IMPERIAL COLLEGE LONDON
DEPARTMENT OF COMPUTING

Pedro Kostelec
CID: 00655960

CELL TRACKING METHODS

LITERATURE SURVEY

Supervisor: Ben Glocker

London, May 2014

Contents

Chapter 1

Introduction

This literature survey describes the background research that has been completed in preparation for the work on the project entitled "Automatic Cell Tracking and Categorisation for Microscopic Image Analysis".

Quantitative analysis of cell populations using time-lapse microscopy is important for understanding cell behaviour. This analysis requires tracking a large number of cells in often low quality image sequences of varying contrast. Often then most effective way to do this is to manually annotate each cell in every frame of the sequence. However, this is slow and tedious, thus limiting the amount of data that we are able to analyse. Several computer vision algorithms have been developed to reduce the amount of manual work required for the analysis of large time-lapse sequences. Ideally, we would like to develop an algorithm that is able to track any type of cells in image sequences, regardless of the imaging technique used to capture them. However, the current state-of-the-art in computer vision is unable to perform this task accurately and automatically. For this reason, several algorithms have been developed to handle specific cases, relying on heuristics to improve their performance.

The main emphasis of this project is to track leukocytes in low quality image sequences acquired with fluorescent reporter technology and light microscopy. Additionally, the behaviour of these cells with be quantified to allow for further studies of leukocytes.

This is a joint project with Dr. Leo Carlin from the Leukocyte Biology Section at the National Heart and Lung Institute (NHLI) and is supervised by Dr. Ben Glocker.

The remaining of the report is divided into two chapters. Chapter 2 summarizes the essential material related to cell detection and tracking. In the Chapter 3 we present an informative work plan that will guide the develop-

ment of the software.

Chapter 2

Cell tracking methods

This chapter is an overview of the background research that has been completed in preparation for the work on the project. The chapter is divided into four sections. Section 2.1 describes methods to perform cell detection on images containing cells. Section 2.2 presents methods used to track cells in a sequence of images. Finally, in Section 2.3 we describe which methods seem most promising for this project.

2.1 Cell detection

Cell detection consists in identifying individual cells in microscopy images. Due to the different imaging techniques and the types of cells we may wish to detect, several algorithms have been developed each to handle a specific case. This section is an overview of these techniques.

2.1.1 Cell segmentation using the Watershed technique

A basic cell detection method relies in binarizing an image to separate the background from the cells, followed by a segmentation step to extract the cells. [1] uses a flooding approach for segmentation. Then entire method can be summarized as follows:

1. Apply a spatial adaptive filter to the image to minimize the effect of noise.

2. Locate the pixels with minimum intensities in a small sliding window.

3. For each minimum point we proceed to the progressive flooding of its neighbouring points

4. Post-processing step to discard false regions.

A similar approach would perform binarization with Otsu's threshold selection method [2], followed by some morphological operations [3] to fill holes and eliminate patches that are too small to correspond to cells, and finally use the Watershed algorithm [4] to segment the binary image into individual cells. The disadvantage of this method is that the number of pixels belonging to either cells or background should be approximately the same, and that the signal-to-noise ratio should be low.

Chen et al [5] have used an improved Watershed algorithm [6] to separate cell nuclei after using Otsu's thresholding method to segment nuclei from the background. Additionally they developed a nuclei-fragment merging method based on the shapes and sizes of the nuclei to deal with the problem of over-segmentation caused by the Watershed algorithm.

2.1.2   Cell segmentation using level sets

Another interesting technique to segment cells is a contour evolution method that makes use of the general appearance of cells to segment them using level sets. Mukherjee [7] makes the observation that leukocyte shapes are nearly circular cells and that at least for a significant part of the border of the cell, the intensity profile is different from the cell cytoplasm and from the background. Using this observation, identification of a leukocyte is formulated as a minimization of an energy function incorporating image gradient and intensity homogeneity within the closed contour encompassing the cell. The benefit of this method is that it can be adjusted to perform well in images with significant clutter and poor contrast by increasing the importance of the homogeneity, or for images with good contrast, where the gradient magnitude term is given more importance. The disadvantage of this method is that cells cannot overlap. The energy function can be minimized with the gradient descent method.

To reduce the solution space for the energy function only the boundaries of connected components within the image-levels sets are evaluated with the energy function. Only the connected components satisfying the size and shape constraints of the cells are extracted. The remaining components are eliminated using area morphology operations. This level-set analysis provides a more efficient solution that is linear in the number of intensity levels in the image in contrast to the much higher complexity of a curve evolution method.

The level set method is contour evolution approach which has good results in segmentation. Tang et al [18] have successfully combined level-sets and local grey thresholding [8] for segmenting neuron stem cells in images which have been obtained by confocal microscopy.

2.1.3 Cell detection by model learning

The previous methods perform efficiently in cells with sufficiently good contrast. In images where the cell borders are unclear, images are of varying intensity, cell density is high, or cells can be of different shapes, these methods would not perform as well. In such cases machine learning methods can perform better by learning a model of a cell based on a large number of annotated examples.

Arteta et al [9] [10], propose an algorithm that uses a highly-efficient MSER region detector [11] to find a broad number of candidate regions. These regions are then scored depending on the similarity to the cell type of interest by a machine learning algorithm.

The authors organized the extremal regions obtained from the MSER detector into trees, so that each tree corresponds to a set of overlapping extremal regions. The non-overlapping regions which achieve high scores can then be selected using dynamic programming of the trees. Two learning strategies are tested: a binary classifier using Support Vector Machines (SVM) and structured learning (structured SVM [12]) which is able to take into account the non-overlap constraint, and achieves better performance. The learning is performed on weakly annotated images – a single dot is necessary on each cell.

The feature vector for each regions is composed of several concatenated histograms: a histogram of intensities within the region, two histograms of differences in intensities between the region border and a dilation of it for two different dilation radii, a shape descriptor and the area of the region. The downside of this approach is that extracting the feature vector from the image is slow, especially because it needs to be extracted for each image in the sequence.

The advantage of this approach is the tolerance to changes in image intensities, cell densities and sizes. The major downside is the non-overlap constraint. Fortunately the authors have also developed an algorithm to detect partially overlapping cells [10].

The idea is to learn to detect overlapping cells, and the number of cells in the region. The algorithm starts by generating a set of nested regions. Each region is then scored using a set of classifiers that evaluate the similarity of

the region to each of the possible classes, where each class corresponds to the number of cells that the region contains. An inference procedure then selects the non-overlapping subset of regions, and assigns each a class label indicating the number of cells that the model believes lie in the region.

2.1.4   Cell detection by image restoration

Another approach is to use an image restoration technique followed by thresholding [13] [14]. Bise et al [13] apply this technique on phase-contrast microscopy images . The technique utilizes the optophysical principle of image formation by phase-contrast microscope to transform the image into an artefact free image by minimizing a regularized quadratic cost function. After this, a simple image thresholding technique can be used for segmentation.

2.2   Cell tracking

After detecting each cell in each image of the sequence we need to determine which cells in an image correspond to which in the next image. This way we can generate tracks of the cells across the sequence. This is the problem of tracking. There are three main approaches to tracking:

Tracking by model evolution Active contours or level sets are used to detect a cell in the first frame of the sequence, and the cell models are then propagated to the next frame. This method can be computationally expensive because of the need to evolve the cell models for each frame.

Tracking by data association Cells in consecutive frames are associated according to a similarity measure, which compares features such as the cell intensity of relative spatial location. This method requires very accurate segmentation to perform well.

Tracking with a dynamics filter This method uses a dynamics filter (e.g. Kalman or Particle filter) to predict the spatial location of the cell in the next frame.

The remaining of this chapter describes how these methods have been implemented in different papers.

2.2.1   Tracking by model evolution

Model evolution tracking methods handle cell deformation well, but can often get stuck in local minima. It is possible to improve the tracking accuracy at the expense of computation time.

Mukherjee [7] models the problem of cell tracking as maximizing a similarity measure between level sets in consecutive frames. The method minimizes the energy associated with leukocyte boundary detection and then match the boundary with that of the previous frame. To maintain spatial coherence, the authors assume that the displacement of cells between frames is marginal. Such an automatic tracker is able to handle slight rotations or deformations of the leukocytes well.

2.2.2   Tracking by data association

When tracking by data association, cells in consecutive frames are associated according to a similarity measure, which compares features such as the cell intensity or relative spatial location. This method is very efficient, but only effective when cells are accurately detected. It is possible to improve the association by analysing several frames in a sequence.

Chen et al [5] perform the matching process based on the distances between the cells. An association matrix is built based on the Euclidean distance between cell nuclei and their sizes. A match is found if the distance is below a threshold. Four possible outcomes are possible in this matching process:

1. A successful one-to-one match is identified.

2. A nucleus moves out of the field of view, which can only happen in the borders.

3. Nucleus separation.

4. Several nuclei in one frame match a single nucleus in the following frame. This can be the result of under-segmentation.

The authors have also developed strategies to overcome the problem of false matched and ambiguous correspondence when nuclei are touching or partially overlapping. In this case information (nuclei size and their relative location – up, down, left, right –) about these nuclei is stored. When they separate, the algorithm can resolve the ambiguities by comparing their current status with the previously stored information.

Over-segmentation is resolved by merging fragments if the sum of sizes of the close fragments corresponds to the size of a nucleus in the previous frame.

Cells that undergo division or occlude other cells make the problem of tracking harder. Baker et al [15] used a contour-based segmentation method. Cells with a common contour are tagged as siblings and undergo additional tests to determine if they are part of division or occlusion events based on whether both cells existed previously or not. Occlusions are further examined and mislabelled cells are corrected.

Cell division detection is crucial for developing an accurate tracking algorithm in cell populations where mitosis is frequent, as is demonstrated in [14]. Based on the segmented blobs in the previous frame frame-by-frame data association is performed in the current frame. The authors categorized five hypothesis about cell behaviour: migration, exit, entrance, clustering and mitosis. The hypothesis regarding mitosis detection often produces too many false positives, and sometimes may not detect mitosis even if it should. To mitigate this effect, a cell mitosis detection algorithm is run independently on the image sequence. The mitosis hypothesis is only evaluated for the cells that have been discovered by this algorithm. After all hypothesis are obtained, a best combination is selected by formulating and solving an integer programming problem. If, after the best association is found, there are remaining cells, these are considered again in the following frames. The association is attempted for the following 10 cells before the track is marked as terminated. A similar method is used to generate new tracks for newly detected cells. Each such cell generates a temporary track, which is confirmed as real track if the track is associated in most of the following 10 frames.

The similarity function used to perform data association between images is of crucial importance. House et al [16] propose a novel cost function that encodes the response of cells to conditions of their environment. The tracking problem is formulated as a standard Bayesian filtering problem. The model allows the state of the cells in the image to evolve in time. To solve the complete assignment two algorithm are evaluated: an iterative probabilistic data association algorithm, which produces an optimal solution, and a two-phase batch algorithm, which produces a suboptimal solution. The performance of both algorithms is dependent on the cost function used. The authors propose and evaluate three different cost functions. The two-phase batch algorithm has difficulties when cells occlude each other, or die or divide in close proximity to another. It performs very well in low-density population with few events occurring. The optimal algorithm, which performs much better in these harder scenarios, has had most difficulties in matching daughter cells to their parent

cell.

2.2.3   Tracking with a dynamics filter

Dynamics filters are a powerful addition to tracking systems because they predict the motion of a cell, and reduce the search space in which we look for matches. These algorithms not only perform frame-by-frame tracking, but also take temporal information into account, resulting in more accurate tracking performance.

Xu et al [17] developed an ant stochastic searching behaviour based tracking system to track multiple cells in fluorescent image sequences. The system is similar to particle filters, but the motion behaviour is modelled to be similar to how ants behave when searching for food. When ants move around, they deposit small quantities of chemical pheromone. Once they find food, they retrace the steps deposing larger amounts of pheromone. Other ants can then use this guide to find food faster, but are allowed to deviate from the path and pursue their own paths. The author propose a system where the food is not static, so that it can be used for cell tracking.

The predicted state of each an is computed based on a dynamic model. Then the importance weights for the predicted states are normalized. An ants moves first in its predicted state. Afterwards, its moving direction can be influenced by the states of other ants. When the ants finish their path selections, the pheromone update process takes place. In this algorithm the updated pheromone refers to the moving ability of an ant. Ants with larger weights are located closer to the food, and move slower. Ants with lower weights are further away from food and move more aggressively, thus giving them a better change to quickly converge on the food.

Compared to particle filters, this algorithm is more computationally expensive, but achieve better accuracy. The algorithm is able to deal with cells entering or leaving the scene, and occlusion.

Kalman filters are another popular method for state prediction. Tang et al [18] use a Kalman filter to guide the tracking of active cells. Active cells are those that move a distance larger than L (supposedly their radius) between frames. Inactive cells are tracked by observing the overlap region. The state vector for the Kalman filter is composed of the cell two dimensional position and the corresponding velocities. The system external input vector is composed of the acceleration in two dimensions. To initialize the Kalman filter, the cells are first tracked in the first three frames by means of a similarity function evaluated in neighbouring locations in successive frames. From the

third frame onward, the Kalman filter is used to predict the cell's location and reduce the search space.

Cell do not always exhibit just one type of motion and a single Kalman filter cannot take this into account. Li et al [19] propose an automated tracking system which runs several Kalman filters in parallel, each corresponding to a different modality. Their system is composed of five modules: cell detector, cell tracker, dynamics filter, track compiler and track linker.

The cell detector separates the background from cell pixels. The cell tracker propagates the cell labelling to the next frame in the sequence. The track compiler compares the results of the cell tracker, cell detector and dynamics filter to create a new track for new or daughter cells, and updates or terminates existing tracks. The track linker connects two or more track segments if they belong to the same cell.

Several assumption are made regarding the appearance and motion of the cells. First they are approximately elliptic in shape. Second, their position and intensity change are governed by independent processes. Third, the dynamic processes governing each cell are independend. Suppose that the cellular motion consists of a finite number of modes. The interacting multiple models (IMM) filter operates several Kalman filters in parallel, each modelling a different mode of motion. The transition between models is determined by a finite state Markov chain. The author use three models of cell dynamics: random walk, first-order and second-order linear extrapolation which correspond to Brownian motion, migration with constant speed and migration with constant acceleration. To initialize the IMM filter the system tracks the cells in the first three frames in which they appear. The parameters of the model are estimated offline using the Expectation-Maximization algorithm.

This model is robust to long-term occlusion and cell detection error.

2.2.4   Cell tracking as a network flow problem

Massoudi et al [20] propose a tracking method that does not rely on perfect segmentation and can deal with uncertainties by exploiting temporal information and aggregating the results of many frames. The system only requires probabilities or potentials that represent cell positions. Tracking is modelled as a network flow problem and is formulated as a linear program based on the occupanty likelihood of the edges of the network. The benefit of this approach is that it aggregates the results of all frames and makes a global decision rather than propagating the results of each frame to the next, which makes it less dependent on the cell detection module. The system can detect false positives

or false negatives and correct itself. The method handles division events, cells entering or exiting the screen at the boundaries.

2.3   Conclusion

This chapter presented an overview of methods used for cell detection and tracking. In the future, specific methods will be selected and analysed based on the quality of image sequences that we will be analysing. The small sample of images made available in the past week by Dr. Leo Carlin show low contrast cells and rapidly changing morphology. The images were low quality so that basic computer vision techniques will probably not suffice for efficient cell segmentation and tracking. First some image restoration preprocessing steps will need to be performed to reduce the artefacts caused by the imaging method. It seems that a machine learning approach, similar to [10] will be most effective to detect the cell position. According to Dr. Leo Carlin, few or no mitosis events will be present in the image sequences that we will study, so a section on mitosis detection has been left out of this literature survey. Because of the low quality images a data association approach will probably not be very accurate for tracking. A dynamic filter approach will most likely be used.

There are several reasons why automatic cell tracking systems are not as popular as we would expect. The main reason is probably that the level of accuracy falls short of manual annotation. Furthermore, these systems are often not robust to changes in cell type, cell culture condition and microscopy image quality. The goal of this research is to develop a system that can track leukocytes in low quality images with an accuracy that is similar to that of a human.

Chapter 3

Work plan

In order to complete the project in due time, a work plan is outlined to guide the development of the software.

The first phase of the project was background literature research and an overview of the learned methods and techniques is outlined in the previous chapter. Additionally, I have also performed background research on cell mitosis. These notes are not included in this background survey, because the image sequences we will study have few or no mitosis events.

3.1 Architecture

Similar to the work by Li et al [19], a modularized approach is going to facilitate development and testability of the software. This approach has the benefit that any of the modules could be swapped without affecting the other parts of the system. Li et al have used the following modules: Cell Detector, Cell Tracker, Dynamics Filter, Track Compiler and Track Linker. These modules will be used as a guideline when developing our own version of the tracker for leukocytes. However, not everything will be the same. For example, their approach connects the Track Compiler to the Cell Detector modules, because the Track Compiler is able to provide the Cell Detector updated histograms that the Cell Detector uses for detection using a Bayesian classifier. This relationship is only necessary if the learning is done on-line, that is if the Cell Detector module is able to use these updated histograms to better detect new cells in the following images.

For the Cell Detector module, an off-line learning approach will probably be tested, similar to the one in [9] [10]. However the details will only be defined once sufficient sample image sequences are analysed.
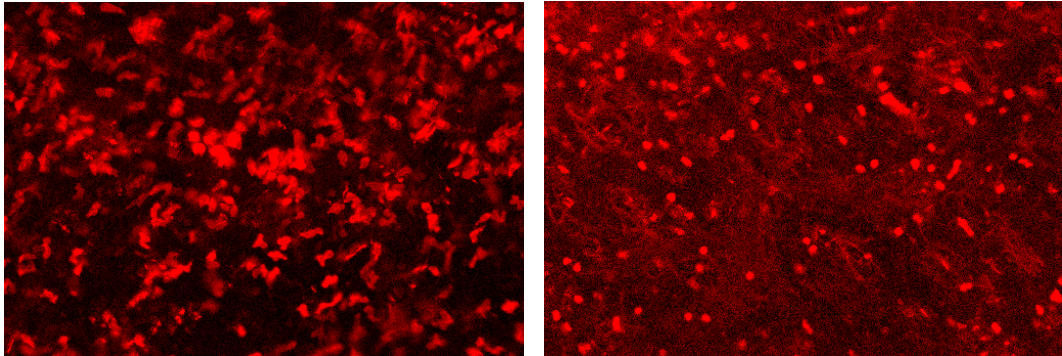
Figure 3.1: Two example frames of cell images that the tracker will need to successfully process and track the individual cells. The colour of the images has been enhanced for this display.

An additional module will be developed to quantify the behaviour of the cells over time. Cell displacement, the proportion of time they move in close proximity to other cells, the changes of cell velocity over time, ... are examples of measurements that it will be responsible for.

The program will be developed in MATLAB and will include an executable GUI that will permit the user to use the application without the entire MAT-LAB package.

3.2 Image sequences

Up to date, only one sample of an image sequence was made available. The data is provided by Dr. Leo Carlin from the Leukocyte Biology Section at the National Heart and Lung Institute (NHLI). He has promised to provide more data as soon as it is obtained.

Figure 3.1 displays two sample frames from the provided image sequence. The images are low quality, and the cell borders have low contrast. Obtaining more sample data is crucial to be able to shortlist which algorithms are appropriate for accurate cell detection and tracking.

3.3 Development schedule

The following is a rough outline of a work plan. This is a preliminary plan, which will be adjusted as work progresses.

May Focus will be on developing cell detection algorithms (the Cell Detection module). For this purpose a basic annotation program may also be written, which should ease annotating image sequences by clicking on cells.

June A basic tracking system that will track cells over time (the Cell Tracker module). Additionally, a module to compute interesting statistics of cell tracks will be developed. If there will be excess time, a Dynamics filter module will be developed.

July Focus will be on developing other modules as necessary, for example a Track Compiler and Track Linker. Also, the Cell Detection and Tracking modules will be improved if needed.

August Writing a GUI for the software, testing and writing the final report.

Bibliography

[1] Y. Chen, K. Biddell, A. Sun, P. Relue, and J. Johnson, "An automatic cell counting method for optical images," in *[Engineering in Medicine and Biology, 1999. 21st Annual Conference and the 1999 Annual Fall Meetring of the Biomedical Engineering Society] BMES/EMBS Conference, 1999. Proceedings of the First Joint*, vol. 2, pp. 819 vol.2–, Oct 1999. 5

[2] N. Otsu, "A threshold selection method from gray-level histograms," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 9, pp. 62–66, Jan 1979. 6

[3] J. Serra, *Image Analysis and Mathematical Morphology.* Orlando, FL, USA: Academic Press, Inc., 1983. 6

[4] S. Beucher and C. Lantuéjoul, "Use of watersheds in contour detection." workshop published, Sept. 1979. 6

[5] X. Chen, X. Zhou, and S.-C. Wong, "Automated segmentation, classification, and tracking of cancer cell nuclei in time-lapse microscopy," *Biomedical Engineering, IEEE Transactions on*, vol. 53, pp. 762–766, April 2006. 6, 9

[6] L. Vincent, "Morphological grayscale reconstruction in image analysis: applications and efficient algorithms," *Image Processing, IEEE Transactions on*, vol. 2, pp. 176–201, Apr 1993. 6

[7] D. Mukherjee, N. Ray, and S. Acton, "Level set analysis for leukocyte detection and tracking," *Image Processing, IEEE Transactions on*, vol. 13, pp. 562–572, April 2004. 6, 9

[8] D. Xu and L. Ma., "Segmentation of image sequences of neuron stem cells based on level-set algorithm combined with local gray threshold.," Master's thesis, Harbin Engineering University, 2010. 7

[9] C. Arteta, V. Lempitsky, J. A. Noble, and A. Zisserman, "Learning to detect cells using non-extremal regions," in *Proceedings of the 15th International Conference on Medical Image Computing and Computer-Assisted Intervention - Volume Part I*, MICCAI'12, (Berlin, Heidelberg), pp. 348–356, Springer-Verlag, 2012. 7, 14

[10] C. Arteta, V. S. Lempitsky, J. A. Noble, and A. Zisserman, "Learning to detect partially overlapping instances.," in *CVPR*, pp. 3230–3237, IEEE, 2013. 7, 13, 14

[11] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide baseline stereo from maximally stable extremal regions," in *Proceedings of the British Machine Vision Conference*, pp. 36.1–36.10, BMVA Press, 2002. doi:10.5244/C.16.36. 7

[12] T. Joachims, T. Finley, and C.-N. J. Yu, "Cutting-plane training of structural svms," *Mach. Learn.*, vol. 77, pp. 27–59, Oct. 2009. 7

[13] R. Bise, T. Kanade, Z. Yin, and S. il Huh, "Automatic cell tracking applied to analysis of cell migration in wound healing assay," in *Engineering in Medicine and Biology Society,EMBC, 2011 Annual International Conference of the IEEE*, pp. 6174–6179, Aug 2011. 8

[14] S. Huh, *Toward an Automated System for the Analysis of Cell Behavior: Cellular Event Detection and Cell Tracking in Time-lapse Live Cell Microscopy*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, March 2013. 8, 10

[15] R. Baker, M. Brasch, J. Henderson, and M. Manning, "Contour-based algorithm for tracking cells and cell-material analyses," in *Bioengineering Conference (NEBEC), 2013 39th Annual Northeast*, pp. 96–97, April 2013. 10

[16] D. House, M. Walker, Z. Wu, J. Wong, and M. Betke, "Tracking of cell populations to understand their spatio-temporal behavior in response to physical stimuli," in *Computer Vision and Pattern Recognition Workshops, 2009. CVPR Workshops 2009. IEEE Computer Society Conference on*, pp. 186–193, June 2009. 10

[17] B. Xu, M. Lu, P. Zhu, Q. Chen, and X. Wang, "Multiple cell tracking using ant estimator," in *Control, Automation and Information Sciences (ICCAIS), 2012 International Conference on*, pp. 13–17, Nov 2012. 11

[18] C. Tang, Y. Wang, and Y. Cui, "Tracking of active cells based on kalman filter in time lapse of image sequences of neuron stem cells." 11

[19] K. Li and T. Kanade, "Cell population tracking and lineage construction using multiple-model dynamics filters and spatiotemporal optimization," in *Proceedings of the 2nd International Workshop on Microscopic Image Analysis with Applications in Biology (MIAAB)*, September 2007. 12, 14

[20] A. Massoudi, D. Semenovich, and A. Sowmya, "Cell tracking and mitosis detection using splitting flow networks in phase-contrast imaging," in *Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE*, pp. 5310–5313, Aug 2012. 12