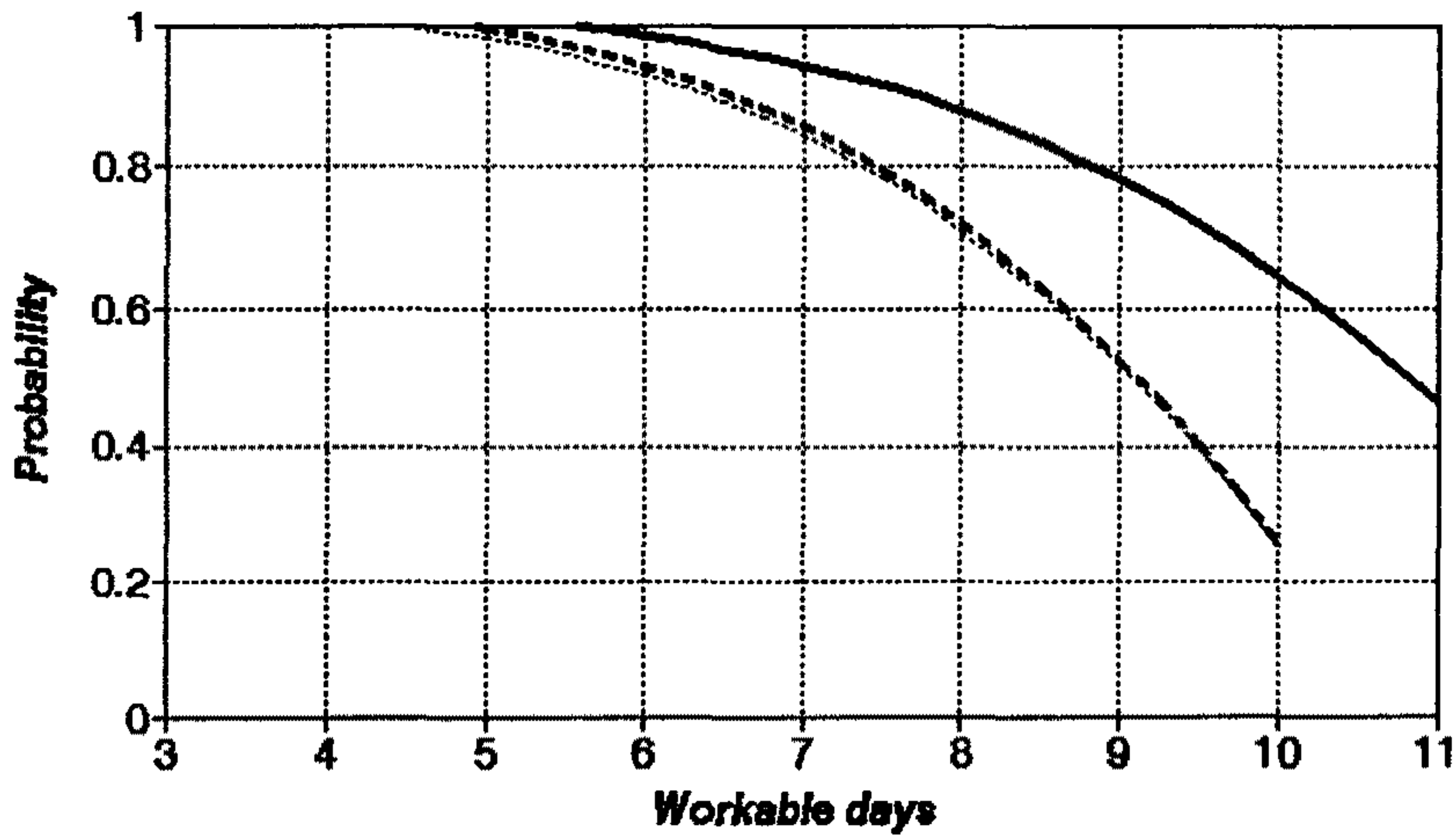
그림 3 - 18. 나주地域의 收穫作業可能日數確率分布曲線

(10월하순 - 11월중순)



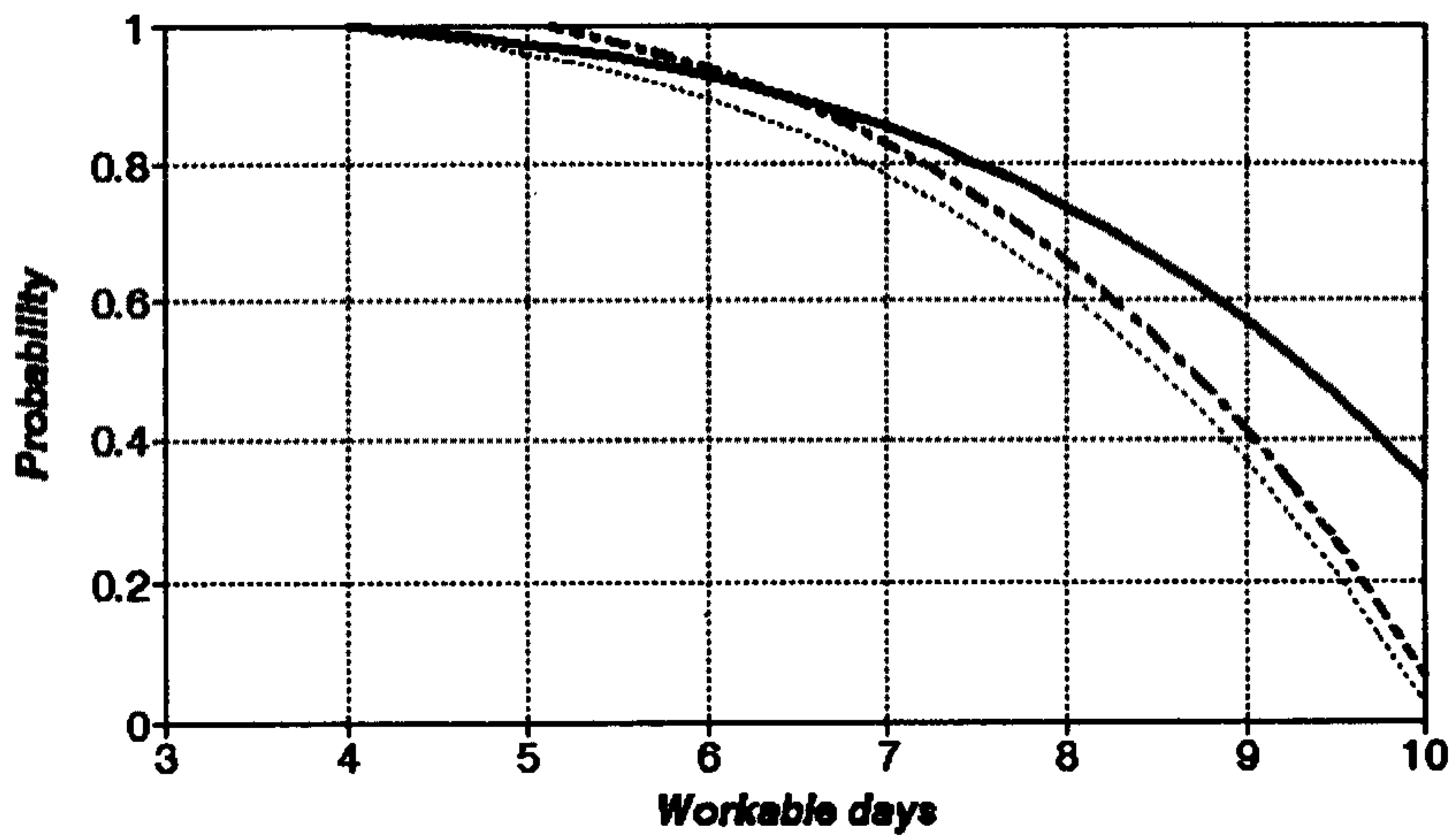
— Last 10-day Sep.    ..... First 10-day Oct.    - - - - - Mid 10-day Oct.

그림 3 - 19. 칠곡地域의 收穫作業可能日數確率分布曲線  
(9월하순 - 10월중순)



— Last 11-day Oct.    ..... First 10-day Nov.    - - - - - Mid 10-day Nov.

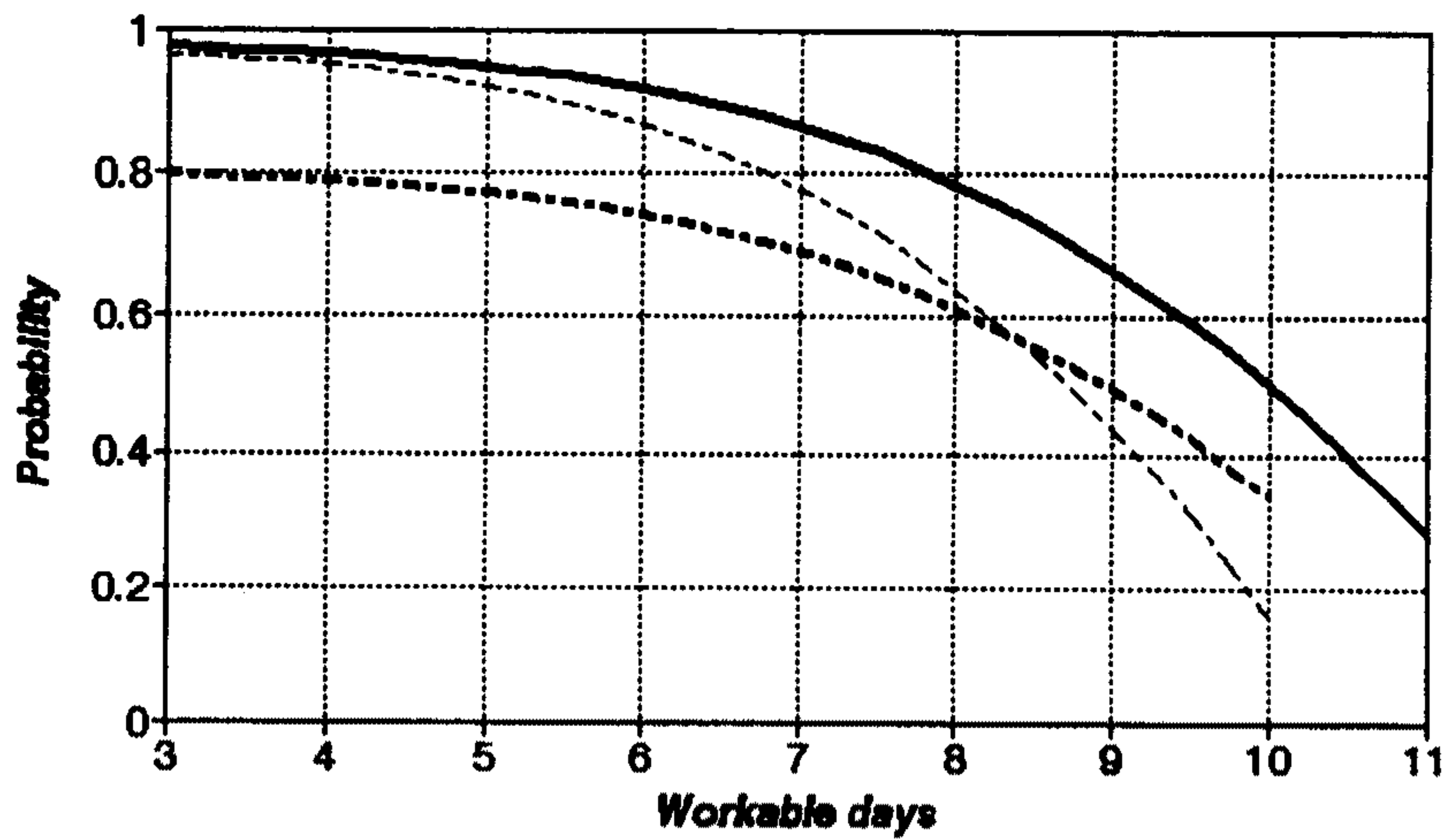
그림 3 - 20. 칠곡地域의 收穫作業可能日數確率分布曲線  
(10월하순 - 11월중순)



— Last 10-day Sep.    ..... First 10-day Oct.    - - - - - Mid 10-day Oct.

그림 3 - 21. 영일地域의 收穫作業可能日數確率分布曲線

(9월하순 - 10월중순)



— Last 11-day Oct.    ..... First 10-day Nov.    - - - - - Mid 10-day Nov.

그림 3 - 22. 영일地域의 收穫作業可能日數確率分布曲線

(10월하순 - 11월중순)

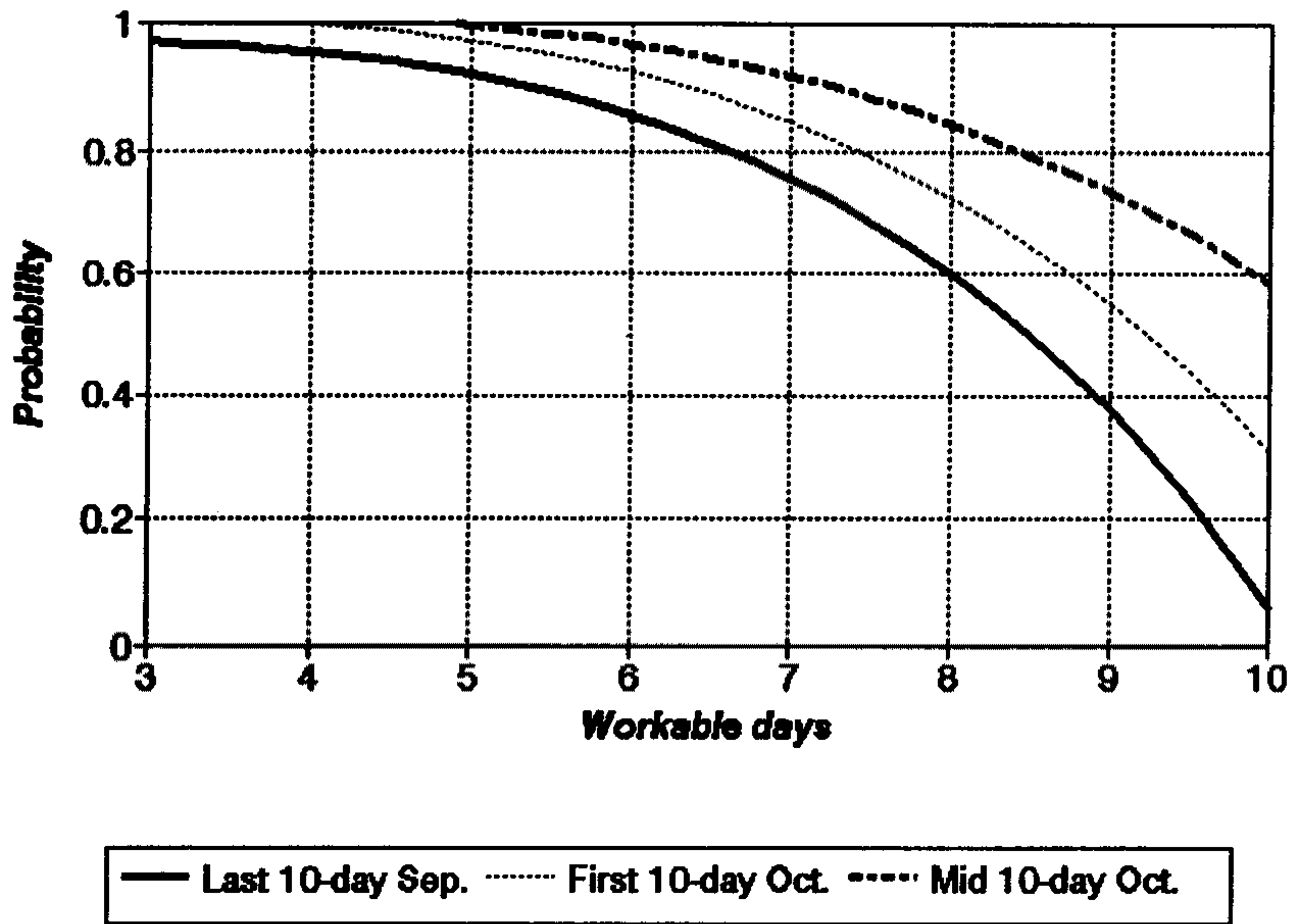


그림 3 - 23. 김해地域의 收穫作業可能日數確率分布曲線

(9월하순 - 10월중순)

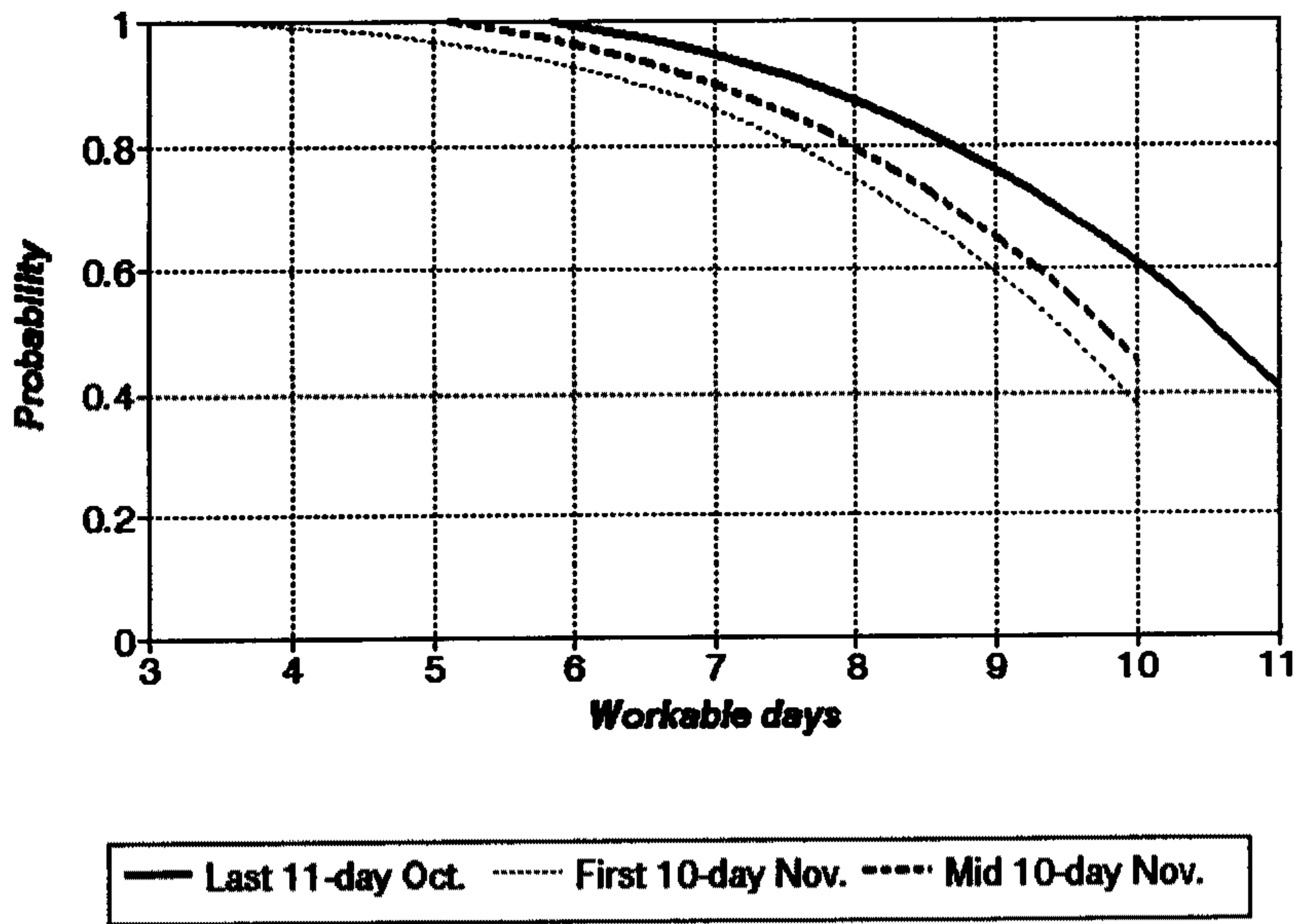


그림 3 - 24. 김해地域의 收穫作業可能日數確率分布曲線

(10월하순 - 11월중순)

### 3. 地域別收穫作業可能日數率

그림 (3 - 1)에서 그림 (3 - 24)까지의 確率分布에 立脚하여 作業計劃의 安定性を 높이기 위하여 66 %의 期待確率値를 적용하여 作業可能日數率을 결정하였으며 그 분포 결과는 표 3-5 에서 보는 바와 같다.

표 3 - 5. 地域別, 月別 및 旬別 收穫作業可能日數率 (%)

지 역	지 역 명	9 월	10 월			11 월	
		하순	상순	중순	하순	상순	중순
경 기도	평 택	82.1	81.2	86.2	80.5	86.1	83.3
	이 천	85.3	80.4	83.6	82.0	79.8	78.6
	김 포	88.8	81.4	87.4	89.6	80.4	83.6
충 청 도	공 주	82.9	83.2	94.3	81.9	80.7	81.3
	서 산	83.2	80.8	85.4	80.2	79.7	73.8
	논 산	84.6	83.8	88.4	86.2	83.7	80.7
전 라 도	정 읍	81.4	78.2	83.8	89.6	81.2	68.8
	김 제	82.2	78.0	82.5	75.8	79.9	74.7
	나 주	80.4	71.4	81.7	79.7	79.8	79.3
경 상 도	칠 곡	92.1	83.2	92.3	90.7	83.5	84.1
	영 일	85.7	78.2	80.4	82.8	79.3	75.5
	김 해	77.3	84.8	96.1	88.8	86.8	90.1

作業可能日數率의 確率分布를 分析했을 때 旬別 作業可能日의 分布는 地域과 作業時期에 따라 큰 차이가 있었으며 一般的인 分布傾向은 단적으로 표시할 수 없을 정도로 극히 다양한 形態를 보였다.

실제로 作業적기의 作業可能日數率は 신뢰도 75% - 85% 사이에 존재한다.



## 第 4 章 農作業計劃의 데이터베이스化

### 4.1 데이터베이스의 概要

일상 생활에서 發生하는 多樣하고 많은 情報를 選別 分析하여 特定 分野에 利用하는 方法은 소규모 마이크로 컴퓨터에서 대형컴퓨터에 이르기까지 거의 모든 컴퓨터에 廣範圍하게 利用되고 있다. 獲得한 데이터를 拔萃, 要約하여 이를 컴퓨터 기억 장치에 貯藏한 후 使用者의 必要에 따라 情報를 處理하여 쉽게 어떤 目的에 利用하는 것이다.

컴퓨터를 使用하여 데이터를 處理함에 있어서 考慮해야 할 중요한 事項은 어떻게 데이터를 貯藏하고 또한 어떻게 效果적으로 利用할 수 있게 하는 方法을 찾아내는 것이다. 데이터를 컴퓨터에 貯藏한 것을 데이터 화일(data file)이라 한다. 어떤 特定 分野에 必要한 데이터를 수집하여 貯藏시킨 후 使用者의 必要에 따라 拔취하여 利用하기 쉽도록 情報를 서로 聯關된 것끼리 모아 데이터 화일을 構築한다. 즉, 必要한 데이터를 統括적인 觀點에서 서로 關係된 情報끼리의 重複을 최소화 하여 한 기억장소에 모아 貯藏함으로써 다수의 使用者로 하여금 동시에 必要한 情報를 共有할 수 있도록 만들어 놓은 情報의 集合體를 데이터 베이스(data base)라고 한다. 다시 말하면, 데이터베이스는 어떤 特定한 目的에 使用하기 便利하도록 構成되고 제공되는 情報의 모임이라 할 수 있다. 여기에는 必要로하는 모든 데이터를 데이터 베이스에 貯藏해야 하고 데이터를 效果的으로 處理 및 利用할 수 있는 어떤 方法으로 기억 장치에 데이터 베이스를 貯藏해야 한다.



## 4.2 데이터베이스화의 目的

未耕地整理된 耕地整理 對象地域의 경지정리 專門家 시스템을 具現하기 위해서는 먼저 專門家 시스템이 學習할 데이터가 必要하다. 이러한 데이터를 利用하여 專門家 시스템은 경지정리 대상 지역에서의 모든 條件을 判斷하여 스스로 대상지역에서 농업기계의 利用 효율을 높힐 수 있는 最適의 耕地整理形狀을 映像으로 具現하도록 한다.

이러한 데이터를 경지구모 및 작업별 포장효율 분석과 專門家 시스템에 利用하기 위해서는 시스템이 스스로 학습할 수 있는 형식으로 구현되어야 한다. 따라서, 必要한 모든 資料를 데이터베이스화하여 專門家 시스템이 학습을 할 수 있도록 하는 것이 데이터베이스화의 주된 目的이 된다.

일반적으로 흔히 使用되고 있는 데이터베이스와는 그 目的이 다르다고 할 수 있다. 一般的인 데이터베이스는 廣範圍한 資料를 컴퓨터를 利用하여 일반적인 使用者의 의도하는 目的에 알맞게 使用하기 便利하도록 만든 것이 보편적이며, 다수의 使用者가 同時に 많은 자료를 使用할 수가 있다.

本 研究에서 開發한 데이터베이스는 使用者가 사람이 아닌 컴퓨터 즉 專門家 시스템이 된다고 할 수 있다. 따라서, 본 연구에서는 專門家 시스템이 데이터를 學習하기 쉽게 만드는 것이 데이터베이스화 作業의 가장 주된 焦點이 되었다.

### 4.3 데이터베이스의 具現

방대한 데이터를 處理하여, 의미를 갖는 情報로 가공하자면 반드시 이들 데이터를 處理/管理하는 시스템이 必要하다. 本 研究에서는 위에서 언급한 目的을 실현하기 위해서 보편적인 常用 프로그램인 데이터베이스용 PC 소프트웨어인 DbaseIII<sup>+</sup>, DbaseIV, Lotus 123, Rbase, Professional File, Q&A, Paradox, Quattro, Foxbase 중에서 DbaseIII<sup>+</sup>를 使用하여 데이터베이스를 完成하였다. 이 데이터베이스용 PC 소프트웨어인 DbaseIII<sup>+</sup>는 한 화일의 크기는 項目數가 32 - 400개, 코드 수는 10억까지 가능하므로 PC용 데이터베이스의 容量은 充分하다고 判斷된다.

dBASEIII+는 미국 에스턴 테이트사에 의해 dBASEII, dBASEIII에 이어 퍼스널 컴퓨터 IBM PC용으로 개발된 소프트웨어로, 각종 데이터 管理 및 處理 업무에 使用된다. 현재 여러가지 종류의 데이터베이스용 소프트웨어가 나와 있지만 주로 dBASEIII+가 많이 使用되고 있다. dBASEIII+는 비교적 使用法이 간단하여 초보자나 專門家에게 모두 유용하게 活用되고 있다.

dBASEIII+는 외형적으로는 하나의 완성된 소프트웨어의 形態를 띄고 있으나 실질적으로는 컴퓨터 프로그래밍의 技能도 동시에 遂行될 수 있다. 즉, dBASEIII+에는 각종 데이터 處理를 프로그램 形式으로 작성하여 使用할 수 있다. 실제로 본 데이터베이스는 dBASEIII+의 後者의 특성인 프로그래밍 기능을 利用하여 構築되었다.

실제로 내년엔 完成할 경지정리 기초설계를 위한 專門家 시스템에서 이러한 데이터베이스 소프트웨어인 DbaseIII<sup>+</sup>가 專門가 시스템과 데이터의 相互 互換이

불가능할 경우에는 기존의 다른 데이터베이스 소프트웨어도 互換이 불가능하리라고 판단이 된다. 따라서, 本 研究에서는 互換이 불가능할 경우에는 다른 方法을 使用하여 데이터베이스를 완성시킬 計劃이다. 그것은 현재 모든 프로그래밍 언어 중에서 互換性이 가장 뛰어난 Turbo C++ 을 利用하여 직접 데이터베이스 프로그램을 만들어서 使用할 計劃이다. 여기에서는 Turbo C++ 언어로 專門家 시스템을 위한 데이터베이스를 構築할 수 있느냐가 問題인데 本 研究에서는 Turbo C++를 利用하여 간단한 데이터베이스를 만들고 있는 중이다. Turbo C++를 利用한 데이터베이스가 완성된다면 互換의 문제는 간단히 解決되리라고 判斷된다. 本 研究에서 利用할 專門家 시스템도 C 언어로 構築되어질 計劃이기 때문이다.

## 4.4 데이터베이스 構成

dBASEIII+를 使用하는 데 있어서 情報의 가장 基本이 되는 최소 단위는 데이터이다. 가령 트랙터의 구체적인 가격인 19,700,000, 제조회사의 구체적인 이름인 대동, 국제 등이 이에 해당한다. 데이터를 구성하는 각 항목이 필드(field)이며 本 데이터 베이스에서 MAKER, MODEL, PRICE(won) 등이 해당된다. 각각의 필드가 모여 하나의 레코드(record)가 구성된다. 즉, MACHINE이라는 하나의 레코드 아래 MAKER, MODEL, PRICE(won) 등의 필드가 包含된다. 이들 여러개의 같은 종류의 레코드들이 모여 하나의 데이터 베이스 화일(data base file)이 구성된다.

트랙터, 경운기, 콤바인, 이앙기 등을 각각 다른 데이터 베이스 화일에 貯藏하면 이 모든 데이터 베이스 화일들이 모여 하나의 總括的인 데이터 베이스가 되는 것이다. 제공된 디스켓내의 확장명이 DBF로 되어 있는 것은 데이터 베이스 화일이며, 확장명이 PRG와 FMT로 되어 있는 것은 dBASEIII+의 컴퓨터 언어의 성격을 利用하여 작성한 프로그램들이다. 실제로 구성된 데이터베이스의 構造를 보면 그림 4-1과 같이 이루어져 있다.

구성된 전체 데이터베이스의 主프로그램은 AG001.PRG이다. 主프로그램의 메뉴를 보면 0번부터 8번까지의 선택事項이 나오고 각각의 선택事項에 따른 副 프로그램인 POWER0.PRG, TRPT1.PRG...등이 작동된다. POWER0.PRG나 TRPT1.PRG의 뒷부분 번호인 0,1은 실제로 메뉴번호에 해당한다. 그 다음 레벨인 TRAC001.PRG, TILL001.PRG등은 실제 情報를 處理하기위한 프로그램으로 副 프로그램인 TRPT1.PRG가 작동하면 다시 使用者가 어떤 情報를 얻을 것인가에 따라 TRAC001.PRG, TILL001.PRG등이 작동하게 된다. FMT가 확장명으로 붙은 화일들은 데이터의 입력시나 조회시의 화면의 상태를 설정해주는 프로그램이다. 001이 붙은 화일들이 작동할때 FMT, DBF가 확장명으로 되어있는 화일들을 참조로 하여 데이터가 照會, 貯藏, 處理된다. 실례를 들어보자. 0~8까지의 메뉴가 나오면 使用者는 1번인 트랙터, 경운기를 선택한다. 이때 TRPT1.PRG가 작동된다. 다시 승용트랙터를 선택하면 TRAC001.PRG가 작동되고 이때 입력 화면의 형태는 TRACTORFMT를 참조로, 자료의 구조나 貯藏은 TRACTOR.DBF를 參照로 이루어지는 것이다.

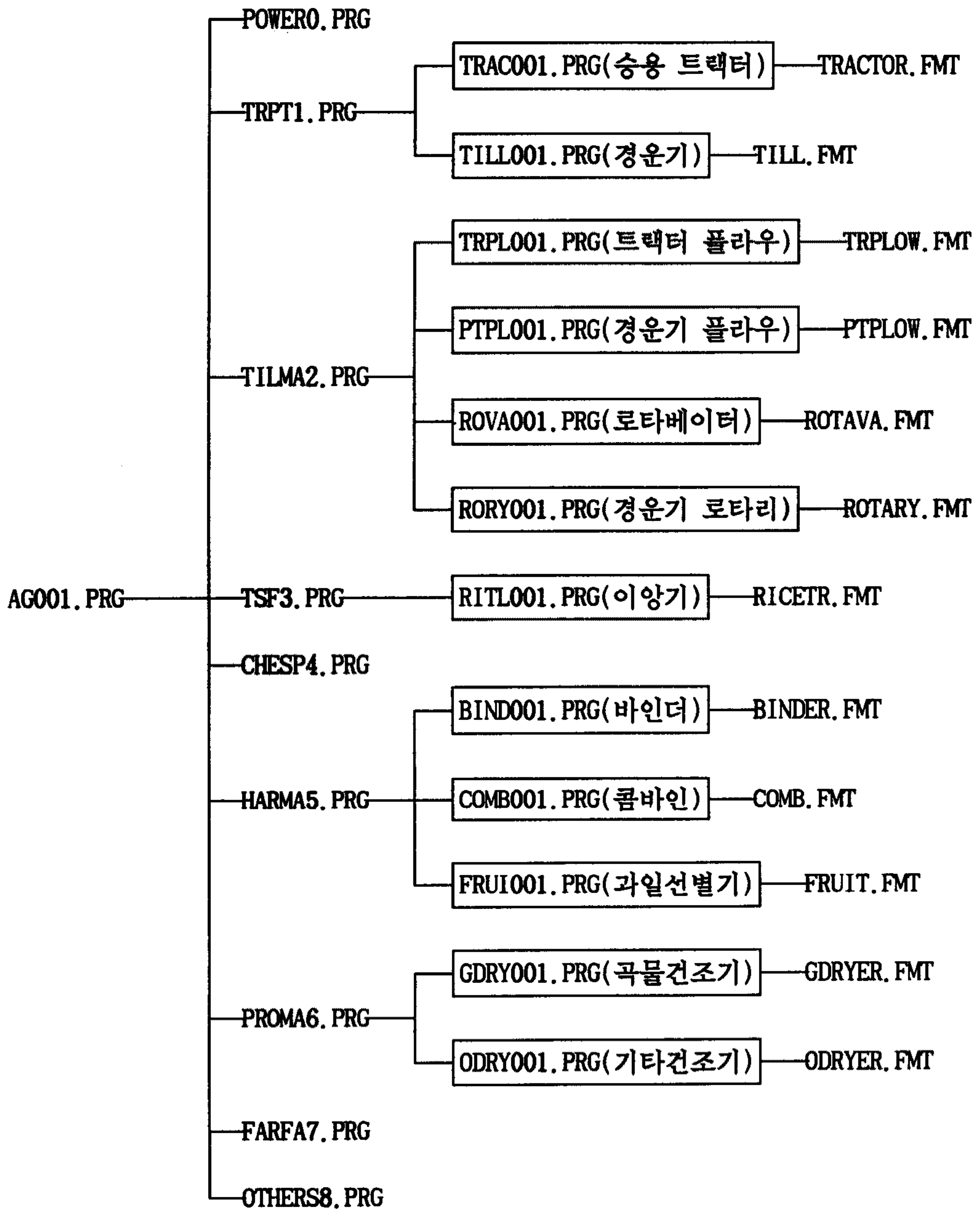


그림 4 - 1. 개발한 데이터베이스의 構造



## 4.5 데이터베이스화 작업

耕地의 形狀이나 크기를 決定하기 위해서는 대상 지역의 諸般 榮農條件을 考慮하여야 한다. 특히, 農業機械化에 초점을 맞추어서 耕地整理를 遂行한다면 각 作業別로 作業體系에 대한 투여 農機械에 대한 圃場作業效率 및 圃場作業能率이 考慮되어야 한다. 그리고, 대상지역의 地形 條件을 考慮하여야 한다. 만약, 대상지역에 불가피한 障礙物이 存在한다면 그 것 또한 중요한 考慮대상이 되며 대상지역의 高度差도 중요한 考慮대상이 될 수 있다. 그래서, 대상지역의 경지의 형상과 크기를 결정하는 데 考慮되는 모든 요소가 데이터베이스화가 되어야 한다. 그래서, 본 연구에서는 크게 세 가지로 분류하여 데이터베이스화 작업을 수행했다.

### 4.5.1 農機械, 農業施設, 裝置에 대한 데이터베이스화

모든 農作業에 소요되는 시설 및 장치 그리고 使用되는 농기계를 총망라하여 각각을 데이터베이스화 하였다. 특히, 農業機械에 대한 데이터베이스는 다른 데이터베이스와 데이터를 서로 互換할 수 있도록 具現을 하였다. 농업기계에 대한 데이터는 포장작업 효율 분석과 기계화 작업체계의 모형개발, 경지정리 기초설계를 위한 專門家 시스템 개발 등에서 가장 중요하고 많이 使用되이질 情報이기 때문이다.

에너지源, 農業機械, 農業施設 등은 그 종류가 많아 코드화하여 1차 分類하고, 1차 분류된 각 항목은 또 다시 항목별로 세분하여 2차 분류하였다. 그리고



2차분류된 항목들을 다시 세분하여 분류하였다.

제공된 디스켓을 컴퓨터의 A드라이브에 넣고 A드라이브의 DOS프롬프트 위에서 "db"를 입력하여 실행시키면 dBASEIII+의 화면이 나오고 다시 거기에 "do ag001"을 입력하여 실행시키면 dBASEIII+의 화면을 얻을 수 있다. 다음의 표는 實際 開發된 데이터베이스에서 승용트랙터를 선택한 것을 예로 보여준다.

표 4 - 1. 데이터베이스 초기 화면

AGRICULTURAL MACHINERY	
0. 에너지원 (Power , Transportation)	
1. 트랙터, 경운기 (Tractor, Power tiller)	
2. 경운정지용 기계 (Tillage machinery)	
3. 이식, 파종, 시비 (Trainsplanter, Seeder, Fertilizer)	
4. 방제기, 제초기 (Chemical sprayer)	
5. 수확기 (Harvesting machinery)	
6. 조제가공기 (Processing machine)	
7. 시설물 (Farm facilities)	
8. 기타 (Others)	
9. EXIT	
선택 1	

표 4 - 1 은 처음에 데이터베이스를 實行시키면 나타나는 畫面이다. 여기서, 1. 트랙터, 경운기 (Tracter , Power tiller) 항목을 선택하면 표 4 - 2 의 화면이 윈도우 화면으로 나타난다. 그 화면에서 또 다시 11. 승용 트랙터(Riding tracter)를 선택하면 표 4 - 3 과 같은 入出力 畫面이 나타난다. 각 項目別의 번호를 選擇하면 원하는 作業을 遂行할 수 있다. 여기서, 1. 추가 항목을 선택하면 표 4 - 4 의 데이터 構造가 화면에 나타난다. 여기에서는 승용트랙터만을 살펴보았는데 모든 다른 項目들도 이러한 構造를 가지고 있다.

표 4 - 2. 1.트랙터, 경운기 항목을 선택했을 경우의 2차 분류 화면

1. 트랙터, 경운기 (Tractor, Power Tiller)
11. 승용트랙터 (Riding trator) 12. 경운기 (Power tiller) 13. Return to MAIN MENU
선택 11

표 4 - 3. 11. 승용트랙터를 선택했을 경우의 入出力 화면

승용트랙터 (Riding tractor)	
1. 추	가
2. 수	정
3. 삭	제
4. 조	회
5. 출	력
0. Return	
선 택 1	

표 4 - 4. 데이터 入出力을 위한 승용트랙터의 데이터베이스의 構造

N u m			
MACHINE			
MAKER	██████████	PTO RPM(2ND)	██████████
MODEL	██████████	PTO RPM(3RD)	██████████
TYPE	██████████	PTO RPM(4TH)	██████████
		PS	██████████
LENGTH	██████████	ENG RPM	██████████
WIDTH	██████████	CYLIND NUM	██████████
HEIGHT	██████████	TOTAL CC	██████████
WEIGHT(FW)	██████████	COM RATIO	██████████
WEIGHT(RW)	██████████	PS(MAX)	██████████
WHEEL(FW)	██████████	FUEL CONS	██████████
WHEEL(RW)	██████████	DRAW BAR PS	██████████
BASE	██████████	DRAW BAR F	██████████
GEAR SHAFT	██████████	F SPEED(MAX)	██████████
PTO RPM(1ST)	██████████	PRICE(won)	██████████

## 4.5.2 栽培作物과 作業體系 데이터베이스화

栽培作物에 따른 農作業活動을 使用農機械와 관련 細分하여 具體的으로 分類하였다. 먼저 作物별로 分類를 하고서 작물별로 作業 體系를 分類하였다. 그리고 각 作業별로 使用되는 農業機械를 分類하였다. 最終的으로는 각 세분된 作業별로 使用되는 農業機械가 선택이 된다. 선택된 농기계에 관련된 데이터는 위에서 개발한 농기계 데이터베이스에서 입력된 자료를 利用할 수 있도록 하는 인터페이스를 개발할 중에 있다.

실제 실행되어지는 데이터베이스의 構造는 다음 표와 같다. 먼저 표 4-5는 초기 화면을 나타낸다. 표 4-6은 수도작용기계를 선택했을 경우의 2차 분류 화면을 보여주고 있다. 표 4-7은 경운작업을 선택했을 경우의 3차 실행 화면을 나타내고있으며, 표 4-8은 승용트랙터를 선택했을 경우의 실행 화면이다.

표 4-9는 입력, 수정, 삭제를 선택했을 때의 실행 화면을 보여주고 있다. 새로운 情報의 追加, 削除 및 訂正이 이 화면에서 이루어 질 수 있다. 표 4-10에서 와 같이 조회 항목을 선택하면, 예를 들어 승용트랙터의 표 4-9에 기억된 여러 가지 제원 중에서 포장작업 효율 분석이나, 경지정리 전분가 시스템 등에 必要한 情報들 만을 별도로 쉽게 利用할 수 있도록 기억되어 있는 화면을 보여준다.

표 4 - 5. 재배작물과 작업체계 데이터베이스의 초기 화면

AGRICULTURAL PROCESS
1. 수도작용 기계 2. 전작용 기계  0. EXIT
————— 선택 1 —————

표 4 - 6. 1.수도작용 기계를 선택했을 경우의 2차 실행 화면

1. 수도작용 기계
11. 경운작업 12. 정지작업 13. 이앙작업 14. 퇴비 및 비료살포 16. 방제 및 제초작업 17. 수확작업  0. Return to MAIN MENU
————— 선택 11 —————

표 4 - 7. 11. 경운작업을 선택했을 경우의 실행 화면

11. 경운 작업	
111. 트랙터	(Riding tractor)
112. 경운기	(Power tiller)
113. 트랙터플라우	(Tracter plow)
114. 경운기플라우	(Power tiller plow)
115. Return	
선택 111	

표 4 - 8. 111.승용트랙터를 선택했을 경우의 실행 화면

승용트랙터 (Riding tractor)	
1. 입	력
2. 수	정
3. 삭	제
4. 조	회
5. 출	력
0. Return	
선택 1	

표 4 - 9. 데이터 入出力을 위한 승용트랙터의 데이터베이스의 構造

N u m			
MACHINE			
MAKER		PTO RPM(2ND)	
MODEL		PTO RPM(3RD)	
TYPE		PTO RPM(4TH)	
		PS	
LENGTH		ENG RPM	
WIDTH		CYLIND NUM	
HEIGHT		TOTAL CC	
WEIGHT(FW)		COM RATIO	
WEIGHT(RW)		PS(MAX)	
WHEEL(FW)		FUEL CONS	
WHEEL(RW)		DRAW BAR PS	
BASE		DRAW BAR F	
GEAR SHAFT		F SPEED(MAX)	
PTO RPM(1ST)		PRICE(won)	



표 4 - 10. 조회 항목을 선택했을 때의 실행 화면

승용트랙터 (Riding tractor)
4. 조회
1. 조건 없음 2. 제조회사 이름으로 3. 모델 이름으로  0. Return
————— 선택 0 —————

각 데이터베이스 간의 농업기계에 대한 데이터 互換性을 위해서 표 4 - 4와 표 4 - 9는 같은 構造로 되어 있다. 따라서, 데이터베이스 간의 데이터를 서로 互換할 수 있도록 만들어 졌다. 이와 같이 서로 다른 데이터베이스 간에 데이터를 자유롭게 互換할 수 있도록 데이터베이스를 一體化시키는 作業은 각각의 데이터베이스가 모두 完成이 된 이후에 遂行될 豫定이다.

### 4.5.3 圃場條件 및 土壤條件 데이터베이스化

多様な 農作業을 수행하기 위해서는 각 作業에 부여되는 농작업기계에 대한 가장 효율적인 경지의 형상과 크기를 결정해야 한다. 또한, 경지의 형상과 규

모를 결정하기 위해서는 각 농작업별로 必要한 농기계에 대한 圃場作業效率에 관한 究明이 必要하다. 圃場作業效率에 관한 分析에서는 圃場の 장변과 단변과 같은 圃場の 크기 및 토양 수분 함유율 등이 主要 變數로 작용한다. 물론, 作業幅과 作業速度와 같은 기계의 변수들은 기계가 선택되면 자동적으로 考慮된다. 여기서, 농작업기계의 변수들은 농기계 데이터베이스를 利用하게 된다.

우선, 圃場조건 및 토양조건에대한 데이터베이스를 具現하기 위해서는 각 主要 農作業別 및 使用農機械別의 圃場作業效率을 提示할 수 있는 模型을 만들어야 한다. 또한 그 모형으로서 분석한 결과가 명확히 提示되어야 한다. 현재 콤바인을 利用한 收穫作業에 대해서는 이러한 작업들이 完了가 되었으며 耕耘, 移秧, 防除作業에 관한 研究는 현재 進行중에 있다. 특히, 이와 같은 농작업들은 適期姓을 要하기 때문에 研究에 相當한 어려움을 가져왔다. 이러한 각 작업별의 圃場作業效率에 관한 研究分析이 끝난 후에 실제의 데이터베이스를 完成할 豫定이다.

## 第 5 章 映像處理를 통한 地形情報 獲得

### 5.1 地圖의 映像獲得

#### 5.1.1 映像處理

디지털 映像處理(digital image processing)란 人間이 視覺情報에 대하여 행하는 모든 처리를 기계 또는 컴퓨터로 실행하는 것으로 인간의 視覺技能을 대행하거나 확대하는 것을 그 궁극적인 目的으로 한다. 映像處理技術은 인간의 시각 정보에 대한 어떤 처리나 작업의 客觀性 및 再演性을 높이며, 인간으로 하여금 보다 높은 수준의 업무에 종사할 수 있도록 한다.

映像處理技術은 일찌기 遠隔探査分野에서 사용되었으며 현재에도 활발히 이용되고 있다. 원격탐사기술은 1970년대 이전에는 위성이나 항공기에 의해 하드웨어적으로 보다 명료한 映像 또는 寫眞을 얻을 수 있는 기술에 치중하였으며 컴퓨터의 발달과 함께 이와같은 영상 또는 사진에 포함되어 있는 정보를 定量化시키는 기술, 즉 映像處理기술과 함께 보다 유용한 기술로 발전하고 있다.

항공기나 위성에 탑재되어 있는 센서에 의하여 획득된 각종 정보는 사진영상이나 마그네틱 테이프에 수치로서 저장된다. 地表의 情報를 표현하는데 전통적으로 사용하여 온 점, 선, 면이 아니라 2次元 平面的 畫素(pixel element, "pixel")를 사용하고 각 畫素에는 입사광의 파장별 複寫強度를 나타내는 수치가 부여된다. 數値化된 화소데이터로부터 원하는 지표에 대한 정보를 얻기 위해서는 적절한 영상처리기술이 요구된다.

그런데, 이러한 기술을 이용하여 지표의 특성을 파악하기 위해서는 지상에서 실제로 획득하여 분석한 결과와 相互 結合이 충분히 가능하여야 한다. 本 研究는 지형으로부터 직접 원하는 映像데이터를 획득하는 것이 아니고 이미 어떤 형태로 표현된 지형에 대한 자료(지형도)로부터 지리적 영상을 얻어 필요한 지형정보를 컴퓨터 영상처리를 이용해 손쉽게 얻고자 한다.

### 5.1.2 디지털 映像

영상을 컴퓨터로 처리하기 위해서는 카메라로 얻어진 아날로그 영상정보를 sampling 및 quantizing에 의하여 디지털(digital) 영상정보로 변환하여야 한다. 本 研究에서 사용한 映像處理장치의 해상도는 512 x 480이며 각 화소는 8비트 (0 - 255)의 값을 가진다. 0 - 255는 畫素值를 나타내며, 큰 수일수록 밝은 映像을 나타낸다.

### 5.1.3 映像處理 시스템構成

地理情報시스템(GIS)은 컴퓨터 하드웨어, 소프트웨어 모듈 및 적절한 시스템 디자인이라는 세가지 主要 要素로 이루어진다. 일반적인 지리정보시스템의 컴퓨터 하드웨어는 컴퓨터 CPU, 터미널 및 자료 저장 장치, 디지털타이저 (digitizer), 도화기(plotter) 등으로 이루어지는데 아직은 그 가격들이 매우 비싸다.

지도의 情報를 數值化하는 機器에는 레스터 스캐너, CCD 디바이스, 벡터 스

캐너 등이 있으나 저렴한 가격으로 寫眞에서 判讀된 境界線을 捕捉하거나 航空 寫眞을 래스터화 하기 위한 간단한 기기들이 開發되고 있다. 이러한 기기중 하나로서 영상 포착기(frame grabber)와 비디오 카메라를 연결시킨 비디오 디지털 이저가 있다.

映像情報를 處理하는 技術은 컴퓨터의 도움없이 불가능하다. 왜냐하면, 2차원이든 3차원이든 처리될 정보량이 엄청나게 많기 때문이다. 일반적으로 영상 정보는 컴퓨터 그래픽이나 CAD에 의해서도 다루어지지만, 지리정보는 정보량이 상대적으로 많기 때문에 컴퓨터 그래픽이나 CAD와 병행해서 사용할 때 처리속도 문제가 있으나, 地理映像畫面위에 耕地整理前後의 전체윤곽을 함께 그래픽으로 圖示하는 研究는 앞으로 계속되어야 할 것이다.

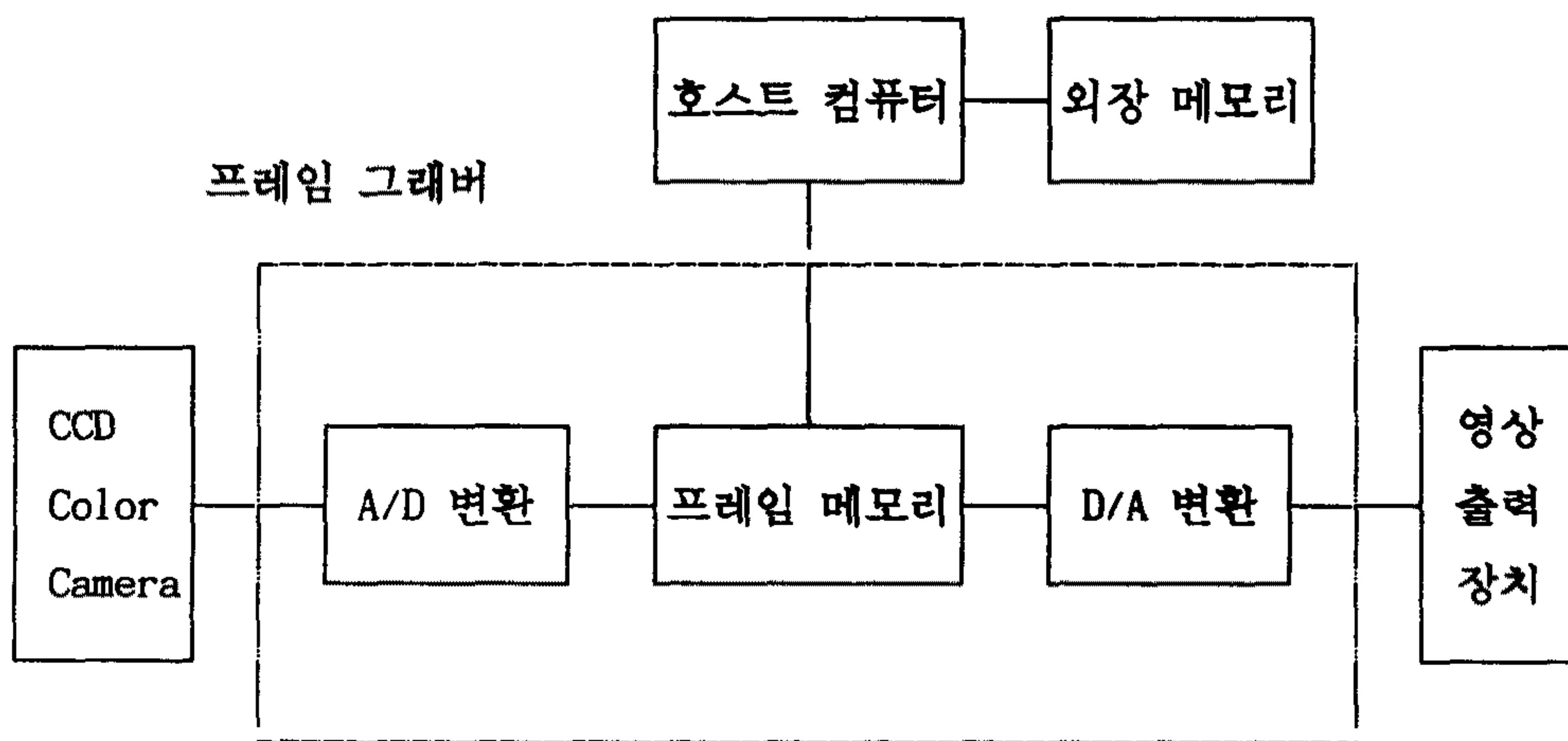


그림 5 - 1. 映像處理시스템의 構成要素

그림 (5-1)은 본 研究에서 地圖 映像을 獲得, 處理 및 貯藏을 위한 映像處理 시스템에 대한 概略圖이다.



## 5.1.4 測度 設定(Calibration)

비디오 디지털라이저를 이용할 경우, 지형도와 디지털 영상 데이터간의 對應關係가 정밀하게 設定되어야 한다. 이들간의 대응관계는 지형도와 카메라간의 거리, 비디오 디지털라이저의 縱橫比(aspect ratio) 및 地形圖의 蓄積度 등을 고려하여 設定된다. 本 研究에서는 모눈종이에 대한 영상을 얻은 후 모눈의 단위크기에 해당하는 영상의 화소의 크기를 分析하여 測度를 設定하였다.

## 5.1.5 異値化

화소치는 곧 지형도에 나타난 각종 선이나 점의 명도를 나타내는데, 배경에 해당하는 화소는 높은 값, 점이나 선에 해당하는 화소는 낮은 값을 가진다. 원래 지형도는 두가지의 값(배경 및 선, 점)으로 표현되므로 어떤 기준에 의하여 최대 256가지의 화소치를 2가지의 화소치로 변환하는 과정이 요구된다. 즉, 문턱값(thresholding value)을 중심으로 이보다 큰 화소치는 255, 작거나 같으면 0으로 변환한다. 이러한 과정을 映像의 異値化라고 말한다. 일반적으로 本 研究에서 이용한 지형도 영상과 같은 도면영상은 영상 異値化를 수행함으로써 훨씬 선명한 영상을 얻을수 있다.

本 研究에서는 研究者가 키보드의 방향키의 조작에 의하여 문턱값을 임의로 변화시킴과 동시에 映像出力 모니터를 보면서 영상異値化를 수행하였으며, 지형도의 각종 선과 점의 鮮明함을 기준으로 최적의 異値映像을 決定하였다.



## 5.1.6 細線化(thinning)

異値化된 映像에 나타난 각 필지의 경계를 나타내는 선은 일정한 굵기를 가지지 않았다. 이는 트래싱하는 과정에서 선의 굵기를 일정하게 할 수 없을 뿐만 아니라 조명 등이 영향을 미쳤기 때문이다. 따라서, 本 研究에서는 각 경계선 등의 연결상태(즉, 폐속선 형태 등)를 바꾸지 않으면서 모든 선과 점의 굵기에 해당하는 화소의 수가 1개가 되도록 하는 영상의 細線化를 다음과 같이 수행하였다.

이치영상(경계선:“255”, 배경:“0”)을 대상으로 세선화를 실시한다. 주어진 처리화소 P를 중심으로 팔방향의 이웃한 화소들을 3시방향의 화소, P<sub>0</sub>로부터 반시계방향으로 돌면서 P<sub>2</sub>, P<sub>3</sub>, P<sub>4</sub>, P<sub>5</sub>, P<sub>6</sub>, P<sub>7</sub> 이라고 하자. 그리고, T(P)를 P<sub>0</sub>, P<sub>1</sub>, . . . , P<sub>7</sub>, P<sub>0</sub>의 순서에서 화소치가 “0”에서 “255”로 변하는 횟수, N(P)를 P와 8근방에 있는 “255”를 가진 화소수라고 하자. 그러면, 처리화소 P를 전 영상 화소로 이동하면서 다음 조건을 모두 만족시키는 P를 “0”으로 하면 세선화 영상을 얻을 수 있다.

- ①  $2 \leq N(P) \leq 6$ ,
- ②  $T(P) = 1$ ,
- ③  $P_0 * P_2 * P_4 = 0$  이거나  $T(P_2) \neq 1$ ,
- ④  $P_0 * P_2 * P_4 = 0$  이거나  $T(P_4) \neq 1$



## 5.2 地形情報 獲得

### 5.2.1 境界線地形圖 製作 및 入力

1/5000 지형도로부터 그림 5-2와 같은 조명장치와 CCD 카메라를 이용하여 대상지역의 지도 映像을 入力하였다. 그림 5-3은 입력된 지도 영상을 모니터에 보여 주고 있다. 이 지형도에서 登高線은 2.5m간격으로 표시되어 있으며, 각 필지의 境界는 斷續線(dashed line)으로, 田畝의 경계는 점선으로 표시되어 있다. 지명, 행정구역 경계 및 기타 색인 부호들은 알고리즘에 의해서 필지 형상, 면적, 고저 등의 지형정보를 신속히 얻는 데 장애 요소로 등장하였다.

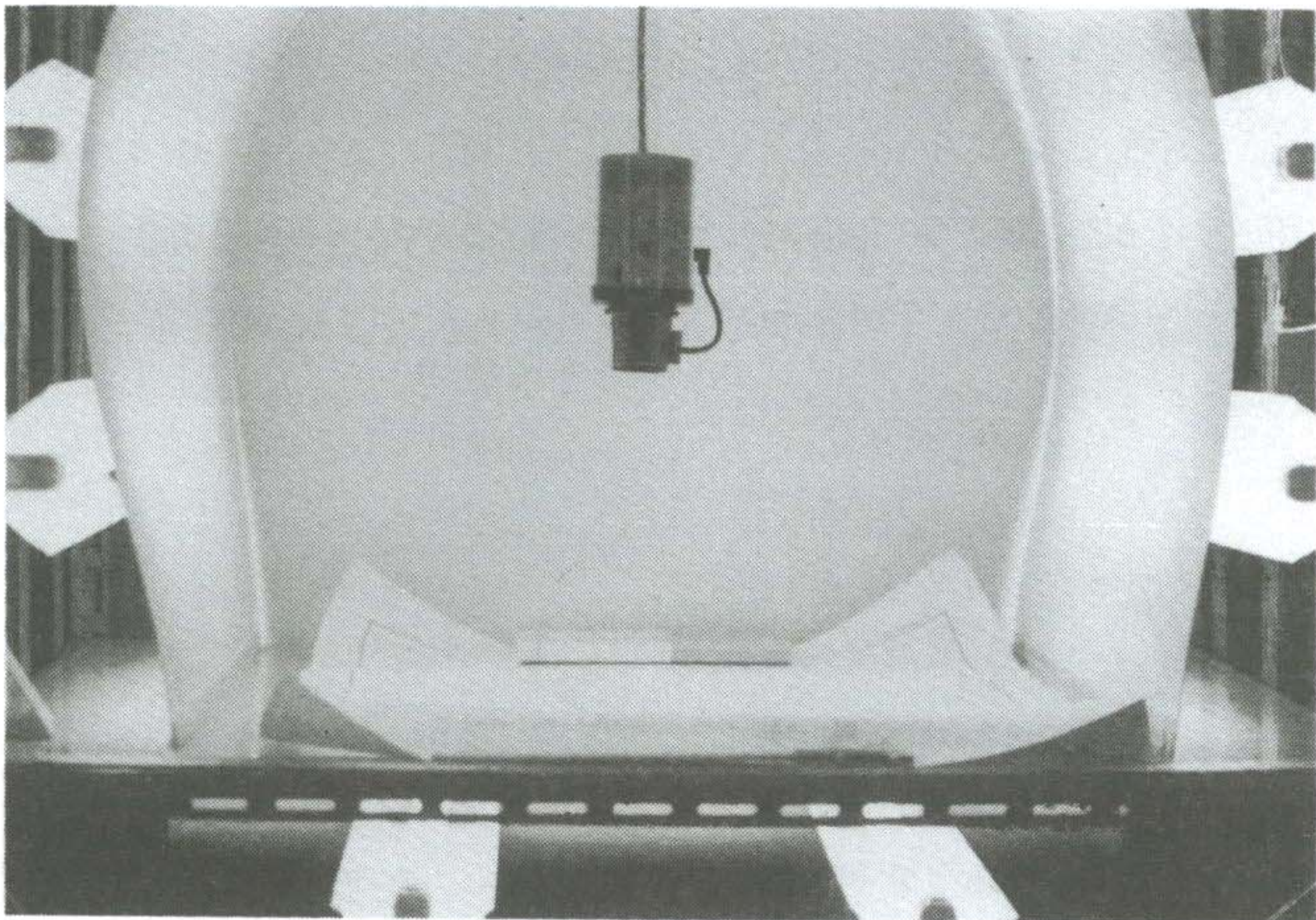


그림 5 - 2. 조명장치와 CCD카메라로 구성된 영상 입력장치



따라서, 本 研究에서는 1/5000 地圖를 이용하여 對象地域을 2배로 擴大 複寫하여 1/2500지도로 만든 다음, 반투명 트레이싱 종이위에 관심있는 未耕地整理 全體地域, 田畝형상, 登高線, 高度點 등을 트레이싱하였다. 그리고, 형광등을 이용한 투과조명에 의하여 트레이싱된 지형도를 CCD 카메라와 프레임 그래버 (Frame grabber)를 통해 디지털 영상으로 변환, 저장하였다.

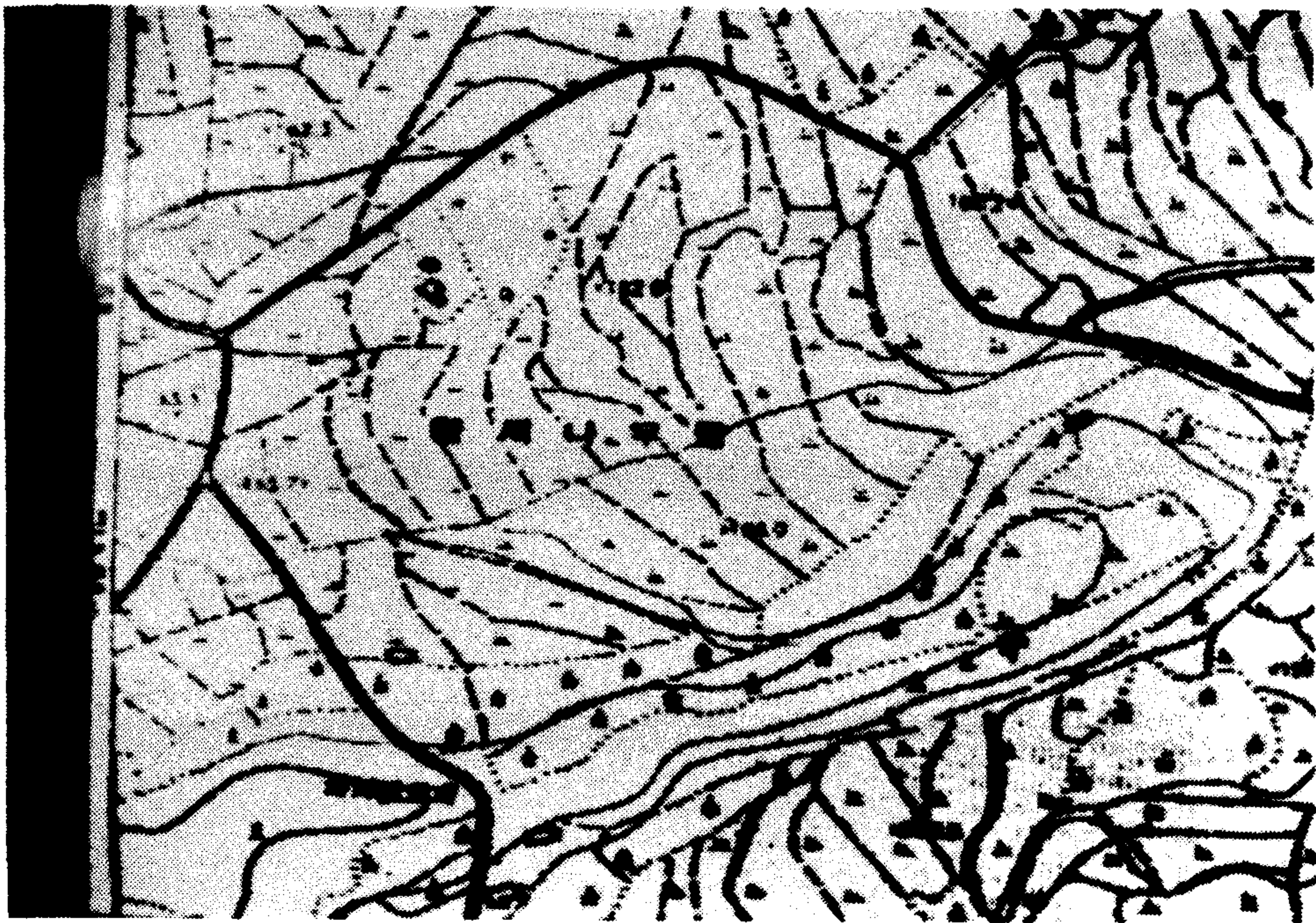


그림 5 - 3. 1/5000 지형도로부터 모니터에 入力된 地圖映像



## 5.2.2 獲得可能 地形情報

저장된 영상 데이터로부터 適當한 문턱값을 設定하여 二值畫像(binary image)를 얻거나 그림 5-4와 같이 윤곽추출(edge detection) 등의 映像處理기법을 이용함으로써 각 필지의 경계를 보다 선명하게 할 수 있을 것이다. 本 研究에서는 이미 설명하였듯이, 단속선, 점선, 문자 및 기호 등은 映像處理에 障礙가 되기 쉬우므로 미리 이들을 없애고 실선화된 지형도를 製作하였다.

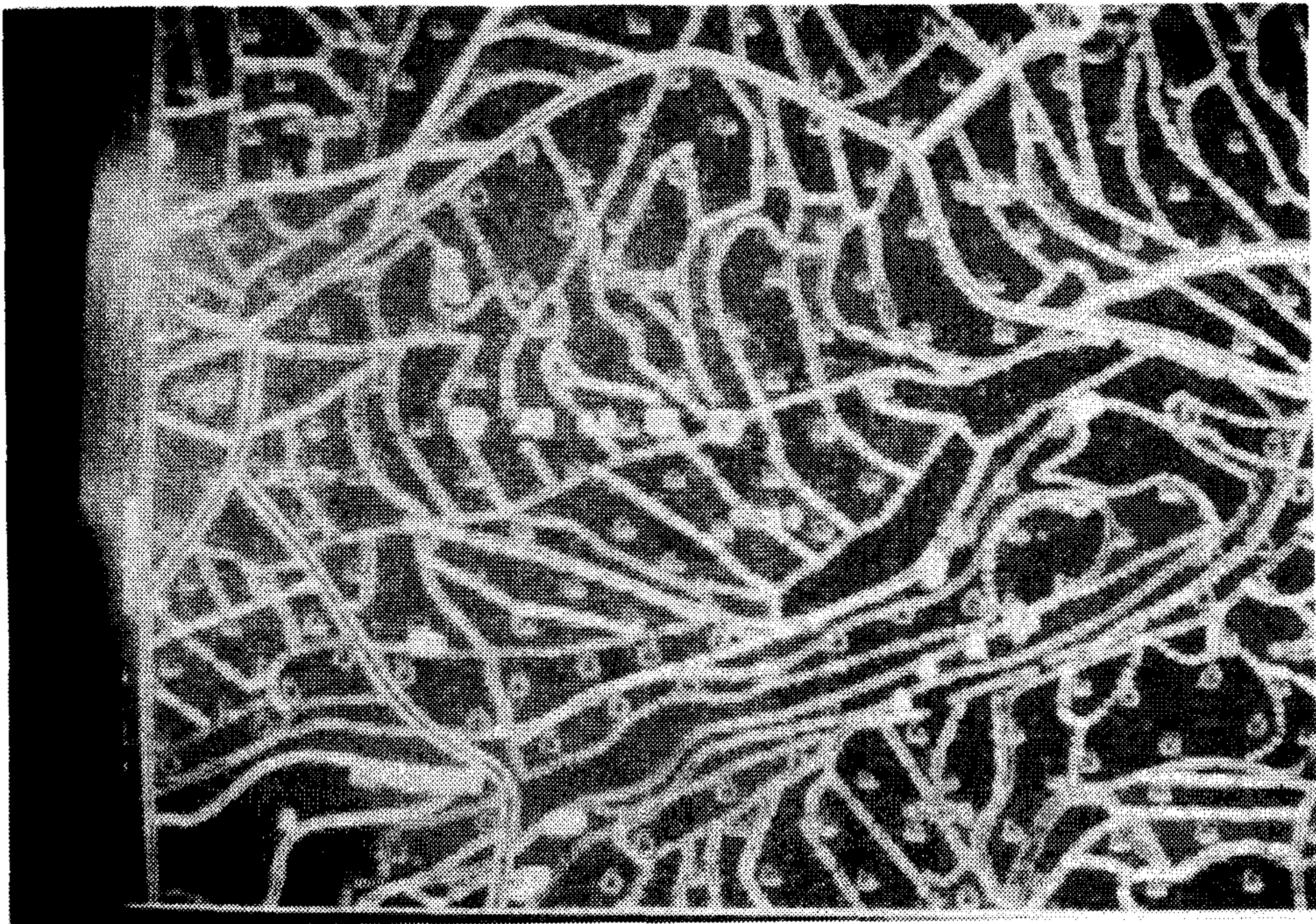


그림 5 - 4. 윤곽추출(edge detection) 映像技法으로 處理된 地圖映像



그림 5-5는 실선화된 지형도로부터 얻어진 영상처리를 하기 전의 초기영상을 보여 주고 있다. 저장된 영상으로부터 다음과 같은 지형정보를 획득할 수 있는 영상처리 소프트웨어를開發하였다.

- ① 미경지정리 전지역의 면적,
- ② 각필지별 면적 계산 및 평균 면적,
- ③ 전답별 평균고도.

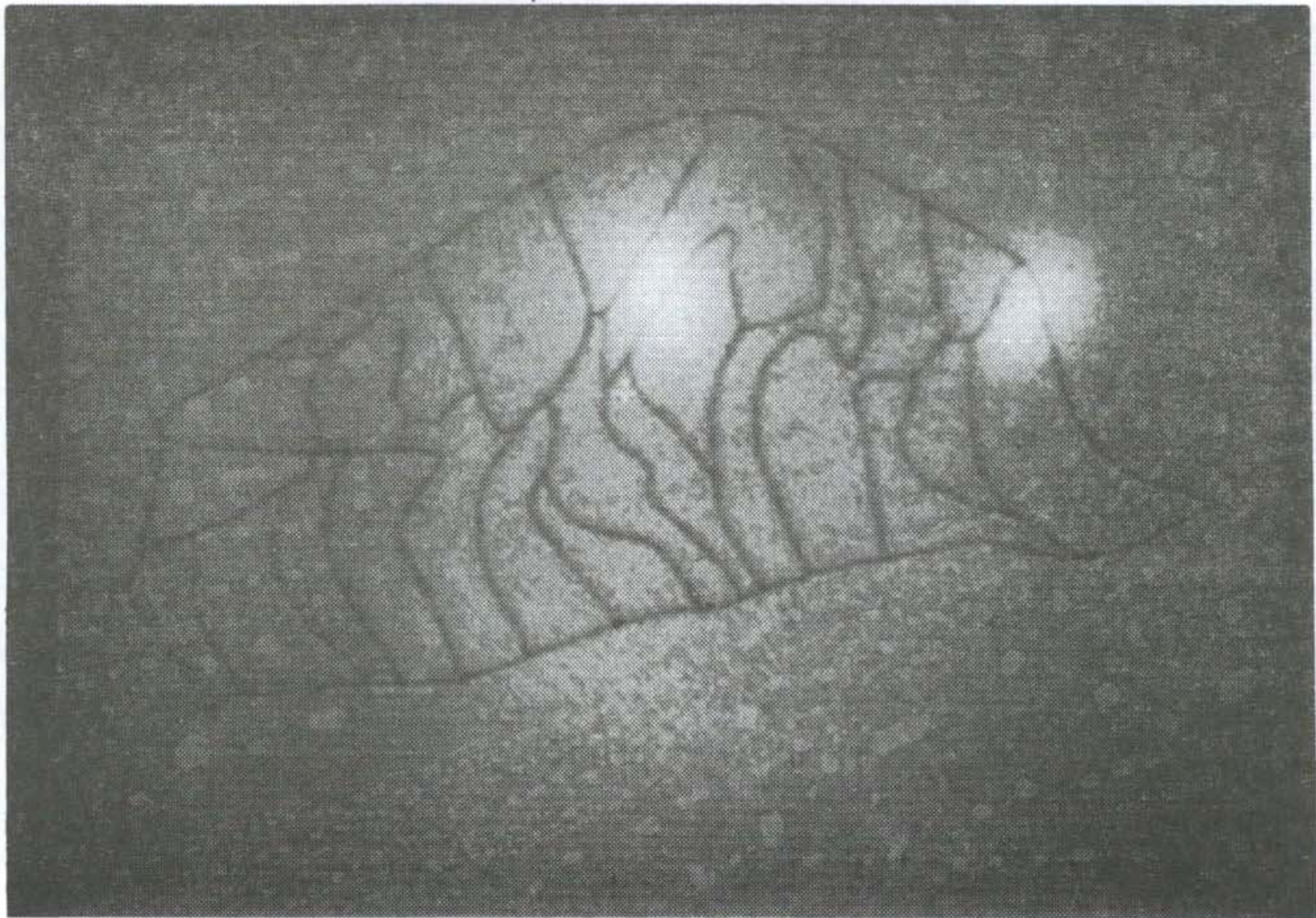


그림 5 - 5. 實線化된 地形圖에서 얻어진 初期映像



다음 그림 5-6은 上記의 項目을 分析하기 위한 處理過程을 도표로 나타낸 것이다.

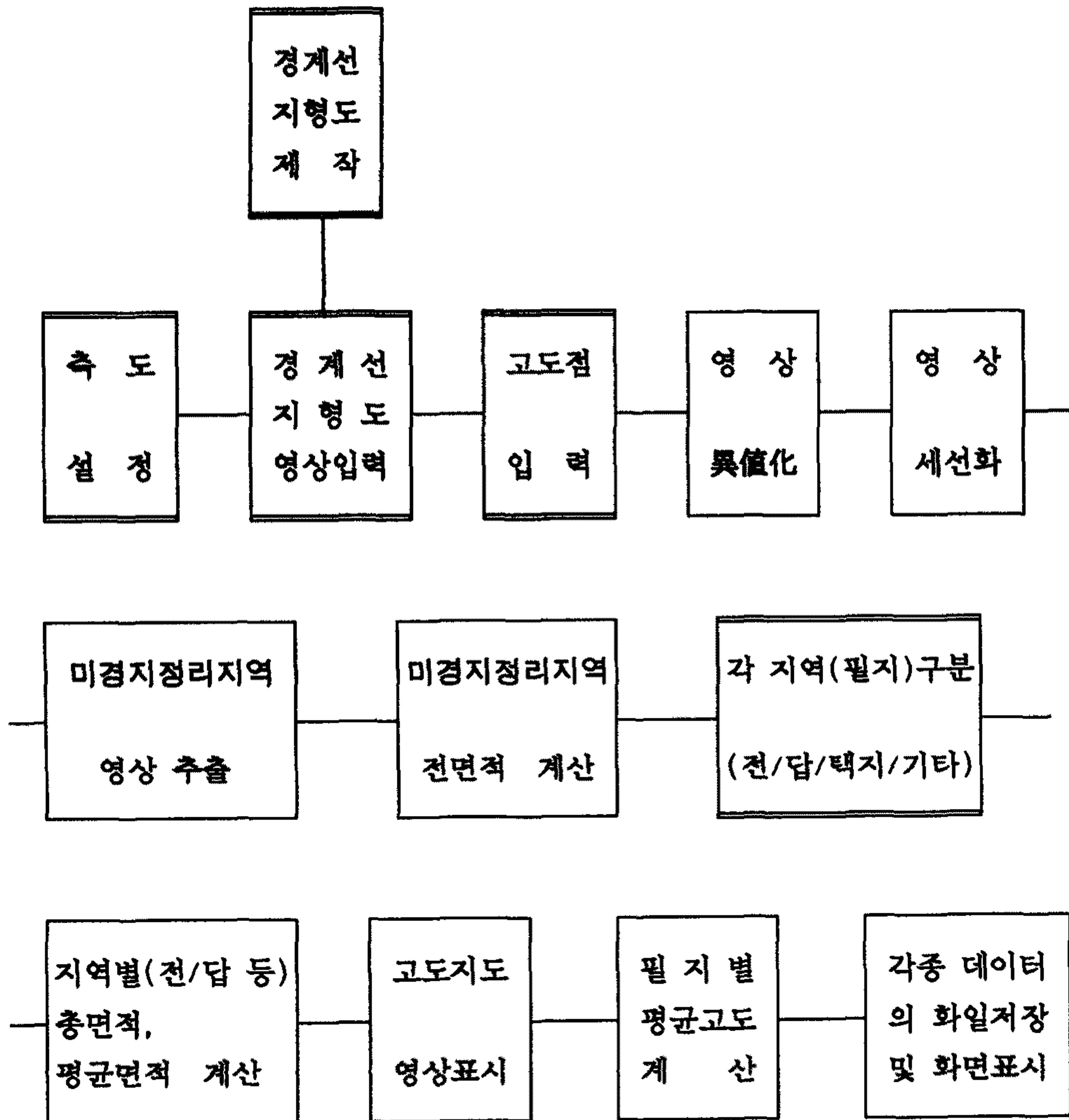


그림 5 - 6. 地形情報 獲得 소프트웨어의 處理過程



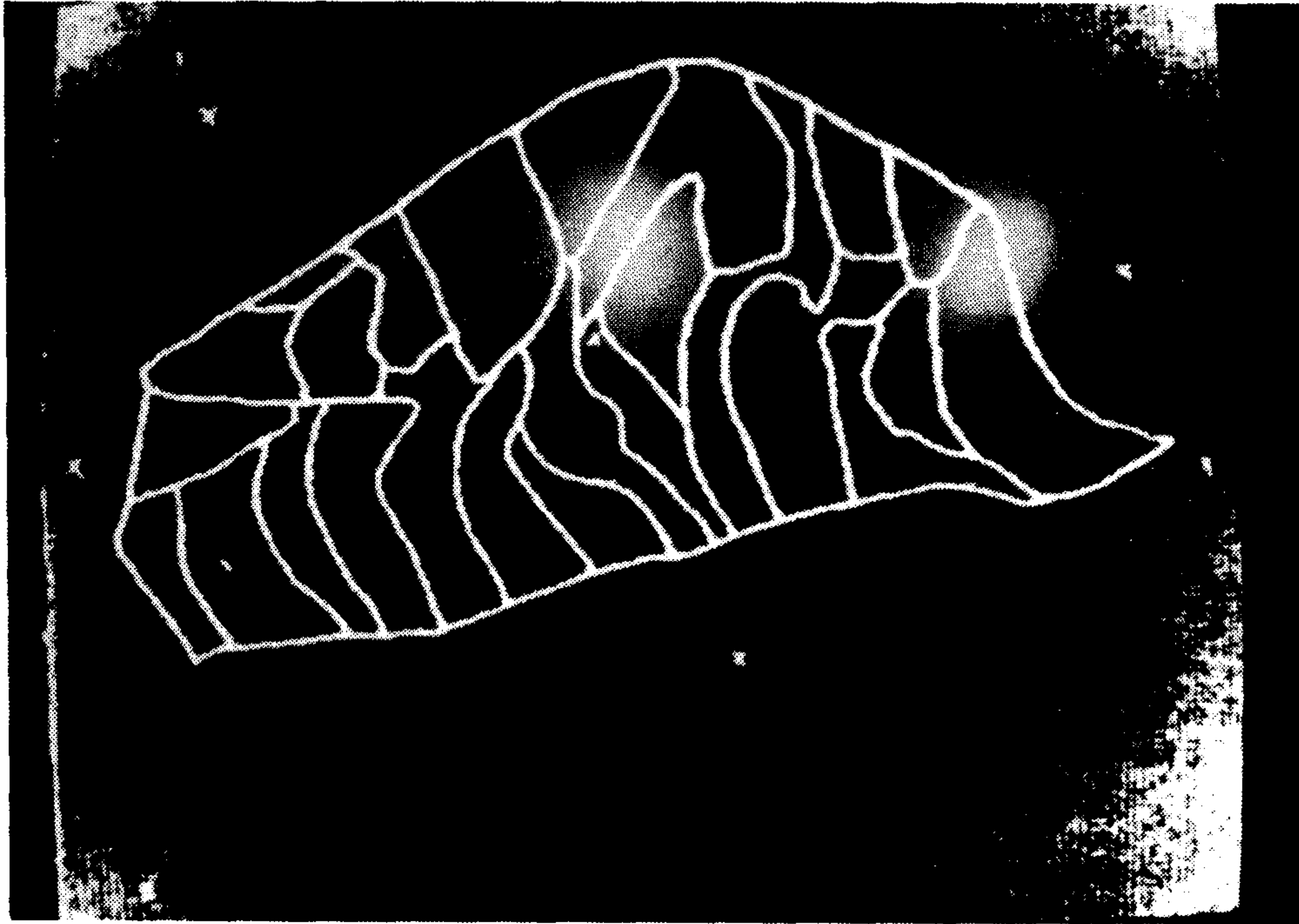


그림 5 - 7. 문턱값을 設定하여 이치화된 映像

그림 5-7은 문턱값을 설정하여 각 필지경계선 화소치를 255로 하고 배경을 화소치 0으로 異値化된 영상을 나타내고 있다.

### 5.2.3 高度粘 入力

각 필지의 平均高度를 구하기 위해서는 기준이 되는 고도점의 위치에 대한 정의가 있어야 한다. 本 研究에서는 각 고도점을 반시계방향으로 순차적으로 연결하여 다각형을 형성하였을 때, 未耕地整理地域이 완전히 포함될 수 있도록 고도점을 미리 정해진 순서에 의하여 시계반대방향으로 입력한다. 여기서, 고도점 입력 순서를 미리 정하는 이유는 각 고도점의 입력 순서가 각 고도점에 대한 고



도값을 할당하기 위한 입력 데이터 화일에서의 高度貯藏順序에 부합되도록 하기 위함이다.

#### 5.2.4 對象地域만의 映像 抽出 및 細線化

일단 미경지정리지역의 경계선이 입력되면 그림 5-8과 같이 저장 영상에서 각 필지의 경계선을 세선화한 다음, 映像處理 대상인 관심 있는 전체 경지정리 대상 지역(가장 큰 폐속선 영역)의 内部만을 남기고 불필요한 영역의 영상 데이터를 消去함으로써 대상지역만의 映像을 抽出한다. 이렇게 함으로써 처리정보량을 줄일 수 있으며, 또한 남겨진 화면의 여백에는 계산된 지형정보에 대한 결과를 표시 할 수 있게 한다.

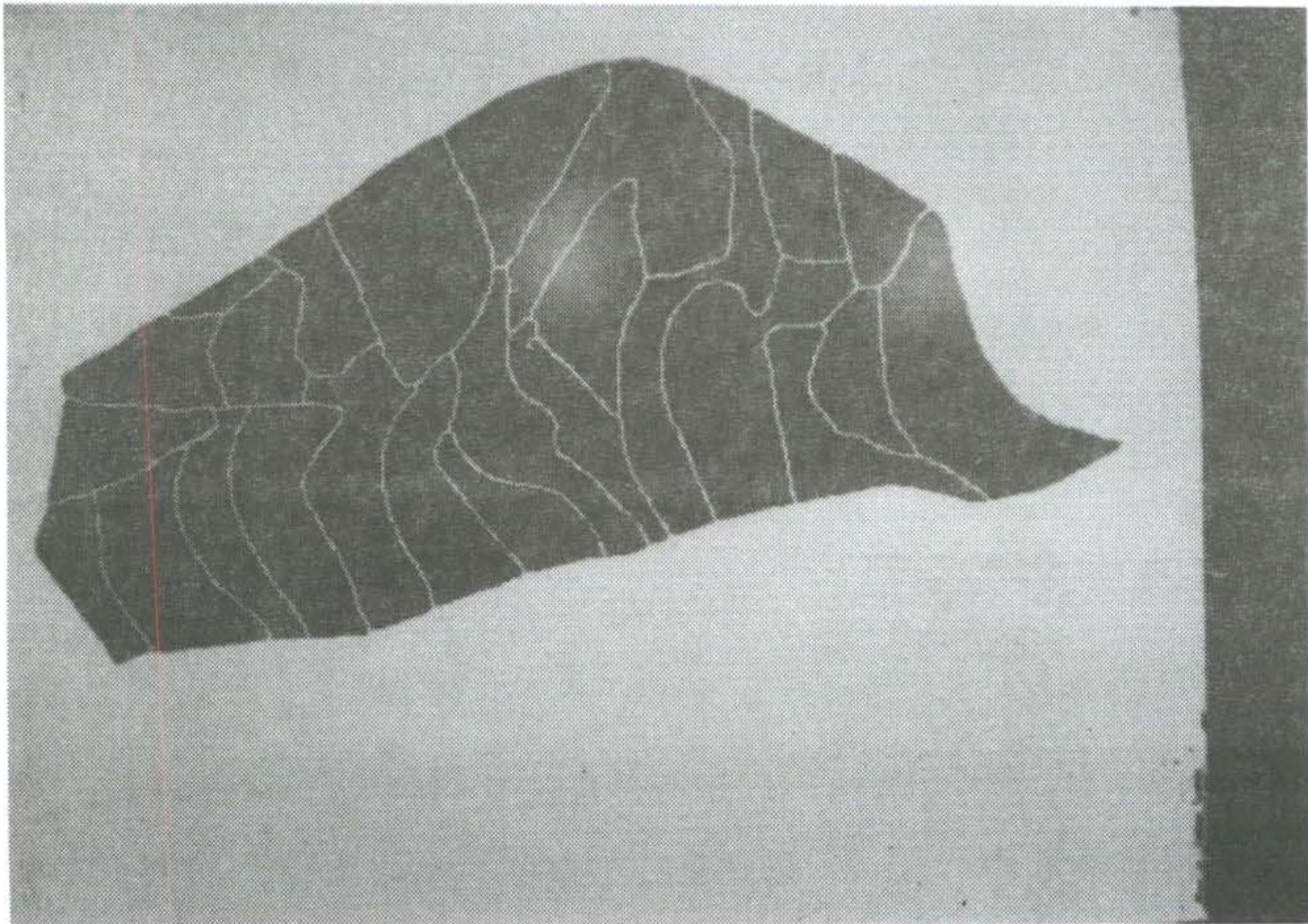


그림 5 - 8. 映像處理를 통한 耕地整理對象地域만의 映像抽出



## 5.2.5 各 地域(필지)의 區分 및 面積 計算

세션화된 영상에는 각 필지의 경계선이 뚜렷하게 나타나 있다. 각 지역마다 하나의 폐속선이 形成되는데 폐속선 내부의 임의 좌표에 지형도에 나타난 전/답/과수원/주거지/기타지역 중의 하나를 마우스를 이용하여 선택하면 그림 5-9와 같이 자동적으로 그 영역에 대한 고유번호가 부여됨과 동시에 면적이 계산된다. 또한, 전/답/과수원/주거지/기타지역 등의 지역별 필지의 수, 총면적 및 필지의 평균면적이 계산된다.



그림 5 - 9. 境界化된 각 필지별 自動的인 面積計算過程

面積은 本 研究에서 開發한 方法으로 計算된다. 즉, 측도설정에 의하여 하나의 화소가 차지하는 實際面積을 비례적으로 계산해서 알 수 있다. 따라서,



選擇된 領域에 대한 그 固有番號가 자동적, 순차적으로 부여되면 고유번호에 해당하는 화소치 (Pixel Value, "Gray Value")로 영역 내부의 모든 화소의 화소치를 대치한 후, 해당 필지의 화소치의 빈도수를 Count하여 얻은 總頻度數를 얻는다. 총빈도수에 측도설정으로 얻어진 화소 하나당 계산된 면적치를 곱해서 필지면적을 구한다.

## 5.2.6 高度地圖表示 및 匹地別 平均高度 計算

임의의 未耕地整理地域에 해당하는 화소는 이웃하는 세개의 알려진 고도점들에 의하여 형성되는 여러가지의 삼각형들 중의 한 삼각형 内部에 속하게 된다.

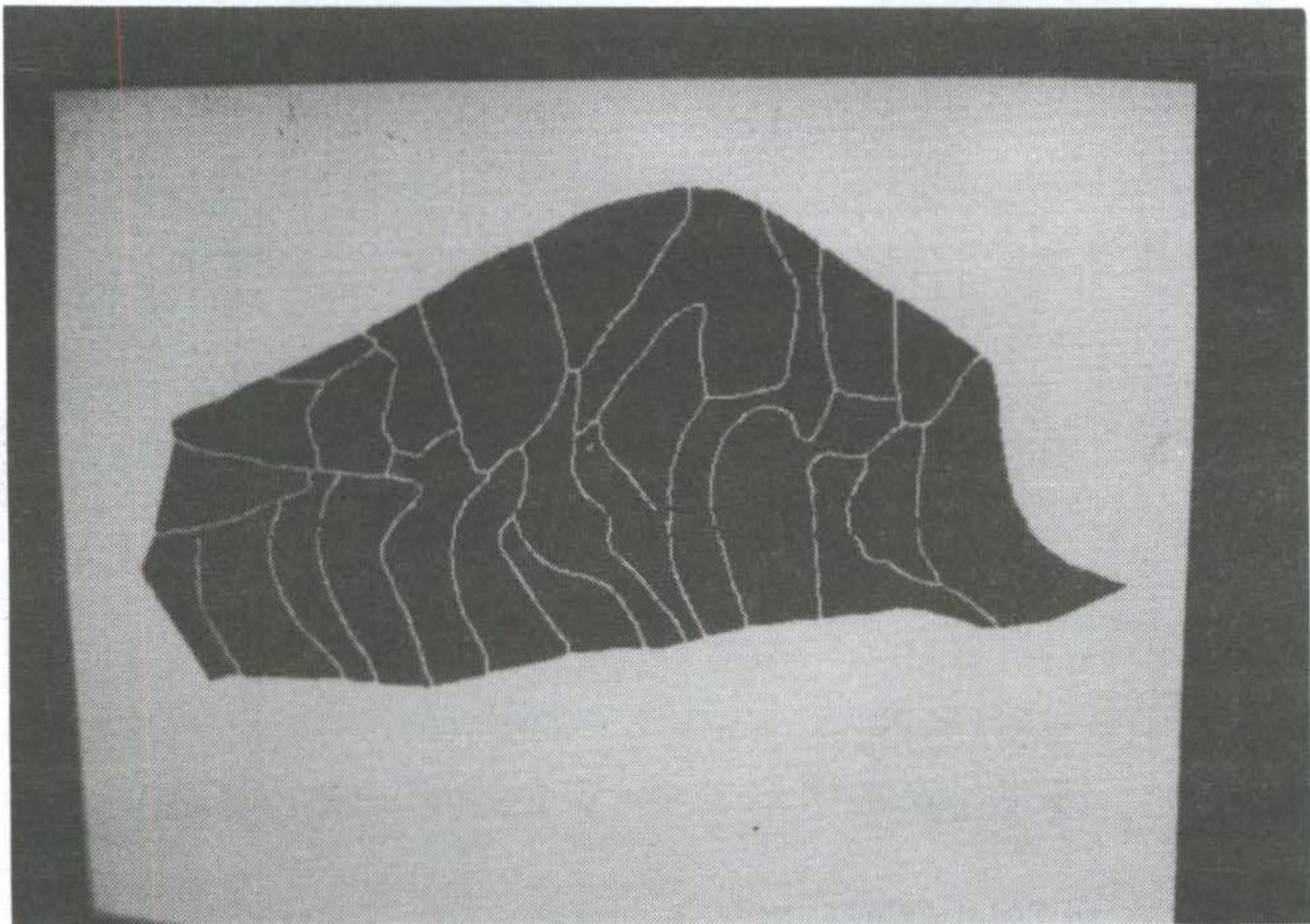


그림 5 - 10. 10번째 화소마다 수행된 高度計算



그리고, 이미 각종 고도점의 高度를 알고 있으므로 주어진 삼각형의 세 꼭지점의 고도를 이용하여 주어진 화소의 고도를 線形補間法을 이용하여 계산할 수 있다.

각 필지의 平均高度를 구하기 위해서는 TV 注射方向으로 화소를 探索하면서 현재의 화소가 어느 삼각형 내부에 속하는지를 결정한 다음, 그 삼각형을 구성하는 세 꼭지점의 고도를 이용하여 현재화소의 高度를 계산한다. 그리고, 각 필지에는 이미 일정한 고유화소치가 割當되어 있으므로 현재화소의 화소치를 읽음으로써 어느 필지에 속하는지를 알 수 있다. 그림 5-10은 매10번째 화소마다 고도를 계산한 화면을 보여주고 있다.

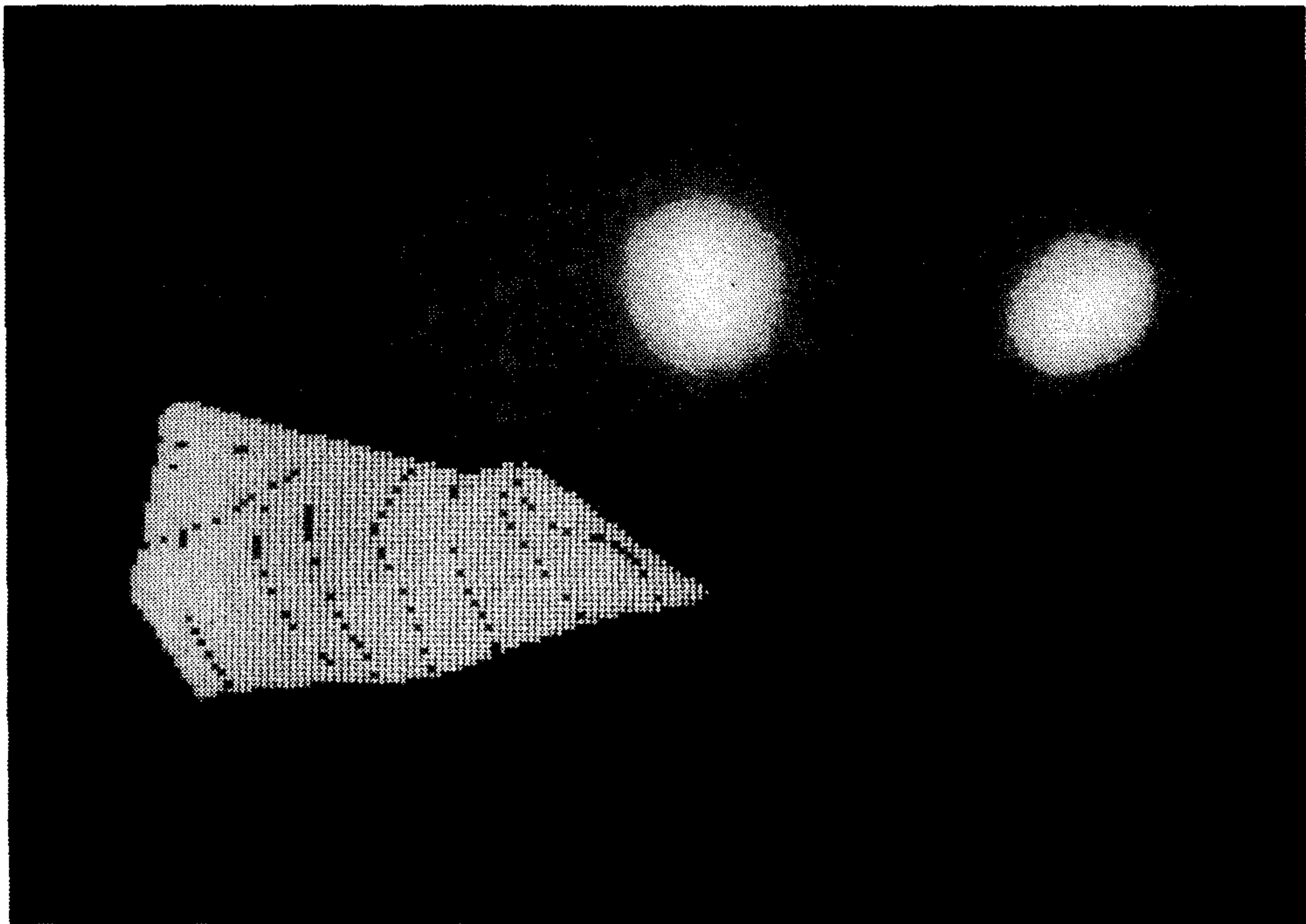


그림 5 - 11. 3번째 화소마다 계산된 고도지도에서 고도가 높은 지역



모든 화소에 대한 고도가 계산되면 계산된 고도를 0~255의 화소값으로 비례 변환하면 映像出力 모니터에는 高度의 分布를 나타내는 映像이 出力되는데, 本 研究에서는 이것을 高度地圖라고 命名하였다. 그림 5-11은 3번째 화소마다 계산된 고도지도에서 고도가 높은 지역을 보여준다. 즉, 높은 화소치를 갖게 되어 밝게 나타난다. 키보드의 방향키를 이용하여 높은 화소치부터 즉, 높은 고도 지역부터 화면에 나타내게 하였다.

그림 5-12는 중간 고도의 지역을 포함한 것이며, 그림 5-13은 가장 낮은 고도까지를 포함한 전체 고도지도를 보여 주고 있다. 따라서, 이 고도지도를 이용하여 耕地定理 對象地域内の 어느 방향으로도 傾斜度를 쉽게 알아 낼 수 있도록 한다.

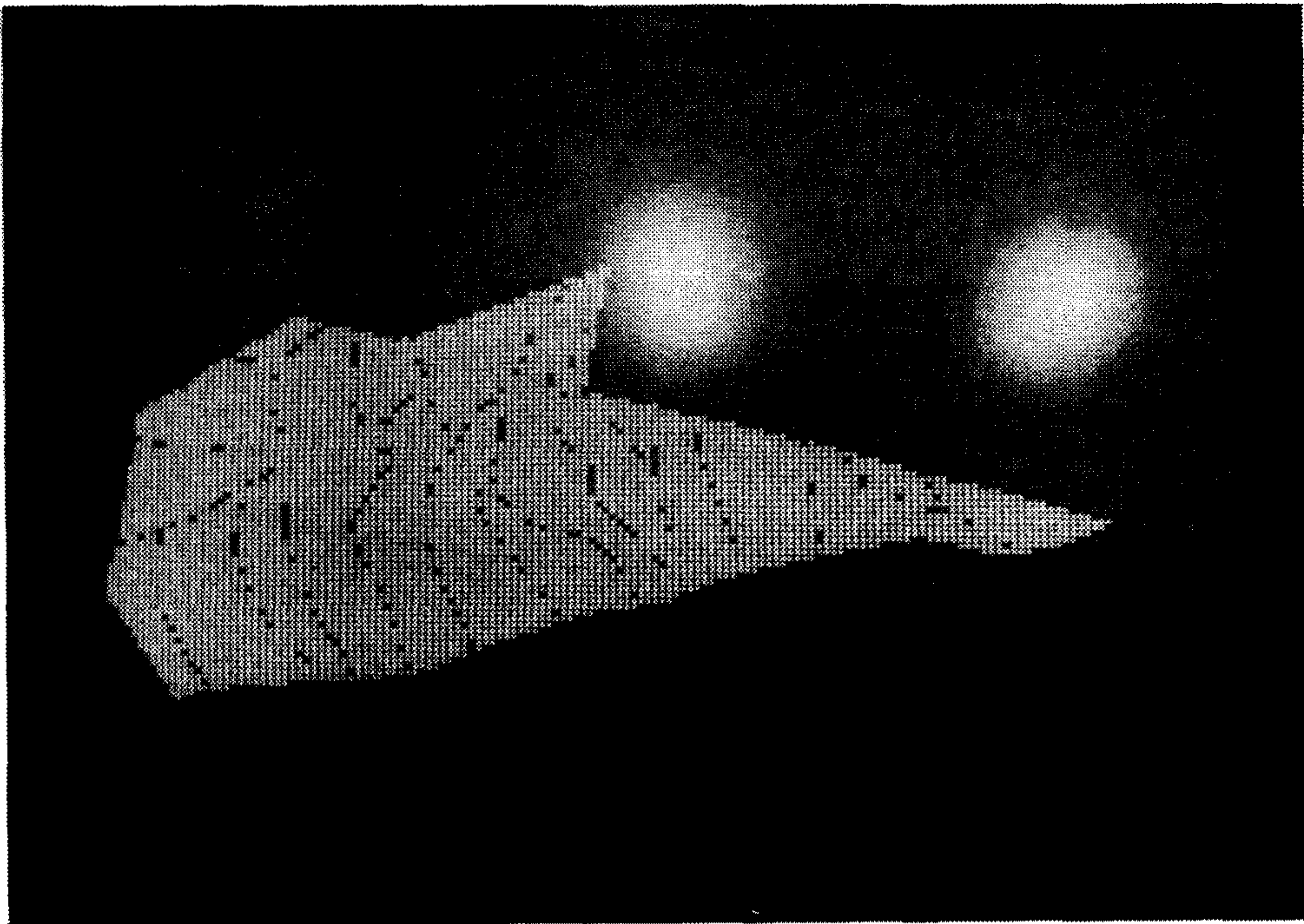


그림 5 - 12. 中間高度 地域을 포함한 高度地圖畫面

한편, 같은 필지에 속하는 화소마다 계산된 고도의 총합을 해당 필지에서 탐색을 수행한 화소의 총갯수로 나눔으로써 해당 필지의 平均高度를 계산할 수 있다.

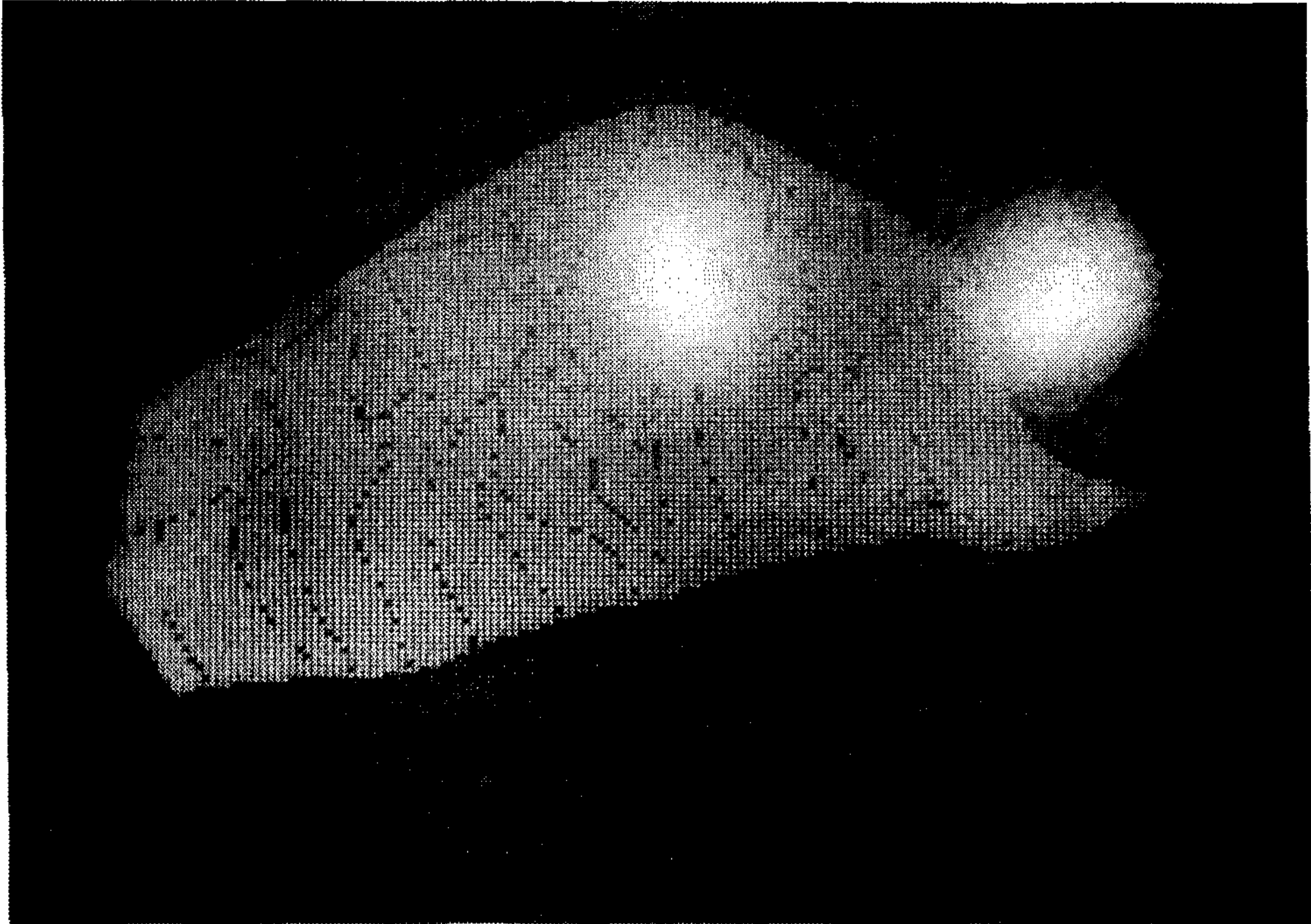


그림 5 - 13. 가장 낮은 지역까지 포함한 全體高度地圖

## 5.2.7 데이터 貯藏 및 出力

各 領域의 面積, 地域區分 및 固有番號, 該當地域의 총 필지수, 총면적 및 평균 면적 등은 背景 映像(비 대상지역 영상)에 出力할 수 있을 뿐만 아니라 데이터 화일로 저장된다. 本 研究에서는 더 나아가 入力된 地形圖 영상 및 作業進行現況 등을 貯藏하도록 함으로써 작업을 일단 중단한 후에 다시 계속하고자 하거나 각종 檢索作業이 可能하도록 하는 소프트웨어 모듈을 開發하였다.

여 백

## 第 6 章 結言 및 2次年度 研究方向

作業別·耕地規模別 作業效率分析에 있어 農作業의 特殊性인 作業의 適期性 때문에 모든 農作業體系에 대하여 作業效率을 研究·分析하는 것은 無理가 있었으며 콤바인작업에 대한 作業效率과 作業能率에 대한 研究를 제외한, 耕耘, 移秧, 防除作業과 같은 未研究作業에 대해서는 2次 年度에 계속 遂行할 豫定이다. 이런 작업들의 圃場實驗과 分析이 完成되면 그 結果는 耕地의 規模와 形態를 規定하는 데 綜合적으로 利用하게 될 것이다.

作業可能日數率의 決定에 관한 연구는 우선, 수확작업에 대해서 분석을 하였으며 경운작업과 이앙작업 및 방제작업 등과 같은 未研究作業은 현재 작업가능日數率결정모형개발에 관한 연구가 진행중에 있다.

농작업계획수립에 필요한 데이터베이스화에 관한 연구는 研究課題의 特性上 短期間에 完成한다는 것은 매우 힘든 일이었으며 따라서, 일차적으로 농업기계 및 시설과 장비, 재배작물, 작업체계 등에 관해서 데이터베이스화 하였다. 앞으로, 2次 年度에 開發豫定인 경지정리 전문가 시스템의 開發과 同時에 계속 補強하면서 연구를 遂行할 예정이다.

영상처리기법을 이용한 지형정보를 獲得하는 연구는 완성되었으며 이러한 기술을 2차년도의 耕地整理 전문가시스템 개발에 이용하기 위한 인터페이스 방법을 계속 研究할 것이다.

이러한 연구를 통해 궁극적으로 목표하는 바는 耕地의 形狀, 경사도, 土質,



區劃의 形狀 및 規模에 따른 農業機械의 圃場作業效率을 究明함으로써 農業機械의 效率的 利用에 관한 資料를 마련하고, 農作業 全般에 대한 데이터베이스의 完成으로 農作業에 대한 資料 獲得의 容易함과 더불어 營農計劃樹立에 全般的인 資料를 提示함으로써 農業振興地區를 中心으로 한 大區劃化, 大型 農機械體系에서 요구되는 適正의 機械選擇과 效率的인 農業機械 利用方案을 提示하고자 한다. 또한, 각 地域實情에 맞는 耕地基盤의 整備를 위한 設計基準을 設定하고, 동시에 耕地整理 전문가시스템을 이용한 農業機械利用 效率提高를 위한 경지정리에 관한 基礎設計方案을 提示하고자 한다. 이러한 전문가시스템을 이용하면 실제 경지정리 作業계획 수립에 앞서, 경지정리 前後의 전체적인 윤곽을 컴퓨터를 이용해 圖示하여, 공사비와 農業機械利用效率등과 같은 투자효과를 쉽게 비교할 수 있어 追後 세부적인 耕地整備 設計 方向을 올바르게 設定할 수 있으리라 생각된다.

## 參 考 文 獻

1. 鄭昌柱, 1988. 農作業機械學.
2. 鄭昌柱, 1991. “地域機械化計劃 및 評價技法의 開發에 관한 研究”.
3. 鄭昌柱 外 16人. 1992. 農作業機械의 分析과 設計. 서울大學校 出版部.
4. 이종호, 정창주. 1975. “벼 收穫機械의 適定所要能力測定을 위한 作業可能 日數率 分析” 韓國農工學會誌 17권.
5. 裴海英. 1992. 데이터베이스論.
6. 金字哲 外 7人. 1991. 現代統計學.
7. 國立農業資材檢査所. 1991. 農業機械檢査年譜.
8. 金基大 外. 1992. “農用原動機 の 燃比特性 と 農作業負荷モード의 適合化” 九州大學 研究報告.
9. 韓國降水資料(1965-1985). 第3圈 中央氣象臺.
10. 剛村俊民. 1991. “農業機械化의 基礎” 北海島大學 研究報告
11. 盧祥夏, 李種煥. 1990. “映像處理技術의 農業的 利用” 韓國農業機械學會 주최 심포지엄 發表文.
12. 日本全國土地改良事業團體連合會. 1988. 21世紀의 農業と 大區劃圓場整備を 考える.
13. 日本農林水産省 構造改善局 計劃部. 1990. “土地改良事業計劃指針, 圓場 整備(大區劃水田)”.
14. 日本農業機械化協會. 1985. “高性能農業機械導入基本方針 及び 參考資料”.
15. 日本農業研究 センター. 1990. “大區劃 水田 生産技術(第2集)”.

16. 日本農林統計協會 作業研究會. 1976. 農作業便覽.
17. 韓國農地改良研究所. 1990. 耕地整理事業施行體制改善과 今後의 發展方向 定立에 關한 研究 (1次年度).
18. 韓國農地改良研究所 農地改良組合聯合會. 1991. “耕地整理事業施行體制 改善과 今後의 發展方向 定立에 關한 研究”.
19. 石田茂樹. 1992. “Simulation of Field Work Efficiency of Agricultural Machines” Japanese Journal of Farm Work 27(1).
20. Date, C. J. 1981. An Introduction to Database Systems. Third Edition. Addison-wesley publishing company.
21. Donnel Huni. 1968. Farm Power and Machinery Management. Fifth Edition.
22. Giarratano, J. and G. Riley. 1989. Expert systems ; Principle and Programming. PWS - KENT publishing company.
23. Jain, A.K. 1989. Fundamentals of digital image precessing. Prentie-hall.
24. Kepner, R. A., R. Bainer and E. L. Barger. 1978. Principles of Farm Machinery. Third Edition. AVI publishing company.
25. Larry J. Segerlind. 1984. Applied Finite Element Analysis. John wiley and sons publishing company.
26. Matrox. 1989. IMAGER-AT MS DOS Software library for the MVP-AT video digitizer. Matrox, Canada.
27. N.R. Draper, H. Smith. 1981. Applied regression analysis.
28. Robert A. Byers. 1984. dBase III PLUS.
29. Rosenfeld, A. and C.K. Avinash. 1982. Digital picture processing. Academic Press.

# 附 錄

## 1. DATABASE PROGRAM LIST

```
* Program..: AG001.PRG
* Author...: KHU
* Date.....: 92.08.31
* Notice...: Copyright (c) 1992, KHU, All Rights Reserved
* Notes....:
* Reserved.: selectnum
CLEAR ALL
SET TALK OFF
SET HELP OFF
SET MENU OFF
SET SAFETY OFF
SET BELL OFF
SET STATUS OFF
SET ESCAPE OFF
SET CONFIRM ON
SET COLOR TO /W/B
DO WHILE .T.
  * ---Display menu options, centered on the screen.
  *   draw menu border and print heading
  CLEAR
  @ 1,2 TO 20,77 DOUBLE
  @ 3,29 SAY [ AGRICULTURAL MACHINERY ]
  @ 5,3 TO 5,76 DOUBLE
  * ---display detail lines
  @ 7,21 SAY [0. 에너지원      (Power,Transportation)]
  @ 8,21 SAY [1. 트랙터, 경운기 (Tractor,Power tiller)]
  @ 9,21 SAY [2. 경운정지용기계 (Tillage machinery)]
  @ 10,21 SAY [3. 이식, 파종, 시비 (Transplanter, Seeder, Fertilizer)]
  @ 11,21 SAY [4. 방제기, 제초기 (Chemical sprayer)]
  @ 12,21 SAY [5. 수확기      (Harvesting machinery)]
  @ 13,21 SAY [6. 조제가공기   (Processing machine)]
  @ 14,21 SAY [7. 시설물      (Farm facilities)]
  @ 15,21 SAY [8. 기타        (Others)]
  @ 18, 35 SAY '9. EXIT'
```



```

STORE 9 TO selectnum
SET COLOR TO /R/W
@ 20,33 SAY " 선택 "
SET COLOR TO /W/B
@ 20,42 GET selectnum PICTURE "9" RANGE 0,9
READ
DO CASE
  CASE selectnum = 9
    @ 6,17 CLEAR TO 20,64
    @ 6,17 TO 20,64
    @ 7,37 SAY [9. EXIT]
    @ 8,18 TO 8,63
    STORE ' ' TO select
    SET COLOR TO /G
    @ 13,29 SAY ' 작업을 마칠까요? (Y/n) '
    SET COLOR TO /W/B
    @ 13,COL( )+1 GET select PICT "A"
    READ
    IF UPPER(select) <> 'N'
      SET BELL ON
      SET TALK ON
      CLEAR ALL
      CLEAR
      RETURN
    ENDIF
  LOOP
  CASE selectnum = 0
    DO POWER0
  CASE selectnum = 1
    DO TRPT1
  CASE selectnum = 2
    DO TILMA2
  CASE selectnum = 3
    DO TSF3
  CASE selectnum = 4
    DO CHESP4
  CASE selectnum = 5
    DO HARMA5
  CASE selectnum = 6
    DO PROMA6
  CASE selectnum = 7
    DO FARFA7

```

```

        CASE selectnum = 8
          DO OTHERS8
        ENDCASE
        ENDDO T
        RETURN
        * EOF: AG001.PRG
        * Program.: AG002.PRG
        * Author...: KHU
        * Date.....: 92.08.31
        * Notice...: Copyright (c) 1992, KHU, All Rights Reserved
        * Notes....:
        * Reserved.: selectnum
        CLEAR ALL
        SET TALK OFF
        SET HELP OFF
        SET MENU OFF
        SET SAFETY OFF
        SET BELL OFF
        SET STATUS OFF
        SET ESCAPE OFF
        SET CONFIRM ON
        SET COLOR TO /W/B
        DO WHILE .T.

```

```

        * ---Display menu options, centered on the screen.
        *   draw menu border and print heading

```

```

        CLEAR
        @ 1,2 TO 20,77 DOUBLE
        @ 3,29 SAY [ AGICULTURAL    PROCESS ]
        @ 5,3 TO 5,76 DOUBLE
        * ---display detail lines
        @ 9,32 SAY [1. 수도작용 기계]
        @ 10,32 SAY [2. 전작용   기계]
        @ 18, 35 SAY '0. EXIT'
        STORE 0 TO selectnum
        SET COLOR TO /R/W
        @ 20,33 SAY " 선   택           "
        SET COLOR TO /W/B
        @ 20,42 GET selectnum PICTURE "9" RANGE 0,2
        READ
        DO CASE
          CASE selectnum = 0

```

```

    @ 6,17 CLEAR TO 20,64
    @ 6,17 TO 20,64
    @ 7,37 SAY [5. EXIT]
    @ 8,18 TO 8,63
    STORE ' ' TO select
    SET COLOR TO B/G
    @ 13,29 SAY ' 작업을 마칠까요? (Y/n) '
    SET COLOR TO /W/B
    @ 13,COL( )+1 GET select PICT "A"
    READ
    IF UPPER(select) <> 'N'
        SET BELL ON
        SET TALK ON
        CLEAR ALL
        CLEAR
        RETURN
    ENDIF
LOOP
CASE selectnum = 1
    DO RICE0
CASE selectnum = 2
    DO RICE2
ENDCASE
ENDDO T
RETURN
* EOF: AG002.PRG
DBFNAME = 'BIND'
TITLE = [바인더          (Binder)]
FORM = 'BIND.FMT'
DO STRUCT
CLEAR MEMORY
CLOSE ALL
CLEAR ALL
RETURN
DO WHILE .T.
    @ 4,10 CLEAR TO 20,70
    @ 4,10 TO 20,70 DOUBLE
    @ 5,24 SAY [4. 방제, 제초기 (Chemical sprayer)]
    @ 6,11 TO 6,69 DOUBLE
    @ 8,24 SAY [41. 인력용      (Use for human power)]
    @ 9,24 SAY [42. 제초기      (Weeder)]
    @ 10,24 SAY [43. 전정기     (Pruning machine)]

```

```

@ 11,24 SAY [44. 연무기 (Fog machine)]
@ 12,24 SAY [45. 분무기 (Sprayer)]

@ 18, 26 SAY '46. Return to MAIN MENU'
STORE 6 TO selectnum
SET COLOR TO /R/W
@ 20,33 SAY " 선택 4 "
SET COLOR TO /W/B
@ 20,42 GET selectnum PICTURE "9" RANGE 1,6
READ
DO CASE
CASE selectnum = 6
EXIT
CASE selectnum = 1
DO NODATA
CASE selectnum = 2
D20 NODATA
CASE selectnum = 3
DO NODATA
CASE selectnum = 4
DO NODATA
CASE selectnum = 5
DO NODATA
ENDCASE
DO CONTI
ENDDO
* EOF: CHESP4.PRG
DBFNAME = 'COMB'
TITLE = [자탈형콤바인 (Head-feeding)]
FORM = 'COMB.FMT'
DO STRUCT
CLEAR MEMORY
CLOSE ALL
CLEAR ALL
RETURN
* CONTI.PRG
SET CONFIRM OFF
STORE ' ' TO wait_subst
SET COLOR TO /G
@ 23,24 SAY '계속하려면 아무키나 누르시오...' GET wait_subst
SET COLOR TO /W/B
READ

```



```

        SET CONFIRM ON
        @ 23,00
        RETURN
* ERRTRAP.PRG
SET TALK OFF
  @ 23,02 SAY REPL(SPACE(1),76)
  @ 23,20 SAY "프린터가 연결되어 있지 않음...확인할 것..."
  CLEAR
* exit
  set device to screen
* SET PRINT OFF
* ON KEY
* CLOSE ALL
* EXIT
* ENDIF
RETURN
      DO WHILE .T.
        @ 4,10 CLEAR TO 20,70
        @ 4,10 TO 20,70 DOUBLE
        @ 5,24 SAY [7. 시설물      (Farm facilities)]
        @ 6,11 TO 6,69 DOUBLE
        @ 8,24 SAY [71. 육묘용시설 (Nursery plant)]
        @ 9,24 SAY [72. 비닐하우스 (Vinyl house)]
        @ 10,24 SAY [73. 도정공장   (Rice milling center)]
        @ 11,24 SAY [74. 선과장     (Assorting house)]
        @ 12,24 SAY [75. 축산용     (Use for live-stock)]
        @ 18, 26 SAY '76. Return to MAIN MENU'
        STORE 6 TO selectnum
        SET COLOR TO /R/W
        @ 20,33 SAY " 선택 7 "
        SET COLOR TO /W/B
        @ 20,42 GET selectnum PICTURE "9" RANGE 1,6
        READ
        DO CASE
          CASE selectnum = 6
            EXIT
          CASE selectnum = 1
            DO NODATA
          CASE selectnum = 2
            DO NODATA
          CASE selectnum = 3
            DO NODATA

```

```

CASE selectnum = 4
  DO NODATA
CASE selectnum = 5
  DO WHILE .T.
    @ 6,17 CLEAR TO 20,64
    @ 6,17 TO 20,64
    @ 7,24 SAY [75. 축산용      (Use for live-stock)]
    @ 8,18 TO 8,63
    @ 10,24 SAY [751. 급이기      (Feeder)]
    @ 11,24 SAY [752. 착유기      (Milker)]
    @ 12,24 SAY [753. 우유냉각기 (Milk cooler)]
    @ 18,33 SAY '754. Return'
  STORE 4 TO selectnum
  SET COLOR TO /R/W
  @ 20,33 SAY " 선 택 75 "
  SET COLOR TO /W/B
  @ 20,43 GET selectnum PICTURE "9" RANGE 1,4
  READ
  DO CASE
    CASE selectnum = 4
      EXIT
    CASE selectnum = 1
      DO NODATAS
    CASE selectnum = 2
      DO NODATAS
    CASE selectnum = 3
      DO NODATAS
  ENDCASE
  DO CONTI
  ENDDO
  LOOP
  ENDCASE
  DO CONTI
  ENDDO
DBFNAME = 'FRUIT'
TITLE = [과실, 야채선별기(Grading machine)]
FORM = 'FRUIT.FMT'
DO STRUCT
CLEAR MEMORY
CLOSE ALL
CLEAR ALL
RETURN

```

```

DBFNAME = 'GDRYER'
TITLE = [곡물건조기(Grain Dryer)]
FORM = 'GDRYER.FMT'
DO STRUCT
CLEAR MEMORY
CLOSE ALL
CLEAR ALL
RETURN
    DO WHILE .T.
        @ 4,10 CLEAR TO 20,70
        @ 4,10 TO 20,70 DOUBLE
        @ 5,24 SAY [5. 수확기      (Harvseting machinery)]
        @ 6,11 TO 6,69 DOUBLE
        @ 8,24 SAY [51. 인력수확      (Harvest by hand)]
        @ 9,24 SAY [52. 바인더      (Binder)]
        @ 10,24 SAY [53. 콤바인      (combine)]
        @ 11,24 SAY [54. 탈곡기      (Thresher)]
        @ 12,24 SAY [55. 사료작물수확 (Harvest for hay)]
        @ 13,24 SAY [56. 과실,야채수확 (Harvest for fruit ]
        @ 14,24 SAY [                & vegitable)]
        @ 18, 26 SAY '57. Return to MAIN MENU'
        STORE 7 TO selectnum
        SET COLOR TO /R/W
        @ 20,33 SAY " 선 택 5 "
        SET COLOR TO /W/B
        @ 20,42 GET selectnum PICTURE "9" RANGE 1,7
        READ
        DO CASE
            CASE selectnum = 7
                EXIT
            CASE selectnum = 1
                DO NODATA
            CASE selectnum = 2
                @ 7,12 CLEAR TO 19,69
                SET COLOR TO /W
                @ 12,24 SAY [52. 바인더      (Binder)]
                SET COLOR TO /W/B
                DO BIND001
                LOOP
            CASE selectnum = 3
                DO WHILE .T.
                    @ 6,17 CLEAR TO 20,64

```

```

@ 6,17 TO 20,64
@ 7,24 SAY [53. 콤바인 (combine)]
@ 8,18 TO 8,63
@ 10,24 SAY [531. 보통형콤바인 (combine)]
@ 11,24 SAY [532. 자탈형콤바인 (Head-feeding)]
@ 12,24 SAY [ (combine)]
@ 18,33 SAY '533. Return'
STORE 3 TO selectnum
SET COLOR TO /R/W
@ 20,33 SAY " 선택 53 "
SET COLOR TO /W/B
@ 20,43 GET selectnum PICTURE "9" RANGE 1,3
READ
DO CASE
CASE selectnum = 3
EXIT
CASE selectnum = 1
DO NODATAS
CASE selectnum = 2
@ 9,19 CLEAR TO 19,63
SET COLOR TO /W
@ 12,24 SAY [532. 자탈형콤바인 (Head-feeding)]
@ 13,24 SAY [ (combine)]
SET COLOR TO /W/B
DO COMB001
LOOP
ENDCASE
DO CONTI
ENDDO
LOOP
CASE selectnum = 4
DO WHILE .T.
@ 6,17 CLEAR TO 20,64
@ 6,17 TO 20,64
@ 7,24 SAY [54. 탈곡기 (Thresher)]
@ 8,18 TO 8,63
@ 10,24 SAY [541. 자동탈곡기 (Self-feeding)]
@ 11,24 SAY [542. 반자동탈곡기 (Power thresher)]
@ 12,24 SAY [543. 주행용탈곡기 (Self-propelled)]
@ 13,24 SAY [544. 기타 (Others except for]
@ 14,24 SAY [ (rice & wheat)]
@ 18,33 SAY '545. Return'

```



```

STORE 5 TO selectnum
SET COLOR TO /R/W
@ 20,33 SAY " 선택 54 "
SET COLOR TO /W/B
@ 20,43 GET selectnum PICTURE "9" RANGE 1,5
READ
DO CASE
  CASE selectnum = 5
    EXIT
  CASE selectnum = 1
    DO NODATAS
  CASE selectnum = 2
    DO NODATAS
  CASE selectnum = 3
    DO NODATAS
  CASE selectnum = 4
    DO NODATAS
  ENDCASE
DO CONTI
ENDDO
LOOP
CASE selectnum = 5
DO WHILE .T.
@ 6,17 CLEAR TO 20,64
@ 6,17 TO 20,64
@ 7,24 SAY [55. 사료작물수확 (Harvest for hay)]
@ 8,18 TO 8,63
@ 10,24 SAY [551. 모우어 (Mower)]
@ 11,24 SAY [552. 헤이콘디셔너 (Hay conditioner)]
@ 12,24 SAY [553. 테더레이커 (Tedder-Rake)]
@ 13,24 SAY [554. 베일러 (Baler)]
@ 14,24 SAY [555. 사료수확기 (Forage harvester)]
@ 15,24 SAY [556. 커터 (Cutter)]
@ 18,33 SAY '557. Return'
STORE 7 TO selectnum
SET COLOR TO /R/W
@ 20,33 SAY " 선택 55 "
SET COLOR TO /W/B
@ 20,43 GET selectnum PICTURE "9" RANGE 1,7
READ
DO CASE
  CASE selectnum = 7

```

```

EXIT
CASE selectnum = 1
  DO NODATAS
CASE selectnum = 2
  DO NODATAS
CASE selectnum = 3
  DO NODATAS
CASE selectnum = 4
  DO NODATAS
CASE selectnum = 5
  DO NODATAS
CASE selectnum = 6
  DO NODATAS
ENDCASE
DO CONTI
ENDDO
LOOP
CASE selectnum = 6
  DO WHILE .T.
    @ 6,17 CLEAR TO 20,64
    @ 6,17 TO 20,64
    @ 7,24 SAY [56. 과실,야채수확 (Harvest for fruit)]
    @ 8,18 TO 8,63
    @ 10,24 SAY [561. 과실,야채수확기 (Harvester)]
    @ 11,24 SAY [562. 과실,야채선별기 (Grading machine)]
    @ 12,24 SAY [563. 과실,야채세척기 (Washer)]
    @ 18,33 SAY '564. Return'
  STORE 4 TO selectnum
  SET COLOR TO /R/W
  @ 20,33 SAY " 선 택 56 "
  SET COLOR TO /W/B
  @ 20,43 GET selectnum PICTURE "9" RANGE 1,4
  READ
  DO CASE
    CASE selectnum = 4
      EXIT
    CASE selectnum = 1
      DO NODATAS
    CASE selectnum = 2
      @ 7, 12 CLEAR TO 19, 69
      SET COLOR TO /W
      @ 12, 18 SAY [562. 과실,야채선별기(Grading

```

machine)]

```
                SET COLOR TO /W/B
                DO FRUI001
                LOOP
                CASE selectnum = 3
                DO NODATAS
                ENDCASE
                DO CONTI
                ENDDO
                LOOP
                ENDCASE
                DO CONTI
                ENDDO
* KEYPRESS.PRG
SET DEVICE TO SCREEN
*SET PRINT OFF
dummy = INKEY()
abort = .T.
@ 23,02 SAY REPL(SPACE(1),76)
@ 23,25 SAY "프린트를 중지할까요 ? (Y/n)" GET abort PICTURE "Y"
READ
IF abort
SET DEVICE TO SCREEN
* SET PRINT OFF
  ON KEY
  CLOSE ALL
  EXIT
ENDIF
RETURN

@ 7,12 CLEAR TO 19,69
SET COLOR TO /W
@ 12,32 SAY ' 해당자료없음...'
SET COLOR TO /W/B
RETURN
@ 9,19 CLEAR TO 19,63
SET COLOR TO /W
@ 12,32 SAY ' 해당자료없음...'
SET COLOR TO /W/B
RETURN

DBFNAME = 'ODRYER'
TITLE = [기타건조기(Other Dryer)]
FORM = 'ODRYER.FMT'
```

```

DO STRUCT
CLEAR MEMORY
CLOSE ALL
CLEAR ALL
RETURN

```

```

DO WHILE .T.
@ 4,10 CLEAR TO 20,70
@ 4,10 TO 20,70 DOUBLE
@ 5,24 SAY [8. 기타 (Others)]
@ 6,11 TO 6,69 DOUBLE
@ 8,24 SAY [81. 양잠용기계 (Machinery for sericulture)]
@ 9,24 SAY [82. 짚가공기계 (Straw processing machine)]
@ 10,24 SAY [83. 낙농기계 (Farming machine)]
@ 11,24 SAY [84. 특수기계 (Special type machine)]
@ 12,24 SAY [85. 수분측정기 (Moisture meter)]
@ 18, 26 SAY '86. Return to MAIN MENU'
STORE 6 TO selectnum
SET COLOR TO /R/W
@ 20,33 SAY " 선택 8 "
SET COLOR TO /W/B
@ 20,42 GET selectnum PICTURE "9" RANGE 1,6
READ
DO CASE
CASE selectnum = 6
EXIT
CASE selectnum = 1
DO NODATA
CASE selectnum = 2
DO NODATA
CASE selectnum = 3
DO NODATA
CASE selectnum = 4
DO WHILE .T.
@ 6,17 CLEAR TO 20,64
@ 6,17 TO 20,64
@ 7,24 SAY [84. 특수기계 (Special type machine)]
@ 8,18 TO 8,63
@ 10,24 SAY [841. 온풍기 (Heater)]
@ 18,33 SAY '842. Return'
STORE 2 TO selectnum
SET COLOR TO /R/W
@ 20,33 SAY " 선택 84 "

```



```

        SET COLOR TO /W/B
        @ 20,43 GET selectnum PICTURE "9" RANGE 1,2
        READ
        DO CASE
            CASE selectnum = 2
                EXIT
            CASE selectnum = 1
                DO NODATAS
            ENDCASE
        DO CONTI
        ENDDO
        LOOP
        CASE selectnum = 5
            DO NODATA
        ENDCASE
    DO CONTI
    ENDDO
    DO WHILE .T.
        @ 4,10 CLEAR TO 20,70
        @ 4,10 TO 20,70 DOUBLE
        @ 5,24 SAY [0. 에너지원 (Power source)]
        @ 6,11 TO 6,69 DOUBLE
        @ 8,24 SAY [00. 인력, 동력 (Human power, engine)]
        @ 9,24 SAY [01. 축력 (Animal)]
        @ 10,24 SAY [02. 자연 (Nature)]
        @ 11,24 SAY [03. 전동기 (Motor)]
        @ 12,24 SAY [04. 엔진 (Engine)]
        @ 13,24 SAY [05. 펌프 (Pump)]
        @ 14,24 SAY [06. 발전기 (Generator)]
        @ 15,24 SAY [07. 운반장비 (Farm transport equip)]
        @ 16,24 SAY [08. 반송장비 (Conveying machine)]
        @ 18, 26 SAY '09. Return to MAIN MENU'
        STORE 9 TO selectnum
        SET COLOR TO /R/W
        @ 20,33 SAY " 선택 0 "
        SET COLOR TO /W/B
        @ 20,42 GET selectnum PICTURE "9" RANGE 0,9
        READ
        DO CASE
            CASE selectnum = 9
                EXIT
            CASE selectnum = 0

```

```

DO WHILE .T.
  @ 6,17 CLEAR TO 20,64
  @ 6,17 TO 20,64
  @ 7,24 SAY [00. 인력,동력 (Human power,engine)]
  @ 8,18 TO 8,63
  @ 10,24 SAY [001. 수동 (Hand)]
  @ 18,33 SAY '002. Return'
STORE 2 TO selectnum
SET COLOR TO /R/W
  @ 20,33 SAY " 선택 00 "
SET COLOR TO /W/B
  @ 20,43 GET selectnum PICTURE "9" RANGE 1,2
READ
DO CASE
  CASE selectnum = 2
    EXIT
  CASE selectnum = 1
    DO NODATAS
  ENDCASE
DO CONTI
ENDDO
LOOP
CASE selectnum = 1
  DO NODATA
CASE selectnum = 2
  DO WHILE .T.
    @ 6,17 CLEAR TO 20,64
    @ 6,17 TO 20,64
    @ 7,24 SAY [02. 자연 (Nature)]
    @ 8,18 TO 8,63
    @ 10,24 SAY [021. 태양 (Sun)]
    @ 11,24 SAY [022. 풍력 (Wind power)]
    @ 12,24 SAY [023. 수력 (Water power)]
    @ 18,33 SAY '024. Return'
STORE 4 TO selectnum
SET COLOR TO /R/W
  @ 20,33 SAY " 선택 02 "
SET COLOR TO /W/B
  @ 20,43 GET selectnum PICTURE "9" RANGE 1,4
READ
DO CASE
  CASE selectnum = 4

```

```

EXIT
CASE selectnum = 1
    DO NODATAS
CASE selectnum = 2
    DO NODATAS
CASE selectnum = 3
    DO NODATAS
ENDCASE
DO CONTI
ENDDO
LOOP
CASE selectnum = 3
    DO NODATA
CASE selectnum = 4
    DO WHILE .T.
        @ 6,17 CLEAR TO 20,64
        @ 6,17 TO 20,64
        @ 7,24 SAY [04. 엔진          (Engine)]
        @ 8,18 TO 8,63
        @ 10,24 SAY [041. 디젤엔진   (Diesel engine)]
        @ 11,24 SAY [042. 가솔린엔진 (Gasoline engine)]
        @ 18,33 SAY '043. Return'
        STORE 3 TO selectnum
        SET COLOR TO /R/W
        @ 20,33 SAY " 선택 04 "
        SET COLOR TO /W/B
        @ 20,43 GET selectnum PICTURE "9" RANGE 1,3
    READ
    DO CASE
        CASE selectnum = 3
            EXIT
        CASE selectnum = 1
            DO NODATAS
        CASE selectnum = 2
            DO NODATAS
    ENDCASE
    DO CONTI
    ENDDO
    LOOP
CASE selectnum = 5
    DO NODATA
CASE selectnum = 6

```

```

DO NODATA
CASE selectnum = 7
DO NODATA
CASE selectnum = 8
DO WHILE .T.
@ 6,17 CLEAR TO 20,64
@ 6,17 TO 20,64
@ 7,24 SAY [08. 반송장비 (Conveying machine)]
@ 8,18 TO 8,63
@ 10,24 SAY [081. 로우더 등 (Loader and others)]
@ 18,33 SAY '082. Return'
STORE 2 TO selectnum
SET COLOR TO /R/W
@ 20,33 SAY " 선택 08 "
SET COLOR TO /W/B
@ 20,43 GET selectnum PICTURE "9" RANGE 1,2
READ
DO CASE
CASE selectnum = 2
EXIT
CASE selectnum = 1
DO NODATAS
ENDCASE
DO CONTI
ENDDO
LOOP
ENDCASE
DO CONTI
ENDDO
* EOF: POWER0.PRG
* PRINT.PRG
ON ERROR DO ERRTRAP
@ 23,01 CLEAR TO 23,78
SET COLOR TO /R/W
@ 23,13 SAY "화면으로 인쇄는 [1]번 키를 아니면 임의의 키를 칠 것..."
SET COLOR TO /W/B
slyn = " "
@ 23,COL()+1 GET slyn PICTURE "X"
READ
pline = IIF (slyn = "1",21,58)
@ 23,01 CLEAR TO 23,78
prinsr = RECNO()

```



```

DO WHILE .NOT.(EOF() .OR. plr < RECNO())
  IF pline = 21
    SET DEVICE TO SCREEN
    CLEAR
    @ 22,00 TO 22,79 DOUBLE
  ELSE
    SET COLOR TO /W/B
    @ 23,01 CLEAR TO 23,78
    SET COLOR TO /W
    @ 23,18 SAY "인쇄 작업 준비완료 후 [ENTER]키를 치면 인쇄..."
    pyn = INKEY()
    DO WHILE pyn <> 13
      pyn = INKEY()
    ENDDO
    SET COLOR TO /W/B
    @ 23,01 CLEAR TO 23,78
    SET COLOR TO /R/W
    @ 23,30 SAY "인쇄 작업 진행 중 ..."
    SET COLOR TO /W/B
    SET DEVICE TO PRINT
    * SET PRINT ON
    * ON KEY DO KEYPRESS
    EJECT
  ENDIF
  @ 03,20 SAY REPL("=",40)
  @ 04,27 SAY TITLE
  @ 05,20 SAY REPL("=",40)
  @ 07,00 SAY REPL("-",76)
  @ 08,02 SAY "RECNO"
  @ 08,09 SAY " MACHINE "
  @ 08,20 SAY "  MAKER  "
  @ 08,30 SAY "  MODEL  "
  @ 08,45 SAY "  TYPE   "
  @ 08,60 SAY "PRICE(won)"
  @ 09,00 SAY REPL("-",76)
  prwn = 10
DO WHILE .NOT.(EOF() .OR. plr < RECNO() .OR. prwn > pline)
  @ prwn,02 SAY STR(RECNO(),5)
  @ prwn,09 SAY MACHINE
  @ prwn,20 SAY MAKER
  @ prwn,30 SAY MODEL
  @ prwn,45 SAY TYPE

```

```

    @ pown, 60 SAY STR(PRICE_WON, 10)
    pown = pown + 1
    &nextrec
ENDDO
IF pline = 21
    SET COLOR TO /W/B
    @ 23, 01 CLEAR TO 23, 78
    SET COLOR TO /W
    @ 23, 10 SAY "다음 화면 표시 [1]번... 출력 장소 변경시 [9]번 키를
    칠 것..."
    pyn = INKEY()
    DO WHILE .NOT. (pyn = 49 .OR. pyn = 57)
        pyn = INKEY()
    ENDDO
    SET COLOR TO /W/B
    IF pyn = 57
        GO prinsr
        pline = 58
    ELSE
        IF .NOT. (plr > RECNO())
            @ 23, 01 CLEAR TO 23, 78
            SET COLOR TO /W
            @ 23, 18 SAY "표시 레코드 마지막 지점 [ENTER]키를 칠 것..."
            pyn = INKEY()
            DO WHILE pyn <> 13
                pyn = INKEY()
            ENDDO
            SET COLOR TO /W/B
            SKIP - 11
            return
        ENDIF
    ENDIF
ELSE
    @ pown+1, 00 SAY SPACE()
ENDIF
ENDDO
*set print off
SET DEVICE TO SCREEN
CLOSE ALL
RETURN
    DO WHILE .T.
        @ 4, 10 CLEAR TO 20, 70

```

```

@ 4,10 TO 20,70 DOUBLE
@ 5,24 SAY [6. 조제가공기 (Processing machine)]
@ 6,11 TO 6,69 DOUBLE
@ 8,24 SAY [61. 인력조제가공 (Processing by hand)]
@ 9,24 SAY [62. 건조기 (Dryer)]
@ 10,24 SAY [63. 사료분쇄기 (Stalk cutter)]
@ 11,24 SAY [64. 사료배합기 (Fodder mixer)]
@ 12,24 SAY [65. 제분기 (Wheat mill)]
@ 13,24 SAY [66. 벼조제가공 (Rice processing)]
@ 18, 26 SAY '67. Return to MAIN MENU'
STORE 7 TO selectnum
SET COLOR TO /R/W
@ 20,33 SAY " 선택 6 "
SET COLOR TO /W/B
@ 20,42 GET selectnum PICTURE "9" RANGE 1,8
READ
DO CASE
CASE selectnum = 7
EXIT
CASE selectnum = 1
DO NODATA
CASE selectnum = 2
DO WHILE .T.
@ 6,17 CLEAR TO 20,64
@ 6,17 TO 20,64
@ 7,24 SAY [62. 건조기 (Dryer)]
@ 8,18 TO 8,63
@ 10,24 SAY [621. 곡물건조기 (Grain dryer)]
@ 11,24 SAY [622. 기타건조기 (Other dryer)]
@ 18,33 SAY '623. Return'
STORE 3 TO selectnum
SET COLOR TO /R/W
@ 20,33 SAY " 선택 62 "
SET COLOR TO /W/B
@ 20,43 GET selectnum PICTURE "9" RANGE 1,3
READ
DO CASE
CASE selectnum = 3
EXIT
CASE selectnum = 1
@ 7, 12 CLEAR TO 19, 69
SET COLOR TO /W

```

```

        @ 12, 18 SAY [621. 곡물건조기(Grain Dryer)]
        SET COLOR TO /W/B
        DO GDRY001
        LOOP
CASE selectnum = 2
        @ 7, 12 CLEAR TO 19, 69
        SET COLOR TO /W
        @ 12, 18 SAY [622. 기타건조기(Other Dryer)]
        SET COLOR TO /W/B
        DO ODRY001
        LOOP
    ENDCASE
DO CONTI
ENDDO
LOOP
CASE selectnum = 3
    DO NODATA
CASE selectnum = 4
    DO NODATA
CASE selectnum = 5
    DO NODATA
CASE selectnum = 6
    DO WHILE .T.
        @ 6, 17 CLEAR TO 20, 64
        @ 6, 17 TO 20, 64
        @ 7, 24 SAY [66. 벼조제가공 (Rice processing)]
        @ 8, 18 TO 8, 63
        @ 10, 24 SAY [661. 도정기 (Rice mill)]
        @ 11, 24 SAY [662. 석발기 (Stone separator)]
        @ 12, 24 SAY [663. 현미기 (Rice huller)]
        @ 13, 24 SAY [664. 현미분리기 (Brown rice separator)]
        @ 14, 24 SAY [665. 정미기 (Rice whitener)]
        @ 15, 24 SAY [666. 선별기 (Separator)]
        @ 18, 33 SAY '667. Return'
        STORE 7 TO selectnum
        SET COLOR TO /R/W
        @ 20, 33 SAY " 선 택 66 "
        SET COLOR TO /W/B
        @ 20, 43 GET selectnum PICTURE "9" RANGE 1, 7
    READ
DO CASE
    CASE selectnum = 7

```



```

EXIT
CASE selectnum = 1
  DO NODATAS
CASE selectnum = 2
  DO NODATAS
CASE selectnum = 3
  DO NODATAS
CASE selectnum = 4
  DO NODATAS
CASE selectnum = 5
  DO NODATAS
CASE selectnum = 6
  DO NODATAS
ENDCASE
DO CONTI
ENDDO
LOOP
ENDCASE
DO CONTI
ENDDO
DBFNAME = 'PTPLOW'
TITLE = [경운기플라우 (Power tiller plow)]
FORM = 'PTPLOW.FMT'
DO STRUCT
CLEAR MEMORY
CLOSE ALL
CLEAR ALL
RETURN
CLEAR
@ 01,00 TO 24,79 DOUBLE
SET COLOR TO /W
@ 02,02 SAY "RECNO"
@ 02,COL()+2 SAY " MACHINE "
@ 02,COL()+2 SAY "  MAKER  "
@ 02,COL()+2 SAY "  MODEL  "
@ 02,COL()+2 SAY "  TYPE   "
@ 02,COL()+2 SAY "PRICE(won)"
SET COLOR TO /W/B
@ 03,01 TO 03,78
@ 22,01 TO 22,78
DO WHILE .NOT.EOF()
  rowno = 4

```

```

@ 04,01 CLEAR TO 21,78
DO WHILE .NOT. (EOF() .OR. rowno = 22)
  @ rowno, COL()+2 SAY MACHINE
  @ rowno, COL()+2 SAY MAKER
  @ rowno, COL()+2 SAY MODEL
  @ rowno, COL()+2 SAY TYPE
  @ rowno, COL()+2 SAY STR(PRICE_WON,10)
  rowno = rowno + 1
  SKIP
  IF MOD(rowno,5) = 0
    SET COLOR TO /W
  ELSE
    SET COLOR TO /W/B
  ENDIF
ENDDO
SET COLOR TO /G
@ 23,02 SAY "[PgUp/PgDn]"
SET COLOR TO /W/B
@ 23,COL() SAY " : 상/하 화면 레코드 데이터 표시 "
SET COLOR TO /G
@ 23,COL()+1 SAY "[Enter]"
SET COLOR TO /W/B
@ 23,COL() SAY " : 레코드 확인 작업 종료 "
keyin = INKEY()
DO WHILE .NOT. (keyin=13 .OR. keyin=18 .OR. keyin=3)
  keyin = INKEY()
ENDDO
IF keyin = 13
  EXIT
* LOOP
ENDIF
IF keyin = 18
  IF RECNO() >= 36
    SKIP - (18 + rowno - 4)
  ELSE
    GO TOP
  ENDIF
ELSE
  IF RECNO() >= RECCOUNT()
    SKIP - 18
  ENDIF
ENDIF
ENDIF

```

```

ENDDO
CLEAR
@ 01,00 TO 24,79 DOUBLE
SET COLOR TO /W
@ 02,02 SAY "RECNO"
@ 02,COL()+2 SAY " MACHINE "
@ 02,COL()+2 SAY "  MAKER  "
@ 02,COL()+2 SAY "  MODEL  "
@ 02,COL()+2 SAY "  TYPE   "
@ 02,COL()+2 SAY "PRICE(won)"
SET COLOR TO /W/B
@ 03,01 TO 03,78
@ 22,01 TO 22,78
DO WHILE .NOT.EOF()
  rowno = 4
  @ 04,01 CLEAR TO 21,78
  DO WHILE .NOT. (EOF() .OR. rowno = 22)
    @ rowno,02 SAY STR(RECNO(),5)
    @ rowno,COL()+2 SAY MACHINE
    @ rowno,COL()+2 SAY MAKER
    @ rowno,COL()+2 SAY MODEL
    @ rowno,COL()+2 SAY TYPE
    @ rowno,COL()+2 SAY STR(PRICE_WON,10)
    rowno = rowno + 1
    SKIP
    IF MOD(rowno,5) = 0
      SET COLOR TO /W
    ELSE
      SET COLOR TO /W/B
    ENDIF
  ENDDO
  SET COLOR TO /G
  @ 23,02 SAY "[PgUp/PgDn]"
  SET COLOR TO /W/B
  @ 23,COL() SAY " : 상/하 화면 레코드 데이터 표시 "
  SET COLOR TO /G
  @ 23,COL()+1 SAY "[Enter]"
  SET COLOR TO /W/B
  @ 23,COL() SAY " : 레코드 확인 작업 종료 "
  keyin = INKEY()
  DO WHILE .NOT. (keyin=13 .OR. keyin=18 .OR. keyin=3)
    keyin = INKEY()

```

```

ENDDO
IF keyin = 13
  EXIT
* LOOP
ENDIF
IF keyin = 18
  IF RECNO() >= 36
    SKIP - (18 + rowno - 4)
  ELSE
    GO TOP
  ENDIF
ELSE
  IF RECNO() >= RECCOUNT()
    SKIP - 18
  ENDIF
ENDIF
ENDDO
CLEAR
@ 01,00 TO 24,79 DOUBLE
SET COLOR TO /W
@ 02,02 SAY "RECNO"
@ 02,COL()+2 SAY " MACHINE "
@ 02,COL()+2 SAY "  MAKER  "
@ 02,COL()+2 SAY "  MODEL  "
@ 02,COL()+2 SAY "  TYPE   "
@ 02,COL()+2 SAY "PRICE(won)"
SET COLOR TO /W/B
@ 03,01 TO 03,78
@ 22,01 TO 22,78
DO WHILE .NOT.EOF()
  rowno = 4
  @ 04,01 CLEAR TO 21,78
  DO WHILE .NOT. (EOF() .OR. rowno = 22)
    @ rowno,02 SAY STR(RECNO(),5)
    @ rowno,COL()+2 SAY MACHINE
    @ rowno,COL()+2 SAY MAKER
    @ rowno,COL()+2 SAY MODEL
    @ rowno,COL()+2 SAY TYPE
    @ rowno,COL()+2 SAY STR(PRICE_WON,10)
    rowno = rowno + 1
  SKIP
  IF MOD(rowno,5) = 0

```



```

        SET COLOR TO /W
    ELSE
        SET COLOR TO /W/B
    ENDIF
ENDDO
SET COLOR TO /G
@ 23,02 SAY "[PgUp/PgDn]"
SET COLOR TO /W/B
@ 23,COL() SAY " : 상/하 화면 레코드 데이터 표시 "
SET COLOR TO /G
@ 23,COL()+1 SAY "[Enter]"
SET COLOR TO /W/B
@ 23,COL() SAY " : 레코드 확인 작업 종료 "
keyin = INKEY()
DO WHILE .NOT. (keyin=13 .OR. keyin=18 .OR. keyin=3)
    keyin = INKEY()
ENDDO
IF keyin = 13
    EXIT
*   LOOP
ENDIF
IF keyin = 18
    IF RECNO() >= 36
        SKIP - (18 + rowno - 4)
    ELSE
        GO TOP
    ENDIF
ELSE
    IF RECNO() >= RECCOUNT()
        SKIP - 18
    ENDIF
ENDIF
ENDDO
CLEAR
@ 01,00 TO 24,79 DOUBLE
SET COLOR TO /W
@ 02,02 SAY "RECNO"
@ 02,COL()+2 SAY " MACHINE "
@ 02,COL()+2 SAY "  MAKER  "
@ 02,COL()+2 SAY "  MODEL  "
@ 02,COL()+2 SAY "  TYPE   "
@ 02,COL()+2 SAY "PRICE(won)"

```

```

SET COLOR TO /W/B
@ 03,01 TO 03,78
@ 22,01 TO 22,78
DO WHILE .NOT.EOF()
  rowno = 4
  @ 04,01 CLEAR TO 21,78
  DO WHILE .NOT. (EOF() .OR. rowno = 22)
    @ rowno,02 SAY STR(RECNO()),5)
    @ rowno,COL()+2 SAY MACHINE
    @ rowno,COL()+2 SAY MAKER
    @ rowno,COL()+2 SAY MODEL
    @ rowno,COL()+2 SAY TYPE
    @ rowno,COL()+2 SAY STR(PRICE_WON,10)
    rowno = rowno + 1
    SKIP
    IF MOD(rowno,5) = 0
      SET COLOR TO /W
    ELSE
      SET COLOR TO /W/B
    ENDIF
  ENDDO
  SET COLOR TO /G
  @ 23,02 SAY "[PgUp/PgDn]"
  SET COLOR TO /W/B
  @ 23,COL() SAY " : 상/하 화면 레코드 데이터 표시 "
  SET COLOR TO /G
  @ 23,COL()+1 SAY "[Enter]"
  SET COLOR TO /W/B
  @ 23,COL() SAY " : 레코드 확인 작업 종료 "
  keyin = INKEY()
  DO WHILE .NOT. (keyin=13 .OR. keyin=18 .OR. keyin=3)
    keyin = INKEY()
  ENDDO
  IF keyin = 13
    EXIT
  * LOOP
  ENDIF
  IF keyin = 18
    IF RECNO() >= 36
      SKIP - (18 + rowno - 4)
    ELSE
      GO TOP
    ENDIF
  ENDIF

```

```

    ENDIF
ELSE
    IF RECNO() >= RECCOUNT()
        SKIP - 18
    ENDIF
ENDIF
ENDDO
CLEAR
@ 01,00 TO 24,79 DOUBLE
SET COLOR TO /W
@ 02,02 SAY "RECNO"
@ 02,COL()+2 SAY " MACHINE "
@ 02,COL()+2 SAY "  MAKER  "
@ 02,COL()+2 SAY "  MODEL  "
@ 02,COL()+2 SAY "  TYPE   "
@ 02,COL()+2 SAY "PRICE(won)"
SET COLOR TO /W/B
@ 03,01 TO 03,78
@ 22,01 TO 22,78
DO WHILE .NOT.EOF()
    rowno = 4
    @ 04,01 CLEAR TO 21,78
    DO WHILE .NOT. (EOF() .OR. rowno = 22)
        @ rowno,02 SAY STR(RECNO(),5)
        @ rowno,COL()+2 SAY MACHINE
        @ rowno,COL()+2 SAY MAKER
        @ rowno,COL()+2 SAY MODEL
        @ rowno,COL()+2 SAY TYPE
        @ rowno,COL()+2 SAY STR(PRICE_WON,10)
        rowno = rowno + 1
        SKIP
        IF MOD(rowno,5) = 0
            SET COLOR TO /W
        ELSE
            SET COLOR TO /W/B
        ENDIF
    ENDDO
    SET COLOR TO /G
    @ 23,02 SAY "[PgUp/PgDn]"
    SET COLOR TO /W/B
    @ 23,COL() SAY " : 상/하 화면 레코드 데이터 표시 "
    SET COLOR TO /G

```

```

@ 23, COL()+1 SAY "[Enter]"
SET COLOR TO /W/B
@ 23, COL() SAY " : 레코드 확인 작업 종료 "
keyin = INKEY()
DO WHILE .NOT. (keyin=13 .OR. keyin=18 .OR. keyin=3)
    keyin = INKEY()
ENDDO
IF keyin = 13
    EXIT
*   LOOP
ENDIF
IF keyin = 18
    IF RECNO() >= 36
        SKIP - (18 + rowno - 4)
    ELSE
        GO TOP
    ENDIF
ELSE
    IF RECNO() >= RECCOUNT()
        SKIP - 18
    ENDIF
ENDIF
ENDDO
CLEAR
@ 01, 00 TO 24, 79 DOUBLE
SET COLOR TO /W
@ 02, 02 SAY "RECNO"
@ 02, COL()+2 SAY " MACHINE "
@ 02, COL()+2 SAY "  MAKER  "
@ 02, COL()+2 SAY "  MODEL  "
@ 02, COL()+2 SAY "  TYPE   "
@ 02, COL()+2 SAY "PRICE(won)"
SET COLOR TO /W/B
@ 03, 01 TO 03, 78
@ 22, 01 TO 22, 78
DO WHILE .NOT. EOF()
    rowno = 4
    @ 04, 01 CLEAR TO 21, 78
    DO WHILE .NOT. (EOF() .OR. rowno = 22)
        @ rowno, 02 SAY STR(RECNO(), 5)
        @ rowno, COL()+2 SAY MACHINE
        @ rowno, COL()+2 SAY MAKER
    
```

```

@ rowno, COL()+2 SAY MODEL
@ rowno, COL()+2 SAY TYPE
@ rowno, COL()+2 SAY STR(PRICE_WON, 10)
rowno = rowno + 1
SKIP
IF MOD(rowno, 5) = 0
    SET COLOR TO /W
ELSE
    SET COLOR TO /W/B
ENDIF
ENDDO
SET COLOR TO /G
@ 23, 02 SAY "[PgUp/PgDn]"
SET COLOR TO /W/B
@ 23, COL() SAY " : 상/하 화면 레코드 데이터 표시 "
SET COLOR TO /G
@ 23, COL()+1 SAY "[Enter]"
SET COLOR TO /W/B
@ 23, COL() SAY " : 레코드 확인 작업 종료 "
keyin = INKEY()
DO WHILE .NOT. (keyin=13 .OR. keyin=18 .OR. keyin=3)
    keyin = INKEY()
ENDDO
IF keyin = 13
    EXIT
* LOOP
ENDIF
IF keyin = 18
    IF RECNO() >= 36
        SKIP - (18 + rowno - 4)
    ELSE
        GO TOP
    ENDIF
ELSE
    IF RECNO() >= RECCOUNT()
        SKIP - 18
    ENDIF
ENDIF
ENDDO
CLEAR
@ 01, 00 TO 24, 79 DOUBLE
SET COLOR TO /W

```



```

@ 02,02 SAY "RECNO"
@ 02,COL()+2 SAY " MACHINE "
@ 02,COL()+2 SAY "  MAKER  "
@ 02,COL()+2 SAY "  MODEL  "
@ 02,COL()+2 SAY "  TYPE   "
@ 02,COL()+2 SAY "PRICE(won)"
SET COLOR TO /W/B
@ 03,01 TO 03,78
@ 22,01 TO 22,78
  SET FILTER TO &FIDNA = TRIM(m_maker)
  GO TOP
  IF (EOF())
    SET COLOR TO /W
    @ 12,30 SAY ' 찾는 자료가 없음 ... '
    SET COLOR TO /W/B
    DO CONTI
  ENDIF
DO WHILE .NOT.EOF()
  rowno = 4
  @ 04,01 CLEAR TO 21,78
  DO WHILE .NOT. (EOF() .OR. rowno = 22)
  * SET FILTER TO MAKER = TRIM(m_maker)
    @ rowno,02 SAY STR(RECNO(),5)
    @ rowno,COL()+2 SAY MACHINE
    @ rowno,COL()+2 SAY MAKER
    @ rowno,COL()+2 SAY MODEL
    @ rowno,COL()+2 SAY TYPE
    @ rowno,COL()+2 SAY STR(PRICE_WON,10)
    rowno = rowno + 1
    SKIP
    IF MOD(rowno,5) = 0
      SET COLOR TO /W
    ELSE
      SET COLOR TO /W/B
    ENDIF
  ENDDO
  SET COLOR TO /G
  @ 23,02 SAY "[PgUp/PgDn]"
  SET COLOR TO /W/B
  @ 23,COL() SAY " : 상/하 화면 레코드 데이터 표시 "
  SET COLOR TO /G
  @ 23,COL()+1 SAY "[Enter]"

```

```

SET COLOR TO /W/B
@ 23,COL() SAY " : 레코드 확인 작업 종료 "
keyin = INKEY()
DO WHILE .NOT. (keyin=13 .OR. keyin=18 .OR. keyin=3)
    keyin = INKEY()
ENDDO
IF keyin = 13
    EXIT
* LOOP
ENDIF
IF keyin = 18
    IF RECNO() >= 36
        SKIP - (18 + rowno - 4)
    ELSE
        GO TOP
    ENDIF
ELSE
    IF RECNO() >= RECCOUNT()
        SKIP - 18
    ENDIF
ENDIF
ENDDO
CLEAR
@ 01,00 TO 24,79 DOUBLE
SET COLOR TO /W
@ 02,02 SAY "RECNO"
@ 02,COL()+2 SAY " MACHINE "
@ 02,COL()+2 SAY "  MAKER  "
@ 02,COL()+2 SAY "  MODEL  "
@ 02,COL()+2 SAY "  TYPE   "
@ 02,COL()+2 SAY "PRICE(won)"
SET COLOR TO /W/B
@ 03,01 TO 03,78
@ 22,01 TO 22,78
SET FILTER TO &FIDNA = TRIM(m_maker)
GO TOP
IF (EOF())
    SET COLOR TO /W
    @ 12,30 SAY ' 찾는 자료가 없음 ... '
    SET COLOR TO /W/B
    DO CONTI
ENDIF

```

```

DO WHILE .NOT.EOF()
  rowno = 4
  @ 04,01 CLEAR TO 21,78
  DO WHILE .NOT. (EOF() .OR. rowno = 22)
  * SET FILTER TO MAKER = TRIM(m_maker)
    @ rowno,02 SAY STR(RECNO(),5)
    @ rowno,COL()+2 SAY MACHINE
    @ rowno,COL()+2 SAY MAKER
    @ rowno,COL()+2 SAY MODEL
    @ rowno,COL()+2 SAY TYPE
    @ rowno,COL()+2 SAY STR(PRICE_WON,10)
    rowno = rowno + 1
    SKIP
    IF MOD(rowno,5) = 0
      SET COLOR TO /W
    ELSE
      SET COLOR TO /W/B
    ENDIF
  ENDDO
  SET COLOR TO /G
  @ 23,02 SAY "[PgUp/PgDn]"
  SET COLOR TO /W/B
  @ 23,COL() SAY " : 상/하 화면 레코드 데이터 표시 "
  SET COLOR TO /G
  @ 23,COL()+1 SAY "[Enter]"
  SET COLOR TO /W/B
  @ 23,COL() SAY " : 레코드 확인 작업 종료 "
  keyin = INKEY()
  DO WHILE .NOT. (keyin=13 .OR. keyin=18 .OR. keyin=3)
    keyin = INKEY()
  ENDDO
  IF keyin = 13
    EXIT
  * LOOP
  ENDIF
  IF keyin = 18
    IF RECNO() >= 36
      SKIP - (18 + rowno - 4)
    ELSE
      GO TOP
    ENDIF
  ELSE
  
```

```

        IF RECNO() >= RECCOUNT()
            SKIP - 18
        ENDIF
    ENDIF
ENDDO
CLEAR
@ 01,00 TO 24,79 DOUBLE
SET COLOR TO /W
@ 02,02 SAY "RECNO"
@ 02,COL()+2 SAY " MACHINE "
@ 02,COL()+2 SAY "  MAKER  "
@ 02,COL()+2 SAY "  MODEL  "
@ 02,COL()+2 SAY "  TYPE   "
@ 02,COL()+2 SAY "PRICE(won)"
SET COLOR TO /W/B
@ 03,01 TO 03,78
@ 22,01 TO 22,78
    SET FILTER TO &FIDNA = TRIM(m_maker)
    GO TOP
    IF (EOF())
        SET COLOR TO /W
        @ 12,30 SAY ' 찾는 자료가 없음 ... '
        SET COLOR TO /W/B
        DO CONTI
    ENDIF
DO WHILE .NOT.EOF()
    rowno = 4
    @ 04,01 CLEAR TO 21,78
    DO WHILE .NOT. (EOF() .OR. rowno = 22)
*   SET FILTER TO MAKER = TRIM(m_maker)
        @ rowno,02 SAY STR(RECNO(),5)
        @ rowno,COL()+2 SAY MACHINE
        @ rowno,COL()+2 SAY MAKER
        @ rowno,COL()+2 SAY MODEL
        @ rowno,COL()+2 SAY TYPE
        @ rowno,COL()+2 SAY STR(PRICE_WON,10)
        rowno = rowno + 1
        SKIP
        IF MOD(rowno,5) = 0
            SET COLOR TO /W
        ELSE
            SET COLOR TO /W/B

```

```

    ENDIF
  ENDDO
  SET COLOR TO /G
  @ 23,02 SAY "[PgUp/PgDn]"
  SET COLOR TO /W/B
  @ 23,COL() SAY " : 상/하 화면 레코드 데이터 표시 "
  SET COLOR TO /G
  @ 23,COL()+1 SAY "[Enter]"
  SET COLOR TO /W/B
  @ 23,COL() SAY " : 레코드 확인 작업 종료 "
  keyin = INKEY()
  DO WHILE .NOT. (keyin=13 .OR. keyin=18 .OR. keyin=3)
    keyin = INKEY()
  ENDDO
  IF keyin = 13
    EXIT
  * LOOP
  ENDIF
  IF keyin = 18
    IF RECNO() >= 36
      SKIP - (18 + rowno - 4)
    ELSE
      GO TOP
    ENDIF
  ELSE
    IF RECNO() >= RECCOUNT()
      SKIP - 18
    ENDIF
  ENDIF
ENDDO

  DO WHILE .T.
    @ 4,10 CLEAR TO 20,70
    @ 4,10 TO 20,70 DOUBLE
    @ 5,30 SAY [2. 전작용 기계]
    @ 6,11 TO 6,69 DOUBLE
    @ 9,30 SAY [21. 경운작업]
    @ 10,30 SAY [22. 정지작업]
    @ 11,30 SAY [23. 복토 및 배토작업]
    @ 12,30 SAY [24. 퇴비 및 비료살포]
    @ 13,30 SAY [25. 방제 및 제초작업]
    @ 14,30 SAY [26. 수확작업]
    @ 18,30 SAY '20. Return to MAIN MENU'

```



```

STORE 0 TO selectnum
SET COLOR TO /R/W
@ 20,33 SAY " 선택 2 "
SET COLOR TO /W/B
@ 20,42 GET selectnum PICTURE "9" RANGE 0,6
READ
DO CASE
  CASE selectnum = 0
    EXIT
  CASE selectnum = 1
    DO WHILE .T.
      @ 6,17 CLEAR TO 20,64
      @ 6,17 TO 20,64
      @ 7,30 SAY [21. 경운작업]
      @ 8,18 TO 8,63
      @ 10,24 SAY [211. 트랙터 (Riding tractor)]
      @ 11,24 SAY [212. 경운기 (Power tiller)]
      @ 12,24 SAY [213. 트랙터플라우 (Tractor plow)]
      @ 13,24 SAY [214. 경운기플라우 (Power tiller plow)]
      @ 18,33 SAY '215. Return'
    STORE 5 TO selectnum
    SET COLOR TO /R/W
    @ 20,33 SAY " 선택 21 "
    SET COLOR TO /W/B
    @ 20,43 GET selectnum PICTURE "9" RANGE 1,5
    READ
    DO CASE
      CASE selectnum = 5
        EXIT
        CASE selectnum = 1
          @ 9,19 CLEAR TO 19,63
          SET COLOR TO /W
          @ 12,24 SAY [211. 트랙터 (Riding tractor)]
          SET COLOR TO /W/B
          DO TRAC001
        LOOP
        CASE selectnum = 2
          @ 9,19 CLEAR TO 19,63
          SET COLOR TO /W

```

DBFNAME = 'RICETR'  
 TITLE = [보행형이앙기 (Walking type)]  
 FORM = 'RICETR.FMT'

```

DO STRUCT
CLEAR MEMORY
CLOSE ALL
CLEAR ALL
RETURN

```

```

    @ 12,24 SAY [212. 경운기 (Power tiller)]
SET COLOR TO /W/B
DO TILL001
LOOP
CASE selectnum = 3
@ 9,19 CLEAR TO 19,63
SET COLOR TO /W
@ 12,24 SAY [213. 트랙터플라우 (Tractor plow)]
SET COLOR TO /W/B
DO TRPL001
LOOP
CASE selectnum = 4
@ 9,19 CLEAR TO 19,63
SET COLOR TO /W
@ 12,24 SAY [214. 경운기플라우 (Power tiller
plow)]
SET COLOR TO /W/B
DO PTPLO01
LOOP
ENDCASE
DO CONTI
ENDDO
LOOP
CASE selectnum = 2
DO WHILE .T.
@ 6,17 CLEAR TO 20,64
@ 6,17 TO 20,64
@ 7,30 SAY [22. 정지작업]
@ 8,18 TO 8,63
@ 10,24 SAY [221. 로타베이터 (Tractor rotavator)]
@ 11,24 SAY [222. 경운기로타리 (Power tiller rotary)]
@ 18,33 SAY '223. Return'
STORE 3 TO selectnum
SET COLOR TO /R/W
@ 20,33 SAY " 선택 22 "
SET COLOR TO /W/B
@ 20,43 GET selectnum PICTURE "9" RANGE 1,3

```

```

READ
  DO CASE
    CASE selectnum = 3
      EXIT
      CASE selectnum = 1
        @ 9,19 CLEAR TO 19,63
        SET COLOR TO /W
        @ 12,24 SAY [221. 로타베이터 (Tractor
rotavator)]

        SET COLOR TO /W/B
        DO ROVA001
      LOOP
      CASE selectnum = 2
        @ 9,19 CLEAR TO 19,63
        SET COLOR TO /W
        @ 12,24 SAY [222. 경운기로타리 (Power tiller
rotary)]

        SET COLOR TO /W/B
        DO RORY001
      LOOP
    ENDCASE
  DO CONTI
  ENDDO
LOOP
CASE selectnum = 3
  DO NODATA
CASE selectnum = 4
  DO NODATA
CASE selectnum = 5
  DO NODATA
CASE selectnum = 6
  DO NODATA
ENDCASE
DO CONTI
ENDDO
* EOF: RICE2.PRG
* RICE0.PRG
  DO WHILE .T.
    @ 4,10 CLEAR TO 20,70
    @ 4,10 TO 20,70 DOUBLE
    @ 5,30 SAY [1. 수도작용 기계]
    @ 6,11 TO 6,69 DOUBLE

```

```

    ⑩ 9,30 SAY [11. 경운작업]
    ⑪ 10,30 SAY [12. 정지작업]
    ⑫ 11,30 SAY [13. 이앙작업]
    ⑬ 12,30 SAY [14. 퇴비 및 비료살포]
    ⑭ 13,30 SAY [15. 방제 및 제초작업]
    ⑮ 14,30 SAY [16. 수확작업]
    ⑯ 18, 30 SAY '10. Return to MAIN MENU'
STORE 0 TO selectnum
SET COLOR TO /R/W
    ⑰ 20,33 SAY " 선 택 1 "
SET COLOR TO /W/B
    ⑱ 20,42 GET selectnum PICTURE "9" RANGE 0,6
READ
DO CASE
CASE selectnum = 0
EXIT
CASE selectnum = 1
DO WHILE .T.
    ① 6,17 CLEAR TO 20,64
    ② 6,17 TO 20,64
    ③ 7,30 SAY [11. 경운작업]
    ④ 8,18 TO 8,63
    ⑤ 10,24 SAY [111. 트랙터 (Riding tractor)]
    ⑥ 11,24 SAY [112. 경운기 (Power tiller)]
    ⑦ 12,24 SAY [113. 트랙터플라우 (Tractor plow)]
    ⑧ 13,24 SAY [114. 경운기플라우 (Power tiller plow)]
    ⑨ 18,33 SAY '115. Return'
STORE 5 TO selectnum
SET COLOR TO /R/W
    ⑩ 20,33 SAY " 선 택 11 "
SET COLOR TO /W/B
    ⑪ 20,43 GET selectnum PICTURE "9" RANGE 1,5
READ
DO CASE
CASE selectnum = 5
EXIT
CASE selectnum = 1
    ① 9,19 CLEAR TO 19,63
SET COLOR TO /W
    ② 12,24 SAY [111. 트랙터 (Riding tractor)]
SET COLOR TO /W/B
DO TRAC001

```

```

        LOOP
        CASE selectnum = 2
        @ 9,19 CLEAR TO 19,63
        SET COLOR TO /W
        @ 12,24 SAY [112. 경운기 (Power tiller)]
        SET COLOR TO /W/B
        DO TILL001
        LOOP
        CASE selectnum = 3
        @ 9,19 CLEAR TO 19,63
        SET COLOR TO /W
        @ 12,24 SAY [113. 트랙터플라우 (Tractor plow)]
        SET COLOR TO /W/B
        DO TRPL001
        LOOP
        CASE selectnum = 4
        @ 9,19 CLEAR TO 19,63
        SET COLOR TO /W
        @ 12,24 SAY [114. 경운기플라우 (Power tiller
plow)]
        SET COLOR TO /W/B
        DO PTPL001
        LOOP
        ENDCASE
        DO CONTI
        ENDDO
        LOOP
        CASE selectnum = 2
        DO WHILE .T.
        @ 6,17 CLEAR TO 20,64
        @ 6,17 TO 20,64
        @ 7,30 SAY [12. 정지작업]
        @ 8,18 TO 8,63
        @ 10,24 SAY [121. 로타베이터 (Tractor rotavator)]
        @ 11,24 SAY [122. 경운기로타리 (Power tiller rotary)]
        @ 18,33 SAY '123. Return'
        STORE 3 TO selectnum
        SET COLOR TO /R/W
        @ 20,33 SAY " 선택 12 "
        SET COLOR TO /W/B
        @ 20,43 GET selectnum PICTURE "9" RANGE 1,3
        READ

```



```

DO CASE
CASE selectnum = 3
EXIT
CASE selectnum = 1
@ 9,19 CLEAR TO 19,63
SET COLOR TO /W
@ 12,24 SAY [121. 로타베이터 (Tractor
rotavator)]

SET COLOR TO /W/B
DO ROVA001
LOOP
CASE selectnum = 2
@ 9,19 CLEAR TO 19,63
SET COLOR TO /W
@ 12,24 SAY [122. 경운기로타리 (Power tiller
rotary)]

SET COLOR TO /W/B
DO RORY001
LOOP
ENDCASE
DO CONTI
ENDDO
LOOP
CASE selectnum = 3
DO WHILE .T.
@ 6,17 CLEAR TO 20,64
@ 6,17 TO 20,64
@ 7,30 SAY [13. 이양작업]
@ 8,18 TO 8,63
@ 10,24 SAY [131. 승용형이양기 (Riding type)]
@ 11,24 SAY [132. 보행형이양기 (Walking type)]
@ 18,33 SAY '133. Return'
STORE 3 TO selectnum
SET COLOR TO /R/W
@ 20,33 SAY " 선택 13 "
SET COLOR TO /W/B
@ 20,43 GET selectnum PICTURE "9" RANGE 1,3
READ
DO CASE
CASE selectnum = 3
EXIT
CASE selectnum = 1

```

```

DO NODATAS
CASE selectnum = 2
@ 9,19 CLEAR TO 19,63
SET COLOR TO /W
@ 12,24 SAY [132. 보행형이앙기 (Walking type)]
SET COLOR TO /W/B
DO RITR001
LOOP
ENDCASE
DO CONTI
ENDDO
LOOP
CASE selectnum = 4
DO NODATA
CASE selectnum = 5
DO NODATA
CASE selectnum = 6
DO WHILE .T.
@ 6,17 CLEAR TO 20,64
@ 6,17 TO 20,64
@ 7,30 SAY [16. 수확작업]
@ 8,18 TO 8,63
@ 10,24 SAY [161. 바인더 (Binder)]
@ 11,24 SAY [162. 보통형콤바인 (combine)]
@ 12,24 SAY [163. 자탈형콤바인 (Head-feeding)]
@ 18,33 SAY '164. Return'
STORE 4 TO selectnum
SET COLOR TO /R/W
@ 20,33 SAY " 선택 16 "
SET COLOR TO /W/B
@ 20,43 GET selectnum PICTURE "9" RANGE 1,4
READ
DO CASE
CASE selectnum = 4
EXIT
CASE selectnum = 1
@ 9,19 CLEAR TO 19,63
SET COLOR TO /W
@ 12,24 SAY [161. 바인더 (Binder)]
SET COLOR TO /W/B
DO BIND001
LOOP

```

```

CASE selectnum = 2
DO NODATAS
CASE selectnum = 3
@ 9,19 CLEAR TO 19,63
SET COLOR TO /W
@ 12,24 SAY [163. 자탈형콤바인 (Head-feeding)]
SET COLOR TO /W/B
DO COMB001
LOOP
ENDCASE
DO CONTI
ENDDO

LOOP
ENDCASE
DO CONTI
ENDDO

* EOF: RICE0.PRG
DBFNAME = 'ROTARY'
TITLE = [경운기로타리 (Power tiller rotary)]
FORM = 'ROTARY.FMT'
DO STRUCT
CLEAR MEMORY
CLOSE ALL
CLEAR ALL
RETURN
DBFNAME = 'ROTAVA'
TITLE = [로타베이터 (Tractor rotavator)]
FORM = 'ROTAVA.FMT'
DO STRUCT
CLEAR MEMORY
CLOSE ALL
CLEAR ALL
RETURN
@ 1, 18 SAY "# Ctrl + U: 레코드에 삭제[Del]표시/해제 한다 #"
@ 4, 3 SAY "총 레코드"
@ 4, 68 SAY "PgUp:"
@ 5, 5 SAY "갯수 :"
@ 5, 68 SAY "전 레코드"
@ 6, 71 SAY "이동"
@ 8, 68 SAY "PgDn:"
@ 9, 68 SAY "다음레코드"
@ 10, 3 SAY "현 레코드"

```

```

⑩ 10, 71 SAY "이동"
⑪ 11, 5 SAY "번호 :"
⑫ 12, 68 SAY "Ctrl+End"
⑬ 13, 68 SAY "/Ctrl + W:"
⑭ 14, 68 SAY "작업종료"
⑯ 16, 3 SAY "필드이동:"
⑰ 16, 68 SAY "ALT+SPACE:"
⑱ 17, 7 SAY CHR(24)
⑲ 17, 68 SAY "한/영전환"
㉘ 18, 4 SAY CHR(27)+SPACE(5)+CHR(26)
㉙ 19, 7 SAY CHR(25)
㉚ 23, 31 SAY '# 입력 / 수정 작업 #'
    ⑦ 7, 1 SAY STR(RECCOUNT( ))
㉛ 13, 1 SAY STR(RECNO( ))
    ② 2, 15 TO 22, 64 DOUBLE
    ⑥ 6, 16 TO 6, 63
    ③ 3, 66 TO 22, 78 DOUBLE
    ③ 3, 1 TO 8, 13 DOUBLE
    ⑨ 9, 1 TO 14, 13 DOUBLE
    ⑮ 15, 1 TO 22, 13 DOUBLE
* SEARCH.PRG
    DO WHILE .T.
        CLEAR
        USE &DBFNAME
        ② 2,12 TO 16,68 DOUBLE
        SET COLOR TO /W
        ③ 3,24 SAY ' '+TITLE+' '
        SET COLOR TO /W/B
        ④ 4,13 TO 4,67 DOUBLE
    *    ⑤ 5,13 CLEAR TO 13,67
        ⑤ 5,30 SAY [4. 조 회]
        ⑥ 6,13 TO 6,67
        ⑧ 8,31 SAY "1. 조건 없음"
        ⑩ 10,31 SAY "2. 제조회사 이름으로"
        ⑪ 11,31 SAY "3. 모델 이름으로"
        ⑭ 14,34 SAY '0. Return'
        STORE 0 TO selectnum
        SET COLOR TO /R/W
        ⑮ 16,33 SAY " 선택 "
        SET COLOR TO /B/W
        ㉚ 23,31 SAY ' 조회 조건 선택 '
        SET COLOR TO /W/B

```

```

    @ 16,43 GET selectnum PICTURE "9" RANGE 0,3
READ
    @ 7,13 CLEAR TO 14,67
DO CASE
    CASE selectnum = 0
        @ 23,00
        SET FILTER TO
        EXIT
    CASE selectnum = 1
        SET FILTER TO
        DO RESULT
        CLOSE ALL
    CASE selectnum = 2
        @ 11,28 SAY '[ENTER] : 모든 데이터 출력 '
        @ 13,28 SAY '[shft+SPACE] : 한/영 전환'
        STORE SPACE(10) TO m_maker
        @ 9,25 SAY '2. 제조회사 이름은 : ' GET m_maker
        READ
        FIDNA = 'MAKER'
        DO RESULT_2
        CLOSE ALL
    CASE selectnum = 3
        @ 11,28 SAY '[ENTER] : 모든 데이터 출력 '
        @ 13,28 SAY '[shft+SPACE] : 한/영 전환'
        STORE SPACE(13) TO m_maker
        @ 9,25 SAY '3. 모델 이름은 : ' GET m_maker
        READ
        FIDNA = 'MODEL'
        DO RESULT_2
        CLOSE ALL
    ENDCASE
ENDDO T
    * Program..: STRUCT.PRG
* Author...: KHU
* Date.....: 92.08.31
* Notice...: Copyright (c) 1992, KHU, All Rights Reserved
* Notes....:
* Reserved.: selectnum
*
    SET TALK OFF
SET HELP OFF
SET MENU OFF

```



```

SET SAFETY OFF
SET BELL OFF
SET STATUS OFF
SET ESCAPE OFF
SET CONFIRM ON
  USE &DBFNAME
  DO WHILE .T.
    * ---Display menu options, centered on the screen
    *   draw menu border and print heading
  CLEAR
  USE &DBFNAME
    @ 2,12 TO 16,68 DOUBLE
  SET COLOR TO /W
    @ 3,24 SAY '   '+TITLE+'   '
  SET COLOR TO /W/B
    @ 4,13 TO 4,67 DOUBLE
    * ---display detail lines
    @ 6,30 SAY [1.   입           력]
    @ 7,30 SAY [2.   수           정]
    @ 8,30 SAY [3.   삭           제]
    @ 9,30 SAY [4.   조           회]
    @ 10,30 SAY [5.  출           력]
    @ 13,34 SAY '0. Return'
  STORE 0 TO selectnum
  SET COLOR TO /R/W
    @ 16,33 SAY " 선   택   "
  SET COLOR TO /B/W
    @ 23,31 SAY ' 데이터 처리 작업 '
  SET COLOR TO /W/B
    @ 16,43 GET selectnum PICTURE "9" RANGE 0,5
  READ
  DO CASE
    CASE selectnum = 0
      @ 23,00
      CLEAR MEMORY
      CLOSE ALL
      CLEAR ALL
      RETURN
    CASE selectnum = 1
    * DO ADD INFORMATION
      SET FORMAT TO &FORM
      APPEND

```

```

SET FORMAT TO
DO CONTI
CASE selectnum = 2
* DO CHANGE INFORMATION
  @ 5,13 CLEAR TO 13,67
  @ 5,30 SAY [2. 수          정]
  @ 6,13 TO 6,67
  editrec = 0
DO WHILE .NOT.(editrec>=1 .AND. editrec<=RECCOUNT( ))
  SET COLOR TO /W
  @ 11,33 SAY ' ( 1 - '+STR(RECCOUNT(),3)+' ) '
  @ 10,20 SAY '조회 후 수정 대상 레코드 번호 입력 ->'
  SET COLOR TO /W/B
  @ 10,COL( )+1 GET editrec PICT "999" RANG 1,RECCOUNT(
)

      READ
ENDDO
GO editrec
  SET FORMAT TO &FORM
EDIT
SET FORMAT TO
DO CONTI
CASE selectnum = 3
* DO REMOVE INFORMATION
  @ 5,13 CLEAR TO 13,67
  @ 5,30 SAY [3. 삭          제]
  @ 6,13 TO 6,67
  STORE ' ' TO select
SET COLOR TO /W
  @ 10,15 SAY '삭제 표시[Del]를 지정한 레코드를 제거할까요?'
(Y/n)'

SET COLOR TO /W/B
  @ 10,COL( )+1 GET select PICT "A"
READ
IF UPPER(select) <> 'N'
*
  SET TALK ON
*
  CLEAR
SET COLOR TO /R/W
  @ 10,15 SAY '삭제 표시[Del]를 지정한 레코드를 파일에서 제
거...

SET COLOR TO /W/B
PACK

```

```

*      SET TALK OFF
      ENDIF
      DO CONTI
      LOOP
      CASE selectnum = 4
*      DO REVIEW INFORMATION
      DO SEARCH
      CASE selectnum = 5
*      PRINT
      USE &DBFNAME
      GO BOTTOM
      plr = RECNO()
      nextrec = "SKIP"
      GO TOP
      DO PRINT
      ENDCASE
      ENDDO T
RETURN
* EOF: STRUCT.PRG
  DBFNAME = 'TILL'
  TITLE = [경운기 (Power tiller)]
  FORM = 'TILL.FMT'
  DO STRUCT
  CLEAR MEMORY
  CLOSE ALL
  CLEAR ALL
  RETURN
      DO WHILE .T.
      @ 4,10 CLEAR TO 20,70
      @ 4,10 TO 20,70 DOUBLE
      @ 5,24 SAY [2. 경운정지용기계(Tillage machinery)]
      @ 6,11 TO 6,69 DOUBLE
      @ 8,24 SAY [21. 플라우 (Plow)]
      @ 9,24 SAY [22. 로타리 (Rotary)]
      @ 10,24 SAY [23. 쇄토기 (Pulverizer)]
      @ 11,24 SAY [24. 중경, 제초 (Cultivation)]
      @ 12,24 SAY [25. 휴립, 배토기(Ridger, Fallow opener)]
      @ 13,24 SAY [26. 답압기 (Press)]
      @ 14,24 SAY [27. 비닐피복기 (Vinyl mulcher)]
      @ 15,24 SAY [28. 트랜처 (Trencher)]
      @ 18, 26 SAY '29. Return to MAIN MENU'
      STORE 9 TO selectnum

```

```

SET COLOR TO /R/W
@ 20,33 SAY " 선택 2 "
SET COLOR TO /W/B
@ 20,42 GET selectnum PICTURE "9" RANGE 1,9
READ
DO CASE
CASE selectnum = 9
EXIT
CASE selectnum = 1
DO WHILE .T.
@ 6,17 CLEAR TO 20,64
@ 6,17 TO 20,64
@ 7,24 SAY [21. 플라우 (Plow)]
@ 8,18 TO 8,63
@ 10,24 SAY [211. 트랙터플라우 (Tractor plow)]
@ 11,24 SAY [212. 경운기플라우 (Power tiller plow)]
@ 18,33 SAY '213. Return'
STORE 3 TO selectnum
SET COLOR TO /R/W
@ 20,33 SAY " 선택 21 "
SET COLOR TO /W/B
@ 20,43 GET selectnum PICTURE "9" RANGE 1,3
READ
DO CASE
CASE selectnum = 3
EXIT
CASE selectnum = 1
@ 9,19 CLEAR TO 19,63
SET COLOR TO /W
@ 12,24 SAY [211. 트랙터플라우 (Tractor plow)]
SET COLOR TO /W/B
DO TRPL001
LOOP
CASE selectnum = 2
@ 9,19 CLEAR TO 19,63
SET COLOR TO /W
@ 12,24 SAY [212. 경운기플라우 (Power tiller
plow)]

SET COLOR TO /W/B
DO PTPL001
LOOP
ENDCASE

```

```

DO CONTI
ENDDO
LOOP
CASE selectnum = 2
DO WHILE .T.
@ 6,17 CLEAR TO 20,64
@ 6,17 TO 20,64
@ 7,24 SAY [22. 로타리 (Rotary)]
@ 8,18 TO 8,63
@ 10,24 SAY [221. 로타베이터 (Tractor rotavator)]
@ 11,24 SAY [222. 경운기로타리 (Power tiller rotary)]
@ 18,33 SAY '223. Return'
STORE 3 TO selectnum
SET COLOR TO /R/W
@ 20,33 SAY " 선택 22 "
SET COLOR TO /W/B
@ 20,43 GET selectnum PICTURE "9" RANGE 1,3
READ
DO CASE
CASE selectnum = 3
EXIT
CASE selectnum = 1
@ 9,19 CLEAR TO 19,63
SET COLOR TO /W
@ 12,24 SAY [221. 로타베이터 (Tractor
rotavator)]

SET COLOR TO /W/B
DO ROVA001
LOOP
CASE selectnum = 2
@ 9,19 CLEAR TO 19,63
SET COLOR TO /W
@ 12,24 SAY [222. 경운기로타리 (Power tiller
rotary)]

SET COLOR TO /W/B
DO RORY001
LOOP
ENDCASE
DO CONTI
ENDDO
LOOP
CASE selectnum = 3

```



```
DO NODATA
CASE selectnum = 4
DO NODATA
CASE selectnum = 5
DO NODATA
CASE selectnum = 6
DO NODATA
CASE selectnum = 7
DO NODATA
CASE selectnum = 8
DO NODATA
ENDCASE
DO CONTI
ENDDO
* EOF: TILMA2.PRG
```

## 2. 映像處理 PROGRAM LIST

```
/* 고도지도표시 함수*/
```

```
#include <c:\msc\include\malloc.h>
#include <c:\msc\include\stdio.h>
#include <c:\msc\include\stdlib.h>
#include <c:\msc\include\conio.h>
#include <c:\msc\include\math.h>
```

```
#define IN      0
#define OUT     1
#define MAXLIMIT 30
```

```
typedef struct tag {
    float x, y, z;
    int in_out;
```

```
} DATA;
```

```
typedef struct tag2 {
    float x, y, z;
```

```
} POINT;
```

```
typedef struct tag3 {
    POINT n1, n2, n3;
```

```
} DATA2;
```

```
DATA2 data[MAXLIMIT];
```

```
extern ZONE zone[];
```

```
int element = 0, left, right, top, bottom, scan_step = 3;
```

```
float maxalti, minalti;
```

```
void allocation( char *filename, int num);
```

```
float Calc_Altitude(float x, float y, int *no, int num);
```

```
void altimap(void);
```

```
int InteriorTest(int no, float x, float y);
```

```
void altitude(void);
```

```
void altimap(void)
```

```
{
```

```
    FILE *fp, *rp;
```

```
    DATA *dat;
```

```
    int i, j, *r1_id, *r2_id;
```

```

int    cnt, cnt2=-1, cnt3=-1, cnt4=-1, dum1, dum5;
int    inpt_id[2], MAX_ID, MIN_ID, discrete, no;
float  dum2, dum3, dum4;
float  altitude;

if( get_peakpoint() < 0 )
    exit(0);

if((rp = fopen("f:/altixy.dat", "r")) == NULL) {
    printf("\a\a"):    exit(0);
}
cnt = -1;
while( !feof(rp) ) {
    fscanf(rp, "%d %f %f", &dum1, &dum2, &dum3);
    cnt++;
}
rewind(rp);

if((fp = fopen("f:/alti.dat", "r")) == NULL) {
    printf("\a\a"):    fclose(rp);    exit(0);
}
i = 0;
while( !feof(fp) ) {
    fscanf(fp, "%d %f %d", &dum1, &dum2, &dum5);
    i++;
}
rewind(fp);

if( i != cnt ) {
    printf("\a\a\a\a\a\a\a\a"):    fcloseall();    exit(0);
}

if( (dat = (DATA *) malloc(sizeof(DATA) * cnt)) == NULL)    {
    printf("\a\a\a"):    exit(0);
}
if( (r1_id = (int *) malloc(sizeof(int) * cnt)) == NULL) {
    printf("\a\a\a"):    exit(0);
}
if( (r2_id = (int *) malloc(sizeof(int) * cnt)) == NULL) {
    printf("\a\a\a"):    exit(0);
}
}

```

```

for(i=0; i<cnt; i++)
    fscanf(rp, "%d %f %f", &dum1, &dat[i].x, &dat[i].y);

for(i=0; i<cnt; i++)
    fscanf(fp, "%d %f %d", &dum1, &dat[i].z, &dat[i].in_out);
fclose(fp);
fclose(rp);

for(i=0; i<cnt; i++)
    if(dat[i].in_out == IN) {
        cnt2++;
        inpt_id[cnt2] = i+1;
    }

MAX_ID = inpt_id[0]; MIN_ID = inpt_id[1];
if( dat[inpt_id[0]-1].x < dat[inpt_id[1]-1].x ) {
    MAX_ID = inpt_id[1]; MIN_ID = inpt_id[0];
}

for(i=0; i<cnt; i++)
    if(dat[i].x < dat[MAX_ID-1].x && dat[i].in_out == OUT) {
        cnt3++;
        r1_id[cnt3] = i+1;
    }
r1_id[++cnt3] = MAX_ID;

for(i=0; i<=cnt3; i++)
    if(r1_id[i] != i+1) {
        discrete = i;
        break;
    }
for(i=cnt3-1; i>=discrete; i--)
    r1_id[i+1] = r1_id[i];
r1_id[discrete] = MAX_ID;

for(i=0; i<cnt; i++) {
    for(j=0; j<=cnt3; j++)
        if( i == (r1_id[j] - 1) )
            goto skip;
    cnt4++;
    r2_id[cnt4] = i+1;
}

```

```

        skip::
    }
    for(i = cnt4 - 1; i >= 0; i--)
        r2_id[i+1] = r2_id[i];

    if(cnt3 != discrete)
        r2_id[++cnt4] = r1_id[discrete+1];
    else
        r2_id[++cnt4] = r1_id[0];

    r2_id[0] = r1_id[discrete-1];

    if((fp=fopen("f:/altiout.dat", "w")) == NULL) {
        printf("\a\a\a");    exit(0);
    }

    for(i=0; i<cnt3; i++)
        fprintf(fp, "%d %3.2f %3.2f %3.2f %3.2f %3.2f %3.2f %3.2f
%3.2f %3.2f\n",
            ++element,    dat[MIN_ID-1].x,    dat[MIN_ID-1].y,
            dat[MIN_ID-1].z,
            dat[r1_id[i]-1].x,                dat[r1_id[i]-1].y,
            dat[r1_id[i]-1].z,
            dat[r1_id[i+1]-1].x,                dat[r1_id[i+1]-1].y,
            dat[r1_id[i+1]-1].z);

        fprintf(fp, "%d %3.2f %3.2f %3.2f %3.2f %3.2f %3.2f %3.2f
%3.2f\n",
            ++element,    dat[MIN_ID-1].x,    dat[MIN_ID-1].y,
            dat[MIN_ID-1].z,
            dat[r1_id[i]-1].x,                dat[r1_id[i]-1].y,
            dat[r1_id[i]-1].z,
            dat[r1_id[0]-1].x,                dat[r1_id[0]-1].y,
            dat[r1_id[0]-1].z);

    for(i=0; i<cnt4; i++)
        fprintf(fp, "%d %3.2f %3.2f %3.2f %3.2f %3.2f %3.2f %3.2f
%3.2f %3.2f\n",
            ++element,    dat[MAX_ID-1].x,    dat[MAX_ID-1].y,
            dat[MAX_ID-1].z,
            dat[r2_id[i ]-1].x,    dat[r2_id[i ]-1].y,
            dat[r2_id[i ]-1].z,

```



```

                                dat[r2_id[i+1]-1].x,          dat[r2_id[i+1]-1].y,
dat[r2_id[i+1]-1].z);

```

```

    minalti = 100000.0;    left = top = 1000000;
    maxalti = -100000.0;   right = bottom = -1000000;
    for(i=0; i < cnt; i++) {
        if(dat[i].x < left)
            left = dat[i].x;
        if(dat[i].x > right)
            right = dat[i].x;
        if(dat[i].y < top)
            top = dat[i].y;
        if(dat[i].y > bottom)
            bottom = dat[i].y;
        if(dat[i].z < minalti)
            minalti = dat[i].z;
        if(dat[i].z > maxalti)
            maxalti = dat[i].z;
    }

```

```

    fclose(fp);
    free(dat);    free(r1_id);    free(r2_id);
}

```

```

void altitude(void)
{

```

```

    int i, j, no, dum1, dum5;
    float alti;
    allocation( "f:/altiout.dat", element);
    printf("\a");    getch();

```

```

    im_opmode(2, 3);          im_pixblt(3, 0, 0, 511, 479, 2, 0, 0, 0,
1);

```

```

    im_opmode(2, 4);          im_clear(4, 0);
    im_opmode(2, 3);          im_clear(3, 0);

```

```

    im_opmode(2, 6);          im_outpath(6, -1, 0, 0);
    for(j=top; j<bottom; j += scan_step)
        for(i=left; i<right; i += scan_step) {
            im_opmode(0, 2);
            if( (dum5 = (int)im_pixr(i, j) ) < 250) {

```

```

        alti = Calc_Altitude((float)i, (float)j,
&no, element);
        alti
        =
150.0*(alti-minalti)/(maxalti-minalti)+50.0;
        if( alti > 0.0 ) {
            dum1 = (int)alti;
            zone[dum5].sumno++;
            zone[dum5].sumx += (float)dum1;
            zone[dum5].sumxx +=
(float)dum1*(float)dum1;
            im_opmode(0, 3);
            im_pixw(i, j, dum1);
            im_opmode(0, 1);
            im_pixw(i, j, dum1 - 10);
            im_opmode(0, 0);
            im_pixw(i, j, dum1 + 10);
        }
    }
else
    im_pixw(i, j, 255 );
}
}

```

```

void allocation( char *filename, int num)
{
    int i, dum;
    FILE *rp;

    if((rp=fopen(filename, "r")) == NULL || num > MAXLIMIT) {
        printf("\a\a\a\a\a\a\a\a\a\a\a\a\a"); exit(0);
    }
    for(i=0; i<num; i++)
        fscanf(rp, "%d %f %f %f %f %f %f %f %f\n", &dum,
            &data[i].n1.x, &data[i].n1.y, &data[i].n1.z,
            &data[i].n2.x, &data[i].n2.y, &data[i].n2.z,
            &data[i].n3.x, &data[i].n3.y, &data[i].n3.z);

    fclose(rp);
}

```

```

float Calc_Altitude(float x, float y, int *no, int num)

```

```

{
    float phi, Ni, Nj, Nk;
    float xi, yi, xj, yj, xk, yk, ai, aj, ak, bi, bj, bk, ci, cj, ck;
    float area, p1, p2, p3;
    int i, success;

    for(i=0; i<num; i++)
        if( (success = InteriorTest(i, x, y)) == 1)
            break;
    if(success == 0) {
        *no = -1;
        return(0.0);
    }

    *no = i+1;

    xi = data[i].n1.x;      yi = data[i].n1.y;      p1 = data[i].n1.z;
    xj = data[i].n2.x;      yj = data[i].n2.y;      p2 = data[i].n2.z;
    xk = data[i].n3.x;      yk = data[i].n3.y;      p3 = data[i].n3.z;

    ai=xj*yk-xk*yj;        aj=xk*yi-xi*yk;        ak=xi*yj-xj*yi;
    bi=yj-yk;              bj=yk-yi;
bk=yi-yj;
    ci=xk-xj;              cj=xi-xk;
ck=xj-xi;

    area = (ai+aj+ak) / 2.0;

    Ni=(ai+bi*x+ci*y)/(2.0*area);  Nj=(aj+bj*x+cj*y)/(2.0*area);
    Nk=(ak+bk*x+ck*y)/(2.0*area);
    return(Ni*p1+Nj*p2+Nk*p3);
}

int InteriorTest(int no, float x, float y)
{
    float x1, y1, x2, y2, x3, y3;
    float normal1, normal2, dot_product, theta, alpha;

    x1 = data[no].n1.x;      y1 = data[no].n1.y;
    x2 = data[no].n2.x;      y2 = data[no].n2.y;
    x3 = data[no].n3.x;      y3 = data[no].n3.y;

```

```

normal1 = ((x1-x2)*(y3-y2)-(y1-y2)*(x3-x2));

theta = acos(((x1-x2)*(x3-x2)+(y1-y2)*(y3-y2)) /
              (sqrt(pow(x1-x2, 2)+pow(y1-y2, 2))*
sqrt(pow(x3-x2, 2)+pow(y3-y2, 2))));

normal2 = ((x1-x2)*(y-y2)-(y1-y2)*(x-x2));

alpha = acos(((x1-x2)*(x-x2)+(y1-y2)*(y-y2)) /
              (sqrt(pow(x1-x2, 2)+pow(y1-y2, 2))*
sqrt(pow(x-x2, 2)+pow(y-y2, 2))));
dot_product = normal1*normal2;
if(dot_product > 0 && alpha <= theta) {
    x1 = data[no].n2.x;          y1 =
data[no].n2.y;
    x2 = data[no].n3.x;          y2 =
data[no].n3.y;
    x3 = data[no].n1.x;          y3 =
data[no].n1.y;

normal1 = ((x1-x2)*(y3-y2)-(y1-y2)*(x3-x2));

theta = acos(((x1-x2)*(x3-x2)+(y1-y2)*(y3-y2)) /
              (sqrt(pow(x1-x2, 2)+pow(y1-y2, 2))*
sqrt(pow(x3-x2, 2)+pow(y3-y2, 2))));

normal2 = ((x1-x2)*(y-y2)-(y1-y2)*(x-x2));

alpha = acos(((x1-x2)*(x-x2)+(y1-y2)*(y-y2)) /
              (sqrt(pow(x1-x2, 2)+pow(y1-y2, 2))*
sqrt(pow(x-x2, 2)+pow(y-y2, 2))));
dot_product = normal1*normal2;

if(dot_product >= 0 && alpha <= theta)
{
    return(1);
}
}
return(0);
}

```

```

int get_peakpoint( void )
{
    FILE *wp;
    int x,y,st,sw=0,t=0,start=0;
    char t_char[20];
    XYG xyg[30];

    if((wp = fopen("f:/altixy.dat", "w")) == NULL) {
        printf("\a\a");    exit(0);
    }

    opnd_1(2, 2);    im_setcolor(MVP_GREEN);
    draw_exit_box(0, 0); hide_mscursor();
    sprintf( t_char, "%d[%3d, %3d]", t, 0, 0);
    outscreen_msg( 400, 20, t_char, MVP_GREEN );

    do {
        st = getmskey(&x, &y, &sw);
        if( x < 50 && y < 50 )
            break;
        switch( st ) {
            case RKDP : /* right key */
                if( start == 0 || t <= 0 ) {
                    fclose(wp);
                    outscreen_msg( 400, 20, t_char,
MVP_GREEN );
                    draw_exit_box(0, 0);
                    shaw_mscursor();    return(-1);
                }
                outscreen_msg( 400, 20, t_char, MVP_GREEN
);
                t--;
                sprintf( t_char, "%d[%3d, %3d]", t, x, y);
                outscreen_msg( 400, 20, t_char, MVP_GREEN
);
                wpixel(xyg[t].x,xyg[t].y,xyg[t].g);
                break;
            case LKDP : /* left key */
                outscreen_msg( 400, 20, t_char, MVP_GREEN
);
                xyg[t].x = x;            xyg[t].y = y;
                xyg[t].g = rpixel(x,y) + 4;

```



```

        wpixel(x, y, xyg[t].g);
        t++;
    sprintf( t_char, "%d[%3d, %3d]", t, x, y);
        outscreen_msg( 400, 20, t_char, MVP_GREEN
);
        start=1;
        break;
    }
}while(t < 30);

for(x=0;x<t;x++)
    fprintf(wp, "%d %3.2f %3.2f\n", x+1, (float)xyg[x].x,
(float)xyg[x].y );

fclose(wp);
outscreen_msg( 400, 20, t_char, MVP_GREEN );
draw_exit_box(0, 0); shaw_mscursor();
return(t);
}

```

/\* 영상이치와 및 세션화 \*/

```

#include <c:\msc\include\stdio.h>
#include <c:\msc\include\io.h>
#include <c:\msc\include\conio.h>
#include <c:\msc\include\stdlib.h>
#include <c:\msc\include\math.h>

void set_lut(int thresh)
{
    int i;
    unsigned char buf[256];

    for(i = 0; i < thresh; ++i)
        buf[i] = 0x00;
    for(i = thresh; i < 256; ++i)
        buf[i] = i;

    im_wlut(1, 4, 0, 256, buf, NULL, NULL);
    im_wlut(1, 4, 0, 256, NULL, buf, NULL);
}

```

```

        im_wlut(1, 4, 0, 256, NULL, NULL, buf);
    }

void make_binary(int fb, int threshval )
{
    int j, i, sb;
    unsigned char rowbuf[512];

    if (threshval > 255 || threshval < 0 ) {
        printf("\a\a\a\a\a .....");
        exit(0);
    }

    if(fb == 2)
        sb = 3;
    else
        sb = fb;

    im_opmode(2, 0);          im_outpath(0, -1, 0, 0);
    for(j=0;j<480;j++) {
        im_opmode(0, sb);
        im_rowr(0, j, 512, rowbuf);
        for(i=0;i<512;i++)
            if(rowbuf[i] > threshval)
                rowbuf[i] = 0x04;
            else
                rowbuf[i] = 0xff;
        im_roww(0, j, 512, rowbuf);
    }
    im_opmode(0, 0);
    im_roww(0, j, 512, rowbuf);
}
im_opmode(2, sb);
}

void thinfeature(void)
{
    int i, j, np, tp;
    unsigned char gray, flag;

    im_thin(1, 1, 3, 0, 7, 1);
    im_opmode(2, 6); /* thinned = 0,  bkgnd = 255 */
    im_clear(fml, 0);
}

```

```

im_connectivity(fm, fml, 7);
for(i=0;i<no;i++) {
    im_opmode(0, fml);
    if(thpt[i].x + thpt[j].y == 0)
        continue;
    gray = im_pixr(thpt[i].x, thpt[i].y);
    np = tp = 0;
    for(j = 0;j < 8;j++) {
        flag = gray & 0xc0;
        if( flag == 0x80 || flag == 0x40)
            tp++;
        if( (gray & 0x80) == 0x80 ) {
            np++;
            gray <<= 1;
            gray |= 0x01;
        }
        else
            gray <<= 1;
    }
    thpt[i].t = (int)(tp / 2);
}/* for(i=0;i<no;i++) */
}

int ms_key(void)
{
    union inkey { char ch[2];
                  int i;
    } c;
    c.i = _bios_keybrd(0);
    if(c.ch[0]) return (c.ch[0]);
    else      return (c.ch[1]+256);
}

int manual_thresholding(void)
{
    int c, i, fb = 1;
    int thres = 80;

    im_opmode(2, fb);      im_outpath(fb, -1, 0, 0);

    im_slut(1, 4); set_lut(thres);
}

```

```

c = 0;
while (c != CR && c != 3) {
    while( (c = mv_inkey()) == 0 );
    switch(c) {
        case UP : ++thres; break;
        case DOWN : --thres; break;
        case PGUP : thres += 16; break;
        case PGDN : thres -= 16; break;
    }
    if (thres > 255) thres = 255;
    if (thres < 0) thres = 0;
    cprintf("\r [ fb = %d ] Threshold = %3d ", fb, thres);
    set_lut(thres);
}
set_lut(0);
return(thres);
}

void main(int argc, char *argv[])
{
    int fm=1, threshold=80;

    if(im_init(0xd000, 0x300) != 1) {
        printf("Init MVP failed\n"); exit(0);
    }
    im_inmode(3); im_outmode(0); im_chan(0); im_sync(1, 0);
    im_opmode(3,6); im_outpath(6, -1, 0, 0); im_video(1, 1);
    im_cgrab(-1); getch(); im_cgrab(0);

    threshold = manual_thresholding();
    make_binary(fm, threshold);
    im_opmode(2, 1); im_outpath(1, -1, 0, 0);
    thinfeature();

    im_opmode(2, 0); im_clear(0, 0);
    im_opmode(2, 1); im_clear(1, 126);
    im_opmode(2, 6); im_outpath(6, -1, 2, 0); im_opmode(1, 3);
    im_setcolor(255); im_setbcolor(255);
    im_move(256, 10); im_paint(0, 255);
}

```

```
/* 필지구분지정 및 면적 계산 */
```

```
#include <c:\msc\include\stdio.h>
#include <c:\msc\include\math.h>
#include <c:\msc\include\string.h>
#include <c:\msc\include\sys\stat.h>
#include <c:\msc\include\fcntl.h>
#include <c:\msc\include\stdlib.h>
#include <c:\msc\include\conio.h>
#include <c:\msc\include\io.h>
#include <c:\msc\include\time.h>
#include <c:\msc\include\graph.h>

#define      In  168
#define      Ib  84
#define      Ig  52
#define      It  34
#define      Ie  22

int      msxc, msyc, typeno, tgray;

typedef struct {
        unsigned int   xc, yc, type, element;
        unsigned long  area;
        float          sumx, sumxx;
        unsigned long  sumno;
        float          farea, alti;
}ZONE;

ZONE  zone[260];

void opmd_l(int z, int fm);
void draw_box( int far *x, int far *y);
void draw_exit_box( int x0, int y0 );
void draw_ok_box( int x0, int y0 );
int det_type(void);
int fillpolydata(int fm, int gn);
int filling(int fm);
int fillout(int fm, int gn);
void areasavedata(int fm);
```



```

void region_color(void);
int shaw_zonedata(void);

int    in = In, ib = Ib, ig = Ig, it = It, ie = Ie, iq = Ie;
char   msg[5][2] = { "N", "B", "G", "T", "E" };

void opmd_1(int z, int fm)
{
    switch(z) {
        case 1 :      im_opmode(2, 6);      im_outpath(6, -1,
2, 0);
                    im_opmode(1, fm);      break;
        case 2 :      im_opmode(2, 6);      im_outpath(6, -1,
2, 1);
                    im_opmode(1,          2);
im_setbcolor(0); im_drawmode(2);
                    break;
        case 11:      opmd_1(1, 3);          im_setcolor(255);
im_setbcolor(1);
                    im_drawmode(0);          break;
        case 12:      opmd_1(1, 3);          im_setcolor(254);
im_setbcolor(1);
                    im_drawmode(0);          break;
    }
}

void draw_box( int far *x, int far *y)
{
    int i,  xx1,yy1,  xx2,yy2;
    for(i=0; i<=1; i++){
        xx1 = *(x+i)-1; yy1 = *(y+i)-1; xx2 = *(x+i)+1; yy2 =
*(y+i)+1;
        mbox( &xx1, &yy1, &xx2, &yy2);
    }
}

void draw_exit_box( int x0, int y0 )
{
    int x1, y1, col = MVP_RED;

    x1 = x0 + 50;  y1 = y0 + 50;
}

```

```

        mbox( &x0, &y0, &x1, &y1 );    msetcolor(&col);
        mvpscreen(x0 + 12, y0 + 32, "EXIT" );    wait_release();
    }

void draw_ok_box( int x0, int y0 )
{
    int x1, y1, col = MVP_RED;

    x1 = x0 + 50;    y1 = y0 + 50;
    mbox( &x0, &y0, &x1, &y1 );    msetcolor(&col);
    mvpscreen(x0 + 12, y0 + 32, "O.K." );    wait_release();
}

int det_type(void)
{
    int  x, y, st, sw = 0, flag=0;

    typeno = 4;
    msxc = x = 200;    msyc = y = 40;

    opnd_1(2, 2);
    draw_exit_box(0, 0);    draw_ok_box(100, 0);
    outscreen_msg(msxc, msyc, msg[typeno], MVP_RED);
    while(1) {
        msxc = x;    msyc = y;
        st = getmskey(&x, &y, &sw);
        outscreen_msg(msxc, msyc, msg[typeno], MVP_RED);

        if( x > 100 && x < 100 + 50 && y < 50 ) {
            draw_exit_box(0,0);    draw_ok_box(100, 0);
            if(flag == 0)
                return(-1);
            switch(typeno) {
                case 0 : tgray = in;    in += 1;
break:
                case 1 : tgray = ib;    ib += 1;
break:
                case 2 : tgray = ig;    ig += 1;
break:
                case 3 : tgray = it;    it += 1;
break:
                case 4 : tgray = ie;    ie += 1;

```

```

break:
    }
    return(tgray);
}
if( x < 50 && y < 50 ) {
    draw_exit_box(0, 0); draw_ok_box(100, 0);
    return(-1);
}
flag = 1;
switch(st) {
    case LKDP :          typeno = (typeno + 1) % 5;
break:
    case RKDP : typeno = (typeno + 4) % 5; break;
}
    outscreen_msg(x, y, msg[typeno], MVP_RED);
}
}

int fillpolydata(int fm, int gn)
{
    opnd_1(1, 3);  im_setcolor(gn);          im_setbcolor(gn);
    im_move(msxc, msyc);  im_paint(0, 255);
}

int filling(int fm)
{
    opnd_1(2, 2);  hide_mscursor();
    tgray = det_type();
    opnd_1(1, 2);  shaw_mscursor();

    if(tgray == -1)
        return(0);
    fillpolydata(fm, tgray);

    return(1);
}

void areasavedata(int fm)
{
    int          i, j;
    unsigned long hist[258];
    char         msg1[30];

```

```

FILE          *wp;
static int    ii = 0;
float        areaf;

static struct {
    float  tl, mn;
    int    no, il;
}tmp[5];

if( ( wp = fopen("f:q.dat", "at" ) ) == NULL ) {
    printf("\a\a\a\a\a\a");
    exit(0);
}

zone[tgray].xc = msxc;  zone[tgray].yc = msyc;
zone[tgray].type = typeno;

opmd_1(1, 3);  im_opmode(2, fm);      im_histo(fm, hist);
opmd_1(2, 2);
zone[tgray].area = hist[tgray+2];
areaf = zone[tgray].farea = (float)zone[tgray].area * AREA;

fprintf(wp, "(%3d) (%3d,%3d) [%d] @%3d %10ld=>%10.2f\n",
        ii++, msxc, msyc, typeno, tgray,
zone[tgray].area, areaf);

sprintf(msg1, "%3d[%s]%8.0f", ii, msg[typeno], areaf);
outscreen_msg( 370, ii*12+14, msg1, MVP_GREEN);
sprintf(msg1, "%3d", ii); outscreen_msg(msxc-20, msyc+4, msg1,
MVP_RED);
fclose(wp);

tmp[0].il = In; tmp[1].il = Ib; tmp[2].il = Ig;
tmp[3].il = It;  tmp[4].il = Ie;

if( ii != 1 )
    for(i = 0; i < 5; i++) {
        sprintf(msg1, "%2d          %s          %10.0f
%8.0f", tmp[i].no, msg[i], tmp[i].tl, tmp[i].mn);
        outscreen_msg(10, 400+12*i, msg1, i+2);
    }

```

```

i = 250;
for( j = 0; j < 5; j++) {
    tmp[j].no = 0; tmp[j].tl = tmp[j].mn = 0.0;
    for( ; i >= tmp[j].il; i--)
        if(hist[i+2] != 0L) {
            ++tmp[j].no;
            tmp[j].tl += (float)hist[i+2] * AREA;
        }
    if( tmp[j].no > 0 )
        tmp[j].mn = tmp[j].tl / (float)tmp[j].no;
}

for(i = 0; i < 5; i++) {
    sprintf(msg1, "%2d          %s          %10.0f",
%8.0f", tmp[i].no, msg[i], tmp[i].tl, tmp[i].mn);
    outscreen_msg(10, 400 + 12*i, msg1, i+2);
}
}

void region_color(void)
{
    int x, y, sw = 0;
    int fm = 3, flag = 1;

    while(flag) {
        if( filling(fm) )
            areasavedata(fm);

        opnd_1(2, 2); hide_mscursor();
        draw_exit_box(0,0); getmskey(&x, &y, &sw);
draw_exit_box(0,0);
        if(x < 50 && y < 50) flag = 0;
        shaw_mscursor();
    }
}

int fillout(int fm, int gn)
{
    im_opmode(2, 0); im_clear(0, 4);
    im_opmode(2, 1); im_clear(1, 4);
    opnd_1(1, 3); im_setcolor(254); im_setbcolor(254);
    im_move(256, 10); im_paint(0, 254);
}

```



```

}

int shaw_zonedata(void)
{
    FILE *wp;
    int i;

    if( (wp = fopen("f:/altieach.dat", "w") ) == NULL ) {
        printf("\a\a\a\a");    exit(0);
    }

    for(i=0; i<256;i++) {
        zone[i].alti = zone[i].sumx / zone[i].sumno;
        if( zone[i].alti > 0.0 )
            fprintf(wp,      "(%3d)    [%s]    %3.2f",    i,
msg[zone[i].type], zone[i].alti);
    }
}

```

/\* 영상처리장치를 이용한 각종 그래픽 및 작업환경조성 \*/

```
#include <c:\msc\include\stdio.h>
#include <c:\msc\include\string.h>
#include <c:\msc\include\dos.h>

#define ON          1
#define OFF        0

#define _MAX_X 512
#define MVP_RED      4
#define MVP_GREEN    2
#define MVP_BLUE     1
#define MVP_YELLOW   6
#define MVP_LIGHTBLUE 8

int _X, _Y, _CHIGH, _CWIDT, _CCOLO, _GCURX, _GCURY, _COLOR=0, _CURSOR = OFF;
int _MOUSE_X, _MOUSE_Y, _CNT_MSKEY, _CNT_VGACURX, _CNT_VGACURY;

void initmvp();
void setupoverlay();
void lgrab(int time);
void initoverlut();
void mptabs(int *x, int *y, int *color);
unsigned char minqclr(int *x, int *y);
void minithcur(int *h, int *w, int *c);
void mmovhcurabs(int *x, int *y);
void mbox(int *xl, int *yl, int *x2, int *y2);
void mmovabs(int *x, int *y);
void mlnabs(int *x, int *y);
void mlnabs_a(int *x, int *y);
void mlnabs_b(int *x, int *y);
void mdelhcur();
void mpolylnabs(int *x, int *y, int *n);
void polyln(int *x, int *y, int *n);
void mpolyrel(int *x, int *y, int *n);
void setgframe(int frame);
void msetxor(int *flag);
void msetcolor(int *color);
int wpixel(int x, int y, int color1);
```

```

int rpixel(int x,int y);
void outscreen ( int x, int y, float value, int color );
void outscreen_msg ( int x, int y, char pos[35], int color );
void mvpscreen( int x, int y, char *str );
int mreadlocator( int *x, int *y, int *sw );
int getmskey(int *x,int *y,int *sw);
void hide_mcursor( void );
void shaw_mcursor( void );
void msetlocator( int *x, int *y );
void wait_release( void );

void setupoverlay()
{
    im_opmode(2,2); im_clear(2,0); im_opmode(3,6); im_outmode(0);
    im_outpath(6,-1,2,1);          initoverlut(); im_opmode(1,2);
}

void lgrab(int time)
{
    im_opmode(3,6); im_sync(1,0); im_inmode(3); im_video(1,1);
    im_opmode(2,6); im_outpath(6,-1,0,0); im_opmode(3,6);
    im_cgrab(time);
    while(!kbhit());
    im_cgrab(0);
    im_opmode(3,6); initoverlut(); im_opmode(1,2);
}

void initoverlut()
{
    int i, section = 0;
    unsigned char OPR[15],OPG[15],OPB[15],TR[256],TG[256],TB[256];

    for(i=0;i<256;i++) TR[i] = TG[i] = TB[i] = i;

    OPR[0] = 0x00; OPG[0] = 0x00; OPB[0] = 0xFF; /* Color 1 */
    OPR[1] = 0x00; OPG[1] = 0xFF; OPB[1] = 0x00; /* Color 2 */
    OPR[2] = 0x00; OPG[2] = 0xFF; OPB[2] = 0xFF; /* .... */
    OPR[3] = 0xFF; OPG[3] = 0x00; OPB[3] = 0x00;
    OPR[4] = 0xFF; OPG[4] = 0x00; OPB[4] = 0xFF;
    OPR[5] = 0xFF; OPG[5] = 0xFF; OPB[5] = 0x00;
    OPR[6] = 0xFF; OPG[6] = 0xFF; OPB[6] = 0xFF;
}

```

```

    OPR[7] = 0x00; OPG[7] = 0x88; OPB[7] = 0xFF;
    OPR[8] = 0x00; OPG[8] = 0x88; OPB[8] = 0x00;
    OPR[9] = 0x00; OPG[9] = 0x88; OPB[9] = 0x88;
    OPR[10] = 0x88; OPG[10] = 0x00; OPB[10] = 0x00;
    OPR[11] = 0x88; OPG[11] = 0x00; OPB[11] = 0x88;
    OPR[12] = 0x88; OPG[12] = 0x88; OPB[12] = 0x00;
    OPR[13] = 0x88; OPG[13] = 0x88; OPB[13] = 0x88;
    OPR[14] = 0x00; OPG[14] = 0x00; OPB[14] = 0x00; /* Color 15 */

    im_olutlay(section, OPR, OPG, OPB, TR, TG, TB);
}

void mptabs(int *x, int *y, int *color)
{
    unsigned char far *p;
    unsigned char current_color;
    int a;

    im_opmode(0,2);
    a = (*y) / 128;
    im_cpuwin( a );
    p = (unsigned char far *)0xD0000000;
    p += ((*y) - a*128) * 512 + *x;
    current_color = *p;
    *p = ( ( (unsigned char)((*color)%16 ) ) ^ current_color);
    im_opmode(1,2);
}

unsigned char minqclr(int *x, int *y)
{
    unsigned char far *p;
    int a;
    unsigned char current_color;

    im_opmode(0,2);
    a = (*y) / 128;
    im_cpuwin( a );
    p = (unsigned char far *)0xD0000000;
    p += ((*y) - a*128) * 512 + *x;
    current_color = *p;
    im_opmode(1,2);
    return(current_color);
}

```

```

}

/*****

void minithcur(int *h, int *w, int *c)
{
    _CHIGH = *h;    _CWIDT = *w;    _CCOLO = *c;
    _MOUSE_X = 0;   _MOUSE_Y = 0;   _CNT_MSKEY = 0;
}

void mmovhcurabs(int *x, int *y)
{
    im_drawmode(2);
    im_setcolor(_CCOLO);
    if(_CURSOR == ON) {
        im_move(_X, _Y-_CHIGH);  im_line(_X, _Y+_CHIGH);
        im_move(_X-_CWIDT, _Y);  im_line(_X+_CWIDT, _Y);
    }
    if((*x != _X) || (*y != _Y)) {
        im_move(*x, *y-_CHIGH);  im_line(*x, *y+_CHIGH);
        im_move(*x-_CWIDT, *y);  im_line(*x+_CWIDT, *y);
        _CURSOR = ON;
        _X = *x;    _Y = *y;
    }
    else _CURSOR = OFF;
    im_setcolor(_COLOR);
}

void mbox(int *x1, int *y1, int *x2, int *y2)
{
    im_drawmode(2);    im_setcolor(MVP_RED);
    im_move(*x1, *y1); im_rect(*x2, *y2);
    im_setcolor(_COLOR);
}

void mmovabs(int *x, int *y)
{
    im_move(*x, *y);
    _GCURX = *x;    _GCURY = *y;
}

void mlnabs(int *x, int *y)

```



```

{
    im_drawmode(2); im_setcolor(MVP_LIGHTBLUE);
    im_move(_GCURX, _GCURY); im_line(*x, *y);
    im_setcolor(_COLOR);
    _GCURX = *x;    _GCURY = *y;
}

```

```

void mlnabs_a(int *x, int *y)

```

```

{
    im_drawmode(2); im_setcolor(MVP_GREEN);
    im_move(_GCURX, _GCURY); im_line(*x, *y);
    im_setcolor(_COLOR);
    _GCURX = *x; _GCURY = *y;
}

```

```

void mlnabs_b(int *x, int *y)

```

```

{
    im_drawmode(2); im_setcolor(MVP_RED);
    im_move(_GCURX, _GCURY); im_line(*x, *y);
    im_setcolor(_COLOR);
    _GCURX = *x;    _GCURY = *y;
}

```

```

void mdelhcur()

```

```

{
    mmovhcurabs(&_X, &_Y);
}

```

```

void mpolylnabs(int *x, int *y, int *n)

```

```

{
    int i;
    im_drawmode(2); im_setcolor(MVP_GREEN);
    im_move(_GCURX, _GCURY);
    for(i=0; i<*n; i++) {
        im_line(*(x+i), *(y+i));
    }
    im_move(_GCURX, _GCURY);
}

```

```

void polyln(int *x, int *y, int *n)

```

```

{
    int i;

```

```

    im_move(_GCURX, _GCURY);
    for(i=0; i<*n; i++) {
        im_line(*(x+i), *(y+i));
    }
    im_move(_GCURX, _GCURY);
}

void mpolyrel(int *x, int *y, int *n)
{
    int i;
    im_drawmode(2); im_setcolor(MVP_GREEN);
    im_move(_GCURX, _GCURY);
    for(i=0; i<*n; i++) {
        im_line( *(x+i)+_GCURX, *(y+i)+_GCURY );
    }
    im_move(_GCURX, _GCURY);
}

void setgframe(int frame)
{
    im_opmode(1, frame);
}

void msetxor(int *flag)
{
    im_drawmode( ((*flag) % 2) * 2 );
}

void msetcolor(int *color)
{
    im_setcolor( (*color) % 16 );
    _COLOR = *color;
}

int wpixel(int x, int y, int color1)
{
    mptabs(&x, &y, &color1);
}

int rpixel(int x, int y)
{
    return( (int)minqclr( &x, &y ) );
}

```

```

}

void outscreen ( int x, int y, float value, int color )
{
    char pos[20];
    int strln;

    sprintf ( pos, "%7.2f", value );
    im_setcolor( color % 16 );
    im_move ( x, y );
    strln = strlen(pos) + 1;
    im_gtext ( 1, strln, pos );
    im_setcolor ( _COLOR );
}

void outscreen_msg ( int x, int y, char pos[35], int color )
{
    int strln;

    im_setcolor( color % 16 );
    im_move ( x, y );
    strln = strlen(pos) + 1;
    im_gtext ( 1, strln, pos );
    im_setcolor ( _COLOR );
}

void mvpscreen( int x, int y, char *str )
{
    char *stp;      int strln = 0;

    for(stp = str; *stp; stp++)      strln++;
    im_move( x, y ); im_gtext ( 1, strln, str );
}

int mreadlocator( int *x, int *y, int *sw )
{
    union REGS ri, ro;
    int ss, sx, sy;

    ri.x.ax = 3;

    while(1) {

```

```

int86( 0x33, &ri, &ro ):

sx = ro.x.cx;  sy = ro.x.dx;  ss = ro.x.bx;

if(sx != _MOUSE_X || sy != _MOUSE_Y || ss != _CNT_MSKEY) {
    if( sx > _MAX_X ) {
        sx = _MAX_X; msetlocator( &sx, &sy );
    }
    *x = _MOUSE_X = sx; *y = _MOUSE_Y = sy;
    *sw = _CNT_MSKEY = ss;
    break;
}
}
return(128);
}

int getmskey(int *x,int *y,int *sw)
{
    union REGS ri, ro;
    int      ss, sx, sy;
    unsigned pre_sw, st;

    ri.x.ax = 3;
    ss = (*sw) & 3;

    do{
        pre_sw = ss;

        while(1) {
            int86( 0x33, &ri, &ro );

            sx = ro.x.cx;  sy = ro.x.dx;  ss = ro.x.bx &
0x03:

            if(sx != _MOUSE_X || sy != _MOUSE_Y || ss !=
_CNT_MSKEY) {
                if( sx > _MAX_X ) {
                    sx = _MAX_X; msetlocator( &sx, &sy
);
                }
                _MOUSE_X = sx; _MOUSE_Y = sy; _CNT_MSKEY =
ss:

```

```

                                break;
                                }
                                }
                                st = ( pre_sw << 4 ) + ss;
                                if(st != 0x11 && st != 0x22 && st != 0x33)
                                    mmovhcurabs( &sx, &sy );
                                }while(st == 0);

                                *x = sx; *y = sy; *sw = ss;
                                return(st);
                                }

void hide_mscursor( void )
{
    union REGS r;

    r.x.ax = 3;
    int86( 0x33, &r, &r );
    _CNT_VGACURX = r.x.cx;
    _CNT_VGACURY = r.x.dx;
    r.x.ax = 2;
    int86( 0x33, &r, &r );
    msetlocator( &_MOUSE_X, &_MOUSE_Y );
}

void shaw_mscursor( void )
{
    union REGS r;

    r.x.ax = 1;
    int86( 0x33, &r, &r );
    msetlocator( &_CNT_VGACURX, &_CNT_VGACURY );
}

void msetlocator( int *x, int *y )
{
    union REGS r;

    r.x.ax = 4;                r.x.cx = *x;    r.x.dx = *y;
    int86(0x33, &r, &r);
}

```



```
void wait_release( void )
{
    union REGS ri, ro;

    ri.x.ax = 6;    ri.x.bx = 3;
    do {    int86( 0x33, &ri, &ro );    } while( ro.x.ax != 0 );
}
```

```
/* 메뉴구동 */
```

```
#include <io.h>
#include <bios.h>
#include <dos.h>
#include <memory.h>
#include <malloc.h>
#include <fcntl.h>
#include <process.h>
#include <stdio.h>
#include <stdarg.h>
#include <conio.h>
#include <stdlib.h>
#include <string.h>
#include <graph.h>
#include <math.h>
```

```
#include <menuname.h>
```

```
#define NUL      0
#define NOWORK   (-1)
#define ENDWORK  ( 0)
#define SHELL    ( 1)
#define DATA_VIEW ( 2)
```

```
static int      USERWORK = NOWORK;
```

```
void WaitKey( void );
int  _getkey_( void );
int  GetKey( void );
int  LoGo( void );
int  help( void );
int  systemFunc( void );
int  fileFunc( void );
int  workFunc( void );
int  RandomInput( void );
int  initialize( void );
int  OSfunc( void );
int  Xitfunc( void );
int  saveFunc( void );
int  savetypeFunc( void );
```

```

int Enterfunc( void );
int dirFunc( void );
int keyclick( void );
int myclock( void );
int drawtuck0( void );
int drawtuck1( void );
int drawtuck6( void );
int Image0Func( void );
int datadisplayFunc( void );
int xfordataFunc( void );
int calibrationFunc( void );
int input_standard_data( void );
int input_ratio_data( void );
int system_configuration( void );

void main( void )
{
    int     done = 0;

    init_start();

    do {
        SetGraph();
        initmouse();
        InitHanGul((unsigned)_RED, (unsigned)_BLUE);
        initmenu(&logo_menu);
        short_funcs = short_cuts;
        max_short_funcs = MAX_SHORT_CUT;
        LoGo();
        switch(USERWORK){
            case ENDWORK : closemenu(); EndHanGul(); exit(0);
            case SHELL   : closemenu(); EndHanGul();
                                done = spawnlp(P_WAIT,
"COMMAND.COM", "COMMAND.COM", NULL);
                                if(done != 0) {
printf("ERROR !!\n");  getch(); }
                                break;
        }
        USERWORK = NOWORK;
    } while (TRUE);
}

```

```

void WaitKey( void )
{
    struct dostime_t t;

    STATE_FLAGS = _bios_keybrd(_KEYBRD_SHIFTSTATUS);

    sprintf( stat_msg, "{%-4s} %-28s *알림*-%-25s",
        HANGUL_ON ? "한글" : "영문",
        MOUSE_EXISTS ? "마우스:좌-선택, 우-취소" :
            "키보드:방향키-이동,RET-선택",
        S_MSSG);

    if (CLOCK_ON) {
        _dos_gettime(&t);
        sprintf((char *)(stat_msg + strlen(stat_msg) - 1),
            " %02d:%02d:%02d", t.hour, t.minute,
t.second);
    }
    else
        sprintf((char *)(stat_msg + strlen(stat_msg) - 1), "
");

    statmsg();
}

int _getkey_( void )
{
    uint key, lo, hi;

    if (MOUSE_EXISTS) {

        IconON();
        setpointer(XMICKY, YMICKY);
        key = getmouse();
        setpointer(MAX_WIDTH, MAX_DEPTH);

        if (key) return(0);
        else goto _key_;
    }
    else while(_bios_keybrd(_KEYBRD_READY) == 0)
        WaitKey();

_key_ :
    key = _bios_keybrd(_KEYBRD_READ);
}

```

```

        lo = key % 256;          /*      General      */
        hi = key / 256;         /*      Scan       */
        return((lo == 0) ? hi + KEY_OFFS : lo);
    }

int GetKey( void )
{
#define getrandom(min,max)  ((rand() % (int)(((max) + 1) - (min))) +
(min))

    if (KEY_CLICK && !MOUSE_FLAG)
        Beep(getrandom(3000, 5000), 4);

    return( _getkey_() );
}

int LoGo( void )
{
    return(openmenu(&logo_menu));
}

/*      MAIN MENU      */

int help( void )
{
    messagebox(200, 100, "Good morning ! [ESC]", ESC);
    return(NO_EXIT);
}

int systemFunc( void )
{
    return(openmenu(&system_menu));
}

int fileFunc( void )
{
    return(openmenu(&file_menu));
}

int workFunc( void )
{
    return(openmenu(&work_menu));
}

/*      System Menu      */

```



```

int RandomInput( void )
{
    altitude();
}

int initialize( void )
{
    inithalo();
    return(NORMAL_EXIT);
}

int OSfunc( void )
{
    USERWORK = SHELL;
    return( EXIT_M_MENU );
}

int Xitfunc( void )
{
    USERWORK = ENDWORK;
    return( EXIT_M_MENU );
}

/*      File Menu      */

int saveFunc( void )
{
    return(openmenu(&save_menu));
}

int savetypeFunc( void )
{
    openmenu(&savetype_menu);
    return( NORMAL_EXIT );
}

int Enterfunc( void )
{
    FILE *fpi;
    char img[30], dat[30];

    static BUFTYPE buf[3] = {
        "c:/mvp/lee/Sam",
        "Sdata",
        "Xdata"
    };
};

```

```

static char *mess[3] = {
    "IMAGE 1 ",
    "IMAGE 2 ",
    "DATA : "
};
int n;

n = gethans(15, 3 * 16, 18, 3, "* File Name *", mess, buf);
if( n == -4 ) return( 0 );
switch( n ) {
    case 0 : case 2 : sprintf( img, "%s.img", buf[n]
);read_image( img );
                                break;
    case 1 : sprintf( dat, "%s.dat", buf[n] ); fpi = fopen(
dat, "rb" );
                                break;
    default: break;
};
im_opmode(2, 3);      im_clear(3, 0);
im_opmode(0, 6);
return( n );
}

int dirFunc( void )
{
    int    ret;

    static ITEMYPE buf[2] = {
        "FILE #1",
        "FILE #2",
    };
    ret = getitem(15, 5 * 16, 4, "* File Name * ", 2, buf);
    return(ret);
}

int keyclick( void )
{
    FLIP(KEY_CLICK);      return(NORMAL_EXIT);
}

int myclock( void )
{
    FLIP(CLOCK_ON);      return(NORMAL_EXIT);
}

```

```

/*      Work Menu      */

int drawtuck0( void )
{
    altimap();
    return( NORMAL_EXIT );
}

int drawtuck1( void )
{
    region_color();
    return( NORMAL_EXIT );
}

int drawtuck6( void )
{
    char    s[30];

    shaw_zonedata();
    return( NORMAL_EXIT );
}

int Image0Func( void )
{
    char s[30];

    strcpy(s, S_MSSG);          strcpy(S_MSSG, "아무키나 치세요(키보드)");
WaitKey();
    lgrab(-1);
    strcpy(S_MSSG, s);
}

/*      Data Didsplay Menu      */

int datadisplayFunc( void )
{
    return(openmenu(&datadisplay_menu));
}

/* XFOR DATA MENU */
int xfordataFunc( void )
{
    return( openmenu(&xfordata_menu) );
}

```

```
}
int calibrationFunc( void )
{
    calibration();
    return(NORMAL_EXIT);
}
int input_standard_data( void )
{
    return(NORMAL_EXIT);
}

int input_ratio_data( void )
{
    return(NORMAL_EXIT);
}

int system_configuration( void )
{
    return(NORMAL_EXIT);
}
```

```
/* 메뉴표시 항목*/
```

```
#define MAX_ITEM      6
ITEM  main_menu_item[MAX_ITEM] = {
    {
        TRUE,          /* ABLE */
        TRUE,          /* has SubMenu */
        None,
        " 시스템 ", '1',
        NULL,
        systemFunc
    },
    {
        TRUE,
        TRUE,
        None,
        " 화 일 ", '2',
        NULL,
        fileFunc
    },
    {
        TRUE,
        TRUE,
        None,
        " 자료 입력 및 작업 ", '3',
        NULL,
        workFunc
    },
    {
        TRUE,
        TRUE,
        None,
        " 데이터 표시 ", '4',
        NULL,
        datadisplayFunc
    },
    {
        TRUE,
        TRUE,
        None,
```



```

        " Data Base ", '5',
        NULL,
        databaseFunc
    },
    {
        TRUE,
        TRUE,
        None,
        " [ @A@-@G@I@S ]", '6',
        NULL,
        xfordataFunc
    }
};

MENU logo_menu = {
    1, 8,
    12, 0, MAX_ITEM,
    NULL,
    TRUE,
    main_menu_item,
};

#undef MAX_ITEM

#define MAX_ITEM 6
ITEM system_menu_item[MAX_ITEM] = {
    {
        TRUE, FALSE,
        None,
        " AGIS에 대하여 ", -1,
        NULL,
        NullFunc
    },
    {
        TRUE, FALSE,
        None,
        " 고 도 지 도 ", -1,
        NULL,
        RandomInput
    },
    {
        TRUE, FALSE,
        None,
        " @I (초기화) ", 'I',
        NULL,

```

```

        initialize
    },

    {
        FALSE, FALSE,
        Line,
        "-", -1,
        NULL,
    },
    {
        TRUE, FALSE,
        None,
        " @DOSHELL  \\@F0 ", 'D',
        NULL,
        OSfunc
    },
    {
        TRUE, FALSE,
        None,
        " @Quit      \\@X ", 'Q',
        NULL,
        Xitfunc
    }
};

```

```

MENU system_menu = {
    1, 16*2,
    14, 0, MAX_ITEM,
    NULL,
    FALSE,
    system_menu_item,
};

```

```
#undef MAX_ITEM
```

```

#define MAX_ITEM      6
ITEM  file_menu_item2[MAX_ITEM] = {
    {
        TRUE, FALSE,
        None,
        " @Load      F3 ", 'L',
        NULL,
    }
};

```

```

        Enterfunc
    },
    {
        TRUE, TRUE,
        None,
        " @Save/Del      F2 ", 'S',
        NULL,
        saveFunc
    },
    {
        TRUE, FALSE,
        None,
        " @Directory ---- ", 'D',
        NULL,
        dirFunc
    },
    {
        TRUE, FALSE,
        None,
        " @Change Dir ---  ", 'C',
        NULL,
        NullFunc
    },
    {
        FALSE, FALSE,
        Line,
        "- ", -1,
        NULL,
        NullFunc
    },
    {
        TRUE, FALSE,
        OOFF,
        " Key click ", -1,
        (void *)&KEY_CLICK,
        keyclick
    }
};

```

```

MENU file_menu = {
    10, 16 * 2,
    16, 0, MAX_ITEM,

```

```

        NULL,
        FALSE,
        file_menu_item2,
};
#undef MAX_ITEM

#define MAX_ITEM 5
ITEM work_menu_item[MAX_ITEM] = {
    {
        TRUE, TRUE,
        None,
        " 01 - 지도 영상 입력 ", '1',
        NULL,
        Image0Func
    },
    {
        FALSE, FALSE,
        Line,
        "-", -1,
        NULL,
        NullFunc
    },
    {
        TRUE, TRUE,
        None,
        " 02 - 각종 경계선 입력하기 ", '2',
        NULL,
        drawtuck0
    },
    {
        TRUE, TRUE,
        None,
        " 03 - 영역에 해당색 지정하기", '3',
        NULL,
        drawtuck1
    },
    {
        TRUE, TRUE,
        None,
        " 04 - 등고선 입력 모드 ", '4',
        NULL,
    }
};

```

```

        drawtuck6
    }
};

MENU work_menu = {
    19, 16 * 2,
    0, 0, MAX_ITEM,
    NULL,
    FALSE,
    work_menu_item,
};
#undef MAX_ITEM

#define MAX_ITEM 3
ITEM datadisplay_menu_item[MAX_ITEM] = {
    {
        TRUE, TRUE,
        None,
        " #1 - 영 상 위 ", '1',
        NULL,
        NullFunc
    },
    {
        TRUE, TRUE,
        None,
        " #2 - 모니터 위 ", '2',
        NULL,
        NullFunc
    },
    {
        TRUE, TRUE,
        None,
        " #3 - 동시 표시 ", '3',
        NULL,
        NullFunc
    }
};

MENU datadisplay_menu = {
    42, 16 * 2,
    0, 0, MAX_ITEM,
    NULL,
    FALSE,

```



```

        datadisplay_menu_item,
};

#undef MAX_ITEM

#define MAX_ITEM      2
ITEM  xfordata_menu_item[MAX_ITEM] = {
    {
        TRUE, FALSE,
        None,
        " CALIBRATION  ", -1,
        NULL,
        calibrationFunc
    },
    {
        TRUE, FALSE,
        None,
        " system config. ", -1,
        NULL,
        system_configuration
    }
};

MENU xfordata_menu = {
    66, 16 * 2,
    0, 0, MAX_ITEM,
    NULL,
    FALSE,
    xfordata_menu_item,
};

#undef MAX_ITEM

```