

天井照明の光度制御を用いた通信手法の検討

村上 広記, 長光 翔一

Hiroki Murakami, Shoichi Nagamitsu

1 はじめに

無線センサネットワークは、無線装置を内蔵した多数のセンサが相互に連携することで、実空間の情報の収集を可能とする技術である¹⁾。無線センサネットワークを構成する全てのノードにデータを送信するためには、一般的にフラッシングを用いる方法がある。しかしながら、フラッシングは無線センサノードの限られた電力資源を消費するとともに転送遅延により同時にデータを取得することはできない。そこで本研究では、天井照明の光度制御を用いて、無線通信を行わないデータ通信手法を提案する。照明の光度制御を用いるデータ通信手法としては可視光通信システム²⁾があるが、高速通信を行うために専用の照明設備と受光端末が必要となる。そのため本研究では、調光可能な照明器具および一般的な照度センサのみを用いてデータ通信を実現する。外光などの照明光以外の外的要因による影響を考慮しない場合における、データ通信手法の基本的検討を行う。

Fig.1 照度取得実験環境

2 天井照明の光度制御を用いたデータ通信手法

2.1 天井照明の光度制御を用いたデータ通信手法の概要

本手法では、照明の明るさ（光度）を変化させることで、照度センサが感知する明るさ（照度）を変化させ、照度変化量からデータ通信を実現する。まず本手法のアルゴリズムを提案するために、光度変化時の照度変化の推移を調査する。照度変化時の課題として、ユーザが照度変化を感知してしまうと、ユーザの快適性が損なわれる。したがって、ユーザが感知できない範囲で照度変化を起こす必要がある。その変化量は、先行研究により、照度変化量が現在照度の 7% 以内であれば、人は感知できないことが確認されている³⁾。本手法でも室内環境での利用を想定し、ユーザの快適性を維持するために、現在照度の 7% 以内の変化量を与え、データ通信を実現する。

2.2 無線センサノードを用いた照度推移実験

現在照度の約 7% 以内の変化量を与えた場合に、無線センサノードに搭載された照度センサが得る照度値がどのように推移するかを検証する。本実験では、無線センサノードとして Crossbow 社の MOTE MICAz を使用した。MOTE MICAz に汎用外部センサ基盤である MDA088 を設置し、リードタイプの NaPiCa 照度センサを組み込むことで、照度値を取得する。本実験は同志社大学香知館の知的システム創造環境実験室にて行い、シャープ製フルカラー LED29 灯と NaPiCa 照度センサ

Fig.2 100 ms 間隔で取得した照度履歴

を搭載した無線センサノード 1 台、シンクノード 1 台を用いた。照度センサの照度取得間隔を 100 ms に設定した。実験環境の俯瞰図を Fig. 1 に示す。なお、照明と無線センサノードの距離は照明の垂直直下に無線センサノードを置いた場合、1.9 m である。無線センサノードは、Fig. 1 に示す通り照明の直下に配置した。

机上面照度が 500 lx となる環境下で、照度を 500 lx の 3% である 15 lx 程度上昇させ、その 1 秒後に元の照度値に戻した場合の照度の推移を取得した。なお、外光など照明以外の光が照度センサに影響を与えない環境下で行った。Fig. 2 に照度取得間隔が 100 ms の場合のある時間における照度推移の履歴を示す。

Fig. 2 からわかる通り、光度変化を感知したとき、照度が急速に変化していることがわかる。この実験結果から、天井照明の光度制御を用いたデータ通信アルゴリズムを提案する。

2.3 天井照明の光度制御を用いたデータ通信アルゴリズム

本手法のアルゴリズムを述べる。なお、本手法では外光などの照明以外の光が照度センサに影響を与えない環境を想定して、アルゴリズムを提案する。

Fig. 2 から、光度変化を感知し照度が急速に変化するとともに、同じ光度で照明が点灯している場合にも、照度値に誤差が発生し、照度値に揺らぎが発生している。したがって、光度変化した際のみ照度変化をセンサが正しくデータ通信時の照度変化であると認知する必要がある。そこで本手法では取得した照度値を前回取得した照度値を用いて微分し、その傾きの大ききで光度変化時の照度変化か否かを判断する。

次に、具体的なアルゴリズムを述べる。データ送信間隔はセンサノードが既知であるという条件で、アルゴリズムを提案する。なお、データ送信間隔を $T[s]$ とする。照明側では送信したいデータを 2 進数のビット列に変換し、1 ビットずつ送信する。本手法では、傾きが閾値未満であれば 0、閾値以上であれば 1 とする。なお、センサノードにデータ通信開始を伝えるため、開始ビットとして 1 を送信する。また、連続して同じビットを送信する場合は同じ光度で照明が点灯するため、照度値は変化しない。したがって、開始ビット受信時から T 秒後に現在照度値を用いて 1 か 0 かを判断する。

- (1) 照度センサが現在照度を取得する
- (2) 照明が現在光度から $x\%$ (ただし, $x < 7\%$) だけ光度値を上昇させる
- (3) 照度センサが照度変化を感知した場合、照度変化前後の傾きを計算する
- (4) 傾きが閾値範囲内であれば、データ通信を開始する
- (5) 照明が光度値を光度変化前に戻す
- (6) 照度取得間隔毎に傾きを計算し、その値が閾値以上なら 1 とする
- (7) 開始ビット受信時から T 秒間傾きが閾値未満の場合、現在照度が照度変化後の照度値域であれば 1、変化前の照度値域であれば 0 とする

3 データ通信時における誤り率の検証実験

3.1 データ通信時における誤り率の検証実験概要

天井照明の光度制御を用いたデータ通信手法を無線センサノードに実装し、データ通信時における誤り率の検証を行った。前章の実験と同様の実験環境および使用機器を用いた。また、外光が入らないよう窓際には白色のパーティションを設置した。本実験では、あらかじめランダムに生成した 2 進数のビット列を用意し、それらをすべて送信した。なお、ビット長は開始ビットを含めて 1000 bit とし、送信間隔は 1 秒とした。

3.2 データ通信時における誤り率の評価

本実験におけるデータ通信時の誤り率の評価を行う。送信ビットと受信ビットとを比較し、異なる値を受信し

た回数を求め、誤り回数/全ビットを誤り率 [%] とした。実験の結果、誤り率は 48.7% となった。これは同じビット列を連続して送信する際に、照明側のデータ送信間隔 T とセンサノード側の内部時間とでずれが生じたために起こったと考えられる。センサノード内時間で T 秒経過した後に照明光度が変化した場合、データ送信側は 1 ビットしか送信していないのにも関わらず、2 ビット受信したとセンサノードが判断する。したがって、誤りが発生する。

この問題を解決するために、 T 秒毎にビット判断を行うのではなく、照明とセンサ側とのずれを考慮した冗長な短い時間 t を T に加え、連続したデータ受信時のビット判断時間を変更する。また、傾きが閾値以上になった際には、 $T+t$ 秒以内でも 1 を受信したと判断し、新たに加えた冗長な時間を補正する。このアルゴリズムを用いた同様の環境で再度実験を行った結果、誤り率は 0.1% となった。この結果から、光度変化を正しく感知できていることがわかる。

4 今後の展望

本研究で提案したデータ通信手法を用いた応用手法として、無線センサノードの時刻同期手法があげられる。本手法を用いて時刻データをすべてのセンサノードに送信することで、複雑なアルゴリズムを用いずに無線通信を行わない時刻同期手法が実現できる。

今後の課題として、データ通信速度の向上が挙げられる。例えば時刻データであれば、時間、分、秒、ミリ秒を送信する場合、それぞれを 7 から 8 ビットで表現するため、最大 32 ビット送信する必要がある。したがって、データ送信間隔 T が 1 秒であれば 32 秒送信に要することとなり、現時点では非常に低速な情報伝達手法であることがわかる。そこで、データ送信間隔を短縮することで高速化を図る。NaPiCa 照度センサの最小照度取得間隔は 8.5 ms である。したがって、最小照度取得間隔毎に照明光度を変化させ、ビットを順次送信することで高速化が実現可能であると考えられる。

また、1 度に送信する情報量を増やすことでも高速化を実現することができる。例えば照度変化量に複数の段階を設けることで 2 進数以外でデータの変復調が可能となり、1 度の送信に際してより多くの情報を送信することができる。しかし、照度センサの分解能の違いにより、同一環境における光度変化でも照度変化量が異なる。そのため複数の異種センサを同時に利用する場合には、同一の光度変化をすべての照度センサが同等の照度変化であると感知可能な手法を検討する必要がある。

参考文献

- 1) YICK, J.: Wireless sensor network survey, Comput. Netw., vol.52, pp.2292-2330(2008)
- 2) 春山真一郎: 可視光通信, 電子情報通信学会論文誌. A, 基礎・境界, Vol.86, pp.1284-1291(2003)
- 3) 鹿倉智明, 森川宏之, 中村芳樹: オフィス照明環境における明るさの変動知覚に関する研究, 照明学会誌, Vol.85, pp.346-351(2001)