1-2004

VISCORS: A visual-content recommender for the mobile Web

Chan Young KIM

Jae Kyu LEE
Singapore Management University, jklee@smu.edu.sg

Yoon ho CHO

Deok Hwan KIM

DOI: https://doi.org/10.1109/MIS.2004.75

Citation

KIM, Chan Young; LEE, Jae Kyu; CHO, Yoon ho; and KIM, Deok Hwan. VISCORS: A visual-content recommender for the mobile Web. (2004). *IEEE Intelligent Systems*. 19, (6), 32-39. Research Collection School Of Information Systems.
Available at: https://ink.library.smu.edu.sg/sis_research/1149

VISCORS: A Visual-Content Recommender for the Mobile Web

Chan Young Kim and Jae Kyu Lee, *Korea Advanced Institute of Science and Technology*

Yoon Ho Cho, *Kookmin University*

Deok Hwan Kim, *Dongyang Technical College*

Current search methods for mobile-Web content can be frustrating to use. To shorten searches for cell phone wallpaper images, VISCORS combines collaborative filtering with content-based image retrieval.

An increasing selection of content is becoming available in the mobile-Web environment, where users navigate the Web using wireless devices such as cell phones and PDAs. The fast growth and excellent prospects of the mobile-Web content market have attracted many content providers. However, as continuing deregulation further lowers the entry barriers for providers, increased competition is quickly eliminating profit opportunities. To survive in this environment, providers must offer an intelligent system that provides customers with a more pleasant mobile-shopping experience.

A particularly popular form of mobile-Web content is wallpaper images for cell phones.[1] Although the market for this content is growing rapidly as related technologies evolve, customers experience much frustration when searching for the images they want, owing to inefficient sequential search (see Figure 1). When a customer logs on to an image-download site using a cell phone, the site presents the customer with a list of the best-selling or newest images. The customer pages through the list and selects an image to inspect. If the customer likes the image, he or she might buy it. Otherwise, the customer repeats these steps until he or she stumbles over the right image or gives up. With this method, the expected number of images the customer views before hitting the desired image far exceeds the acceptable level.

These difficulties are partly attributable to the cell phone's characteristics. Compared to PCs, cell phones have smaller screens, fewer input keys, and less sophisticated browsers. So, the user interface of mobile-Web applications isn't as friendly as that of typical Web applications. Consequently, many customers use their PCs to select images and then request a download to their mobile devices. Nevertheless, searching generally remains inconvenient and complex. To make searching more acceptable, a more efficient search aid that suggests only the images meeting the customer's preference is necessary.

As a solution to these problems, we propose VISCORS (*Vis*ual *Co*ntents *R*ecommender *S*ystem). To reduce customers' search effort, VISCORS combines the two most popular information-filtering techniques: *collaborative filtering* and *content-based image retrieval*. Combined, these techniques properly handle the distinct characteristics of visual content while taking into account the mobile Web's constraints.

Collaborative filtering

Recommender systems help customers find the items they'd like to purchase. The most successful recommendation technique is collaborative filtering.[2,3] CF identifies customers (*neighbors*) whose tastes are like those of the target customer and recommends items those customers have liked.

However, CF has two major shortcomings. First, when there's a shortage of customer ratings, it suffers from a *sparsity problem.*[3–6] Most similarity measures used in CF work properly only when there's a sufficient number of ratings on common items from similar customers. An increase in the number of customers and items worsens this problem because the likelihood of different customers rating common items decreases. Such sparsity in ratings makes the formation of a neighborhood (a group of customers with similar tastes) inaccurate, thereby resulting in poor recommendations. Second, CF suffers from a *new-item problem.*[4,6] Because CF recommends an item on

1541-1672/04/$20.00 © 2004 IEEE
Published by the IEEE Computer Society

the basis of previous customers' ratings of that item, it doesn't recommend a newly introduced item until sufficient ratings of that item are available.

To address these shortcomings, researchers have proposed many variations of hybrid approaches that combine CF with *content-based filtering*.[4–6] Content-based filtering recommends items with properties similar to those of items the target customer liked in the past. Despite these approaches' success in some applications, none of them is adequate for wallpaper image recommendation because of visual content's distinct characteristics.

A customer's preference of images is ambiguous and more changeable over time than that of the usual items, because the same customer might perceive the same image differently at different times.[7] So, CF that recommends items entirely on the basis of the customer's past preferences yields lower-quality recommendations for images than it does for ordinary items. Any hybrid approach without a countermeasure to this drawback can't give acceptable results for image recommendation. A mechanism for learning about the customer's current preference is essential to deliver good recommendations.

Figure 1. A typical user interface for downloading wallpaper images for cell phones.

Content-based image retrieval

CBIR, the most common image retrieval technique, uses images' visual features to retrieve images similar to the given query.[7–9] However, its effectiveness is limited because of the gap between high-level concepts of images and their representation in low-level features. For CBIR to handle this semantic gap, it needs the ability to learn about the customer's true intention through iterative interactions. The customer's preference regarding presented images needs to be fed back so that CBIR learns from this preference to retrieve, in the next iteration, images more similar to the one the customer really wants. This learning process, *preference feedback*, is essential for faster search.

To express an image's degree of preference, you can use binary weights (for example, *preferred* or *unpreferred*) or multilevel weights (for example, *highly preferred*, *preferred*, *neutral*, *unpreferred*, or *highly unpreferred*).[7] We call a set of preferred images a *preferred set*. CBIR can use the preferred set's images to refine the query to learn the customer's current preference.

Because CBIR operates generally in the PC-based Web environment, applications that use CBIR typically assume that the user interface can display many images at once, eliciting multiple levels of preference feedback. However, these assumptions don't apply to mobile-Web devices. So, CBIR that takes into account mobile-Web devices' user interface constraints is necessary.

Viscors overview

Viscors consists of a CF module and a CBIR module (see Figure 2). The CF module produces the initial list of recommended

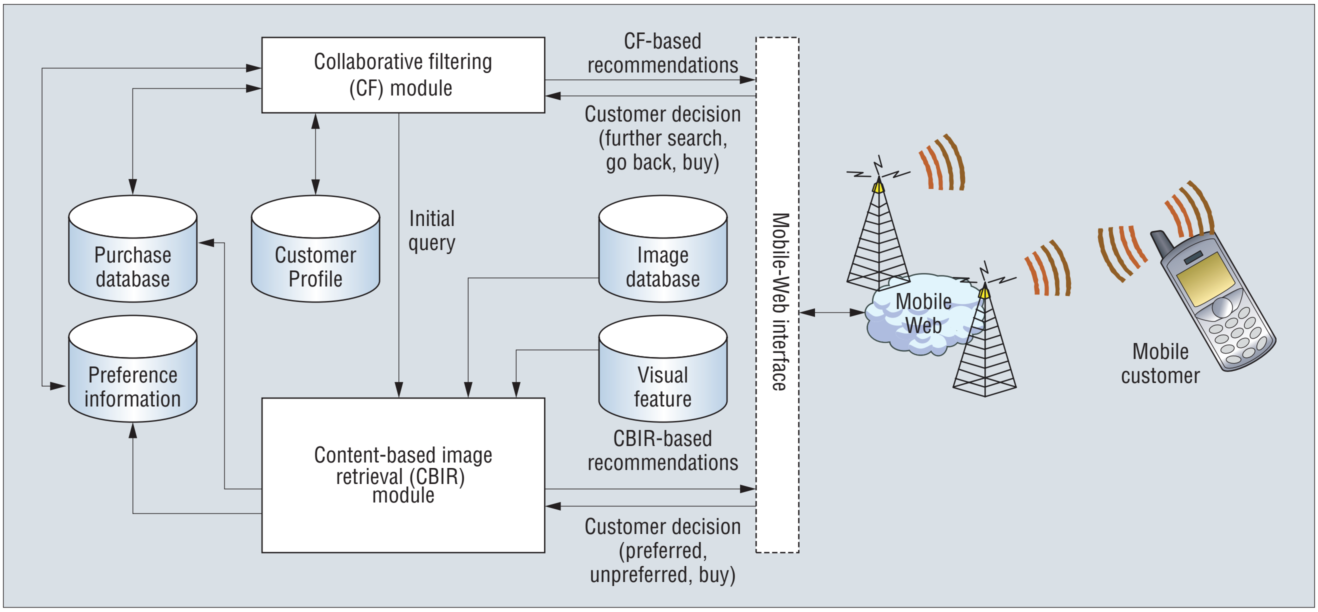


Figure 2. Viscors (Visual Contents Recommender System) architecture.

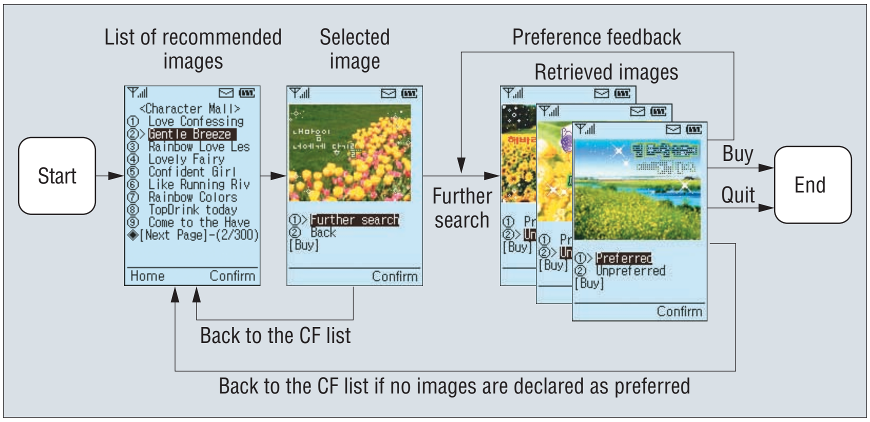


Figure 3. Viscors iterative search process.

images (the left screen shot in Figure 3). For new customers, Viscors generates the list using the best-seller-based method; otherwise, it uses CF. The customer skims through the list to see if any images are of interest. Then, the customer selects an entry to view the image (see the center screen shot in Figure 3). After viewing the image, the customer might decide to purchase it, use it as a query for a CBIR-based search of similar images, or go back to the CF-generated recommendation list.

If the customer decides to use the viewed image as a query, the CF module passes that image to the CBIR module. For all images in the database, the CBIR module calculates their distances from the query and generates a list of the most similar images. This module retrieves k images as recommendations, presents them to the customer one by one, and interactively elicits the user's preferences. (The right screenshots in Figure 3 illustrate the case of $k = 3$, where the system is requesting customer feedback on three recommended images.) For each of the k images, the customer must declare whether he or she prefers it (that is, the CBIR module uses binary preference information). At any point in this presentation session, the customer can also decide to buy an image or quit.

After the customer makes all k declarations, the CBIR module updates the preference information and purchase databases with the related information, for the CF module to use later. If the customer declares all k images as unpreferred, the search session returns to the CF-generated recommendation list. Otherwise, the CBIR module learns the customer's current preference using the preferred set and applies this information to refine the query and update the distance function. It then uses the refined query and updated distance function in the next iteration of retrieval. These iterations continue until the customer finds the desired image or quits the application.

Because Viscors adds the fed-back preference and purchase information to the customer profile as ratings, it alleviates the sparsity problem. Because the CBIR module recommends images on the basis of their visual features, Viscors can recommend images with no customer ratings, thus eliminating the new-item problem. In this way, our hybrid approach leads to higher-quality recommendations.

CF-based recommendation

Our customer profile is the matrix of preference ratings $\mathbf{P} = (p_{ij})$:

$$p_{ij} = \begin{cases} +2 & \text{if customer } i \text{ has purchased image } j \\ +1 & \text{if customer } i \text{ has marked image } j \text{ as preferred} \\ -1 & \text{if customer } i \text{ has marked image } j \text{ as unpreferred} \\ 0 & \text{if customer } i \text{ has not seen image } j \end{cases} \quad (1)$$

where $i = 1$ to M, $j = 1$ to N, M is the total number of customers, and N is the total number of images. As Equation 1 shows, the rating matrix's cells have four possible values. We rate the previously purchased images highest because they should reflect the customer's taste the most strongly. The preferred and unpreferred images have ratings that are one half or a negative one half, respectively, of the magnitude of the purchased ones. (Viscors records the initial CF-recommended image as preferred or unpreferred depending on whether the customer selects it as an initial query for CBIR.) We assign –1 to unpreferred images so that the CBIR module will recommend unseen images over unpreferred ones.

The CBIR module constantly replaces ratings in the customer profile with newly obtained purchase and preference information to reflect dynamically the customer's most recent preference. This is significantly different from the customer profiles that traditional CF techniques use.[2,3]

Given the customer profile **P**, the CF-based recommendation procedure for a target customer c takes two steps.

Step 1: Customer neighborhood formation

We use $sim(a, b)$ to denote the similarity between two customers a and b. First, we determine the neighborhood $H = \{h_1, h_2, \ldots, h_m\}$ such that $c \notin H$ and $sim(c, h_1)$ is the highest similarity, $sim(c, h_2)$ is the next highest, and so on. We calculate the similarity using the *Pearson-r* correlation[2]:

$$sim(a,b) = corr_{ab} = \frac{\sum_{j=1}^{N}\left(p_{aj} - \bar{p}_a\right)\left(p_{bj} - \bar{p}_b\right)}{\sqrt{\sum_{j=1}^{N}\left(p_{aj} - \bar{p}_a\right)^2 \sum_{j=1}^{N}\left(p_{bj} - \bar{p}_b\right)^2}} \quad (2)$$

where N is a total number of images, p_{aj} and p_{bj} are customer a's and b's ratings on image j, and $\bar{p}_a$ and $\bar{p}_b$ are customer a's and b's average ratings on all images.

Step 2: Recommendation generation

$PLS(c, j)$ denotes the *purchase likeliness score* of the target customer c for image j. We generate a list of n images, $R = \{r_1, r_2, \ldots, r_n\}$, such that $r_j \notin$ {the images that c has already purchased} and $PLS(c, r_1)$ is the highest PLS, $PLS(c, r_2)$ is the next highest, and so on. We compute the PLS as

$$PLS(c,j) = \frac{\sum_{i \in H}\left(p_{ij} - \bar{p}_i\right) \times sim(c,i)}{\sum_{i \in H} sim(c,i)} \quad (3)$$

CBIR-based recommendation

During CBIR, Viscors continuously refines the query to reflect the customer's latest preference. Viscors uses a multipoint query, because recent studies have indicated that such a query handles the semantic gap better than a single-point query.[8,9]

There are two different approaches in refining a query using multiple query points. The first approach clusters example images and uses a centroid of each cluster as a query point.[8,9] This approach works fine when enough example images are available. However, it's difficult to use with a very small number of example images.

The other approach uses every example image in a query as a query point.[10] This approach offers a breakthrough in cases with few example images. In the mobile-Web environment, having a preferred set of sufficient size is difficult because the small screens make it impossible to obtain preference information on multiple images in one interaction. So, VISCORS refines a query by replacing its query points with the newly fedback preferred images.

Because a query in VISCORS can have multiple query points, the distance function between an image x and a query Q should aggregate multiple distance components from the image to related query points. VISCORS uses this aggregate distance function:

$$Dist\left(x,Q\right)=\sqrt{\frac{g}{\sum_{j=1}^{g} 1/dist^2\left(x,q_j\right)}} \quad (4)$$

where g is the number of query points in a query Q, q_j is the jth query point of Q, and $dist(x, q_j)$ is a distance function between an image x and a query point q_j. We derived Equation 4 from FALCON's formula.[10] This aggregate distance function supports a disjunctive query that captures high-level semantics of images better than a conjunctive query, especially when preferred images are widely spread out in the feature space (that is, a case of a high level of heterogeneity). This enables faster search of the desired images.

Just as VISCORS continuously refines the query in CBIR, it also updates Equation 4 to reflect the customer's current preference. For this purpose, we define the $dist(x, q_j)$ in Equation 4 as

$$dist\left(x,q_j\right)=\sqrt{\sum_{s=1}^{S} w_s\left(x_s-q_{js}\right)^2} \quad (5)$$

where S is the number of dimensions of the feature space, w_s is a weight of the sth dimension in the feature space, and x_s and q_{js} are coordinates of an image x and a query point q_j on the sth dimension. The relative weight for the sth feature, w_s in Equation 5, is $1/\sigma_s$, where σ_s is a standard deviation of coordinates of the sth dimension of images. We calculate σ_s using all images in the preferred set accumulated during the search session. This distance function update better reflects a customer's current preference by allowing different weights by dimension and emphasizing the features with smaller variance.

Figure 4 shows the three steps of CBIR-based recommendation.

1. Initialization
   a. Set Q to the image selected from the CF list.
   b. Set the preferred set R_p and the accumulated preferred set during the search session R_a to empty sets.
   c. Apply equal weights by setting w_s to 1 for $s = 1, 2, \ldots, S$.
2. Image retrieval
   Retrieve k images, $R = \{r_1, r_2, \ldots, r_k\}$, such that $Dist(r_1, Q)$ is the lowest distance between an image and the query, $Dist(r_2, Q)$ is the next lowest, and so on.
3. Query refinement and distance function update
   a. For each r_j in R
      Recommend r_j to the target customer c.
      If r_j is marked as preferred, add r_j to R_p and R_a.
      Endfor
   b. Refine Q by replacing the images in Q with the ones in R_p.
   c. Update $Dist(x, q_j)$ by recalculating w_s for $s = 1, 2, \ldots, S$ using R_a.
   d. Set R_p to an empty set.
   e. Go to step 2 for the next iteration.

Figure 4. The three steps of CBIR-based recommendation.

Evaluating Viscors

To evaluate the performance of VISCORS, we developed a Web-based application system running on a PC with exactly the same user interface as the cell-phone-based VISCORS system. We wanted to answer two main questions:

- How much performance improvement does VISCORS deliver compared to other recommender systems?
- How does CF's effectiveness affect overall performance?

The experiments

For the experiments, we used the 230 wallpaper images that Korea Telecom Freetel (KTF), a leading Korean CDMA (code division multiple access) carrier, offered at the time of our experiment. To characterize images, VISCORS used three color moments based on HSV (hue, saturation, value), a well-known visual feature. That is, we calculated the mean, standard deviation, and skewness of HSV values of all pixels to represent images as vectors in a 9D feature space.

The experiment involved 200 mobile-Web customers who had previously purchased wallpaper images from KTF. Their past purchase information was stored as an initial customer profile. Each participant had purchased an average of 7.4 images. Before the participants started a search session, we asked them to select a target image to search for. Using the PC Web interface instead of the cell phone interface let participants navigate freely for a target image. A search session continued until the system returned the target image. For each session, we logged information on target images, recommended images, and images marked as preferred or unpreferred for later analysis.

Figure 5 illustrates how a participant finds a target image (the bouquet of roses in Figure 5a) in an actual search session, for $k = 3$. In Figure 5b, the CF module generates a list of recommendations and presents the list's first page. The participant views the first image, determines that it isn't similar to the bouquet image, and decides to return to the CF-generated list.

In Figure 5c, the participant views the second image (of lilies) on the CF list and determines that it's somewhat similar to the target image because it consists entirely of flowers. The participant then initiates further searching using the image as a starting image. In Figure 5d, VISCORS uses the lilies image to retrieve three images with similar visual content from the image database, and presents them in sequence. The participant declares that all three images are unpreferred and returns to the CF-generated list.

In Figure 5e, the participant views the third image on the CF list and determines that the presented image of pinkish flowers is somewhat similar to the target image. The partici-

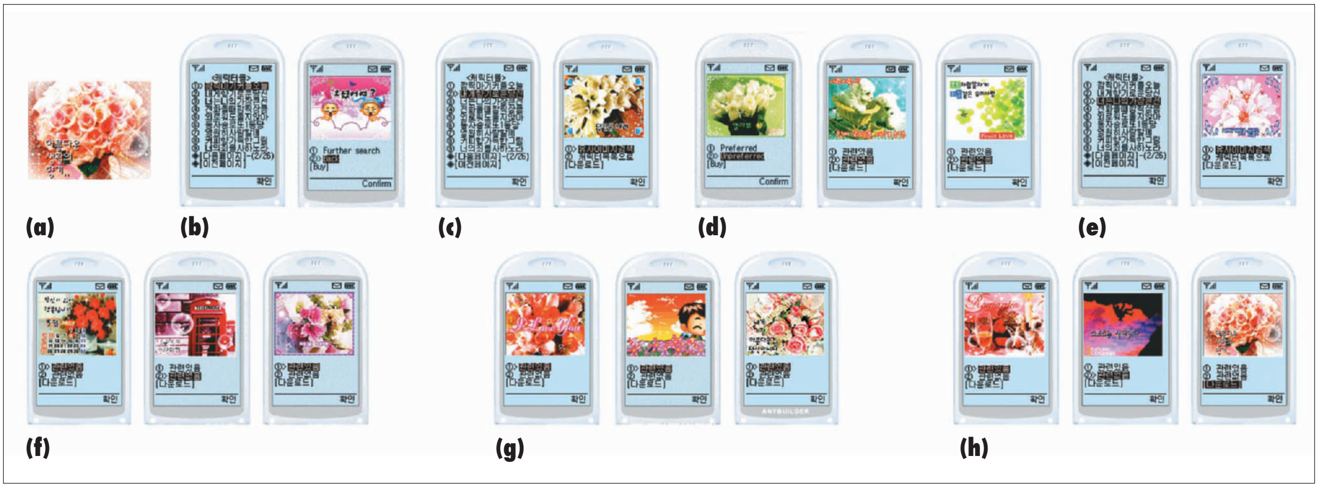


Figure 5. A customer's actual search session (part of some screens are in English for this article): (a) the target image; (b) first CF view; (c) second CF view; (d) first CBIR view; (e) third CF view; (f) second CBIR view; (g) third CBIR view; (h) fourth CBIR view.

pant then initiates further searching using the image as a starting image. In Figure 5f, Viscors uses the image to retrieve three images, which it presents in sequence. The participant declares the first and third as preferred because they're images of flowers in similar colors, and declares the second as unpreferred.

In Figure 5g, Viscors uses the two preferred images to retrieve three similar images, which it presents in sequence. The participant declares them all as preferred. In Figure 5h, Viscors again retrieves and presents three images in the same manner using the three preferred images. The participant declares the first as preferred and the second as unpreferred. The third image is the target image.

For this evaluation, we devised the *views per success* metric, which is the number of images a customer views before he or she purchases an image. Basically, VPS measures the customer's effort for a successful search. We used VPS to compare Viscors to two other recommender systems: a best-seller-based system and a typical CF-based recommender system (pure-CF). The pure-CF procedure is identical to that of CF-based recommendation that Viscors uses, except that it uses only purchase information to build a customer profile.

Each participant performed three search sessions per period for five periods, with a different target image each session. This was to see how system's performance changes over time. For each target image, participants repeated the experiment using the three different recommender systems. In the case of Viscors, participants performed the experiment for $k = 3$, 5, and 7 to see how changes in k affect the overall performance. Because the quality of CF recommendations varies with the neighborhood's size,[3] we performed an initial experiment of a single period to determine the optimal size. A neighborhood size of 30 yielded the best performance, so we used that size in our other experiments.

To analyze the experiment's results, we conducted two statistical tests. One was a t-test comparing the average performance of all three systems. The other was a two-way Anova (*an*alysis *o*f *va*riance) test with repetition to assess how each period and k affected VPS.

Results and discussion

As Table 1 shows, the average VPS of Viscors is about 38 percent lower than pure-CF and 52 percent lower than the best-seller-based system, at a significance level of 1 percent. (That is, Viscors produced performance gains of 38 and 52 percent.) Table 1 also shows that the rate of improvement in VPS over the five periods (that is, the system's learning speed) is 25 percent for Viscors, 18 percent for pure-CF, and 8 percent for the best-seller-based system. These results indicate that Viscors offers not only the lowest VPS but also the fastest performance improvement. Its superior performance over pure-CF stems from its accelerated learning of customer preference from

Table 1. Performance comparison of Viscors and two benchmark systems. VPS (views per success) measures the number of images a customer views before purchasing an image.

System	Performance (VPS)						VPS reduction over	
	Period 1	Period 2	Period 3	Period 4	Period 5	Average	five periods (%)	t value
Viscors (V)	23.01	22.32	19.96	18.57	17.24	20.22	25	
Pure-CF (C)	37.95 (39%)*	33.05 (32%)	31.48 (37%)	31.19 (40%)	31.00 (44%)	32.93 (38%)	18	–106.77‡
Best-seller-based system (B)	43.98 (48%)†	42.64 (48%)	41.12 (51%)	41.89 (56%)	40.66 (58%)	42.06 (52%)	8	-66.49‡

* The performance gain of Viscors over pure-CF is (C – V)/C.
† The performance gain of Viscors over the best-seller-based system is (B – V)/B.
‡ $p < 0.01$. p is the probability that the null hypothesis is true.

the additional preference-rating information fed back from CBIR. This indicates that VISCORS successfully overcomes CF's sparsity problem.

As we mentioned before, search sessions in the experiment lasted until the system found the predetermined target image. However, in a real environment, a search session might end when the system retrieves an image somewhat close to the customer's desired image. This suggests that customers in a real environment would be able to search for images with much less effort than our results indicate.

According to the ANOVA results, the variation in VPS over the five periods (that is, the *period effect*) is significant (F = 292.46, $p < 0.01$). The F statistic provides a test for the statistical significance of the observed VISCORS performance differences over periods. A large value for F indicates that performance varies by period. p is the probability that the null hypothesis—in this case, that the performance of VISCORS doesn't vary by period—is true. Figure 6 illustrates the variation as a decreasing curve. As periods progress, more rating information becomes available. When the customer profile contains more ratings, neighborhood formation becomes more accurate, thereby improving the quality of CF recommendations.

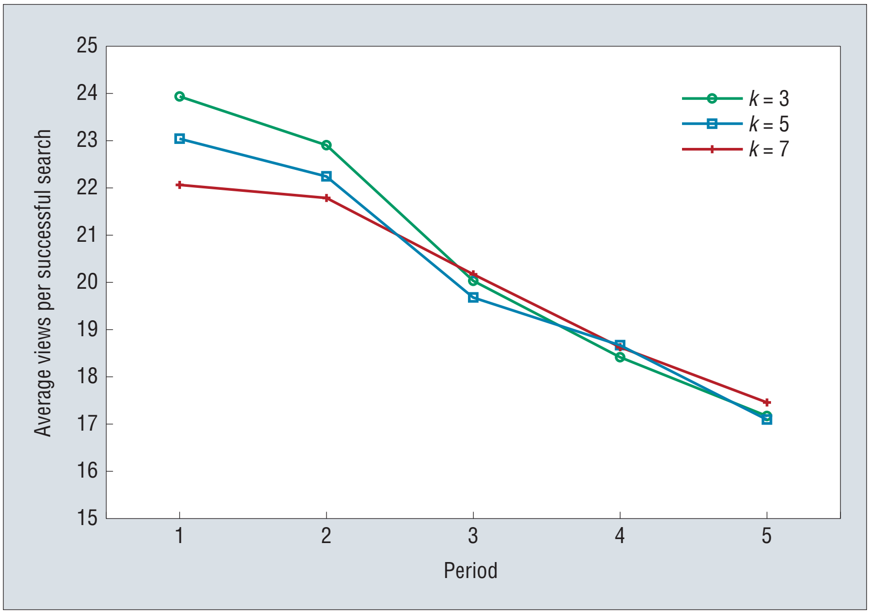


Figure 6. The overall performance of VISCORS.

The results in Table 2 show that better CF recommendations help decrease VPS in three ways. First, the images on the CF list are used more often as an initial query for CBIR because the CF list's upper part contains more preferred images. As Part A of Table 2 shows, the average rate of viewed images on the CF list's first page (that is, the top nine recommended images) that become an initial query of CBIR increases from 0.33 in Period 1 to 0.39 in Period 5.

Second, the number of iterations of CBIR per success decreases because the first image used in CBIR is closer to the target image in the feature space. As Part B of Table 2 shows, the average number of iterations of CBIR per success decreases from 4.23 in Period 1 to 3.66 in Period 5.

Third, the VPS that CBIR consumes decreases because the likelihood increases that the target image is in the CF list's upper part. As Part C of Table 2 shows, the rate of the target image being on the CF list's first page increases from 0.18 in Period 1 to 0.28 in Period 5.

The two-way ANOVA results also indicate that the variation in VPS due to k (that is, the *k effect*) is significant ($F = 4.52$, $p < 0.05$). In addition, the interaction between the period and k significantly affects performance (F = 4.09, $p < 0.01$). Figure 6 illustrates this as the

Table 2. Collaborative filtering's effect over five periods. *k* is the number of retrieved images per iteration of content-based image retrieval.

		Period				
Effect	***k***	**1**	**2**	**3**	**4**	**5**
A. The rate of viewed images on the CF list's first page that become an initial query of content-based image retrieval	3	0.33	0.34	0.37	0.35	0.37
	5	0.31	0.34	0.36	0.37	0.40
	7	0.35	0.31	0.36	0.37	0.41
	Average	0.33	0.33	0.36	0.36	0.39
B. Iterations of CBIR per success	3	6.20	5.96	5.37	5.30	5.08
	5	3.81	3.67	3.45	3.50	3.40
	7	2.69	2.60	2.54	2.53	2.51
	Average	4.23	4.08	3.79	3.78	3.66
C. The rate of the target image being found on the CF list's first page	3	0.18	0.20	0.24	0.25	0.28
	5	0.18	0.21	0.24	0.24	0.28
	7	0.18	0.20	0.23	0.25	0.27
	Average	0.18	0.20	0.24	0.25	0.28
D. The ratio of the image views that CF consumed to the total views per success	3	0.56	0.56	0.55	0.54	0.53
	5	0.48	0.50	0.48	0.47	0.45
	7	0.45	0.47	0.45	0.44	0.42
	Average	0.50	0.51	0.49	0.48	0.47

difference in VPS for the three values of k in the early periods and as the diminishing difference in VPS over all five periods.

As Part D of Table 2 shows, CF's relative importance to overall performance decreases as k increases (for example, from 0.56 when $k = 3$ to 0.45 when $k = 7$ in Period 1). As we mentioned previously, CF's effectiveness also improves as the periods progress. So, a performance with a lower k should improve faster than that with a higher k. This means that the k effect is obvious when CF's performance is poor and gradually diminishes as CF improves. An additional one-way ANOVA analysis confirms the claim with its results of ($F = 13.54$, $p < 0.01$) on the k effect for the first two periods and ($F = 1.10$, $p = 0.33$) for the last three periods.

From these results, we conclude that VISCORS is a viable solution to the problems encountered when customers download wallpaper images on the mobile Web. However, the results also suggest two aspects of VISCORS to consider for its real-world application.

First, the most important factors influencing CF's effectiveness are the customer profile's sparsity level and heterogeneity in customers' buying behaviors.[3,5] In cases with a high level of sparsity or heterogeneity, CF performs poorly, so we recommend using a higher value of k. In other cases, because k's impact on performance is insignificant, you can select k's value by considering other factors such as the customer's burden of feedback in CBIR or the service provider's marketing strategy concerning the diversity of recommended images.

Second, our experiment used a less heterogeneous image database consisting of a few images belonging to a small number of categories (for example, love, friendship, the bizarre, and so on). So, we used only the color moment for image characterization even though we knew that CBIR's performance improves as it considers more visual features. However, using a single feature, as our experiment did, might not give such positive results in real-world applications, where the image database's heterogeneity will likely be higher. Therefore, in a real-world implementation of VISCORS, we recommend expanding the feature set to include other widely used features such as texture and shape.

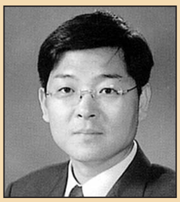
VISCORS offers two main benefits. First, customers can purchase content with much less search effort and much lower connection time because they can much more easily find the desired content. Second, mobile-Web content providers can improve their businesses' profitability because lower customer frustration in finding desired content increases revenue through an improved purchase conversion rate. (This rate constitutes the number of search sessions that end with a purchase divided by the total number of search sessions.)

As mobile-Web services rapidly grow, mobile-Web-based recommender systems for other types of multimedia content, such as music on demand or video on demand, will increasingly be an area of research interest. Our approach is applicable to these types of content as long as you can represent the content as a vector in the feature space, as in the case of wallpaper images. However, users can easily provide preference feedback on wallpaper images after a short viewing; this might

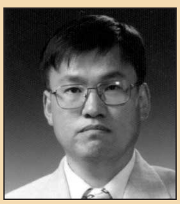
The Authors

Chan Young Kim is an associate professor of Internet business at Dongyang Technical College and is a PhD candidate in the management information systems program at the Korea Advanced Institute of Science and Technology. His research interests include recommender systems, e-business, and mobile business. He received his BS in electrical engineering from Seoul National University and his MBA from the University of Chicago. Contact him at the Dept. of Internet Business, Dongyang Technical College, 62-160 Kochuk, Kuro, Seoul, 152-714, Korea; cykim@dongyang.ac.kr.

Jae Kyu Lee is a professor at the Korea Advanced Institute of Science and Technology's Graduate School of Management and a director of the International Center for Electronic Commerce. His research interests include e-commerce, the Semantic Web, recommender systems, data mining, and expert systems. He's the editor in chief of *Electronic Commerce Research and Applications*. He received his PhD from the Wharton School, University of Pennsylvania. Contact him at the Graduate School of Management, Korea Advanced Institute of Science and Technology, 207-43 Cheongryangri, Dongdaemun, Seoul, 130-012, Korea; jklee@kgsm.kaist.ac.kr.



Yoon Ho Cho is an assistant professor at Kookmin University's School of E-Business. His research interests include recommender systems, e-business, mobile business, customer relationship management, and data mining. He received his PhD in management information systems from the Korea Advanced Institute of Science and Technology. Contact him at the School of E-Business, Kookmin Univ., 861-1 Jungnung, Sungbuk, Seoul, 136-702, Korea; www4u@kookmin.ac.kr.



Deok Hwan Kim is an associate professor of mobile Internet at Dongyang Technical College. He's now visiting the University of Arizona's Computer Science Department. His research interests include multimedia information retrieval, multimedia databases, recommender systems, and data mining. He received his PhD in information and communication engineering from the Korea Advanced Institute of Science and Technology. Contact him at the Dept. of Mobile Internet, Dongyang Technical College, 62-160 Kochuk, Kuro, Seoul, 152-714, Korea; dhkim@dongyang.ac.kr.

not be true for music or videos. So, successful application of our approach to these types of content will require research on the proper interfaces for preference feedback.

Acknowledgments

The Korea Science & Engineering Foundation's Postdoctoral Fellowship Program partially supported this work.

References

1. *Korea Internet White Paper 2002*, Korea Network Information Center, 2002.
2. U. Shardanand and P. Maes, "Social Information Filtering: Algorithms for Automating 'Word of Mouth,'" *Proc. 1995 ACM/SIGCHI Conf. Human Factors in Computing Systems* (CHI 95), Addison-Wesley, 1995, pp. 210–217.
3. B. Sarwar et al., "Analysis of Recommendation Algorithms for E-Commerce," *Proc. ACM Conf. Electronic Commerce* (EC 00), ACM Press, 2000, pp. 158–167.
4. M. Balabanovic and Y. Shoham, "Fab: Content-Based Collaborative Recommendation," *Comm. ACM*, vol. 40, no. 3, 1997, pp. 66–72.
5. Y.H. Cho, J.K. Kim, and S.H. Kim, "A Personalized Recommender System Based on Web Usage Mining and Decision Tree Induction," *Expert Systems with Applications*, vol. 23, no. 3, 2002, pp. 329–342.
6. P. Melville, R.J. Mooney, and R. Nagarajan, "Content-Boosted Collaborative Filtering for Improved Recommendations," *Proc. 18th Nat'l Conf. Artificial Intelligence* (AAAI 02), AAAI Press, 2002, pp. 187–192.
7. X.S. Zhou and T.S. Huang, "Relevance Feedback for Image Retrieval: A Comprehensive Review," *ACM Multimedia Systems J.*, vol. 8, no. 6, 2003, pp. 536–544.
8. D.H. Kim and C.W. Chung, "Qcluster: Relevance Feedback Using Adaptive Clustering for Content-Based Image Retrieval," *Proc. 2003 ACM SIGMOD Int'l Conf. Management of Data* (SIGMOD 03), ACM Press, 2003, pp. 599–610.
9. K. Porkaew, K. Chakrabarti, and S. Mehrotra, "Query Refinement for Multimedia Similarity Retrieval in MARS," *Proc. 7th ACM Int'l Multimedia Conf.* (MM 99), Part 1, ACM Press, 1999, pp. 235–238.
10. L. Wu et al., "FALCON: Feedback Adaptive Loop for Content-Based Retrieval," *Proc. 26th Int'l Conf. Very Large Data Bases* (VLDB 00), Morgan Kaufmann, 2000, pp. 297–306.