Capabilities and Objectives of Distributed Image Processing on Smart Camera Systems

Rene Golembewski, Steve Goering, Guenter Schaefer
Ilmenau University of Technology, Germany
Email: [rene.golembewski, steve.goering, guenter.schaefer]@tu-ilmenau.de

***Abstract*—The challenge of bringing more intelligence to the infrastructure of modern cities requires a change of thinking and states a demand for new algorithms and strategies. One important outcome of those algorithms is a realtime estimation of the prevalent spatio-temporal conditions of public transportation networks by making use of distributed image processing on networked smart camera systems. This paper provides a detailed analysis of two exemplary networked applications that can use the derived data. A conducted simulation study based on the infrastructure of real cities shows the potential of using autonomously generated knowledge, that smart camera systems can provide. Especially, inter-camera object tracking, as well as adaptive and smart navigation tasks can benefit considerably and substantiate the need for autonomous and confidential image processing.**

I. Introduction

In recent years, a series of efforts are undertaken in order to bring more intelligence to the public infrastructure of big cities to solve essential problems like public security, smart navigation or traffic management [1], [2], [3]. The city of New York just spent 40 million dollars for a super computer system developed by Microsoft, in order to analyze the output of thousands of surveillance camera systems [4]. A common approach for this is to capture all the camera systems that are mounted at the most frequented and important transportation hubs, and perform an offline analysis of the recorded video material by security personnel in case of an incident. Nevertheless, this is a rather trivial approach because realtime live surveillance by humans is not only very expensive but also not feasible for really large scenarios with thousands of cameras. Furthermore, the aggregation of surveillance data at a distinct point in the network creates a single point of failure (SPoF), and raises serious privacy concerns, e.g., attackers would have a well known target to concentrate on and unauthorized persons could potentially get access to all surveillance data at one central point. Additionally, this approach requires an enormous amount of network bandwidth and processing power, concerning the steadily improving camera sensors and image resolutions.

In the course of the general trends in decreasing prices and increasing processing power of miniaturized and integrated devices, new forms of Distributed Smart Cameras (DSCs) have been developed over the last years that allow for distributed analysis of video material on the spot and enable to avoid the expensive transport of raw camera data. Additionally, a meanwhile cheap and ubiquitous networking enables the setup of a connected camera infrastructure in order to exchange pre-processed data (e.g., characteristic fingerprints of faces, etc.). The idea is to use this infrastructure for innovative applications, e.g., the tracing of criminal acts, or navigation tasks that are adapted to the actual traffic conditions of the system. Figure 1 illustrates a basic task in distributed image processing: The estimation of traffic flows in urban scenarios without any further requirements to the camera systems, like a shared field of view, calibrated devices, or special training procedures. Even the available bandwidth for communication can be very low, because DSCs are assumed to exchange only anonymized metadata about the locally analyzed events.

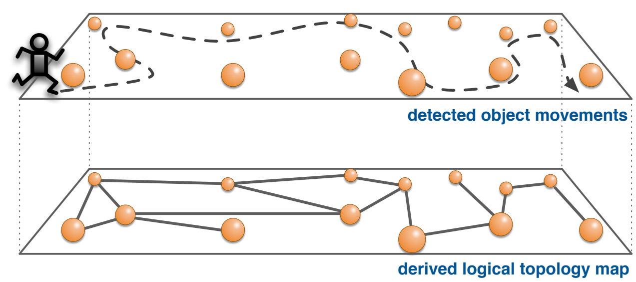

Fig. 1. Smart camera nodes estimate a logical topology map based on monitored object transitions and distributed vision algorithms. Therefore, no video material has to be transmitted, and communication overhead can be saved.

Upcoming strategies for distributed image processing address the basic problem of topology estimation [5], [6], [7], and raise the question of how much networked applications can benefit from the derived information. Therefore, this paper provides an outline and a detailed analysis of two exemplary networked applications that can use the derived data. On the one hand, the problem of inter-camera object tracking in highly diverse scenarios and without any further knowledge about the underlying camera setup is investigated. On the other hand, a new approach of smart navigation in public transportation networks is introduced that exclusively uses networked smart cameras and anonymized image processing as a source of information. Both applications provide the background for a following requirement analysis that states the main non-functional objectives for DSC systems.

A conducted simulation study based on the infrastructural properties of real cities shall reveal the potential of using autonomously generated knowledge that smart camera systems can provide, in contrast to common approaches with limited and rather static knowledge. Therefore, traffic networks of a variety of big cities are analyzed and used as input for simulation tasks and the generation of synthetic simulation topologies following the extracted metrics [8]. The implementation of the described applications is kept simple and concentrates on basic algorithms in order to provide a generic model for further research in the field of distributed and networked applications in the context of smart cities and image processing.

II. Illustrative Applications

The following section provides two demonstrative applications that can profit from an autonomous and distributed image processing, directly on the smart camera devices. Of course, a centralized processing of images on dedicated servers can provide the same output, but requires a lot more communication bandwidth and creates bottlenecks as well as privacy issues, concerning the aggregation of intimate surveillance data (see Section III-A).

A. Inter-Camera Object Tracking

The main scope of object tracking between autonomously operating camera nodes is to identify certain trajectories of moving objects while keeping the efforts for communication and processing power at a minimum. This can be done by continuously identifying spatio-temporal relationships between those nodes in advance, and use this topological knowledge to save resources afterwards. A detailed and mathematical formulation of how the topology information is derived in realtime is given in [7].

A rather static scenario arises if only the geographical coordinates of all camera nodes are known without any further information about paths or means of transportation. Objects of interest that have been recognized at time t at a certain node in the scenario are assumed to have a maximum motion speed. That means they can reside anywhere in the surrounding of that node, with steadily increasing radius r over time $t+\Delta t$ (see Fig. 2). The growing geographical area is therefore illustrated as concentric circles with the object's origin as centre.

Due to the fact that none of the nodes has the information about potential trajectories in the scenario, all nodes within the reachable area must be monitored to recognize the object of interest (see Fig. 2). Nevertheless, without this geographical knowledge, all existing nodes in the system would have to be monitored. This approach for shrinking the search space is therefore expected to work, but requires a manual setup of coordinates or special GPS functionality on each participating system.

Thus, an adaptive and autonomous algorithm that estimates the actual traffic flow conditions without any prior information about location and neighborhood is expected to outperform the location-based approach. Figure 3 depicts the same scenario but now with the previously estimated spatio-temporal relationships between camera nodes. Based on that information, some of the former interesting nodes can be excluded from the set of relevant and reachable nodes and thus, the search space can be limited even more because only a subset of paths is avaiable for the object's movement through the surrounding.

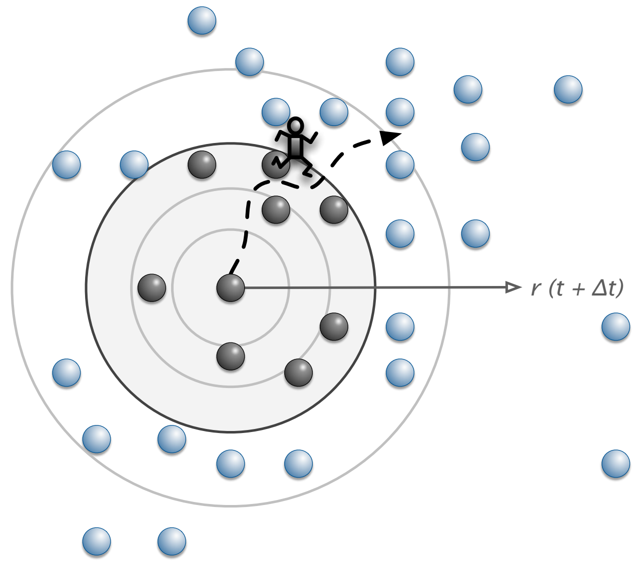

Fig. 2. An object (e.g., a suspicious person) shall be located by distributed cameras, after it has been identified at a single point in the network, a certain amount of time Δt before. The main challenge: Limiting the search effort to some few camera nodes. If the geographical position of every single camera node is known, the search radius $r(t + \Delta t)$ must be extended based on a constant moving speed of the object.

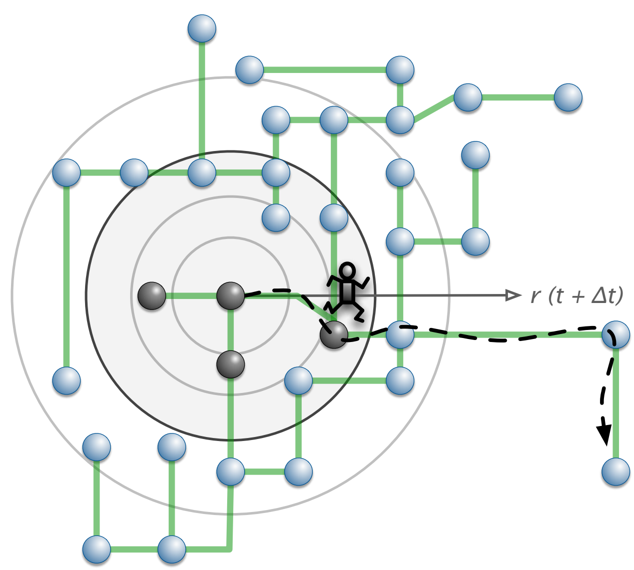

Fig. 3. Even without any knowledge about positions or distances of nodes, a logical topology based on estimated transition times can support search algorithms by limiting the number of plausible destinations and paths.

B. Smart Navigation

Another important task in modern cities is smart and adaptive navigation that takes into account the actual traffic conditions, like congestion, road works, or rush hours with long waits. Depending on the time of the day, there may be alternative routes in the transportation networks or on the streets, that finally lead to a shorter overall travel time, although the distance increases (see Fig. 4). This is a plausible fact for any person that ever used transportation networks during rush hours, but it is hard to predict or to analyze such situations in realtime and without computational intelligence.

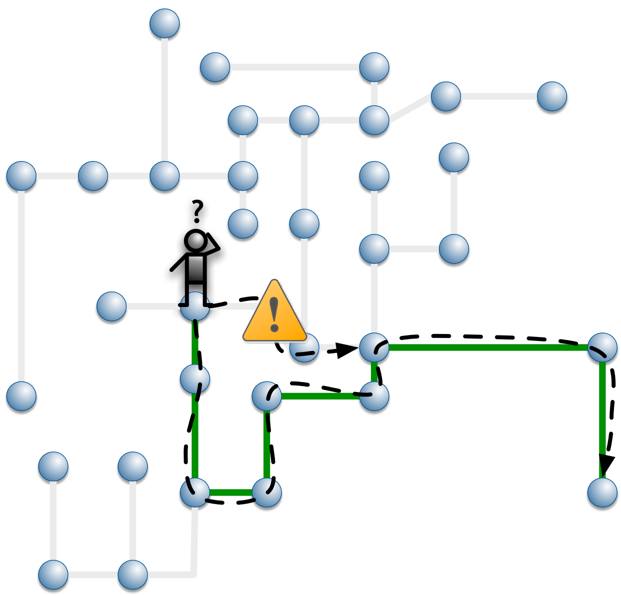

Fig. 4. Smart and adaptive navigation if the former shortest path is no longer available or heavily congested.

In order to realize an adaptive navigation application, the knowledge about transition times between particular stations is essential. Of course, even with overcrowded stations, the metro transit times are quite constant (according to the schedule), but it takes much longer for a person to reach the stop and get on the trains. And this is actually the interesting Problem: How long does it take for the passengers to move between different stations?

III. Objectives for Distributed Image Processing

Besides the functional requirements for distributed and autonomous processing of video material, there are some main and yet important non-functional requirements that have to be fulfilled. Future distributed networked applications must be designed with respect to these objectives, in order to be applicable in growing public environments.

A. Privacy

Probably the most important point when camera systems should be applied in public areas is the privacy aspect. Administrative instances have to follow very strong guidelines whenever intimate and private data, such as camera images, is recorded or transmitted over communication networks. That is because of the potential presence of electronical eavesdropping operations at any readily accessible and thus vulnerable point in the infrastructure. Image processing on centralized systems is a comparatively simple approach, and once compromized, an enormous amount of aggregated data is accessible for unauthorized people. Additionally, also security personnel should not be able to get access to any kind of recorded material, unless it is really necessary. An innovative approach should therefore provide the possibility of preprocessing the private data and use anonymized metadata in combination with intelligent algorithms instead. Furthermore, smart algorithms that try to extract useful information about the monitored scenarios can overcome the paradigm of saving all the binary image data for later analysis.

B. Scalability

With an actual size of thousands of cameras mounted almost everywhere in big cities, scalability problems will arise with steadily increasing image resolutions in combination with networked data transport. Although data rates can be reduced drastically by distributed preprocessing of images, another class of problems arises without centralized coordination and management functions. DSCs will have to communicate on a peer-to-peer basis and work autonomously to derive infrastructural behavior. Therefore, such a system can be seen as scalable concerning the number of nodes, if it is guaranteed that there are enough resources available to communicate over the transport network and to process all gathered information in real time. A smart communication overlay can support location awareness and reduce the required communication bandwidth [7].

C. Adaptivity/Agility

Due to the continuous evolution of traffic situations, e.g., because of congestion, road works, or special events, the whole public transportation infrastructure can be seen as a dynamic system. Special training phases or unique training objects for camera calibration must be disregarded as a consequence. In order to use infrastructural information for smart applications, the estimation of traffic flows and transit times should be up-to-date at any time. Hence, collaborative image processing over communication networks must be realized with realtime capabilities to support smart features.

D. Robustness and Precision

With increasing distributed processing power, also object recognition and tracking algorithms improve steadily. But there still exists the fundamental challenge of classifying complex objects in a diverse environment, like people in public places. The main problems that arise are “False Negative” object detections because of coverage or perspective issues on the one hand, and “False Positive” recognized objects because of the limited size of feature descriptors and mis-recognition on the other hand. Since algorithms for autonomous image processing are computed in a distributed fashion, they should be robust against corrupted or distorted input data [9]. In this respect, the most important component is the computer vision part that provides information about detected and classified objects in video material. As an example, with more than 10.000 people moving in a public transportation system simultaneously, object tracking and object recognition algorithms still are stretched to their limits. This would inevitably result in high error rates and influence the functioning of further applications.

Conclusion

The main target to enable autonomous processing of video material on distributed devices, is to build up a smart communication overlay that takes into account the actual knowledge about traffic flows. But the logical relationship between distributed instances often differs from geographical circumstances. Applications for smart navigation or inter-camera object tracking are assumed to benefit from information about the logical topology of smart camera nodes and the traffic flows between them.

IV. Evaluation

In order to evaluate the advantage of spatio-temporal topology knowledge, we extracted some real traffic networks of different cities based on the free datasets of OpenStreetMap (OSM) [10] in the first place. We chose big cities from different countries to get a representative sample. Afterwards, the cities' XML files are analyzed, and corrected with respect to the public transportation networks (stations, stops, routes, etc.). This extraction process consists of the following steps. First, the relations of the XML files are parsed and filtered, because only light rail and subway information is needed, especially the stations. Based on this results, the needed paths and nodes (with GPS positions) are selected. The resulting graph contains all GPS measurements and routes. In order to clean this data and reduce the number of additional waypoints, the GPS graph is generalized as a routing graph where every station is a node for routing. To get the egdes/weights between the routing nodes, an algorithm for finding shortest paths (e.g., Dijkstra) is applied on the GPS graph. Furthermore, overlapping stations (e.g., OSM tagging failures or parallel tracks) are merged and the largest connected component of the graph is extracted. Then the resulting graphs of all cities were analyzed concerning average node degree and clustering coefficient. Table I shows the calculated outcome for a selection of nine cities. As expected, the (global) clustering coefficients for all cities were very low, because the establishment of clusters in a transportation network is not economic and indicates high redundancies of transport routes within a small geographical area. The more interesting metric is given with the average vertex degree for the particular cities. It seems that transportation networks of big cities have similar properties, even if the absolute size (points of interest) is very different [8].

Based on these numbers and some other, minor important details (e.g., the geographical density of nodes), we can generate synthetic topologies that have the same characteristics as the evaluated cities. A following simulation environment, based on the OMNeT++ Framework [11], can provide any number of random topologies in order to get statistically significant results and reliable statements.

TABLE I
Evaluation of real OSM-Data (light rail and subway transportation network)

City	Points of Interest	Edges	Vertex degree min	max	avg	Clustering coefficient
Berlin	376	983	1	7	2.61	0.14
Paris	314	763	1	12	2.43	0.04
New-York	219	603	1	13	2.75	0.10
Shanghai	210	452	1	5	2.15	0.02
Frankfurt	208	526	1	11	2.53	0.10
Moscow	161	352	1	4	2.19	0.01
Barcelona	140	305	1	5	2.18	0.00
Rome	75	152	1	4	2.03	0.02
London	54	125	1	8	2.31	0.15

A. Inter-Camera Object Tracking

The first application is based on the scenario described in section II-A. The core idea is to find the number of nodes an object can reach while starting from a node s within a certain time window Δt. Initially, to evaluate the given situation we need at least the number $n(s, \Delta t)$ from a set of camera nodes V an object can reach when starting at node s, an amount of time Δt before. Considering that every node can be the starting point, it is necessary to do an evaluation without a special node s. So we define $\overline{n}(\Delta t)$ the average number of possible target nodes for the tracked object as

$$\overline{n}(\Delta t) = \frac{1}{|V|} \sum_{s \in V} n(s, \Delta t).$$

Apparently, $\overline{n}(\Delta t)$ is in the interval $[0, |V|]$, but to compare different topologies we need to scale it to the uniform interval $[0, 1]$ and define it as $\underline{n}(\Delta t)$.

Now we can differentiate between two basic approaches. The first does not use any topological knowledge and is just based on the geographical information of every node. The calculation of $n(s, \Delta t)$ is a simple distance lookup of which nodes are reachable in the radius $r(t) = v \cdot t$ where v is the assumed object speed. The second approach uses the estimated spatio-temporal topology properties and calculates $n(s, \Delta t)$ with a truncated breadth-first search (tBFS). During the insertion phase, the tBFS ignores nodes that are not reachable in the time window Δt. The reachability is then calculated based on the topology knowledge, especially the observed time to reach a neighboring node.

The fundamental simulation uses a synthetically generated set of topologies of city terrains with $10 \times 10\ km^2$. Objects move with a variable moving speed of $v : N(\mu = 1.5\frac{m}{s}, \sigma = 0.3\frac{m}{s})$. Supposing a person moves from one corner to the opposite corner of the city region with a speed of $v = 1.5\frac{m}{s}$, the needed time is $t = \frac{10 \cdot \sqrt{2} \cdot 10^3}{1.5} s \approx 157$ minutes. We calculated $\underline{n}(t)$ in equidistant steps of 5 minutes with the two introduced methods. All measures are calculated means over 32 runs and additionally with 99% confidence intervals. The used topologies consist of 500 nodes with an average node degree of 2.521 and $\sigma(99\%) = 0.011$.

Figure 5 shows the measured $\underline{n}(\Delta t)$ values for all Δt starting from 0 to 100 minutes. First of all, the "Geographical Knowledge" series evaluates the approach based on geographical distances without any topology knowledge. E.g., at time $\Delta t = 50$ minutes the search space is approximately 80% of all nodes. On the other hand, the "Topology Knowledge" series shows a remaining search space of less than 55% of nodes. It uses the tBFS approach in combination with a shortest path metric, because in this scenario only the paths between nodes are known, but not the spatio-temporal relationship. The "Topology Knowledge (timed)" series considers delays or heavily congested paths, realized by modified distances between two congested nodes based on an exponential distribution with $Exp(\lambda = 1.0)$. This distribution was chosen because it ensures the similarity to the unmodified topology concerning the average path lengths. Summarizing, the global view of all three series shows that the search space increases slower for the topology based approaches, because only the previously identified paths are available for the object of interest. Furthermore, all series apply to the fact that in time $\Delta t > 100$ minutes, all nodes are potentially reachable for all of the implemented strategies. Additionally, we claim that the topology knowledge is quite useful for the search-task scenario. For all cases, topology knowledge could speed up the search algorithms, and in the best case with approximately 20%. Just for search scenarios during the first few minutes, the geographical approach is able to compete with the topology based ones.

B. *Adaptive/Smart Navigation*

In section II-B the second application was introduced. The general problem of adaptive/smart navigation is the well known shortest path problem of finding the best route from a given starting point to any destination. But for the simulative evaluation we have to solve the all pairs shortest path (APSP) problem, e.g., with the algorithm from Bellman and Ford, because every node could be the starting point. As input, at least the topology graph $G = (V, E)$ and a cost matrix $c[u, v]$ based on geographical distances are needed. An algorithm for the APSP problem returns a distance matrix $d(u, v)$ that holds the costs for the shortest path from node u to node v. Subsequently, we define $\overline{d}$ the average costs for all paths as

$$\overline{d} = \frac{1}{|V|^2} \sum_{v \in V} \sum_{u \in V} d(u, v).$$

In order to compare different topologies, we need to scale the $\overline{d}$ values to the interval $[0, 1]$ and define it as $\underline{d}$. For an evaluation of the adaptive navigation, two kinds of topologies were used. The first one uses static distances based on geographical information comparable to a situation with no information about congestion or traffic disruption, but with well known

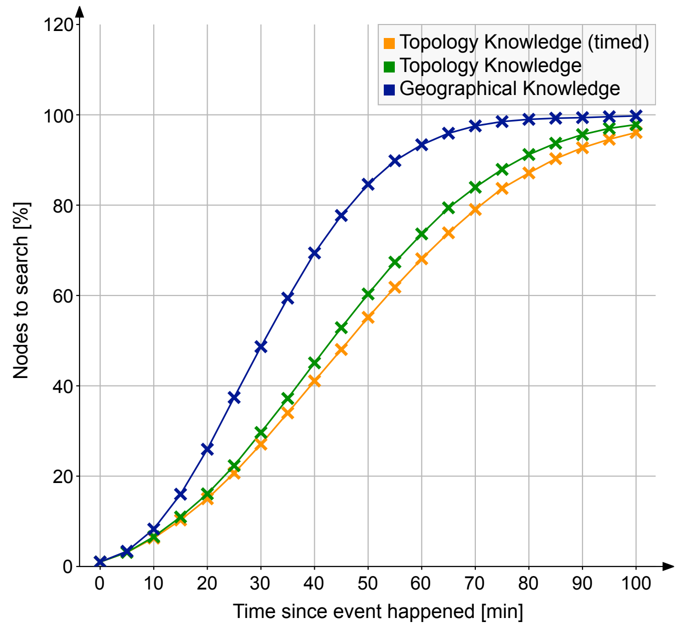

Fig. 5. Increase of the search space, depending on the elapsed time: Simulated pedestrians with a variable moving speed $v : N(\mu = 1.5\frac{m}{s}, \sigma = 0.3\frac{m}{s})$ on a synthetically generated set of topologies (10x10 km^2), mean over 32 runs.

neighborhood relationships (see "Topology Knowledge" measurements). The second one uses modified distances between two congested nodes based on an exponential distribution with $Exp(\lambda = 1.0)$, identical to the "Topology Knowledge (timed)" experiment that was introduced in the last section.

For the following simulation study, both topologies – the static ones and the dynamically congested – have been generated and the $\underline{d}$ values were calculated. Of course, the simulated results for $\underline{d}$ are graph dependent and a representative number of experiments had to be made in order to reach statistical certainty. One basic adjustable property of the synthetic topologies is the average node degree. Referring to the OSM Table I, typical cities have an average node degree in the interval $[2, 3]$. Notice the fact that crossroads usually consists of 4–5 roads the interval can be extended to $[2, 5]$ if streets are taken into account. We measured the $\underline{d}$ values for graphs with an average node degree starting from 2.2 to 5.8 with equidistant steps of 0.4 for the two different types of topologies. All measurements are calculated means over 32 runs with 99% confidence intervals on synthetically generated topologies with 500 nodes. Therefore it was not possible to use real cities' topologies, because the average node degree would be fixed and not adjustable for statistical evaluation runs.

Figure 6 shows the average navigation costs depending on the average node degree of the evaluated topologies. For higher node degrees, there are more "backup paths" with a growing number of disjoint paths in the topology. The adaptive navigation algorithm has the ability to discover and use those additional paths in the case of congestion (cf., Fig. 4). In a city with an average node degree of 2.6 (e.g., Berlin) the paths for geographical navigation have costs of about 0.4 in contrast to

≈ 0.3 with the adaptive version. For real world scenarios the measured values in the interval $[2, 3]$ are important. Generally, the adaptive shortest path algorithm outperformes the static one and saves approximately 20% of costs.

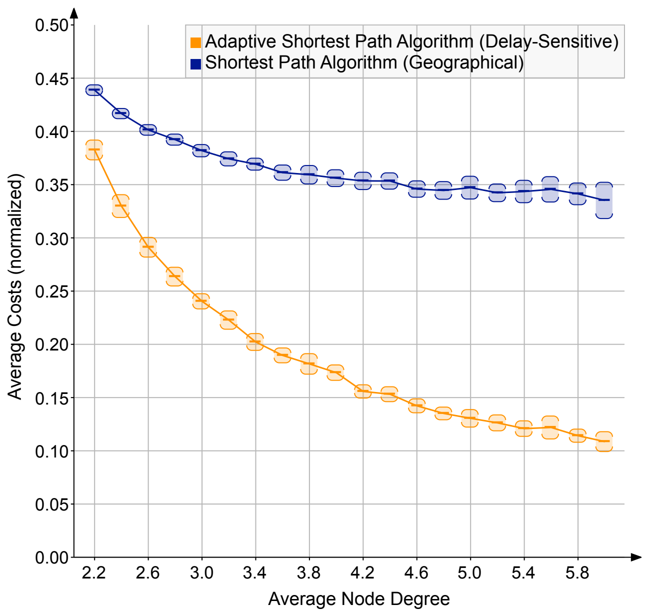

Fig. 6. Average costs (covered distance, [0,1] normalized) for all paths, depending on the navigation algorithm, depicted over the average node degree (99% Confidence).

Summarizing, we showed that navigation can benefit from collected topology knowledge, because in all samples the adaptive navigation could save approximately 10% – 60% of costs. The static navigation approach should only be used in cases where no congestion occurs.

V. Conclusion and Future Work

In this article, a new deployment scenario for distributed image processing was discussed. Two demonstrative applications are presented and motivated in the context of smart navigation and public surveillance. An analysis of the most important requirements reveals the gap of security and functionality on the one hand, as well as privacy, scalability, and robustness on the other hand. In order to meet all the stated objectives, distributed networked applications are a promising approach. Remembering the research on peer-to-peer networks, also the smart city scenario can benefit from distributed algorithms and the advances in scalability as well as robustness against network failures and sabotage. As a basic feature, distributed image processing can provide topological properties of public transportation networks in realtime. Completely pseudonymised, and without any intervention of administrative personnel, the spatio-temporal processes can be analyzed and evaluated using only localized image processing.

To evaluate the profit for synthesizing applications, a simulation study was conducted and discussed. Two different and demonstrative approaches have been presented that can be built upon distributed topology estimation algorithms, a basic challenge of research on image processing. In order to generate realistic simulation topologies, real cities and their public transportation networks have been extracted and surveyed, based on OSM data. Afterwards, similar synthetic topologies were used to serve as input for the demonstrative applications. On the one hand, only the geographical coordinates were used by the applications. On the other hand, traffic relationships were estimated by distributed image processing procedures, at first without timings, and subsequently, with estimated transition times between pairwise neighbored DSC nodes in the network. For both networked applications, it turns out that timed topological knowledge about the public transportation network can support the introduced and yet elementary approaches to a high degree. Smart navigation and route planning is a very complicated feature if it has to be done in realtime, regarding the actual environmental conditions. We claim that distributed networked applications have the potential to support these tasks in an efficient, scalable, and robust way. Also the mentioned algorithms utilized in the given applications can be computed in a distributed manner.

However, some issues remain to be addressed in further research. By using more sophisticated algorithms, the basic approaches can be extended, e.g., with user-generated information via smartphones, or the addition of timetable data and machine learning strategies. Also, the introduced mechanisms should be tested under real conditions, but this would require an extensive pilot project with the support of a city government and access to the core infrastructure for surveillance tasks – resources that were not available in this project.

References

[1] H. Chourabi, T. Nam, S. Walker, J. R. Gil-Garcia, S. Mellouli, K. Nahon, T. a. Pardo, H. J. Scholl, Understanding Smart Cities: An Integrative Framework, in: 2012 45th Hawaii International Conference on System Sciences, IEEE, 2012, pp. 2289–2297. doi:10.1109/HICSS.2012.615.

[2] K. Pentikousis, H. Wang, Network infrastructure at the crossroads the emergence of smart cities, in: 2011 15th International Conference on Intelligence in Next Generation Networks, IEEE, 2011, pp. 109–114.

[3] M. Naphade, G. Banavar, C. Harrison, J. Paraszczak, R. Morris, Smarter Cities and Their Innovation Challenges, Computer 44 (6) (2011) 32–39.

[4] http://www.nydailynews.com/new-york/nypd-unveils-new-40-million-super-computer-system-data-network-cameras-license-plate-readers-crime-reports-article-1.1132135.

[5] R. Farrell, D. Doermann, L. S. Davis, Learning Higher-order Transition Models in Medium-scale Camera Networks, in: 2007 IEEE 11th International Conference on Computer Vision, IEEE, 2007, pp. 1–8.

[6] R. Farrell, L. Davis, Decentralized discovery of camera network topology, in: Distributed Smart Cameras, 2008. ICDSC 2008. Second ACM/IEEE International Conference on, IEEE, 2008, pp. 1–10.

[7] R. Golembewski, G. Schafer, T. Gerlach, Efficient communication for large-scale robust image processing with smart camera devices, in: IEEE Symposium on Computational Intelligence for Security and Defence Applications, Ottawa, Canada, 2012, pp. 1–8.

[8] C. Roth, S. M. Kang, M. Batty, M. Barthelemy, A long-time limit for world subway networks., Journal of the Royal Society, Interface / the Royal Society 9 (75) (2012) 2540–50. doi:10.1098/rsif.2012.0259.

[9] V. Gupta, C. Langbort, R. Murray, On the robustness of distributed algorithms, in: Decision and Control, 2006 45th IEEE Conference on, IEEE, 2006, pp. 3473–3478.

[10] M. M. Haklay, P. Weber, OpenStreetMap: User-Generated Street Maps, IEEE Pervasive Computing 7 (4) (2008) 12–18.

[11] A. Varga, The OMNeT++ Discrete Event Simulation System, in: Proceedings of the European Simulation Multiconference (ESM2001), 2001, pp. 319–324.