

通信遅延を考慮したセンサ選択手法

鏡 慎吾^{†a)} 石川 正俊[†]

A Sensor Selection Method Considering Communication Delays

Shingo KAGAMI^{†a)} and Masatoshi ISHIKAWA[†]

あらまし ネットワークで接続されたセンサの中から、対象の情報を効率良く得ることができるようなセンサを選択するための手法について、特にネットワークを介して情報を受け取る際の通信遅延の存在を明示的に取り扱った形で議論する。まず、通信遅延を考慮しながら複数のセンサからの情報をカルマンフィルタで統合する際の、センサ数の増大に関してスケーラブルな計算アルゴリズムを提案する。複数のセンサがそれぞれ異なる通信遅延をもち、またその遅延が変動して、結果としてセンサ情報が実際の時刻とは順不同で到着する場合でも、それらの情報を正しく統合できるように構成されている。次いで、このアルゴリズムによる状態推定を行う際に、適切なセンサを選択するための手法を提案する。この手法は、センサを選択する基準として従来から用いられている相互情報量を利用しているが、現在時刻の推定値にもたらされる情報量が通信遅延によって減少する点を、定量的に表現できている点を特徴とする。これらのアルゴリズムを数値実験で検証し、その有効性を示す。

キーワード センサフュージョン、カルマンフィルタ、相互情報量、センサネットワーク、分散センシング

1. ま え が き

無線通信やモバイルネットワークといったネットワーク技術、並びに VLSI や MEMS といったデバイス技術の急速な進歩を背景として、多数のセンサをネットワークで接続し、その膨大な情報を利用するための研究が盛んに行われている [1], [2]。

特に近年は、例えば工場やプラント等で従来から用いられてきたような特定用途の閉じたセンサネットワークとは一線を画す、開放性や相互接続性をもったセンサネットワークを実現するための技術が注目を集めている。センサノードのデバイス、通信機構、ネットワークアーキテクチャやソフトウェアアーキテクチャなど、様々な側面からの研究が精力的に行われている [3] ~ [5]。

このように、膨大な数の様々なセンサが接続されたセンサネットワークを有効に利用するためには、そこで行われるセンサ情報処理に関しても新たな技術の導入が必要とされる。その中でも重要な課題として、あ

るセンシングタスクをセンサネットワーク内にどのようにマッピングするか、すなわち、活性化すべきセンサの集合をどのように定め、それらから得られる情報をどのように統合するかという問題が挙げられる。

この問題の解が自明でない理由の一つとして、単に「より良いセンシングを行う」ことを目指すだけではなく、計算量、通信量、通信遅延や電力消費などといった様々な条件を考慮する必要があることが挙げられる。例えば、センシングの精度のみを追求して、考え得るすべてのセンサを活性化してその全情報を利用することは、ネットワークの帯域幅の制限やセンサノードのバッテリー利用効率などを考えると、必ずしも望ましくない。個々のセンサの価値を正しく評価して、活性化すべきセンサの集合を適切に定めることが重要となる。

本論文ではこの問題を、特にネットワークを介して情報を受け取る際の通信遅延の存在に焦点を当てて議論する。ユーザが様々なセンサ情報を、それぞれの遅延を伴って、しかも順不同で得るような状況下において、センサ選択を適切に行うためには、これらの通信遅延に関する情報も考慮した上で、各センサの価値を評価する必要がある。

本論文が取り扱うのは、何らかのセンシング対象の状態を実時間で推定する問題である。この場合、いか

[†] 東京大学大学院情報理工学系研究科，東京都
Graduate School of Information Science and Technology,
University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo, 113-
8656 Japan

a) E-mail: shingo_kagami@ipc.i.u-tokyo.ac.jp

に精度の良いセンサでも、その情報が到着するまでに長い遅延があるようでは使いものにならない。逆にネットワーク的に近く遅延が小さなセンサでも、その精度があまりに悪いようでは、その重要性は薄れてしまう。このようなトレードオフは、例えばネットワーク的に遠いセンサから観測しやすい位置に対象がある場合や、移動ロボットにおいて、自身に装備されたセンサは小型で低精度であるが、インフラに設置された高精度なセンサにネットワークを介してアクセスできる場合など、システムや環境のレベルの条件によってしばしば発生する。これらの問題を包括的に取り扱いたいという点に、本研究の動機がある。

以下ではこのような実時間センシングの枠組を実現するための基礎技術を議論する。まず 2. で、複数のセンサとユーザ間の通信遅延を許容しながら各センサの情報をカルマンフィルタで統合する際の、効率的なアルゴリズムを導入する。3. ではそれをベースとして、遅延を伴う観測がもたらす情報量を評価するアルゴリズムを提案し、それによるセンサ選択を議論する。4. にて、数値実験による検証を行う。

2. 遅延のある環境下でのカルマンフィルタ

2.1 関連研究と基本方針

遅延がある場合のカルマンフィルタリングという問題は、到着していない観測情報は推定に用いることはできないため、「ある時刻における対象の状態の、その時刻までに到着したすべての観測情報に基づく線形最小分散推定量を求める問題」であると位置づけることができる。

通常カルマンフィルタは 1 時刻ごとの再帰的な形をしており、前の時刻の最適推定値から、対象が従うダイナミクスのモデルを用いて予測を行い（時間更新）、この時間更新値を現時刻の観測情報を用いて更新する（観測更新）ことで、現時刻の最適推定値を得るという流れで処理が行われる。観測情報が遅延なく得られることを前提としているため、遅延がある場合にそのまま適用することはできない。

センサからの観測情報が到着した際に、その情報の観測時刻までの推定を行い、その時刻から現在時刻までは予測のみを行うことで、部分的に遅延を取り扱うことができるが、観測情報が順不同で届く場合には、遅れて到着した情報を統合することができない。

系ら [6] はこのように順不同に到着する観測情報にも対処できるように拡張したカルマンフィルタを提案

している。しかし同手法では観測値を 2 個以上さかのぼることはできず、通信遅延がサンプリング間隔に対して十分に小さい場合にしか適用できない。

また、現実的なアプローチとしては、各センサの観測モデルに遅延を組み込んでしまうという手法がとられることもある。例えば江村らはサンプリングレートが異なる複数のセンサの情報をカルマンフィルタで統合する手法を提案しているが [7]、ここでは遅延は観測モデルによって吸収可能であるとして、特別扱いはしていない。このアプローチは、遅延が既知で固定であるとみなせる場合は有効であるが、そう仮定できない場合には用いることができない。また、実際には過去のものである観測情報を用いて現在時刻の推定をするため、その分の精度の低下を観測ノイズとして定量化する必要があるが、その方法も明確には与えられていない。

一方、1 時刻前の推定値から計算を始めるのではなく、 L 時刻前の推定値から計算を始めるように変更すると、最大で $L - 1$ 時刻分の遅延を許容できるようになる。

しかし、この方法を単純に行った場合の問題点は、系ら [6] も指摘しているように記憶量と計算量が膨大となり得る点にある。センサから得られた観測情報を、それが現在時刻から数えて L 時刻以上古くなるまでは記憶しておく必要があり、その記憶されているすべての観測情報による観測更新を、毎ステップに繰り返さなくてはならない。そのため、センサの数の増大に対応することが困難である。

前山ら [8] は、カルマンフィルタに基づいた遡及的な情報統合アルゴリズムを提案しており、遅延を伴って到着する観測情報を計算量の増大を防ぎながら統合することを可能としている。この手法は、移動ロボットに備えられた複数のセンササブシステムからの情報を統合するために導入されたものであり、センサ数に関するスケーラビリティよりも、許容し得る最大遅延に関するスケーラビリティに主眼を置いている。各センサが観測を行った瞬間が検出可能であることを前提とし、観測時刻によって時間更新を分割することによって遡及的な情報統合を可能としているため、いつ観測が行われたか分からない—極限的には全時刻において観測がなされているとみなせる—大規模センサネットワークへの適用に向けたものではない^(注1)。

(注1): 逆に、本論文が提案する手法は許容最大遅延に関するスケーラビリティはなく、これが重要となる応用には不向きである。

センサネットワークにおける情報統合に対するカルマンフィルタの適用例としては、Durrant-Whyteらの脱集権型カルマンフィルタに関する一連の研究が挙げられる [9] ~ [11] . これは並列カルマンフィルタ [12] をベースとして、そこで仮定されていた中央プロセッサの存在を取り除き、脱集権化したものと位置づけることができる .

Rao ら [10] による最初の脱集権化では、すべてのノードが互いに全結合されたトポロジーが仮定されていた . Grime ら [11] は、ノード間の通信時に情報を選別するためのフィルタ (チャネルフィルタと呼ばれる) を追加することで、トリー型トポロジーにこれを拡張した .

これらのいずれにおいても、ノード間の通信には遅延がないことが前提とされている . ただし、Grime らの手法は全結合を仮定しておらず、したがって観測情報は一般に複数のノードを経由して、すなわち遅延を伴って到着することが許されており、遅延のある状況での最適推定が実現されている .

しかしこれは、経由する各ノードにおける予測・更新フィルタの処理と、ノード間通信におけるチャネルフィルタの処理によって実現されているため、これらの処理が組み込まれていないノード群を経由して情報の伝達を行うことは不可能である . そのためこの手法の適用は、閉じた系での専用センサネットワークに限られる .

例えば、TCP/IP に代表される汎用ネットワークを途中で経由する場合を考えると、途中ノードでは単なるパケット転送しか行えないため、この手法は適用できない . TCP/IP に限らなくとも、一般にセンサとユーザとの間の通信経路が両者の管轄外であるようなケースや、パケット転送がハードウェアによる固定の処理として実現されているケース、あるいは無線通信のように、そもそも 1 ホップ分の遅延が無視できないケースなど、これが問題となる場合は多く存在する .

そこで、以下ではこの点を克服する手法を提案する . 提案手法では、センサからユーザまでの情報伝達の経路には単なるデータ転送以外の処理を仮定せずに、最適フィルタリングを実現する .

2.2 アルゴリズム

通常カルマンフィルタ [13] の仮定から議論を始める . 時刻 k における対象の状態 $x(k)$ と観測値 $y(k)$ が、以下のモデルに従うとする .

$$\begin{aligned} x(k+1) &= A(k)x(k) + B(k)u(k) \\ y(k) &= C(k)x(k) + w(k) \\ E[u(k)] &= \bar{u}(k), \quad E[w(k)] = \bar{w}(k) \\ E[(u(k) - \bar{u}(k))(u(l) - \bar{u}(l))^T] &= \delta_{kl}U(k) \\ E[(w(k) - \bar{w}(k))(w(l) - \bar{w}(l))^T] &= \delta_{kl}W(k) \end{aligned}$$

ここで、 $u(k)$ と $w(k)$ はそれぞれ時刻 k におけるダイナミックスのノイズと観測ノイズを表し、 $u(k)$ 、 $w(l)$ と $x(0)$ は互いに独立とする . $E[\cdot]$ は期待値を、 δ_{ij} はクロネッカーのデルタをそれぞれ表す .

よく知られているとおり、時刻 $k-1$ までのすべての観測に基づく、時刻 k における対象の状態の最適な推定値 $\hat{x}(k | k-1)$ は、

$$\begin{aligned} \hat{x}(k | k-1) &= A(k-1)\hat{x}(k-1 | k-1) \\ &\quad + B(k-1)\bar{u}(k-1) \\ P(k | k-1) &= A(k-1)P(k-1 | k-1)A^T(k-1) \\ &\quad + B(k-1)U(k-1)B^T(k-1) \end{aligned}$$

と表される時間更新により求められる . また、時刻 k までのすべての観測に基づく、時刻 k における対象の状態の最適な推定値 $\hat{x}(k | k)$ は、

$$\begin{aligned} K(k) &= P(k | k-1)C^T(k) \{W^{-1}(k) \\ &\quad + C(k)P(k | k-1)C^T(k)\}^{-1} \\ P(k | k) &= P(k | k-1) \\ &\quad - K(k)C(k)P(k | k-1) \quad (1) \\ \hat{x}(k | k) &= \hat{x}(k | k-1) + K(k) \{y(k) \\ &\quad - (C(k)\hat{x}(k | k-1) + \bar{w}(k))\} \quad (2) \end{aligned}$$

または

$$\begin{aligned} P(k | k) &= \{P^{-1}(k | k-1) \\ &\quad + C^T(k)W^{-1}(k)C(k)\}^{-1} \quad (3) \\ \hat{x}(k | k) &= \hat{x}(k | k-1) \\ &\quad + P(k | k)C^T(k)W^{-1}(k) \{y(k) \\ &\quad - (C(k)\hat{x}(k | k-1) + \bar{w}(k))\} \quad (4) \end{aligned}$$

で表される観測更新により求められる . ここで $P(i | j)$ は、時刻 j までのすべての観測が与えられたときの、時刻 i での推定誤差共分散を表す . すなわち、

$$P(i | j) \equiv E[(\hat{x}(i | j) - x(i))(\hat{x}(i | j) - x(i))^T]$$

である。

観測が m 個のセンサによるものであるとし、センサごとの観測機構に分割する。ここで、異なるセンサに乗るノイズは無相関であることを仮定する。

$$y_i(k) = C_i(k)x(k) + w_i(k) \quad (i = 1, 2, \dots, m)$$

$$y(k) = [y_1^T(k), y_2^T(k), \dots, y_m^T(k)]^T$$

$$C(k) = [C_1^T(k), C_2^T(k), \dots, C_m^T(k)]^T$$

$$w(k) = [w_1^T(k), w_2^T(k), \dots, w_m^T(k)]^T$$

$$W(k) = \text{blockdiag} \{W_1(k), W_2(k), \dots, W_m(k)\}$$

この記法では、形式的に各センサが同一レートで同期的に観測するとしている。このうち、同一レートであるという条件は実際には必要なく、全センサの観測時刻列を表現できる最小クロックでシステムを記述すればよい。各センサは、実際に観測を行った時刻にのみ情報をユーザに送信する。観測周期の違いによる観測精度の違いは、観測ノイズとして個々に表現することになる。一方、各センサの時刻が同期していることは実際に必要となるが、時刻の同期に関しては多くの研究がなされており [14]、本論文では議論しない。

上記のようにセンサごとに分割された記法を用いると、式 (3), (4) の観測更新は以下のように表せる。

$$P^{-1}(k | k) = P^{-1}(k | k - 1) + \sum_{i=1}^m C_i^T(k)W_i^{-1}(k)C_i(k) \quad (5)$$

$$P^{-1}(k | k)\hat{x}(k | k) = P^{-1}(k | k - 1)\hat{x}(k | k - 1) + \sum_{i=1}^m C_i^T(k)W_i^{-1}(k) \{y_i(k) - \bar{w}_i(k)\} \quad (6)$$

このように共分散行列の逆行列を基本的な量として表した形は、information form のカルマンフィルタ、あるいは information filter として古くから知られており [15]、並列カルマンフィルタ等でも積極的に利用されている。

この形式をもとにして、遅延を許容するカルマンフィルタの処理を行うことを考えると、観測更新の 1 時刻分の処理を行うには、式 (5) と式 (6) に総和の形で現れている各センサによる観測情報の、各々ではなく総和値（それぞれ行列とベクトル）だけが必要であることが分かる。すなわち、各時刻に対応したバッファを許容したい最大遅延時間の時刻数だけ用意し、センサからの観測情報が到着することに、対応する時刻の

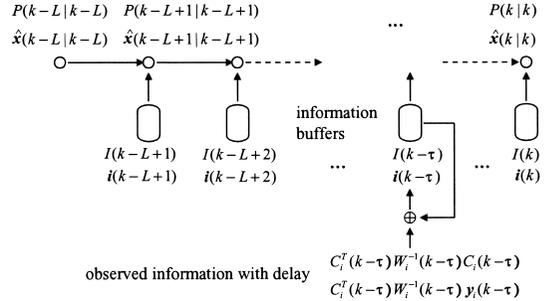


図 1 通信遅延を許容するカルマンフィルタの動作
Fig. 1 Operation of the delay-tolerant Kalman filter.

バッファにこれを加算しておけばよい。

以上の議論により、通信遅延を許容するカルマンフィルタのアルゴリズムは以下のように記述できる。その動作を図 1 に模式的に示す。

[Algorithm 1] 各センサはセンシングを行うと、状態情報 $C_i^T(k)W_i^{-1}(k) \{y_i(k) - \bar{w}_i(k)\}$ と誤差共分散情報 $C_i^T(k)W_i^{-1}(k)C_i(k)$ を、観測時刻のタイムスタンプとともにユーザに送信する。

ユーザ側には、現時刻 k からさかのぼって L 時刻分の状態情報バッファ $i(j)$ と誤差共分散情報バッファ $I(j)$ （それぞれ $j = k - L + 1, \dots, k$ ）を用意する。フィルタリングの 1 ステップの処理は以下に行われる。

(1) センサからの情報を受け取り、タイムスタンプ j に対応するバッファに加算する：

$$i(j) := i(j) + C_i^T(j)W_i^{-1}(j) \{y_i(j) - \bar{w}_i(j)\}$$

$$I(j) := I(j) + C_i^T(j)W_i^{-1}(j)C_i(j)$$

これを、その時刻に到着する情報の分だけ繰り返す。ただし $j < k - L + 1$ の場合はこれを破棄する。

(2) $\hat{x}(k - L | k - L)$ から $\hat{x}(k | k)$ まで、1 時刻ずつ時間更新・観測更新を繰り返す。すなわち

```
for  $j = k - L + 1, k - L + 2, \dots, k$  begin
     $P(j | j - 1)$ 
        :=  $A(j - 1)P(j - 1 | j - 1)A^T(j - 1)$ 
        +  $B(j - 1)U(j - 1)B^T(j - 1)$ 
     $\hat{x}(j | j - 1)$  :=  $A(j - 1)\hat{x}(j - 1 | j - 1)$ 
        +  $B(j - 1)\bar{u}(j - 1)$ 
     $P(j | j)$  :=  $\{P^{-1}(j | j - 1) + I(j)\}^{-1}$ 
     $\hat{x}(j | j)$  :=  $P(j | j) \{P^{-1}(j | j - 1)\hat{x}(j | j - 1)$ 
        +  $i(j)\}$ 
end
```

を行う。またこの計算の途中で現れる $\hat{x}(k-L+1 | k-L+1)$ と $P(k-L+1 | k-L+1)$ を次のステップのために記憶しておく。他の時刻の推定値・誤差共分散は記憶する必要はない。

(3) バッファ $i(k-L+1)$ と $I(k-L+1)$ の記憶領域を破棄する。新たにバッファ $i(k+1)$ と $I(k+1)$ の記憶領域を割り当て、それぞれ零ベクトル $\mathbf{0}$ 、零行列 \mathbf{O} で初期化する。□

許容したい最大遅延を表すパラメータ L は、様々な要素に応じて定める必要がある。ネットワークの遅延発生状況によって、また推定対象のダイナミクスやセンサの構成、アプリケーションの要求などに応じて、考慮する価値のある L の上限が影響される。一方、後述するように、このアルゴリズムの計算量は L に比例して増加するため、要求を満たす範囲でできるだけ小さくするのが望ましい。

以降ではこのアルゴリズムを DTKF (Delay-Tolerant Kalman Filter) と略称する。

2.3 計算量の評価

提案したアルゴリズムの計算量を示す。推定する対象の状態を n 次元とし、簡単のため、 m 個のセンサの観測ベクトルはすべて p 次元とする。すべてのセンサが毎時刻に一つの観測情報を送信するとすると、Algorithm 1 の (1) は、1 時刻当たり平均 m 回繰り返されると考えられる。以上の前提のもとで、1 時刻の推定を得るためにユーザ側で実行される浮動小数点演算回数を評価した。

$n = 6, p = 2$ と固定し、 L と m を $L = 5, 10, \dots, 25$ 及び $m = 1, 2, \dots, 40$ と変化させた場合の、DTKF の演算量を図 2 に実線で示す。

比較対象として、通常のカルマンフィルタにおいて遅延を許容するため、最大許容遅延 L 時刻分の $\hat{x}(j | j)$ 、 $P(j | j)$ 、 $\mathbf{y}(j)$ 、 $\mathbf{W}(j)$ を保存し、 L 回分の推定を単純に繰り返す場合に必要となる演算量を同図に一点鎖線で示す。これは基本的には通常のカルマンフィルタの演算量の L 倍であるが、異なるセンサに乗るノイズが無相関であると仮定して簡単化された提案手法と同じ条件とするため、 mp 次元の観測について式 (1)、(2) の観測更新を 1 回実行する代わりに、 p 次元の観測更新を、ユーザのもとに到着している観測情報の分だけ繰り返し実行するものとした。全センサの観測情報 m 個のうちいくつが実際に到着しているかは一概には分からないが、ここでは仮に平均 $m/2$ 個であるとして評価した。

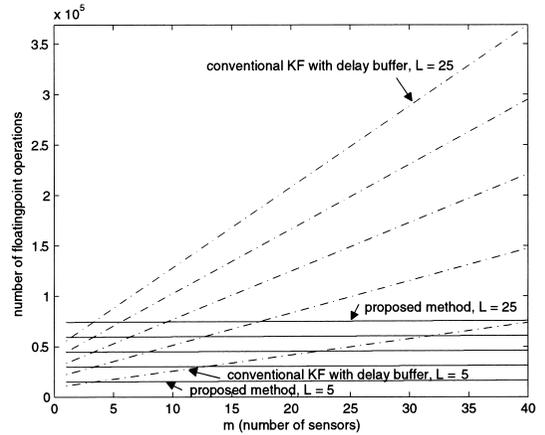


図 2 時間計算量の評価
Fig. 2 Computational complexity of DTKF.

この結果より、提案手法がセンサ数 m に関するスケーラビリティにおいて優位であることが分かる。 L, m 及び n に関するオーダ計算量は従来手法が $O(Lm + Ln^3)$ なのに対し、提案手法では $O(mn^2 + Ln^3)$ となる。従来手法では Lm の項が存在するため、センサ数の増大に対して遅延の取扱いが相乗的に困難になる。提案手法では、 m に関する繰返しと L に関する繰返しが分離されており、 m に関する繰返しは n 次元ベクトル及び $n \times n$ 行列の加算を各 1 回ずつ含むのみであるため、計算量の深刻な増加につながらない。

3. 遅延を考慮したセンサ選択手法

以上で導入した DTKF を用いることによって、遅延を伴って到着する観測情報を取り扱うことが可能となった。本章ではこの枠組みのもとで、遅延を考慮したセンサ選択手法を提案する。

3.1 基本方針

対象の状態をよりよく推定するためには、できるだけ多くのセンサの情報を統合することが望ましい。その一方で、センサの数が多い場合、すべてのセンサと通信し、その全情報を統合することは現実的ではない。そのため、対象に関する情報をできるだけ多くもたすようなものを優先的に選べるのが望ましい。

このような考え方から、カルマンフィルタにおいて観測が状態推定に対してもたらす相互情報量 [13] によってセンシングを評価すること、あるいはそれが最大となるようにセンシングや行動を選択することが従

来から行われている [16], [17]. ここでは, これを通信遅延のある場合の推定問題に適用する. 既に, 遅延がある場合へ適用できるようなカルマンフィルタを導入しているため, これをもとに相互情報量を求めることができれば, 適切なセンサが選択できると考えられる.

さて, このような相互情報量の評価は, 厳密には選択し得るすべての組合せについて行う必要があるが, そのような組合せの数は膨大であり, 実時間処理は困難である. また, 対象の状態を継続的に推定し続けたい場合, 観測の時系列全体を評価するには定常的な過程における極限を考えざるを得ず, 状況に応じた動的なセンサ選択には利用できない.

そのため以下では, 現在時刻の誤差共分散を基準として, 各センサについて, その 1 個の観測情報が遅延を伴って到着した際に, 現在時刻の推定値にもたらされる情報量をそれぞれ計算し, これに基づいて近似的にセンサ選択を行うことを考える.

3.2 アルゴリズム

カルマンフィルタにおいて, ダイナミックスのノイズと観測ノイズ, 及び対象の初期状態がガウス性であるとき, 観測値が推定値にもたらす相互情報量は, 誤差共分散行列の行列式を用いて計算できることが知られている [13], [16]. 文献 [13] にはその計算方法が詳しく述べられている. 以下のように考えることで, 遅延がある場合の相互情報量も, これと同様の手順で求められることが分かる.

現時刻 k において, あるセンサからユーザのもとに到着した観測値 \mathbf{y} と, この観測値を得る前に得ているすべての観測値の集合 \mathcal{Y} を既知としたときの現時刻での対象の状態 \mathbf{x}_y を考える. この両者間の相互情報量 $I_{\text{mutual}}(\mathbf{x}_y; \mathbf{y})$ が求めたいものである. これは定義により,

$$\begin{aligned} I_{\text{mutual}}(\mathbf{x}_y; \mathbf{y}) &= I_{\text{mutual}}(\mathbf{y}; \mathbf{x}_y) \\ &= \iint p(\mathbf{y})p(\mathbf{x} | \mathcal{Y}, \mathbf{y}) \log \frac{p(\mathbf{x} | \mathcal{Y}, \mathbf{y})}{p(\mathbf{x} | \mathcal{Y})} d\mathbf{x}d\mathbf{y} \\ &= E \left[\log \frac{p(\mathbf{x} | \mathcal{Y}, \mathbf{y})}{p(\mathbf{x} | \mathcal{Y})} \right] \end{aligned} \quad (7)$$

と表される. $p(\cdot)$ と $p(\cdot | \cdot)$ はそれぞれ確率密度分布と条件付き確率密度分布を表す.

この計算に必要な $p(\mathbf{x} | \mathcal{Y})$ と $p(\mathbf{x} | \mathcal{Y}, \mathbf{y})$ は, それぞれ \mathbf{y} が与えられていないときと与えられているときの \mathbf{x} の条件付き確率密度なので, 以下のとおり表

すことができる.

$$\begin{aligned} p(\mathbf{x} | \mathcal{Y}) &= \frac{\exp \left\{ -\frac{1}{2}(\mathbf{x} - \hat{\mathbf{x}}_y)^T P_y^{-1}(k)(\mathbf{x} - \hat{\mathbf{x}}_y) \right\}}{\sqrt{(2\pi)^n |P_y(k)|}} \\ p(\mathbf{x} | \mathcal{Y}, \mathbf{y}) &= \frac{\exp \left\{ -\frac{1}{2}(\mathbf{x} - \hat{\mathbf{x}}_{y,y})^T P_{y,y}^{-1}(k)(\mathbf{x} - \hat{\mathbf{x}}_{y,y}) \right\}}{\sqrt{(2\pi)^n |P_{y,y}(k)|}} \end{aligned}$$

ここで, $\hat{\mathbf{x}}_y(k)$ と $\hat{\mathbf{x}}_{y,y}(k)$ はそれぞれ \mathbf{y} が到着していないときと到着しているときに DTKF によって求められる推定値であり, $P_y(k)$ と $P_{y,y}(k)$ は同じく誤差共分散行列である. ただし $\mathbf{x} \in \mathbf{R}^n$ とした.

これらを式 (7) に代入し, 一般に $\mathbf{m}^T Q \mathbf{m} = \text{tr} Q \mathbf{m} \mathbf{m}^T$ が成り立つことを用いて整理すると, 相互情報量は

$$I_{\text{mutual}}(\mathbf{x}_y; \mathbf{y}) = \frac{1}{2} \log |P_y(k) P_{y,y}^{-1}(k)| \quad (8)$$

と得られる.

これを用いると, DTKF の実行時に, あるセンサ i ($i = 1, 2, \dots, m$) がもたらす相互情報量を計算することができる. ただし, 各センサ i の通信遅延 τ_i と誤差共分散情報 $C_i^T(k - \tau_i) W_i^{-1}(k - \tau_i) C_i(k - \tau_i)$ は既知とする.

センサ i が選択されているとき, DTKF によって求めた $P(k | k)$ は, センサ i の観測値を含んだ観測情報に基づくものなので, これが $P_{y,y}(k)$ である. また, このときの $P_y(k)$ は, $P(k | k)$ に対するセンサ i の寄与を取り除くことで求められるが, これは $I(k - \tau_i)$ に単純に加算されているため, これを減じた上で DTKF による再計算を行えばよい.

逆に, センサ i が選択されていないとき, DTKF によって求めた $P(k | k)$ が $P_y(k)$ である. $P_{y,y}(k)$ の計算には, i の寄与を加えればよい.

計算の手順は以下のようにまとめられる.

[Algorithm 2] 現在選択されているセンサ (のインデックス) の集合を S と書く. DTKF によって現時刻の誤差共分散行列 $P(k | k)$ を計算した後, 以下のとおり実行する.

if $i \in S$ then

$$P_{y,y}(k) := P(k | k)$$

$$\alpha := -1$$

else

$$P_y(k) := P(k | k)$$

$$\alpha := 1$$

```

end
for  $j = k - L + 1, k - L + 2, \dots, k$  begin
   $P(j | j-1) := A(j-1)P(j-1 | j-1)A^T(j-1)$ 
     $+ B(j-1)U(j-1)B^T(j-1)$ 
  if  $j = k - \tau_i$  then
     $P(j | j) := \{P^{-1}(j | j-1) + I(j)$ 
       $+ \alpha C_i^T(j)W_i^{-1}(j)C_i(j)\}^{-1}$ 
  else
     $P(j | j) := \{P^{-1}(j | j-1) + I(j)\}^{-1}$ 
  end
end
end

if  $i \in S$  then
   $P_Y(k) := P(k | k)$ 
else
   $P_{Y,y}(k) := P(k | k)$ 
end

```

以上により $P_{Y,y}(k)$ と $P_Y(k)$ の両方を求めることができるので、式 (8) により相互情報量が得られる。□

ここで導入した、通信遅延・誤差共分散情報が既知であるという仮定は、相互情報量の計算にのみ必要なものであり、DTKF の実行には必要ない。センサ選択時にこれらが未知だった場合、何らかの近似値または推定値で代替する必要があるが、それによって精度が劣化するのはあくまでもセンサ選択の結果のみであり、DTKF による最終的な状態推定の出力は、選択されたセンサ集合による正確な最適推定となることが保証される。

以上で述べたアルゴリズムにより、あるセンサの集合が選択されている状態において、ある一つのセンサが追加または削除されたときに、その観測情報をもたらすであろう情報量を評価することができる。この情報量の計算自体は、前節の DTKF と同様に各センサの観測レートが一致していなくても可能である。一方、これはあくまで 1 個の観測情報をもたらす情報量であるため、これをもとに各センサを選択する際には、全センサが同一のレートで観測していると仮定しなければ、価値の正確な比較ができなくなる。

また、既に述べたとおり複数のセンサの組合せを厳密に評価するのは困難であり、近似的解法が必要となる。例として、各時点において相互情報量が最大となるものを一つ追加して集合を更新し、これを推定誤差が一定の基準を満たすまで行う貪欲解法、相互情報量の大きな順に計算コストや通信コストが許す数まで選

択する方法などが考えられる。後述する数値実験では、あらかじめ定められた一定数を相互情報量の大きな順に選択する方法を評価している。

取捨選択を行うために個々のセンサを評価するためには、評価したいセンサの数に比例する計算時間が必要となる。フィルタリングの計算とは異なり、センサの評価は必ずしも毎ステップにすべてのセンサについて実行する必要はないため、これは深刻な問題ではない。例えば一定周期ごとに評価を更新する、選択・不選択の境界付近に位置する一部のセンサのみを評価する、などの方策を取ることが可能である。

4. 数値実験

高速視覚センサによる対象追跡 [18] を想定して数値実験を行った。センサのフレームレートを 1000 frame/s、すなわち観測周期を 1 ms とした。許容する最大遅延については、アプリケーションの要請として通常のビデオフレームと同程度のオーダーで遅延した観測情報までは有用な情報をもつと考え、10 ms までの遅延を取り扱うこととした。 $L = 10$ となる。図 2 によれば、この処理に必要な計算量は 1 ms 当りの浮動小数点演算回数で 10^5 以下、すなわち 100 MFLOPS 以下であり、近年のマイクロプロセッサの能力を考えると十分実行可能である。

対象の位置を $(x, y, z)^T$ として $x \equiv (x, y, z, \dot{x}, \dot{y}, \dot{z})^T$ とし、そのダイナミックスを

$$A(k) = \begin{pmatrix} I_3 & \Delta t I_3 \\ O_3 & 0.99 I_3 \end{pmatrix}, \quad B(k) = I_6,$$

$$U(k) = \sigma_u^2 \text{diag}\{0, 0, 0, 1, 1, 1\}$$

とにおいて、 $\Delta t = 0.001$ [s] とした。

センサは画像座標での対象位置を出力する正射影カメラであるとし、それぞれランダムに与えられた方向と観測ノイズ共分散をもつとして、観測モデルを以下のように定めた。

$$C_i = (I_2 \quad \mathbf{0}_2) R(\theta_i, \phi_i, \gamma_i) (I_3 \quad O_3)$$

$$W_i = \sigma_i^2 I_2$$

ここで $R(\theta, \phi, \gamma)$ は回転行列で、 $\theta_i, \phi_i, \gamma_i$ 及び σ_i^2 は各センサ i ごとにランダムに定めた。

センサからユーザへの通信遅延には、ネットワークやソフトウェアの構成によって様々な要因が影響する。ここでは特定のネットワークの構成を仮定せずに、簡

単なる乱数シミュレーションを採用する．各センサ i はある定められた平均遅延 \bar{d}_i をもち、実際の送信ごとの遅延は、この平均遅延の周りの正規分布 $N(\bar{d}_i, \sigma_d^2)$ に従う正規白色過程であるとした．これはごく単純化されたモデルであり、現実の具体的なネットワークへ適用する場合の挙動については、より精密なモデル化と評価が必要となる．

まず、DTKF の性能を通常のカルマンフィルタと比較して評価した．比較対象とした通常のカルマンフィルタでは、到着した最新の情報までの推定とそれ以降の予測のみを行い、順不同で遅れて到着した情報は破棄することとした．実験は四つの異なるパラメータ設定について行った．一つのパラメータ設定に関する実験は 50 試行からなる．試行ごとに、対象の軌跡とセンサ配置をランダムに生成し、この同一の対象軌跡・センサ配置に対して、通常のカルマンフィルタ及び DTKF による連続 100 時刻の状態推定を行った．

表 1 に示すパラメータは、四つの設定すべてについて共通とした．一方 σ_i^2 は、各設定について以下のように定めた．

設定 1 区間 $(0, 1.0^2)$ の一様乱数として生成

設定 2 区間 $(0, 5.0^2)$ の一様乱数として生成

設定 3 $\bar{d}_i > L/2$ ならば $\sigma_i^2 = 0.1 \times 1.0^2$ とし、そうでなければ $\sigma_i^2 = 0.8 \times 1.0^2$

設定 4 $\bar{d}_i > L/2$ ならば $\sigma_i^2 = 0.1 \times 5.0^2$ とし、そうでなければ $\sigma_i^2 = 0.8 \times 5.0^2$

設定 1 と 2 は、センサが無作為に配置された状況を模擬するためのものであり、一方、設定 3 と 4 は、センサ単体の精度と通信遅延がトレードオフの関係にあるような状況を恣意的に模擬するものである．

各設定における平均二乗誤差を図 3 に示す．ここでは 1 試行内の 100 時刻及びその 50 試行の全体にわたる平均が計算されている．横軸が設定 1 から 4 を、縦軸が平均二乗誤差を表す．グループ化された 2 本の棒グラフの左側が通常のカルマンフィルタによる結果を、右側が提案手法による結果を表す．明らかに提案手法

が良い結果を得ていることが分かる．特に、観測ノイズが相対的に大きく、かつ恣意的に定められている設定 4 では、その差が顕著に現れている．通常のカルマンフィルタでは、通信遅延は大きい精度の高いセンサからの重要な情報の多くが、到着が逆順になることによって破棄されていると考えられ、それによる効果が大きく現れていると考えられる．実際に全試行において情報が破棄された割合を調べると、通常のカルマンフィルタでは 82.7% であり、ほぼ各時刻における最新の情報しか統合されていなかった．一方、提案手法でも許容最大遅延を超えると情報の破棄が生じるが、その割合は 5.7% にとどまった．

次に、提案するセンサ選択手法を評価するため、他の評価基準による選択結果との比較を行った．比較対象は、ランダム選択、遅延を考慮せず相互情報量の大きな順に選択、平均遅延の小さな順に選択、の 3 種類である．5 時刻ごとに、提案手法を含む 4 種類の評価値の計算を行い、それぞれの評価値のよい順に、あらかじめ定められた数のセンサを選択した．いずれの場合も選択されたセンサからの情報の統合を DTKF によって 1 時刻ごとに行い、同様に推定値の平均二乗誤差を評価した．実験は、表 2 に示す四つのパラメータ設定について行い、各設定について、上記 4 種類の評価基準による選択センサ群を用いた連続 100 時刻の状態推定を、50 試行を行った．

各設定に関する、50 試行にわたる平均二乗誤差を図 4 に示す．図 3 と同様に、横軸に設定 1 から 4 を、縦軸に平均二乗誤差を示す．グループ化された 4 本の棒グラフは、左からランダム選択、相互情報量による

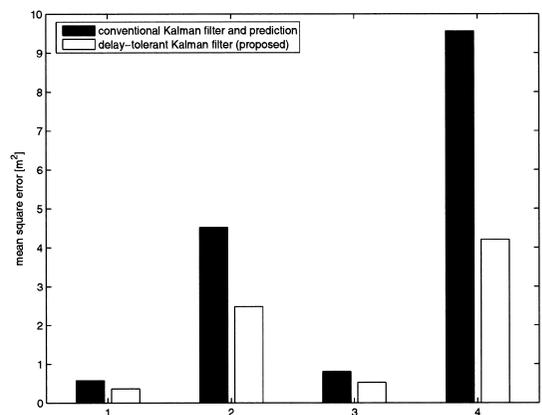


図 3 DTKF の評価実験結果
Fig. 3 Performance evaluation of the DTKF.

表 1 DTKF 評価実験のパラメータ設定
Table 1 Parameter setup in DTKF evaluation.

parameter	value
# of sensors m	20
maximum delay L	10
σ_d^2	2.0^2
σ_u^2	10.0^2
\bar{d}_i	uniformly distributed in $(1, L)$

表 2 センサ選択手法の評価実験で用いたパラメータ設定
Table 2 Parameter setups in the evaluation of sensor selection.

setup	1	2	3	4
# of sensors m	20			
# of selected sensors	5		3	
maximum delay L	10			
σ_i^2 (interval in which σ_i^2 is uniformly distributed)	$(0, 1.0^2)$	$(0, 5.0^2)$	$(0, 0.2^2)$	$(0, 1.0^2)$
σ_u^2	10.0^2			
σ_d^2	2.0^2			
d_i	uniformly distributed in $(1, L)$			

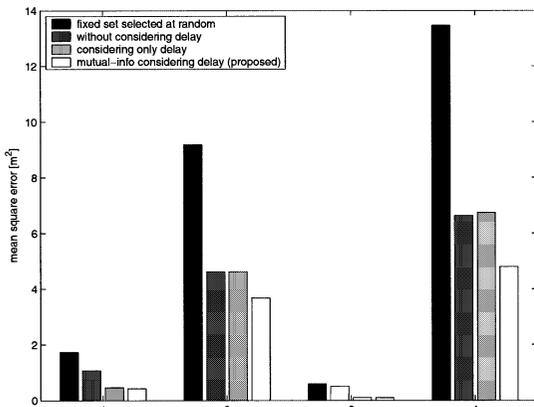


図 4 センサ選択手法の評価実験結果

Fig. 4 Performance evaluation of the proposed sensor selection method.

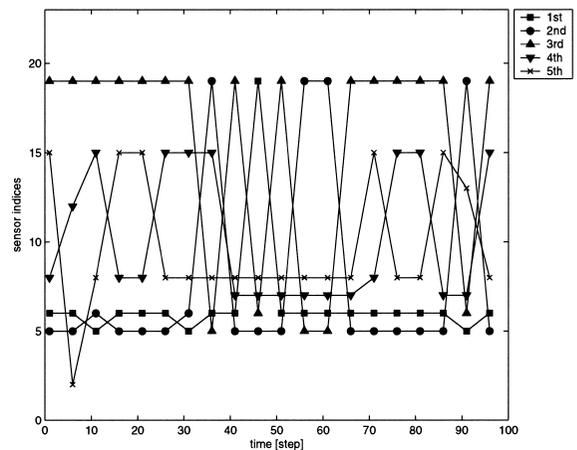
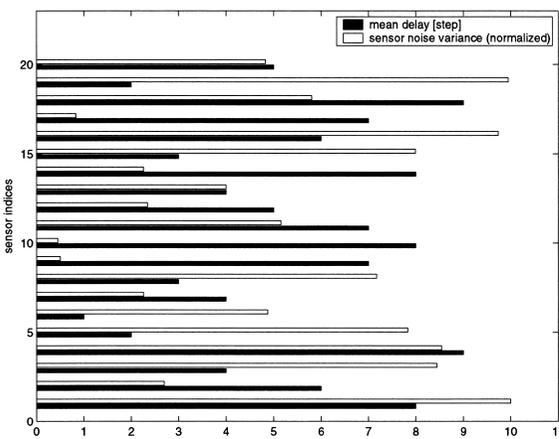


図 5 センサ選択結果の詳細 . 設定 3 (観測ノイズが小さい) における試行の一例

Fig. 5 Profiles of the sensors and the sequence of selection; a sample among the trials for the setup No. 3 (small observation noise).

選択, 平均遅延による選択, 提案手法による選択による結果をそれぞれ表す .

全体として, 提案手法が良好な, または少なくとも悪くない結果を得ている . 特に, 観測ノイズが大きい場合や選択するセンサ数が小さい場合に, 提案手法の優位性が明確に現れている .

その一方で, 観測ノイズがあまり大きくない場合は, 平均遅延のみに基づいた選択でも提案手法と同等の性能が得られている . この理由を明らかにするため, 各試行結果の詳細を検討した .

図 5 に, 設定 3, すなわち観測ノイズが小さい状況で行った 50 試行のうち, 一つの試行で用いられたセンサの設定と, その試行において提案手法によって選択されたセンサの時系列を示す . 図の左の棒グラフには, 各センサの平均遅延と, 正規化された観測ノイズが示されている . 図の右に示すグラフは, 各時刻で選択されていたセンサを, その選択順位とともに示している . この図より, 平均遅延の小さなセンサが優先的に選択されていることが見て取れる . これは, この状況では, 提案手法による選択と平均遅延のみによる選択の結果がほぼ等価になっていることを意味している . 観測ノイズが全体的に小さいときは, その中で精度の高いセンサを特に選ぶことによる効果は相対的に低くなると考えられ, この実験結果が示す挙動は妥当なものであるといえる .

逆に, 設定 2 のように観測ノイズが大きい場合は,

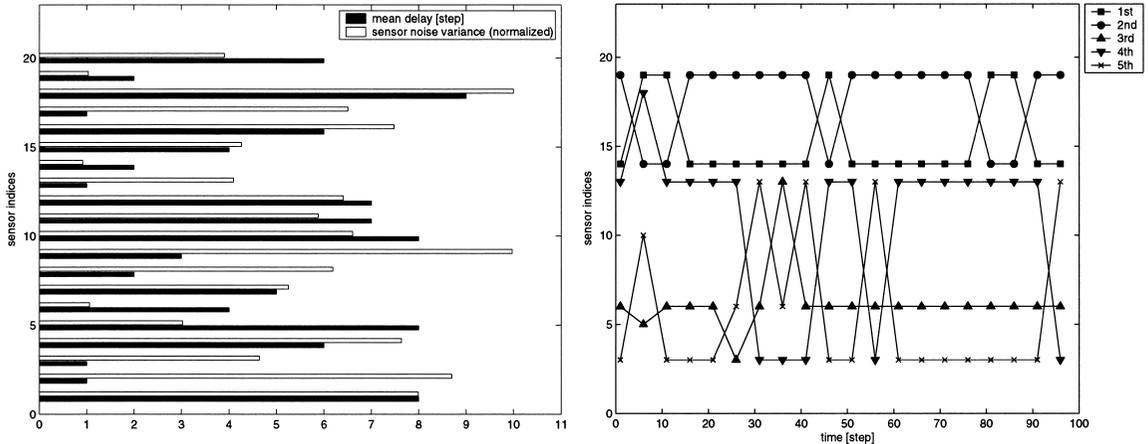


図 6 センサ選択結果の詳細．設定 2 (観測ノイズが大きい) における試行の一例
 Fig. 6 Profiles of the sensors and the sequence of selection; a sample among the trials for the setup No. 2 (large observation noise).

通信遅延の小さなセンサは必ずしも魅力的ではなくなる．図 6 に設定 2 の 1 試行におけるセンサ設定と，提案手法によるセンサ選択結果を示す．この場合は，通信遅延とセンサの精度の両方を勘案して選択が行われていることが見て取れ，結果として図 4 に示されるように良い推定が行われている．このように，提案手法は様々な状況に適応した，適切なセンサ選択を実現できているといえる．

5. む す び

本論文では，通信遅延を許容するカルマンフィルタの，センサ数の増大にスケーラブルな計算アルゴリズムを提案した．またその枠組みを用いて通信遅延の影響を考慮してセンサの観測情報をもたらず情報量を評価するアルゴリズムを提案し，それに基づくセンサ選択を検討した．これらの手法を数値実験により検証し，その有効性を確認した．

今後の課題としては，具体的なネットワーク構成を仮定して，パケットのロスやふくそう等も考慮したより精密な評価を行うことが挙げられる．また本論文では，各センサから単純に送信される情報を単一ユーザによって集中的に統合・センサ選択するという基本的な構造を議論したが，更なるスケーラビリティ向上のため分散化やデータ集約方法の検討なども重要な課題である．

文 献

[1] D. Estrin, D. Culler, K. Pister, and G. Sukhatme, "Connecting the physical world with pervasive net-

works," IEEE Pervasive Computing, vol.1, no.1, pp.59–69, 2002.

[2] 鏡 慎吾, 石川正俊, "分散リアルタイムセンシングによる高速動作獲得技術," 情報処理, vol.44, no.1, pp.34–39, 2003.

[3] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, "Next century challenges: Scalable coordination in sensor networks," Fifth Annual ACM/IEEE Int. Conf. Mobile Computing and Networking, pp.263–270, 1999.

[4] J.M. Kahn, R.H. Katz, and K.S.J. Pister, "Next century challenges: Mobile networking for "Smart Dust"," Fifth Annual ACM/IEEE Int. Conf. Mobile Computing and Networking, pp.271–278, 1999.

[5] G. Asada, M. Dong, T.S. Lin, F. Newberg, G. Pottie, W.J. Kaiser, and H.O. Marcy, "Wireless integrated network sensors: Low power systems on a chip," 24th IEEE European Solid State Circuits Conference, pp.9–12, 1998.

[6] 系 正義, 辻道信吾, 小菅義夫, "広域複数センサシステムにおける遅延データ対処型目標追尾フィルタ," 信学論 (B), vol.J84-B, no.10, pp.1857–1868, Oct. 2001.

[7] 江村 暁, 館すずむ, "シーケンシャルなセンサフュージョンのアルゴリズムとその理論," 計測自動制御学会論文集, vol.31, no.6, pp.705–711, 1995.

[8] 前山祥一, 大矢晃久, 油田信一, "移動ロボットのための避及的現在位置推定法—処理時間を要する外界センサデータの利用," 日本ロボット学会誌, vol.15, no.7, pp.1075–1081, 1997.

[9] H. Durrant-Whyte and M. Stevens, "Data fusion in decentralised sensing networks," Fourth International Conference on Information Fusion, 2001.

[10] B.S.Y. Rao, H.F. Durrant-Whyte, and J.A. Sheen, "A fully decentralized multi-sensor system for track-

- ing and surveillance,” Int. J. Robotics Research, vol.12, no.1, pp.20–44, 1993.
- [11] S. Grime and H.F. Durrant-Whyte, “Data fusion in decentralized sensor networks,” Control Engineering Practice, vol.2, no.5, pp.849–863, 1994.
- [12] H.R. Hashemipour, S. Roy, and A.J. Laub, “Decentralized structures for parallel Kalman filtering,” IEEE Trans. Autom. Control, vol.33, no.1, pp.88–94, 1988.
- [13] 有本 卓, カルマン・フィルター, 産業図書, 東京, 1977.
- [14] F. Sivrikaya and B. Yener, “Time synchronization in sensor networks: A survey,” IEEE Netw., vol.18, no.4, pp.45–50, 2004.
- [15] B.D.O. Anderson and J.B. Moore, Optimal Filtering, Prentice-Hall, Englewood Cliffs, 1979.
- [16] 有本 卓, “情報理論から見たシステムの同定と推定,” 計測制御, vol.9, no.4, pp.276–284, 1970.
- [17] 阪口 豊, “計測データの統合と融合,” 計測工学ハンドブック (山崎弘郎, 他 (編)), pp.220–227, 朝倉書店, 東京, 2001.
- [18] 石川正俊, 小室 孝, “デジタルビジョンチップとその応用,” 信学論 (C), vol.J84-C, no.6, pp.451–461, June 2001.

(平成 16 年 3 月 4 日受付, 11 月 2 日再受付,
12 月 28 日最終原稿受付)



鏡 慎吾

平 10 東大・工・計数卒・平 15 同大学院博士課程了。科学技術振興事業団研究員を経て、現在東京大学大学院情報理工学系研究科システム情報学専攻助手。実時間センサ情報処理アーキテクチャ、アルゴリズム、システムの研究に従事。博士 (工学)。



石川 正俊 (正員)

昭 52 東大・工・計数卒・昭 54 同大学院修士課程了。同年通産省工業技術院製品科学研究所に入所。平元東大・工・計数助教授。現在同大学院情報理工学系研究科システム情報学専攻教授。超並列・超高速ビジョン、センサフュージョン、光コンピューティング等に関する研究に従事。工博。