Supplementary Material of Face Video Retrieval with Image Query via Hashing across Euclidean Space and Riemannian Manifold

Yan Li[1,2], Ruiping Wang[1], Zhiwu Huang[1,2], Shiguang Shan[1], Xilin Chen[1,3]
[1]Key Lab of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS, Beijing, 100190, China
[2]University of Chinese Academy of Sciences, Beijing, 100049, China
[3]Department of Computer Science and Engineering, University of Oulu, Oulu 90570, Finland
{yan.li, zhiwu.huang}@vipl.ict.ac.cn, {wangruiping, sgshan, xlchen}@ict.ac.cn

This material is the supplementary document to the CVPR 2015 submission #34 and provides additional experimental results that support the method proposed in the main paper. The rest of this document is organized as follows: Section 1 shows the evaluation of the proposed method on a surveillance video database. Section 2 compares the proposed method with two classical key-frame extraction based video classification methods. Then Section 3 gives the complete experimental results of both retrieval scenarios, i.e., image query vs. video database, and video query vs. image database. Section 4 shows the evaluation of different initialization choices of Algorithm 1 in the main paper. Section 5 gives the results on algorithm convergence. Character distributions of the two TV-Series are shown in Section 6, and followed by more comparisons of our method with the state-of-the-art multiple modalities hash learning methods in Section 7.

1. Evaluation on Surveillance Video Database

To further evaluate the proposed method, we conduct retrieval experiment on a surveillance video database, i.e., COX-S2V [4], which is a public still and video face database containing 1,000 subjects. In COX-S2V, still images are captured with SLR, and videos are captured by video cameras located at different positions. Faces in COX-S2V contain lots of variations, e.g., illumination, head pose. In this experiment, we randomly select 300 subjects as training (one still image and three videos for each subject), and use the rest 700 subjects as testing (one still image of each subject for query, and three videos per subject for database). Here only the performances of HER and the second best method under 128 bits are listed: HER 0.3952, SITQ (SMH) 0.3241, MM-NN (MMH) 0.3087.

2. Comparison with Key-frame Extraction Methods

We also carefully implemented two classical key-frame extraction methods, i.e., clustering based method [18] and saliency based method [9], and fixed the back-end hash learning part as the best baseline SITQ [2]. 0.4810, 0.4712 are achieved (BBT/I2V/128b) respectively, which are very close to our current SITQ baseline (0.4799 in Table 1). That is because: a) our current implementation can be viewed as an over-complete key-frame extraction method; b) different from general videos (e.g., sports) always with huge visual variance, face videos have no that explicit definition of "key-face", and all faces in the video play similar importance. Though it is not easy to say whether covariance is superior to key-frame for general videos, we believe for face videos the answer is YES.

3. Complete Experimental Results of Two Retrieval Scenarios

As the space limitation, we only gave the experimental result of one retrieval scenario, i.e., using image query to retrieve video database. Here we give the complete experimental results of both scenarios. The results can be found in Table 1 and Table 2, respectively corresponding to comparison with single modality hashing and multiple modalities hashing.

978-1-4673-6964-0/15/$31.00 ©2015 IEEE

Table 1: Comparison with the state-of-the-art single modality hash learning methods with mAP on two databases. K means the length of hash code.

Task	Method	*the Big Bang Theory*					*Buffy the Vampire Slayer*				
		K = 8	K = 16	K = 32	K = 64	K = 128	K = 8	K = 16	K = 32	K = 64	K = 128
Image Query vs. Video Database	LSH [5]	0.1977	0.2086	0.2092	0.1963	0.1994	0.1532	0.1508	0.1517	0.1568	0.1578
	SH [14]	0.2617	0.2652	0.2665	0.2623	0.2673	0.1832	0.2046	0.2237	0.2177	0.2222
	ITQ [2]	0.2798	0.3025	0.2989	0.3029	0.3060	0.1693	0.1848	0.1972	0.2265	0.2457
	SSH [13]	0.3209	0.2855	0.2662	0.2584	0.2586	0.2262	0.2193	0.2202	0.2141	0.2120
	DBC [11]	0.4391	0.4495	0.4235	0.4005	0.3867	0.2965	0.3858	0.4460	0.4707	0.4547
	KSH [7]	0.4059	0.4366	0.4454	0.4567	0.4604	**0.3481**	0.3542	0.4149	0.4385	0.4517
	SITQ [2]	0.3442	0.3909	0.4298	0.4576	0.4799	0.3345	**0.3869**	0.4580	0.4738	0.4990
	HER	**0.4626**	**0.5049**	**0.5227**	**0.5490**	**0.5539**	0.3195	0.3770	**0.4852**	**0.5281**	**0.5877**
Video Query vs. Image Database	LSH [5]	0.2016	0.2089	0.1983	0.2108	0.2087	0.1500	0.1498	0.1523	0.1492	0.1486
	SH [14]	0.2499	0.2656	0.2666	0.2650	0.2659	0.1647	0.1738	0.1783	0.1772	0.1729
	ITQ [2]	0.2804	0.3012	0.3020	0.3106	0.3151	0.1547	0.1609	0.1659	0.1747	0.1866
	SSH [13]	0.2696	0.2638	0.2591	0.2560	0.2548	0.1738	0.1737	0.1725	0.1728	0.1736
	DBC [11]	0.3609	0.4030	0.3901	0.3785	0.3724	0.2020	0.2290	0.2542	0.2772	0.2729
	KSH [7]	0.3303	0.3626	0.3859	0.3930	0.3947	0.1911	0.1941	0.2324	0.2370	0.2362
	SITQ [2]	0.3154	0.3630	0.4004	0.4323	**0.4593**	0.2004	0.2196	0.2453	0.2609	0.2722
	HER	**0.3743**	**0.4080**	**0.4125**	**0.4451**	0.4476	**0.2262**	**0.2571**	**0.2932**	**0.3180**	**0.3414**

Table 2: Comparison with the state-of-the-art multiple modalities hash learning methods with mAP on two databases. K means the length of hash code.

Task	Method	*the Big Bang Theory*					*Buffy the Vampire Slayer*				
		K = 8	K = 16	K = 32	K = 64	K = 128	K = 8	K = 16	K = 32	K = 64	K = 128
Image Query vs. Video Database	CMSSH [1]	0.2109	0.2047	0.2143	0.2024	0.2478	0.1504	0.1569	0.1559	0.1593	0.1688
	CVH [6]	0.2085	0.2110	0.2092	0.2231	0.2407	0.1566	0.1579	0.1570	0.1644	0.1900
	PLMH [16]	0.2387	0.2447	0.2461	0.2487	0.2608	0.1847	0.1859	0.1800	0.1828	0.1853
	PDH [10]	0.2998	0.2949	0.2903	0.3095	0.2916	0.1698	0.1769	0.1865	0.1846	0.1980
	MLBE [17]	0.3214	0.2600	0.2648	0.3917	0.3858	0.1123	0.1550	0.1720	0.1759	0.1840
	MM-NN [8]	0.3263	0.3955	0.4664	0.5124	0.4922	0.2207	0.2207	0.2681	0.3671	0.4045
	HER	**0.4626**	**0.5049**	**0.5227**	**0.5490**	**0.5539**	**0.3195**	**0.3770**	**0.4852**	**0.5281**	**0.5877**
Video Query vs. Image Database	CMSSH [1]	0.2002	0.1953	0.1966	0.1996	0.2152	0.1555	0.1559	0.1608	0.1664	0.1645
	CVH [6]	0.2080	0.2044	0.2070	0.2182	0.2377	0.1497	0.1502	0.1527	0.1551	0.1621
	PLMH [16]	0.2287	0.2318	0.2330	0.2391	0.2479	0.1577	0.1559	0.1558	0.1587	0.1618
	PDH [10]	0.2630	0.2661	0.2600	0.2672	0.2692	0.1612	0.1657	0.1676	0.1706	0.1736
	MLBE [17]	0.3222	0.2467	0.2408	0.3991	0.3656	0.1288	0.1379	0.1848	0.1582	0.1883
	MM-NN [8]	0.2567	0.3302	0.4090	0.3941	0.4077	0.2001	0.2001	0.2081	0.2423	0.2600
	HER	**0.3743**	**0.4080**	**0.4125**	**0.4451**	**0.4476**	**0.2262**	**0.2571**	**0.2932**	**0.3180**	**0.3414**

4. Initialization Effects

As mentioned in the main paper, our method is a general framework for heterogeneous hash learning. Any one of the Generalized Multiview Analysis (GMA) [12] methods is competent for the initialization of Algorithm 1 in the main paper. Here we give a comparison of two representative initialization choices, i.e., Kernelized Canonical Correlation Analysis (KC-CA) [3] and Kernelized Generalized Multiview Marginal Fisher Analysis (KGMMFA) [12]. We choose these two because CCA [3] is a classical multi-view learning method, and MFA [15] is a general and state-of-the-art framework for multi-view learning proposed most recently. The comparison is shown in Table 3, and it is easy to observe that KGMMFA shows relatively better results compared with KCCA in most test cases. This is mainly because KGMMFA utilizes more discriminant information compared with KCCA in which only side information is used, and this superiority gets more significant as the length of hash code increases.

5. Algorithm Convergence

While it is hard to find the global minimum of the objective function, usually in practice a couple of iterations can lead to good hash codes which are capable of yielding desirable results. To evaluate the convergence, average Hamming distances of intra- and inter-category pairs of each modality (i.e., image and video) in every iteration are shown in Fig. 1 (without loss of generality, we fix the test database and hash code length to BBT and 128, respectively). Usually, we iterate 2 or 3 times to reach the optimization of the Algorithm in practice.

Table 3: Comparison of different initialization methods, i.e., KCCA and KGMMFA, on BBT and BVS. K means the length of hash code.

Task	Initialization Method	Code Length				
		K = 8	K = 16	K = 32	K = 64	K = 128
Image Query vs. Video Database on BBT	KCCA	0.4131	0.3860	0.4646	0.4640	0.4546
	KGMMFA	0.4626	0.5049	0.5227	0.5490	0.5539
Video Query vs. Image Database on BBT	KCCA	0.3307	0.3148	0.3657	0.3677	0.3622
	KGMMFA	0.3743	0.4080	0.4125	0.4451	0.4476
Image Query vs. Video Database on BVS	KCCA	0.3836	0.4002	0.4767	0.5018	0.4966
	KGMMFA	0.3195	0.3770	0.4852	0.5281	0.5877
Video Query vs. Image Database on BVS	KCCA	0.2480	0.2619	0.2914	0.3038	0.2953
	KGMMFA	0.2262	0.2571	0.2932	0.3180	0.3414

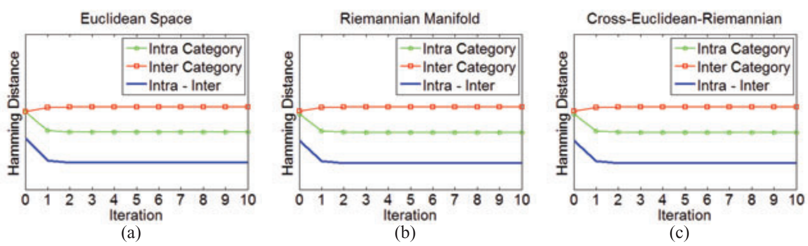


Figure 1: Average Hamming distances of intra- and inter-category pairs of each modality in every iteration. In fact, the blue lines in (a), (b), and (c) correspond to the terms E_e, E_r, and E_{er} of Equ. (3), Equ. (4), and Equ. (5) respectively in the main paper, and the green (red) lines in (a), (b), and (c) correspond to the first (second) terms of Equ. (3), Equ. (4), and Equ. (5), respectively.

6. Character Distribution

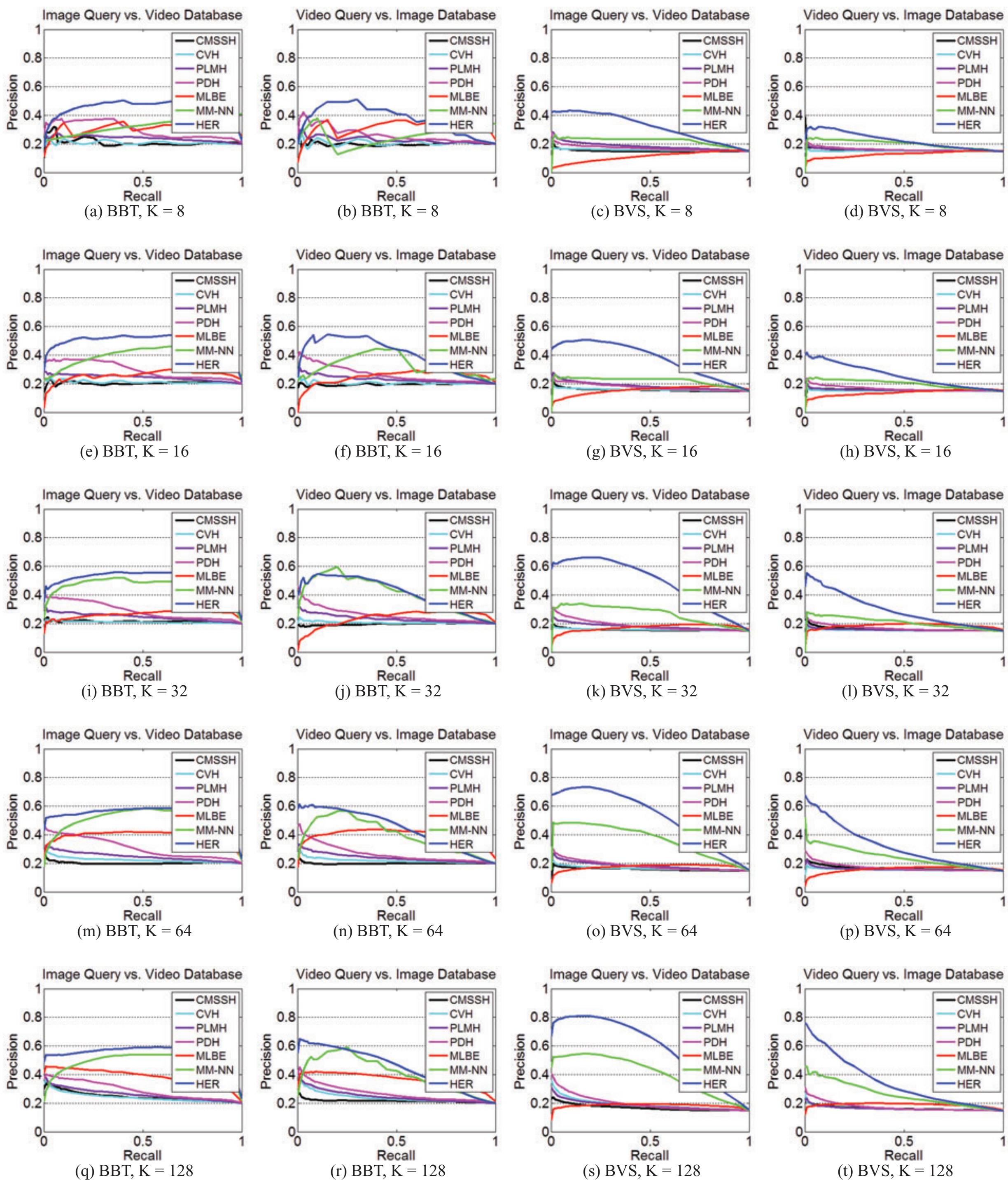
The first TV-Series database consists of face videos of the first 6 episodes from season 1 of the Big Bang Theory (BBT), and the second one consists of face videos of the first 6 episodes from season 5 of Buffy the Vampire Slayer (BVS). Table 4 shows the distributions of face videos of all characters in BBT and BVS. Specifically, there are 3341 face videos of 12 characters and 4779 face videos of 29 characters in BBT and BVS, respectively, where extras (usually appear in the background) are labeled as "Unknown".

Table 4: Distributions of face videos of all characters in BBT and BVS.

			the Big Bang Theory				
Character	Doug	Gabelhauser	Howard	Kurt	Leonard	Leslie	Mary
Video Num	8	15	263	30	932	78	88
Character	Penny	Raj	Sheldon	Summer	Unknown		
Video Num	474	249	860	4	340		
			Buffy the Vampire Slayer				
Character	__None__	Anya	Ben	Beth	Buffy	BuffyDoll	Dawn
Video Num	7	249	18	51	1102	2	304
Character	Donny	Dracula	Giles	Glory	Graham	Harmony	Joyce
Video Num	31	63	286	66	39	172	89
Character	Leiach	Maclay	Manager	Mort	Overheiser	Riley	Sandy
Video Num	17	51	26	30	32	464	10
Character	Spike	Tara	Toth	watchman	Willow	Xander	Xander2
Video Num	175	236	1	9	438	442	109
Character	Unknown						
Video Num	261						

7. More Comparison with State-of-the-art Multiple Modalities Hash Learning Methods

Fig. 2 shows the comparisons of our method with the state-of-the-art multiple modalities hash learning methods in BBT and BVS, respectively. Compared with the main paper, more hash code lengths are listed.



(a) BBT, K = 8 (b) BBT, K = 8 (c) BVS, K = 8 (d) BVS, K = 8

(e) BBT, K = 16 (f) BBT, K = 16 (g) BVS, K = 16 (h) BVS, K = 16

(i) BBT, K = 32 (j) BBT, K = 32 (k) BVS, K = 32 (l) BVS, K = 32

(m) BBT, K = 64 (n) BBT, K = 64 (o) BVS, K = 64 (p) BVS, K = 64

(q) BBT, K = 128 (r) BBT, K = 128 (s) BVS, K = 128 (t) BVS, K = 128

Figure 2: Comparison with the state-of-the-art multiple modalities hash learning methods with precision recall curves on two databases. K means the length of hash code.

References

[1] M. Bronstein, A. Bronstein, F. Michel, and N. Paragios. Data fusion through cross-modality metric learning using similarity-sensitive hashing. In *CVPR*, pages 3594–3601. IEEE, 2010. 2

[2] Y. Gong and S. Lazebnik. Iterative quantization: A procrustean approach to learning binary codes. In *CVPR*, pages 817–824. IEEE, 2011. 1, 2

[3] D. Hardoon, S. Szedmak, and J. Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural computation*, 16(12):2639–2664, 2004. 2

[4] Z. Huang, S. Shan, H. Zhang, S. Lao, A. Kuerban, and X. Chen. Benchmarking still-to-video face recognition via partial and local linear discriminant analysis on cox-s2v dataset. In *ACCV*, pages 589–600. Springer, 2012. 1

[5] P. Indyk and R. Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *ACM Symposium on Theory of Computing*, pages 604–613. ACM, 1998. 2

[6] S. Kumar and R. Udupa. Learning hash functions for cross-view similarity search. In *IJCAI*, pages 1360–1365. AAAI Press, 2011. 2

[7] W. Liu, J. Wang, R. Ji, Y.-G. Jiang, and S.-F. Chang. Supervised hashing with kernels. In *CVPR*, pages 2074–2081. IEEE, 2012. 2

[8] J. Masci, M. Bronstein, A. Bronstein, and J. Schmidhuber. Multimodal similarity-preserving hashing. *PAMI*, 2013. 2

[9] E. Mendi and C. Bayrak. Shot boundary detection and key frame extraction using salient region detection and structural similarity. In *Proceedings of the 48th Annual Southeast Regional Conference*, page 66. ACM, 2010. 1

[10] M. Rastegari, J. Choi, S. Fakhraei, D. Hal, and L. Davis. Predictable dual-view hashing. In *ICML*, pages 1328–1336, 2013. 2

[11] M. Rastegari, A. Farhadi, and D. Forsyth. Attribute discovery via predictable discriminative binary codes. In *ECCV*, pages 876–889. Springer, 2012. 2

[12] A. Sharma, A. Kumar, H. Daume, and D. Jacobs. Generalized multiview analysis: A discriminative latent space. In *CVPR*, pages 2160–2167. IEEE, 2012. 2

[13] J. Wang, S. Kumar, and S.-F. Chang. Semi-supervised hashing for scalable image retrieval. In *CVPR*, pages 3424–3431. IEEE, 2010. 2

[14] Y. Weiss, A. Torralba, and R. Fergus. Spectral hashing. In *NIPS*, pages 1753–1760, 2008. 2

[15] S. Yan, D. Xu, B. Zhang, H.-J. Zhang, Q. Yang, and S. Lin. Graph embedding and extensions: a general framework for dimensionality reduction. *PAMI*, 29(1):40–51, 2007. 2

[16] D. Zhai, H. Chang, Y. Zhen, X. Liu, X. Chen, and W. Gao. Parametric local multimodal hashing for cross-view similarity search. In *IJCAI*, pages 2754–2760. AAAI Press, 2013. 2

[17] Y. Zhen and D.-Y. Yeung. A probabilistic model for multimodal hash function learning. In *KDD*, pages 940–948. ACM, 2012. 2

[18] Y. Zhuang, Y. Rui, T. Huang, and S. Mehrotra. Adaptive key frame extraction using unsupervised clustering. In *ICIP*, volume 1, pages 866–870. IEEE, 1998. 1