A Descriptive Model Based on the Mining of Web Map Server Logs for Tile Prefetching in a Web Map Cache

Ricardo García, Juan Pablo de Castro, María Jesús Verdú, Elena Verdú, Luisa María Regueras and Pablo López

***Abstract*—Web mapping has become a popular way of distributing interactive digital maps over the Internet. Traditional web map services generated map images on the fly each time a request was received, which limited service scalability and offered a poor user experience. Most popular web map services, such as Google Maps or Microsoft Virtual Earth, have demonstrated that an optimal delivery of online mapping can be achieved by serving pre-generated map image tiles from a server-side cache. However, these caches can grow unmanageably in size, forcing administrators to use partial caches containing just a subset of the total tiles. By assuming that users access patterns are slow to change, service history can be used to determine in advance which areas are likely to be requested in the future, based exclusively on past accesses. Those tiles with high probability of being requested shortly can be pre-generated and cached on advance for faster retrieval. This work proposes the use of a descriptive model based on the mining of web map server logs for predicting popular areas in a web map, considered good candidates for tile prefetching. However, as the number of tiles grows exponentially with the rendering resolution level, it is rarely feasible to work with statistics of individual tiles. To overcome this issue, a simplified model is proposed which combines statistics from multiple tiles to reduce the dimension of the tiling space. This model has been tested using real-world logs from several nationwide public web map services in Spain. Simulations demonstrate that significant savings of storage requirements can be achieved by using a partial cache with the proposed model, while maintaining a high cache hit ratio.**

***Index Terms*—Web mapping, Map tile, WMTS, SDI, WMS, Descriptive model, Logs, Proxy cache.**

I. Introduction

THE Web Map Service (WMS) standard of the Open Geospatial Consortium (OGC) offers a standardized and flexible way of serving cartographic digital maps of spatially referenced data through HTTP requests [1]. However, spatial parameters in requests are not constrained, which forces images to be generated on the fly each time a request is received, limiting the scalability of these services. This process has been proved to be ineffective to satisfy the requirements of some massive applications, as explained in [2] after NASA's experience with the OnEarth map server.

A common approach to improve the cachability of requests consists of dividing the map into a discrete set of images, called tiles, and restrict user requests to that set. Several specifications have been developed to address how cacheable image tiles are advertised from server-side and how a client requests cached image tiles. The Open Source Geospatial Foundation (OSGeo) developed the WMS Tile Caching (usually known as WMS-C) proposal [4], while the OGC has recently released the Web Map Tile Service Standard (WMTS) [5] inspired by the former and other similar initiatives.

Most popular commercial services, like Google Maps, Yahoo Maps or Microsoft Virtual Earth, have already shown that significant performance improvements can be achieved by adopting this methodology, using their custom tiling schemes.

When an OGC service is used in a demanding environment, with stationary and bit configurable parameters, the proxy web cache pattern can be used to improve the perceived quality of service. A proxy is a device placed seamlessly anywhere between the client and the final service, intercepting user's requests [6].

Using this approach, providers have to cope with serious decisions about the design and maintenance of their tile caches. The immediate option is to pre-generate all tiles from all available scales. However, only big corporations have currently enough storage resources to store all the tiles. This is not a problem for them, but smaller companies must decide carefully which part of the content should be pre-generated.

Anyway, there are cartographic layers that should be updated frequently and every of them start from an empty cache. In general, to pre-generate all the objects is not a good approach when serving frequently updated maps, such as weather or traffic information maps.

The rest of the paper is organized as follows. Section II introduces a formal characterization of the tiling space. Section III defines a descriptive model to predict which tiles should be cached from logs of past server accesses. This model is analyzed and the experiment results are presented later. Finally, Section IV includes the main conclusions of this work.

II. Tiling space

In order to offer a tiled web map service, the web map server renders the map across a fixed set of scales through progressive generalization. Rendered map images are then divided into tiles, describing a tile pyramid as depicted in Fig.1.

At any moment a cache status can be defined where its managed objects can be available with a certain probability.

Spatial cache objects can be identified by their coordinates. Each tile is defined tile as $T(i,j,n)$, having $i,j,n \in \mathbb{N}$ (where n is the resolution level in the scale pyramid and i,j are the spatial indexes for this level). Therefore, $P_h\{T(i,j,n)\}$

The authors are with the Laboratory of Spatial Data Infraestructures at the University of Valladolid. Web Site: www.idelab.uva.es

is defined as the probability of getting a cache hit for the requested tile $T(i,j,n)$, at time t.

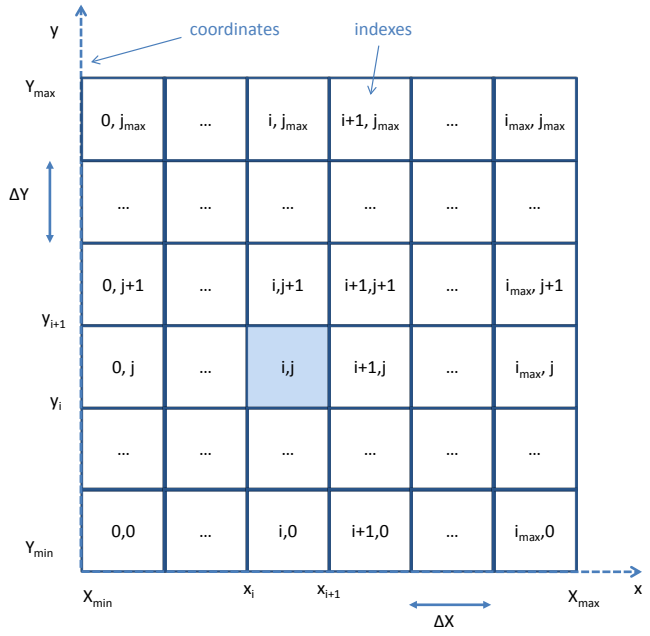


Fig. 1: Uniform tile space for a certain level of the scale pyramid

Denominate τ_h to the cost in seconds to delivery an object directly from the cache and τ_m to the cost in seconds to build a tile through the original services.

Being $f_{req}(x,y,n)$ the spatial probability density that characterizes the spatial distribution of the centroids of the requested map tiles for the scale n at time t. In general, for any request distribution, tiles are requested with probability (1).

$$P_{req}\{T(i,j,n)\} = \int_{y=y_{n,j}}^{y_{n,j+1}} \int_{x=x_{n,i}}^{x_{n,i+1}} f_{req}(x,y,n)\,dxdy \quad (1)$$

Although this result can be useful to analyze non tiled requests to a proxy-cache, it can be simplified assuming that requests are constrained to a reference grid cell (Fig.1). In this case, the probability of receiving a request of the tile $T(i,j,n)$ with size $\Delta x \Delta y$ is (2):

$$P_{req}\{T(i,j,n),t\} = f_{req}(x,y,n,t)\,\Delta x \Delta y \quad (2)$$

The latency to serve a request for a given tile with coordinates (i,j,n) at time t can be determined by (3):

$$\begin{aligned}\tau(i,j,n,t) &= P_h\{T(i,j,n)\}(t)\tau_h + \\ &+ (1 - P_h\{T(i,j,n)\}(t)\tau_m)\end{aligned} \quad (3)$$

Combining (3) and (1) a probabilistic expression can be obtained to estimate the average latency of the service:

$$\begin{aligned}\tau(t) &= \sum_{\langle ijn\rangle} (\tau_m - P_h\{T(i,j,n)\}(t)(\tau_m - \tau_h)) \\ P_{req} &\{T(i,j,n)\}\end{aligned} \quad (4)$$

Some components can be identified in (4), which should be parametrized at least locally.

Some important characteristics of this model are:

- There is no independence between $P_h\{T(i,j,n)\}(t)$ and $P_{req}\{T(i,j,n),t\}$ because the cache status is intimately connected to the service requests history.
- There is no temporal invariance in the transient state.
- The probability density function is not uniform and it probably represents a direct relationship with the spatial structure of the underlying information.

Another factor that must be addressed is the problem of managing exponential data structures. In a cache with pyramidal scales, the number of objects increases exponentially according to the n value. Therefore, the use of analytic or predictive algorithms can be impractical, even with the support of heuristic algorithms, if they aim to be used with all the pyramid levels.

For this reason, it is very useful to obtain a statistical relationship model between different scale levels. The statistics of a level can be extrapolated to other near levels. Assuming that the geographic location is a relevant information for all the scale levels, and that requests have significant spatial correlation, heuristic algorithms like Locality Principle [7] can be used. These algorithms should manage the whole cache through statistical probes within levels containing a manageable number of objects (tiles).

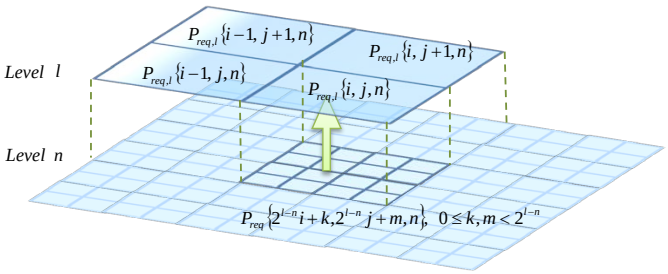


Fig. 2: Probability estimation of requests to tile $T(i,j,n)$ calculated from lower levels

III. Descriptive Model

Descriptive models determine the most requested map areas from map servers logs. For example, the web application *Microsoft HotMap*[1] represents in a heatmap the requests to Bing Maps service [8], [9], [10]. However, it is not possible to access data itself, which limits the possible analysis to a visual exploration and makes the use of automatic algorithms to extract patterns of interest difficult.

A. Analyzed map services

This study has been carried out with data retrieved from requests made to the WMS-C services *Cartociudad*[2], *IDEE-BASE* and *PNOA*, provided by the National Geographic Institute (IGN)[3] of Spain. A total of 3.778.369, 16.978.535 and

[1] http://hotmap.msresearch.us/
[2] http://www.cartociudad.es
[3] http://www.ign.es

9.816.747 requests have been processed for these services, respectively.

Tiled versions of these services use caches implemented by Metacarta *Tilecache* [11]. This cache system follows the OSGeo WMS-C specification.

B. Retrieving request data

Tile map requests are extracted from the Apache's standard access log configured using the Common Log Format [12]. Information retrieved from these records is as follows:

- Request date, with precision in seconds.
- IP address or hostname of the remote client that made the request to the server.
- Server status code returned to the the client. This information is very valuable, because it reveals whether the request was successfully returned or not.
- Size of the returned object. This value does not include the response headers, and it is expressed in bytes.

The following WMS-C request parameters are extracted: service, version, request, layers, width, height, format, styles, exceptions and coordinate reference system.

Fig.3 plots the number of requests made versus the resolution level for the analyzed services. *IDEE-BASE* and Cartociudad present an anomalous peak in level 4, which can be justified because it is the more suitable level for displaying the whole Spain area in a single map screen.

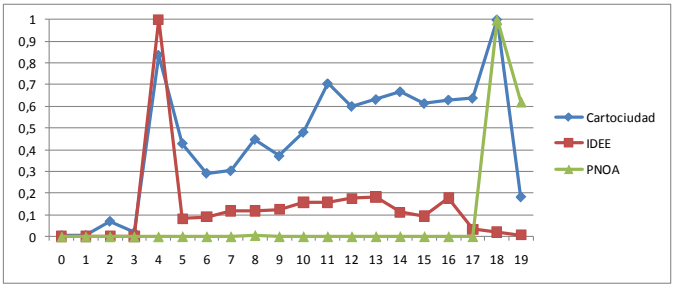


Fig. 3: Normalized request distribution along the different scales

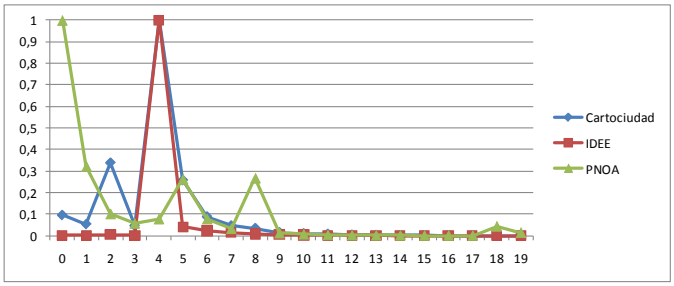


Fig. 4: Normalized request density distribution along the different scales

Although the higher resolution levels receive a greater number of requests, request density (requests per area unit) is larger in the lower ones (Fig.4). Because of this, a common approach consists of pre-generating tiles from the lower levels. High resolution tiles can be included in the cache later, or even let them be included in the cache as they are requested.

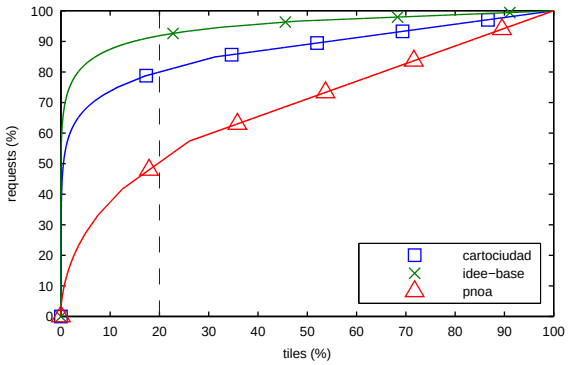


Fig. 5: Percentile of requests for the analyzed services.

It must be noted that the performance gain achieved by the use of a tile cache will vary depending on how the tile requests are distributed over the tiling space. If those were uniformly distributed, the cache gain would be proportional to the cache size. However, it has been found that tile requests describe a Pareto distribution, as shown in Figure 5. Tile requests to the CartoCiudad map service follow the 20:80 rule, which means that the 20% of tiles receive the 80% of the total number of requests. In the case of IDEE-Base, this behaviour is even more prominent, where the 10% of tiles receive almost a 90% of total requests. PNOA requests are more scattered. This happens because about the 90% of requests belong to the two higher resolution levels (19 and 20), the ones with larger number of tiles.

Services that show Pareto distributions are well-suited for caching, because high cache hit ratios can be found by caching a reduced fraction of the total tiles.

Main part of the analyzed requests to these services are referred to the Spain . So, the studied area has been reduced to the bounding box [-11.9971, 32.8711, 5.5371, 46.0107], which represents Spain in a grid of 400 tiles width and 300 tiles height in level 12.

C. Simplified model

Given the exponential nature of the scale pyramid, and the impossibility of working with statistics from individual tiles, a simplified model has been proposed. This model tries to approximate the probability of receiving a tile request in a particular level from the statistics retrieved from another level covering the same area. Specifically the model has been simplified to the 400x300 tiles area defined above. The pyramidal structure of scales is transformed in some way in a prism-like structure with the same number of items in all the scales (see Fig.6).

Given the boolean nature of caching (a tile can be placed in the cache or not), it is necessary to define a probability threshold that indicates whether a tile must be cached or not. This value determines the number of cached tiles and the hit rate. If the threshold value is 0, the hit rate will be 100%, at the expense of storing all the tiles. However, if the threshold

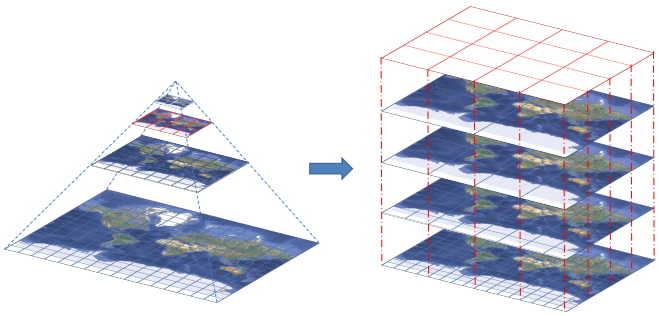


Fig. 6: Conceptual translation from a scale pyramid to prism through the simplified model

value is 1, any tile will be stored and the hit rate will be 0. The value for this threshold must be chosen so it maximizes the cache hit rate while keeping the resource requirements under a given level.

D. Experiment and results

In order to experiment with the proposed model, request logs were divided in two time ranges. The first one was used as source to make predictions and the second one was used to prove the prediction created previously. The experiment was conducted with the simplified model to the grid cell defined by the level of resolution 12.

Fig.7 shows the heatmaps of requests extracted from the web server logs of IDEE-Base service, propagated to level 12 through the proposed model. These figures demonstrate that some entities such as coast lines, cities and major roads are highly requested. These elements could be used as entities for a predictive model to identify priority objects, as explained in [13].

These figures show that near levels are more related than distant levels, but all of them share certain similarity. This relationships between resolution levels encourages the use of statistics collected in a level to predict the map usage patterns in another level with detailer resolution. For example, as shown in Fig.7c and Fig.7e, resolution levels 14 and 16 are very correlated. It is easier to work with the statistics of level 14 than with those of level 16 which has much more elements.

Tables I, II and III represent the hit percentage for each service. These tables show the percentage of hits obtained for the level identified by the column index from the statistics collected in the level identified by the row index. Last column shows the resources consumption, as a percentage of cached tiles. Last row collects the results of combining the statistics of all levels to make predictions over every level. Shadowed cell in Table I indicates that using retrieved statistics of level 13 as the prediction source, a hit rate of 91.95% is obtained for predictions made in the level 18, being necessary the storage of a 8.83% of the tiles in cache.

Nevertheless, it must be noted that the main benefit of using a partial cache is not the reduction in the number of cached tiles. The main benefits are the savings in storage space and generation time. As explained in [13], the amount of saved tiles is bigger than the storage saving. It reveals that the most interesting tiles come at a bigger cost. Mainly, popular areas

	12	13	14	15	16	17	18	19	resources
12	87.7	90.4	91.5	90.2	94.4	95.4	96.0	92.3	18.08
13	69.6	78.8	82.3	83.0	88.6	80.8	91.9	85.0	8.83
14	56.6	69.8	76.4	78.8	84.6	78.8	89.0	80.2	5.40
15	41.4	54.9	65.8	74.5	78.4	74.5	85.5	73.3	3.00
16	33.2	45.6	57.6	66.3	73.8	71.2	84.1	69.0	2.04
17	29.3	42.4	53.3	63.5	54.8	78.2	81.5	66.4	1.61
18	26.3	37.7	50.4	61.5	52.8	67.8	81.7	63.4	1.35
19	15.2	23.4	33.6	44.7	37.9	48.6	72.7	44.5	0.76
prop	67.7	76.3	80.6	84.4	87.4	90.8	92.7	83.8	6.68

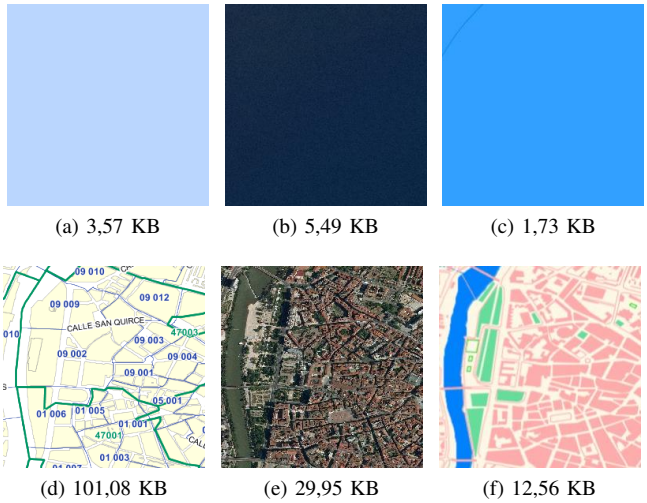
TABLE I: Percentage (%) of cache hits through the simplified model obtained from Cartociudad logs, using the mean of the normalized frequencies as the probability threshold

	12	13	14	15	16	17	18	19	resources
12	88.0	85.7	93.7	94.5	96.0	93.7	93.5	79.3	21.02
13	74.8	85.9	90.7	92.1	94.0	89.7	82.4	54.8	15.28
14	48.1	57.7	84.3	86.6	89.1	78.5	59.8	38.7	7.63
15	34.6	45.6	76.0	82.7	81.7	66.9	50.6	31.5	4.58
16	41.6	53.1	80.9	84.0	89.8	74.7	56.2	37.5	6.10
17	30.2	37.6	57.0	60.1	62.2	69.5	55.5	23.4	3.26
18	23.5	25.5	41.7	46.1	45.8	41.4	61.9	33.3	2.33
19	8.8	8.6	12.4	13.1	14.3	12.2	13.6	44.1	1.23
prop	67.1	78.0	87.7	90.1	92.3	86.8	83.9	71.6	12.50

TABLE II: Percentage (%) of cache hits through the simplified model obtained from *IDEE-BASE* logs, using the mean of the normalized frequencies as the probability threshold

	12	13	14	15	16	17	18	19	resources
12	21.1	25.1	27.4	18.8	20.3	32.0	11.3	12.7	0.64
13	11.2	22.8	21.8	17.1	30.6	29.3	11.5	13.1	0.25
14	6.6	15.9	21.4	17.8	28.1	23.2	10.5	12.2	0.17
15	3.4	9.9	15.6	17.8	29.4	18.8	8.4	9.7	0.11
16	2.4	6.5	11.1	12.4	28.0	13.4	6.1	7.1	0.06
17	0.9	4.5	7.5	9.6	12.6	5.9	4.5	5.2	0.03
18	45.2	64.9	73.6	75.5	86.0	89.3	86.4	88.6	9.06
19	39.4	61.7	69.8	67.1	72.8	69.0	82.5	85.2	7.71
prop	40.0	55.9	64.0	61.2	72.2	64.5	76.7	80.2	8.93

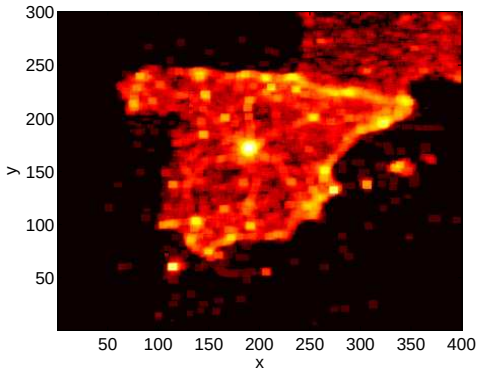
TABLE III: Percentage (%) of cache hits through the simplified model obtained from PNOA logs, using the mean of the normalized frequencies as the probability threshold



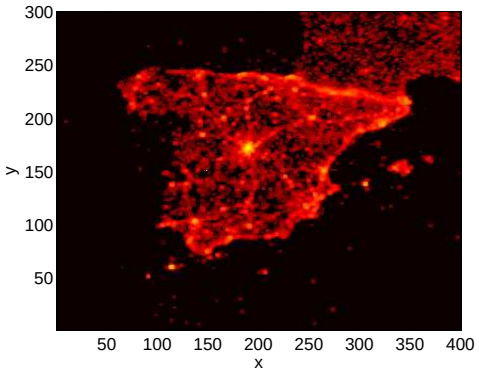
(a) 3,57 KB (b) 5,49 KB (c) 1,73 KB

(d) 101,08 KB (e) 29,95 KB (f) 12,56 KB

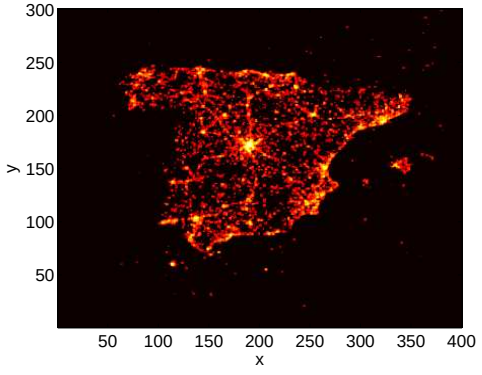
Fig. 9: Storage size for a tile corresponding to a sea region (top) and a city centre (bottom) for Cartociudad (left), PNOA (center) and IDEE-Base (right).



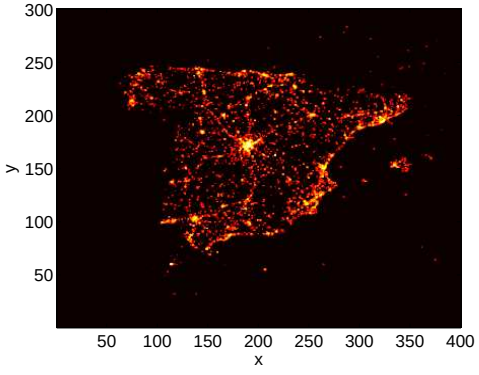
(a) Scale 12



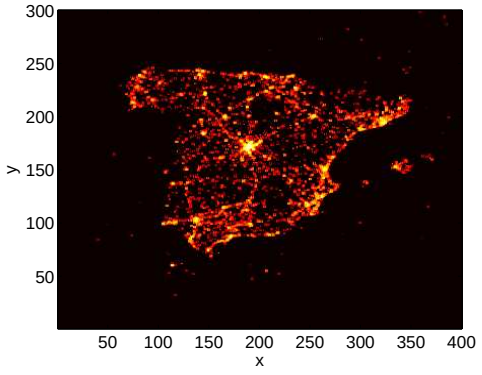
(b) Scale 13



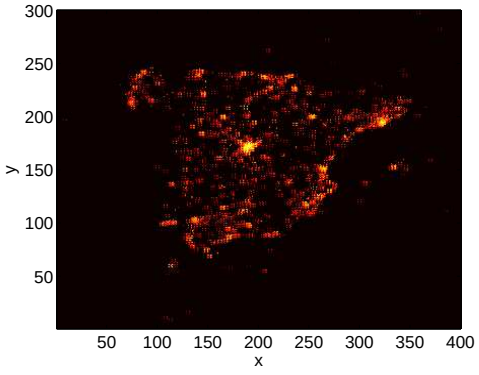
(c) Scale 14



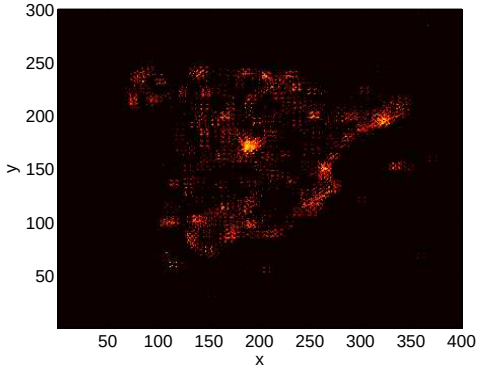
(d) Scale 15



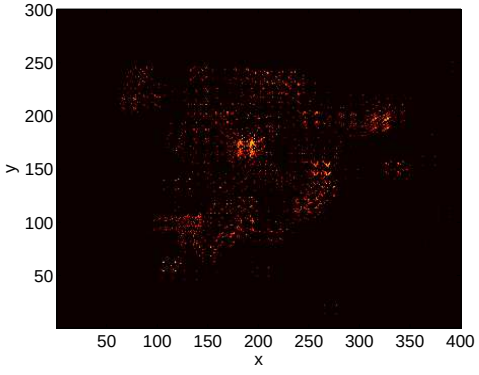
(e) Scale 16



(f) Scale 17

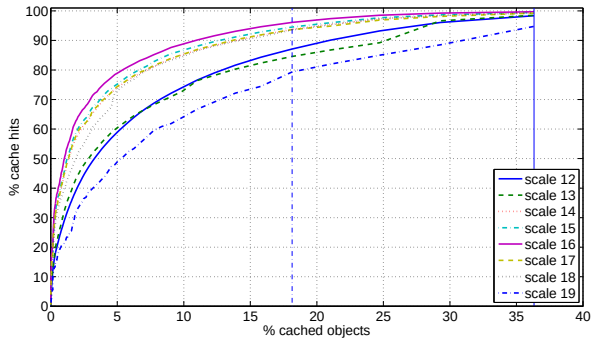


(g) Scale 18

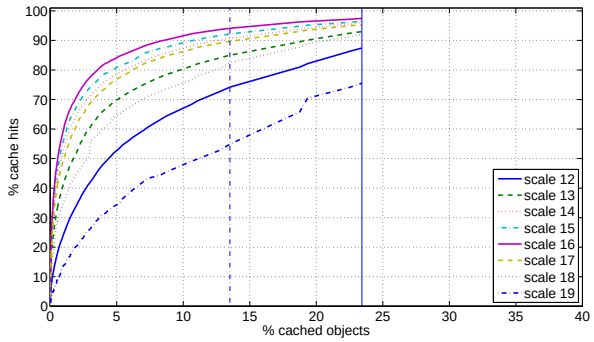


(h) Scale 19

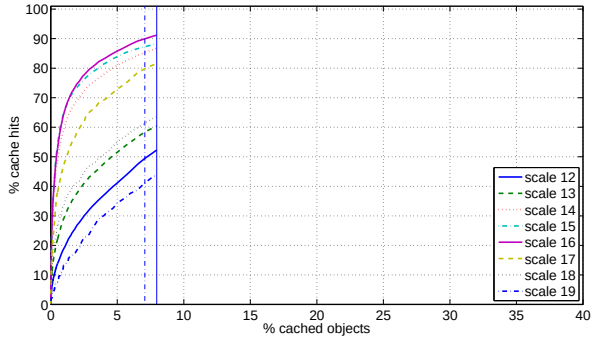
Fig. 7: Heatmap of the requests to the *IDEE-BASE* service propagated from levels 12-19 to level 12.



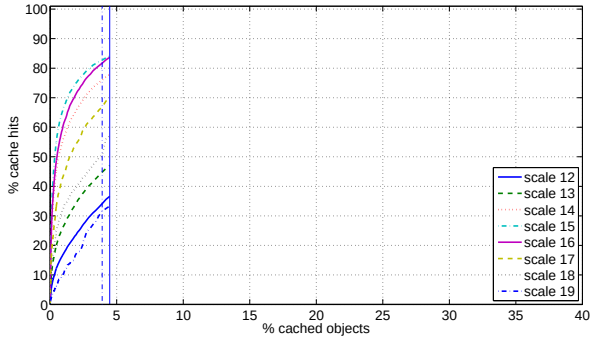
(a) Scale 12



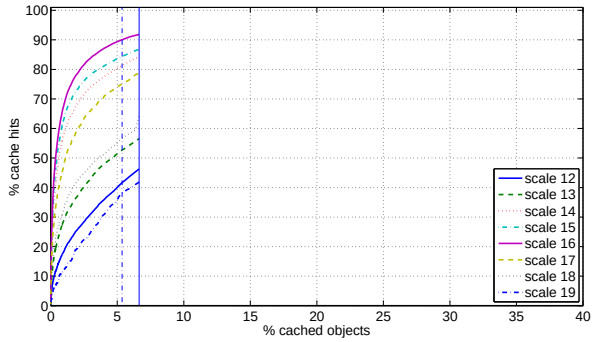
(b) Scale 13



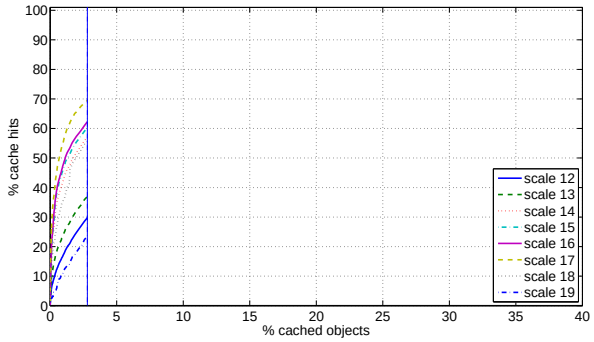
(c) Scale 14



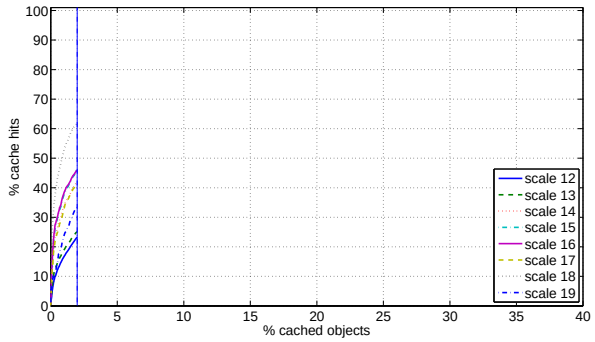
(d) Scale 15



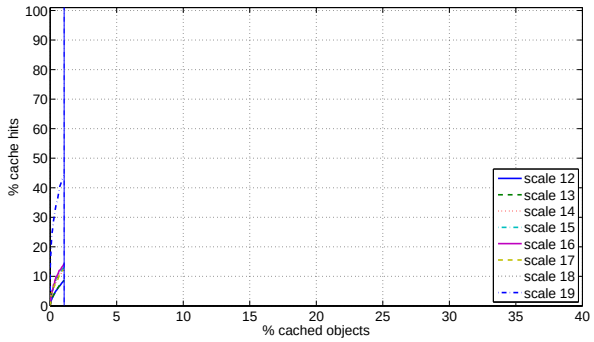
(e) Scale 16



(f) Scale 17



(g) Scale 18



(h) Scale 19

Fig. 8: Percentage of hits vs cached objects for *IDEE-BASE* service through the simplified model.

are more complex, and it is necessary more disk space to store them. Fig. 9 shows the storage size of a popular tile corresponding to a city centre versus an unpopular tile that renders a sea region.

Low hit rates for the *PNOA* service are caused by the strange request distribution for this service. The simplified model is not able to make precise predictions for this kind of services.

Fig.8 represents the cache hit ratios obtained by the simplified model for the *IDEE-BASE* service. From a certain percentage of cached objects, identified by the continuous vertical line, the simplified model is not able to make predictions. Tiles situated at the right of this line correspond to objects that have never been requested so are not collected in server logs.

The simplified model obtains better results for predicting user behavior from near resolution levels. Descending in the scale pyramid, the requested objects percentage decreases, so the model prediction range decreases too.

The proposed model obtains high cache hit ratios using a reduced subset of the total tiles.

IV. Conclusion

Web map tiled services have reached high popularity in recent times, improving response times and scalability versus traditional mapping services, by serving pre-generated images from cache. In environments with reduced storage capabilities or where the cartography is updated frequently, it is not suitable to pre-generate the whole cartographic content. In this cases, it is necessary to work with incomplete caches. The solution proposed in this paper is based on the definition of priority areas for pre-fetching and replacement mechanisms, maximizing the user QoS while keeping resource consumption under a given level. It tries to keep in cache the tiles which are likely to be requested in the future. To determine those priority areas, a descriptive model has been proposed. This model has been tested with different Spanish national map services logs. High hit rates obtained prove that it is possible to predict the future accesses to a Web map service based solely on the information collected from past. This model can be applied when user behavior is relatively stationary. The multi-scale analysis made supports the use of statistics collected in a certain level to predict the behavior at other near levels.

In the future, higher cache hit ratios could be achieved by combining the weighted information collected from different scale levels, instead the single level analysis in the current model.

From the usage logs of analyzed map services some conclusions have been obtained. Areas like coast lines, cities or major roads are more requested than others. These elements could be used as entities for a predictive model to identify priority objects that should be take into account during the cache maintenance tasks.

Acknowledgment

This work has been partially supported by the Spanish Ministry of Science and Innovation through the project “España Virtual” (ref. CENIT 2008-1030), National Centre for Geographic Information (CNIG) and the National Geographic Institute of Spain (IGN).

References

[1] J. de la Beaujardiere, editor. *OpenGIS Web Map Server Implementation Specification*. Open Geospatial Consortium Inc, OGC 06-042, 2006.

[2] Lucian Plesea. The design, implementation and operation of the JPL On-Earth WMS server. In J.T. Sample, K. Shaw, S. Tu, and M. Abdelguerfi, editors, *Geospatial Services and Applications for the Internet*, pages 93–109. Springer, Berlin, 2008.

[3] Joe Schwartz. Bing maps tile system. http://msdn.microsoft.com/en-us/library/bb259689.aspx, 2009.

[4] OSGeo. WMS tiling client recommendation - OSGeo wiki. http://wiki.osgeo.org/wiki/WMS_Tiling_Client _Recommendation, 2008.

[5] Núria Juliá Juan Masó, Keith Pomakis, editor. *OpenGIS Web Map Tile Service Implementation Standard*. Open Geospatial Consortium Inc, OGC 07-057r7, 2010.

[6] Kai Cheng, Y. Kambayashi, and M. Mohania. Efficient management of data in proxy cache. In *Database and Expert Systems Applications, 2001. Proceedings. 12th International Workshop on*, pages 479–483, 2001.

[7] Peter J. Denning. The locality principle. *Commun. ACM*, 48(7):19–24, 2005.

[8] D. Fisher. Hotmap: Looking at geographic attention. *Visualization and Computer Graphics, IEEE Transactions on*, 13:(6) 1184–1191, 2007.

[9] D. Fisher. The impact of hotmap. 2009.

[10] D. Fisher. How we watch the city: Popularity and online maps. In *Workshop on Imaging the City, ACM CHI 2007 Conference*, 2007.

[11] MetaCarta. TileCache, from MetaCarta labs. http://tilecache.org/, 2008.

[12] Archivos de registro (Log files) - servidor HTTP apache. http://httpd.apache.org/docs/2.0/es/logs.html.

[13] S. Quinn and M. Gahegan. A predictive model for frequently viewed tiles in a web map. *Transactions in GIS*, 14(2):193–216, 2010.

Ricardo García received the Master's degree in telecommunications engineering from the University of Valladolid, Valladolid, Spain, in 2008.

He is currently working toward the Ph.D degree on artificial intelligence methods applied to cache mechanisms for improving the performance of Web Map Services in the Spatial Data Infrastructures.

Juan P. de Castro (M'95) received the Master's degree in telecommunications engineering from the University of Valladolid, Valladolid, Spain, in 1996, and the Ph.D. degree in Telecommunications Engineering from the Technical University of Madrid (UPM), Madrid, Spain, in 2000.

He is currently an Associate Professor with the Department of Signal Theory, Communications and Telematics Engineering, University of Valladolid. He was the Research Director of a technological centre from February 2001 to June 2003. He currently acts as R&D consultant for various enterprises.

María Jesús Verdú (M'95) received the Master's and Ph.D. degrees in telecommunications engineering from the University of Valladolid, Valladolid, Spain, in 1996 and 1999, respectively.

She is currently an Associate Professor with the Department of Signal Theory, Communications and Telematics Engineering, University of Valladolid. She has experience in coordinating projects in the fields of new telematic applications for the Information Society and telecommunication networks, especially related to e-learning. Her research interests include new e-learning technologies.

Elena Verdú (M'09-SM'10) received the Master's and Ph.D. degrees in telecommunications engineering from the University of Valladolid, Valladolid, Spain, in 1999 and 2010, respectively.

She is currently the Coordinator of Research and a Project Manager with the Centre for the Development of Telecommunications of Castilla y León (CEDETEL). She is also an Adjunct Professor with the Department of Signal Theory, Communications and Telematics Engineering, University of Valladolid. Her research interests include new e-learning technologies.

Luisa M. Regueras (M'10) received the Master's and Ph.D. degrees in telecommunications engineering from the University of Valladolid, Valladolid, Spain, in 1998 and 2003, respectively.

She is currently an Assistant Professor with the Department of Signal Theory, Communications and Telematics Engineering at the University of Valladolid. Her research interests mainly include new e-learning technologies.

Pablo López received the Master's degree in Communications from the University of Valladolid, Valladolid, Spain, in 2009. He obtained the Master in Research on Information & Communications Technologies from the same University in 2010.

His research interests focuses on neogeography and spatial map caches.