

筑波大学大学院修士課程

理工学研究科修士論文

携帯電話を用いたコンピュータの
遠隔操作システムの研究

中須正人

平成14年3月

筑波大学大学院修士課程

理工学研究科修士論文

携帯電話を用いたコンピュータの
遠隔操作システムの研究

中 須 正 人

主任指導教官 電子・情報工学系 田中二郎

目次

論文要旨

第1章	序論	1
1.1	研究の目的	2
1.2	本論文の構成	2
第2章	携帯電話版 VNC システム	3
2.1	SVNC プロキシ	3
2.1.1	画像情報の保存, 切り取り, 変換	5
2.1.2	入力デバイスの違いの吸収	5
2.2	CRFB プロトコル	6
2.3	SVNC ビューア	6
第3章	携帯電話向けの表示・操作インタフェース	7
3.1	画面領域の登録と呼び出し	8
3.2	操作モード	10
3.2.1	視点変更モード	10
3.2.2	マウスポインタモード	12
3.2.3	文字入力モード	14
3.3	表示モード	17
3.3.1	Normal View	17
3.3.2	Split View	19
3.3.3	Global View	19
3.4	ガイド機能	20
3.4.1	表示インタフェース	21
3.4.2	Split View におけるガイド機能	21
3.4.3	Global View におけるガイド機能	22
第4章	実装システム	24
4.1	システム構成	24
4.2	設定と起動	24
4.3	CRFB プロトコル	24
4.3.1	VNC セッションの初期化	25
4.3.2	入力イベントの通知	28

4.3.3	画面の通知	29
4.3.4	画面領域登録の通知	29
4.4	スライドショー機能	30
4.5	画面更新の頻度	30
4.6	実験環境	30
4.7	動作確認	31
4.7.1	操作例：Windows の終了操作	31
第5章	関連研究	41
5.1	携帯電話やPDAを用いた遠隔操作	41
5.1.1	コンテンツ変換に基づいた遠隔操作	41
5.1.2	画像に基づいた遠隔操作	41
5.2	小画面インタフェース	42
5.2.1	小画面インタフェースの議論	42
第6章	結論	43
	謝辞	
	参考文献	

論文要旨

携帯電話から遠隔計算機のデスクトップにアクセスする汎用的な仕組みとして、ビットマップ転送に基づく VNC を利用したシステムを提案する。本システムでは、デスクトップのビットマップを VNC を用いて携帯電話に転送し、遠隔計算機のデスクトップを携帯電話から操作可能にしている。さらに、携帯電話の小画面およびキー入力デバイスを考慮したインタフェースを提供するために、遠隔計算機と携帯電話の間にプロキシを置く。携帯電話には、遠隔計算機のデスクトップを操作する際にユーザの負担を減らすための工夫として、画面領域の記憶および表示モードを組み込んだ。画面領域の記憶とは、ユーザが表示した画面領域の位置情報を任意に記憶させ、少ないキー操作数で記憶領域に移動可能にする仕組みである。表示モードは3つあり、ユーザは使用用途に合わせて切り替える。Normal View は携帯電話の全画面を使用する。Split View は携帯電話の画面を2つに分けて2画面同時に表示する。Global View はサーバ計算機の全画面を携帯電話の全画面に表示する。我々は、本システムのプロトタイプを Java で実装し、iモード Java 対応携帯電話エミュレータ上でその動作を確認した。

第1章 序論

携帯電話が携帯される機会は、パームサイズ計算機やハンドヘルド等のPDA(Personal Digital Assistant)よりも多い。したがってこの携帯電話を用いて遠隔情報機器(ネットワーク経由でアクセス可能な遠隔計算機や情報家電)を操作できれば、以下のような場合に好都合である。

1. 遠隔計算機のデスクトップに保存されているデータを、出張先から確認する。
2. 緊急時に、出先からサーバを再起動する。
3. 携帯電話で表示がうまくできないWebコンテンツを閲覧する。

1の場合は、ユーザが出張先で、取引先に商品の在庫情報を提示しようとしたが、在庫情報を印刷するのを忘れてしまい、すぐに在庫情報のデータを確認しなくてはならないという状況である。在庫情報のデータは会社にあるユーザのパソコンに保存されていて、Web上にはなくデスクトップ上におかれている。

2の場合は、ユーザが会社から離れていた場所で、携帯電話で会社からサーバがダウンしたという情報を知らされ、ユーザはダウンしたサーバ計算機の管理者であるので、すぐに復旧させる必要があるという状況である。

3の場合は、ユーザは出先である目的地の場所を忘れてしまい、急遽目的地を示す地図を確認しようと考えたが、地図データは会社の計算機に保存されており、その地図データをすぐに確認したいという状況である。

4の場合は、ユーザが携帯電話向けに作成されていないWebコンテンツを閲覧する状況である。携帯電話向けに作成されていないWebコンテンツとはフレームやテーブルを多用したもの、サイズの大きい画像ファイルが貼り付けてあるもの、FlashやShockwaveの技術が使われたものといった携帯電話のブラウザから閲覧することが上手く出来ないようなWebコンテンツである。

個々の場合を想定して、対処可能にするための仕組みを予め準備することは可能である。すなわち、操作対象となりうる遠隔情報機器に、適切な処理が行われるようにソフトウェアを設定しておけば良い。しかしながら、新しい処理を追加する度にソフトウェアを設定することになるので効率が悪い。そこで、なるべく汎用的な仕組みを、携帯される機会が高い携帯電話に実現しておくことが望ましいと考え、我々は研究を進めている [1]。

1.1 研究の目的

本研究では、携帯電話から遠隔情報機器にアクセスするための汎用的な仕組みを実現することを目的として、ビットマップベースの画面転送機能を有する VNC(Virtual Network Computing)[2] を用いたアーキテクチャを提案する。本システムは、VNC を用いて遠隔情報機器を操作するためのビットマップを携帯電話に転送し、遠隔計算機のデスクトップを携帯電話から操作することを実現している。さらに本アーキテクチャでは、VNC サーバと携帯電話の間にプロキシを挟むことによって、携帯電話に対するトラフィックを抑制し、携帯電話が備える入出力に適切な操作インタフェースを提供する。本論文では、特に遠隔計算機(パーソナルコンピュータやワークステーション、サーバ計算機など)のデスクトップを操作する際に生じる操作上の困難を軽減するための操作インタフェースについても述べる。

1.2 本論文の構成

本論文の構成は以下の通りである。

第 2 章では、本研究で提案する携帯電話向けの VNC システムの構成について述べる。特に、プロキシサーバが行う仕事について詳しく述べる。

第 3 章では、携帯電話で動作する VNC ビューア上でユーザに提供する携帯電話向けの表示・操作インタフェースについて述べる。

第 4 章では、本研究で実装した携帯電話向けの VNC システムについての実験環境や動作検証について紹介する。

第 5 章では、本研究の関連研究を紹介および考察を述べる。

第 6 章では、本論文の結論を述べる。

第2章 携帯電話版 VNC システム

遠隔情報機器，特に遠隔計算機のデスクトップの状態をそのまま転送すると，一般には大きなネットワークトラフィックが生じる．さらに，携帯電話の小画面に適した表示を提供するとともに，遠隔情報機器と携帯電話との入力デバイスの差異を吸収する必要がある．これらの処理を行う計算機は，多くの計算量とメモリを必要とする．しかしながら，携帯電話のメモリは 200Kbyte 程度，CPU の処理能力は数 10MHz であり，デスクトップ計算機と比べるとかなり低い．

また，携帯電話は移動しながら利用するため，現在の基地局エリアから異なる基地局エリアに移動する場合にネットワークが一時的に切断される．長時間送受信が行われない場合は，ネットワークタイムアウトが発生することがあり，処理を復帰させるための状態保持が重要となる．本研究では，これらの処理を携帯電話のために肩代わりするプロキシを導入した携帯電話版 VNC システムを構築した．図 2.1 は本システムにおける携帯電話版 VNC ビューアの動作イメージである．

本アーキテクチャは，VNC を利用することによって，種々の遠隔情報機器に適用可能な枠組みとなっている．VNC は，VNC サーバ (fat サーバ) と VNC ビューア (thin クライアント) からなるリモートディスプレイシステムである．VNC サーバが遠隔計算機の画面のビットマップ情報を転送して，VNC ビューアがそのビットマップを描画する単純な仕組みである．その転送には，RFB (Remote Frame Buffer) プロトコル [3] を用いる．VNC は，RFB プロトコルのシンプルな設計により高いポータビリティを実現しており，結果として様々なプラットフォーム (Windows, Macintosh, Unix 系 OS など) を操作する他に，ネットワーク経由で情報機器を操作するためにも用いられている [4] ．

携帯電話版 VNC システムの構成を図 2.2 に示す．本システムは，携帯電話上で動作する VNC ビューア (SVNC ビューア)，および遠隔情報機器で動作する VNC サーバと SVNC ビューアの間には挟まれるプロキシ (SVNC プロキシ) から構成される．また，SVNC ビューアと SVNC プロキシとの通信には CRFB (Compact RFB) プロトコルを用いる．

2.1 SVNC プロキシ

SVNC プロキシは以下の基本処理を行う¹．

¹ここでは遠隔計算機のデスクトップを操作対象とした説明をしているが，操作対象が情報家電の操作パネルにも同様の処理を行う．



図 2.1: 携帯電話版 VNC ビューアの動作イメージ

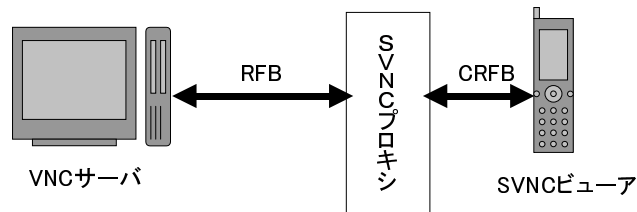


図 2.2: 携帯電話版 VNC システム

- VNC サーバから RFB プロトコルに従い遠隔計算機のデスクトップの画像情報をビットマップとして受信し蓄える。さらに、そのビットマップに対し、携帯電話の画面領域に合わせた切り取り、縮小等の変換処理を行う。
- 変換処理したビットマップを CRFB プロトコルに従い SVNC ビューアに送信する。
- SVNC ビューアから CRFB プロトコルに従い受信したユーザの操作情報を変換し、RFB プロトコルに従い VNC サーバに送信する。

VNC サーバと VNC ビューアは RFB プロトコルにより通信を行う。RFB は、ネットワーク上にある計算機間でフレームバッファやユーザの入力などを転送するための手続きを規定している。VNC サーバが RFB プロトコルを用いて遠隔計算機のデスクトップのビットマップをビューアに送り、ユーザは VNC ビューア上でそのビットマップ閲覧することで、デスクトップを操作することが可能である。

RFB プロトコルでは、VNC サーバが監視する遠隔計算機のデスクトップが更新される度に、更新内容がビットマップとして VNC ビューア側へ転送される。よって、携帯電話へのビットマップの転送に RFB プロトコルで送られるビットマップをそのまま用いることは、携帯電話の伝送速度を考えると現実的ではない。例えば NTT DoCoMo の 503i 系携帯電話で利用可能な伝送速度は最大 9600bps であり、IMT-2000 の FOMA 対応携帯電話であっても伝送速度は下り最大 384Kbps である。これに対

してデスクトップが頻繁に更新される状態で生じるトラフィックは膨大である．そこでSVNC プロキシを設け，VNC サーバとSVNC ビューア間の通信のトラフィックを緩和する役目を与える．

2.1.1 画像情報の保存，切り取り，変換

携帯電話で扱う画像領域は 120 × 130 ピクセル程度であるから，遠隔計算機側のデスクトップの画面サイズ(例えば，1280 × 1024 ピクセル)で送信する必要がない．よって，遠隔計算機から送られるデスクトップ画像は直接SVNC ビューア側へ送らずに，SVNC プロキシに蓄える．SVNC プロキシは蓄えた画像情報から携帯電話で表示する画像領域の部分のみを切り取り，携帯電話のSVNC ビューア側へ渡す．

携帯電話の小画面から遠隔計算機のデスクトップを閲覧する際に，デスクトップの全体を表示する場合はデスクトップに表示されているアイコンなどの詳細を見ることは難しく，一部分を詳細表示する場合は周りの情報が把握できないという問題がある．よって，SVNC プロキシは画面領域の拡大/縮小により目的の場所を表示するための機能を提供する．SVNC プロキシはSVNC ビューア側から表示すべき領域(左上の座標，幅，高さ，拡大率)の情報を受け取り，蓄えたデスクトップ画像からその領域を切り出す．つづいて，SVNC プロキシは拡大率に応じて縮小した上で，SVNC ビューアに渡す．

2.1.2 入力デバイスの違いの吸収

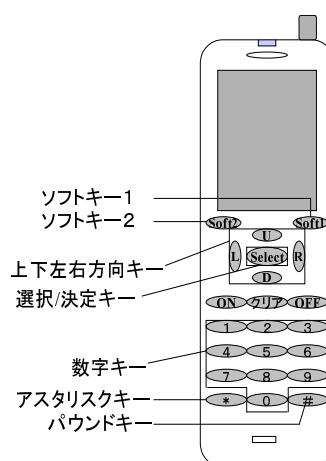


図 2.3: 携帯電話で用いられるキー

SVNC ビューアが動作する携帯電話が標準的に備える入力デバイスは，図 2.3 のように，0, 1, 2, 3, 4, 5, 6, 7, 8, 9 の数字キー，アスタリスク(*)キー，パウンド(#)キー，ソフトキー1，ソフトキー2，上方向キー，下方向キー，左方向キー，右方向キー，選択/決定キーである．しかし，ワークステーション，パーソナルコン

コンピュータ等の遠隔計算機は、入力デバイスとしてマウス、キーボードを備える。そこで、SVNC プロキシはSVNC ビューアから受信した携帯電話で行ったキー入力情報を遠隔計算機のマウス操作情報とキー入力情報に変換して、RFB プロトコルに従い遠隔計算機側へ送信する。

2.2 CRFB プロトコル

CRFB(Compact RFB) プロトコルは、RFB プロトコルを携帯電話向けに簡易化した本研究独自のプロトコルである。SVNC プロキシと携帯電話のSVNC ビューアの間の手続きをCRFB プロトコルとして規定する。

2.3 SVNC ビューア

SVNC ビューアは以下の基本処理を行う。SVNC ビューアは、SVNC プロキシに対して描画のリクエストを送信する。SVNC プロキシからCRFB プロトコルに従い画像情報を受信する表示モードに応じて画像を描画する。ユーザの操作情報をCRFB プロトコルに従いSVNC プロキシに送信する詳細については第3章で述べる。

第3章 携帯電話向けの表示・操作インタフェース

ユーザがデスクトップ計算機の画面上で行う操作を携帯電話の小画面上で行う場合は、Small Screen Problem を議論する必要がある [5, 6]。そこで、携帯電話向けの表示・操作インタフェースを設計する上で、本論文でも携帯電話における Small Screen Problem について以下に考察を行う。

画面サイズ

一般のデスクトップ計算機の画面は、サイズが 17 インチ、解像度が 1600 × 1200 ピクセルであり、ユーザが画面サイズの大きさをあまり考慮せずに、操作が出来ることを前提にしている。

対して、携帯電話の画面は、サイズが 2 インチ、解像度が 130 × 120 ピクセル程度であり、携帯電話の画面サイズはデスクトップ計算機と比べると極めて小さい。よって、デスクトップ計算機のように画面サイズを気にせずに利用することは難しい環境である。この時点で、デスクトップ計算機の画面インタフェースを携帯電話の画面インタフェースとして適用することが難しいと筆者は感じる。

よって、SVNC ビューアを設計する際には、携帯電話の画面に適した独自のインタフェースを使用する。

アイコン

デスクトップ計算機におけるアイコン表示は、ユーザに対して視覚的にファイルのデータの情報を直接訴えかける面があるので、ユーザの認識力を高められ効果的であるといえる。アイコンを携帯電話に適用してみると、携帯電話の小画面上でアイコンが効果的にユーザに認識できれば、アイコンの有用性が高いと感じる。但し、小画面ゆえにアイコンのサイズや数によっては、操作する上での混乱の元にもなるため注意が必要である。

よって、SVNC ビューアを設計する際には、VNC サーバのデスクトップにあるアイコンを活用したい。そこで、画面領域の拡大/縮小の操作を可能にして、アイコンの形や色を認識できる工夫をする。また、利用用途に応じた画面の見せ方を提供する。



図 3.1: 閲覧例：左端を拡大率 1 倍で表示

マルチウィンドウ方式

デスクトップ計算機においてのマルチウィンドウ方式は、ユーザにとって実践的でありかつ魅力もある。携帯電話の小画面インタフェースにマルチウィンドウ方式を適用する際には、携帯電話の「小画面」という限られた表示領域の問題がある。何故なら、マルチウィンドウ方式はある程度の画面サイズが必要であり、ウィンドウ操作に直接操作方式を使用するからである。具体的に述べると、小画面ではウィンドウシステムの特徴でもある垂直方向・水平方向のスクロール操作はユーザには煩わしい。そして、頻繁にスクロール操作をする必要が生じるため、ユーザにとって負担が大きい。また小空間である事から各ウィンドウの配置にユーザは時間を獲られると考えられる。更にマルチウィンドウ方式はウィンドウのサイズ変更、移動等のウィンドウ操作において絶えず、マウスによる直接操作(クリック、ドラッグ、ドロップなど)をユーザに求めるので、マウスのような入出力機器を持たない携帯電話においては不都合な面が生じる。このように、デスクトップ計算機で大変有益であると考えられているマルチウィンドウ方式を携帯電話の小画面インタフェースへの単なる適用は効果的に働かない面もあるので、注意をする必要がある。

よって、SVNC ビューアを設計する際には、ウィンドウの数(2つ)を制限して、サイズ変更や移動を行わないようにする。また、スクロール量の軽減をしてユーザが行うキー操作量を軽減する工夫をする。

以上のような考察を踏まえて、本研究では携帯電話向けの表示・操作インタフェースを設計を行う。

3.1 画面領域の登録と呼び出し

携帯電話の小画面上でユーザが同じ場所を何度も行き来することを想定した場合、ユーザは毎回同じ拡大/縮小や視点移動の操作をしなければならない。例えば、遠隔

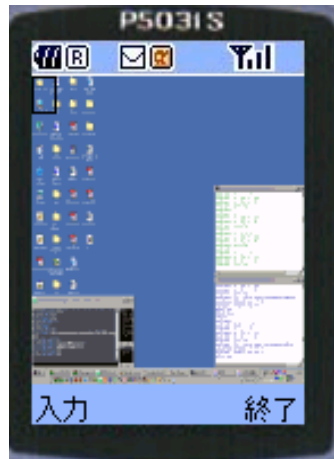


図 3.2: 閲覧例：デスクトップ全体を表示



図 3.3: 閲覧例：右端を拡大率 1 倍で表示

計算機のデスクトップの左端と右端を交互に閲覧したいような場合がある。始めに左端を拡大率 1 倍で閲覧している状態(図 3.1)から右端を拡大率 1 倍で閲覧するには、デスクトップ全体を表示させる表示モード(3.3.3 節：Global View)にして(図 3.2)、目的とする右端まで視点を移動する。そして、また拡大操作をして右端を表示させる(図 3.3)。携帯電話の拡大/縮小、視点移動は、キーを用いて行うわけであるから連続したキー操作が要求される。一般的なデスクトップ計算機であれば、入力デバイスにマウスを用いることが可能であるので、マウスポインタの移動はマウスで行えばよい。また、画面が広いので視点移動や拡大/縮小といった操作自体が必要ない。そして、右端まで視点移動を行った後、拡大の操作を行う。最後に視点移動を行い、表示位置の微調整を行う。更に、交互に閲覧する必要性から、毎回同じ拡大/縮小及び視点移動の操作を行う必要があり、ユーザにとっては大変苦痛である。

本研究ではこの問題への解決策として、ユーザが任意に画面領域を登録できる機能を提供する。ユーザが任意に移動する頻度の高い画面領域を登録することで、画

面の拡大/縮小や視点移動といった操作を何度も繰り返さずに円滑に移動することを可能にする。

ユーザが画面領域を登録する手順，および呼び出す手順は以下の通りである。

- ユーザは登録させる画面領域 (現在表示されている画面) を決め，*キーを2回押してから数字キーを押す。
- ユーザは*キーを1回押してから，呼び出す画面に割り当てた数字キーを押す。

SVNC ビューアは，ユーザが画面領域を登録するキー操作 (*キー2回押し+数字キーの操作) を行うと，押された数字キーの値，および画面領域 (左上点座標，縦横幅) を登録情報として SVNC プロキシに送信する。SVNC プロキシは受け取った登録情報を保存しておく。ユーザは，次回接続した時に，保存した登録情報を SVNC プロキシから呼び出すことが可能になる。よって，前回登録した画面領域情報を利用できるため，ユーザは起動時から，スムーズに視点変更を行うことができる。

また SVNC ビューアは，ユーザが画面領域に割り当てた数字キーを押した場合，数字キーに対応した画面領域情報を SVNC プロキシにリクエストする。リクエストされた SVNC プロキシは，登録情報にもとづいて，デスクトップから領域を切り取り，拡大/縮小処理を行った後，SVNC ビューアに送信する。

3.2 操作モード

SVNC ビューアが SVNC プロキシに接続すると，ユーザは用途に応じて，画面領域の拡大/縮小の操作を行う。また，視点変更，マウスポインタ，文字入力の各モードに切り替えて操作を行う。

画面領域の拡大/縮小操作

画面領域の拡大率は Normal View の表示モード (3.1 節) では通常は等倍で表示されている。この拡大率を各表示モードでユーザは自由に変更することが可能である。画面領域の拡大率を変更するには，拡大/縮小にそれぞれ割り当てられたキーを押すことで変更する。

3.2.1 視点変更モード

視点変更モードでは，上下左右に視点を移動し画面を切り替えることが出来る。視点とは，携帯電話に表示される遠隔計算機のデスクトップの画面領域である。ユーザが視点移動のキー操作を行うと，数字キーの値を SVNC ビューアから SVNC プロキシに送信する。SVNC プロキシは受信した数字キーの値をもとにデスクトップの視点を上下左右に動かす。



図 3.4: 視点変更モード～拡大率大



図 3.5: 視点変更モード～拡大率小

視点移動の方法として単純に携帯電話のキーイベントに割り当てるとすると、携帯電話のキーを1回押すごとに表示画面を1ピクセル分移動させることになり、大量に移動する場合には多くのキー操作を必要とする。

そこで本研究では視点移動量を拡大率に応じて変更するように工夫している。拡大率が高い場合には移動量が少なく、拡大率が低い場合には移動量が大きくなる。

例えば、図 3.4 は拡大率が等倍、図 3.5 は拡大率が 1/4 倍の表示であり、それぞれ図の左が視点移動前で、図の右が右方向にカーソル移動を1回行った後の図である。図 3.4 と図 3.5 を比較すると、拡大率の低い図 3.5 の方がキー操作1回あたりの移動量が大きいが分かる。

よって、ユーザはキー操作1回あたりのスクロール量を多くしたい場合には縮小をして拡大率を下げると良い。また、キー操作1回あたりのスクロール量を少なくしたい場合には拡大をして拡大率を上げると良い。



図 3.6: マウスポインタモード

3.2.2 マウスポインタモード

マウスポインタモードは遠隔計算機のデスクトップで、マウスポインタの移動、およびクリック操作、ドラッグ&ドロップの操作を行うことができる。マウスポインタモードでは携帯電話の画面に仮想ポインタが表示される。仮想ポインタは、図 3.6 のような携帯電話の画面上に黒い小さな四角い枠である。

ポインタの移動

ユーザは携帯電話上の仮想ポインタを携帯電話のキーを操作して移動させることで、遠隔計算機のポインタを移動する。また、ポインタが画面領域を超えて移動しようとする時、視点もポインタの移動に追従して移動する。また、遠隔計算機のデスクトップの端に到達すると、それ以上は移動しない。

クリック操作

クリックするには、左クリック、中クリック、右クリック操作に割り当てた携帯電話の各キーを対象(アイコンやウィンドウ)の上で押すことで、クリック操作を実現する。クリック操作には、シングルクリックとダブルクリックの2種類があり、ユーザは用途によってクリックのモードを変更する。クリックのモードの切り替えに割り当てたキーを押すことで、シングルクリックとダブルクリックのモードを交互に入れ替える。

ドラッグ&ドロップ

ドラッグ&ドロップの操作を行う際には、ドラッグ&ドロップのモード切り替えに割り当てたキーを対象の上で押し、仮想ポインタを動かすことで対象をドラッグ



図 3.7: 操作例 1 : アイコン操作

する．そして，ドロップする場所で，再度ドラッグ&ドロップに割り当てたキーを押すことにより，その場所に対象をドロップする．

操作例

例えば，以下のような操作が可能である．

- アプリケーションのアイコンを選択して起動する (ダブルクリック操作)
- ウィンドウからメニューを選択する (シングルクリック操作，ポインタの移動)
- アプリケーションのスクロールバーを操作する (ドラッグ&ドロップ)
- アイコンをドラッグ&ドロップする (ドラッグ&ドロップ)
- 簡単な絵や図を描く (ドラッグ&ドロップ，シングルクリック操作)

図 3.7 では，携帯電話から遠隔計算機のデスクトップにある「マイ コンピュータ」アイコンをマウスポインタでダブルクリック操作をして，「マイ コンピュータ」のウィンドウを表示させたイメージである．



図 3.8: 操作例 2 : 図の描画操作

図 3.8 では、携帯電話から遠隔計算機の「ペイント」アプリケーションを起動して、マウスポインタでドラッグ操作を行いながら、簡単な図を書いているイメージである。

3.2.3 文字入力モード

文字入力モードでは、英字、数字といった通常文字と、Ctrl、Alt などの特殊なキー入力を行う。通常文字の入力には、携帯電話に備わっている文字入力機能を用いる。携帯電話に備わっている文字入力機能とは、テキストボックスを開き、文字の種類(英字、数字)を選択して入力し、テキストボックスを閉じて文字入力を終了するものである。また、文字が入力される場所は、マウスポインタでクリックしてフォーカスを当てた対象である。

例えば、「HelloWorld」と入力する場合は、図 3.9 のように、文字を入力する対象として Windows の notepad のウィンドウをマウスポインタでクリックし、notepad にフォーカスを当てる。そして、文字入力モードに切り替え、テキストボックスを選択して開く。次に、文字種を英字モードに切り替え、文字列「HelloWorld」を入力する。入力が終了したら、テキストボックスを閉じる。最後に、ソフトキー 2 を押して、SVNC プロキシに入力した文字情報を送信することで、文字入力を終了する。結果として、VNC サーバの notepad に「HelloWorld」が文字入力された。



図 3.9: 文字列「HelloWorld」の入力

特殊キーの入力

また、特殊キー入力すなわち Ctrl, Alt, Esc, Backspace などを入力する場合には携帯電話に備わっている文字入力機能では入力できない。特殊キー入力を実現する手法としては、以下のような手法が考えられる。

1. 特殊キー専用のキーを新たにハードウェア的に実装する
2. 特殊キー入力の専用モードをソフトウェア的に実現する
3. 特殊キー入力を、キーの長押しを用いて実現する
4. 既存の携帯電話のキーインタフェースは維持して、特殊な記法を用いて実現する

1 の場合は、特殊キー専用のキーを新しく携帯電話のキーとして追加し、特殊キーを入力する場合には、その特殊キー専用のキーを押すという方法である。この手法は、ハードウェア的に実装しなければならないので、現在の携帯電話では動作しない。また、特殊キーの数だけキーを増やす必要もあり、現在の携帯電話に備わっている 19 個程度のキー数より更に増加することになる。携帯電話のキー数の増加は、ユーザの負担が更に大きくなるという問題がある。

2 の場合は、現在の携帯電話に備わっている文字入力機能では、英語、数字、仮名漢字のモードを切り替えることで、0 から 9 の数字キーだけで入力できる。そこで、



図 3.10: 特殊キー「Ctrl-S」の入力

新たに特殊キー入力のモードを追加することで、特殊キーの入力を可能にするという手法である。この手法の場合は、結局特殊キーの入力する度にモード切り替えの操作をしなくてはならないのでユーザの操作数が増える。例えば、コンソールボックスで“ Ctrl-D ”という入力をする場合には、“ 特殊キーモードに切替 ”+“ Ctrl ”+“ 英語モードに切替 ”+“ D ”のように4回のキー操作が必要となる。またこの例より多いエスケープシーケンスを入力しなくてはならないような場面を想定すると、2の手法ではユーザは素早く入力することが困難であると考えた。

3の場合は、特殊キーの入力をするのに、1や2のように特殊キーの専用なキーやモードを用いない。特殊キーの入力には、キーの長押しを用いる。例えば、“ Ctrl ”を入力したい場合には、英字モードでCに割り当てられた数字キーを長押しする。この手法には、携帯電話における長押しの判定基準が問題になる。ユーザによって携帯電話の長押しの感覚が異なり、個人個人で長押しの設定が必要になり、汎用的な基準が作れない。また、特殊キーを入力するのにキーの長押しを行うため、1や2の手法と比べて入力に時間がかかる問題がある。

そこで、本システムでは4の手法を採用した。この手法は、特殊キーの入力に特殊な記法を用いて行う。特殊な記法とは、“ !c ”, “ !a ”, “ !e ”, “ !b ”のような文字“ ! ”+「エスケープシーケンスを表す文字列」である。通常の文字列の中に“ ! ”が現れるとシステム側は“ ! ”の次に続く文字列を、エスケープシーケンスを表す文字列と認識して解釈する。

表 3.1: エスケープシーケンスの記法

記法	特殊キー
!a	Alt
!b	Backspace
!c	Ctrl
!d	Delete
!e	Esc
!f1 ~ !f12	f1 ~ f12
!n	改行 (Enter)
!t	Tab
!!	!

例えば，図 3.10 のように Windows の Notepad で“ Ctrl-S ”という入力を行いたい場合には，文字入力モードのテキストボックスに“ !cs ”という文字列を記述し，SVNC プロキシにその文字列を送信する．SVNC プロキシは，受け取った文字列から“ Ctrl-S ”と解釈して，VNC サーバ側に伝える．結果として，Notepad のファイルに保存するウィンドウが開く．

また，“！”を文字として入力したい場合は，“!!”という表記で SVNC プロキシに与えると，VNC サーバ側に文字“！”が入力される．エスケープシーケンスの記法は表 3.1 のようになる．

3.3 表示モード

SVNC ビューアではユーザの利用場面に応じて Normal View と Split View および Global View の 3 種類の表示モードをユーザに提供する．

3.3.1 Normal View

Normal View は，携帯電話の全画面を使用する表示モードである．Normal View では画面の拡大/縮小，視点変更，マウスポインタの操作を提供する．Normal View では携帯電話の全画面を利用するため閲覧性が良い．従って，文字の閲覧，編集作業を行う場合に適している．Normal View の使用例を以下に示す．

1. アイコンの閲覧，選択
2. テキスト文書の閲覧，編集
3. スケジュールを管理するソフトの閲覧，編集



図 3.11: Normal View

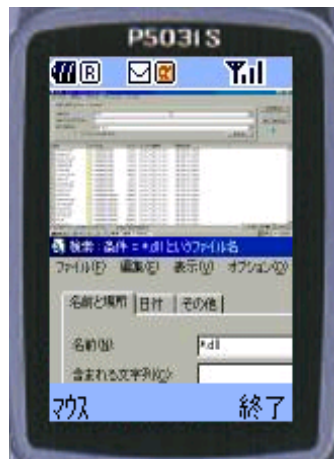


図 3.12: 2画面同時作業の例：ファイル検索

1の例は、図3.11のように、ユーザが遠隔計算機のデスクトップの左上端に並んでいるアイコンを閲覧し、目的とするアプリケーションのアイコンを選択するという場面である。

2の例は、遠隔計算機上にあるテキスト文書を、ユーザがファイル転送を行わずに閲覧および編集する場面である。

3の例は、スケジュールを管理するソフトを起動して、今日の予定を確認する。また、会議の予定をこと細かに書き込むような場面である。

以上の3つの例にあるように、Normal Viewでは主に、文書の閲覧、編集作業に適している。逆に、文書の比較などの画面を切り替える必要がある作業にはあまり向いていない。比較については、次節で述べる Split View が適している。

3.3.2 Split View

Normal View が携帯電話の全画面を使っているのに対して，Split View は図 3.12 のように携帯電話の画面を 2 つに分割 (split) して用いる．分割した 2 つの画面に対しても，Normal View で提供した画面の拡大/縮小，視点変更，マウスポインタの操作をユーザは実行できる．Split View は以下に挙げるような用途を想定している．

2画面同時作業

携帯電話の画面内で，2 つの画面領域を同時に閲覧して作業をするような場면을 Split View の用途の一つとして考えている．例えば，図 3.12 は，Windows の「検索アシスタント」アプリケーションを起動して，上の画面に検索結果を表示する部分の画面領域，下の画面に検索式を入力する部分の画面領域を表示している．ユーザは，下の画面の入力部で検索式「*.dll」を入力し，上の画面で検索結果を得られる．Normal View であれば，適切な検索式を入力し，検索結果を得るまでに画面を行き来する必要がある．画面移動する作業は，画面領域登録 (3.1 節) を用いて行っても，キー操作を何度も行う必要があり，ユーザにとって効率が悪い．Split View を用いれば，画面を行き来する為の煩雑なキー操作は必要としない．ユーザは下の画面で検索式を入力し，検索した結果を上の画面で確認することが可能である．

スライドショー機能 (過去の画面を再現する)

SVNC ビューアに現在表示されている画面領域が変化した場合は，SVNC ビューアの描画が更新されるためユーザは連続的に確認できる．しかしながら，SVNC ビューアに表示されていない画面領域の変化についてはユーザが知ることは難しい．よって，本研究では SVNC ビューアに表示されていない画面領域で過去に起こった変化を再現するスライドショー機能を提案する．

スライドショーを実行すると，下の画面から上の画面へ一定間隔で保存された過去の画面が過去から現在に向かって順々に表示される．ユーザは，必要に応じてスライドショーを一時停止させ，どのような変化が生じたのかを前後の画面を比較して把握するといったことが可能である．

3.3.3 Global View

Global View は図 3.13 のように遠隔計算機の全画面を携帯電話の全画面に表示する．Normal View と違い，拡大・縮小表示，視点移動/マウスポインタの操作は行わず，遠隔計算機の全体を捉えてユーザが操作する View である．



図 3.13: Global View

操作対象領域の指定

ユーザは、Normal View で操作対象とする領域を指定することができる。Normal View では拡大・縮小、視点移動を行い、操作対象とする領域を決定した。しかし、現在位置よりも遠い場所に操作対象領域を移動したい場合、拡大・縮小、視点移動するキー操作数は近い場所に移動する場合に比べて増加し、ユーザの負担は大きい。

そこで本研究では、遠隔計算機の全画面を表示し、操作対象となる領域を黒色の四角い枠(画面領域枠)で表示し、画面領域枠の位置を移動、枠のサイズ変更を行うことが出来るようにした。

以下の順で、ユーザは目的の操作対象領域を指定する。

- デスクトップの全体から操作対象領域とする場所を確認する。
- 画面領域枠を操作対象領域へ移動する。
- 画面領域枠を適切なサイズに変更する。

といった操作を行う。

Normal View と違い、遠隔計算機のデスクトップ全体のイメージを常時見ていることが可能であるので、周りにどのような画面があるか、どのような変化がおきているかという状況確認のための操作を行わず、操作対象領域を指定できる。

3.4 ガイド機能

ユーザが登録させた画面を呼び出す際に、ユーザがどの画面領域を登録したかを覚えていない場合がある。また、目的の場所が登録した画面領域に近い場合に、その画面領域を経由すると便利な場合がある。そこで、全体画面において登録させた画面領域が何処に位置しているのかを一覧表示できるガイド機能を考えた。ガイド機能は、Split View 及び Global View にて使用可能である。

3.4.1 表示インタフェース

ガイド機能は以下のような表示インタフェースをもっている。

1. 登録した画面領域を四角い枠で囲む(「登録画面領域枠」の表示)
2. 登録画面領域枠の中に割り当てた数字キーの数字を表示する(「登録画面番号」の表示)
3. 背景画像をモノクロに表示する

1の登録画面領域枠と2の登録画面番号は、登録画面番号(1~9)ごとに色を設定している(表3.2)。よって、枠および番号ごとに色が異なるため、他の登録画面領域枠と重なっていても、ユーザは目的の画面領域を発見しやすい。

3の背景画像のモノクロ表示は、1および2と併用したときに大変効果がある。登録画面領域枠と登録画面番号を色分けして表示する際に、表示される背景画像の色によって大変見難い場合がある。背景画像をモノクロに表示することで、背景画像の色に左右されずに1および2をユーザに見やすい表示が可能にした。ただし、1および2の色は白や灰色は避ける必要があり、表3.2の色設定に反映されている。

表 3.2: 各登録画面番号の設定色

登録画面番号	設定色
1	BLUE(青色)
2	LIME(緑色)
3	AQUA(水色)
4	RED(赤色)
5	FUCHSIA(紫色)
6	YELLOW(黄色)
7	GREEN(暗い緑色)
8	TEAL(暗い水色)
9	NAVY(暗い青色)

各表示モードにおける動作について、以下で詳しく説明する。

3.4.2 Split View におけるガイド機能

Split View でガイド機能を表示している画面が図3.14上である。図3.14では、数字キー1にデスクトップの左上隅が、数字キー2に中央部、数字キー3に右下隅付近が登録されていることが示されている。ユーザは分割された画面上のガイド表示から、移動したい画面領域、あるいは移動したい領域に近い画面領域を探して、その画面領域に移動する。移動先の画面を表示する場所は画面下である。



図 3.14: Split View におけるガイド機能の使用例



図 3.15: Global View におけるガイド機能の使用例

ガイド機能を使わない場合は、ユーザ自身が登録した画面領域のイメージと割り当てた数字キーを全て覚えておく必要が生じたが、ガイド機能を使用することによりユーザの負担を減らすことができる。

3.4.3 Global View におけるガイド機能

Global View では、Split View と違い常時ガイドを表示している(図 3.15)。ユーザは、ガイドにより画面全体を通して登録した画面領域を知ることができる。また、全画面で画面領域を登録するとその結果がすぐガイドに反映され、どの領域を登録したのかをすぐ知ることが可能である。よって、ユーザの画面領域の登録作業がスムーズに行うことが可能になる。また、ユーザはガイド表示から事前に画面位置を登録した場所を確認して、登録した画面領域を呼び出すことによって、素早く画面領域枠を移動させることが可能である。

例えば，図 3.15 では遠隔計算機のデスクトップの左上が 1，右上が 2，左下が 3 に登録されており，中央付近にユーザが閲覧している現在位置があることを示している．ユーザが左に並んでいるアイコンを操作するには，1 か 3 の登録領域に画面領域枠を移動するとよいことが一見して分かる．

第4章 実装システム

この章では，携帯電話版 VNC システム (以下，本システム) について説明する．

4.1 システム構成

本システムは，VNC サーバ，SVNC プロキシ，SVNC ビューアという 3 つのシステムで構成されている．このうち，SVNC プロキシは Java Servlet として，SVNC ビューアは携帯電話上で Java アプリケーションとして動作する．SVNC プロキシの開発には Sun Microsystems の JDK 1.3.1 を，SVNC ビューアの開発には NTT DoCoMo の i モード Java[7] を利用した．VNC サーバについては，既存のアプリケーション VNC server for Win32 version 3.3.3 を利用した．

4.2 設定と起動

VNC サーバ，SVNC プロキシ，SVNC ビューアの 3 つのシステムの起動と設定について述べる．

VNC サーバ，SVNC プロキシ，SVNC ビューアは，携帯電話の SVNC ビューアから操作する遠隔計算機で起動しておく．VNC サーバは，ユーザ名，パスワード，ディスプレイ番号等の設定を行い，他の計算機の VNC ビューアから接続可能な状態にしておく．SVNC プロキシは，Servlet コンテナである Apache Tomcat 4.0 を起動して，SVNC ビューアからの要求に応じて SVNC プロキシを起動できるようにしておく．最後に，携帯電話上で SVNC ビューアの Java プログラムを起動する．SVNC ビューアの Java プログラムは，起動時に SVNC プロキシにネットワーク接続するよう設定をインストール時に行う．

以上のシステム設定と起動を行うことにより，SVNC ビューアから SVNC プロキシに接続し，SVNC プロキシを介して VNC サーバに接続できる．

4.3 CRFB プロトコル

この節では，本研究で提案する CRFB プロトコルについて述べる．CRFB プロトコルは，図 2.2 に示すように SVNC プロキシと携帯電話の SVNC ビューアの間の手続きを規定している．

CRFB プロトコルは

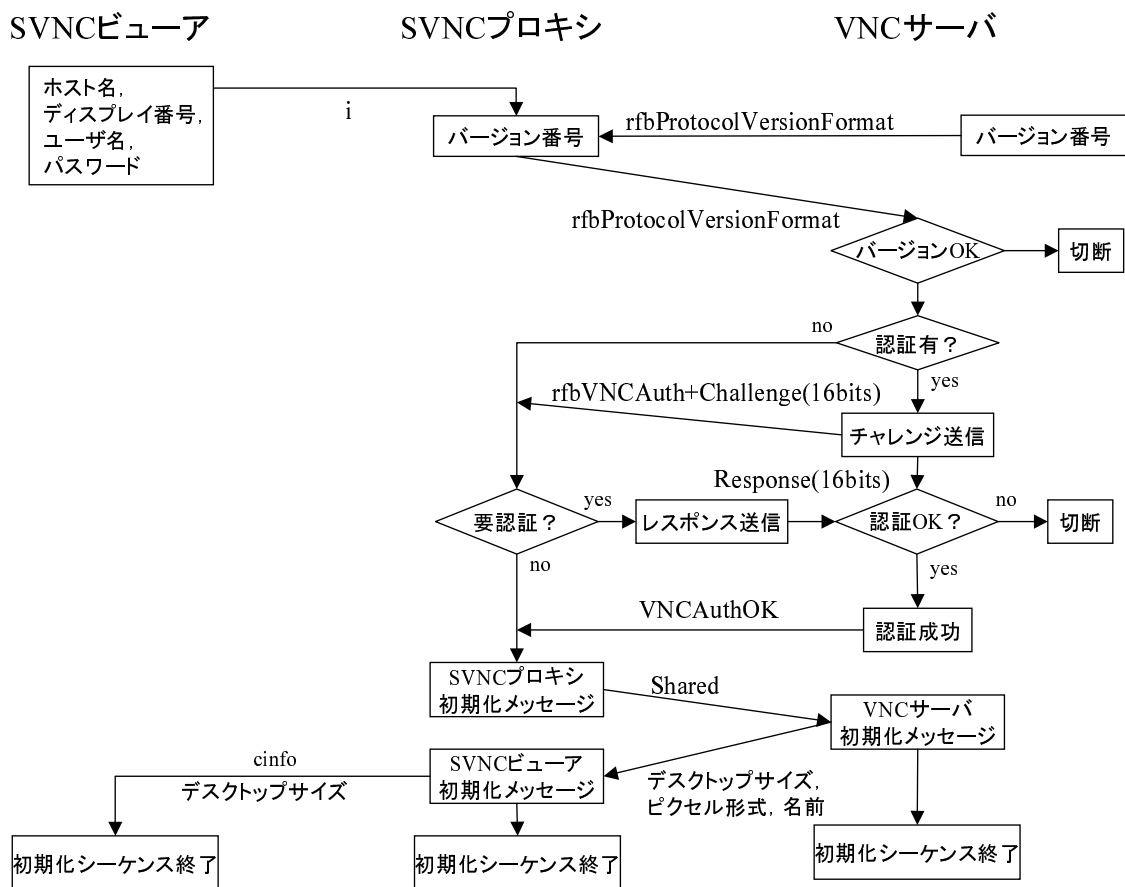


図 4.1: VNC セッションの初期化のイベントトレース

- VNC セッションの初期化
- SVNC ビューアからの入力イベントの通知
- SVNC プロキシから SVNC ビューアへの画面の通知

から構成される。以下の各節では、それぞれの過程、処理における CRFB プロトコルの詳細を述べる。

4.3.1 VNC セッションの初期化

VNC セッションの初期化まとめたイベントトレースが図 4.1 である。

VNC セッションは、SVNC ビューアが HTTP(Hyper Text Transfer Protocol) を利用して SVNC プロキシに接続することから始まる。

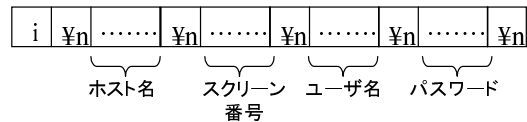


図 4.2: VNC サーバへの接続情報

SVNC ビューアから SVNC プロキシへの接続

SVNC プロキシに接続後，SVNC プロキシと VNC サーバの接続に必要な認証情報を SVNC ビューアが SVNC プロキシに伝える．まず，SVNC ビューアは SVNC プロキシに対して VNC セッションの初期化を始めるメッセージを以下のように通知する．

```
os.write(("i"+"\\n").getBytes());
```

続いて，SVNC ビューアは SVNC プロキシに対して，図 4.2 のように，接続する VNC サーバのホスト名，ディスプレイ番号，ユーザ名，パスワードを通知する．

SVNC プロキシは SVNC ビューアから認証情報を受け取ると，受け取った認証情報を元に VNC サーバに対して接続を開始する．SVNC プロキシから VNC サーバの通信には，RFB プロトコルを用いる．

プロトコルバージョンの交換

SVNC プロキシと VNC サーバの接続が確立すると，最初に RFB プロトコルのバージョンメッセージ (rfbProtocolVersionFormat) を交換する．RFB プロトコルバージョンメッセージは，12byte の ASCII 文字列の以下の形式である．

```
'RFB xxx.yyy\\n'
```

ここで，xxx は RFB プロトコルのメジャーバージョン，yyy は RFB プロトコルのマイナーバージョンを示す．

接続が確立すると，最初に VNC サーバがサポートするプロトコルバージョンを SVNC プロキシに通知する．SVNC プロキシはこのメッセージを受け取り，接続先が VNC サーバであることと，VNC サーバのバージョンを確認する．通知されたバージョンが利用可能であれば SVNC プロキシは，自分の側のプロトコルのバージョンのメッセージを VNC サーバに返す．

```
'RFB 003.003\\n'
```

SVNC プロキシは，RFB プロトコルの Version 003.003 を使用しているため，上記のようなメッセージを VNC サーバに通知する．

表 4.1: 認証方式

値	認証方式
0	接続失敗 (ConnFailed)
1	認証なし (NoAuth)
2	VNC 認証 (rfbVNCAuth)

認証

プロトコルバージョンの交換後，VNC サーバはSVNC プロキシに認証方式を通知する．認証方式は，表 4.1 のようになる．それぞれの認証方式に該当する値が，SVNC プロキシに対して通知される．

VNC 認証 (VNCAuth) を使用することを指示された場合には，VNC サーバから後続データとしてランダムに選ばれた 16bit の Challenge 値が送られる．SVNC プロキシは，SVNC ビューアから受け取ったパスワードによって暗号化された 16bit の Response 値を VNC サーバに送り返す．

VNC サーバは，SVNC プロキシからの Response 値と VNC サーバに登録されているパスワードを比較して，最終的に接続を受け付けるかどうかを判断して，その結果を SVNC プロキシに返す．接続を許可する場合は VncAuthOK，不許可の場合は VncAuthFailed である．

SVNC プロキシ初期化

認証なし (NoAuth) の場合と認証が成功 (VncAuthOK) の場合に，最初に SVNC プロキシが VNC サーバに送るメッセージ (rfbClientInitMsg) である．メッセージのメンバは，他のユーザにデスクトップの共有を許可するかを示す shared だけをメンバとして持っている．shared の値は，許可が 0，不許可が 1 である．実装システムでは，共有の選択をサポートしていないため，shared の値は不許可 (1) である．

VNC サーバの初期化

rfbClientInitMsg を受け取ると，VNC サーバはSVNC プロキシに対して初期化メッセージ (rfbServerInitMsg) を通知する．通知内容は VNC サーバのデスクトップのサイズ (幅，高さ)，ピクセル形式，名前の長さ，名前の文字列が通知される．

SVNC ビューアの初期化

rfbServerInitMsg を受け取ると，SVNC プロキシはSVNC ビューアに対して初期化メッセージ (cinfo) を通知する．VNC サーバのデスクトップのサイズ (幅，高さ) が通知される．

以上で、VNC セッションの初期化シーケンスが終了する。

4.3.2 入力イベントの通知

SVNC ビューアで発生した入力イベントは、マウスイベント、キーイベントとして SVNC プロキシに通知される。

マウスイベントの通知

操作モードがマウスポインタの時に発生したキーイベントを、マウスポインタイベントメッセージとして SVNC プロキシに通知される。まず、SVNC ビューアは SVNC プロキシに対して、マウスイベントを始めるメッセージを以下のように通知する。

```
os.writeln(("p"+"\\n").getBytes());
```

次に、マウスポインタのイベントの種類を通知する。マウスポインタのイベントの種類は、マウスポインタの移動 (set)、シングルクリック操作 (click)、ダブルクリック操作 (twoclick)、ドラッグ操作 (drag)、ドロップ操作 (drop) がある。イベントの種類によって、次に通知するメッセージが変わる。

set イベントの場合は、マウスポインタの位置 (x 座標, y 座標) を SVNC プロキシに通知する。

click イベントの場合は、マウスポインタの位置 (x 座標, y 座標) と押したマウスボタンの種類 (左ボタン (left) or 中ボタン (middle) or 右ボタン (right)) を SVNC プロキシに通知する。

twoclick イベントの場合は、マウスポインタの位置 (x 座標, y 座標) と押したマウスボタンの種類 (左ボタン (left) or 中ボタン (middle) or 右ボタン (right)) を SVNC プロキシに通知する。

drag イベントの場合は、マウスポインタの位置 (x 座標, y 座標) を SVNC プロキシに通知する。

drop イベントの場合は、ドロップする位置 (x 座標, y 座標) を SVNC プロキシに通知する。

SVNC プロキシは、SVNC ビューアからのマウスイベントメッセージを受け取ると、VNC サーバ向けのメッセージに変換する。変換したメッセージを VNC サーバに通知する。

キーイベントの通知

操作モードが、文字入力の時に発生したキーイベントを、キーイベントメッセージとして SVNC プロキシに通知される。まず、SVNC ビューアは SVNC プロキシに対して、キーイベントを始めるメッセージを以下のように通知する。

```
os.write(("t"+"\\n").getBytes());
```

次に，入力された文字列と文字列長を SVNC プロキシに通知する．

SVNC プロキシは，SVNC ビューアからのキーイベントメッセージを受け取ると，1文字ごと読み込み，キーイベントメッセージとして VNC サーバに通知する．この時，読み込んだ文字に，数字，英字，エスケープシーケンス以外の文字が含まれていたら，無視して VNC サーバには通知しない．

4.3.3 画面の通知

画面の通知は，SVNC ビューアからの通知要求と，それに対して SVNC プロキシからの画面更新データから構成される．

更新通知要求

SVNC ビューアから，更新通知要求として以下のようなメッセージを通知する．更新通知要求は，視点変更を行ったときなどの再描画を必要とする場合に以下のように通知する．

```
os.write(("s"+"\\n").getBytes());
```

続いて，画面領域の左上の座標 (x 座標，y 座標)，サイズ (幅，高さ)，拡大率を通知する．SVNC プロキシは SVNC ビューアから更新通知要求を受け取ると，VNC サーバに対して蓄えておいたデータを返す．

画面更新データ

VNC サーバは画面更新要求を SVNC プロキシから受け取ると，SVNC プロキシに対して画面更新データを通知する．SVNC プロキシは，VNC サーバから送られた画面更新データを通知された画面領域の情報を元に加工して，SVNC ビューア用の画面更新データとして SVNC ビューアに通知する．

4.3.4 画面領域登録の通知

画面領域の登録または画面領域の呼び出しを行うと，SVNC ビューアから画面領域登録メッセージとして SVNC プロキシに通知する．

```
os.write(("r"+"\\n").getBytes());
```

次に，画面領域の登録 (save) か画面領域の呼び出し (load) かを通知する．save か load かで，次に通知するメッセージが変わる．

save の場合は，登録した画面領域の情報を SVNC プロキシに通知する．登録画面領域の情報とは，登録画面領域の左上の座標 (x 座標，y 座標)，サイズ (幅，高さ) 拡

大率及び登録画面番号(1~9)である。SVNC プロキシは、この登録情報をファイルに保存する。

load の場合は、呼び出す登録画面番号を通知する。SVNC プロキシは登録情報を保存したファイルから登録画面番号に対応した画面領域の情報を読み出す。読み出した画面領域情報をもとに、蓄えたデスクトップの画像から切り出し縮小をした上で、SVNC ビューアに渡す。

4.4 スライドショー機能

SVNC プロキシは、VNC サーバのデスクトップ画像を一定の間隔で定期的に保存する。同時に、保存した時間と保存したファイル名をログファイルとして保存する。

スライドショー機能を実行すると、SVNC ビューアは、スライドショーをする画面領域の情報を SVNC プロキシに通知する。通知後、SVNC プロキシは保存した時間の最新 20 個についてを古い順にファイル名を読み出す。そして、読み出したファイル名に該当する画像から、スライドショーを実行する画面領域を切り出し縮小した上で、SVNC ビューアに渡す。

4.5 画面更新の頻度

実験に用いる携帯電話の下りパケット通信速度が 9600bps であることから通信のトラフィックを考慮し、SVNC プロキシが行う画面更新の頻度を最大 5 秒に 1 回に設定した。実際の使用環境には FOMA 対応携帯電話を考えており、下りパケット通信速度が 384kbps であるので、1 秒に 2 回の更新頻度に設定する予定である。

4.6 実験環境

SVNC ビューアを実行する携帯電話には、現行で Java プログラムが実行できることを踏まえて NTT DoCoMo の i モード Java 対応携帯電話を選択した。実際の使用環境には FOMA 対応携帯電話を考えている。実装中の SVNC ビューアの動作検証には i モード Java 対応携帯電話エミュレータである Zentek Technology の i-JADE 1.4.1 を使用した。

SVNC プロキシが動作する計算機は、Windows 2000 が動作するパーソナルコンピュータを使用した。SVNC プロキシを Servlet として実行するため、Servlet engine として Apache Tomcat 4.0 を選択した。

SVNC ビューアが遠隔操作する計算機には、VNC サーバが起動し他の計算機の VNC ビューアから接続可能な計算機を用意した。また、Windows 2000 が動作するパーソナルコンピュータを使用し、起動する VNC サーバには AT&T Research Labs Cambridge の VNC server for Win 32 version 3.3.3 を選択した。

4.7 動作確認

実装したプロトタイプを動作させ、携帯電話のSVNCビューアからSVNCプロキシに接続し、VNCサーバに接続を確認した。次に、遠隔計算機のデスクトップを携帯電話の画面上に表示を確認した。さらに、遠隔計算機にあるアプリケーションの起動や終了等の一般的な操作が携帯電話の入出力インタフェースを使って行えることを確認した。また、Normal View、Split View、Global Viewの各表示モードを使って、閲覧、編集、比較の作業を実際に行い動作を確認した。

4.7.1 操作例：Windowsの終了操作

本システムを用いて、Windows 2000が動作する遠隔計算機の「Windowsの終了操作」を実行する過程を、それぞれの操作に要するキー操作回数とともに、以下に示す。

1. 現在の状態 (作業開始)



図 4.3: 現在の状態

図 4.3 は、Global View の表示モードでガイドを表示している場面である。ユーザは、この状態から「Windowsの終了操作」を実行する。

2. 「スタートボタン」の画面領域の表示



図 4.4: 登録された画面領域への移動

図 4.4 は、「スタートボタン」が含まれる左下の 3 番の登録領域に数字キー 3 を押して、画面領域枠を移動した場面である。これにより、Normal View の表示モードで表示される画面領域が、3 番の登録領域になる。以上のキー操作数は 1 回である。

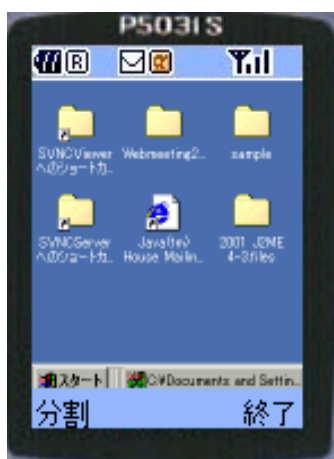


図 4.5: 表示モードの変更

図 4.5 は、Normal View の表示モードに切り替えて、3 番の画面領域を表示した場面である。以上のキー操作数は 1 回である。



図 4.6: 画面領域の拡大

図 4.6 は、画面領域の拡大率を 1/2 倍から 1 倍に変更して、画面領域を拡大した場面である。以上のキー操作数は 1 回である。



図 4.7: 視点変更

図 4.7 は、視点変更モードに切り替えて、スタートボタンが見える位置になるまで、下方向に視点を移動した場面である。以上のキー操作数は 2 回である。

3. 「スタートボタン」の選択と「スタートメニュー」の操作



図 4.8: 操作モードの変更

図 4.8 は、操作モード切替に割り当てたキーを押し、マウスポインタモードに切り替えた場面である。画面領域の中央付近に、仮想ポインタが表示される。以上のキー操作数は 1 回である。



図 4.9: マウスポインタの移動

図 4.9 は、仮想ポインタを「スタートボタン」の位置まで移動した場面である。仮想ポインタを移動すると、それに追従して遠隔計算機のマウスポインタも移動する。以上のキー操作数は 3 回である。



図 4.10: マウスクリック操作

図 4.10 は、「スタートボタン」の位置で、マウスクリック操作に割り当てたキーを押し、遠隔計算機の「スタートボタン」をクリックした場面である。以上のキー操作数は1回である。



図 4.11: マウスポインタの移動

図 4.11 は、「スタートメニュー」の「シャットダウン (U)」に仮想ポインタを移動した場面である。遠隔計算機の「スタートメニュー」の「シャットダウン (U)」が選択状態になる。以上のキー操作数は1回である。



図 4.12: マウスクリック操作

図 4.12 は、「シャットダウン (U)」の位置で、マウスクリック操作に割り当てたキーを押し、遠隔計算機の「シャットダウン (U)」をクリックした場面である。以上のキー操作数は 1 回である。

4. 「Windows のシャットダウン」画面領域の表示



図 4.13: 画面領域の縮小

図 4.13 は、「Windows のシャットダウン」のウィンドウが SVNC ビューアに表示されるまで、画面領域の拡大率を変更して、画面領域を縮小した場面である。ユーザーはこの操作によって、「Windows のシャットダウン」のウィンドウが遠隔計算機のデスクトップ上の中央に表示されていることを確認する。以上のキー操作数は 5 回である。



図 4.14: 視点変更

図 4.14 では、操作モード切替に割り当てたキーを押し、視点変更モードに切り替える。そして、上方向、右方向に視点の移動を行い、画面中央の「Windows のシャットダウン」のウィンドウが表示された状態である。以上のキー操作数は 3 回である。

5. 「Windows のシャットダウン」の操作



図 4.15: 操作モードの変更

図 4.15 は、操作モード切替に割り当てたキーを押し、マウスポインタモードに切り替えた場面である。画面領域の中央付近に、仮想ポインタが表示される。以上のキー操作数は 1 回である。



図 4.16: マウスクリック操作

図 4.16 は、「シャットダウンメニュー」の位置で、マウスクリック操作に割り当てたキーを押し、遠隔計算機の「シャットダウンメニュー」をクリックして開いた状態である。以上のキー操作数は2回である。



図 4.17: マウスポインタの移動

図 4.17 は、「シャットダウンメニュー」の「シャットダウン」に仮想ポインタを移動した場面である。遠隔計算機の「シャットダウンメニュー」の「シャットダウン」が選択状態になる。以上のキー操作数は1回である。



図 4.18: マウスクリック操作

図 4.18 は、「シャットダウンメニュー」の「シャットダウン」の位置で、マウスクリック操作に割り当てたキーを押し、遠隔計算機の「シャットダウンメニュー」の「シャットダウン」をクリックした場面である。以上のキー操作数は 1 回である。



図 4.19: マウスポインタの移動

図 4.19 は、「Windows のシャットダウン」ウィンドウの OK ボタンに仮想ポインタを移動した場面である。以上のキー操作数は 2 回である。



図 4.20: 画面領域の縮小とマウスクリック操作

図 4.20 は、遠隔計算機のデスクトップ全体が SVNC ビューアの画面領域に表示されるまで、画面領域の拡大率を変更して、画面領域を縮小した場面である。ユーザは、これにより、遠隔計算機のデスクトップ全体を閲覧しながら、Windows をシャットダウンできる。この状態でマウスクリック操作に割り当てたキーを押すことで、遠隔計算機の「Windows のシャットダウン」の OK ボタンが押され、Windows がシャットダウンされた。以上のキー操作数は 4 回である。

結果

携帯電話から遠隔計算機の「Windows の終了操作」をキー操作数 31 回で行うことができた。

第5章 関連研究

5.1 携帯電話やPDAを用いた遠隔操作

携帯電話およびPDAを用いて、遠隔の計算機、および種々の情報機器を操作する試みは数多く行われている。

5.1.1 コンテンツ変換に基づいた遠隔操作

上向らは、iモード携帯電話からWWWブラウザを遠隔制御するシステム[8]を提案している。[8]では、ブラウザに表示されているWWWページの構造を解析し、解析結果に基づいて操作(リンクの選択、フォームへの入力、およびスクロールなど)ページをiモード携帯電話向けに生成し、携帯電話に送信する。ユーザが操作ページを操作(ページ中のリンクをクリック)すると、CGIが起動され、CGIによってその操作がWWWブラウザに反映される。このシステムの他にも、コンテンツ変換を行うことにより情報家電をiモード携帯電話から制御するためシステムとして[9][10]などが提案されている。

岡田らは、デスクトップに表示されているGUIを携帯電話から制御するために、GUIの構造を解析してiモード携帯電話に表示可能なようにテキスト化するコンテンツ変換を実現した[11]。ユーザは、iモード携帯電話を用いて、GUI部品の単位でデスクトップを操作することが可能である。

上記のアプローチは、計算機が表示するコンテンツ、あるいは情報機器を制御するためのコンテンツを、それぞれのモデルに則して変換し、携帯電話に転送する方式を取っている。結果として、モデルに合致しないコンテンツの制御は不可能となっている。

5.1.2 画像に基づいた遠隔操作

PalmVNC[12]は、PDAを用いて遠隔計算機のデスクトップを制御するためのシステムとして興味深い。このシステムは、PDAの小画面に合わせて遠隔計算機のデスクトップのイメージの縮小版を生成し転送する機能、および特殊なコントロールコードを生成するための機能を持つ。本研究は携帯電話を対象としており、PDAと比べて更に小画面で、入力デバイスもスタイラスではなくキーのみである。その対策として本研究では新しいインタフェースを提案している点が異なる。

本研究は、携帯電話の小画面に対して広大なデスクトップ画面を制御するための

様々な処理をプロキシに付与している点、およびネットワークトラフィックの削減等のためにプロキシを用いているが、画像変換を行うプロキシを用いている点において、本研究のアーキテクチャは長谷川らが提案するアーキテクチャ[13, 14]と同様である。このアーキテクチャでは、ネットワーク経由で情報家電を制御することを目的とし、PDA や i モード携帯電話に適した操作形態のインタフェースを生成する VNC プロキシを用意している。VNC プロキシは、情報家電を制御する GUI 画面を読み取り、PDA に対して減色・縮小処理、携帯電話に対してはテキスト版の操作ページを生成する等のコンテンツ変換を行う。

Su らは、携帯電話から遠隔計算機を操作するシステムを提案している [15]。遠隔計算機のデスクトップを携帯電話に描画している点と文字入力と特殊キーの入力操作ができる点は、本システムと Rajicon は似ている。しかし、本研究は VNC を用いた汎用システムを目標としている点、および本研究の提案する 2 画面同時作業やスライドショー機能を実現している点が異なる。

Desktop On-Call [16] は、VNC と同様に遠隔計算機をパーソナルコンピュータや PDA から操作可能にするアプリケーションである。携帯電話のブラウザから、遠隔計算機のデスクトップを横方向に 5 分割したそれぞれの部分を閲覧することができる。しかし、携帯電話から遠隔計算機を操作することを目的としていないため、本システムが提供するようなマウスポインタの操作などが実現されていない点が異なる。

5.2 小画面インタフェース

携帯電話および PDA に代表される小画面を用いたユーザインタフェースに関する研究は数多く行われている。

5.2.1 小画面インタフェースの議論

Matt Jones らは、携帯型計算機上で従来の大画面向けに設計された Web サイトを閲覧した際にどのような影響があるのかの実験を行っている [6]。この実験では、大画面と小画面のブラウザを利用する 2 つのユーザを作り、同じサイトデザインのサイトを閲覧してもらい、ユーザにはタスクを与え、一定時間内に答えを導いてもらう。実験結果、大画面のほうが小画面よりも 2 倍多い正答を得た。アンケートでは、小画面ユーザの 80 % が画面サイズが小さいので仕事のパフォーマンスが妨げられたという報告がある。小画面向けの Web サイトの設計方法として、次の点を提案している。

- 直接アクセスを提供する
- スクロール量を減らす

本研究では、数字キーを用いてユーザが記憶させた場所に移動できる機能を提供している。これにより、本システムでは上記 2 点を解決している。

第6章 結論

携帯電話を用いて遠隔情報機器を利用する手段として、ビットマップベースの画面転送を有する VNC システムを用い、携帯電話が備える入出力機器のみを用いた操作体系を適切な形態で提供するプロキシを VNC サーバと携帯電話の間に挟むシステムを提案した。また、携帯電話が備える入出力インタフェースを用いて、広大なデスクトップを制御する際に生じる操作上の困難を軽減するインタフェースとして、画面領域の登録と表示モードの提案を行った。さらに、提案したシステムのプロトタイプを i モード Java 対応携帯電話に実装した。今後の課題は、実装したシステムの評価を行い、新たな問題点を探り、インタフェースの改善を行うことである。特に、現在は数字キーに画面領域を登録する機能を提供しているが、これに文字入力やポインタ操作等を組み合わせた操作列をマクロとして登録できるような仕組みを考えている。

謝辞

本研究を進めるにあたり、ご指導下さった指導教官の田中二郎教授には心から感謝いたします。志築文太郎助手、Web グループリーダーの三浦元喜助手には本研究の方向性やシステムの実装、論文作成についてさまざまな点についてご助言を頂き、深く感謝いたします。また、田中研究室の皆様にも多くのご助言や励ましの言葉を頂き、この場を借りて感謝いたします。

参考文献

- [1] 中須正人, 志築文太郎, 田中二郎. 携帯電話版 VNC システム. インタラクシオン 2002 論文集 (掲載予定). 情報処理学会, March 2002.
- [2] Tristan Richardson, Quentin Stanfford-Fraser, Kenneth R. Wood, and Andy Hopper. Virtual Network Computing. *IEEE Internet Computing*, Vol. 2, No. 1, pp. 33–38, January/February 1998.
- [3] Tristan Richardson and Kenneth R. Wood. The RFB Protocol Version 3.3. January 1998.
- [4] Tomohito Haraikawa, Tadashi Sakamoto, Tomohiro Hase, Tadanori Mizuno, and Atsushi Togashi. μ VNC: A Proposal for Internet Connectivity and Interconnectivity of Home Appliances based on Remote Display Framework. *IEEE Transactions on Consumer Electronics*, Vol. 47, No. 3, pp. 512–519, August 2001.
- [5] Lars Erik Holmquist. Will Baby Faces Ever Grown up? In *Proceedings of Human-Computer Interaction International (HCII)*, pp. 706–709, August 1999.
- [6] Matt Jones, Gary Marsden, Norliza Mohd-Nasir, and Kevin Boone. Improving Web Interaction on Small Displays. In *Proceedings of 8th International World Wide Web Conference*, pp. 51–59, May 1999.
- [7] 株式会社 NTT ドコモ. i モード対応 Java API 向け Java™2 Micro Edition Wireless SDK ユーザガイド, 2000.
- [8] 上向俊晃, 萩野浩明, 原隆浩, 塚本昌彦, 西尾章治郎. リモートディスプレイ環境における WWW ブラウジングシステム. 情報処理学会論文誌, Vol. 41, No. 9, pp. 2364–2373, September 2000.
- [9] Olivier Liechti and Kenji Mase. iRemote: a Platform for Controlling Public Displays with Mobile Phones. インタラクシオン 2001 論文集, pp. 35–36. 情報処理学会, March 2001.
- [10] インプローブ・ネットワークス. CAFEMOON@ HOME. [http:// www.cafemoon.org/](http://www.cafemoon.org/).

- [11] 岡田英彦, 加藤清志, 池上輝哉, 辰巳勇臣, 旭敏之. 携帯端末による PC 遠隔操作システムの提案. 情報処理学会研究報告 2001-HI-93, pp. 1–6. 情報処理学会, May 2001.
- [12] Harakan Software. PalmVNC: Virtual Network Computing Client for Palm Platform. [http:// www. harakan. btinternet. co. uk/ PalmVNC/](http://www.harakan.btinternet.co.uk/PalmVNC/).
- [13] Atsushi Hasegawa and Tatsuo Nakajima. A User Interface System for Home Appliances with Virtual Network Computing. In *IEEE International Workshop on Smart Appliances and Wearable Computing (IWSAWC2001)*, April 2001.
- [14] 長谷川温, 中島達夫. 家電機器向けユーザインタフェースシステム. SPA2001 予稿集, March 2001.
- [15] Norman Su, Masahiko Tsukamoto, and Shojiro Nishio. Rajicon: A System for Remote PC Access. マルチメディア, 分散, 協調とモバイル (DICOMO2001) シンポジウム, pp. 349–354, June 2001.
- [16] IBM Corporation. Desktop On-Call 「PC リモコン」 Version 5 ユーザーズ・ガイド第 1.0 版, November 2001.