JSPS-CNRS 二国間共同研究の枠組みでの学生派遣についての報告

加藤　碧 (katoon@sfc.wide.ad.jp)

概 要

ユーザの利用状況の把握，異常検出のために，ISP のような大きなネットワークにおける効率的なトラフィック監視は欠かせない．効率的にネットワーク状態を把握するために，トラフィック解析によって，オペレータが把握できるコンパクトな分量で，ネットワークの問題発見に発見するために冗長な情報を提供する必要がある．これを実現するため，フロー集約というトラフィック解析が注目を集めている．

フロー集約は，フロー情報を抽象化する．フローは 5-tuple (送信元/宛先アドレス，プロトコル，送信元/宛先ポート) を含む．フロー集約では，この 5 つのアトリビュートについて各々抽象化を行い，複数のフローをひとつの「集約フロー」で表す．集約フローによって膨大なネットワークのトラフィックデータは圧縮される．同時に，集約フローはネットワークの状態を把握するためのトラフィックパターンを示す要素になる．

集約フローは，トラフィック監視の際のトラフィックパターンに相当するため，情報の信頼性に影響する．不用意に集約フローのアトリビュートの抽象度が高くすると，トラフィックパターンが隠蔽されてしまう．情報量と情報の抽象度のトレードオフを考慮して，集約フローを生成する必要がある．

従来の研究では，このトレードオフについて直感的な議論が行われてきた．そのため，フロー集約がトラフィック監視に必要十分な情報量を含んでいるか，私たちは科学的な視点から把握できていない．このことからフロー集約の結果を定量的に評価して，このトレードオフに関して定式化することが重要であると私たちは考えている．

これを実現するためのスタートポイントとして，私たちは情報理論の応用を試みた．情報理論では，フロー集約と同様，情報量と情報の抽象化にはトレードオフが考えられる．例えば，画像処理において，画像に含まれるシグナルのエントロピーの集約を行い，情報の抽象化を実現する．画像は，データ圧縮が求められるほど，画像のパターンを抽象化する．これにより，被写体は画像から把握しづらくなる．画像処理において被写体を認識するための必要十分な情報を含ませることは，フロー集約においてトラフィックデータからトラフィックパターンを抽出することと同位である．

JSPS-CNRS 二国間共同研究（代表：江崎浩，Patrice Abry）の枠組みでの学生派遣として，慶應義塾大学大学院 政策・メディア研究科 徳田研究室 修士課程 1 年の加藤 碧が Ecole Normale Sup´erieure de Lyon (ENS-Lyon) で 1 ヶ月間 Patrice Abry の下で研究活動を行った．その研究内容として，MCA という画像処理を応用してフロー集約の定式化に取り組んだ．本報告書では，Monophological Component Analysis (MCA) のトラフィック解析の応用について述べる．

1　まえがき

JSPS-CNRS 二国間共同研究においてフランス国立科学センター（CNRS: Centre National de la Recherche Scientifique）との協力活動があり，その一環として学生派遣を行っている．本年度は，慶應義塾大学大学院政策・メディア研究科 徳田研究室 修士課程 1 年の加藤 碧が対象者の 1 人として，2012 年 9 月 4 日から 2012 年 10 月 4 日までの 1 ヶ月間，フランスのリヨンにおいて研究活動を行った．受け入れ先は Ecole Normale Sup´erieure de Lyon (ENS-Lyon) の Patrice Abry らの研究室である．Patrice は，同大学の Pierre Borgnat とともに，WIDE プロジェクトメンバの江崎浩（東京大学），長健二郎（IIJ 技術研究所），福田健介（国立情報学研究所）との協力体制でインターネットトラフィック解析の研究を行っている．

トラフィック集中やDDoS攻撃，ポートスキャン，設定ミスなどをトラフィックパターンから把握するために，ネットワークオペレーションにトラフィック監視は欠かせない．また，ネットワークの使用状況や長期的なインフラ増強のための指標として，トラフィック監視は重要な役割を持つ．

これらの目的を達成するために，ネットワークオペレータはベンダーや独自開発の監視ツールを使用する．監視ツールは，効率的にネットワークのトラフィックデータを解析して，ネットワークオペレータに簡潔なサマリーチャートを提供する．

サマリーチャートは，短期間と長期的なチャートの2種類ある。短期的なサマリーチャートは，異常検出を行いやすいもの，長期的なサマリーチャートは，ネットワークオペレータがサービス傾向を一目で理解できるものである必要がある．そのため，サマリーチャートは，人間が把握しやすい分量で，問題発見に十分なトラフィックパターンを含んでいることが重要だ．

効率的なトラフィックパターンを生成するために，フロー単位のトラフィック解析が行われる．フローとは，固有の5-tupleをもつパケットのことで，IPヘッダ，TCPヘッダに含まれる送信元/宛先アドレス，プロトコル，送信元/宛先ポート，5つのアトリビュート情報の組み合わせからなる．

最もシンプルで典型的なフロー単位のトラフィック解析では，トラフィックで観測されたフローのトップランキングの表示する．しかし，ネットワークの異常検出や使用状態はフローの集合によって観測できる．例えばDDoS攻撃は，特定の宛先アドレスと宛先ポート番号，プロトコル，不特定の送信元アドレスと送信元ポート番号の組み合わせである．この攻撃は，各フローのデータ通信量が少ないため，トップランキングの分析だけでは，ネットワーク攻撃の発見を行うことが難しい．

この問題を回避するために，5-tupleの中の特定のアトリビュートを抽象化した「集約フロー」がある．この集約フローを使ったトラフィック解析手法として「フロー集約」[2,3,6]が注目を集めている．図2に，フロー集約の概要を示した．平面は2次元アドレススペース（送信元/宛先アドレス），高さはトラフィック量を表している．説明を簡単にするために，本報告書では5-tupleのうち，2次元アドレススペースにおけるトラフィックデータを用いて説明する．右図はある期間に観測されたトラフィック，左図はフロー集約によって生成された集約フローのイメージである．右図で凸凹に観測されたフローのトラフィック量は，フロー集約によって均一化され，左図のような集約フローに変換される．

フロー集約を使うと，トラフィックデータに含まれるフロー数は大幅に減少する．そのため，フロー集約はデータ圧縮技術と言い換えることができる．データ圧縮率は集約フローに依存する．集約フローが1つアトリビュートを抽象化するとき圧縮率は小さく，4つのアトリビュートを抽象化するとき圧縮率は高くなる．

トラフィック監視において，このデータ圧縮率の調節は難しい．集約フローのひとつひとつは，ネットワークのトラフィックパターンを示す要素となりとなるため，高い圧縮率でパターンを隠蔽してはならない．しかし，ネットワークオペレータがネットワークの状態を一目で把握できるように，十分なデータ圧縮が必要になる．

データ圧縮率とデータの詳細を把握のための情報量はトレードオフである．従来のフロー集約の研究では，このトレードオフを認めた上で，集約フローを使ってトラフィックパターンを適切に抽出できているのかという点に対して，直感的な理解がされてきた．そのため，フロー集約を使った監視ツールのサマリーチャートは，トラフィックデータをどの程度具現化した情報なのか，信用性のある情報なのか，定量的な評価をすることができない．

私たちは，生成された集約フローの妥当性を検証するために，JSPS/CRNSプロジェクトの一環として，フロー集約の定式化を試みた．今回の交換留学プログラムでは，そのスタートポイントとして，Monophological Component Analysis (MCA) [4,5]という画像処理をトラフィックパターン抽出に応用した．

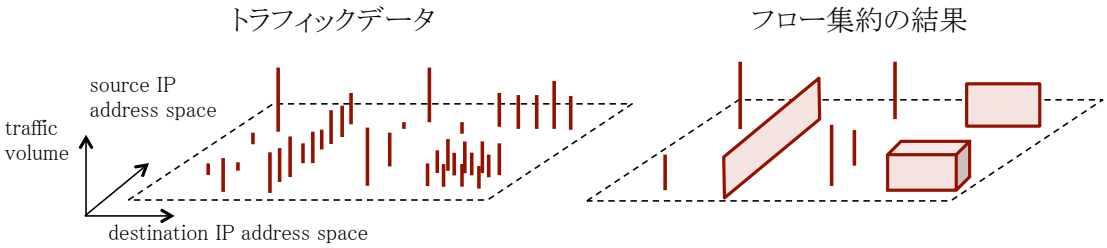

図 1: フロー集約の概要

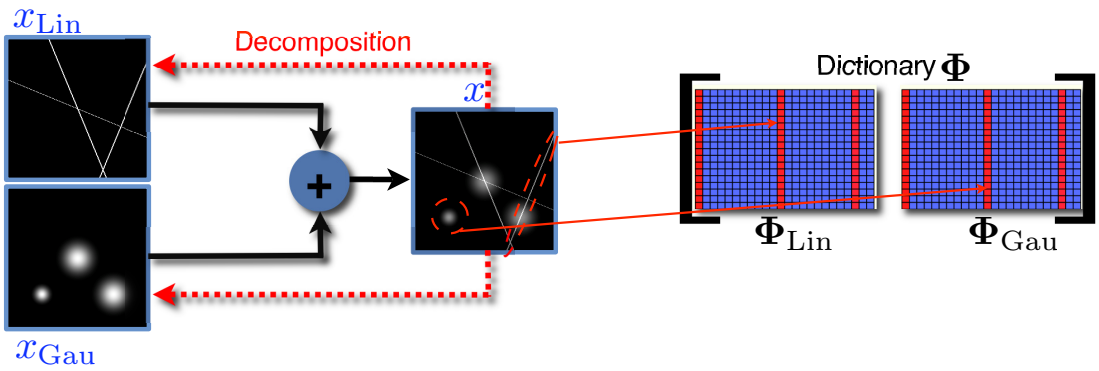

図 2: MCA の概要 ([4] の Fig. 1 より)

画像処理では，先に述べたフロー集約のトレーとオフが画像の解像度と同様に観測される．高解像度の画像は情報量が多いため，細部まで被写体を認識できる．一方，低解像度の画像は，情報量が少なくなるが，被写体の特徴が曖昧になる．

MCA は，画像から被写体を認識するために開発された技術である．私たちは，この技術を応用することで，トラフィックデータをフロー集約したときに，抽出されなくてはならない部分の比較ができると考えた．

本報告書は，以下のような構成になっている．2 章は MCA の概要を述べる．3 章では，トラフィックデータから抽出する要素の定義を行い，4 章では，MCA をトラフィックデータに適応した実験の結果，考察を述べる．5 章で MCA の問題点を述べた後，6 章は，MCA の問題点を考慮して改良した Ligt-weigh MCA の概要と，トラフィックデータでの実験結果について述べる．7 章で結論と今後の展望について述べる．

2 MCA の概要

本章では，トラフィックデータから要素を抽出するために使用した画像処理技術「Monophological Component Analysis (MCA)」を説明する．MCA は，画像から定義された要素の抽出および補正を行う技術である．漫画の 1 シーンからキャラクターと背景を分裂させる，画像のノイズを取り除き，ノイズを取り除かれた画像に補正をする，などの用途で使用されている．

MCA の概念を図 2 に示した．ここでは，線分と円を抽出するために，それらの要素が混在する画像から MCA を使って線分と円を分離している．MCA の要素抽出プロセスを，順に説明する．MCA は抽出する要素を辞書として定義する．この辞書は，大小のスケールを問わない．MCA で

は，抽出プロセスの途中，ウェーブレット変換やカーブレット変換を利用して調べることで，大きさの異なる要素を正確に抽出する．例の場合，円（Φ_{Gau}）と線分（Φ_{Lin}）を抽出要素として定義している．次に，オリジナルの画像（x）と辞書の類似度を示す係数（α）を計算する．閾値化の結果，残ったエントロピーが要素として抽出される．MCA は，辞書に定義された要素を異なる閾値で計算を繰り返し，最終的に，x_{Lin}，x_{Gau} のようにそれぞれの要素が抽出される．

MCA は具体的な計算プロセスを示す．初期条件を以下に示す．

- 残像 $r_0 = \emptyset$
- 画像 $x_0 = x$
- 抽出された画像 $x_{1,\cdots,k} = \emptyset$
- 係数の閾値 $\lambda_0 = \lambda$

次に，計算プロセスを示す．

- $l(l = 0, ...L)$ 回目の繰り返しにおいて，以下のプロセスを繰り返す
- それぞれの要素 $(k = 1, ..., K)$ について，
  - 残像を計算する $r_k = r_l + x_k$
  - 要素 k について，係数と閾値の比較を行う $\alpha_k = TH_{\lambda_l}(\Phi_k^{-1} r_k(l))$
  - 要素 k の抽出を行う $x_k(l) = \Phi_k \alpha_k(l)$
- 残像を更新する $r(l+1) = x - \sum_k x_k(l)$
- 閾値を更新する $\lambda(l+1) = \lambda(l) - \delta$
- 閾値 $\lambda(l+1)$ が予め定められた値より小さければ，反復をやめる．それ以外の場合は，プロセスの開始に戻る．

MCA は画像に存在する要素を抽出するため，最終的に残像と抽出された要素の合計が定常状態になる．これは，以下のような条件式で表すことができる．この条件式が不成立であるとき，抽出された要素は不十分であるということを示す．

$$min_{(k=x_1,\ldots,x_K)} \sum_{(k=1,\ldots,K)} \Phi_k^{-1} x_k \quad s.t. \quad ||x - \sum_{k=1,\ldots,K} x_k < \epsilon|| \tag{1}$$

3　トラフィックデータから抽出する要素の定義

フロー集約がトラフィックデータから集約フローとして抽出される要素は，具体的に 3 つに大別できる（図 1 を参照）．1 つ目は，送信元／宛先アドレスが特定されて通信量が多いパターン（1 対 1）．2 つ目は，特定の送信元（または宛先）アドレスから不特定の宛先（または送信元）アドレスへの通信が多いパターン（1 対多数）．このパターンは，例えば web サーバへのトラフィック集中が起きたときに観測される．最後は，不特定の宛先（または送信元）アドレスから不特定の送信元（または宛先）アドレスへの通信量が多いパターン（多数対多数）．特定のネットワーク同士のデータ通信を示す．以後，これらのパターンを「点」,「線分」,「矩形」と表記する．

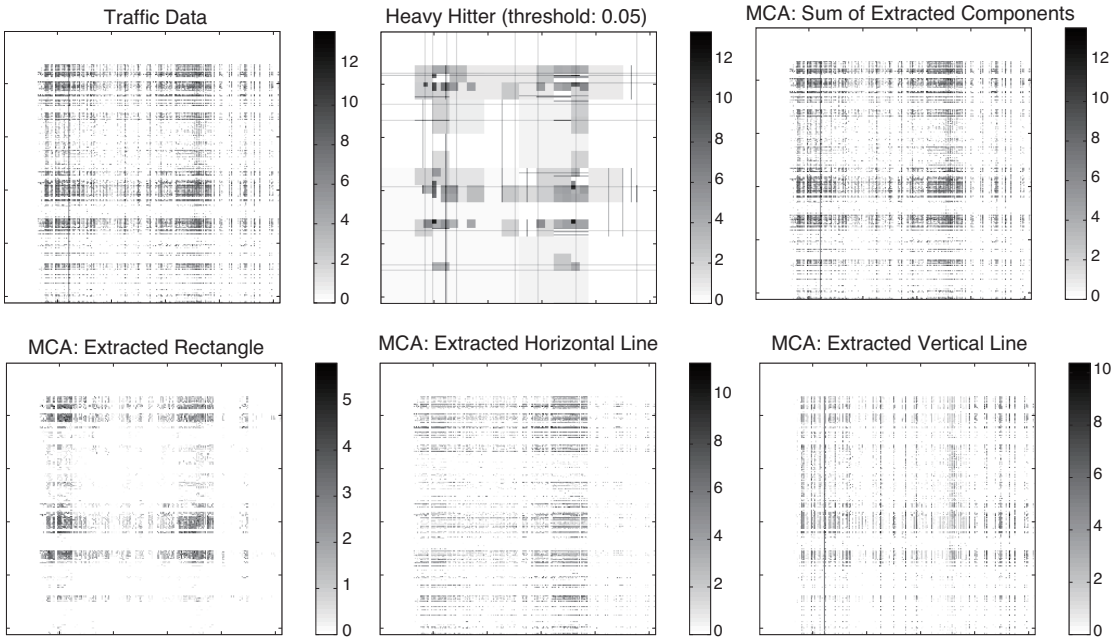

図 3: Light-weight MCA の実験結果

4 トラフィックデータを使った MCA の実験

CAIDA [1] が公開しているトラフィックデータを使用して，MCA による要素抽出を実験した．2 次元アドレススペースで観測されたトラフィックをネットマスク /8 で圧縮して、256x256 ピクセルの画像を生成した．また、この実験では，辞書として線分と矩形を定義した．

図 3 は，可視化した実験結果を示している．色の濃淡は，観測されたトラフィック量を log スケールで表している．左上図は，実験に使用したトラフィックデータをプロットしたもの．真ん中上図は，全体の 0.05%以上のトラフィック量が観測されるように，点，線分，矩形をプロットしたもの．これはフロー集約の結果の指標である．右上図は，MCA によって抽出された要素をあわせた結果である．下図に，MCA で抽出された要素を，辞書ごとに可視化した．

MCA で抽出された各要素と，一定以上のトラフィック量が観測された要素を比較すると，その差は大きいことが観察できる．MCA で線分として認められた要素数は，一定以上のトラフィック量が観測された線分の要素数より圧倒的に多く存在する．また，一定のトラフィック量として観測された要素のほとんどは矩形だが，MCA は矩形よりも線分を多く抽出している．図から，トラフィックデータ中にすべての要素が一定の通信量をもつわけではないことが分かる．

5 MCA の欠点

前章の実験結果から，MCA をトラフィックデータに適応する際の問題点が浮上した．まず，画像に存在する要素を抽出する技術であるため，MCA の技術をそのまま適応すると，トラフィックデータから一定以上の通信量（エントロピー）をもつホスト群という条件は無視されるということ．そのため，データ通信量を基準に要素抽出するフロー集約とは，大きく異なる結果になることは避けられない．

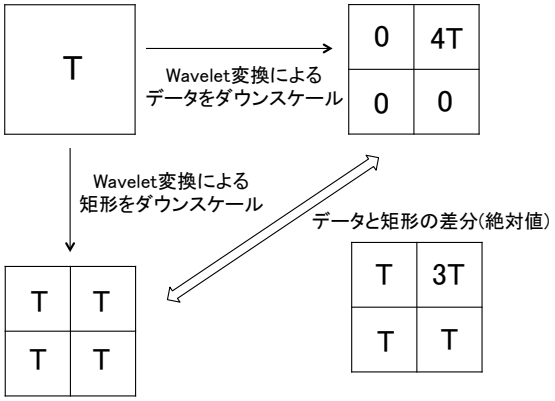

図 4: Light-weight MCA の概要

次に，MCA は要素の優先順位をつけることができないということ．一般的に，ネットワークオペーレータがネットワーク状態を把握する際には，点，線分，矩形の順に要素の重要性を認識する．そのため，この優先順位を考慮して，要素の抽出されることが好ましい．しかし，MCA にはこの優先順位が考慮されない．加えて,「点」や「正方形以外の矩形」の要素を抽出することができない．MCA では，大きさの異なる正方形の集合として矩形を認識する．これは矩形をひとつの集約フローとして扱うフロー集約と異なるため，フロー集約と MCA の結果の比較が行えない．

最も大きい問題点は，計算コストだ．MCA は繰り返し要素の抽出を行うため，計算量が多い．図 3 に示したデータは，オリジナルのトラフィックデータを何百億分の 1 以下になるように圧縮したものであるが，この計算に 4 時間かかった．実際のトラフィックデータは何十億倍も大きなデータになるため，MCA で計算するのは，非現実的な計算コストがかかると予想できる．以上の 3 点の問題点を解決するために，私たちは Light-weight MCA の開発に取り組んだ．

6 Light-weight MCA

Light-weight MCA の概要を説明する．Light-weight MCA は，要素のエントロピーの大きさを考慮した上で，計算を繰り返すコストを抑えるために，係数の計算を簡単にした．そのため，繰り返し計算を最小限にするために，ネットマスクの大きさ分だけ処理を行う．加えて，計算の効率性を考慮するために，大きな要素から順に抽出する．また，要素を抽出する際，Light-weight MCA では，2 つの閾値化を行う．1 つは，要素のエントロピーの閾値化．2 つ目は，係数の閾値化．前者で MCA の 1 つ目の問題点を考慮し，後者で抽出する要素の大きさを決定する．エントロピーの閾値は，抽出する要素の大きさが変わる（異なるネットマスクごと）に再計算される．エントロピーの閾値化だけで抽出する要素を決定してしまうと，大きい要素として抽出されてしまうため，係数の閾値を加えた．この閾値は，抽出する要素ごとに決定される．

矩形を抽出するときのポリシーを図 4 を使って説明する．左図は，エントロピーの閾値化の後の矩形のエントロピーを示し，右図はウェーブレット変換を使ってダウンスケールしたときの矩形のエントロピーを示している．要素を特定するために，要素の大きさはより小さい方が好ましい．そのため，左図の状態ではなく，右の状態で矩形を抽出される必要がある．この状態判断をするために，以下のように係数の閾値 λ を求める．まず，下図のように左図をダウンスケールする．下図と右図から，実際のデータと矩形のデータをダウンスケールした差分が求まる．差分と実際のデータ

のエントロピーは，$\frac{3}{2}$ 倍の合計になっている．これを正規化して，$\lambda = \frac{3}{8}$ と求まる．線分の要素に関しても同様に係数の閾値を考えると，$\lambda = \frac{1}{2}$ と求まる．

次に具体的なアルゴリズムを示す．まずパラメータの初期条件を以下に示す．

- 残像 $r_0 = \emptyset$
- 画像 $x_0 = x$
- 抽出された画像 $x_{1,\cdots,k} = \emptyset$

次に，計算プロセスを示す．

- それぞれのスケール係数 $j(j = 0, \cdots, J)$ について，以下のプロセスを繰り返す
  - それぞれの要素 $k(k = 1, \cdots, K)$ について，
    - 残像の計算をする $r_k = r_j + x_k$
    - エントロピーの最大値を画像から求める $M_j = max(x_{i=1,\cdots,n})$
    - 閾値 $\lambda_{(1,j)}$ を決める $\lambda_{(1,j)} = M_j\theta$
    - 閾値以上のエントロピーをもつ要素を画像から $x_{i=1,\cdots,n}$ 選択した $\tilde{x}_j$ を作成 $\tilde{x}_j = TH_{\lambda_{(1,j)}}(x_i)$
    - 残像 r_j をダウンスケール（$j = j + 1$ のときの状態 $r_{(j+1)}$）した画像 $\tilde{r}_j$ を作成 $\tilde{r}_j = \uparrow_2 r_{(j+1)}$
    - 閾値以上のエントロピーが観測された画像 x_j をダウンスケールして，画像 $\tilde{x}_j$ を作成 $\tilde{x}_j = \uparrow_2 x_{(j+1)}$
    - $\tilde{x}_j$ と $\tilde{r}_j$ の差分について閾値化を行う $\alpha_k = TH_{\lambda_{(2,j,k)}}(\Phi_k^{-1} r_k(l))$
    - 要素 k の抽出を行う $x_k = \Phi_k \alpha_k$

4 章で使ったデータセットを用いて，Light-weight MCA の実験をした．図 5 は実験結果である．Light-weight MCA は，MCA で 4 時間程度かかった処理時間を 10 分程度に削減することができた．また Light-weight MCA では，抽出された要素がスケールごとに決められた閾値以上であるときに抽出されるため，MCA より大きな要素が観測できる．

しかし，Light-weight MCA も MCA 同様，フロー集約の定式化を行うためには，いくつか不十分な点が残る．Light-weight MCA は，

- 抽出する要素の通信量が固定されていない．
- 正方形を除く矩形の抽出が考慮されていない．
- トラフィックデータから抽出するべきパターンの優先順位を考慮していない．

これらの問題点を考慮して， 私たちは今後 Light-weight MCA を改良していく予定である．

7 結論

CNRS 二国間共同研究（代表：江崎浩，Patrice Abry）の枠組みでの学生派遣として，慶應義塾大学大学院 政策・メディア研究科 徳田研究室 修士課程 1 年の加藤碧が Ecole Normale Sup ´

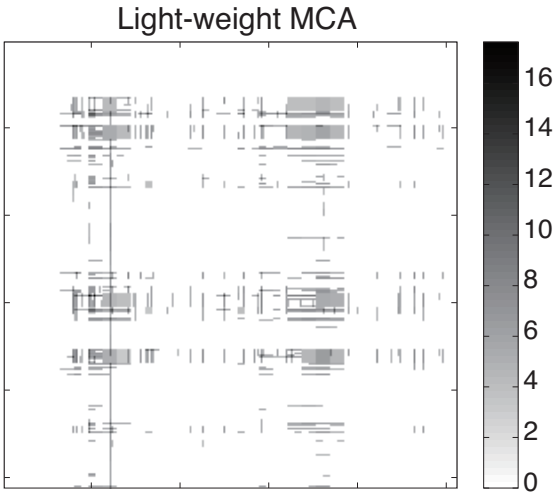

図 5: Light-weight MCA の実験結果

erieure de Lyon (ENS-Lyon) で 1 ヶ月間 Patrice Abry らの下で研究活動を行った．トラフィック解析の結果はネットワークのトラフィック監視の指標になるにも関わらず，フロー集約は，科学的に定式化されていない．そのため，本学生派遣において，画像処理技術 MCA を応用してフロー集約の定式化を試みた．

トラフィックデータを用いた実験により，MCA は計算コストが高く，要素抽出方法に問題があるということが明らかになった．そのため，MCA を改良して light-weight MCA を開発したが，light-weight MCA もいくつかの問題点を残している．フロー集約の定式化を実現するために，Light-weigh MCA の改良を今後の展望とする．

参考文献

[1] Caida. URL http://www.caida.org/home/.

[2] K. Cho, R. Kaizaki, and A. Kato. Aguri: An aggregation-based traffic profiler. In *Quality of Future Internet Services*, 2001.

[3] C. Estan, S. Savage, and G. Varghese. Automatically inferring patterns of resource consumption in network traffic. In *SIGCOMM*, 2003.

[4] J. Fadili, J.-L. Starck, M. Elad, and D. Donoho. Mcalab: Reproducible research in signal and image decomposition and inpainting. *Computing in Science Engineering*, 12(1):44 –63, jan.-feb. 2010.

[5] J. M. F. Y. M. Jrme Bobin, Jean-Luc Starck and D. L. Donoho. Morphological component analysis: An adaptive thresholding strategy. In *IEEE TRANSACTIONS ON IMAGE PROCESSING*, volume 16, Nov. 2007.

[6] M. Kato, K. Cho, M. Honda, and H. Tokuda. Monitoring the dynamics of network traffic by recursive multi-dimensional aggregation. In *MAD, OSDI workshop*, 2012.